High order low-bit Sigma-Delta quantization for fusion frames and algorithms for hypergraph signal processing

By

Zhen Gao

Dissertation
Submitted to the Faculty of the
Graduate School of Vanderbilt University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY
in
Mathematics

June 30, 2021

Nashville, Tennessee

Approved:
Alexander M Powell, Ph.D.
Akram Aldroubi, Ph.D.
Mark Does, Ph.D.
Douglas Hardin, Ph.D.
Larry Schumaker, Ph.D.

To the memory of my grandfather

ACKNOWLEDGMENTS

First and foremost, I would like to say great thanks to my supervisor Prof. Alexander Powell, for his endless support, guidance, and encouragement. I could still remember the first moment when he agreed to be my supervisor. It is my honor to work with him for the past four years. He is like a beacon for both my study and my life. For my research, Alex and I had meetings every week. He taught me how to think and work as a mathematician. Without him, I couldn't achieve what I get today. As an international student far away from home, sometimes, I felt that he is more like a father than a supervisor. He gave me tremendous help outside of the research as well when I have difficulties.

I am also very grateful to Prof. Akram Aldroubi, Prof. Douglas Hardin, Prof. Larry Schumaker, and Prof. Mark Does, for their serves as my dissertation committee members and their comments and suggestions on my research work. In addition, I want to thank Dr. Dylan Domel-White, Dr. Jonathan Ashbrock, and other participants for the weekly seminars where I learnt a lot. I also want to thank all my friends at Vanderbilt, especially, Wenhao Wang, Jiawei Han and Sifan Yu. They made the time I spent in Nashville enjoyable.

Most importantly, none of my work would have been accomplished without the love of my family. I must thank my parents and younger brother. They provided onward motivation throughout my past journey in life. Finally, I would like to express my deepest memories to my grandfather, who gave me countless love during my childhood. He passed away earlier this year, but I couldn't attend his funeral due to covid travel restriction.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

Introduction: High-Order Low-Bit Quantization on Fusion Frames

Fusion frames provide a mathematical setting for representing signals in terms of projections onto a redundant collection of closed subspaces. Fusion frames were introduced in [11] as a tool for data fusion, distributed processing, and sensor networks, e.g., see [12, 13]. In this work we consider the question of how to perform quantization, i.e., analog-to-digital conversion, on a collection of fusion frame measurements.

Our motivation comes from the stylized sensor network in [24]. Suppose that one seeks to measure a signal $x \in \mathbb{R}^d$ over a large environment using a collection of remotely dispersed sensors s_n that are constrained by limited power, limited computational resources, and limited ability to communicate. Each sensor is only able to make local measurements of the signal, and the goal is to communicate the local measurements to a distantly located base station where the signal x can be accurately estimated. The sensor network is modeled as a fusion frame and is physically constrained in the following manner:

- Each local measurement y_n is a projection of x onto a subspace W_n associated to s_n.
- Each sensor has knowledge of the proximities W_n of a small number of nearby sensors.
- Each sensor can communicate analog signals to a small number of nearby sensors.
- Each sensor can transmit a *low-bit* signal to the distant base station.
- The base station is relatively unconstrained in power and computational resources.

Mathematically, the above sensor network problem can be formulated as a quantization problem for fusion frames, e.g., [24]. Suppose that $\{W_n\}_{n=1}^N$ are subspaces of $\mathbb{R}^d$ and suppose that each $\mathcal{A}_n \subset W_n$ is a finite quantization alphabet. Given $x \in \mathbb{R}^d$ and the orthogonal projections $y_n = P_{W_n}(x)$, we seek an efficient algorithm for rounding the continuum-valued measurements $y_n \in W_n$ to finite-valued $q_n \in \mathcal{A}_n$. This rounding process is called quantization and it provides a digital representation of y_n through q_n. Here, q_n corresponds to the low-bit signal that the sensor s_n

transmits to the central base station. We will focus on the case where the q_n are computed sequentially and we allow the algorithm to be implemented with a limited amount of memory. The memory variables correspond to the analog signals that sensors communicate to other nearby sensors. Finally, once the quantized measurements $\{q_n\}_{n=1}^N$ have been computed, we seek a reconstruction procedure for estimating x; this corresponds to the role of the base station.

We address the above problem with a new low-bit version of Sigma-Delta ($\Sigma\Delta$) quantization for fusion frames. Sigma-Delta quantization is a widely applicable class of algorithms for quantizing oversampled signal representations. Sigma-Delta quantization was introduced in [23], underwent extensive theoretical development in the engineering literature [18], and has been widely implemented in the circuitry of analog-to-digital converters [30]. Starting with [14], the mathematical literature provided approximation-theoretic error bounds for Sigma-Delta quantization in a variety of settings, starting with bandlimited sampling expansions [14, 15, 19, 20, 22, 32]. The best known constructions yield error bounds decaying exponentially in the bit budget [19, 15], which is also the qualitative behavior that one encounters when quantizing Nyquist rate samples at high precision. The rate of this exponential decay, however, is provably slower for Sigma-Delta [28].

Subsequently, Sigma-Delta was generalized to time-frequency representations [33] and finite frame expansions [4]. As it turned out, the direct generalization to frames has significant limitations unless the frame under consideration has certain smoothness properties [8]. A first approach to overcome this obstacle was to recover signals using a specifically designed dual frame, the so-called Sobolev Dual [6, 26]; this approach has also been implemented for compressed sensing [21, 27, 16]. Another class of dual frames that sometimes outperform Sobolev duals are the so-called beta duals [10]. In the context of compressed sensing, Sobolev duals have inspired a convex optimization approach for recovery [31], which has also been analyzed for certain structured measurements [17].

Since fusion frames employ vector-valued measurements, our approach in Definition 3.1.1 may be viewed as a vector-valued analogue of Sigma-Delta quantization. For perspective, we point out related work on $\Sigma\Delta$ quantization of finite frames with complex alphabets on a lattice [5], hexagonal $\Sigma\Delta$ modulators for power electronics [29], and dynamical systems motivated by error diffusion in digital halftoning [1, 2, 3].

The work in [24] constructed and studied stable analogues of Sigma-Delta quantization in the setting of fusion frames. The first order $\Sigma\Delta$ algorithms in [24] were stably implementable using

very low-bit quantization alphabets. Unfortunately, the higher order $\Sigma\Delta$ algorithms in [24] required large quantization alphabets for stability to hold. Stable high order $\Sigma\Delta$ algorithms are desirable since quantization error generally decreases as the order of a Sigma-Delta algorithm increases, e.g., [14]. The main contribution of the current work is that we provide the first examples of stable high-order *low-bit* Sigma-Delta quantizers for fusion frames.

Our results achieve the following:

- We construct stable rth order fusion frame Sigma-Delta (FF$\Sigma\Delta$) algorithms with quantization alphabets that use $\log_2(\dim(W_n) + 1)$-bits per subspace W_n, see Theorems 3.2.1 and 3.3.2. This resolves a question posed in [24].

- We provide numerical examples to show that the FF$\Sigma\Delta$ algorithm performs well when implemented together with a version of Sobolev dual reconstruction.

CHAPTER 2

Fusion frames and quantization

In this section, we provide background on fusion frames and quantization.

2.1 Fusion frames

Let $\{W_n\}_{n=1}^N$ be a collection of subspaces of $\mathbb{R}^d$ and let $\{c_n\}_{n=1}^N \subset (0,\infty)$ be positive scalar weights. The collection $\{(W_n, c_n)\}_{n=1}^N$ is said to be a *fusion frame* for $\mathbb{R}^d$ with *fusion frame bounds* $0 < A \leqslant B < \infty$ if

$$\forall x \in \mathbb{R}^d, \quad A\|x\|^2 \leqslant \sum_{n=1}^N c_n^2 \|P_{W_n}(x)\|^2 \leqslant B\|x\|^2.$$

If the bounds A, B are equal, then the fusion frame is said to be *tight*. If $c_n = 1$ for all $1 \leqslant n \leqslant N$, then the fusion frame is said to be *unweighted*. Given a fusion frame, the associated *analysis operator* $T : \mathbb{R}^d \longrightarrow \bigoplus_{n=1}^N W_n$ is defined by

$$T(x) = \{c_n P_{W_n}(x)\}_{n=1}^N.$$

The problem of recovering a signal $x \in \mathbb{R}^d$ from fusion frame measurements $y_n = P_{W_n}(x)$ is equivalent to finding a left inverse to the analysis operator. There is a canonical choice of left inverse which can be described using the synthesis operator and the frame operator.

The adjoint $T^* : \bigoplus_{n=1}^N W_n \longrightarrow \mathbb{R}^d$ of the analysis operator is called the *synthesis operator* and is defined by $T^*(\{f_n\}_{n=1}^N) = \sum_{n=1}^N c_n f_n$. The *fusion frame operator* $S : \mathbb{R}^d \to \mathbb{R}^d$ is defined by $S(x) = (T^*T)(x) = \sum_{n=1}^N c_n^2 P_{W_n}(x)$. It is well-known [11] that S is a positive self-adjoint operator. Moreover, $L = S^{-1}T^*$ is a left inverse to T since $LT = S^{-1}T^*T = S^{-1}S = I$. This provides the following canonical reconstruction formula for recovering $x \in \mathbb{R}^d$ from fusion frame measurements $y_n = P_{W_n}(x)$

$$\forall x \in \mathbb{R}^d, \quad x = LTx = S^{-1}Sx = \sum_{n=1}^N c_n^2 S^{-1}(y_n).$$

Although the canonical choice of left inverse $L = S^{-1}T^*$ is natural, other non-canonical left-inverses will be more suitable for the problem of reconstructing a signal x from quantized measurements.

2.2 Norms and direct sums

The direct sum space $\bigoplus_{n=1}^N W_n$ arises naturally in the study of fusion frames. In the interest of maintaining simple notation, we use the norm symbol $\|\cdot\|$ in different contexts throughout the paper to refer to norms on both Euclidean space and direct sum spaces, as well as operator norms on such spaces.

The following list summarizes different ways in which norm notation is used throughout the paper.

- If $x \in \mathbb{R}^d$ then $\|x\|$ denotes the Euclidian norm.
- If $H : \mathbb{R}^d \to \mathbb{R}^d$ then $\|H\| = \sup_{x\in\mathbb{R}^d} \frac{\|Hx\|}{\|x\|}$.
- If $w \in \bigoplus_{n=1}^N W_n$, then $\|w\| = \left(\sum_{n=1}^N \|w_n\|^2\right)^{1/2}$ and $\|w\|_\infty = \sup_{1\leqslant n\leqslant N} \|w_n\|$.
- If $G : \bigoplus_{n=1}^N W_n \to \bigoplus_{n=1}^N W_n$ then $\|G\| = \sup_{w\in\bigoplus W_n} \frac{\|Gw\|}{\|w\|}$.
- If $\mathcal{L} : \bigoplus_{n=1}^N W_n \to \mathbb{R}^d$ then $\|\mathcal{L}\| = \sup_{w\in\bigoplus W_n} \frac{\|\mathcal{L}w\|}{\|w\|}$.

2.3 Quantization

Let $x \in \mathbb{R}^d$ and suppose that $\{W_n\}_{n=1}^N$ are subspaces associated with a fusion frame for $\mathbb{R}^d$. For each $1 \leqslant n \leqslant N$, let $\mathcal{A}_n \subset W_n$ be a finite set which we refer to as a *quantization alphabet*, and let $Q_n : W_n \to \mathcal{A}_n$ be an associated vector quantizer with the property that

$$\forall w \in W_n, \quad \|w - Q_n(w)\| = \min_{q\in\mathcal{A}_n} \|w - q\|. \tag{2.1}$$

Memoryless quantization is the simplest approach to quantizing a set of fusion frame measurements $y_n = P_{W_n}(x), 1 \leqslant n \leqslant N$. Memoryless quantization simply quantizes each y_n to $q_n = Q_n(y_n)$. See [24] for basic discussion on the performance of memoryless quantization for fusion frames. This approach works well when the alphabets $\mathcal{A}_n$ are sufficiently fine and dense,

and is also suitable when the subspaces are approximately orthogonal. On the other hand, it is not very suitable for our sensor network problem which requires coarse low-bit alphabets and involves correlated subspaces W_n. We will see that Sigma-Delta quantization is a preferable approach.

We will make use of the low-bit quantization alphabets provided by the following lemma. These alphabets use $\log_2(\dim(W_n)+1)$ bits to quantize each subspace W_n. For perspective, in the scalar-valued setting, it is known that stable $\Sigma\Delta$ quantizers of arbitrary order can be implemented using a 1-bit quantization alphabet to quantize each scalar-valued sample, [14]. The vector-valued alphabet in the following lemma provides a suitable low-bit analogue of this for fusion frames.

Lemma 2.3.1. *Let W be an m-dimensional subspace of $\mathbb{R}^d$. There exists a set $\mathcal{U}(W) = \{u_k\}_{k=1}^{m+1}$ in W such that $\sum_{j=1}^{m+1} u_j = 0$, and each u_j is unit-norm $\|u_j\| = 1$, and*

$$\langle u_j, u_k \rangle = -\frac{1}{m}, \text{ for } j \neq k. \tag{2.2}$$

Moreover, if $\theta_0 = cos^{-1}(\frac{1}{m})$, then for every $w \in W\backslash\{0\}$, there exists k such that

$$angle(w, u_k) \leqslant \theta_0. \tag{2.3}$$

For references associated to Lemma 2.3.1, see the discussion following Lemma 1 in [24].

CHAPTER 3

Fusion frame Sigma-Delta quantization

3.1 Algorithm

Throughout this section we shall assume that $\{W_n\}_{n=1}^{\infty}$ are subspaces of $\mathbb{R}^d$ and that each finite collection $\{W_n\}_{n=1}^{N}$ is an unweighted fusion frame for $\mathbb{R}^d$ when $N \geqslant d$. We also assume that $\mathcal{A}_n = \mathcal{U}(W_n) \subset W_n$ is a set of $1 + \dim(W_n)$ vectors as in Lemma 2.3.1, and that $Q_n : W_n \to \mathcal{A}_n$ is a vector quantizer satisfying (2.1). Observe that by (2.1) and (2.3) one has that for arbitrary $w \in W_n$ with $\|w\| = 1$

$$\langle Q_n(w), w\rangle = 1 - \tfrac{1}{2}\|Q_n(w) - w\|^2 = 1 - \tfrac{1}{2}\min_{q\in\mathcal{A}_n}\|w - q\|^2 = \min_{q\in\mathcal{A}_n}\langle q, w\rangle \geqslant \tfrac{1}{m}. \tag{3.1}$$

Given $x \in \mathbb{R}^d$, we shall investigate the following algorithm for quantizing fusion frame measurements $y_n = P_{W_n}(x)$.

Definition 3.1.1 (Fusion frame Sigma-Delta algorithm)**.** *For each $n \geqslant 1$, fix operators $H_{n,j} : \mathbb{R}^d \to W_n$, $1 \leqslant j \leqslant L$. Initialize the state variables $v_0 = v_{-1} = \cdots = v_{1-L} = 0 \in \mathbb{R}^d$.*

The fusion frame Sigma-Delta algorithm *(FF$\Sigma\Delta$) takes the measurements $y_n = P_{W_n}(x)$ as inputs and produces quantized outputs $q_n \in \mathcal{A}_n$, $n \geqslant 1$, by running the following iteration for $n \geqslant 1$*

$$q_n = Q_n\left(y_n + \sum_{j=1}^{L} H_{n,j}(v_{n-j})\right), \tag{3.2}$$

$$v_n = y_n - q_n + \sum_{j=1}^{L} H_{n,j}(v_{n-j}). \tag{3.3}$$

The algorithm (3.2), (3.3) may be applied to an infinite stream of inputs, but, in practice, the algorithm will usually be applied to a fusion frame of finite size and will terminate after finitely many steps. The operators $H_{n,j}$ must be chosen carefully for the algorithm (3.2), (3.3) to perform well. We shall later focus on a specific choice of operators $H_{n,j}$ in Section 3.3, but to understand its

motivation, it is useful to first discuss reconstruction methods for the FF$\Sigma\Delta$ algorithm and to keep $H_{n,j}$ general for the moment.

The fusion frame Sigma-Delta algorithm must be coupled with a reconstruction procedure for recovering x from the quantized measurements $\{q_n\}$. We consider the following simple reconstruction method that uses left inverses of fusion frame analysis operators. At step N of the FF$\Sigma\Delta$ algorithm, one has access to the quantized measurements $\{q_n\}_{n=1}^N$. Henceforth, $q \in \bigoplus_{n=1}^N W_n$ will denote the element of $\bigoplus_{n=1}^N W_n$ whose nth entry is $q_n \in \mathcal{A}_n \subset W_n$. Since $\{W_n\}_{n=1}^N$ is a fusion frame with analysis operator $T = T_N$, let $\mathcal{L} = \mathcal{L}_N$ denote a left inverse of T, so that $\mathcal{L}Tx = x$ holds for all $x \in \mathbb{R}^d$. A specific choice of left inverse will be specified in Section 3.6, but for the current discussion let $\mathcal{L}$ be an arbitrary left inverse. After step N of the iteration (3.2), (3.3), we reconstruct the following $\widetilde{x}$ from $\{q_n\}_{n=1}^N$

$$\widetilde{x} = \widetilde{x}_N = \mathcal{L}q. \tag{3.4}$$

We now introduce notation that will be useful for describing the error $x - \widetilde{x}$ associated to (3.4). Let $v \in \bigoplus_{n=1}^N W_n$ denote the element of $\bigoplus_{n=1}^N W_n$ whose nth entry is $v_n \in W_n$. Let $\mathcal{I}_N : \bigoplus_{n=1}^N W_n \longrightarrow \bigoplus_{n=1}^N W_n$ denote the identity operator, and let $\mathcal{H}_N : \bigoplus_{n=1}^N W_n \longrightarrow \bigoplus_{n=1}^N W_n$ denote the $N \times N$ block operator with entries

$$\forall\ 1 \leqslant n, k \leqslant N, \quad (\mathcal{H}_N)_{n,k} = \begin{cases} H_{n,n-k}, & \text{if } 1 \leqslant n-k \leqslant L, \\ 0, & \text{otherwise.} \end{cases} \tag{3.5}$$

Note that (3.3) and (3.5) can be expressed in matrix form as $y - q = (\mathcal{I}_N - \mathcal{H}_N)v$. Combining this and (3.4) allows the error $x - \widetilde{x}$ to be expressed as

$$x - \widetilde{x} = \mathcal{L}_N(y - q) = \mathcal{L}_N(\mathcal{I}_N - \mathcal{H}_N)v. \tag{3.6}$$

We aim to design the operators $\mathcal{H}_N$ in the quantization algorithm and the reconstruction operator $\mathcal{L}_N$ so that the error $\|x - \widetilde{x}\| = \|\mathcal{L}_N(\mathcal{I}_N - \mathcal{H}_N)v\|$ given by (3.6) can be made quantifiably small. We pursue the following design goals:

- Select $\mathcal{H}_N$ so that the iteration (3.2), (3.3) satisfies a *stability condition* which controls the

norm of the state variable sequence v.

- Select $\mathcal{H}_N$ and $\mathcal{L}_N$ so that $\mathcal{L}_N(\mathcal{I}_N - \mathcal{H}_N)$ has small operator norm. This can be decoupled into separate steps. First, $\mathcal{H}_N$ is chosen to ensure that operator $\mathcal{I}_N - \mathcal{H}_N$ satisfies an *rth order condition* that expresses $\mathcal{I}_N - \mathcal{H}_N$ in terms of a generalized rth order difference operator. Secondly, $\mathcal{L}_N$ is chosen to be a *Sobolev left inverse* which is well-adapted to the operator $\mathcal{I}_N - \mathcal{H}_N$.

For the above points, Section 3.2 discusses stability, Section 3.3 discusses the rth order property, and Section 3.6 discusses reconstruction with Sobolev left inverses.

3.2 Stability

The following theorem shows that the fusion frame $\Sigma\Delta$ algorithm is stable in the sense that control on the size of inputs $\|y_n\|$ ensures control on the size of state variables $\|v_n\|$. For perspective, the stable higher order fusion frame $\Sigma\Delta$ algorithm in [24] requires relatively large alphabets $\mathcal{A}_n$.

Theorem 3.2.1. *Let $\{W_n\}_{n=1}^N$ be subspaces of $\mathbb{R}^d$ with $d^\star = \max_{1\leqslant n\leqslant N} \dim(W_n)$. Suppose that a sequence $\{y_n\}_{n=1}^N$ with $y_n \in W_n$ is used as input to the algorithm* (3.2)*,* (3.3)*.*

Suppose that $0 < \delta < \frac{1}{d^\star}$, and let

$$\alpha_1 = \sqrt{\frac{1 - \frac{2\delta}{d^\star} + \delta^2}{1 - (\frac{1}{d^\star})^2}} \quad \textit{and} \quad \alpha_2 = \frac{1}{2}\left(\left(\frac{1}{d^\star} - \delta\right) + \sqrt{\left(\frac{1}{d^\star} - \delta\right)^2 + 4}\right).$$

Suppose that $\alpha = \sup_{1\leqslant n\leqslant N} \sum_{j=1}^L \|H_{n,j}\|$ satisfies $1 < \alpha \leqslant \min\{\alpha_1, \alpha_2\}$, and let

$$C = \left(\frac{1}{d^\star} - \delta\right)\left(\frac{\alpha}{\alpha^2 - 1}\right).$$

If $\|y_n\| \leqslant \delta$ for all $1 \leqslant n \leqslant N$, then the state variables v_n in (3.2)*,* (3.3) *satisfy $\|v_n\| \leqslant C$ for all $1 \leqslant n \leqslant N$.*

Proof. Step 1. We begin by noting that the assumption $1 < \alpha \leqslant \min\{\alpha_1, \alpha_2\}$ is not vacuous. The condition $1 < \alpha_2$ directly follows from the assumption. If $d^\star = 1$ then $1 < \alpha_1 = \infty$ automatically

holds. For $d^\star > 1$, we rewrite

$$\alpha_1 = \sqrt{1 + \frac{(\frac{1}{d^\star} - \delta)^2}{1 - (\frac{1}{d^\star})^2}},$$

which is strictly larger than 1.

Step 2. Next, we note that $C \geqslant 1$. By the definition of C, it can be verified that $C \geqslant 1$ holds if and only if

$$\alpha^2 - \left(\frac{1}{d^\star} - \delta\right)\alpha - 1 \leqslant 0.$$

It follows that $C \geqslant 1$ holds if and only if

$$\frac{\left(\frac{1}{d^\star} - \delta\right) - \sqrt{\left(\frac{1}{d^\star} - \delta\right)^2 + 4}}{2} \leqslant \alpha \leqslant \frac{\left(\frac{1}{d^\star} - \delta\right) + \sqrt{\left(\frac{1}{d^\star} - \delta\right)^2 + 4}}{2}.$$

Since $\frac{\left(\frac{1}{d^\star} - \delta\right) - \sqrt{\left(\frac{1}{d^\star} - \delta\right)^2 + 4}}{2} < 0$ and $1 < \alpha_2 = \frac{\left(\frac{1}{d^\star} - \delta\right) + \sqrt{\left(\frac{1}{d^\star} - \delta\right)^2 + 4}}{2}$, the assumption $1 < \alpha \leqslant \alpha_2$ implies that $C \geqslant 1$, as required.

Step 3. We will prove the theorem by induction. The base case holds by the assumption that $v_0 = v_{-1} = \cdots = v_{1-L} = 0 \in \mathbb{R}^d$. For the inductive step, suppose that $n \geqslant 1$ and that $\|v_j\| \leqslant C$ holds for all $j \leqslant n - 1$.

Let $z_n = y_n + \sum_{j=1}^{L} H_{n,j}(v_{n-j})$, so that $v_n = z_n - q_n$ with $q_n = Q_n(z_n)$. If $z_n = 0$ then $\|v_n\| = \|q_n\| = 1 \leqslant C$, as required. So, it remains to consider $z_n \neq 0$.

When $z_n \neq 0$, let $\gamma_n = \frac{\langle z_n, q_n \rangle}{\|z_n\|}$. Combining the definition of $d^\star$ and the fact that the quantizer Q_n is scale invariant with (3.1), we obtain that $\gamma_n \geqslant \frac{1}{d^\star}$. Thus,

$$\begin{aligned} \|v_n\|^2 &= \|z_n\|^2 + \|q_n\|^2 - 2\|z_n\|\gamma_n \\ &\leqslant \|z_n\|^2 - \frac{2}{d^\star}\|z_n\| + 1. \end{aligned} \tag{3.7}$$

Since $\sum_{j=1}^{L} \|H_{n,j}\| \leqslant \alpha$, $\|y_n\| \leqslant \delta$, and $\|v_{n-j}\| \leqslant C$, the definition of z_n gives that

$$\|z_n\| \leqslant \|y_n\| + \sum_{j=1}^{L} \|H_{n,j}\| \, \|v_{n-j}\| \leqslant \delta + C\alpha. \tag{3.8}$$

Recall, we aim to show $\|v_n\| \leqslant C$. Let $f(t) = t^2 - (\frac{2}{d^\star})t + 1$. By (3.7) and (3.8), it suffices to prove

$$f([0, \alpha C + \delta]) \subseteq [0, C^2].$$

For that, we note that

$$\min\{f(t) : t \in [0, \alpha C + \delta]\} \geqslant f(1/d^\star) = 1 - (1/d^\star)^2 \geqslant 0.$$

and

$$\max\{f(t) : t \in [0, \alpha C+\delta]\} = \max\{f(0), f(\alpha C+\delta)\} = \max\{1, f(\alpha C+\delta)\} \leqslant \max\{C^2, f(\alpha C+\delta)\}.$$

Hence it only remains to show that $f(\alpha C + \delta) \leqslant C^2$.

Step 4. Consider the polynomial

$$p(x) = (\alpha^2 - 1)x^2 + 2\alpha\left(\delta - \frac{1}{d^\star}\right)x + \left(1 - \frac{2\delta}{d^\star} + \delta^2\right).$$

Since $1 < \alpha \leqslant \alpha_1$, it can be verified that the polynomial p has real roots $r_1 \leqslant r_2$. Since $\alpha > 1$, one has that $p(x) \leqslant 0$ for all $x \in [r_1, r_2]$. In particular, $p\left(\frac{r_1+r_2}{2}\right) \leqslant 0$. Moreover, it can be checked that

$$\frac{r_1 + r_2}{2} = \left(\frac{1}{d^\star} - \delta\right)\left(\frac{\alpha}{\alpha^2 - 1}\right) = C.$$

Thus, $p(C) \leqslant 0$.

Step 5. Note that

$$\begin{aligned} f(\alpha C + \delta) &= (\alpha C + \delta)^2 - \left(\frac{2}{d^\star}\right)(\alpha C + \delta) + 1 \\ &= \alpha^2 C^2 + 2\alpha\left(\delta - \frac{1}{d^\star}\right)C + \left(1 - \frac{2\delta}{d^\star} + \delta^2\right) \\ &= p(C) + C^2. \end{aligned}$$

Since $p(C) \leqslant 0$ holds by Step 4, it follows that $f(\alpha C + \delta) \leqslant C^2$, as required.

□

3.3 Rth order algorithms and feasibility

Classical scalar-valued rth order Sigma-Delta quantization expresses coefficient quantization errors as an rth order difference of a bounded state variable, e.g., see [14, 19]. In this section we describe an analogue of this for the vector-valued setting of fusion frames.

Let $D_N : \bigoplus_{n=1}^N W_n \longrightarrow \bigoplus_{n=1}^N W_n$ be the $N \times N$ block operator defined by

$$\forall\ 1 \leqslant n, k \leqslant N, \qquad (D_N)_{n,k} = \begin{cases} I, & \text{if } n = k, \\ -P_{W_n}, & \text{if } n = k+1, \\ 0, & \text{otherwise.} \end{cases} \tag{3.9}$$

Definition 3.3.1 (rth order algorithm)**.** *The fusion frame Sigma-Delta iteration* (3.2)*,* (3.3) *is an* rth order algorithm *if for every* $N \geqslant d$ *there exist operators* $G_N : \bigoplus_{n=1}^N W_n \longrightarrow \bigoplus_{n=1}^N W_n$ *that satisfy*

$$\mathcal{I}_N - \mathcal{H}_N = (D_N)^r G_N, \tag{3.10}$$

and

$$\sup_N \|G_N\| < \infty. \tag{3.11}$$

Moreover, given $\epsilon > 0$*, we say that* $\{(G_N, \mathcal{H}_N)\}_{N=d}^{\infty}$ *is* ϵ-feasible *if the operators* $H_{n,j}$ *that define* $\mathcal{H}_N$ *by* (3.5) *satisfy*

$$\sup_{n \geqslant 1} \sum_{j=1}^{L} \|H_{n,j}\| \leqslant 1 + \epsilon. \tag{3.12}$$

The rth order condition (3.10) should be compared with the scalar-valued analogue in equation (4.2) in [19], cf. [15]. For perspective, the condition (3.12) ensures that the stability result from Theorem 3.2.1 can be used. The rth order conditions (3.10), (3.11) will later be used in Section 3.6 to provide control on the quantization error $\|x - \widetilde{x}\|$.

We now show that it is possible to select $\mathcal{H}_N$ so that the low-bit fusion frame Sigma-Delta algorithm in (3.10), (3.11) is rth order and ϵ-feasible with small $\epsilon > 0$.

We make use of the following sequences n_j, d_j, h defined in [19]. The constructions have subsequently been improved in [25, 15], but we will work with the (suboptimal) first construction, as it allows for closed form expressions. Given $r \in \mathbb{N}$, define the index set $\mathbb{N}_r = \mathbb{N} \cap [1, r]$. Let $r, \sigma \in \mathbb{N}$ be fixed and define the sequences $\{n_j\}_{j=1}^r$ and $\{d_j\}_{j=1}^r$ by

$$n_j = \sigma(j-1)^2 + 1 \quad \text{and} \quad d_j = \prod_{i \in \mathbb{N}_r \setminus \{j\}} \frac{n_i}{n_i - n_j}. \tag{3.13}$$

Next, define $h \in \ell^1(\mathbb{N})$ by

$$h = \sum_{j=1}^{r} d_j \delta_{n_j}, \tag{3.14}$$

where $\delta_n \in \ell^1(\mathbb{N})$ is defined by $\delta_n(j) = 1$ if $j = n$, and $\delta_n(j) = 0$ if $j \neq n$. We will later use the property, proven in [19], that h satisfies

$$\|h\|_{\ell^1} < \cosh(\pi \sigma^{-1/2}). \tag{3.15}$$

Let $L = n_r$ and define the $N \times N$ block operator $\mathcal{H}_N$ using (3.5) with $1 \leqslant n \leqslant N$ and $1 \leqslant j \leqslant L$ and

$$H_{n,j} = \begin{cases} h_j P_{W_n} P_{W_{n-1}} \cdots P_{W_{n-j+1}}, & \text{if } n > j, \\ 0, & \text{otherwise.} \end{cases} \tag{3.16}$$

In the following result, we prove that the fusion frame Sigma-Delta algorithm with operators (3.16) is rth order and ϵ-feasible.

Theorem 3.3.2. *Fix $r \geqslant 2$. Given $\epsilon > 0$, if $\sigma \in \mathbb{N}$ is sufficiently large and if the operators $H_{n,j} : \bigoplus_{n=1}^N W_n \longrightarrow \bigoplus_{n=1}^N W_n$ are defined by* (3.13)*,* (3.14)*,* (3.16)*, then the fusion frame Sigma-Delta algorithm* (3.2)*,* (3.3) *is an rth order algorithm and is ϵ-feasible.*

The proof of Theorem 3.3.2 is given in Section 3.5.

3.4 Background lemmas

In this section, we collect background lemmas that are needed in the proof of Theorem 3.3.2. The following result provides a formula for the entries of the block operator $(D_N)^{-r}$.

Lemma 3.4.1. *Fix $r \geqslant 1$. If D_N is the $N \times N$ block operator defined by* (3.9) *then D_N is invertible and D_N^{-r} satisfies*

$$(D_N^{-r})_{i,j} = \begin{cases} \binom{r+i-j-1}{r-1}(P_{W_i}P_{W_{i-1}}\dots P_{W_{j+1}}) & \text{if } i > j \\ I & \text{if } i = j \\ 0 & \text{if } i < j. \end{cases} \tag{3.17}$$

Proof. The proof proceeds by induction. For the base case $r = 1$, a direct computation shows that

$$\forall\, 1 \leqslant i,j \leqslant N, \quad (D_N^{-1})_{i,j} = \begin{cases} P_{W_i}P_{W_{i-1}}\cdots P_{W_{j+1}} & \text{if } i > j \\ I & \text{if } i = j \\ 0 & \text{if } i < j. \end{cases}$$

For the inductive step, suppose that (3.17) holds. Using $(D_N^{-(r+1)})_{i,j} = \sum_{k=1}^{N}(D_N^{-r})_{i,k}(D_N^{-1})_{k,j}$, shows that $(D_N^{-(r+1)})_{i,i} = I$, and that if $i < j$ then $(D_N^{-(r+1)})_{i,j} = 0$. If $i > j$, then

$$(D_N^{-(r+1)})_{i,j} = \sum_{k=j}^{i}(D_N^{-r})_{i,k}(D_N^{-1})_{k,j} = \left(\sum_{k=j}^{i}\binom{r+i-k-1}{r-1}\right)(P_{W_i}P_{W_{i-1}}\dots P_{W_{j+1}}). \tag{3.18}$$

The combinatorial identity $\sum_{k=0}^{i-j}\binom{r-1+k}{k} = \sum_{k=0}^{i-j}\binom{r-1+k}{r-1} = \binom{r+i-j}{r}$, e.g., see page 1617 in [19], shows that $\sum_{k=j}^{i}\binom{r+i-k-1}{r-1} = \sum_{k=j}^{i}\binom{r-1+i-k}{i-k} = \sum_{k=0}^{i-j}\binom{r-1+k}{k} = \binom{r+i-j}{r}$. In particular, (3.18) reduces to $(D_N^{-(r+1)})_{i,j} = \binom{r+i-j}{r}(P_{W_i}P_{W_{i-1}}\dots P_{W_{j+1}})$ when $i > j$. □

Lemma 3.4.2. *Fix $\sigma \in \mathbb{N}$, $r \in \mathbb{N}$, and define $\{h_l\}_{l\in\mathbb{N}}$, $\{n_j\}_{j=1}^{r}$, $\{d_j\}_{j=1}^{r}$ by* (3.13) *and* (3.14). *If $n \geqslant n_r - r + 1$, then*

$$\binom{r+n-1}{r-1} = \sum_{l=1}^{n}\binom{r+n-1-l}{r-1}h_l. \tag{3.19}$$

Sketch of Proof. The result is contained in the proof of Proposition 6.1 in [19]. We provide a brief summary since [19] proves a more general result.

First, note that $\{n_j\}_{j=1}^r$ is an increasing sequence of strictly positive, distinct integers, which satisfies the requirements of Proposition 6.1 in [19]. The final sentence in step (i) of the proof Proposition 6.1 in [19] shows that

$$\binom{n+r-1}{r-1} - \sum_{n_j \leqslant n} d_j \binom{n-n_j+r-1}{r-1} = g_n,$$

where g_n is given by (6.1) in [19]. Moreover, the first two sentences in step (ii) in the proof of Proposition 6.1 in [19] give that $g_n = (\prod_{i=1}^r n_i)\, G(n)$ where $G(n) = 0$ whenever $n \geqslant n_r - r + 1$. Finally, recalling the definition h_l in (3.14) gives the desired conclusion. □

3.5 Proof of Theorem 3.3.2

In this section we prove Theorem 3.3.2.

Step 1. We first show that the operators $H_{n,j} : \bigoplus_{n=1}^N W_n \to \bigoplus_{n=1}^N W_n$ defined by (3.13), (3.14), (3.16) satisfy (3.12) when $\sigma \in \mathbb{N}$ is sufficiently large.

Note that $f(x) = \cosh(x)$ is decreasing on $(0, \infty)$ and $\lim_{x\to 0+} \cosh(x) = 1$. Given $\epsilon > 0$, it follows from (3.15) that there exists $N_0 = N_0(\epsilon)$ so that $\sigma > N_0$ implies

$$\|h\|_{\ell^1} < \cosh(\pi\sigma^{-1/2}) < 1 + \epsilon. \tag{3.20}$$

By (3.16) we have

$$\sup_{n\geqslant 1} \sum_{j=1}^L \|H_{n,j}\| = \sup_{n\geqslant 1} \sum_{j=1}^L \|h_j P_{W_n} P_{W_{n-1}} \dots P_{W_{n-j+1}}\| \leqslant \sup_{n\geqslant 1} \sum_{j=1}^L |h_j| \leqslant \|h\|_{\ell^1} < 1 + \epsilon. \tag{3.21}$$

Step 2. Define the $N \times N$ block operator $G_N = (D_N^{-r})(\mathcal{I}_N - \mathcal{H}_N)$. Using (3.5), (3.16), Lemma 3.4.1, and $(G_N)_{i,j} = \sum_{k=1}^N (D_N^{-r})_{i,k}(\mathcal{I}_N - \mathcal{H}_N)_{k,j}$ it can be shown that

$$(G_N)_{i,j} = \begin{cases} \left(\binom{r+i-j-1}{r-1} - \sum_{l=1}^{i-j} \binom{r+i-j-l-1}{r-1} h_l\right) (P_{W_i} P_{W_{i-1}} \cdots P_{W_{j+1}}) & \text{if } i > j, \\ I & \text{if } i = j, \\ 0 & \text{if } i < j. \end{cases} \tag{3.22}$$

Let $K = n_r - r + 1$. Lemma 3.4.2 shows that if $i - j \geqslant K$ then $\binom{r+i-j-1}{r-1} = \sum_{l=1}^{i-j} \binom{r+i-j-l-1}{r-1} h_l$. This shows that G_N is banded and satisfies

$$(G_N)_{i,j} = \begin{cases} \left(\binom{r+i-j-1}{r-1} - \sum_{l=1}^{i-j} \binom{r+i-j-l-1}{r-1} h_l \right) (P_{W_i} P_{W_{i-1}} \cdots P_{W_{j+1}}) & \text{if } K > i-j > 0, \\ I & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{3.23}$$

Step 3. Recall that $K = n_r - r + 1$ and let $M_r = \binom{r+K-2}{r-1}$. We next show that if $1 \leqslant i, j \leqslant N$ and $0 < i - j < K$ then

$$\|(G_N)_{i,j}\| \leqslant (2+\epsilon) M_r. \tag{3.24}$$

Since $\binom{r+m-2}{r-1}$ increases as m increases, it follows that if $0 < i - j < K$ then $\binom{r+i-j-1}{r-1} \leqslant \binom{r+K-2}{r-1} = M_r$. Likewise, if $1 \leqslant l \leqslant i - j < K$ then $\binom{r+i-j-l-1}{r-1} \leqslant \binom{r+K-2}{r-1} = M_r$. Also, recall that by (3.20) we have $\|h\|_{\ell^1} < 1 + \epsilon$. So, if $0 < i - j < K$ then (3.23) implies that

$$\begin{aligned} \|(G_N)_{i,j}\| &\leqslant \left| \binom{r+i-j-1}{r-1} \right| + \left| \sum_{l=1}^{i-j} \binom{r+i-j-l-1}{r-1} h_l \right| \\ &\leqslant M_r + M_r \sum_{l=1}^{i-j} |h_l| \leqslant M_r + M_r \|h\|_{\ell^1} \leqslant (2+\epsilon) M_r. \end{aligned}$$

Step 4. Next, we prove that

$$\sup_{N > K} \|G_N\| \leqslant (2+\epsilon) M_r K < \infty. \tag{3.25}$$

Suppose that $v \in \bigoplus_{n=1}^{N} W_n$ satisfies $\|v\|_2 = 1$. By (3.23), it follows that $G_N v$ satisfies

$$\forall\ 1 \leqslant n \leqslant N, \quad (G_N v)_n = \begin{cases} v_n + \sum_{k=1}^{n-1} G_{n,k} v_k & \text{if } 1 \leqslant n \leqslant K, \\ v_n + \sum_{k=n+1-K}^{n-1} G_{n,k} v_k & \text{if } K+1 \leqslant n \leqslant N. \end{cases} \tag{3.26}$$

Using (3.24), (3.26), and noting that $(2+\epsilon) M_r \geqslant 1$, gives

$$\|G_N v\|^2 = \sum_{n=1}^{K} \|v_n + \sum_{k=1}^{n-1} G_{n,k} v_k\|^2 + \sum_{n=K+1}^{N} \|v_n + \sum_{k=n+1-K}^{n-1} G_{n,k} v_k\|^2$$

$$\leqslant (2+\epsilon)^2 M_r^2 \sum_{n=1}^{K} \left(\|v_n\| + \sum_{k=1}^{n-1} \|v_k\| \right)^2 + (2+\epsilon)^2 M_r^2 \sum_{n=K+1}^{N} \left(\|v_n\| + \sum_{k=n+1-K}^{n-1} \|v_k\| \right)^2$$
$$= (2+\epsilon)^2 M_r^2 \left[\sum_{n=1}^{K} \left(\sum_{k=1}^{n} \|v_k\| \right)^2 + \sum_{n=K+1}^{N} \left(\sum_{k=n+1-K}^{n} \|v_k\| \right)^2 \right]. \tag{3.27}$$

Applying the Cauchy-Schwarz inequality to (3.27) gives

$$\|G_N v\|^2 \leqslant (2+\epsilon)^2 M_r^2 \left[\sum_{n=1}^{K} n \sum_{k=1}^{n} \|v_k\|^2 + \sum_{n=K+1}^{N} K \sum_{k=n+1-K}^{n} \|v_k\|^2 \right]$$
$$\leqslant (2+\epsilon)^2 M_r^2 K \left[\sum_{n=1}^{K} \sum_{k=1}^{n} \|v_k\|^2 + \sum_{n=K+1}^{N} \sum_{k=n+1-K}^{n} \|v_k\|^2 \right]. \tag{3.28}$$

Next, a computation shows that

$$\sum_{n=K+1}^{N} \sum_{k=n+1-K}^{n} \|v_k\|^2 = \sum_{n=1}^{K} \sum_{k=n+1}^{N-K+n} \|v_k\|^2. \tag{3.29}$$

Combining (3.28) and (3.29) completes the proof

$$\|G_N v\|^2 \leqslant (2+\epsilon)^2 M_r^2 K \sum_{n=1}^{K} \sum_{k=1}^{N-K+n} \|v_k\|^2 \leqslant (2+\epsilon)^2 M_r^2 K^2 \sum_{k=1}^{N} \|v_k\|^2 = (2+\epsilon)^2 M_r^2 K^2.$$

□

3.6 Reconstruction and error bounds

In this section, we describe the choice of left inverse $\mathcal{L}$ used for the reconstruction step (3.4). Combining the error expression (3.6) with the rth order condition (3.10) gives

$$x - \widetilde{x} = \mathcal{L}(\mathcal{I}_N - \mathcal{H}_N)v = \mathcal{L} D_N^r G_N v. \tag{3.30}$$

If $T : \mathbb{R}^d \to \bigoplus_{n=1}^N W_n$ is the analysis operator of the unweighted fusion frame $\{W_n\}_{n=1}^N$, we seek a left inverse $\mathcal{L} : \bigoplus_{n=1}^N W_n \to \mathbb{R}^d$ that satisfies $\mathcal{L}T = I$ and for which the quantization error in (3.30) is small.

By the stability result in Theorem 3.2.1, the state variable v satisfies $\|v\|_\infty \leqslant C < \infty$ and $\|v\| \leqslant \sqrt{N}\|v\|_\infty$. Also, the rth order condition (3.11) ensures that $C' = \sup_N \|G_N\| < \infty$ is finite.

So

$$\begin{aligned}\|x-\widetilde{x}\| &\leqslant \|\mathcal{L}D_N^r G_N\|\,\|v\| \\ &\leqslant \|\mathcal{L}D_N^r\|\,\|G_N\|\,\|v\| \\ &\leqslant C'\sqrt{N}\,\|\mathcal{L}D_N^r\|\|v\|_\infty \\ &\leqslant C'C\sqrt{N}\|\mathcal{L}D_N^r\|. \end{aligned} \tag{3.31}$$

In view of (3.31), the work in [24] selected $\mathcal{L}$ as a left inverse to T that makes $\|\mathcal{L}D_N^r\|$ small. The *rth order Sobolev left inverse* is defined by

$$\mathcal{L}_{r,Sob} = ((D_N^{-r}T)^* D_N^{-r}T)^{-1}(D_N^{-r}T)^* D_N^{-r}. \tag{3.32}$$

It is easily verified that $\mathcal{L}_{r,Sob}\,T = I$; see [24] for further discussion of Sobolev duals in the setting of fusion frames.

In general, it can be difficult to bound the operator norm $\|\mathcal{L}D_N^r\|$ in (3.31). It would be interesting to find quantitative bounds on $\|\mathcal{L}D_N^r\|$ when $\mathcal{L}$ is the Sobolev left inverse for nicely structured classes of deterministic or random fusion frames. For perspective, [6, 16, 21, 26, 31] provides analogous bounds for the scalar-valued setting of frames, and [24] contains examples for fusion frames when $\mathcal{L}$ is the canonical left inverse.

CHAPTER 4

Numerical experiments

This section contains two numerical examples which illustrate the performance of the low-bit fusion frame Sigma-Delta algorithm. For each $N \geqslant 3$, define the unit-vectors $\{\varphi_n^N\}_{n=1}^N \subset \mathbb{R}^3$ by

$$\varphi_n^N = \left(\tfrac{1}{\sqrt{3}}, \quad \sqrt{\tfrac{2}{3}} \cos(\tfrac{2\pi n}{N}), \quad \sqrt{\tfrac{2}{3}} \sin(\tfrac{2\pi n}{N}) \right),$$

and define the unweighted fusion frame $\mathcal{W}_N = \{W_n^N\}_{n=1}^N$ by

$$W_n^N = \{x \in \mathbb{R}^3 : \langle x, \varphi_n^N \rangle = 0\}.$$

For each fixed $N \geqslant 3$, $\{W_n^N\}_{n=1}^N$ is an unweighted tight fusion frame for $\mathbb{R}^3$ with fusion frame bound $A = A_N = \frac{2N}{3}$, e.g., see Example 1 in [24]. Moreover, the vectors $e_{1,n}^N = \big(0, \sin(\frac{2\pi n}{N}), -\cos(\frac{2\pi n}{N})\big)$ and $e_{2,n}^N = \sqrt{\frac{1}{3}} \big(-\sqrt{2}, \cos(\frac{2\pi n}{N}), \sin(\frac{2\pi n}{N})\big)$ form an orthonormal basis for W_n^N.

Let $\mathcal{A}_n^N \subset W_n^N$ be the low-bit quantization alphabet given by Lemma 2.3.1. Since $\dim(W_n^N) = 2$, each alphabet $\mathcal{A}_n^N$ contains 3 elements, and can be defined by

$$\mathcal{A}_n^N = \left\{ e_{1,n}^N, \left(-\frac{1}{2} e_{1,n}^N + \frac{\sqrt{3}}{2} e_{2,n}^N \right), \left(-\frac{1}{2} e_{1,n}^N - \frac{\sqrt{3}}{2} e_{2,n}^N \right) \right\}.$$

Let Q_n be a vector quantizer associated to $\mathcal{A}_n^N$ by (2.1).

4.1 Example 1 (second order algorithm)

This example considers the performance of the second order fusion frame Sigma-Delta algorithm. By Theorems 3.2.1 and 3.3.2 we can choose appropriate σ and h, as in Section 3.3, to ensure that the algorithm (3.2), (3.3) is stable and second order. In Theorem 3.2.1, let $\delta = 0.1$, so that $\alpha_1 \approx 1.1015$ and $\alpha_2 \approx 1.2198$ allows us to pick $\alpha = 1.101$. Using (3.15) and (3.21), the condition $\sup_{n \geqslant 1} \sum_{j=1}^L \|H_{n,j}\| \leqslant \|h\|_{\ell^1} < \alpha$ will be satisfied if $\pi \sigma^{-1/2} = \cosh^{-1}(\alpha) = \ln(\alpha + \sqrt{\alpha^2 - 1}) \leqslant$

0.4458, which occurs when $\sigma \geqslant 49.67$. We pick $\sigma = 50$, so that (3.13) gives $n_1 = 1, n_2 = 51$, and (3.14) gives

$$h_j = \begin{cases} \frac{n_2}{n_2 - n_1} = \frac{51}{50} & \text{if } j = n_1, \\ \frac{n_1}{n_1 - n_2} = -\frac{1}{50} & \text{if } j = n_2, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

The second order low-bit $\Sigma\Delta$ quantization algorithm takes the following form

$$q_n^N = Q_n \left(y_n^N + h_1 P_{W_n^N}(v_{n-1}^N) + h_{n_2} \prod_{k=0}^{n_2-1} P_{W_{n-k}^N}(v_{n-n_2}^N) \right), \tag{4.2}$$

$$v_n^N = y_n^N - q_n^N + \left(h_1 P_{W_n^N}(v_{n-1}^N) + h_{n_2} \prod_{k=0}^{n_2-1} P_{W_{n-k}^N}(v_{n-n_2}^N) \right). \tag{4.3}$$

Let $x = (\frac{1}{25}, \frac{\pi}{57}, \frac{1}{2\sqrt{57}})$ and define the fusion frame measurements by $y_n^N = P_{W_n^N}(x)$. Note that $\|y_n\| \leqslant \|x\| \leqslant \delta$. Run the second order low-bit fusion frame Sigma-Delta algorithm with inputs $\{y_n^N\}_{n=1}^N$, to obtain the quantized outputs $q^N = \{q_n^N\}_{n=1}^N$.

Let T_N be the analysis operator for the unweighted fusion frame $\{W_n^N\}_{n=1}^N$. The canonical left inverse of T_N is $\mathcal{L}_N = S_N^{-1} T_N^* = (A_N^{-1} I) T_N^* = A_N^{-1} T_N^*$. Since the fusion frame is tight with bound $A_N = 2N/3$, it follows that $\mathcal{L}_N = \frac{3}{2N} T_N^*$, e.g., [24]. Also let $\mathcal{L}_{2,Sob}^N$ be the second order Sobolev left inverse of T_N, as defined in (3.32). Consider the following two different methods of reconstructing a signal from q^N

$$\widetilde{x}_N = \mathcal{L}_N(q^N) \quad \text{and} \quad \widetilde{x}_{N,Sob} = \mathcal{L}_{2,Sob}^N(q^N).$$

Figure 4.1 shows log-log plots of $\|x - \widetilde{x}_N\|$ and $\|x - \widetilde{x}_{N,Sob}\|$ against N. For comparison, log-log plots of $2/N$ and $100/N^2$ against N are also given.

4.2 Example 2 (third order algorithm)

We consider the same experiment as in Example 1, except that we use an algorithm of order $r = 3$.

We again use the parameters $\delta = 0.1$ and $\sigma = 50$. Using (3.13) with $\sigma = 50$ and $r = 3$ gives

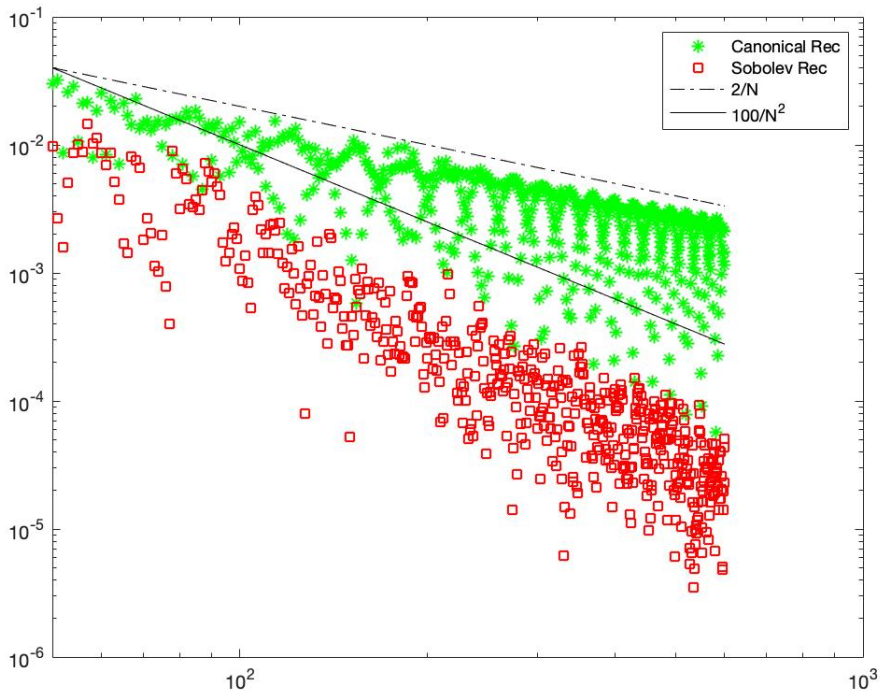

Figure 4.1: Error for the second order algorithm in Example 1.

$n_1 = 1, n_2 = 51, n_3 = 201$ and

$$h_j = \begin{cases} \frac{n_2 n_3}{(n_2-n_1)(n_3-n_1)} & \text{if } j = n_1, \\ \frac{n_1 n_3}{(n_1-n_2)(n_3-n_2)} & \text{if } j = n_2, \\ \frac{n_1 n_2}{(n_1-n_3)(n_2-n_3)} & \text{if } j = n_3, \\ 0 & \text{otherwise.} \end{cases} \tag{4.4}$$

The third order low-bit fusion frame Sigma-Delta quantization algorithm takes the following form

$$q_n^N = Q_n\left(y_n^N + h_1 P_{W_n^N}(v_{n-1}^N) + h_{n_2}\prod_{k=0}^{n_2-1} P_{W_{n-k}^N}(v_{n-n_2}^N) + h_{n_3}\prod_{k=0}^{n_3-1} P_{W_{n-k}^N}(v_{n-n_3}^N)\right), \tag{4.5}$$

$$v_n^N = y_n^N - q_n^N + \left(h_1 P_{W_n^N}(v_{n-1}^N) + h_{n_2}\prod_{k=0}^{n_2-1} P_{W_{n-k}^N}(v_{n-n_2}^N) + h_{n_3}\prod_{k=0}^{n_3-1} P_{W_{n-k}^N}(v_{n-n_3}^N)\right). \tag{4.6}$$

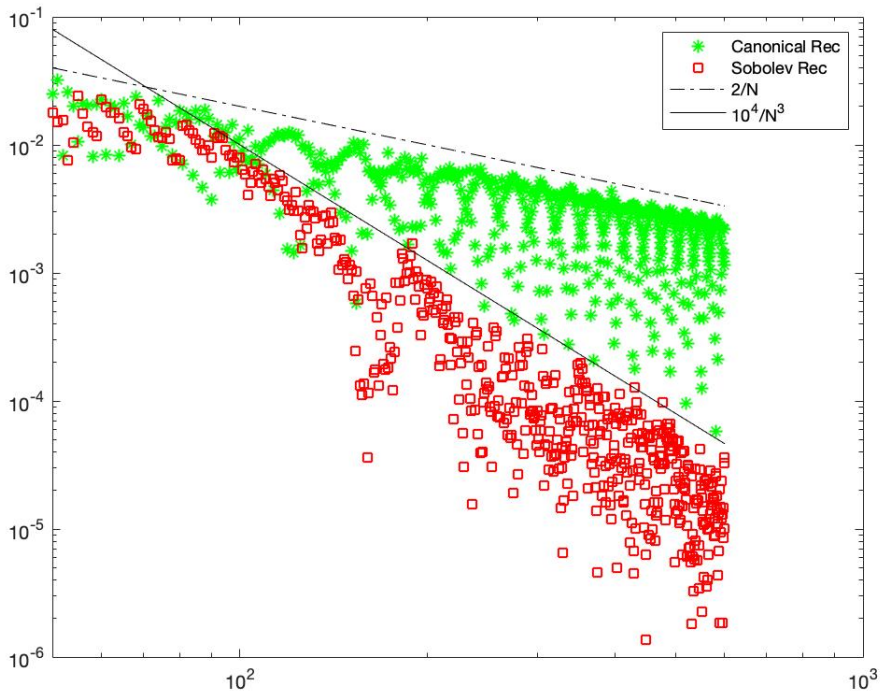

Figure 4.2: Error for the third order algorithm in Example 2.

Let $x = (\frac{1}{25}, \frac{\pi}{57}, \frac{1}{2\sqrt{57}})$, and use the third order fusion frame Sigma-Delta algorithm with inputs $\{y_n^N\}_{n=1}^N$ to obtain the quantized outputs $q^N = \{q_n^N\}_{n=1}^N$. For the reconstruction step, let $\mathcal{L}_N = \frac{3}{2N} T_N^*$ be the canonical left inverse of T_N and let $\mathcal{L}_{3,Sob}^N$ be the third order Sobolev left inverse of T_N, as defined in (3.32). We consider the following two different methods of reconstructing a signal from q^N

$$\widetilde{x}_N = \mathcal{L}_N(q^N) \quad \text{and} \quad \widetilde{x}_{N,Sob} = \mathcal{L}_{3,Sob}^N(q^N).$$

Figure 4.2 shows log-log plots of $\|x - \widetilde{x}_N\|$ and $\|x - \widetilde{x}_{N,Sob}\|$ against N. For comparison, log-log plots of $2/N$ and $10^4/N^3$ against N are also given.

CHAPTER 5

Outlook

In this paper we have discussed higher order Sigma-Delta modulators for fusion frame measurements and proved their stability. As for finite frames, the reconstruction accuracy of such approaches will depend on the fusion frame under consideration. In particular, we expect that for certain adversarial fusion frame constructions, only very slow error decay can be achieved.

On the other hand, our numerical experiments in the previous section show that for certain deterministic fusion frames the error decays polynomially of an order that corresponds to the order of the Sigma-Delta scheme. For random frames, such error bounds have been established with high probability [21, 27, 16]. These result have important implications for compressed sensing with random measurement matrices. Given that the theory of compressed sensing generalizes to the setting of fusion frames [9], and there exists analysis of random fusion frames which parallels the restricted isometry property [7]; it will be interesting to understand if the aforementioned results generalize to the stable low-bit rth order fusion frame Sigma-Delta algorithms discussed in this paper, or whether modifications are necessary. The crucial quantity to estimate is the last factor in (3.31) for the Sobolev dual of a random fusion frame. In any case, we expect that the stability analysis provided in this paper will be of crucial importance even in the latter case.

5.1 Experiments of random fusion frames

In this section, we will do numerical experiments when we take random fusion frames. We consider the same experiment as in Example 1 and Example 2, except that we take random frames.

We again use the parameters $\delta = 0.1$ and $\sigma = 50$. First we will do the experiment when $r = 2$, then we have another figure when $r = 3$. n_i are still calculated as in previous examples and h_j, q_n^N and v_n^N are again calculated by formula (8.1), (4.5) and (4.6).

Let $x = (\frac{1}{25}, \frac{\pi}{57}, \frac{1}{2\sqrt{57}})$, and use the fusion frame Sigma-Delta algorithm with inputs $\{y_n^N\}_{n=1}^N$ to obtain the quantized outputs $q^N = \{q_n^N\}_{n=1}^N$. For the reconstruction step, let $\mathcal{L}_N$ be the canonical left inverse of T_N and let $\mathcal{L}_{2,Sob}^N$ or $\mathcal{L}_{3,Sob}^N$ be the second or third order Sobolev left inverse of T_N.

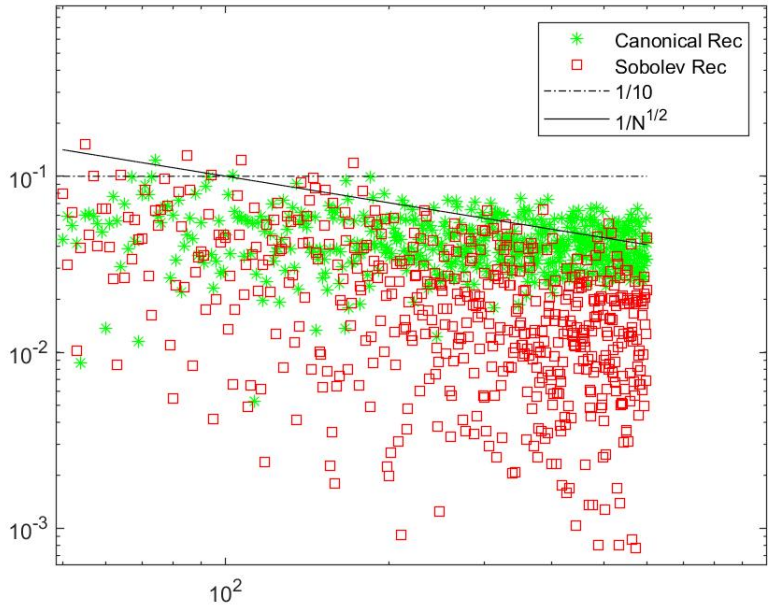

Figure 5.1: Error for the second order random frames in section 5.1.

We consider the following two different methods of reconstructing a signal from q^N

$$\widetilde{x}_N = \mathcal{L}_N(q^N) \quad \text{and} \quad \widetilde{x}_{N,Sob} = \mathcal{L}^N_{2,Sob}(q^N) \text{ or } \mathcal{L}^N_{3,Sob}(q^N).$$

Figure 5.1 and Figure 5.2 shows log-log plots of $\|x - \widetilde{x}_N\|$ and $\|x - \widetilde{x}_{N,Sob}\|$ against N. For comparison, log-log plots of $1/10$ and $1/N^{1/2}$ against N are also given.

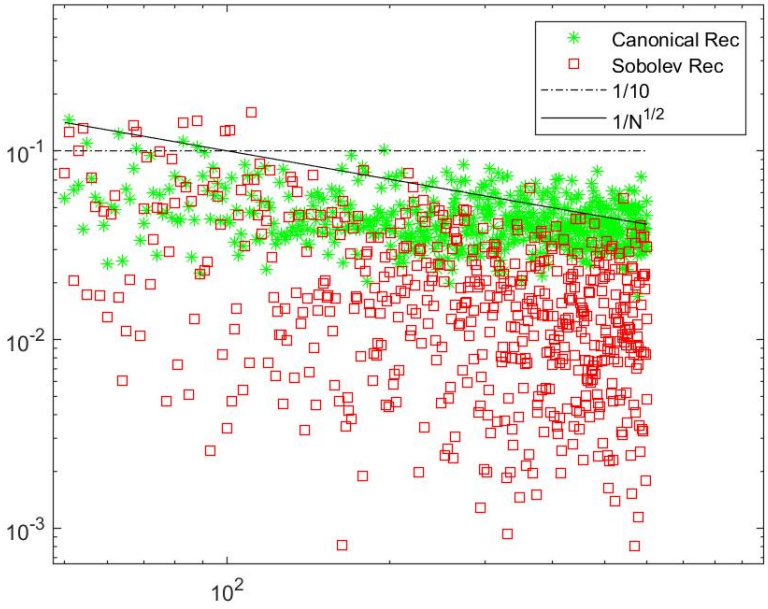

Figure 5.2: Error for the third order random frames in section 5.1.

CHAPTER 6

Introduction: Hypergraph Signal Processing and Applications

The second part of this thesis is on hypergraph signal processing.

Hypergraphs are a generalization of the concept of graphs. In mathematics, a graph is a structure for some objects in which some pairs of the objects have some kind of relation. We use vertices (sometimes also called as nodes or points) to denote the objects and edges to denote such relations. It can be written as $G = (V, E)$, where G corresponds to the graph, V corresponds to the vertices and E corresponds to the edges. More precisely, $V = \{v_1, v_2, \ldots, v_n\}$, where v_i is one vertex, and $E \subset \{(v_i, v_j) | (v_i, v_j) \in V^2 \text{ and } 1 \leqslant i, j \leqslant n, i \neq j\}$, where (v_i, v_j) is one edge. We can use a $n \times n$ matrix to formulate the graph $A = \{(a_{i,j})\}$ where $a_{i,j} \in \{0, 1\}$ when $i \neq j$, and $a_{i,i} = 0$. So $a_{i,j} = 1$ means objects i has relation i with j and they share an edge.

Graphs are a very powerful tool in discrete mathematics because they can model the relations between discrete objects, such as social networks (where each person or organization is one vertex), digital maps (i.e. google map, where each building is one vertex) or review systems (where each product and each customer could be one vertex). We can modify a simple graph to allow the graph to contain some more complicated information, like letting $w_{i,j}$ be a positive number associated to each edge to include the weights, which can represent the importance for the relation, or letting $a_{i,j} \neq 0$ but $a_{j,i} = 0$ to give a direction between the objects.

However, there are situations where more than two objects have relations. In this situation, a model of graph may lose some very important information. For example, suppose we want to use a graph to describe relation between books, and if two books belong to same category, they share an edge. Now suppose we have three books, where book 1 and book 2 are history, book 2 and book 3 are politics, book 1 and book 3 are science. Then we will have a fully connected graph of three vertices. Consider another case, where book 1, book 2 and book 3 all belong to history. In this case, we also get a fully connected graph. These two cases would generate the same graph, but they are in fact have totally different meanings.

For hypergraphs, we require the concept of hyperedge rather than edge. A hyperedge is a set

of vertices that share some common relations, in contrast, an edge is only a set of two vertices. So we can use the same notations and also denote a hypergraph as $G = (V, E)$, where V is the set of vertices and E is the set of hyperedges. $V = \{v_1, v_2, \ldots, v_n\}$, where $v_i (i = 1, \ldots, n)$ is one vertex, and $E = \{e_1, e_2, \ldots, e_m\}$, where e_i is one hyperedge and consists of a set of vertices. A weighted hypergraph $G = (V, E, w)$, where $w = \{w(e)\}$, is a hypergraph that has a positive number $w(e)$ associated with each hyperedge e. If we do a simple math, we can see for an undirectioned graph, there would be at most $n(n-1)/2$ edges, but for a hypergraph graph, there would be at most $2^n - 1$ hyperedges. So a hypergraph can take more information than a graph. Besides, if all hyperedges only contains exactly two vertices, a hypergraph is the same as a graph.

Here we are going to introduce some more notation for hypergraphs.

- A hypergraph G can be represented by a $|V| \times |E|$ matrix H, $h(v, e) = 1$ if $v \in e$ and 0 otherwise.
- For a vertex $v \in V$, the corresponding degree is defined as $d(v) = \sum_{\{e \in E | v \in e\}} w(e) = \sum_{e \in E} h(v, e) w(e)$.
- For a hyperedge $e \in E$, the corresponding degree is defined as $\delta(e) = |e|$, i.e., number of vertices contained in this hyperedge. It is also $\delta(e) = \sum_{v \in V} h(v, e)$.
- D_v and D_e can denote the diagonal matrix containing the vertex degree and hyperedge degree. Obviously, D_v is $|V| \times |V|$ and D_e is $|E| \times |E|$.
- W can denote the diagonal matrix containing the weights $w(e)$ of each hyperedge. So W is $|E| \times |E|$.
- The weighted adjacency matrix A of hypergraph $G = (V, E, W)$ can be defined as $A = HWH^T$. Then A would be $|V| \times |V|$.

We consider the question of how to adapt tools from classical signal processing to the setting of hypergraphs. We focus on the following three methods: diffusion maps, wavelets, and the empirical mode decomposition (EMD).

Diffusion maps are a dimensionality reduction or feature extraction algorithm proposed by Coifman and Lafon [reference]. It could embed data into Euclidean space, i.e., representing weighted

graph $G = (V, E, W)$ in $\mathbb{R}^d$. The coordinates of the embedding given by diffusion map usually come from eigenvectors and eigenvalues of a diffusion operator. Other dimensionality reduction methods, such as principal component analysis (PCA), take advantage of linear transforms. However, diffusion maps are a non-linear method. It tries to discover the underlying manifold where the data comes from.

Many signal processing techniques are based on transform methods. In many fields, Fourier transforms, which can decompose a continuous time signal into sines and cosines, are widely used. The wavelet transform is similar to the Fourier transform with a completely different basis function. The wavelet transform is more powerful in some sense by using functions that are localized in both the time and frequency. It is a mapping from $L^2(\mathbb{R}) \to L^2(\mathbb{R}^2)$. We can write a continuous wavelet transform W_s of signal $f(x)$ as

$$W_f(a,b) =< \psi_{a,b}, f >= \int_{-\infty}^{\infty} f(x) a^{-1} \psi^*(\frac{x-b}{a}) dx$$

where ψ is the wavelet function, a is dilation term and b is translation term and

$$\psi_{a,b}(x) := \frac{1}{|a|}\psi(\frac{x-b}{a}).$$

The Empirical mode decomposition (EMD) is another transform method. A big difference between EMD and the wavelet transform or Fourier transform is that EMD can break down a signal without leaving the time domain. It is very helpful when we analyze natural signals. Because natural time series are often generated by multiple causes, and each of these causes may occur at different time intervals. EMD can decompose this kind of data easily and clearly, whereas the Fourier transform or wavelet transform cannot.

The EMD will break down a signal into its component intrinsic mode functions (IMFs). For a given signal $f(x) \in L^2(\mathbb{R})$, the operator $\mathcal{S}[f]$ can be defined by the following steps:

• Find all extrema of $f(x)$.

• Use a cubic spline to interpolate minima, write the interpolation function as $e_{\text{min}}(x)$; similarly interpolate maxima and get $e_{\text{max}}(x)$.

• Take the mean of the two interpolation functions $m(x) = (e_{\text{min}}(x) + e_{\text{max}}(x))/2$.

• The remaining part of the signal is $\mathcal{S}[f](x) = f(x) - m(x)$.

The above steps would be repeated until some conditions are met. Suppose it terminates after n iterations, we would have $d_1[f](x) = S^n[f](x)$ and $m_1[f](x) = f(x) - d_1[f](x)$. $d_1[f](x)$ here is an oscillatory signal called as an intrinsic mode function (IMF). It can be shown that [34] an IMF has only one extrema between zero crossings, and has a mean value of zero. If we do not stop with $m_1[f](x)$, we can get $m_1[f](x) = m_2[f](x) + d_2[f](x)$ and so on. At last, we can represent $f(x)$ as

$$f(x) = m_K[f](x) + \sum_{k=1}^{K} d_k[f](x).$$

CHAPTER 7

Diffusion Map, Wavelet and Empirical Mode Decomposition

In this chapter, we will construct diffusion maps, wavelet, and the emperical mode decomposition on hypergraphs.

7.1 Diffusion maps on hypergraphs

Given hypergraph $G = (V, E, w)$, again, here we want to embed the hypergraph to Euclidean space $\mathbb{R}^d$. Consider a random walk on the vertices of V with transition probabilities

$$\text{Prob}\{X(t+1) = j | X(t) = i\} = \frac{\sum_{e \in E} h(v_i, e) w(e) \frac{1}{|e|} h(v_j, e)}{d(v_i)}.$$

The explanation for this formula can be, if we start from vertex v_i, we can look at all hyperedges that v_i is in. Next, because each hyperedge may contain more two vertices, if e contains vertex v_j, we need to normalize the probability by dividing $|e|$, since we may walk to other vertices. Of course, we need to have the weight of e in the denominator. Let M be the matrix of probabilities,

$$M_{i,j} = \frac{\sum_{e \in E} h(v_i, e) w(e) \frac{1}{|e|} h(v_j, e)}{d(v_i)}.$$

It is straightforward to see $M_{i,j} \geqslant 0$ and

$$(M\mathbf{1})_i = \frac{\sum_{e \in E} h(v_i, e) w(e) \sum_{1 \leqslant j \leqslant |V|} \frac{1}{|e|} h(v_j, e)}{d(v_i)} = \frac{\sum_{e \in E} h(v_i, e) w(e)}{d(v_i)} = 1.$$

where $\mathbf{1}$ is a vector with each entry has value 1. So the above definitions are valid. We can write M as $M = D_v^{-1} H W D_e^{-1} H^T$.

If we start the random walk at vertex v_i at time $t_0 = 0$, then at time point t, the probability of ending at vertex v_j would be

$$\text{Prob}\{X(t) = j | X(0) = i\} = M^t[i, j].$$

Of course, we can map v_i to the probability cloud $M^t[i,:]$. This could be an embedding from the hypergraph onto the Euclidean space $\mathbb{R}^d$. However, it requires $d = |V|$, which could be unacceptably large. We want to use spectral methods to reduce the dimensionality. Unfortunately M is not symmetric, so we need to find an alternative that is similar to M and is symmetric. Define

$$S = D_v^{\frac{1}{2}} M D_v^{-\frac{1}{2}} = D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}}.$$

It is not hard to verify S is a symmetric matrix. Consider the spectral decomposition of S. Then $S = U\Lambda U^T$, where $U = [u_1, \ldots, u_{|V|}]$ satisfies the property $U^T U = I_{|V|\times|V|}$ and Λ is a diagonal matrix with $\Lambda_{i,i} = \lambda_i$. Here we can assume $\{\lambda_i\}$ are ordered that $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_{|V|}$. So we can write M as

$$M = D_v^{-\frac{1}{2}} S D_v^{\frac{1}{2}} = (D_v^{-\frac{1}{2}} U)\Lambda(D_v^{\frac{1}{2}} U)^T = \Phi\Lambda\Psi^T.$$

where $\Phi = D_v^{-\frac{1}{2}} U = [\phi_1, \phi_2, \ldots, \phi_{|V|}]$ and $\Psi = D_v^{\frac{1}{2}} U = [\psi_1, \psi_2, \ldots, \psi_{|V|}]$. Here Φ and Ψ build a biorthogonal system such that $\Phi^T\Psi = \Psi^T\Phi = I_{|V|\times|V|}$. We also have $M\phi_k = \lambda_k\phi_k$ and $\psi_k^T M = \lambda_k \psi_k^T$, thus

$$M = \sum_{k=1}^{|V|} \lambda_k \phi_k \psi_k^T.$$

and after t steps,

$$M^t = \sum_{k=1}^{|V|} \lambda_k^t \phi_k \psi_k^T.$$

Then we can construct a map from a vertex in the hypergraph to $\mathbb{R}^d$:

$$v_i \mapsto M^t[i,:] = \sum_{k=1}^{|V|} \lambda_k^t \phi_k(i) \psi_k^T.$$

If we use the set $\{\psi_k\}$ as the basis, we have

$$v_i \mapsto M^t[i,:] = \left[\lambda_1^t\phi_1(i), \lambda_2^t\phi_2(i), \ldots, \lambda_{|V|}^t\phi_{|V|}(i)\right]^T.$$

Proposition 7.1.1. *All eigenvalues λ_k of M have property that $|\lambda_k| \leqslant 1$.*

Proof. For any eigenvalue λ_k, let ϕ_k be the corresponding eigenvector. Suppose i_k is the index for

which $|\phi_k(i_k)| \geqslant |\phi_k(j)|$ for all $j \neq i_k$. Then,

$$|\lambda_k \phi_k(i_k)| = |M\phi_k(i_k)| = |\sum_{j=1}^{|V|} M_{i_k,j}\phi_k(j)| \leqslant \sum_{j=1}^{|V|} |M_{i_k,j}||\phi_k(i_k)| = |\phi_k(i_k)|.$$

The last equality is because $M\mathbf{1} = \mathbf{1}$. So $|\lambda_k| \leqslant 1$. □

Again by $M\mathbf{1} = \mathbf{1}$, we could conclude $\lambda_1 = 1$. So we do not need to look at the first eigenvector.

Definition 7.1.2 (Diffusion Map)**.** *Given a hypergraph $G = (V, E, w)$ and an interger d, we can construct M and have a map from the hypergraph to a d-dimensional Euclidean space, the map is $f_t : V \to \mathbb{R}^d$, where*

$$f_t^{(d)}(v_i) = \left[\lambda_2^t \phi_2(i), \lambda_3^t \phi_3(i), \ldots, \lambda_{d+1}^t \phi_{d+1}(i)\right]^T.$$

when $d = |V| - 1$, this is a diffusion map, and when $d < |V| - 1$, this is truncated diffusion map.

7.2 Spectral hypergraph wavelet transform

Let $G = (V, E, w)$ be a given hypergraph. For a vertex subset $S \subset V$ and the compliment S^c. Similar to the cut in a graph, a cut in a hypergraph is also to partition of V into subsets S and S^c. If a hyperedge e contains vertices from both S and S^c, we say e is a cut. The boundary of S can be defined as

$$\partial S = \{e \in E | e \cap S \neq \varnothing, e \cap S^c \neq \varnothing\}.$$

In another word, ∂S is the set of all cuts. Similar to the volume in a graph, a volume of S in a hypergraph is the sum of the degrees of the vertices in S, which is vol$S = \sum v \in Sd(v)$. So the volume of the boundary of S is

$$\text{vol}\partial S = \sum_{e \in \partial S} w(e) \frac{|e \cap S||e \cap S^c|}{\delta(e)}.$$

If we take a hyperedge $e \in E$ as a fully connected subgraph ($W(v, u) = 1$ for any $v, u \in e$),

then the cut in the graph view is

$$\text{cut}(e \cap S, e \cap S^c) = \sum_{v \in e \cap S, u \in e \cap S^c} W(v,u) = |e \cap S||e \cap S^c|.$$

Then the hypergraph cut $\text{Hcut}(S, S^c) := \sum_{e \in E} \text{cut}(e \cap S, e \cap S^c) = \sum_{e \in E} |e \cap S||e \cap S^c|$. Suppose we take the same weight $w(e)/\delta(e)$ for all edges in this fully connected subgraph, $\text{cut}(e \cap S, e \cap S^c) = w(e)\frac{|e \cap S||e \cap S^c|}{\delta(e)}$. Then we get

$$\text{vol}\partial S = \sum_{e \in \partial S} \text{cut}(e \cap S, e \cap S^c).$$

By symmetry, it is not hard to see that $\text{vol}\partial S = \text{vol}\partial S^c$. Similar to the Ncut in a graph, Ncut in a hypergraph could be defined as

$$\begin{aligned}
\text{Ncut}(S, S^c) :=& \sum_{e \in E} \left(\frac{\text{cut}(e \cap S, e \cap S^c)}{\text{vol}S} + \frac{\text{cut}(e \cap S^c, e \cap S)}{\text{vol}S^c} \right) \\
=& \sum_{e \in E} \text{cut}(e \cap S, e \cap S^c) \left(\frac{1}{\text{vol}S} + \frac{1}{\text{vol}S^c} \right) \\
=& \text{vol}\partial S \left(\frac{1}{\text{vol}S} + \frac{1}{\text{vol}S^c} \right).
\end{aligned}$$

Consider an optimization problem

$$\text{argmin}_{\varnothing \neq S \subset V} \text{Ncut}(S, S^c) \tag{7.1}$$

This problem (7.1) is NP-hard, we can relex it as

$$\text{argmin}_f \frac{1}{2} \sum_{e \in E} \sum_{v,u \in e} \frac{w(e)}{\delta(e)} \left(\frac{f(v)}{\sqrt{d(v)}} - \frac{f(u)}{\sqrt{d(u)}} \right)^2$$

$$\text{subject to} \sum_{v \in V} f^2(v) = 1, \sum_{v \in V} f(v)\sqrt{d(v)} = 0.$$

In Section 7.1, we defined $M = D_v^{-1} H W D_e^{-1} H^T$ and a similar symmetric alternative to M as

$S = D_v^{\frac{1}{2}} M D_v^{-\frac{1}{2}} = D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}}$. Define $\Delta = I - S$, it could be shown that

$$\sum_{e \in E} \sum_{v,u \in e} \frac{w(e)}{\delta(e)} \left(\frac{f(v)}{\sqrt{d(v)}} - \frac{f(u)}{\sqrt{d(u)}} \right)^2 = 2 f^T \Delta f. \tag{7.2}$$

Then the normalized hypergraph Laplacian matrix is defined as

$$L_{sym} := I - D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2},$$

We also define the unnormalized hypergraph Laplacian matrix as

$$L = D_v^{1/2} L_{sym} D_v^{1/2} = D_v - H W D_e^{-1} H^T.$$

The Fourier transform is defined by

$$f(x) = \frac{1}{2\pi} \int_{\mathbb{T}^d} \hat{f}(\omega) e^{i\omega x} d\omega = < \frac{1}{2\pi} e^{-i\omega x}, \hat{f} >,$$

$$\hat{f}(\omega) = \int_{\mathbb{T}^d} f(x) e^{-i\omega x} dx = < e^{i\omega x}, f > .$$

In fact, we can notice that $\frac{d}{dx^2}(e^{i\omega x}) = -\omega^2 (e^{i\omega x})$, so $e^{i\omega x}$ is eigenfunction of Laplacian operator $\frac{d}{dx^2}$. Similarily, we can define hypergraph Fourier transform.

$$\hat{f}(l) = < \chi_l, f > = \sum_{n=1}^{N} \chi_l^*(n) f(n),$$

$$f(n) = < \chi_l^*, \hat{f} > = \sum_{l=0}^{N-1} \chi_l(n) \hat{f}(l),$$

where L is the symmetric hypergraph Laplacian matrix, and χ_l $(l = 0, \ldots, N-1)$ are eigenfunctions to eigenvalues λ_l,

$$L \chi_l = \lambda_l \chi_l.$$

If we consider Φ as the eigenfunction matrix, i.e. $\Phi = (\chi_l)_{l=1,\ldots,N}$, then Φ is an unitary matrix. Also if one treats $f, \hat{f}$ as two $N \times 1$ vectors, then we can arrange the hypergraph Fourier transform

in matrix form as

$$\hat{f} = \Phi^* f,$$

$$f = \Phi \hat{f}.$$

Theorem 7.2.1. *The Parseval relation holds for the hypergraph Fourier transform, in particular for any* $f, g \in \mathbb{R}^N$

$$< f, g >=< \hat{f}, \hat{g} > .$$

Proof.

$$< \hat{f}, \hat{g} >= \hat{f}^* \hat{g} = (\Phi^* f)^* \Phi^* g = f^* \Phi \Phi^* g = f^* g =< f, g >$$

□

The spectral hypergraph wavelet transform will be determined by the choice of a kernel function $g : \mathbb{R}^+ \to \mathbb{R}^+$, which is analogous to $\hat{\psi}^*$. This kernal g should behave as a band-pass filter, i.e. $g(0) = 0$ and $\lim_{x\to\infty} g(x) = 0$. Recall $\hat{\psi}(0) = \int_{-\infty}^{\infty} \psi(t)dt = 0$ is the admissibility condition.

Lemma 7.2.2. *For the spectral hypergraph wavelet kernel g, the wavelet operator* $T_g = g(L)$ *acts on a given function* f *by*

$$\widehat{(T_g f)}(l) = g(\lambda_l)\hat{f}(l).$$

Proof. Since $g(L)f = diag\{g(\lambda_0), \ldots, g(\lambda_{N-1})\}f$, then

$$\widehat{g(L)f} = diag\{g(\lambda_0), \ldots, g(\lambda_{N-1})\}\hat{f}.$$

□

Applying the inverse Fourier transform to $\widehat{T_g f}$, we have

$$(T_g f)(m) = \sum_{l=0}^{N-1} g(\lambda_l)\hat{f}(l)\chi_l(m).$$

The wavelet operator at scale t is defined by $T_g^t = g(tL)$. Note: even though the domain on a hypergraph is discrete, since the kernel g is continuous, the scaling t may be defined for any positive number $t > 0$.

The spectral hypergraph wavelets can be realized as through localizing these operators by applying them to the impulse on a single vertex, so on a hypergraph domain, we have

$$\begin{aligned}\psi_{t,n}(m) = T_g^t \delta_n(m) &= \sum_{l=0}^{N-1} g(t\lambda_l)\hat{\delta}_n(l)\chi_l(m) \\ &= \sum_{l=0}^{N-1} g(t\lambda_l)\left(\sum_{m=1}^{N} \chi_l^*(m)\delta_n(m)\right)\chi_l(m) = \sum_{l=0}^{N-1} g(t\lambda_l)\chi_l^*(n)\chi_l(m).\end{aligned}$$

Again, define the analogous spectral hypergraph wavelet transform as follows,

$$W_f(t,n) =< \psi_{t,n}, f > .$$

Lemma. *If $n \times n$ matrix A is symmetric, then eigenvectors corresponding to different eigenvalues are orthogonal.*

Proof. Suppose χ_1, χ_2 are two eigenvectors corresponding to λ_1, λ_2 and $\lambda_1 \neq \lambda_2$, then

$$\lambda_1\chi_1^*\chi_2 = (A\chi_1)^*\chi_2 = \chi_1^*(A\chi_2) = \chi_1^*(\lambda_2\chi_2) = \lambda_2\chi_1^*\chi_2.$$

□

Because of the orthogonality of $\{\chi_l\}$, we can express the spectral hypergraph wavelet transform explicitly,

$$\begin{aligned}W_f(t,n) = (T_g^t f)(n) &=< \psi_{t,n}, f > \\ &= \sum_{m=1}^{N} \psi_{t,n}^*(m) f(m) \\ &= \sum_{m=1}^{N} \psi_{t,n}^*(m)\left(\sum_{l=0}^{N-1} \hat{f}(l)\chi_l(m)\right) \\ &= \sum_{m=1}^{N} \left(\sum_{l=0}^{N-1} g(t\lambda_l)\chi_l^*(n)\chi_l(m)\right)^* \left(\sum_{l=0}^{N-1} \hat{f}(l)\chi_l(m)\right) \\ &= \sum_{m=1}^{N} \left(\sum_{l=0}^{N-1} g(t\lambda_l)\chi_l(n)\chi_l^*(m)\right)\left(\sum_{l=0}^{N-1} \hat{f}(l)\chi_l(m)\right) \\ &= \sum_{m=1}^{N}\sum_{l=0}^{N-1}\sum_{l'=0}^{N-1} g(t\lambda_l)\chi_l(n)\chi_l^*(m)\hat{f}(l')\chi_{l'}(m)\end{aligned}$$

$$
\begin{aligned}
&= \sum_{l=0}^{N-1} \sum_{l'=0}^{N-1} \left(\sum_{m=1}^{N} g(t\lambda_l)\chi_l(n)\chi_l^*(m)\hat{f}(l')\chi_{l'}(m) \right) \\
&= \sum_{l=0}^{N-1} \sum_{l'=0}^{N-1} g(t\lambda_l)\chi_l(n)\hat{f}(l')\mathbf{1}_l(l') \\
&= \sum_{l=0}^{N-1} g(t\lambda_l)\hat{f}(l)\chi_l(n).
\end{aligned}
$$

7.3 Hypergraph Empirical Mode Decomposition

Let $G = (V, E, w)$ be given hypergraph and f be a signal defined on V. Then a node i is a local maxima if for all its neighbors j in G, $f(i) > f(j)$. A node i is a local minima if for all its neighbors j in G, $f(i) < f(j)$.

In last section, we defined the unnormalized hypergraph Laplacian matrix as $L = D_v - HWD_e^{-1}H^T$ and the normalized hypergraph Laplacian matrix as $L_{sym} := D_v^{-1/2}LD_v^{-1/2}$. In this section, we will simply use L as either of the unnormalized hypergraph Laplacian matrix or the normalized hypergraph Laplacian matrix.

Next, we will try to interpolate these minima and maxima to get $e_{\min}$ and $e_{\max}$. Let B be the sets of nodes where the signal is known and U be the sets of nodes where the signal is unknown. Then will will solve a Dirichlet problem on the hypergraph. Because of formula (7.2), the problem is finding s that minimizes $s^T Ls$ with restriction $s(b) = s_B(b)$ for $b \in B$. We can re-order the nodes and write L as $\begin{pmatrix} L_B & R \\ R^T & L_U \end{pmatrix}$. Then the Dirichlet problem is reduced as a problem solving a linear system $L_U s_U = -R^T s_B$.

Definition 7.3.1 (Hypergraph Empirical Mode Decomposition)**.** *Like the traditional EMD, HEMD will break down a signal into its component intrinsic mode functions (IMFs). For a given signal $f \in V$), the operator $\mathcal{S}[f]$ can be defined as following steps:*

- *Find all extrema of $f(i), i \in V$.*
- *Use the interpolate method introduced above, write the interpolation function as $e_{min}(i)$ and $e_{max}(i)$.*
- *Take the mean of the two interpolation functions $m(i) = (e_{min}(i) + e_{max}(i))/2$.*
- *The remaining part of the signal is $\mathcal{S}[f](i) = f(i) - m(i)$.*

The above steps would be repeated until some criteria conditions are met. At last, we can decompose f into

$$f(i) = m_K[f](i) + \sum_{k=1}^{K} d_k[f](i).$$

CHAPTER 8

Experiments

In this chapter, we will test our proposed algorithms on some data sets.

8.1 Diffusion Map

In this section, we use a dataset consists of 101 animals from a zoo(from UCI machine learning, link: https://www.kaggle.com/uciml/zoo-animal-classification). There are 16 different variables with various traits to describe the animals. Here are a list for these traits:

name of attribute	type of value domain
hair	Boolean
feathers	Boolean
eggs	Boolean
milk	Boolean
airborne	Boolean
aquatic	Boolean
predator	Boolean
toothed	Boolean
backbone	Boolean
breathes	Boolean
venomous	Boolean
fins	Boolean
legs	Numeric (set of values: 0,2,4,5,6,8)
tail	Boolean
domestic	Boolean
catsize	Boolean

Each of the animal belongs to one of the categories, which are mammal, bird, reptile, fish, amphibian, bug and invertebrate. Our goal will be, based on these traits, trying to predict the

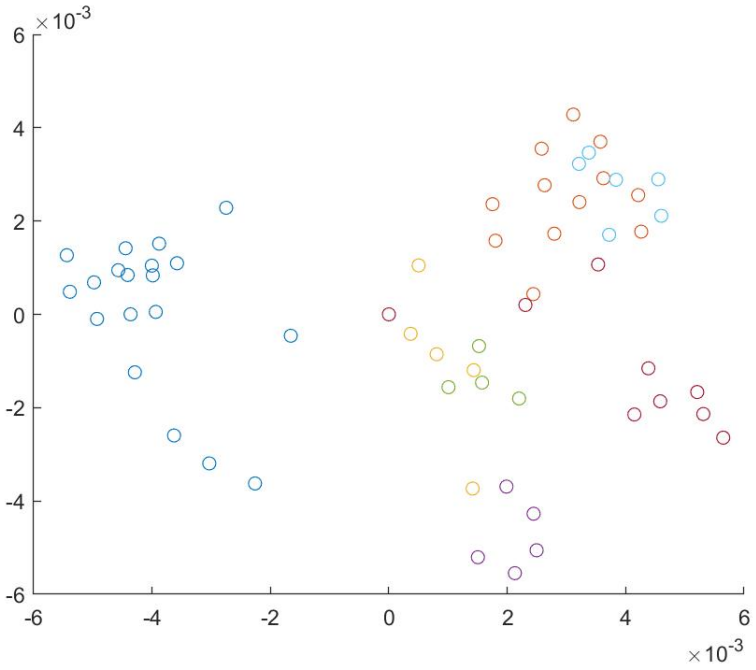

Figure 8.1: Visualization of Truncated Diffusion Map in section 8.1.

classification of the animal.

Based on the data, we can build a hypergraph with 101 vertices and 36 hyperedges. 30 of the hyperedges come from the Boolean value variables. For example, one hyperedge would be the set of animals with hair and another hyperedge would be the set without hair. 6 of the hyperedges come from the variable 'legs'. For example, one hyperedge could be the set of animals with 0 leg. Then we can build our diffusion map as described before. Usually, we can take some small integer d and have the corresponding truncated diffusion map. A visualization of truncated ($d = 2$) diffusion map is shown in Figure 8.1.

At last, we can use some unsupervised classification algorithms, such as KNN or SVM, to make the classification.

8.2 Wavelet

When the wavelet kernel g is carefully chosen, the spectral hypergraph wavelet transform would perform localization in good scales. In our experiments, we take our wavelet kernel function as

$$g(x) = \begin{cases} x & \text{for } x < 1 \\ -5 + 11x - 6x^2 + x^3 & \text{for } 1 \leqslant x \leqslant 2 \\ \frac{1}{x} & \text{for } x > 2. \end{cases} \tag{8.1}$$

The wavelet scales t_j will be picked between t_1 and t_J, in which t_1 and t_J will be determined by the values of the hypergraph Laplacian L. Suppose λ_{max} is the largest eigenvalue of L and $\lambda_{min} = \frac{\lambda_{max}}{K}$, where K is a parameter for the wavelet transform. Pick $t_1 = \frac{2}{\lambda_{min}}$ and $t_J = \frac{1}{\lambda_{max}}$. Then the function $g(t_1 x)$ has decay for $x > \lambda_{min}$ and $g(t_J x)$ has linear for $x < \lambda_{max}$.

8.2.1 Swiss Roll

In the first example, we will make the spectral hypergraph wavelet transform on a Swiss roll data cloud. The data points are generated by a map $\mathbb{R}^2 \to \mathbb{R}^3 : (s,t) \mapsto \left(\frac{t\cos(t)}{4\pi}, s, \frac{t\sin(t)}{4\pi}\right)$, where (s,t) is uniformily random distributed on $[-1,1] \times [\pi, 4\pi]$. We will generate 400 points for our experiment.

Each point would be one vertex in our hypergraph. We then will construct same number of hyperedges as the number of vertices. For each vertex v_i, the corresponding hyperedge e_{v_i} contains itself v_i and vertices v_j if $\mathrm{d}(v_i, v_j) < D$, where $D > 0$. Then we could get the matrix H. The weight matrix is $I_{400\times 400}$, which means the hypergraph is unweighted.

Suppose the original siginal is a Dirac delta function on one node of the Swiss roll. Figure 8.2 shows the wavelets with $J = 3$ scales and $K = 20$.

8.2.2 Minnesota Road Hypergraph

In this example, we build a hypergraph of Minnesota road network. The original data is a graph, so we generate the hypergraph from the original graph. First, we take all vertices in the graph as the vertices in the hypergraph. Here each vertex stands one town or one intersection of roads. Second, we construct one hyperedge for each vertex v_i. The hyperedge consists 1), v_i, 2); v_j if $\mathrm{d}(v_i, v_j) < D$; 3), v_i and v_j share a edge in the graph.

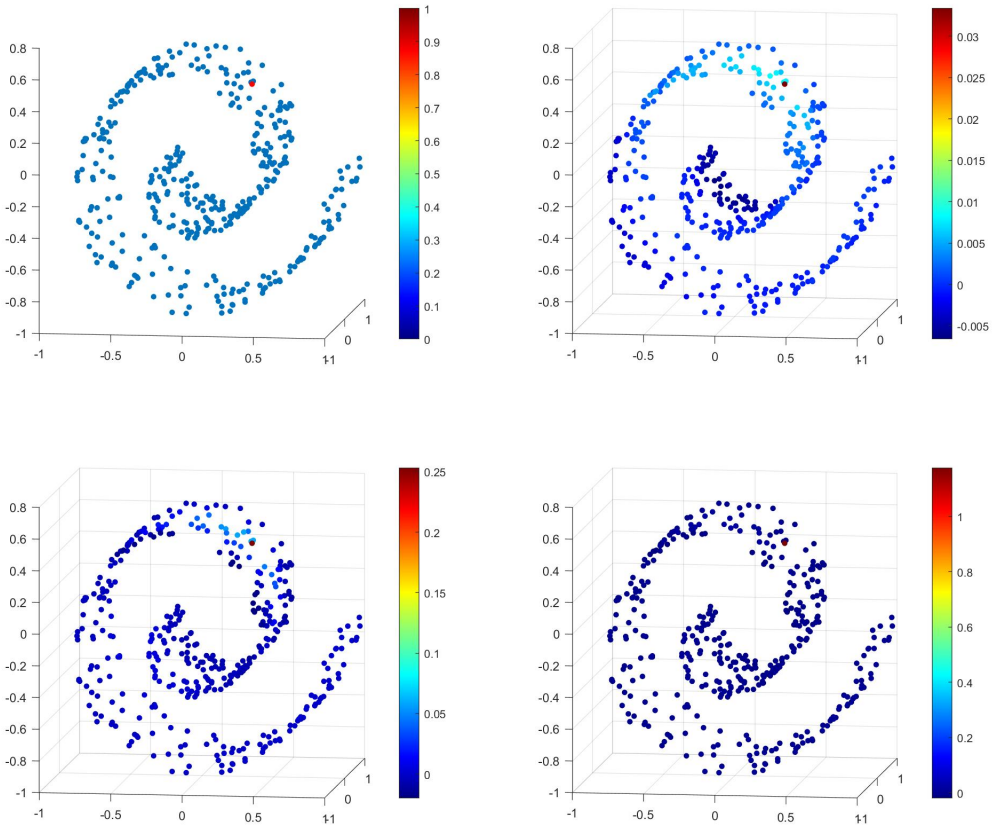

Figure 8.2: Spectral hypergraph wavelet transform on a Swiss roll in section 8.2.1.

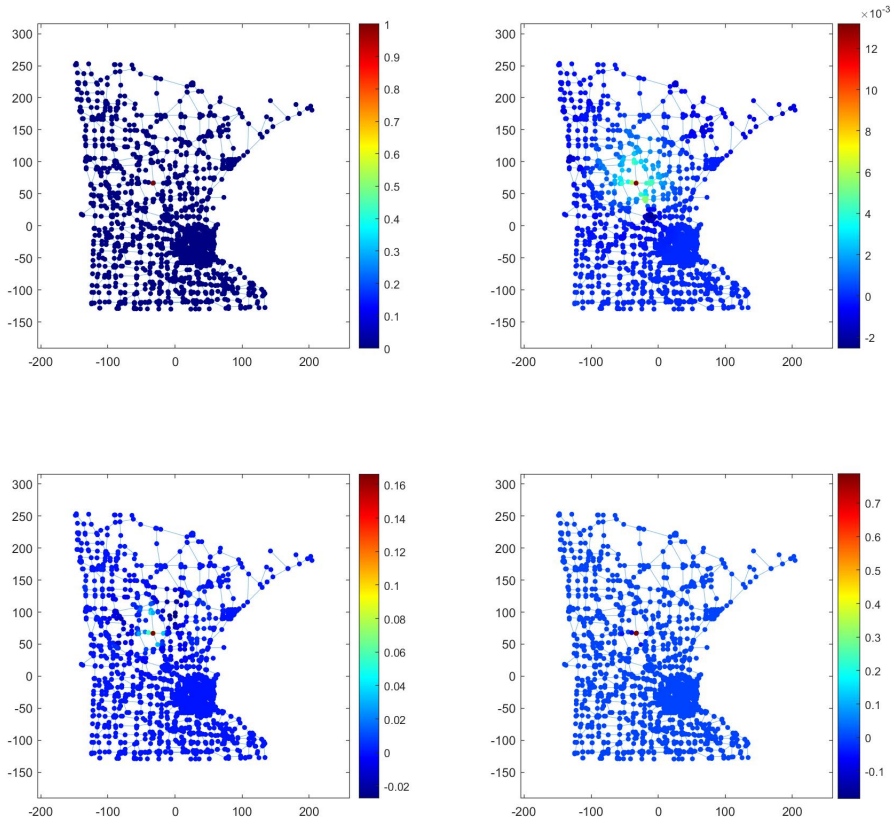

Figure 8.3: Spectral hypergraph wavelet transform on Minnesota road map in section 8.2.2.

Suppose the original siginal is a Dirac delta function on one node of the Minnesota road. Figure 8.3 shows the wavelets with $J = 3$ scales and $K = 100$.

8.3 Hypergraph Empirical Mode Decomposition

Suppose there are $N = 1600$ sensors uniformly randomly distributed on the 2D manifold $[0, 1] \times [0, 1]$. Let's assume the orignal signal was a superposition of two sine waves. Then there would be three parts including the noise: 1), $f_1(x, y) = \sin(7\sqrt{2}\pi|x + y|)$, which is a sine function with an angle $\pi/4$ with the horizontal; 2), $f_2(x, y) = 2\sin(4\pi x)$which is a horizontal sine function; 3), a uniform noise from $[-0.5, 0.5]$.

Each sensor would be one vertex in our hypergraph. We then will construct same number of hyperedges as the number of vertices. For each vertex v_i, the corresponding hyperedge e_{v_i} contains

itself v_i and vertices v_j if $\mathrm{d}(v_i, v_j) < D$, where $D = 0.025$ in our example.

Figure 8.4 shows the first two IMFs and the residue of the Hypergraph EMD algorithm.

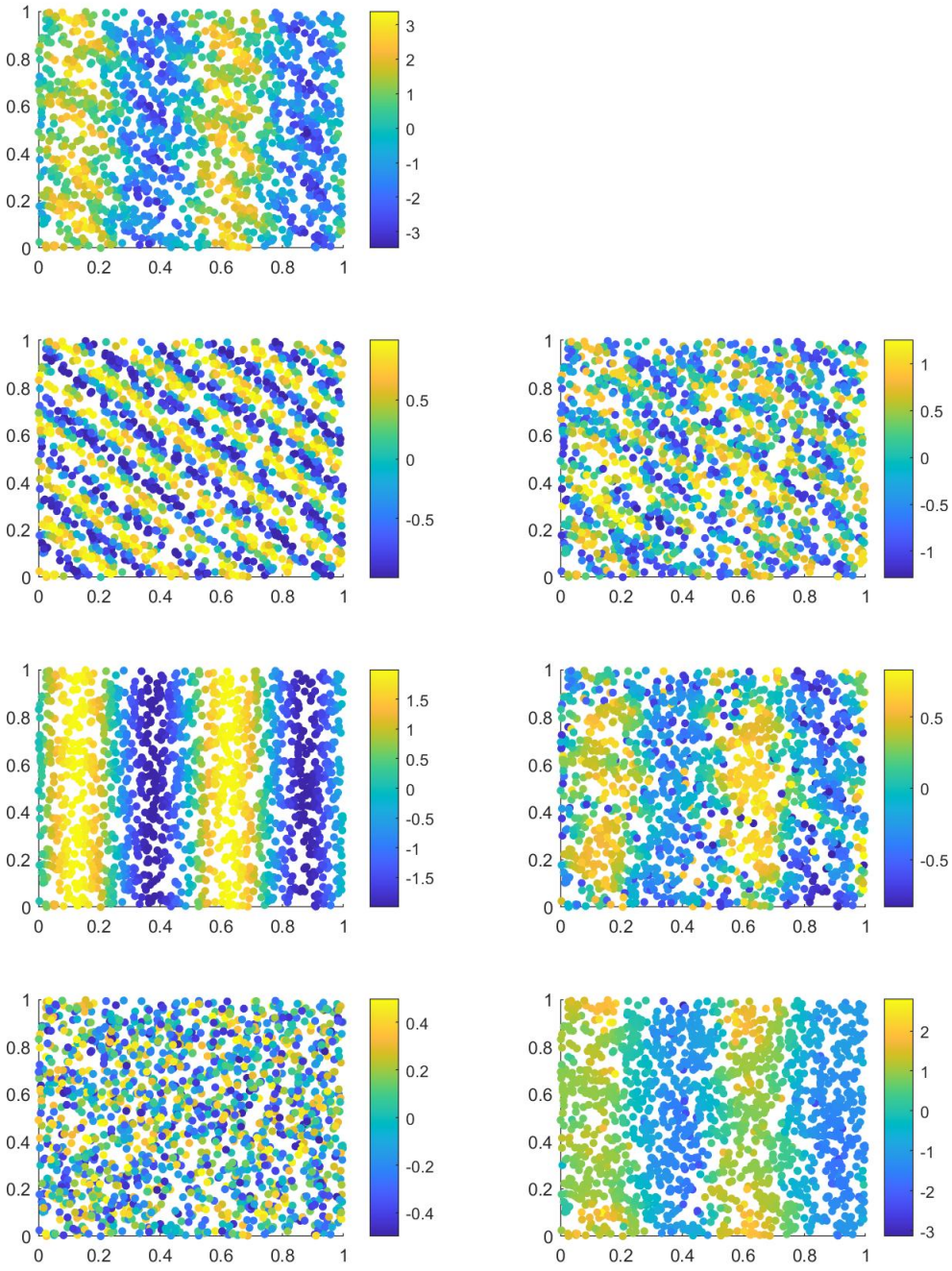

Figure 8.4: Hypergraph EMD on sensor network. Left column: the original signal and its three components. Right column: the first two IMFs and the residue which is uncovered by HEMD in section 8.3.

BIBLIOGRAPHY

[1] R. Adler, B. Kitchens, N. Martens, C. Pugh, M. Shub, C. Tresser, Convex dynamics and applications, Ergodic Theory and Dynamical Systems 25, 321–352 (2005).

[2] R. Adler, T. Nowicki, G. Świrszcz, C. Tresser, S. Winograd, Error diffusion on acute simplices: geometry of invariant sets, Chaotic Modelling and Simulation 1, 3–26 (2015).

[3] R. Adler, T.G. Nowicki, Świrszcz, C. Tresser, Convex dynamics with constant input, Ergodic Theory and Dynamical Systems 30, 957–972 (2010).

[4] J. Benedetto, A.M. Powell and Ö. Yılmaz, Sigma-Delta ($\Sigma\Delta$) quantization and finite frames. IEEE Transactions on Information Theory 52, 1990–2005 (2006)

[5] J. Benedetto, O. Oktay and A. Tangboonduangjit, Complex sigma-delta quantization algorithms for finite frames. In: Ólafsson, G., Grinberg, E., Larson, D., Jorgesen, P., Massopust, P., Quinto, E., Rudin, B. (eds.) Radon transforms, geometry, and wavelets, pp. 27–49. Contemporary Mathematics 464, American Mathematical Society, Providence, RI, (2008)

[6] J. Blum, M. Lammers, A.M. Powell and Ö. Yılmaz Sobolev duals in frame theory and Sigma-Delta quantization. Journal of Fourier Analysis and Applications 16, 365–381 (2010)

[7] B. Bodmann, Random fusion frames are nearly equiangular and tight. Linear Algebra and its Applications 439, no. 5, 1401–1414 (2013).

[8] B. Bodmann and V. Paulsen, Frame paths and error bounds for sigma-delta quantization. Applied and Computational Harmonic Analysis 22, no. 2, 176–197 (2007).

[9] P. Boufounos, G. Kutyniok and H. Rauhut, Holger, Sparse recovery from combined fusion frame measurements. IEEE Transactions on Information Theory 57, no. 6, 3864–3876 (2011).

[10] E. Chou and C.S. Güntürk, Distributed Noise-Shaping Quantization: I. Beta Duals of Finite Frames and Near-Optimal Quantization of Random Measurements. Constructive Approximation 44, no. 1, 1—22 (2016).

[11] P.G. Casazza and G. Kutyniok, Frames of subspaces, In: C. Heil, P. Jorgensen and D. Larson (eds.) Wavelets, frames and operator theory, pp. 87–113, Contemporary Mathematics 345, American Mathematical Society, Providence, RI (2004).

[12] P.G. Casazza, G. Kutyniok and S. Li, Fusion frames and distributed processing. Applied and Computational Harmonic Analysis 25, 114–132 (2008).

[13] P.G. Casazza, G. Kutyniok, S. Li and C. Rozell, Modelling sensor networks with fusion frames. Wavelets XII (San Diego, CA, 2007), SPIE Proceedings 6701, SPIE, Bellingham, WA (2007).

[14] I. Daubechies and R. DeVore, Approximating a bandlimited function using very coarsely quantized data: A family of stable sigma-delta modulators of arbitrary order. Annals of Mathematics 158, 679–710 (2003).

[15] P. Deift, C.S. Güntürk and F. Krahmer, An optimal family of exponentially accurate Sigma-Delta quantization schemes, Communications on Pure and Applied Mathematics, Vol. LXIV, 0883–0919 (2011).

[16] J.-M. Feng and F. Krahmer, An RIP-based approach to $\Sigma\Delta$ quantization for compressed sensing, IEEE Signal Processing Letters 21, no. 11, 1351–1355 (2014).

[17] J.-M. Feng, F. Krahmer and R. Saab, Quantized compressed sensing for random circulant matrices. Applied and Computational Harmonic Analysis 47, no. 3, 1014–1032 (2019).

[18] R. Gray, Quantization noise in single-loop Sigma-Delta modulation with sinusoidal inputs, IEEE Transactions on Communications 37, no. 9, 956 – 968 (1989).

[19] C.S. Güntürk, One-bit Sigma-Delta quantization with exponential accuracy, Communications on Pure and Applied Mathematics, Vol. LVI, 1608–1630 (2003).

[20] C.S. Güntürk, Approximating a bandlimited function using very coarsely quantized data: improved error estimates in sigma-delta modulation. Journal of the American Mathematical Society 17, 229–242 (2004).

[21] C.S. Güntürk, M. Lammers, A.M. Powell, R. Saab and Ö. Yılmaz, Sobolev duals for random

frames and $\Sigma\Delta$ quantization of compressed sensing measurements. Foundations of Computational Mathematics 13, 1–36 (2013)

[22] C.S. Güntürk and N.T. Thao, Ergodic dynamics in sigma-delta quantization: tiling invariant sets and spectral analysis of error. Advances in Applied Mathematics 34, 523–560 (2005).

[23] H. Inose, Y. Yasuda and J. Murakami, A telemetering system by code modulation - $\Delta\Sigma$ modulation, IRE Transactions on Space Electronics and Telemetry, Volume: SET-8, Issue 3, 204–209 (1962).

[24] J. Jiang and A.M. Powell, Sigma-Delta quantization for fusion frames and distributed sensor networks, in: "Frames and other bases in abstract functions spaces," 101–124, Birkhäuser, 2017.

[25] F. Krahmer, An improved family of exponentially accurate sigma-delta quantization schemes. Proceedings SPIE 6701 (2007).

[26] F. Krahmer, R. Saab, R. Ward, Root-exponential accuracy for coarse quantization of finite frame expansions. IEEE Transactions on Information Theory 58, no. 2, 1069–1079 (2012).

[27] F. Krahmer, R. Saab, Ö. Yılmaz, Sigma-Delta quantization of sub-Gaussian frame expansions and its application to compressed sensing. Information and Inference 3, no. 1, 40–58 (2014).

[28] F. Krahmer, R. Ward, Lower bounds for the error decay incurred by coarse quantization schemes. Applied and Computational Harmonic Analysis 31, 131–138 (2012).

[29] G. Luckjiff and I. Dobson, Hexagonal $\Sigma\Delta$ modulators in power electronics, IEEE Transactions on Power Electronics 20, 1075–1083 (2005).

[30] S. R. Norsworthy, R. Schreier and G. C. Temes, Delta-Sigma Data Converters, Theory, Design, and Simulation, IEEE Press, 1997.

[31] R. Saab, R. Wang and Ö. Yılmaz, Quantization of compressive samples with stable and robust recovery. Applied and Computational Harmonic Analysis 44, no. 1, 123–143 (2018).

[32] Ö. Yılmaz, Stability analysis for several second-order sigma-delta methods of coarse quantization of bandlimited functions. Constructive Approximation 18, 599–623 (2002).

[33] Ö. Yılmaz, Coarse quantization of highly redundant time-frequency representations of square-integrable functions, Applied and Computational Harmonic Analysis, 14, no. 2, 107–132 (2003).

[34] G. Rilling and P. Flandrin, One or Two Frequencies? The Empirical Mode Decomposition Answers, IEEE Transactions on Signal Processing, vol. 56, no. 1, pp. 85-95, Jan. 2008.