Model-based compressive sensing with Earth Mover's Distance constraints

by

Ludwig Schmidt

B.A., University of Cambridge (2011)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

Author .....
Department of Electrical Engineering and Computer Science
May 22, 2013

Certified by .....
Piotr Indyk
Professor of Computer Science and Engineering
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering
Chairman, Department Committee on Graduate Students

Model-based compressive sensing with Earth Mover's Distance constraints

by

Ludwig Schmidt

Submitted to the Department of Electrical Engineering and Computer Science
on May 22, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

In compressive sensing, we want to recover a k-sparse signal $x \in \mathbb{R}^n$ from linear measurements of the form $y = \Phi x$, where $\Phi \in \mathbb{R}^{m \times n}$ describes the measurement process. Standard results in compressive sensing show that it is possible to exactly recover the signal x from only $m = O(k \log \frac{n}{k})$ measurements for certain types of matrices. Model-based compressive sensing reduces the number of measurements even further by limiting the supports of x to a subset of the $\binom{n}{k}$ possible supports. Such a family of supports is called a structured sparsity model.

In this thesis, we introduce a structured sparsity model for two-dimensional signals that have similar support in neighboring columns. We quantify the change in support between neighboring columns with the Earth Mover's Distance (EMD), which measures both how many elements of the support change and how far the supported elements move. We prove that for a reasonable limit on the EMD between adjacent columns, we can recover signals in our model from only $O(k \log \log \frac{k}{w})$ measurements, where w is the width of the signal. This is an asymptotic improvement over the $O(k \log \frac{n}{k})$ bound in standard compressive sensing.

While developing the algorithmic tools for our proposed structured sparsity model, we also extend the model-based compressed sensing framework. In order to use a structured sparsity model in compressive sensing, we need a model projection algorithm that, given an arbitrary signal x, returns the best approximation in the model. We relax this constraint and develop a variant of IHT, an existing sparse recovery algorithm, that works with approximate model projection algorithms.

Thesis Supervisor: Piotr Indyk
Title: Professor of Computer Science and Engineering

Acknowledgments

I want to thank my advisor Piotr Indyk for much helpful advice and many insightful comments while working on the research problems that lead to this thesis. I also want to thank my coauthor Chinmay Hegde for his insights and teaching me about compressive sensing. All results in this thesis are based on work with Chin and Piotr.

Moreover, I would like to thank my friends in the theory group and all of MIT for making this place such a great environment.

Contents

List of Figures

Chapter 1

Introduction

Sensing signals is a cornerstone of many modern technologies such as digital photography, medical imaging, wireless networks and radar. While some settings allow the efficient acquisition of large signals (e.g. CCDs for visible light), physical constraints make individual measurements in other settings much more expensive (e.g. nuclear magnetic resonance in MRI). Moreover, after acquiring a signal, it is often possible to significantly reduce its size by utilizing known structure in the signal. *Compressive sensing* combines these two insights and offers a framework for acquiring structured signals with few measurements. In compressive sensing, the number of measurements depends almost exclusively on the inherent complexity of the signal and not on its size.

Model-based compressive sensing extends this core idea of compressive sensing by utilizing more knowledge of the signal – a so called "model" – in order to reconstruct the signal from even fewer measurements. In this thesis, we introduce a particular signal model inspired by seismic processing and develop the necessary algorithms for using our model in compressive sensing. In the process, we extend the model-based compressive sensing framework to a wider range of recovery algorithms.

1.1 Compressive sensing

A traditional signal processing system often proceeds in two basic steps. First, the system *senses* the signal by taking coordinate-wise measurements in one or multiple dimensions. Second, the signal is *compressed* by utilizing known structure in the signal. *Compressive sensing* [8, 4] combines these two steps in order to reduce the number of measurements. The idea is to take few measurements that capture information about the entire signal in a compressive way. Afterwards, the signal can be reconstructed from the measurements by utilizing its known structure.

More formally, let $x \in \mathbb{R}^n$ be the n-dimensional *signal vector*. We represent the known structure in x with the constraint that x is k-sparse, i.e. only k components of x are non-zero. A linear measurement process can be described by a *measurement matrix* $\Phi \in \mathbb{R}^{m \times n}$. Each row of the matrix Φ corresponds to one linear measurement of the signal vector x. The result of the measurement process is the *measurement vector* $y \in \mathbb{R}^m$:

$$y = \Phi x \,. \tag{1.1}$$

We want to find a measurement matrix Φ with a small number of rows that still allows efficient reconstruction of the signal x from the measurements y. A useful property for measurement matrices is the *restricted isometry property* (RIP) [5]. A measurement matrix Φ satisfies the (δ, k)-RIP if the following two inequalities hold for all k-sparse vectors x:

$$(1 - \delta)\|x\|_2^2 \leq \|\Phi x\| \leq (1 + \delta)\|x\|_2^2, \tag{1.2}$$

where $\|x\|_p$ denotes the ℓ_p-norm. Matrices with i.i.d. Gaussian or Rademacher (± 1) entries satisfy the RIP with only $\Theta(k \log(n/k))$ rows [6].

Given a measurement vector y and a measurement matrix Φ satisfying the RIP, we can reconstruct the original k-sparse signal x by solving the following optimization problem:

$$\arg\min_x \|x\|_1 \quad \text{subject to} \quad \Phi x = y \tag{1.3}$$

This problem is known as *basis pursuit* and can be solved efficiently by converting it to a linear program [5]. If there is a k-sparse x satisfying $\Phi x = y$, the problem (1.3) is guaranteed to recover such an x.

This short introduction only scratches the surface of compressive sensing. The theory extends to settings with measurement noise and to approximately sparse signals [3]. Much more is known about measurement matrices (e.g. partial Fourier matrices [6] and sparse matrices [9]) and there are numerous sparse recovery algorithms [15]. Since the formulation as a sparse linear inverse problem is very general, compressive sensing has spread to a wide range of areas beyond the simple signal processing model outlined above.

1.2 Model-based compressive sensing

Standard compressive sensing requires that the signal vector x is sparse. In some applications, we know more about the structure of the signal. One example is the wavelet decomposition of a natural image, where the large coefficients form a tree-like structure. *Model-based compressive sensing* [1] introduces more sophisticated sparsity models to compressive sensing. The goal of this approach is to reduce the number of measurements necessary for successful reconstruction of the signal.

Note that the set of k-sparse signals can be defined as the union of k-dimensional subspaces. Model-based compressive sensing adds more structure to signals by restricting the set of such subspaces. More formally, we define a *structured sparsity model* $\mathcal{M}_k$, given by m_k k-dimensional subspaces, as follows:

$$\mathcal{M}_k = \bigcup_{i=1}^{m_k} \{x \,|\, \mathrm{supp}(x) \subseteq \Omega_i\}, \tag{1.4}$$

where $\Omega_i \subseteq \{1, \ldots, n\}$ is the index set of non-zero components in the subspace i and $|\Omega_i| \leq k$ for all i. The support of a vector x, $\mathrm{supp}(x)$, is the index set of non-zero components.

It can be shown that standard measurement matrices (e.g. i.i.d. Gaussian) satisfy the counterpart of the restricted isometry property in model-based compressive sensing. For

reconstruction, we can modify CoSaMP [12] and IHT [2], two popular sparse recovery algorithms, to work with structured sparsity models. These modified algorithms use projection oracles $\mathbb{P}(x, k)$ that find the best approximation of an arbitrary signal x in the model $\mathcal{M}_k$:

$$\mathbb{P}(x, k) = \arg\min_{x' \in \mathcal{M}_k} \|x - x'\|_2. \tag{1.5}$$

The algorithm used for implementing the projection oracle depends on the specific model $\mathcal{M}_k$.

1.3 Seismic processing

Seismology is the study of waves propagating through the earth. The sources of seismic waves can be natural events like earthquakes or controlled man-made devices, e.g. trucks shaking the ground. Since the propagation of seismic waves depends on the subsurface, seismology allows us to learn about the interior structure of the earth. The investigated scales can range from a few hundred meters to the entire earth.

In *reflection seismology*, we have a controlled source and record the reflections of the seismic wave with an array of receivers (geophones) at the surface. Since the seismic waves are reflected at boundaries between different types of rocks, we can infer properties of the subsurface from the recorded data. Reflection seismology is the primary tool used in hydrocarbon exploration. Figure 1-1 illustrates a simple seismic experiment.

A seismic experiment with a single source is called a *shot* and the result of such an experiment is a *shot record*. A shot record contains the amplitudes recorded at each receiver during the experiment. Typical sampling rates are between 1 ms and 10 ms, with a total duration of several seconds. In order to gather more information about the subsurface, this process is usually repeated several times with different source locations. Figure 1-2 shows a simple subsurface model with two corresponding shot records.

Due to the underlying physical mechanisms, shot records typically exhibit more structure than mere sparsity. In particular, the signals received by neighboring sensors are similar.

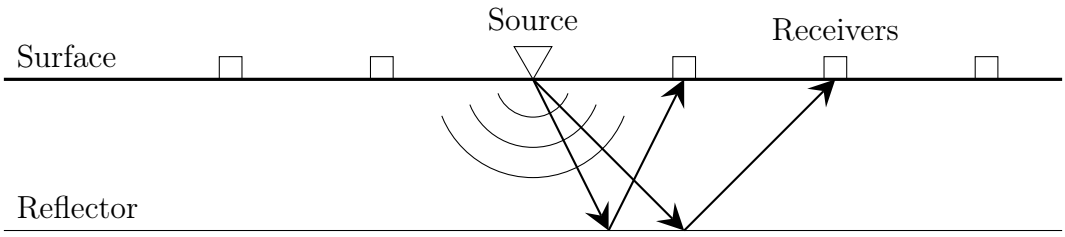
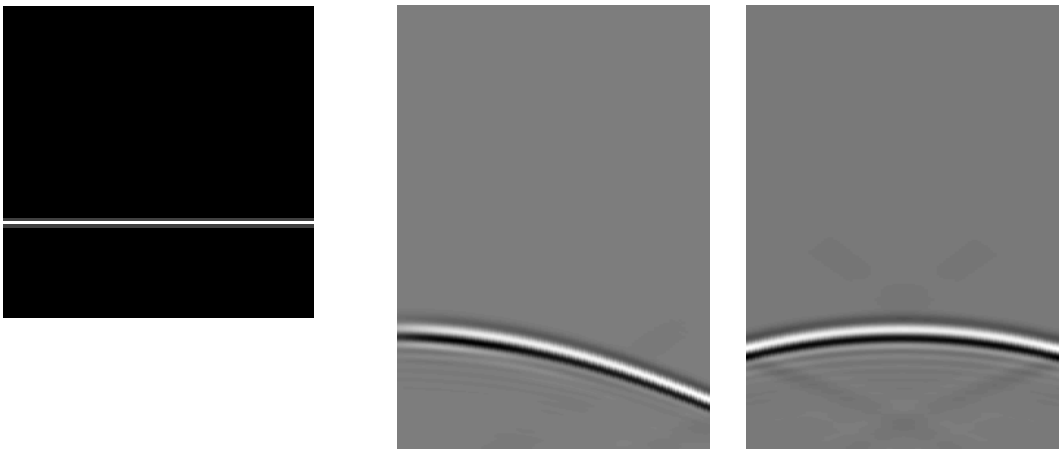


Figure (1-1): A simple seismic experiment. The source causes a seismic wave to travel through the subsurface. The wave is reflected and the receivers record its amplitude at the surface. The two arrows indicate the paths along which the wave travels from the source to the two receivers.



(a) Velocity model (b) Left shot record (c) Center shot record

Figure (1-2): Part (a) shows the subsurface velocity model used in this example (the surface is at the top). The colors indicate the propagation velocities at each point in the subsurface. Figures (b) and (c) show two shot records obtained from a simulation on the velocity model. The source locations are on the very left and in the center of the surface, respectively. Each column in a shot record corresponds to the signal recorded by a single receiver. The y-axis of the shot records is the time axis (origin at the top).

Figure 1-3 illustrates this phenomenon. We use this observation in the design of our signal model.

The signal vector $x \in \mathbb{R}^n$ can be interpreted as a matrix with h rows and w columns, where h is the number of time steps and w is the number of receivers ($hw = n$). We then require

that each column is k-sparse, which represents the fact that there is only a small number of large reflections from the subsurface. Moreover, the support sets of adjacent columns should be similar. We formalize this notion with the *Earth Mover's Distance* (EMD) [13] and require that the support sets of neighboring columns have a small total EMD.

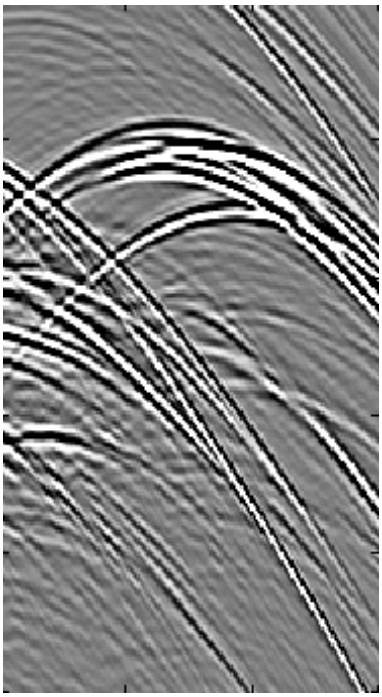


Figure (1-3): Section of a shot record from the Sigsbee2A data set. Note the parabola-like shapes in the signal. The large coefficients in neighboring columns appear at similar indices.

Chapter 2

Preliminaries

We now formally introduce our notation and the relevant background.

2.1 Notation

We generally use lower case letters like to denote vectors and scalars and capital letters for matrices and sets. We use $[n]$ as a shorthand for the set $\{1, 2, \ldots, n\}$.

Support of vectors For a vector $x \in R^n$, the support of x is the set of indices with nonzero components: $\text{supp}(x) = \{i \in [n] \,|\, x_i \neq 0\}$. Given a set of indices $\Omega \subseteq [n]$, we define the vector x_Ω to agree with x on all coordinates in Ω and to be 0 outside Ω, i.e. $(x_\Omega)_i = x_i$ for $i \in \Omega$ and $(x_\Omega)_i = 0$ for $i \notin \Omega$. A vector x is k-sparse if it has k nonzero components. Then $|\text{supp}(x)| = k$.

Vector norms We denote the ℓ_p-norm ($p \geq 1$) of a vector $x \in \mathbb{R}^n$ with $\|x\|_p$. We have $\|x\|_p = (\sum_{i=1}^n x_i^p)^{1/p}$. The ℓ_0-"norm" is $\|x\|_0 = |\{i \,|\, x_i \neq 0\}| = \text{supp}(x)$. We sometimes omit the subscript of the ℓ_2-norm for succinctness, i.e. $\|x\| = \|x\|_2$.

Matrices We denote the individual columns of a matrix $X \in \mathbb{R}^{m \times n}$ with $X_{*,i} \in \mathbb{R}^m$ for $i \in [n]$. For a set of indices $\Omega \subseteq [n]$, we define the matrix X_Ω to agree with X on all

columns with index in Ω and to be 0 everywhere else. Formally, for any $j \in [m]$, we have $(X_\Omega)_{i,j} = X_{i,j}$ for $i \in \Omega$ and $(X_\Omega)_{i,j} = 0$ for $i \notin \Omega$. Column restriction has precedence over the transpose operator, so $X_\Omega^T = (X_\Omega)^T$. Note that $\left(X^T y\right)_\Omega = X_\Omega^T y$.

2.2 Model-based compressive sensing

The key ingredient in model-based compressive sensing [1] is the idea of a structured sparsity model, which is a union of subspaces corresponding to a set of permissible support sets. In standard compressive sensing, we allow all $\binom{n}{k}$ of the k-sparse subspaces for an n-dimensional vector. By requiring some structure in the permissible support sets, model-based compressive sensing allows us to reduce this number of subspaces. Hence a smaller number of rows is necessary for a measurement matrix to have the RIP for our set of allowed signals, which leads to a smaller number of measurements. This way, model-based compressive utilizes a priori information about the signal in order to recover it from a smaller number of measurements.

2.2.1 Basic definitions

We now formally define the notion of a structured sparsity model. Since structured sparsity models usually depend on more parameters than merely the sparsity k, we introduce the concept of a *model parameter* $p \in \mathcal{P}$. Together with the ambient dimension of the signal vector, the model parameter p contains all information necessary for enumerating all permissible supports sets in the model.

Definition 1 (Structured sparsity model (definition 2 in [1]))**.** *The structured sparsity model $\mathcal{A}_p \subseteq \mathbb{R}^n$ is the set of vectors*

$$\mathcal{A}_p = \{x \in \mathbb{R}^n \,|\, \mathrm{supp}(x) \in \mathbb{A}_p\}\,, \tag{2.1}$$

where $\mathbb{A}_p = \{\Omega_1, \ldots, \Omega_{a_p}\}$ is the set of allowed, structured supports with $\Omega_i \subseteq [n]$. We call $a_p = |\mathbb{A}_p|$ the size of the structured sparsity model $\mathcal{A}_p$.

Later in this thesis we look at the vectors of the form $x + y$ with $x \in \mathcal{A}_p$ and $y \in \mathcal{A}_q$. Hence we define the addition on model parameters.

Definition 2 (Addition of model parameters (similar to definition 8 in [1]))**.** *Let $x \in \mathcal{A}_p$ and $y \in \mathcal{A}_q$. Then we define $\mathcal{A}_{p \oplus q}$ as the structured sparsity model*

$$\mathcal{A}_{p \oplus q} = \{x + y \,|\, x \in \mathcal{A}_p \text{ and } y \in \mathcal{A}_q\}\,. \tag{2.2}$$

Hence $\mathbb{A}_{p \oplus q} = \{\Omega \cup \Gamma \,|\, \Omega \in \mathcal{A}_p \text{ and } \Gamma \in \mathcal{A}_q\}$.

For many natural models, $p \oplus q$ corresponds to simple operations on the model parameters. For example, in a model that enforces only sparsity, the model parameter is the sparsity and we have $\mathcal{A}_{p \oplus q} = \mathcal{A}_{p+q}$.

Based on the definition of a structured sparsity model, we can define a variant of the RIP that applies only to vectors in the model $\mathcal{A}_p$.

Definition 3 (Model-RIP (definition 3 in [1]))**.** *The matrix $\Phi \in \mathbb{R}^{m \times n}$ has the (δ, p)-model-RIP if the following inequalities hold for all $x \in \mathcal{A}_p$:*

$$(1 - \delta)\|x\|_2^2 \;\leq\; \|\Phi x\|_2^2 \leq (1 + \delta)\|x\|_2^2\,. \tag{2.3}$$

The following result connects the size of a model with the model-RIP.

Fact 4 (Theorem 1 in [1])**.** *Let $\mathcal{A}_p$ be a structured sparsity model and let k be the size of the largest support in the model, i.e. $k = \max_{\Omega \in \mathbb{A}_p} |\Omega|$. Let $\Phi \in \mathbb{R}^{m \times n}$ be a matrix with i.i.d. Gaussian entries. Then there is a constant c such that for fixed δ, any $t > 0$ and*

$$m \geq c\,(k + \log a_p)\,, \tag{2.4}$$

Φ has the (δ, p)-model-RIP with probability at least $1 - e^{-t}$.

Note that for a sparsity model that only enforces k-sparsity, we recover the standard compressive sensing bound for Gaussian matrices: we have $a_k = \binom{n}{k}$ and hence $m = O(\log \binom{n}{k}) =$

$O(k \log \frac{n}{k})$. Moreover, the additive term k is information-theoretically necessary because we can recover the k nonzero components of x from our m compressive samples.

2.2.2 Structured sparse recovery

The authors of [1] modify CoSaMP [12] and IHT [2], two popular sparse recovery algorithms, to work for model-based compressive sensing. Informally, the modified algorithms solve the problem of recovering $x \in \mathcal{A}_p$ from compressive samples $y = \Phi x$, where Φ has the model-RIP. An important subroutine in the structured sparse recovery algorithms is a *model projection algorithm*. Given an arbitrary $x \in \mathbb{R}^n$, such an algorithm returns the best approximation to x in the model $\mathcal{A}_p$. The modified versions of CoSaMP and IHT use the model approximation algorithm to recover $x \in A_p$ from the compressive samples. Since the model approximation algorithm is used many times, its running time is very important for the overall sparse recovery procedure.

We now formally define the guarantee required for a model approximation algorithm.

Definition 5 (Model projection oracle (section 3.2 in [1]))**.** *A model projection oracle is a function $M : \mathbb{R}^n \times \mathcal{P} \to \mathbb{R}^n$ such that the following two properties hold for all $x \in R^n$ and $p \in \mathcal{P}$:*
Output model sparsity: $M(x, p) \in \mathcal{A}_p$.
Optimal model projection: $\|x - M(x, p)\|_2 = \min_{x' \in \mathcal{A}_p} \|x - x'\|_2$.
We usually write $M_p(x)$ instead of $M(x, p)$ for clarity of presentation.

Using a model projection oracle, we can now define a variant of IHT. The algorithm uses the same updates as IHT but uses a model projection algorithm instead of the hard thresholding operator. The input to the algorithm are following four parameters:

- The measurements $y \in \mathbb{R}^m$.
- The measurement matrix $\Phi \in \mathbb{R}^{m \times n}$.
- The model parameter $p \in P$.

- The number of iterations $t \in \mathbb{N}$. A larger number of iterations gives a better approximation guarantee but also leads to a longer running time.

Algorithm 1 contains the description of Model-IHT.

Algorithm 1 Model-IHT (algorithm 2 in [1])

function MIHT(y,Φ,p,t)
  $x^1 \leftarrow 0$
  for $i \leftarrow 1, \ldots, t$ **do**
    $x^{i+1} \leftarrow M_p(x^i + \Phi^T(y - \Phi x^i))$
  return x^{t+1}

It is possible to prove the following recovery guarantee for Model-IHT.

Fact 6 (Appendix C in [1]). *Let $x \in \mathcal{A}_p$ and let $y = \Phi x + e$ where Φ has $(0.1, 3p)$-model-RIP. Then*

$$\|x - \mathrm{MIHT}(y, \Phi, p, t)\| \leq 2^{-t}\|x\| + 4\|e\| \,. \tag{2.5}$$

2.3 Consequences of the RIP

A matrix with the RIP behaves like a near-isometry when restricted to a small set of columns and / or rows. The following properties are a direct result of the RIP.

Fact 7 (from section 3 in [12]). *Let $\Phi \in \mathbb{R}^{m \times n}$ be a matrix with (δ, p)-model-RIP. Let Ω be a support in the model, i.e. $\Omega \in \mathbb{A}_p$. Then the following properties hold for all $x \in \mathbb{R}^n$.*

$$\left\|\Phi_\Omega^T x\right\|_2 \leq \sqrt{1+\delta}\|x\|_2 \tag{2.6}$$

$$\left\|\Phi_\Omega^T \Phi_\Omega x\right\| \leq (1+\delta)\|x\|_2 \tag{2.7}$$

$$\left\|\left(I - \Phi_\Omega^T \Phi_\Omega\right)x\right\|_2 \leq \delta\|x\|_2 \tag{2.8}$$

Due to this near-isometry, a matrix with RIP is also "almost orthogonal" when restricted to a small set of columns and / or rows. The following property is therefore known as

approximate orthogonality.

Fact 8 (from section 3 in [12])**.** *Let $\Phi \in \mathbb{R}^{m \times n}$ be a matrix with (δ, p)-model-RIP. Let Ω and Γ be two disjoint supports with their union in the model, i.e. $\Omega \cup \Gamma \in \mathbb{A}_p$. Then the following property holds for all $x \in \mathbb{R}^n$:*

$$\left\|\Phi_\Omega^T \Phi_\Gamma x\right\|_2 \leq \delta \|x\|_2 \,. \tag{2.9}$$

2.4 Related work[1]

There has been prior research on reconstructing time sequences of spatially sparse signals (e.g., [16]). Such approaches assume that the support of the signal (or even the signal itself) does not change much between two consecutive time steps. However, the variation between two columns a and b was defined as the number of changes in the support from one column to the next, i.e. $|(\text{supp}(a) \cup \text{supp}(b)) \setminus (\text{supp}(a) \cap \text{supp}(b))|$. In contrast, we measure the difference in the support of a and b with the Earth Mover's Distance (EMD). As a result, our model not only quantifies in *how many* places the support changes but also *how far* the supported elements move. Hence our model easily handles signals such as those in figure 7-1, where the supports of any two consecutive columns can potentially even be disjoint, yet differ very little according to the EMD.

Another related paper is [10], whose authors propose the use of the EMD to measure the *approximation error* of the recovered signal in compressive sensing. In contrast, we are using the EMD to constrain the *support set* of the signals.

[1] This section is mostly taken from an earlier paper [14].

Chapter 3

The EMD model

We now define the EMD model, one of the main contributions of this thesis.

3.1 Earth Mover's Distance

A key ingredient in our proposed signal model is the Earth Mover's Distance (EMD). Originally, the EMD was defined as a metric between probability distributions and successfully applied in image retrieval problems [13]. The EMD is also known as Wasserstein metric or Wallows distance [11]. Here, we define the EMD for two sets of natural numbers.

Definition 9 (EMD)**.** *The EMD of two finite sets $A, B \subset \mathbb{N}$ with $|A| = |B|$ is*

$$\mathrm{EMD}(A, B) = \min_{\pi: A \to B} \sum_{a \in A} |a - \pi(a)| \tag{3.1}$$

where π ranges over all one-to-one mappings from A to B.

Note that $\mathrm{EMD}(A, B)$ is the cost of a min-cost matching between A and B.

We are particularly interested in the EMD between two sets of supported indices. In this case, the EMD not only measures how many indices change, but also how far the supported indices move. Hence we define a variant of the EMD which we use in our signal model.

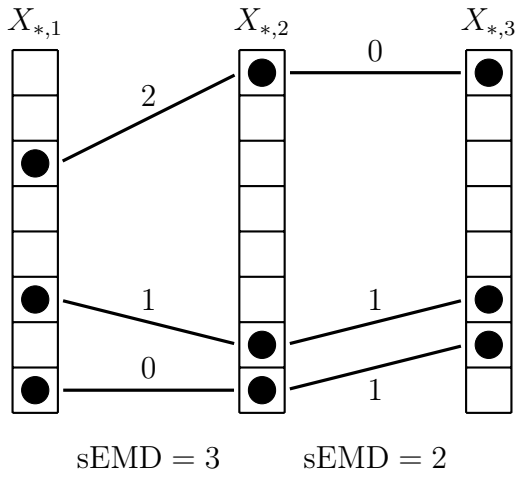


Figure (3-1): The support EMD for a matrix with three columns and eight rows. The circles stand for supported elements in the columns. The lines indicate the matching between the supported elements and the corresponding EMD cost. The total support EMD is $\text{sEMD}(X) = 2 + 3 = 5$.

Definition 10 (support-EMD for two vectors)**.** *The support-EMD (*sEMD*) of two k-sparse vectors $x, y \in \mathbb{R}^n$ is*

$$\text{sEMD}(a, b) = \text{EMD}(\text{supp}(a), \text{supp}(b)). \tag{3.2}$$

Since we want to quantify the change in support for more than two vectors, we extend the definition of the support-EMD to an entire matrix by summing the support-EMD of adjacent columns. Figure 3-1 illustrates this definition.

Definition 11 (support-EMD for a matrix)**.** *The support-EMD (*sEMD*) of a matrix $X \in \mathbb{R}^{h \times w}$ is*

$$\text{sEMD}(X) = \sum_{i=1}^{w-1} \text{sEMD}(X_{*,i}, X_{*,i+1})\,. \tag{3.3}$$

3.2 EMD model

For the definition of our EMD model, we interpret the signal $x \in \mathbb{R}^n$ as a matrix $X \in \mathbb{R}^{h \times w}$ with $n = h\,w$. For given dimensions of the signal X, the EMD model has two parameters:

- k, the total sparsity of the signal. For simplicity, we assume here and in the rest of this thesis that k is divisible by w. Then the sparsity of each column $X_{*,i}$ is $s = k/w$.
- B, the support EMD of X. We call this parameter the EMD budget.

Hence the set of model parameters for the EMD model is $\mathcal{P} = \mathbb{N} \times \mathbb{N}$.

Definition 12 (EMD model)**.** *The EMD model is the set of signals*

$$\begin{aligned} \mathcal{A}_{k,B} = \{X \in \mathbb{R}^{h \times w} : \|X_{*,i}\|_0 = s \text{ for } i \in [w], \\ sEMD(X) \leq B\}\,. \end{aligned} \tag{3.4}$$

The parameter B controls how much the support can vary from one column to the next. Setting $B = 0$ forces the support to remain constant across all columns, which corresponds to block sparsity (the blocks are the rows of X). A value of $B \geq kh$ effectively removes the EMD constraint because each supported element is allowed to move across the full height of the signal. In this case, the model demands only s-sparsity in each column. Choosing an EMD-budget B between these two extremes allows the support to vary but still gives improved sampling bounds over simple k-sparsity (see section 3.3).

It is important to note that we only constrain the EMD of the column *supports* in the signal, not the actual amplitudes. Figure 3-2 illustrates the EMD model with an example. Moreover, we can show that the sum of two signals in the EMD model is again in the EMD model (with reasonably adjusted parameters). This property is important for model our algorithm in recovery algorithms.

Theorem 13. *Let $X, Y \in \mathbb{R}^{h \times w}$. Moreover, assume that $X \in \mathcal{A}_{k_1,B_1}$ and $Y \in \mathcal{A}_{k_2,B_2}$. Then*

$$X + Y \in \mathcal{A}_{k_1+k_2,B_1+B_2}\,. \tag{3.5}$$

Proof. Each column of $X+Y$ is $\frac{k_1+k_2}{w}$ sparse. Moreover, we can use the matchings in X and Y to construct a matching for $X + Y$ with support-EMD at most $B_1 + B_2$. □

This also means that for our model, $\mathcal{A}_{p_1 \oplus p_2} \subseteq \mathcal{A}_{p_1+p_2}$.

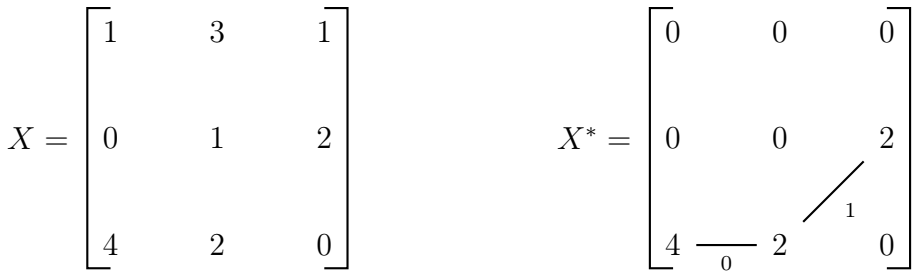


Figure (3-2): A signal X and its best approximation X^* in the EMD model $\mathcal{A}_{3,1}$. A sparsity constraint of 3 with 3 columns implies that each column has to be 1-sparse. Moreover, the total support-EMD between neighboring columns in X^* is 1. The lines in X^* indicate the support-EMD.

3.3 Sampling bound

In order to establish a sampling bound for our EMD model $\mathcal{A}_{k,B}$, we need to bound the number of subspaces $a_{k,B}$. Using fact 4, the number of required samples then is $m = O(\log a_{k,B} + k)$.

For simplicity, we assume here and in the rest of this thesis that $w = \Omega(\log h)$, i.e. the following bounds apply for all signals X besides very thin and tall matrices X.

Theorem 14.

$$\log a_{k,B} = O\left(k \log \frac{B}{k}\right) \tag{3.6}$$

Proof. For given h, w, B and k, the support is fixed by the following three decisions:

- The choice of the supported elements in the first column of X. There are $\binom{h}{s}$ possible choices.

- The distribution of the EMD budget B over the k supported elements. This corresponds to distributing B balls into k bins and hence there are $\binom{B+k-1}{k}$ possible choices.

- For each supported element, the direction (up or down) to the matching element in the next column to the right. There are 2^k possible choices.

Multiplying the choices above gives an upper bound on the number of supports. Using the inequality $\binom{a}{b} \leq \left(\frac{a\,e}{b}\right)^b$, we get

$$\log a_{k,B} \leq \log\left(\binom{h}{s}\binom{B+k-1}{k}2^k\right) \tag{3.7}$$

$$\leq s\log\frac{h}{s} + k\log\frac{B+k}{k} + k + O(s+k) \tag{3.8}$$

$$= O\left(k\log\frac{B}{k}\right). \tag{3.9}$$

□

If we allow each supported element to move a constant amount from one column to the next, we get $B = O(k)$ and hence $m = O(k)$. As mentioned above, this bound is information-theoretically optimal. Furthermore, for $B = kh$ (i.e. allowing every supported element to move anywhere in the next column) we get $m = O(k\log n)$, which almost matches the standard compressive sensing bound of $O(k\log\frac{n}{k})$. Again, choosing an EMD-budget B between these two extremes controls how many measurements we need to take in order to reconstruct x. Hence the EMD model gives us a smooth trade-off between the allowed support variability and the number of measurements necessary for reconstruction.

Chapter 4

Approximate model-IHT

In order to use our EMD model in the model-based compressive sensing framework of [1], we need to implement a model projection oracle that finds the best approximation of any given signal with a signal in the model. While our projection algorithms for the EMD model give provably good approximations, they are not necessarily optimal. Hence we extend the model-based compressive sensing framework to work with approximate projection oracles. This extension enables us to use model-based compressive sensing in cases where optimal model projections are beyond our reach but approximate projections are still efficiently computable. Since this extension might be of independent interest, we present the results in this section in a more general setting than the EMD model.

Our sparse recovery algorithm with approximate projection oracles is based on Iterative Hard Thresholding (IHT) [2], an iterative sparse recovery algorithm for standard compressive sensing. IHT iterates updates of the following form until a convergence criterion is reached:

$$x^{i+1} \leftarrow H_k(x^i + \Phi^T(y - \Phi x^i)), \tag{4.1}$$

where $H_k(x)$ returns a vector containing only the k largest components of x (hard thresholding). IHT has already been modified to work for model-based compressive sensing [1] and we extend this variant of IHT here.

4.1 Approximate projection oracles

We now give a formal definition of the approximate projection oracles that work in conjunction with our model-based sparse recovery algorithm.

Definition 15 (Head approximation oracle)**.** *A (c, f)-head approximation oracle is a function $H : \mathbb{R}^n \times \mathcal{P} \to \mathbb{R}^n$ such that the following two properties hold for all $x \in \mathbb{R}^n$ and $p \in \mathcal{P}$:*

Output model sparsity: $H(x, p) = x_\Omega$ for some $\Omega \in \mathbb{A}_{f(p)}$.

Head approximation: $\|H(x, p)\|_2 \geq c\|x_\Omega\|_2$ for all $\Omega \in \mathbb{A}_p$.

We usually write $H_p(x)$ instead of $H(x, p)$ for clarity of presentation.

Definition 16 (Tail approximation oracle)**.** *A (c, f)-tail approximation oracle is a function $T : \mathbb{R}^n \times \mathcal{P} \to \mathbb{R}^n$ such that the following two properties hold for any $x \in \mathbb{R}^n$ and $p \in \mathcal{P}$:*

Output model sparsity: $T(x, p) = x_\Omega$ for some $\Omega \in \mathbb{A}_{f(p)}$.

Tail approximation: $\|x - T(x, k)\|_2 \leq c\|x - x'\|_2$ for all $x' \in \mathcal{A}_p$.

We usually write $T_p(x)$ instead of $T(x, p)$ for clarity of presentation.

We have two different notions of approximate projections because the corresponding oracles fulfill different roles in iterative sparse recovery algorithms. A head approximation oracle allows us to find a support containing a large amount of mass compared to all other possible supports. Used on a residual proxy of the form $\Phi^T \Phi r$, such a support identifies a large part of the relevant update. However, even for input signals that are in the model, a head approximation oracle does not necessarily return a signal with zero approximation error.

In contrast, a tail approximation oracle returns a signal with an approximation error not much larger than that of the best signal approximation in the model. This notion of approximation is the guarantee we want to achieve for the overall sparse recovery algorithm. It is worth noting that combining IHT with only a tail approximation oracle instead of the optimal projection does not give a useful recovery guarantee (see appendix B).

Another important feature of the above definitions is that they not only allow head and tail approximations but also projections into *larger* models. While a projection into a significantly larger model incurs a penalty in the number of measurements necessary for recovery, this relaxation allows a much wider range of approximation algorithms.

4.2 Algorithm

We now define Approximate Model IHT (AMIHT), our sparse recovery algorithm for model-based compressive sensing with approximate projection oracles. In the algorithm, we use a (c_H, f_H)-head approximation oracle H and a (c_T, f_T)-tail approximation oracle T.

AMIHT requires the following four parameters:

- The measurements $y \in \mathbb{R}^m$.
- The measurement matrix $\Phi \in \mathbb{R}^{m \times n}$.
- The model parameter $p \in P$.
- The number of iterations $t \in \mathbb{N}$. A larger number of iterations gives a better approximation guarantee but also leads to a longer running time.

Besides the placement of the head and tail approximation oracles, AMIHT follows the form of IHT.

Algorithm 2 Approximate model-IHT

---

function AMIHT(y,Φ,p,t)

$x^1 \leftarrow 0$

for $i \leftarrow 1, \ldots, t$ **do**

$x^{i+1} \leftarrow T_p(x^i + H_{p \oplus f_T(p)}(\Phi^T(y - \Phi x^i)))$

return x^{t+1}

4.3 Analysis

We now prove the main result about AMIHT: geometric convergence to the original signal x. We make the following assumptions in the analysis of AMIHT:

- $x \in \mathbb{R}^n$ and $x \in \mathcal{A}_p$.
- $y = \Phi x + e$ for an arbitrary $e \in \mathbb{R}^m$ (the measurement noise).
- Φ has (δ, t)-model-RIP for $t = f_H(p \oplus f_T(p)) \oplus p \oplus f_T(p)$.

Moreover, we define the following quantities as shorthands:

- $r^i = x - x_i$.
- $a^i = x^i + H_{p \oplus f_T(p)}(\Phi^T(y - \Phi x^i))$.
- $b^i = \Phi^T(y - \Phi x^i)$.
- $\Omega = \operatorname{supp}(r^i)$.
- $\Gamma = \operatorname{supp}(H_{p \oplus f_T(p)}(b^i))$.

In the first lemma, we show that we can use the RIP of Φ on relevant vectors.

Lemma 17. *For all $x \in \mathbb{R}^n$ with $\operatorname{supp}(x) \in \Omega \cup \Gamma$ we have*

$$(1 - \delta)\|x\|^2 \leq \|\Phi x\|^2 \leq (1 + \delta)\|x\|^2 . \tag{4.2}$$

Proof. By the definition of T, we have $\operatorname{supp}(x^i) \in \mathbb{A}_{f_T(p)}$. Since $\operatorname{supp}(x) \in \mathbb{A}_p$, we have $\operatorname{supp}(x - x^i) \in \mathbb{A}_{p \oplus f_T(p)}$ and hence $\Gamma \in \mathbb{A}_{p \oplus f_T(p)}$. Moreover, $\operatorname{supp}(H_{p \oplus f_T(p)}(b^i)) \in \mathbb{A}_{f_H(p \oplus f_T(p))}$ by the definition of H. Therefore $\Omega \cup \Gamma \in \mathbb{A}_{f_H(p \oplus f_T(p)) \oplus p \oplus f_T(p)}$, which allows us to use the model-RIP of Φ on x with $\operatorname{supp}(x) \in \Omega \cup \Gamma$. □

As a result of lemma 17, we can use the standard consequences of the RIP such as approximate orthogonality (see section 2.3).

Just as in IHT, we use the residual proxy $\Phi^T(y - \Phi x^i)$ as update in each iteration. We now show that H preserves the relevant part of the residual proxy.

Lemma 18.

$$\left\|\Phi^T_{\Gamma\setminus\Omega}\Phi r^i\right\| \leq \frac{\sqrt{1-c_H^2}(1+\delta)+\delta}{c_H}\left\|r^i\right\| + \sqrt{1+\delta}\left(\frac{\sqrt{1-c_H^2}+1}{c_H}+1\right)\|e\|. \tag{4.3}$$

Proof. The head approximation guarantee of H gives us

$$\left\|b^i_\Gamma\right\|^2 \geq c_H^2\left\|b^i_\Omega\right\|^2 \tag{4.4}$$

$$\left\|b^i_{\Gamma\cap\Omega}\right\|^2 + \left\|b^i_{\Gamma\setminus\Omega}\right\|^2 \geq c_H^2\left\|b^i_{\Gamma\cap\Omega}\right\|^2 + c_H^2\left\|b^i_{\Omega\setminus\Gamma}\right\|^2 \tag{4.5}$$

$$\frac{(1-c_H^2)\|b^i_{\Gamma\cap\Omega}\|^2 + \left\|b^i_{\Gamma\setminus\Omega}\right\|^2}{c_H^2} \geq \left\|b^i_{\Omega\setminus\Gamma}\right\|^2 \tag{4.6}$$

$$\frac{\|\sqrt{1-c_H^2}\,b^i_{\Gamma\cap\Omega} + b^i_{\Gamma\setminus\Omega}\|}{c_H} \geq \left\|b^i_{\Omega\setminus\Gamma}\right\|. \tag{4.7}$$

We now expand b^i and apply consequences of the RIP several times:

$$\begin{aligned}\frac{1}{c_H}\|\sqrt{1-c_H^2}\,\Phi^T_{\Gamma\cap\Omega}\Phi r^i + \sqrt{1-c_H^2}\,\Phi^T_{\Gamma\cap\Omega}e + \Phi^T_{\Gamma\setminus\Omega}\Phi r^i + \Phi^T_{\Gamma\setminus\Omega}e\| \\ \geq \left\|\Phi^T_{\Omega\setminus\Gamma}\Phi r^i + \Phi^T_{\Omega\setminus\Gamma}e\right\|\end{aligned} \tag{4.8}$$

$$\begin{aligned}\frac{\sqrt{1-c_H^2}}{c_H}\left\|\Phi^T_{\Gamma\cap\Omega}\Phi r^i\right\| + \frac{\sqrt{1-c_H^2}}{c_H}\left\|\Phi^T_{\Gamma\cap\Omega}e\right\| + \frac{1}{c_H}\|\Phi^T_{\Gamma\setminus\Omega}\Phi r^i\| + \frac{1}{c_H}\left\|\Phi^T_{\Gamma\setminus\Omega}e\right\| \\ \geq \left\|\Phi^T_{\Omega\setminus\Gamma}\Phi r^i\right\| - \left\|\Phi^T_{\Omega\setminus\Gamma}e\right\|\end{aligned} \tag{4.9}$$

$$\begin{aligned}\frac{\sqrt{1-c_H^2}(1+\delta)}{c_H}\left\|r^i\right\| + \frac{\sqrt{1-c_H^2}\sqrt{1+\delta}}{c_H}\|e\| + \frac{\delta}{c_H}\left\|r^i\right\| + \frac{\sqrt{1+\delta}}{c_H}\|e\| \\ \geq \left\|\Phi^T_{\Omega\setminus\Gamma}\Phi r^i\right\| - \sqrt{1+\delta}\|e\|.\end{aligned} \tag{4.10}$$

Rearranging and grouping terms gives the statement of the lemma:

$$\left\|\Phi^T_{\Gamma\setminus\Omega}\Phi r^i\right\| \leq \frac{\sqrt{1-c_H^2}(1+\delta)+\delta}{c_H}\left\|r^i\right\| + \sqrt{1+\delta}\left(\frac{\sqrt{1-c_H^2}+1}{c_H}+1\right)\|e\|\,. \tag{4.11}$$

□

We now prove the main theorem: geometric convergence for approximate model-IHT.

Theorem 19.

$$\begin{aligned}\left\|r^{i+1}\right\| \leq (1+c_T)\left(\frac{\sqrt{1-c_H^2}(1+\delta)+\delta}{c_H}+2\delta\right)\left\|r^i\right\| + \\ (1+c_T)\sqrt{1+\delta}\left(\frac{\sqrt{1-c_H^2}+1}{c_H}+4\right)\|e\|\,.\end{aligned} \tag{4.12}$$

Proof. The triangle inequality gives us

$$\left\|r^{i+1}\right\| = \left\|x-x^{i+1}\right\| \tag{4.13}$$

$$\leq \left\|x-a^i\right\| + \left\|x^{i+1}-a^i\right\| \tag{4.14}$$

$$\leq (1+c_T)\left\|x-a^i\right\|, \tag{4.15}$$

where the last line follows because T is a (c_T, f_T)-tail approximation oracle.

We now bound $\|x-a^i\|$:

$$\left\|x-a^i\right\| = \left\|x-x^i-H_{p\oplus f_T(p)}(b^i)\right\| \tag{4.16}$$

$$= \left\|r^i - H_{p\oplus f_T(p)}(b^i)\right\| \tag{4.17}$$

$$\leq \left\|r^i-b^i_\Omega\right\| + \left\|H_{p\oplus f_T(p)}(b^i)+b^i_\Omega\right\|. \tag{4.18}$$

Looking at each term individually:

$$\left\|r^i - b^i_\Omega\right\| = \left\|r^i - \Phi^T_\Omega \Phi r^i + \Phi^T_\Omega e\right\| \tag{4.19}$$

$$\leq \left\|\left(I - \Phi^T_\Omega \Phi_\Omega\right) r^i\right\| + \left\|\Phi^T_\Omega e\right\| \tag{4.20}$$

$$\leq \delta \left\|r^i\right\| + \sqrt{1+\delta}\|e\|. \tag{4.21}$$

And

$$\left\|H_{p \oplus f_T(p)}(b^i) - b^i_\Omega\right\| = \left\|b^i_\Gamma - b^i_\Omega\right\| \tag{4.22}$$

$$= \left\|\Phi^T_\Gamma \Phi r^i - \Phi^T_\Gamma e - \Phi^T_\Omega \Phi r^i - \Phi^T_\Omega e\right\| \tag{4.23}$$

$$\leq \left\|\Phi^T_\Gamma \Phi r^i - \Phi^T_\Omega \Phi r^i\right\| + 2\sqrt{1+\delta}\|e\| \tag{4.24}$$

$$= \left\|\Phi^T_{\Gamma\setminus\Omega} \Phi r^i - \Phi^T_{\Omega\setminus\Gamma} \Phi r^i\right\| + 2\sqrt{1+\delta}\|e\| \tag{4.25}$$

$$\leq \left\|\Phi^T_{\Gamma\setminus\Omega} \Phi r^i\right\| + \left\|\Phi^T_{\Omega\setminus\Gamma} \Phi r^i\right\| + 2\sqrt{1+\delta}\|e\| \tag{4.26}$$

$$\leq \delta \left\|r^i\right\| + \left\|\Phi^T_{\Omega\setminus\Gamma} \Phi r^i\right\| + 2\sqrt{1+\delta}\|e\|. \tag{4.27}$$

Using lemma 18 gives

$$\begin{aligned} &\|H_{p \oplus f_T(p)}(b^i) - b^i_\Omega\| \\ &\quad \leq \left(\frac{\sqrt{1-c_H^2}(1+\delta)+\delta}{c_H} + \delta\right)\left\|r^i\right\| + \sqrt{1+\delta}\left(\frac{\sqrt{1-c_H^2}+1}{c_H} + 3\right)\|e\|. \end{aligned} \tag{4.28}$$

Combining the inequalities above:

$$\begin{aligned} \left\|r^{i+1}\right\| \leq (1+c_T)&\left(\frac{\sqrt{1-c_H^2}(1+\delta)+\delta}{c_H} + 2\delta\right)\left\|r^i\right\| + \\ &(1+c_T)\sqrt{1+\delta}\left(\frac{\sqrt{1-c_H^2}+1}{c_H} + 4\right)\|e\|. \end{aligned} \tag{4.29}$$

□

Corollary 20. *For $\delta = 0.01$, $c_T = 1.5$ and $c_H = 0.95$ we get*

$$\left\|r^{i+1}\right\| \leq 0.91\left\|r^{i}\right\| + 13.53\|e\|\,. \tag{4.30}$$

Corollary 21. *For $\delta = 0.01$, $c_T = 1.5$ and $c_H = 0.95$ we get*

$$\|x - \mathrm{AMIHT}(y, \Phi, p, t)\| \leq 0.91^{t}\|x\| + 150.34\|e\|\,. \tag{4.31}$$

Proof. We iterate corollary 20 and use

$$13.53\sum_{i=0}^{\infty} 0.91^{i} \leq 150.34\,. \tag{4.32}$$

□

These results shows that we get an overall recovery guarantee that is comparable to that of model-based compressive sensing with optimal projections, in spite of using only approximate projections.

Chapter 5

Head approximation algorithm

We now study the problem of finding a good head approximation for the EMD model. Ideally, we could find an algorithm H mapping arbitrary signals to signals in $\mathcal{A}_{k,B}$ with the following guarantee:

$$\|H_{k,B}(x)\| = \max_{\Omega \in \mathbb{A}_{k,B}} \|x_\Omega\| . \tag{5.1}$$

Instead of finding the best projection in the model, we propose an efficient approximation algorithm satisfying the guarantees of a head approximation oracle (definition 15). Although we only have an approximate oracle, our sparse recovery algorithm still gives us a recovery guarantee comparable to that of model-based compressive sensing with optimal projections (see chapter 4).

Therefore, we develop an algorithm for the following problem. Given an arbitrary signal x and an approximation ratio c, find a support $\Omega \in \mathbb{A}_{O(k),O(B \log k)}$ such that

$$\|x_\Omega\| \geq c \max_{\Gamma \in \mathbb{A}_{k,B}} \|x_\Gamma\| . \tag{5.2}$$

As before, we interpret our signal x as a matrix $X \in \mathbb{R}^{h \times w}$. Because of the structure of the EMD-constraint, each support for x can be seen as a set of s paths from the leftmost to the rightmost column in X (recall that s is the per-column sparsity parameter, i.e. $s = k/w$). Hence the goal of our algorithm is to find a set of s such paths that cover a large amount of

amplitudes in the signal. We describe an algorithm that allows us to get a constant fraction of the optimal amplitude sum and then repeat this algorithm several times in order to boost the approximation ratio (while increasing the sparsity and EMD budget of the result only moderately).

For simplicity, we now look at the closely related problem where $X_{i,j} \geq 0$ for all i, j and we are interested in the ℓ_1-guarantee

$$\|x_\Omega\|_1 \geq c \max_{\Omega \in \mathbb{A}_{k,B}} \|x_\Omega\|_1 . \tag{5.3}$$

This modification allows us to add amplitudes along paths in the analysis of the head approximation algorithm. We can easily convert the input signal x into a matrix satisfying these constraints by squaring each amplitude.

In the rest of this chapter, we use OPT to denote the largest amplitude sum achievable with a support in $\mathbb{A}_{k,B}$, so

$$OPT = \max_{\Omega \in \mathbb{A}_{k,B}} \|x_\Omega\|_1 . \tag{5.4}$$

Moreover, we set $\Omega_{OPT} \in \mathbb{A}_{k,B}$ to be a support with $\|x_\Omega\|_1 = OPT$.

5.1 Basic algorithm

We first make the notion of paths in the input matrix X more formal.

Definition 22 (Path in a matrix)**.** *Given a matrix $X \in \mathbb{R}^{h \times w}$, a path $p \subseteq [h] \times [w]$ is a set of w locations in X with one location per column, i.e. $|p| = w$ and $\bigcup_{(i,j) \in p} i = [w]$.*

Definition 23 (Weight of a path)**.** *Given a matrix $X \in \mathbb{R}^{h \times w}$ and a path p in X, the weight of p is the sum of amplitudes on p:*

$$w_X(p) = \sum_{(i,j) \in p} X_{i,j} . \tag{5.5}$$

Definition 24 (EMD of a path)**.** *Given a matrix $X \in \mathbb{R}^{h \times w}$ and a path p in X, the EMD of p is the sum of the EMDs between locations in neighboring columns. Let $j_1, \ldots, j_w$ be the locations of p in columns 1 to w. Then*

$$\mathrm{EMD}(p) = \sum_{i=1}^{w-1} |j_i - j_{i+1}| \,. \tag{5.6}$$

Note that the a path p in X is a support with $w_X(p) = \|X_p\|_1$ and $\mathrm{EMD}(p) = \mathrm{sEMD}(X_p)$. We use this fact to iteratively build a support Ω by finding s paths in X. Algorithm 3 contains the description of HeadApproxBasic.

Algorithm 3 Basic head approximation algorithm

---

function HeadApproxBasic(X, k, B)
  $X^{(1)} \leftarrow X$
  for $i \leftarrow 1, \ldots, s$ **do**
    Find the path q_i from column 1 to column w in $X^{(i)}$ that maximizes $w^{(i)}(q_i)$
      and uses at most EMD-budget $\left\lfloor \frac{B}{i} \right\rfloor$.
    $X^{(i+1)} \leftarrow X^{(i)}$
    for $(u, v) \in q_i$ **do**
      $X^{(i+1)}_{u,v} \leftarrow 0$
  return $\bigcup_{i=1}^{s} q_i$

We now show that HeadApproxBasic finds a constant fraction of the amplitude sum of the best support while only moderately increasing the size of the model. For simplicity, we introduce the following shorthands:

- $w(p) = w_X(p)$.
- $w^{(i)}(p) = w_{X^{(i)}}(p)$.

Theorem 25. *Let Ω be the support returned by* HeadApproxBasic*. Let $B' = \lceil H_s \rceil B$, where H_s is the s-th harmonic number. Then $\Omega \in \mathcal{A}_{k,B'}$ and $\|X_\Omega\|_1 \geq \frac{1}{4} OPT$.*

Proof. We can decompose Ω_{OPT} into s disjoint paths in A. Let $p_1, \ldots, p_s$ be such a decomposition with $\mathrm{EMD}(p_1) \geq \mathrm{EMD}(p_2) \geq \ldots \geq \mathrm{EMD}(p_s)$. Note that $\mathrm{EMD}(p_i) \leq \left\lfloor \frac{B}{i} \right\rfloor$: otherwise $\sum_{j=1}^{i} \mathrm{EMD}(p_i) > B$ and since $\mathrm{EMD}(\Omega_{OPT}) \leq B$ this would be a contradiction.

Since Ω is the union of s paths in A, Ω has column-sparsity s. Moreover, we have

$$\mathrm{EMD}(\Omega) = \sum_{i=1}^{s} \mathrm{EMD}(q_i) \leq \sum_{i=1}^{s} \left\lfloor \frac{B}{i} \right\rfloor \leq \lceil H_s \rceil B. \tag{5.7}$$

So $\Omega \in \mathcal{A}_{k,B'}$.

When finding path q_i in $X^{(i)}$, there are two cases:

1. $w^{(i)}(p_i) \leq \frac{1}{2} w(p_i)$, i.e. the paths $q_1, \ldots, q_{i-1}$ have already covered more than half of the amplitude sum of p_i in X.

2. $w^{(i)}(p_i) > \frac{1}{2} w(p_i)$, i.e. there is still more than half of the amplitude sum of p_i remaining in $X^{(i)}$. Since $\mathrm{EMD}(p_i) \leq \left\lfloor \frac{B}{i} \right\rfloor$, the path p_i is a candidate when searching for the optimal q_i and hence we find a path q_i with $w^{(i)}(q_i) > \frac{1}{2} w(p_i)$.

Let $C = \{i \in [s] \mid \text{case 1 holds for } q_i\}$ and $D = \{i \in [s] \mid \text{case 2 holds for } q_i\}$ (note that $C = [s] \setminus D$). Then we have

$$\begin{aligned}
\|A_\Omega\|_1 &= \sum_{i=1}^{s} w^{(i)}(q_i) && (5.8)\\
&= \sum_{i \in C} w^{(i)}(q_i) + \sum_{i \in D} w^{(i)}(q_i) && (5.9)\\
&\geq \sum_{i \in D} w^{(i)}(q_i) && (5.10)\\
&\geq \frac{1}{2} \sum_{i \in D} w(p_i). && (5.11)
\end{aligned}$$

For each p_i with $i \in C$, let $E_i = p_i \cap \bigcup_{j<i} q_j$, i.e. the locations of p_i already covered by

some p_j when searching for p_i. Then we have

$$\sum_{(x,y)\in E_i} X_{u,v} = w(p_i) - w^{(i)}(p_i) \geq \frac{1}{2} w(p_i) \tag{5.12}$$

and

$$\sum_{i\in C} \sum_{(u,v)\in E_i} X_{u,v} \geq \frac{1}{2} \sum_{i\in C} w(p_i). \tag{5.13}$$

Since the p_i are pairwise disjoint, so are the E_i. Moreover, for every $i \in C$ we have $E_i \subseteq \bigcup_{j=1}^{s} q_j$. Hence

$$\|X_\Omega\|_1 = \sum_{i=1}^{s} w^{(i)}(q_i) \tag{5.14}$$

$$\geq \sum_{i\in C} \sum_{(u,v)\in E_i} X_{u,v} \tag{5.15}$$

$$\geq \frac{1}{2} \sum_{i\in C} w(p_i). \tag{5.16}$$

Combining equations 5.11 and 5.16 gives

$$2\|X_\Omega\|_1 \geq \frac{1}{2} \sum_{i\in C} w(p_i) + \frac{1}{2} \sum_{i\in D} w(p_i) = \frac{1}{2} OPT. \tag{5.17}$$

So $\|X_\Omega\|_1 \geq \frac{1}{4} OPT$. □

We now analyze the running time of HeadApproxBasic. The most expensive part of HeadApproxBasic is finding the paths with largest weight for a given EMD budget.

Theorem 26. HeadApproxBasic *runs in* $O(s\,n\,B\,h)$ *time.*

Proof. We can find a largest-weight path in X by dynamic programming over the graph given by $whB = nB$ states. We have one state for each combination of location in X and amount of EMD budget currently used. At each state, we store the largest weight achieved by a path ending at the corresponding location in X and using the corresponding amount of EMD

budget. Moreover, each state has h outgoing edges to the states in the next column (given the current location, the decision on the next location also fixes the new EMD amount). Hence the time complexity of finding one largest-weight path is $O(nBh)$. Since we repeat this procedure s times, the overall time complexity is $O(snBh)$. □

Note that the EMD model is mainly interesting in cases where $B = O(n)$ (especially $B = O(sw)$. Hence HeadApproxBasic has a strongly polynomial running time for our purposes.

5.2 Better approximation ratio

We now use HeadApproxBasic to get a head approximation guarantee

$$\|x_\Omega\|_1 \geq c \max_{\Omega \in \mathbb{A}_{k,B}} \|x_\Omega\|_1 \tag{5.18}$$

for arbitrary $c < 1$. We achieve this by running HeadApproxBasic several times to get a larger support that contains a larger fraction of OPT. We call the resulting algorithm HeadApprox (see algorithm 4).

Algorithm 4 Head approximation algorithm

function HeadApprox(X, k, B, c)
  $d \leftarrow \left\lceil \frac{4c}{1-c} \right\rceil$
  $X^{(1)} \leftarrow A$
  for $i \leftarrow 1, \ldots, d$ **do**
    $\Omega_i \leftarrow$ HeadApproxBasic($X^{(i)}, k, B$)
    $X^{(i+1)} \leftarrow X^{(i)}$
    $X^{(i+1)}_{\Omega_i} \leftarrow 0$
  return $\bigcup_{i=1}^{d} \Omega_i$

We now show that HeadApprox solves the head approximation problem for arbitrary c. We use $d = \left\lceil \frac{4c}{1-c} \right\rceil$ as a shorthand throughout the analysis.

Theorem 27. *Let Ω be the support returned by* HeadApproxBasic. *Let $k' = dk$ and $B' = d\lceil H_s \rceil B$. Then $\Omega \in \mathcal{A}_{k',B'}$ and $\|A_\Omega\|_1 \geq cOPT$.*

Proof. From theorem 25 we know that for each i, $\Omega_i \in \mathcal{A}_{k,\lceil H_s \rceil B}$. Since $\Omega = \bigcup_{i=1}^{d} \Omega_i$, we have $\Omega \in \mathcal{A}_{k',B'}$.

Before each call to HeadApproxBasic, at least one of the following two cases holds:

Case 1: $\|X^{(i)}_{\Omega_{OPT}}\|_1 < (1-c)OPT$. So the supports Ω_j found in previous iterations already cover amplitudes with a sum of at least $cOPT$.

Case 2: $\|X^{(i)}_{\Omega_{OPT}}\|_1 \geq (1-c)OPT$. Since Ω_{OPT} is a candidate solution with parameters k and B in $X^{(i)}$, we have that $\|X^{(i)}_{\Omega_i}\|_1 \geq \frac{1-c}{4} OPT$.

After d iterations of the for-loop in HeadApprox, one of the following two cases holds:

Case A: Case 1 holds for at least one iteration. Hence $\|X_\Omega\|_1 \geq cOPT$.

Case B: Case 2 holds in all d iterations. Since we set $X^{(i+1)}_{\Omega_i} \leftarrow 0$ in each iteration, we have $\left\|X^{(i)}_{\Omega_j}\right\| = 0$ for all $j < i$. In particular, this means that $\left\|X^{(i)}_{\Omega_i}\right\|_1 + \left\|X^{(i+1)}_{\Omega_{i+1}}\right\|_1 = \left\|X^{(i)}_{\Omega_i \cup \Omega_{i+1}}\right\|_1$ and hence

$$\|X_\Omega\|_1 = \sum_{i=1}^{d} \left\|X^{(i)}_{\Omega_i}\right\|_1 \tag{5.19}$$

$$\geq \left\lceil \frac{4c}{1-c} \right\rceil \frac{1-c}{4} OPT \tag{5.20}$$

$$\geq cOPT. \tag{5.21}$$

So in both cases A and B, at the end of the algorithm we have $\|X_\Omega\|_1 \geq cOPT$. □

The running time of HeadApprox follows directly from the running time of HeadApproxBasic (see theorem 26).

Theorem 28. HeadApprox *runs in $O(d\,s\,n\,B\,h)$ time.*

We can now conclude that HeadApprox satisfies the definition of a head-approximation algorithm (see definition 15).

Corollary 29. HeadApprox *is a* $\left(c, \left(\left\lceil \frac{4c^2}{1-c^2} \right\rceil k, \left\lceil \frac{4c^2}{1-c^2} \right\rceil \lceil H_s \rceil B\right)\right)$*-head approximation algorithm. Moreover,* HeadApprox *runs in* $O(s\,n\,B\,h)$ *time for fixed* c*.*

Proof. Let $X' \in \mathbb{R}^{h \times w}$ with $X'_{i,j} = X^2_{i,j}$. We run HeadApprox with parameters X', k, B and c^2.

Let Ω be the support returned by HeadApprox. Let $k' = \left\lceil \frac{4c^2}{1-c^2} \right\rceil k$ and $B' = \left\lceil \frac{4c^2}{1-c^2} \right\rceil \lceil H_s \rceil B$. Then according to theorem 27 we have $\Omega \in \mathbb{A}_{k',B'}$.

Moreover, we get the following ℓ_1-guarantee:

$$\|x'_\Omega\|_1 \geq c^2 \max_{\Omega \in \mathbb{A}_{k,B}} \|x'_\Omega\|_1 \,, \tag{5.22}$$

which directly implies

$$\|x_\Omega\|_2^2 \geq c^2 \max_{\Omega \in \mathbb{A}_{k,B}} \|x_\Omega\|_2^2 \,. \tag{5.23}$$

And hence

$$\|x_\Omega\| \geq c \max_{\Omega \in \mathbb{A}_{k,B}} \|x_\Omega\| \,. \tag{5.24}$$

The running time bound follows directly from theorem 28. □

Chapter 6

Tail approximation algorithm

In addition to the head approximation algorithm introduced in the previous chapter, we also need a *tail approximation algorithm* in order to use our EMD model in compressive sensing. Ideally, we could give an algorithm $T_{k,B}$ with the following guarantee (an optimal projection algorithm of this form would actually make the head approximation algorithm unnecessary):

$$\|x - T_{k,B}(x)\| = \min_{x' \in \mathcal{A}_{k,B}} \|x - x'\| . \tag{6.1}$$

Instead, we develop an algorithm with an approximate tail guarantee. As before, we interpret our signal x as a matrix $X \in \mathbb{R}^{h \times w}$. Since we also establish a connection between paths in the matrix X and support sets for X, we study the ℓ_1-version of the tail approximation problem: given an arbitrary signal x and an approximation ratio c, find a support $\Omega \in \mathbb{A}_{k,O(B)}$ such that

$$\|x - x_\Omega\|_1 \leq c \min_{\Gamma \in \mathbb{A}_{k,B}} \|x - x_\Gamma\|_1 . \tag{6.2}$$

Note that we allow a constant factor increase in the EMD budget of the result.

6.1 EMD flow networks

For the tail-approximation algorithm, we use a graph-based notion of paths in order to find good supports for X. Extending the connection between the EMD and min-cost matching, we convert the tail approximation problem into a min-cost flow problem. The min-cost flow problem with a single cost function is a *Lagrangian relaxation* of the original problem with two separate objectives (signal approximation and EMD constraint). A core element of the algorithm is the corresponding flow network, which we now define.

Definition 30 (EMD flow network)**.** *For a given signal X, sparsity k and trade-off parameter λ, the flow network $G_{X,k,\lambda}$ consists of the following elements:*

- *The* nodes *are a source, a sink and a node $v_{i,j}$ for $i \in [h]$, $j \in [w]$, i.e. one node per entry in X (besides source and sink).*
- *G has an* edge *from every $v_{i,j}$ to every $v_{k,j+1}$ for $i, k \in [h]$, $j \in [w-1]$. Moreover, there is an edge from the source to every $v_{i,1}$ and from every $v_{i,w}$ to the sink for $i \in [h]$.*
- *The* capacity *on every edge and node is 1.*
- *The* cost *of each node $v_{i,j}$ is $-|X_{i,j}|$. The cost of an edge from $v_{i,j}$ to $v_{k,j+1}$ is $\lambda|i-k|$. The cost of the source, the sink and all edges incident to the source or sink is 0.*
- *The* supply *at the source is s and the demand at the sink is s.*

Figure 6-1 illustrates this definition with an example. An important property of a EMD flow network $G_{X,k,\lambda}$ is the correspondence between flows in $G_{X,k,\lambda}$ and supports in X. As before, recall that s is the per-column sparsity in the EMD-model, i.e. $s = k/w$. We first formally define the support induced by a set of paths.

Definition 31 (Support of a set of paths)**.** *Let $X \in \mathbb{R}^{h \times w}$ be a signal matrix, k be a sparsity parameter and $\lambda \geq 0$. Let $P = \{p_1, \ldots, p_s\}$ be a set of disjoint paths from source to sink in $G_{X,k,\lambda}$ such that no two paths in P intersect vertically (i.e. if the p_i are sorted vertically and*

$$X = \begin{bmatrix} 1 & 3 \\ 0 & -1 \\ 2 & 1 \end{bmatrix}$$

$G_{X,k,\lambda} =$

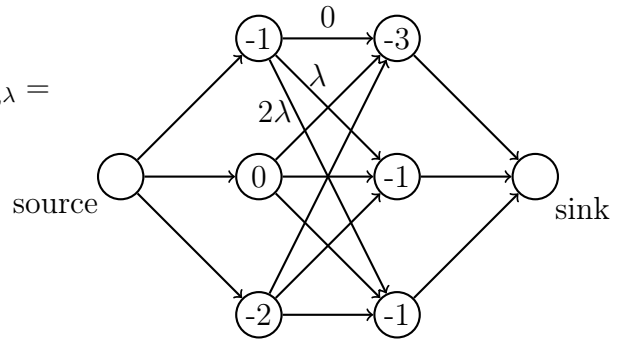


Figure (6-1): A signal X with the corresponding flow network $G_{X,k,\lambda}$. The node costs are the negative absolute values of the corresponding signal components. The numbers on edges indicate the edge costs (most edge costs are omitted for clarity). All capacities in the flow network are 1. The edge costs are the vertical distances between the start and end nodes.

$i \leq j$, then $(u, v) \in p_i$ and $(w, v) \in p_j$ implies $u < w$). Then the paths in P define a support

$$\Omega_P = \{(u, v) \,|\, (u, v) \in p_i \text{ for some } i \in [s]\}\,. \tag{6.3}$$

Now, we introduce the property connecting paths and supports.

Theorem 32. *Let $X \in \mathbb{R}^{h \times w}$ be a signal matrix, k be a sparsity parameter and $\lambda \geq 0$. Let $P = \{p_1, \ldots, p_s\}$ be a set of disjoint paths from source to sink in $G_{X,k,\lambda}$ such that no two paths in P intersect vertically. Finally, let f_P be the flow induced in $G_{X,k,\lambda}$ by sending a single unit of flow along each path in P and let $c(f_P)$ be the cost of f_P. Then*

$$c(f_P) = -\|X_{\Omega_P}\|_1 + \lambda\, \mathrm{sEMD}(X_{\Omega_P})\,. \tag{6.4}$$

Proof. The theorem follows directly from the definition of $G_{X,k,\lambda}$ and Ω_P. The node costs of P result in the term $-\|X_{\Omega_P}\|_1$. Since the paths in P do not intersect vertically, they are a min-cost matching for the elements in Ω_P. Hence the cost of edges between columns of X sums up to $\lambda\, \mathrm{sEMD}(X_{\Omega_P})$. □

So finding a min-cost flow in $G_{X,k,\lambda}$ allows us to find the best support for a given trade-off

between signal approximation and support EMD. In the next section, we use this connection to find a support set with a tail approximation guarantee.

6.2 Algorithm

We first formalize the connection between min-cost flows in $G_{X,k,\lambda}$ and good supports in X.

Lemma 33. *Let $G_{X,k,\lambda}$ be an EMD flow network and let f be a min-cost flow in $G_{X,k,\lambda}$. Then f can be decomposed into s disjoint paths $P = \{p_1, \ldots, p_s\}$ which do not intersect vertically. Moreover,*

$$\|X - X_{\Omega_P}\|_1 + \lambda \text{sEMD}(X_{\Omega_P}) = \min_{\Omega \in \mathbb{A}_{k,B}} \|X - X_\Omega\|_1 + \lambda \text{sEMD}(X_\Omega)\,. \tag{6.5}$$

Proof. Note that $\|X - X_\Omega\|_1 = \|X\|_1 - \|X_\Omega\|_1$. Since $\|X\|_1$ does not depend on Ω, minimizing $\|X - X_\Omega\|_1 + \lambda \text{sEMD}(X_\Omega)$ with respect to Ω is equivalent to minimizing $-\|X_\Omega\|_1 + \lambda \text{sEMD}(X_\Omega)$. Hence we show

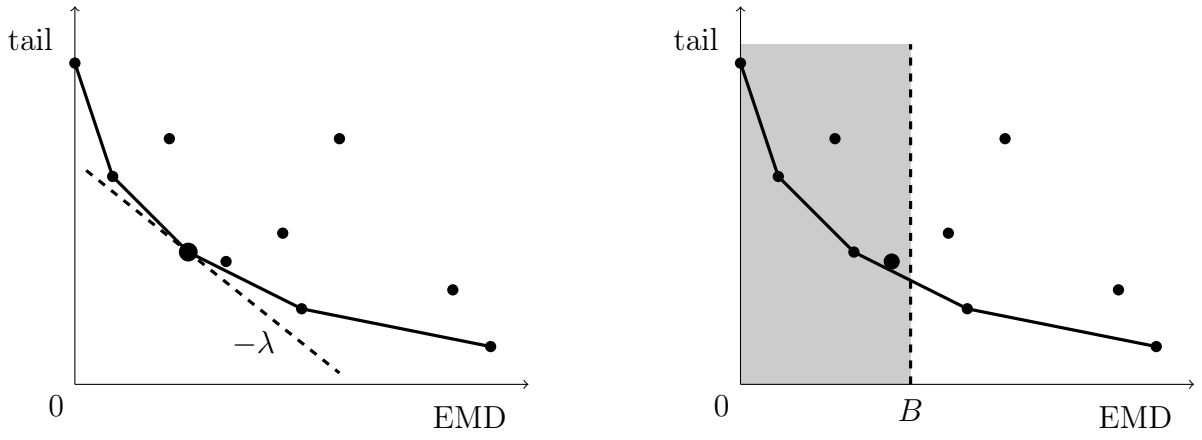
$$-\|X_{\Omega_P}\|_1 + \lambda \text{sEMD}(X_{\Omega_P}) = \min_{\Omega \in \mathbb{A}_{k,B}} -\|X_\Omega\|_1 + \lambda \text{sEMD}(X_\Omega) \tag{6.6}$$

instead of equation 6.5.

All edges and nodes in $G_{X,k,\lambda}$ have capacity one, so f can be composed into disjoint paths P. Since $G_{X,k,\lambda}$ has integer capacities, the flow f is integral and therefore P contains exactly s paths. Moreover, the paths in P are not intersecting vertically: if p_i and p_j intersect vertically, we can relax the intersection to get a set of paths P' with smaller support EMD and hence a flow with smaller cost – a contradiction.

Moreover, each support $\Omega \in \mathbb{A}_{k,B}$ gives rise to a set of disjoint, not vertically intersecting paths Q and thus also to a flow f_Q with $c(f_Q) = -\|X_\Omega\|_1 + \lambda \text{sEMD}(X_\Omega)$. Since f is a min-cost flow, so $c(f) \leq c(f_Q)$. The statement of the theorem follows. □

We can use this connection to probe the set of supports for a given signal X. Each choice



(a) Result of a min-cost flow run.

(b) Optimal solution

Figure (6-2): Each point corresponds to a support Ω for the matrix X. The x-coordinate of Ω is sEMD(X_Ω) and the y-coordinate is $\|X - X_\Omega\|_1$. Finding min-cost flows in $G_{X,k,\lambda}$ allows us to find points on the convex hull of support points.

The dashed line in figure (a) illustrates the result of MinCostFlow$(G_{X,k,\lambda})$, which is also the slope of the line. The point found is the first point we hit when we move a line with slope $-\lambda$ from the origin upwards (the larger dot in the figure).

For a given EMD budget B, we want to find the support with the smallest tail. The shaded region in figure (b) indicates the region where supports with support-EMD at most B lie. We want to find the point in this region with minimum y-coordinate. Note that this point (the larger dot in the figure) does not necessarily lie on the convex hull.

of λ defines a trade-off between the size of the tail and the EMD cost. A useful geometric perspective on this problem is illustrated in figure 6-2. Each support gives rise to a point in the plane defined by EMD-cost in x-direction and size of the tail in y-direction. A choice of λ defines a set of lines with slope $-\lambda$. The point corresponding to the result returned by MinCostFlow$(G_{X,k,\lambda})$ can be found geometrically as follows: start with a line with slope $-\lambda$ through the origin and move the line upwards until it hits the first point p. The support Ω corresponding to p then minimizes $-\|X_\Omega\|_1 + \lambda\text{sEMD}(X_\Omega)$, which is the same result as the min-cost flow.

Therefore, finding min-cost flows allows us to explore the convex hull of possible supports. While the best support for a given B does not necessarily lie on the convex hull, we still get a provably good result by finding nearby points on the convex hull. Algorithm 5 describes the overall tail approximation algorithm implementing this idea. The parameters d and δ for TailApprox quantify the acceptable tail approximation ratio (see theorem 35). In the algorithm, we assume that MinCostFlow($G_{X,k,\lambda}$) returns the support corresponding to a min-cost flow in $G_{X,k,\lambda}$.

Algorithm 5 Tail approximation algorithm

function TailApprox(X, k, B, d, δ)
  $x_{\min} \leftarrow \min_{|X_{i,j}|>0}|X_{i,j}|$
  $\varepsilon \leftarrow \frac{x_{\min}}{wh^2}\delta$
  if X is s-sparse in every column **then**
    $\lambda_0 \leftarrow \frac{x_{\min}}{2wh^2}$
    $\Omega \leftarrow$ MinCostFlow(G_{X,k,λ_0})
    if $\Omega \in \mathbb{A}_{k,B}$ **then**
      return X_Ω
  $\lambda_r \leftarrow 0$
  $\lambda_l \leftarrow \|X\|_1$
  while $\lambda_l - \lambda_r > \varepsilon$ **do**
    $\lambda_m \leftarrow (\lambda_l + \lambda_r)/2$
    $\Omega \leftarrow$ MinCostFlow(G_{X,k,λ_m})
    if sEMD(X_Ω) $\geq B$ and sEMD(X_Ω) $\leq dB$ **then**
      return X_Ω
    if sEMD(X_Ω) $> B$ **then**
      $\lambda_r \leftarrow \lambda_m$
    else
      $\lambda_l \leftarrow \lambda_m$
  $\Omega \leftarrow$ MinCostFlow(G_{X,k,λ_l})
  return X_Ω

We now prove the main result for TailApprox: a bicriterion approximation guarantee that allows us to use TailApprox as a tail approximation algorithm. We show that one of the following two cases occurs:

- We get a solution with tail approximation error at least as good as the best support with support-EMD B. The support-EMD of our solution is at most a constant times larger than B.
- We get a solution with bounded tail approximation error and support-EMD at most B.

Before we prove the main theorem, we show that TailApprox always returns the optimal result for signals $X \in \mathcal{A}_{k,B}$, i.e. X itself.

Lemma 34. *For any $X \in \mathcal{A}_{k,B}$,* TailApprox*(X, k, B, d, δ) returns X for any d and δ.*

Proof. Since $X \in \mathcal{A}_{k,B}$, every column of X is s-sparse. We show that the following call MinCostFlow$(G_{X,k,\lambda_{\min}})$ returns an $\Omega \in \mathbb{A}_{k,B}$ with $\text{supp}(X) \subseteq \Omega$.

First, we prove that MinCostFlow$(G_{X,k,\lambda_{\min}})$ returns a support set covering all nonzero entries in X. As a result, $\text{supp}(X) \subseteq \Omega$.

Let Γ be any s-column-sparse support set not covering all entries in X and let Δ be any s-column-sparse support set covering all entries in X. So $\|X - X_\Gamma\| \geq x_{\min}$ and $\|X - X_\Delta\| = 0$. Hence

$$\begin{aligned}
-\|X_\Delta\|_1 + \lambda_0 \text{sEMD}(X_\Delta) &= -\|X_\Delta\| + \frac{x_{\min}}{2wh^2}\text{sEMD}(X_\Delta) \\
&< -\|X_\Delta\|_1 + x_{\min} \\
&\leq -\|X_\Gamma\|_1 \\
&\leq -\|X_\Gamma\|_1 + \lambda_0 \text{sEMD}(X_\Gamma)\,.
\end{aligned}$$

So the cost of the flow corresponding to Δ is always less than the cost of the flow corresponding to Γ.

Next, we show that among the support sets covering all nonzero entries in X, MinCostFlow($G_{X,k,\lambda_{\min}}$) returns a support set with minimum support-EMD.

Since $X \in \mathcal{A}_{k,B}$, there is a $\Gamma \in \mathbb{A}_{k,B}$ with $\|X_\Gamma\|_1 = \|X\|_1$. Moreover, Ω is the support returned by MinCostFlow($G_{X,k,\lambda_{\min}}$), so we have $\|X_\Omega\|_1 = \|X\|_1$ and

$$-\|X_\Omega\|_1 + \lambda_0 \text{sEMD}(X_\Omega) \leq -\|X_\Gamma\| + \lambda_0 \text{sEMD}(X_\Gamma)\,. \tag{6.7}$$

So $\text{sEMD}(X_\Omega) \leq \text{sEMD}(X_\Gamma) \leq B$. Since X_Ω is also s-sparse in each column, $\Omega \in \mathbb{A}_{k,B}$. □

We now prove the main theorem. In order to simplify the derivation, we introduce the following shorthands. The intuition for the subscripts l and r is that the corresponding variables correspond to bounds on the optimal solution from the *left* and *right* in the plane of supports.

- $\Omega_l = \text{MinCostFlow}(G_{X,k,\lambda_l})$
- $\Omega_r = \text{MinCostFlow}(G_{X,k,\lambda_r})$
- $b_l = \text{sEMD}(X_{\Omega_l})$
- $b_r = \text{sEMD}(X_{\Omega_r})$
- $t_l = \|X - X_{\Omega_l}\|_1$
- $t_r = \|X - X_{\Omega_r}\|_1$

Theorem 35. *Let Ω be the support returned by* TailApprox*(X, k, B, d, δ). Let OPT be the tail approximation error of the best support with EMD-budget at most B, i.e. $OPT = \min_{\Gamma \in \mathbb{A}_{k,B}} \|X - X_\Gamma\|_1$. Then at least one of the following two guarantees holds for Ω:*

- $B \leq \text{sEMD}(X_\Omega) \leq dB$ *and* $\|X - X_\Omega\|_1 \leq OPT$.
- $\text{sEMD}(X_\Omega) \leq B$ *and* $\|X - X_\Omega\|_1 \leq (1+\delta)\frac{d}{d-1} OPT$.

Proof. If $X \in \mathcal{A}_{k,B}$, TailApprox returns X, which means both guarantees hold (see lemma 34). If $X \notin \mathcal{A}_{k,B}$ but X is s-sparse in each column, the following call MinCostFlow($G_{X,k,\lambda_{\min}}$) returns an Ω covering all nonzero entries in X and using the minimum amount of support-EMD among all supports covering all nonzero entries in X (again, see lemma 34). However, since $X \notin \mathcal{A}_{k,B}$, we have $\text{sEMD}(X_\Omega) > B$ and hence $\Omega \notin \mathbb{A}_{k,B}$. So TailApprox does not terminate early for $X \notin \mathcal{A}_{k,B}$. In the following, we assume that $X \notin \mathcal{A}_{k,B}$ and hence $OPT \geq x_{\min}$.

In the binary search, we maintain the invariant that $b_l \leq B$ and $b_r > B$. Note that this is true before the first iteration of the binary search due to our initial choices of λ_r and λ_l[1]. Moreover, our update rule maintains the invariant.

We now consider the two ways of leaving the binary search. If we find an Ω with $\text{sEMD}(X_\Omega) \geq B$ and $\text{sEMD}(X_\Omega) \leq dB$, this also means $\|X - X_\Omega\|_1 \leq OPT$ due to convexity of the convex hull of supports. Hence the first guarantee in the theorem is satisfied.

If $\lambda_l - \lambda_r \leq \varepsilon$, we return $\Omega = \Omega_l$ and hence the $\text{sEMD}(X_\Omega) \leq B$ part of the second guarantee is satisfied. We now prove the bound on $\|X - X_\Omega\|_1 = t_l$. Figure 6-3 illustrates the geometry of the following argument.

Let P_{OPT} be the point corresponding to a support with tail error OPT and minimum support-EMD, i.e. the optimal solution. Since the point (b_r, t_r) was the result of the corresponding MinCostFlow(G_{X,k,λ_r}), P_{OPT} has to lie above the line with slope $-\lambda_r$ through (b_r, t_r). Moreover, P_{OPT} has to have x-coordinate less than B. We can use these facts to establish the following bound on OPT:

$$OPT \geq t_r + \lambda_r(b_r - B)\,. \tag{6.8}$$

Let λ be the slope of the line through (t_r, b_r) and (t_l, b_l), i.e.

$$\lambda = \frac{t_r - t_l}{b_r - b_l}\,. \tag{6.9}$$

[1]Intuitively, our initial choices make the support-EMD either very cheap or very expensive compared to the tail approximation error.

Then we have $\lambda_r \leq -\lambda \leq \lambda_l$. Together with $\lambda_l - \lambda_r \leq \varepsilon$ this gives

$$\lambda_r \geq \frac{t_l - t_r}{b_r - b_l} - \varepsilon\,. \tag{6.10}$$

We now use this bound on λ_r to derive a bound on OPT:

$$\begin{aligned}
OPT &\geq t_r + \lambda_r(b_r - B) && (6.11)\\
&\geq t_r + (b_r - B)\frac{t_l - t_r}{b_r - b_l} - \varepsilon(b_r - B) && (6.12)\\
&\geq t_r + (b_r - B)\frac{t_l - t_r}{b_r} - \varepsilon(wh^2 - B) && (6.13)\\
&\geq t_l - \frac{B}{b_r}(t_l - t_r) - \varepsilon(wh^2 - B) && (6.14)\\
&\geq t_l - \frac{B}{dB}t_l - \frac{x_{\min}}{wh^2}\delta(wh^2 - B) && (6.15)\\
&\geq \frac{d-1}{d}t_l - x_{\min}\delta && (6.16)\\
&\geq \frac{d-1}{d}t_l - \delta\,OPT\,. && (6.17)
\end{aligned}$$

And hence

$$t_l \leq (1+\delta)\frac{d}{d-1}\,OPT\,, \tag{6.18}$$

which shows that the second guarantee of the theorem is satisfied. □

We now analyze the running time of TailApprox. For simplicity, we assume that $h = \Omega(\log w)$, i.e. the matrix X is not very wide and low.

Theorem 36. TailApprox *runs in* $O(snh \log \frac{\|X\|_1 n}{x_{\min}\delta})$ *time.*

Proof. We can solve our instances of the min-cost flow problem by finding s augmenting paths because all edges and nodes have unit capacity. Moreover, $G_{X,k,\lambda}$ is a directed acyclic graph, so we can compute the initial node potentials in linear time. Each augmenting path

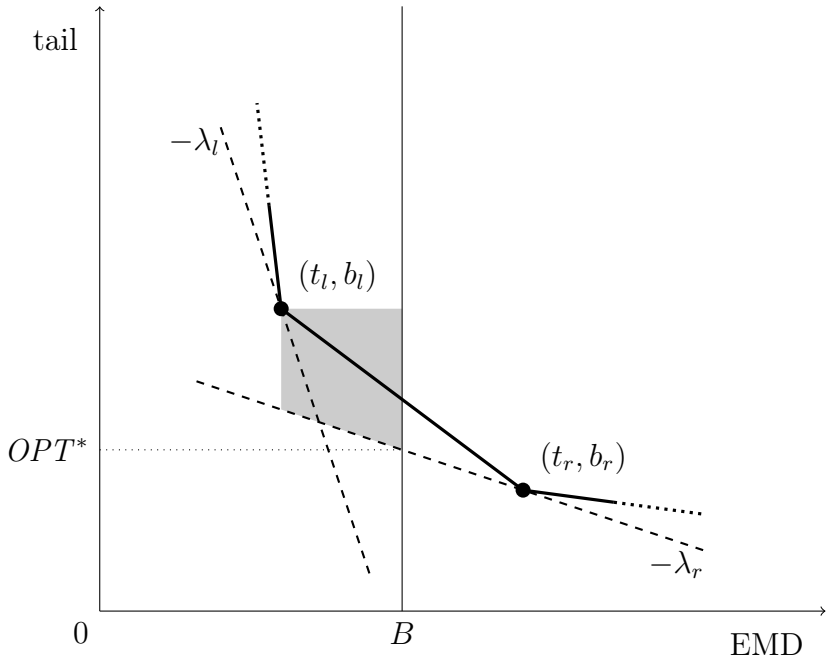


Figure (6-3): The local region of the convex hull during the binary search. The point (t_l, b_l) corresponds to MinCostFlow(G_{X,k,λ_l}) and (t_r, b_r) corresponds to MinCostFlow(G_{X,k,λ_r}). All support points between the two points have to lie above the dashed lines with slopes $-\lambda_l$ and $-\lambda_r$. We also use the fact that the optimal support has to have a x-coordinate between b_l and B and a y-coordinate below t_l. In the proof of theorem 35 we use only the line corresponding to λ_r, which leaves the gray area. As a result, OPT^* is a lower bound on OPT.

can then be found with a single run of Dijkstra's algorithm, which costs $O(wh \log(wh) + wh^2)$ time.

The binary search takes at most

$$\log \frac{\|X\|_1}{\epsilon} = \log \frac{\|X\| nh}{x_{\min} \delta} \tag{6.19}$$

iterations. Using $n = wh$ gives the stated running time. □

We can now conclude that TailApprox satisfies the definition of a tail approximation oracle (see definition 16).

Corollary 37. *Let $\delta \geq 0$ and $d = 1 + \frac{1}{c^2/(1+\delta) - 1}$. Then* TailApprox *is a $(c, (k, dB))$-tail approximation algorithm.*

Proof. Let $X' \in \mathbb{R}^{h \times w}$ with $X'_{i,j} = X^2_{i,j}$. We run TailApprox with parameters X', k, B, d and δ. Let Ω be the support returned by TailApprox.

The tail approximation guarantee follows directly from theorem 35. Note that we can not control which of the two guarantees the algorithm returns. However, in any case we have $\text{sEMD}(X_\Omega) \leq dB$, so $\Omega \in \mathbb{A}_{k,dB}$.

Moreover, note that $(1 + \delta)\frac{d}{d-1} = c^2$. So we get the following ℓ_1-guarantee:

$$\|x' - x'_\Omega\|_1 \leq c^2 \min_{x^* \in \mathcal{A}_{k,B}} \|x' - x^*\|_1 , \tag{6.20}$$

which directly implies

$$\|x - x_\Omega\| \leq c \min_{x^* \in \mathcal{A}_{k,B}} \|x - x^*\| . \tag{6.21}$$

□

Chapter 7

Compressive sensing with the EMD model

We now bring the results from the previous chapters together: we show that our EMD-model can be used for model-based compressive sensing and that doing so significantly reduces the number of measurements necessary for recovering signals in our model. The main result is the theoretical guarantee which builds on AMIHT and the head and tail approximation algorithms for the EMD model. Moreover, we also present results from experiments which empirically validate the performance of our model. The algorithms used in the experiment are from an earlier paper [14] that predates the main theoretical developments for the EMD model. Therefore, the experiments are based on slightly different algorithms than those described in the previous chapters.

7.1 Theoretical guarantee

The main theoretical result is the following recovery guarantee.

Theorem 38. *Let $x \in \mathcal{A}_{k,B}$ be an arbitrary signal in the EMD model with dimension $n = wh$ and $B = O(k)$. Let $\Phi \in \mathbb{R}^{m \times n}$ be a measurement matrix with i.i.d. Gaussian entries and let $y \in \mathbb{R}^m$ be a noisy measurement vector, i.e. $y = \Phi x + e$ with arbitrary $e \in \mathbb{R}^m$. Then we can*

recover a signal approximation $\widehat{x} \in \mathcal{A}_{k,2B}$ *satisfying*

$$\|x - \widehat{x}\|_2 \leq C\|e\|_2 \tag{7.1}$$

for some constant C *from* $O(k \log\log \frac{k}{w})$ *measurements.*

Moreover, the recovery algorithm runs in time $O(snh \log \frac{\|x\|}{\|e\|}(B + d\log(\|x\|n)))$ *if* x, Φ *and* e *are specified with at most* d *bits of precision.*

Proof. We use AMIHT, TailApprox and HeadApprox. The output $\widehat{x}$ of AMIHT is the result of TailApprox with parameters k, B and c_T. As shown in corollary 21, $c_T = 1.5$ suffices for geometric convergence of AMIHT. With this choice of c_T, TailApprox is a $(1.5, (k, 2B))$-tail approximation algorithm (corollary 37 and choosing $\delta = 0.1$). Hence $\widehat{x} \in \mathcal{A}_{k,2B}$.

We now show that $m = O(k \log\log \frac{k}{w})$ suffices for Φ to have the (δ, t)-model-RIP for $t = f_H(p \oplus f_T(p)) \oplus p \oplus f_T(p)$. Note that for the EMD-model, we have $p = (k, B)$ and $\mathcal{A}_{p \oplus q} \subseteq \mathcal{A}_{p+q}$ (theorem 13), so we are interested in $t = f_H(p + f_T(p)) + p + f_T(p)$.

For $c_H = 0.95$ (corollary 21), HeadApprox is a $(0.95, (38k, 38\lceil H_s \rceil B))$-head approximation oracle. So we get $t = (k', B')$ with $k' = \Theta(k)$ and $B' = \Theta(B \log s)$.

Using the sampling bound from theorem 14 we get

$$\log a_{k',B'} = O(k' \log \frac{B'}{k'}) \tag{7.2}$$

$$= O(k \log\log s) \tag{7.3}$$

$$= O(k \log\log \frac{k}{w}) \,. \tag{7.4}$$

Combining this with fact 4 shows that $m = O(k \log\log \frac{k}{w})$ is sufficient for Φ to have the desired (δ, t)-model-RIP for fixed δ.

Therefore, all assumptions in the analysis of AMIHT are satisfied. Using corollary 21 with a sufficiently large t (e.g. $25 \log \frac{\|x\|}{\|e\|}$) gives the desired approximation error bound with $C = 152$.

The running time bound follows directly from this choice of t, theorem 28 and theorem 36. □

Note that this measurement bound $m = O(k \log\log \frac{k}{w})$ is a significant improvement over the standard compressive sensing measurement bound $m = O(k \log \frac{n}{k})$.

7.2 Experiments

Before fully developing the theory of the EMD-model, we conducted several experiments to test the performance of our model. For these experiments, we used algorithms that are conceptually similar to AMIHT and TailApprox but do not come with recovery guarantees. Due to their similarity, we do not explain the algorithms here in detail. Appendix A contains the relevant pseudocode. The following results show that the EMD-model enables recovery from far fewer measurements than standard compressive sensing. Note that the experiments are not an evaluation of the algorithms introduced in previous chapters (AMIHT, HeadApprox and TailApprox).

The main goal of using model-based compressive sensing (or even compressive sensing in general) is reducing the number of measurements necessary for successful recovery of the signal. Hence we demonstrate the advantage of our model by showing that signals in the EMD model can be recovered from fewer measurements than required by standard compressive sensing methods. In each experiment, we fix a signal x in our model, generate a random Gaussian measurement matrix Φ and then feed Φx as input into a range of recovery algorithms. We compare the popular general sparse recovery algorithms CoSaMP [12] and IHT [2] with our variants EMD-CoSaMP and EMD-IHT (see appendix A).

Figure 7-1 shows the result of a single experiment with a signal of size $w = 10$, $h = 100$. The column sparsity is $s = 2$, which gives a total sparsity of $k = 20$. The support-EMD of the signal is 18 and we run our algorithms with $B = 20$. The figure shows the result of recovering the signal from $m = 80$ measurements. While standard CoSaMP fails, our variant recovers the signal perfectly.

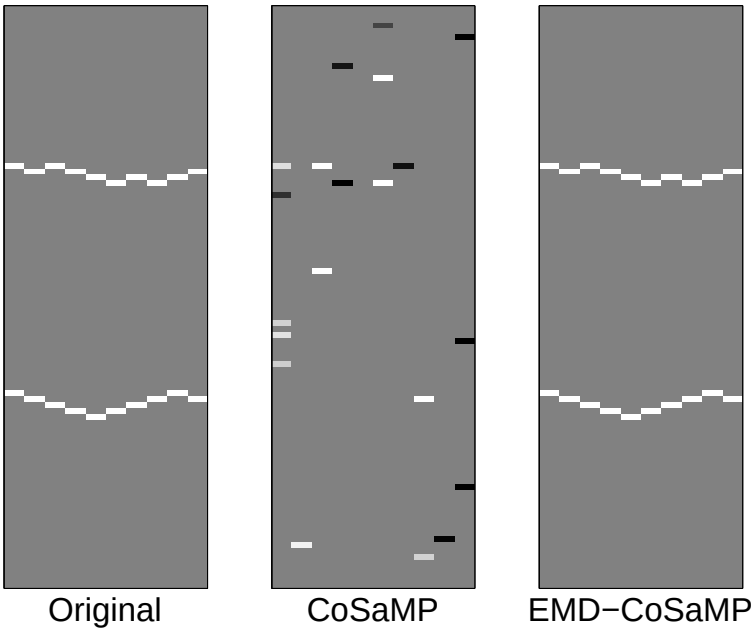


Figure (7-1): We compare the recovery performance of CoSaMP and EMD-CoSaMP on the same signal and measurement matrix. (left) Original image with parameters $h = 100$, $w = 10$, $k = 2$, $B = 20$, $m = 80$. (center) Recovery result using CoSaMP. (right) Recovery result using EMD-CoSaMP. CoSaMP fails to recover the signal while our algorithm recovers it perfectly.

We also study the trade-off between the number of measurements and the probability of successful recovery. Here, we define successful recovery in terms of the relative ℓ_2-error. We declare an experiment as successful if the signal estimate $\widehat{x}$ satisfies $\frac{\|x-\widehat{x}\|}{\|x\|} \leq 0.05$. We vary the number of measurements from 60 to 150 in increments of 10 and run 100 independent trials for each value of m. For each trial, we use the same signal and a new randomly generated measurement matrix (so the probability of recovery is with respect to the measurement matrix). The results in figure 7-2 show that our algorithms recover the signal successfully from far fewer measurements than the unmodified versions of CoSaMP and IHT.

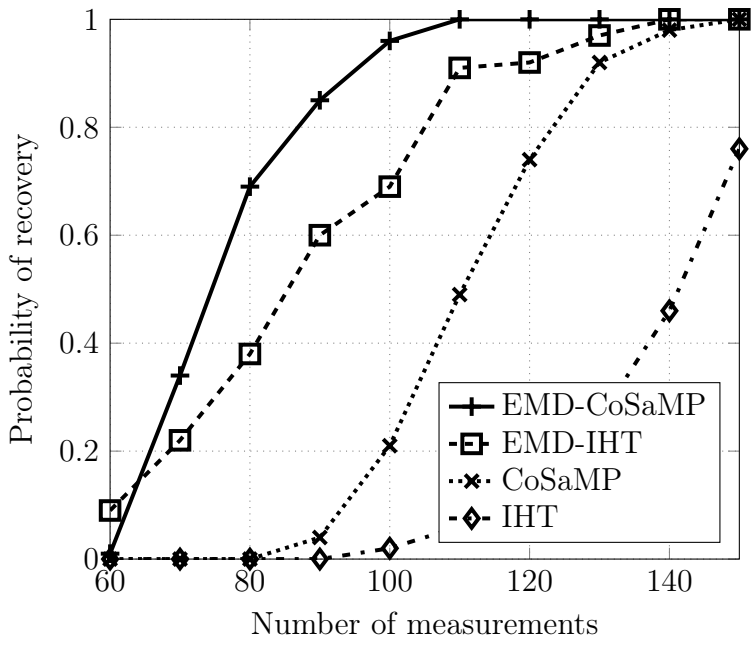


Figure (7-2): Comparison of several reconstruction algorithms. The signal is the same as in Figure 7-1. We declare the signal as recovered successfully if the signal estimate is within a relative ℓ_2-distance of 5% of the original signal. The probability of recovery is with respect to the measurement matrix and averaged over 100 trial runs. The recovery algorithms using the EMD model have a higher probability of recovery than standard algorithms.

Chapter 8

Conclusion

We have proposed the EMD-model, a structured sparsity model for signals that – when seen as a matrix – have similar support in neighboring columns. Due to its generality, the EMD-model is potentially useful in a wide range of settings. Moreover, the EMD-model is a generalization of existing approaches to measuring how much the support of a set of vectors changes. The EMD budget parameter gives the model additional flexibility: varying the EMD budget allows us to interpolate between block sparsity and a simple sparsity constraint on each column.

In order to use the EMD-model in compressive sensing, we have extended the model-based compressive sensing framework to approximate projection oracles. This extension allows us to use model-based compressive sensing in cases where a optimal model projections are intractable but approximations are still feasible. For the EMD-model, we have introduced corresponding head and tail approximation algorithms that allow us to recover signals in our model from noisy compressive measurements. For reasonable values of the EMD budget, we can show that we need asymptotically fewer measurements than standard compressive sensing. Our experiments show that this advantage also holds in practice.

8.1 Future work

We now outline directions for future work based on the EMD-model.

Approximately sparse signals Currently, our recovery guarantees only apply to measurements of the form $y = \Phi x + e$ where x is in the EMD-model. In practice, signals are typically not exactly k-sparse but rather compressible, i.e. the signal has a small number of large components and then decays quickly. This also means that a compressible signal x is well-approximated by the closest x' in the model and consequently finding the best approximation in the model can still result in a useful recovery result. Extending the EMD-model and AMIHT to model-compressible signals is the subject of further work.

Fault detection We have preliminary results indicating that the EMD-model is useful for pattern recognition tasks such as fault detection in seismic processing. Images of the subsurface usually consist of horizontal layers that sometimes have shear-like discontinuities, so-called faults. Since faults often trap hydrocarbons, detecting faults is an important goal in seismic processing. In our experiments, the tail projection algorithm was able to identify the horizontal layers as paths through an image, which allows us to detect faults as large local spikes in the EMD between neighboring columns.

Better head approximation algorithms Currently, our head approximation algorithm incurs an increase in the EMD budget by a factor of $\Theta(\log s)$, which leads to the $\log\log\frac{k}{w}$ term in the necessary number of measurements $O(k\log\log\frac{k}{w})$. A $(c, (O(k), O(B)))$-head approximation algorithm would bring the sample complexity in the $B = O(k)$ regime down to $O(k)$, which is optimal for a k-sparse signal.

Appendix A

Experimental algorithms

The algorithms here are used for the experiments in section 7.2. While they do not come with theoretical guarantees, EMDFlow can be seen as a variant of TailApprox. EMD-CoSaMP and EMD-IHT are the standard variants of CoSaMP [12] and IHT [2] form model-based compressive sensing [1]. However, we use them here with an approximate model projection algorithm instead of an optimal model projection algorithm.

The min-cost flow routine in EMDFlow is implemented with the LEMON library [7].

Algorithm 6 Approximate model projection algorithm

function EMDFlow(X, k, B)
  for all $(i, j) \in [h] \times [w]$ **do**
    $X'_{i,j} \leftarrow X^2_{i,j}$
  $\lambda_l \leftarrow 0$
  $\lambda_h \leftarrow 1$
  repeat
    $\lambda_h \leftarrow 2\lambda_h$
    $\Omega \leftarrow \text{MinCostFlow}(G_{X',k,\lambda_h})$
  until $\text{sEMD}(X'_\Omega) \leq B$
  repeat
    $\lambda_m \leftarrow \frac{\lambda_h+\lambda_l}{2}$
    $\Omega \leftarrow \text{MinCostFlow}(G_{X',k,\lambda_m})$
    if $\text{sEMD}(X'_\Omega) > B$ **then**
      $\lambda_l \leftarrow \lambda_m$
    else
      $\lambda_h \leftarrow \lambda_m$
  until $\lambda_h - \lambda_l \leq \epsilon_\lambda$
  $\Omega \leftarrow \text{MinCostFlow}(G_{X',k,\lambda_h})$
  return X_Ω

Algorithm 7 EMD-model version of CoSaMP

function EMD-CoSaMP(Φ, y, k, B)

$x^0 \leftarrow 0$

$r \leftarrow y$

$i \leftarrow 0$

while not converged **do**

$i \leftarrow i + 1$

$e \leftarrow \Phi^T r$

$\Omega \leftarrow \text{supp}(\text{EMDFlow}(e, 2k, 2B))$

$\Gamma \leftarrow \Omega \cup \text{supp}(x^{i-1})$

$z_\Gamma \leftarrow \Phi_\Gamma^\dagger y$

$z_{\Gamma^C} \leftarrow 0$

$x^i \leftarrow \text{EMDFlow}(z, k, B)$

$r \leftarrow y - \Phi x^i$

return x^i

Algorithm 8 EMD-model version of IHT

function EMD-IHT(Φ, y, k, B)

$x^0 \leftarrow 0$

$i \leftarrow 0$

while not converged **do**

$x^{i+1} \leftarrow \text{EMDFlow}(x^i + \Phi^T(y - \Phi x^i), k, B)$

$i \leftarrow i + 1$

return x^i

Appendix B

Counterexample for IHT with tail approximation oracles

In order to present a simple explanation, we look at the standard k-sparse compressive sensing setting[1]. Let $a \in \mathbb{R}^n$ and let $T'_k(x)$ be a tail approximation oracle with the following guarantee:

$$\|a - T'_k(a)\| \leq c\|a - T_k(a)\| \tag{B.1}$$

where c is an arbitrary constant and T_k is an optimal projection oracle, i.e. returns a k-sparse vector with the k largest components of a. We now show that an "adversarial" tail approximation oracle with $T'_k(a) = 0$ satisfies this definition for inputs a occurring in the first iteration of IHT with high probability. This shows that IHT cannot make progress and consequently we cannot recover the signal.

Recall that IHT with tail approximation oracle T' iterates

$$x^{i+1} \leftarrow T'_k(x^i + \Phi^T(y - \Phi x^i))\,, \tag{B.2}$$

which in the first iteration gives

$$x^1 \leftarrow T'_k(\Phi^T y)\,. \tag{B.3}$$

[1]This counterexample is taken from a question on the problem set for the class 6.893 (Spring 2013), which in turn was derived from research that lead to this thesis.

We now look at the case that the signal x is 1-sparse with $x_1 = 1$ and $x_i = 0$ for $i \neq 1$, i.e. $x = e_1$. Given a measurement matrix Φ with $(\delta, O(1))$-RIP for small δ, IHT needs to perfectly recover x from Φx. Matrices $\Phi \in \mathbb{R}^{m \times n}$ with $\Phi_{i,j} = \pm\sqrt{m}$ i.i.d. uniformly at random are known to have this RIP for $m = O(\log n)$ with high probability (so called Rademacher matrices). We prove that our adversarial tail approximation oracle satisfies the tail approximation guarantee for its input $a = \Phi^T \Phi e_1$ with high probability. Hence $x^1 = x^0 = 0$ and IHT cannot make progress. Intuitively, in spite of the RIP, the tail of a contains so much "noise" that the adversarial tail approximation oracle does not have to find a good sparse support for a and can get away with simply returning 0.

Consider the components of the vector $a \in \mathbb{R}^n$: a_i is the inner product of the first column of Φ with the i-th column of Φ. We have $a_1 = 1$ and $-1 \leq a_i \leq 1$ for $i \neq 1$. Hence $T_k(a) = e_1$ is an optimal projection and so $\|a - T_k(a)\|^2 = \|a\|^2 - 1$ We want to show that $\|a\|^2 \geq \frac{c^2}{c^2-1}$. This statement is equivalent to

$$\|a\|^2 \leq c^2(\|a\|^2 - 1)\,, \tag{B.4}$$

which then implies that T' satisfies the desired guarantee

$$\|a - T'_k(a)\|^2 \leq c^2 \|a - T_k(a)\|^2\,. \tag{B.5}$$

Note that $\|a\|^2 = 1 + \sum_{i=2}^n a_i^2$, where the a_i are independent. For $i \neq 1$, each a_i is the sum of m independent $\pm\frac{1}{m}$ random variables ($p = 1/2$). We have $\mathbb{E}[a_i^2] = \frac{1}{m}$. We can use Hoeffding's inequality to show that $\sum_{i=2}^n a_i^2$ does not deviate from its mean $\frac{n-1}{m}$ by more than $O(\sqrt{n \log n})$ with high probability. Since $m = O(\log n)$, this shows that

$$\|a\|^2 = 1 + \sum_{i=2}^{n} a_i^2 \geq \frac{c^2}{c^2 - 1} \tag{B.6}$$

with high probability for sufficiently large n.

Bibliography

[1] R.G. Baraniuk, V. Cevher, M.F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56(4):1982 –2001, 2010.

[2] T. Blumensath and M.E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2009.

[3] E. Candes, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[4] E.J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[5] E.J. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[6] E.J. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[7] B. Dezs, A. Jüttner, and P. Kovács. LEMON: an open source C++ graph template library. *Electron. Notes Theor. Comput. Sci.*, 264(5):23–45, 2011.

[8] D.L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[9] A. Gilbert and P. Indyk. Sparse recovery using sparse matrices. *Proceedings of the IEEE*, 98(6):937–947, 2010.

[10] Piotr Indyk and Eric Price. K-median clustering, model-based compressive sensing, and sparse recovery for Earth Mover Distance. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC '11, pages 627–636, 2011.

[11] E. Levina and P. Bickel. The Earth Mover's distance is the Mallows distance: some insights from statistics. In *Proceedings of the eighth IEEE International Conference on Computer Vision, ICCV 2001*, volume 2, pages 251–256, 2001.

[12] D. Needell and J.A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

[13] Y. Rubner, C. Tomasi, and L.J. Guibas. The Earth Mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[14] Ludwig Schmidt, Chinmay Hegde, and Piotr Indyk. The constrained earth mover distance model, with applications to compressive sensing. In *10th International Conference on Sampling Theory and Applications*, SampTA '13, 2013.

[15] J.A. Tropp and S.J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, 2010.

[16] N. Vaswani and Wei Lu. Modified-CS: Modifying compressive sensing for problems with partially known support. *IEEE Transactions on Signal Processing*, 58(9):4595–4607, 2010.