Check for updates

# Detecting Edges from Non-uniform Fourier Data via Sparse Bayesian Learning

**Victor Churchill[1] · Anne Gelb[1]**

## Abstract
In recent investigations, the problem of detecting edges given non-uniform Fourier data was reformulated as a sparse signal recovery problem with an $\ell_1$-regularized least squares cost function. This result can also be derived by employing a Bayesian formulation. Specifically, reconstruction of an edge map using $\ell_1$ regularization corresponds to a so-called type-I (maximum a posteriori) Bayesian estimate. In this paper, we use the Bayesian framework to design an improved algorithm for detecting edges from non-uniform Fourier data. In particular, we employ what is known as type-II Bayesian estimation, specifically a method called sparse Bayesian learning. We also show that our new edge detection method can be used to improve downstream processes that rely on accurate edge information like image reconstruction, especially with regards to compressed sensing techniques.

**Keywords** Edge detection · Non-uniform Fourier data · Sparse Bayesian learning · Signal reconstruction · Regularization

**Mathematics Subject Classification** 94A12 · 62F15 · 62J05 · 65F22

## 1 Introduction

Edge detection is an important tool in identifying physical structures and regions of interest in signals and images. In magnetic resonance imaging (MRI), edge detection helps tissue boundary identification [26,29]. In synthetic aperture radar (SAR), it can improve target identification. In both of these applications, data are collected as non-uniform Fourier samples. Detecting edges specifically from non-uniform[1] Fourier data has been explored in [16,18,27]. Most recently, the fact that the edges of many signals and images are sparse was utilized to

✉ Victor Churchill
   victor.a.churchill.gr@dartmouth.edu

1  Department of Mathematics, Dartmouth College, Hanover, USA

[1] Note that while here we only explicitly consider non-uniform Fourier samples, all methods described here apply to uniform Fourier samples as well.

reformulate edge detection from non-uniform Fourier data as a sparse signal recovery (SSR) problem [18]. Groundwork for that reformulation was laid in [16,27,31,37]. The technique in [18] used Fourier frames to construct a forward model for edge detection from given non-uniform Fourier data. An edge map of a piecewise smooth function was then reconstructed using an $\ell_1$-norm regularization optimization procedure which is known to encourage sparsity [11,33].[2] For a variety of reasons including noise, non-uniform sampling, and the magnitude dependence of the $\ell_1$ norm, the solution via $\ell_1$ regularization to SSR is sometimes not as sparse as desired. Specifically regarding magnitude dependence, because it simply sums the absolute values of a sparse signal, the $\ell_1$ norm is not as sensitive in penalizing small spikes when signals contain both small and large spikes. While the edge reconstructions in [18] may be suboptimal due to these downsides of $\ell_1$ regularization, it was indeed a key development of [18], although not explicitly addressed, to view edge detection as SSR. We note that edge detection was also formulated as SSR in [24,32], albeit in different settings. With that door open, in this paper we explore another algorithm that has both empirically and theoretically outperformed $\ell_1$ regularization for SSR.

Capturing the sparsity of solutions more accurately in the SSR problem has been widely studied [4,8–10,21,25,34,38]. More recently, there has been increased interest in Bayesian probabilistic approaches to SSR [4,21,25,34,38]. Within the probabilistic approaches there are two categories. The first is type-I, or maximum a posteriori (MAP) Bayesian estimation which uses a fixed prior. The most popular examples of type-I methods for SSR are $\ell_1$ regularization [11,33], iteratively reweighted $\ell_1$ regularization [9], and iteratively reweighted $\ell_2$ regularization [10]. The second category is type-II, or evidence maximization Bayesian estimation which employs a hierarchical, flexible parametrized prior that is *learned* from the given data. For SSR [21] provides the most general and comprehensive comparison of type-I and type-II methods, and in particular it is shown there through extensive empirical results that type-II estimates outperform corresponding type-I estimates in terms of accuracy. In addition, type-II methods have the ability to retrieve a full posterior distribution for the solution rather than just a point estimate as in type-I. Also advantageous is the automatic estimation of the crucial regularization parameter, the choice of which is typically difficult and subjective in MAP schemes. In this paper we focus our efforts on a particular type-II method called sparse Bayesian learning (SBL), also called the relevance vector machine (RVM) [34]. In [21], SBL typically outperformed all other methods including $\ell_1$ regularization [8], iteratively reweighted $\ell_1$ [9], and iteratively reweighted $\ell_2$ regularization methods [10], as well as a variety of type-II methods for SSR. The strength of SBL is also supported by evidence from [25], which looked at SBL from the perspective of compressed sensing [8,14], and showed that SBL could recover a sparse signal with better accuracy than $\ell_1$ regularization using the same amount of data. There are also theoretical results from [28,39] that show that SBL is a closer surrogate to the $\ell_0$ norm than $\ell_1$ regularization, and proved that even in the worst case SBL still outperforms the most widely employed algorithms from compressed sensing. In [38], for the noiseless case, it is proved that the global minimum of the SBL cost function is achieved at a solution such that the posterior mean equals the maximally sparse solution. Furthermore, it was shown that local minima are achieved at sparse solutions, regardless of noise [38]. SBL employs the expectation–maximization (EM) algorithm [13]. Due to properties of the EM algorithm, SBL is globally convergent [38].

---

[2] Although ideally the $\ell_0$ semi-norm should be used to regularize this problem, the resulting optimization problem is NP-hard. Hence the $\ell_1$ norm has become a popular convex surrogate that makes the problem computationally tractable and also offers theoretical guarantees for exact reconstruction [8], as well as a variety of other benefits related to compressed sensing [14].

This paper adapts and implements the SBL algorithm from [34] for the non-uniform Fourier edge detection problem using the SSR formulation from [18]. We demonstrate improved accuracy compared with the results of [18] that used $\ell_1$ regularization in both noise-free and noisy scenarios. The organization of the paper is as follows. In Sect. 2, we set up the problem of edge detection from non-uniform Fourier data. In Sect. 3, we describe the reformulation of the problem as SSR and reproduce the edge detection results from [18]. Section 4 introduces the probabilistic approach, adapts SBL for the edge detection problem, and explains the resulting algorithm. We also look at numerical results for the new SBL technique, comparing with the results of Sect. 3. Edge detection is not only useful in and of itself, but can be used to improve signal and image reconstruction as well. In Sect. 5, we use our new edge detection algorithm to create a weighting matrix in order to apply $\ell_2$ regularization away from edges to achieve full image reconstruction results. This is so-called edge-adaptive $\ell_2$ regularization [12]. Finally in Sect. 6, we summarize this work and look at future directions for our research.

## 2 Problem Setup

We consider a one-dimensional periodic piecewise smooth function $f : [-1, 1] \to \mathbb{R}$. We define the jump function, $[f]$, of $f$ as the difference between the left- and right-hand limits of the function:

$$[f](x) = f(x^+) - f(x^-). \tag{1}$$

In smooth regions, $[f](x) = 0$. At a discontinuity, $[f](x)$ is equal to the height of the jump. Suppose we use $2J + 1$ grid points, $x_j = \frac{j}{J}$ for $j = -J, \ldots, J$. Assuming that the discontinuities of $f$ are separated such that there is at most one jump per cell $I_j = [x_j, x_{j+1})$, we can write

$$[f](x) = \sum_{j=-J}^{J-1} [f](x_j)\delta_{x_j}(x). \tag{2}$$

where the coefficient $[f](x_j)$ is the jump value in $I_j$ and $\delta_{x_j}(x)$ is the indicator function with

$$\delta_{x_j}(x) = \begin{cases} 1 & x = x_j \\ 0 & x \neq x_j \end{cases} . \tag{3}$$

Suppose we are given a finite sequence of non-uniform Fourier samples of $f$,

$$\hat{f}(\lambda_k) = \frac{1}{2} \int_{-1}^{1} f(x)e^{-i\pi\lambda_k x}dx, \tag{4}$$

where $\lambda_k \in \mathbb{R}$ and $k = -M, \ldots, M$. Note here that we are considering *continuous* Fourier samples, as generating data via the same model we use to solve the inverse problem commits the inverse crime. Specifically, we look at three types of non-uniform sampling that were considered in [18]. The first is *jittered sampling*, defined by

$$\lambda_k = k - \left\lfloor \frac{2M+1}{2} \right\rfloor - 1 + \frac{1 - 2\xi_k}{4} \tag{5}$$
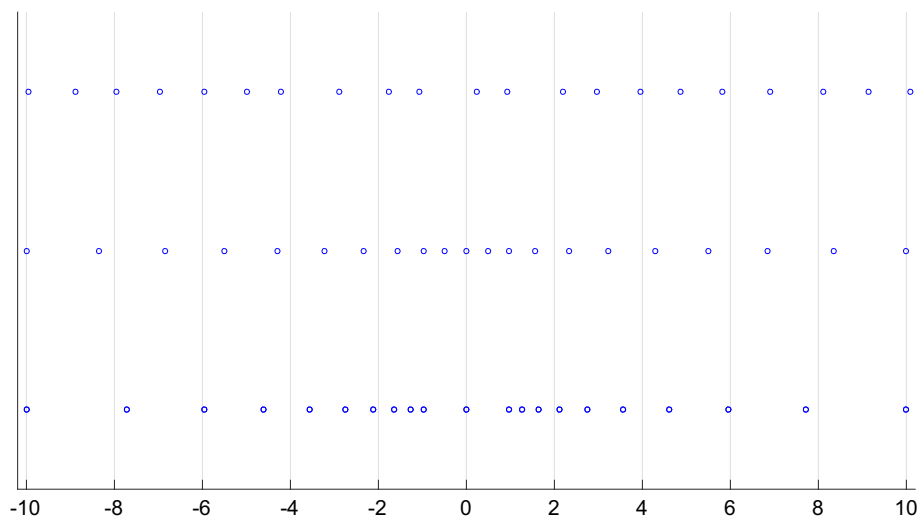
**Fig. 1** Modes $\lambda_k$ ($M = 10$) given by (top) jittered; (middle) quadratic; and (bottom) logarithmic sampling

where $\xi_k \sim U([0, 1])$. Jittered sampling imitates Cartesian grid sampling with small errors that often occur in imaging systems. The next is *quadratic sampling*, defined by

$$\lambda_k = \frac{\text{sign}(k)k^2}{M}. \tag{6}$$

This pattern simulates a cross-section of a non-Cartesian sampling pattern that undersamples high frequencies and oversamples low frequencies. Lastly, we consider *logarithmic sampling*, which oversamples low frequencies and, even more sparsely than quadratic, undersamples high frequencies. In particular, $\log|\lambda_k|$ is evenly distributed between $-v$ and $\log M$ with $v > 0$ and $2M + 1$ being the total number of samples. Figure 1 gives a visualization of these three sampling types. We will also consider the case where the underlying Fourier data in (4) are noisy, given by

$$\hat{f}^\eta(\lambda_k) = \hat{f}(\lambda_k) + \eta_k, \tag{7}$$

for $k = -M, \ldots, M$. Here $\eta_k \sim \mathcal{CN}(0, \sigma^2)$, meaning $\eta_k$ is a complex Gaussian random variable with mean 0 and variance $\sigma^2$.

From data given by (4) or (7), our goal is to accurately reconstruct the edge function $[f](x)$ defined in (1). In the following section, we review the method of [18], which shows how non-uniform Fourier edge detection can be formulated as SSR. Later, we see the same formulation in probabilistic terms, and then go further into the SBL framework to achieve superior results.

## 3 Formulating Edge Detection as SSR

In [18], the authors solve an optimization problem to choose a concentration factor [19], followed by a reconstruction via optimizing an $\ell_1$-regularized cost function. The $\ell_1$ regularized edge detection method described below follows the derivation in [18].

### 3.1 Concentration Factor Edge Detection for Uniform Fourier Data

The concentration factor edge detection method, originally developed in [19], approximates (2) from $2M + 1$ uniform Fourier coefficients given in (4) where $\lambda_k = k$ as

$$S_M^\sigma[f](x) = i \sum_{k=-M}^{M} \hat{f}(k)\operatorname{sgn}(k)\sigma\left(\frac{|k|}{M}\right)e^{\pi ikx}. \tag{8}$$

Here $\sigma$, coined the concentration factor in [19], satisfies certain admissibility conditions

1. $\sum_{k=1}^{M} \sigma\left(\frac{|k|}{M}\right)\sin(k\pi x)$ is odd
2. $\frac{\sigma(u)}{u} \in C^2(0,1)$
3. $\int_v^1 \frac{\sigma(u)}{u} du \to -1$, $v = v(M) > 0$ being small.

If all three conditions are satisfied, $S_M^\sigma[f]$ concentrates at the singular support of $f$ and the jump function approximation observes the concentration property

$$S_M^\sigma[f](x) = [f](x) + \begin{cases} \mathcal{O}\left(\frac{\log M}{M}\right) & d(x) \leq \frac{\log M}{M} \\ \mathcal{O}\left(\frac{\log M}{(Md(x))^s}\right) & d(x) \gg \frac{1}{M} \end{cases} \tag{9}$$

where $d(x)$ is the distance between a point in the domain and the nearest discontinuity, and $s > 0$ depends on $\sigma$. Generally, the convergence of (8) depends on the particular choice of $\sigma$.

### 3.2 Fourier Frame Approximation

The concentration factor method cannot be extended directly to non-uniform Fourier coefficients because $\{e^{i\pi\lambda_k x}\}_{k=-M}^{M}$ is generally not an orthogonal basis. However, it can be adapted by using the finite Fourier frame approximation [18,30]. We will employ the Fourier frame framework for our new edge detection method. The technique in [18] is briefly described below.

We require the following two definitions:

**Definition 1** A **frame** for a Hilbert space $\mathcal{H}$ is a sequence of vectors $\{\varphi_k : k \in \mathbb{Z}\} \subseteq \mathcal{H}$ for which there exists constants $0 < A \leq B < \infty$ such that, for every $f \in \mathcal{H}$, we have

$$A||f||^2 \leq \sum_{k\in\mathbb{Z}} |\langle f, \varphi_k\rangle|^2 \leq B||f||^2. \tag{10}$$

**Definition 2** If $\{\varphi_k : k \in \mathbb{Z}\} \subseteq \mathcal{H}$ is a frame for $\mathcal{H}$ then the associated **frame operator** $S : \mathcal{H} \to \mathcal{H}$ is defined as

$$Sf = \sum_{k\in\mathbb{Z}} \langle f, \varphi_k\rangle\varphi_k. \tag{11}$$

Frame elements span $\mathcal{H}$ but are not necessarily linearly independent. The frame operator is bounded, invertible, positive, and self-adjoint. In this investigation we consider $\mathcal{H} = L^2(-1, 1)$ and $\varphi_k(x) = e^{i\pi\lambda_k x}$. Hence we can reconstruct $f$ via

$$f = S^{-1}Sf = \sum_{k\in\mathbb{Z}} \langle f, \varphi_k\rangle S^{-1}\varphi_k = \sum_{k\in\mathbb{Z}} \hat{f}(\lambda_k)\tilde{\varphi}_k, \tag{12}$$

where $\tilde{\varphi}_k = S^{-1}\varphi_k$ for $k \in \mathbb{Z}$ is the canonical dual frame and $\hat{f}(\lambda_k)$ is the given non-uniform Fourier data as in (4). In general there is no closed form of $S^{-1}$. Various algorithms construct (finite-dimensional) approximations to $S^{-1}$. Here we use the *admissible frame* method developed in [30], which obtains a finite-dimensional approximation of $\tilde{\varphi}_k$ by projecting $\{\varphi_k : k \in \mathbb{Z}\}$ onto an admissible frame $\{\psi_l : l \in \mathbb{Z}\}$ defined as

**Definition 3** A frame $\{\psi_l : l \in \mathbb{Z}\}$ is **admissible** with respect to the frame $\{\varphi_k : k \in \mathbb{Z}\}$ if the following two conditions hold:

1. It is intrinsically self-localized, that is, there exists $c_0 \in \mathbb{R}^+$ and $t > 1$ such that

$$|\langle \psi_k, \psi_l \rangle| \le c_0(1 + |k - l|)^{-t}.$$

2. There exists $c_1 \in \mathbb{R}^+$ and $s > 1/2$ such that

$$|\langle \varphi_k, \psi_l \rangle| \le c_1(1 + |k - l|)^{-s}.$$

If $\{\psi_l : l \in \mathbb{Z}\}$ is admissible with respect to $\{\varphi_k : k \in \mathbb{Z}\}$, then $\{\tilde{\varphi}_k : k \in \mathbb{Z}\}$ can be approximated by

$$\tilde{\varphi}_k \approx \sum_{l=-N}^{N} b_{l,k}\psi_l =: \tilde{\varphi}_{N,k}, \qquad (13)$$

where $\{b_{l,k}\}_{l=-N,k=-M}^{N,M}$ are the entries of $\mathbf{B} = \Psi^\dagger$. That is, $\mathbf{B}$ is the Moore–Penrose pseudoinverse of $\Psi$ where

$$\Psi_{k,l} = [\langle \varphi_k, \psi_l \rangle] \quad k = -M, \dots, M, \quad l = -N, \dots, N. \qquad (14)$$

Since in this investigation $\varphi_k(x) = e^{i\pi\lambda_k x}$, as shown in [30], a practical admissible frame is $\psi_l(x) = e^{i\pi l x}$, yielding the *sinc* approximation

$$\langle \varphi_k, \psi_l \rangle = 2\frac{\sin(\pi(\lambda_k - l))}{\pi(\lambda_k - l)}. \qquad (15)$$

**Remark 1** We note that except for the case of jittered sampling in (5), $\{\varphi_k\}$ may not actually constitute a frame except for its finite span. Nevertheless, as discussed in [17], the admissible frame approach is still effective.

Since the conditioning of (14) depends on the difference $\lambda_k - l$, more uniformly spaced samples yield better conditioning. To complete the reconstruction of $f$, we plug in the approximation of $\tilde{\varphi}_k$ in (13) into (12) to get

$$T_M f = \sum_{l=-N}^{N} \sum_{k=-M}^{M} \hat{f}(\lambda_k)b_{l,k}\psi_l = \sum_{k=-M}^{M} \hat{f}(\lambda_k)\tilde{\varphi}_{N,k}(x). \qquad (16)$$

### 3.3 Concentration Factor Design

Returning to detecting edges from non-uniform Fourier data, since we cannot use (8), we instead use (16) to approximate (2) by

$$T_M^\sigma f(x) = \sum_{k=-M}^{M} \sigma_k \hat{f}(\lambda_k)\tilde{\varphi}_{N,k}(x). \qquad (17)$$

The problem of choosing a concentration factor still remains. We need $\sigma \in \mathbb{C}^{2M+1}$ so that for each reconstruction point $x_j = \frac{j}{J}$, $j = -J, \ldots, J$, we have

$$T_M^\sigma f(x_j) = \sum_{k=-M}^{M} \sigma_k \hat{f}(\lambda_k) \tilde{\varphi}_{N,k}(x_j) \approx [f](\xi) \delta_\xi(x_j). \tag{18}$$

An optimization problem for determining the concentration factor $\{\sigma_k\}_{k=-M}^{M}$ is developed in [18], which is reviewed below for the case when $f$ has a single jump discontinuity at $x = \xi \in (-1, 1)$. Since (18) is linear with respect to the Fourier data, these results also hold for functions with multiple jumps.

To proceed, we first approximate $f$ as a superposition of scaled and shifted ramp functions, $r : [-1, 1] \to \mathbb{R}$, defined by

$$r(x) = \begin{cases} -\frac{x+1}{2} & \text{if } x \leq 0 \\ -\frac{x-1}{2} & \text{if } x > 0 \end{cases}. \tag{19}$$

Observe that the jump function for $r_\xi(x) = r(x - \xi)$ is

$$[r_\xi](x) = \begin{cases} 1 & \text{if } x = \xi \\ 0 & \text{if } x \neq \xi \end{cases}. \tag{20}$$

While $f(x)$ is only approximated to first order by

$$f(x) \approx a r_\xi(x), \tag{21}$$

where $a$ is the corresponding jump height, the jump function is exactly $[f](x) = a[r_\xi](x)$.[3] Substituting (21) into (18), translating $\xi$ to 0, and canceling $a$, leads to

$$T_M^\sigma r(x_j) = \sum_{k=-M}^{M} \sigma_k \hat{r}(\lambda_k) \tilde{\varphi}_{N,k}(x_j) \approx \delta_0(x_j). \tag{22}$$

Here $\hat{r}(\lambda_k)$ are the Fourier coefficients for $r(x)$ given by

$$\hat{r}(\lambda_k) = \begin{cases} 0 & \lambda_k = 0 \\ \frac{(\sin(\pi \lambda_k) - \pi \lambda_k)i}{(\pi \lambda_k)^2} & \lambda_k \neq 0. \end{cases} \tag{23}$$

Because $\delta_\xi(x)$ has a trivial Fourier expansion, we consider a regularized approximation of $\delta_\xi(x)$,

$$h_\xi(x) = h\left(\frac{x - \xi}{\epsilon}\right) \approx \delta_\xi(x), \quad \epsilon > 0,$$

for some $h$ that is (essentially) compactly supported in $[\xi - \epsilon, \xi + \epsilon]$ with $h_\xi(\xi) = 1$. As the parameter $\epsilon$ increases the approximation is more regularized, but the edges are not as well localized. The authors of [18] suggest using $\epsilon = 0.07$. Replacing $\delta_0(x)$ in (22) gives

$$\sum_{k=-M}^{M} \sigma_k \hat{r}(\lambda_k) \tilde{\varphi}_{N,k}(x_j) \approx h_0(x_j) \approx \sum_{k=-M}^{M} \hat{h}_0(\lambda_k) \tilde{\varphi}_{N,k}(x_j). \tag{24}$$

---

[3] A complete derivation explaining how approximating periodic piecewise smooth functions as sums of scaled and shifted ramp functions can be effectively used to design concentration factors from uniform Fourier coefficients can be found in [36].

Hence

$$\sigma_k = \frac{\hat{h}_0(\lambda_k)}{\hat{r}(\lambda_k)}, \quad k = -M, \ldots, M. \tag{25}$$

Now we know how to compute $\sigma$, but the problem of choosing an optimal $h$ still remains. In many cases, $f$ is in a more restrictive class of functions like piecewise polynomials or piecewise trigonometric polynomials. Assuming $f$ has just one edge at $x = \xi$,

$$f(x) = s(x) + [f](\xi) r_\xi(x) \tag{26}$$

where $s(x)$ is the continuous part of $f$. By adding the constraint

$$T_M^\sigma s(x_j) \approx 0$$

for all $x_j \in (-1, 1)$, we are requiring that smooth regions of $f$ be drawn to zero in the approximation of $[f](x)$. With these tools in hand, we can now develop an algorithm to construct an optimal $\hat{\mathbf{h}} = \{\hat{h}_0(\lambda_k)\}_{k=-M}^{M}$ by formulating an optimization problem rather than using a pre-determined regularization that makes no assumption on $s(x)$. Specifically, we determine that $\hat{\mathbf{h}}$ must satisfy two contraints

$$\sum_{k=-M}^{M} \frac{\hat{h}(\lambda_k)}{\hat{r}(\lambda_k)} \hat{s}(\lambda_k) \tilde{\varphi}_{N,k}(x_j) \approx 0 \quad \text{and} \quad \sum_{k=-M}^{M} \hat{h}(\lambda_k) \tilde{\varphi}_{N,k}(x_j) \approx \delta_0(x_j). \tag{27}$$

---

**Algorithm 1** Concentration factor design for non-uniform Fourier data

---

1: Given $2M + 1$ Fourier coefficients, $\hat{f}(\lambda_k)$, of a piecewise smooth function, $f$, as in (4).
2: Choose $s(x)$ to be consistent with $f(x) = s(x) + [f](\xi) r_\xi(x)$ and define

$$\mathbf{F} = \left[\tilde{\varphi}_{N,k}(x_j)\right]_{j=-J, k=-M}^{J,M} \quad \text{and} \quad \mathbf{S} = \left[\frac{\hat{s}(\lambda_k)}{\hat{r}(\lambda_k)} \tilde{\varphi}_{N,k}(x_j)\right]_{j=-J, k=-M}^{J,M}. \tag{28}$$

3: Determine $\hat{\mathbf{h}}$ as the minimizer of

$$\min_{\hat{\mathbf{h}} \in \mathbb{R}^{2M+1}} ||\mathbf{F}\hat{\mathbf{h}}||_1 + \mu ||\mathbf{S}\hat{\mathbf{h}}||_1 \quad \text{subject to} \quad h(0) = 1 \tag{29}$$

for $x_j = \frac{j}{J}$, $j = -J, \ldots, J$, and $\mu > 0$.
4: Define $\sigma_k = \frac{\hat{h}_k}{\hat{r}(\lambda_k)}$, $k = -M, \ldots, M$.

---

Per [18], we choose $\mu = 1000$ for Algorithm 1. We still have to choose $s(x)$, though, which depends on prior information about $f$. For example, if $f$ has discontinuities in its derivative as well, we may choose a hat function

$$s_1(x) = \begin{cases} -\frac{x+1}{2} & \text{if } -1 \leq x \leq 0 \\ \frac{x-1}{2} & \text{if } 0 < x \leq 1 \end{cases}. \tag{30}$$

On the other hand if $f$ is essentially piecewise constant or linear with smooth variation between jumps, a smooth varying function like

$$s_2(x) = \begin{cases} -\frac{(x+1)^3}{12} & \text{if } -1 \leq x \leq 0 \\ \frac{(x-1)^3}{12} - \frac{1}{6} & \text{if } 0 < x \leq 1 \end{cases}, \tag{31}$$

is more appropriate.

### 3.4 Sparsity Model for Edge Detection

We now develop a method for reconstructing the edge function $[f](x)$ using Algorithm 1 that takes advantage of our assumption that $[f](x)$ is sparse. We seek a solution $[f](x)$ that most closely matches the projection of $\sum_{l=1}^{L}[f](\xi_l)h_{\xi_l}(x)$ where $L$ is the number of edges onto the space spanned by the Fourier frame elements given by (18). To this end, we adopt the waveform matching idea developed in [31] for uniform Fourier data. As will be evident in what follows, this approach allows us to build a forward model for reconstructing $[f](x)$. We start by defining the waveform kernel as the partial Fourier sum approximation of $h_0(x)$ given by

$$W_M^{\sigma}(x) = \frac{1}{\gamma_M^{\sigma}} \sum_{k=-M}^{M} \sigma_k \frac{\cos kx}{k}, \tag{32}$$

where $\sigma$ is the concentration factor vector found using Algorithm 1 and $\gamma_M^{\sigma}$ is a normalization constant. We can then approximate

$$W_M^{\sigma} * [f] \approx S_M^{\sigma}(f), \tag{33}$$

where $S_M^{\sigma}(f)$ is defined in (8) for some admissible $\sigma$. For non-uniform Fourier data, the corresponding waveform kernel is given by

$$W_{N(M)}^{\sigma}(x) = \frac{1}{\gamma_{N(M)}^{\sigma}} \sum_{k=-M}^{M} \sigma_k \hat{r}(\lambda_k) \tilde{\varphi}_{N,k}(x) = \frac{1}{\gamma_{N(M)}^{\sigma}} \sum_{k=-M}^{M} \sum_{l=-N}^{N} \sigma_k \hat{r}(\lambda_k) b_{l,k} e^{i\pi lx}, \tag{34}$$

where

$$\gamma_{N(M)}^{\sigma} = \sum_{k=-M}^{M} \sum_{l=-N}^{N} b_{l,k} \sigma_k \hat{r}(\lambda_k)$$

is the normalization constant. Analogous to (33) we now have

$$W_{N(M)}^{\sigma} * [f] \approx T_M^{\sigma}(f), \tag{35}$$

which is satisfied by requiring

$$\frac{1}{\gamma_{N(M)}^{\sigma}} \sum_{k=-M}^{M} b_{l,k} \sigma_k \hat{r}(\lambda_k) \widehat{[f]}(l) \approx \sum_{k=-M}^{M} b_{l,k} \sigma_k \hat{f}(\lambda_k) \quad l = -N, \ldots, N. \tag{36}$$

Hence we see that using the waveform kernel approach allows us to construct the model

$$\frac{1}{\gamma_{N(M)}^{\sigma}} (\mathbf{B}(\sigma \cdot \hat{\mathbf{r}})) \cdot (\mathbf{F}[f]) \approx \mathbf{B}(\sigma \cdot \hat{\mathbf{f}}) \tag{37}$$

where $\cdot$ denotes elementwise multiplication and $\mathbf{F}$ is the Fourier transform. We simplify the model as

$$\Theta[f] \approx \mathbf{y} \tag{38}$$

where

$$\Theta = \frac{1}{\gamma_{N(M)}^{\sigma}} \text{diag} \left[ \mathbf{B}(\sigma \cdot \hat{\mathbf{r}}) \right] \mathbf{F} \quad \text{and} \quad \mathbf{y} = \mathbf{B}(\sigma \cdot \hat{\mathbf{f}}). \tag{39}$$

Algorithm 2 combines (38) with $\ell_1$ regularization, used because of our assumption that the edges are sparse, to construct an approximation to $[f](x)$ at a set of grid points $x_j$, $j = -J, \ldots, J$. As in [18], we choose $\lambda = 0.01$ for Algorithm 2 when starting from noise-free data, and $\lambda = 0.025$ when starting from noisy data, reflecting the fact that in the noisy case we have less confidence in the fidelity term. We also consider the same three examples given there to compare our methods, given in Examples 1–3 below, and shown in Fig. 2. The results using $M = 100$ displayed in Figs. 3, 4, and 5, show generally good results for the jittered sampling case, but spurious oscillations appear for both quadratic and logarithmic sampling. Thus we see that the solutions are not sufficiently sparse, which may have undesirable consequences in downstream processing such as reconstruction. Furthermore, as displayed in Fig. 6, the results clearly deteriorate when Gaussian noise with mean 0 and standard deviation 0.02 is added to the given Fourier data. Note that throughout this paper to compare results we use the relative error defined by

$$RE(\mathbf{g}) = \frac{||\mathbf{g} - [\mathbf{f}]||_2}{||[\mathbf{f}]||_2}, \tag{40}$$

where $[\mathbf{f}]$ is the discretized true edge function.

---

**Algorithm 2** Reconstruction of $[f](x)$ from non-uniform Fourier data via $\ell_1$ regularization

---

1: Given $2M + 1$ Fourier coefficients, $\hat{f}(\lambda_k)$, of a piecewise smooth function, $f$, as in (4).
2: Determine $\sigma$ from Algorithm 1.
3: Reconstruct the jump function $[f]$ on gridpoints as

$$\mathbf{g}^* = \arg\min_{\mathbf{g}} ||\Theta\mathbf{g} - \mathbf{y}||_2 + \lambda||\mathbf{g}||_1. \tag{41}$$

---

**Example 1**

$$f_1(x) = \begin{cases} \cos\left(\pi x - \frac{\pi x}{2}\,\text{sign}(-x - 1/2)\right) & -1 \leq x \leq 0 \\ \cos\left(\frac{5\pi x}{2} + \pi x\,\text{sign}(x - 1/2)\right) & 0 < x \leq 1 \end{cases} \tag{42}$$
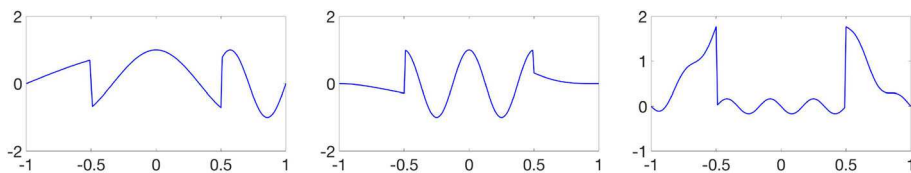


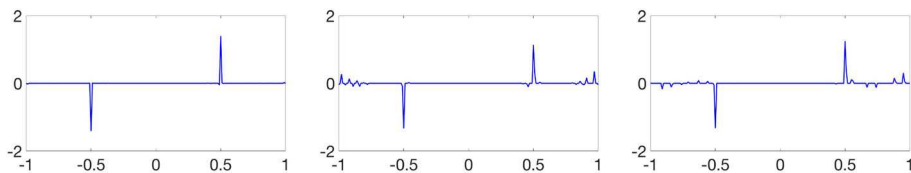**Fig. 2** (left) $f_1(x)$, (center) $f_2(x)$, and (right) $f_3(x)$



**Fig. 3** $[f_1]$ reconstructed via Algorithm 2 using (left) jittered; (center) quadratic; and (right) logarithmic sampling
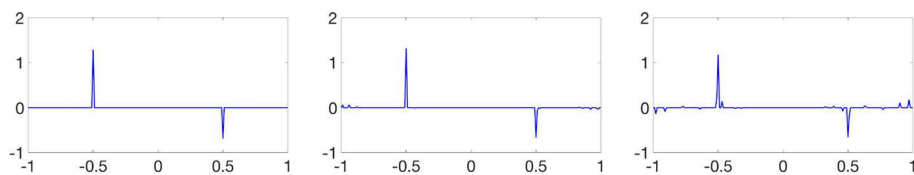
**Fig. 4** $[f_2]$ reconstructed via Algorithm 2 using (left) jittered; (center) quadratic; and (right) logarithmic sampling
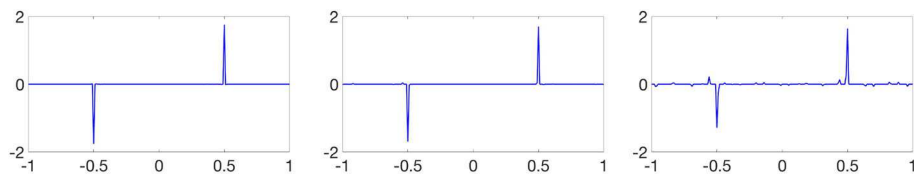


**Fig. 5** $[f_3]$ reconstructed via Algorithm 2 using (left) jittered; (center) quadratic; and (right) logarithmic sampling

with

$$[f_1](x) = \begin{cases} -\sqrt{2} & x = -1/2 \\ \sqrt{2} & x = 1/2 \\ 0 & |x| \neq 1/2 \end{cases}. \tag{43}$$

*Example 2*

$$f_2(x) = \begin{cases} -\frac{1}{2}(1 - x^2)^2 & -1 \leq x \leq -1/2 \\ \cos(4\pi x) & |x| < 1/2 \\ (1 - x^2)^4 & 1/2 \leq x \leq 1 \end{cases} \tag{44}$$

with

$$[f_2](x) = \begin{cases} \frac{42}{31} & x = -1/2 \\ -\frac{175}{256} & x = 1/2 \\ 0 & |x| \neq 1/2 \end{cases}. \tag{45}$$

*Example 3*

$$f_3(x) = \begin{cases} \pi(1 - x^2)^2 & 1/2 \leq |x| \leq 1 \\ -\frac{1}{6}\sin(6\pi x) & |x| < 1/2 \end{cases} \tag{46}$$

with

$$[f_3](x) = \begin{cases} -\frac{9}{16}\pi & x = -1/2 \\ \frac{9}{16}\pi & x = 1/2 \\ 0 & |x| \neq 1/2 \end{cases}. \tag{47}$$

## 4 Probabilistic Approach

As Figs. 3, 4 and 5 demonstrate, Algorithm 2 is somewhat effective in recovering edges of piecewise smooth functions from non-uniform Fourier data in cases without noise. However,
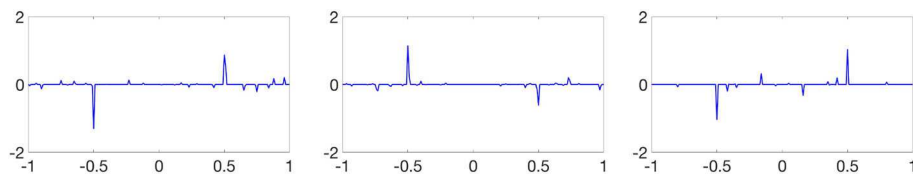
**Fig. 6** Reconstructions with Algorithm 2 from non-uniform Fourier data with zero-mean Gaussian noise with 0.02 standard deviation. (left)[$f_1$] from logarithmic sampling; (center) [$f_2$] from quadratic sampling; and (right) [$f_3$] from jittered sampling. The relative errors left to right are 0.295, 0.364, and 0.472. Because of the

as Fig. 6 shows, the quality of the edge approximation deteriorates as noise is introduced. While this suggests that the $\ell_1$ regularization term is no longer mitigating the poor fidelity, it is also apparent that as more noise is introduced, it is more difficult to obtain a truly sparse approximation. Hence choosing the regularization parameter $\lambda$ a-priori and without oracle knowledge becomes more challenging. As mentioned in the introduction, one way to address this problem is to use the *sparse Bayesian learning* (SBL) approach, which we describe below.

Following (38), we now assume that the recovery of the presumably sparse edge map **g** can be modeled as

$$\mathbf{y} = \Theta\mathbf{g} + \mathbf{n}. \tag{48}$$

Here **y** is a transformation of the acquired Fourier data from (4), $\Theta$ is the forward model described in Sect. 3.4, and **n** is added noise, assumed zero-mean complex Gaussian with unknown variance $\nu^2$. Rather than using the strategy in Algorithm 2, where we find edges by minimizing an explicit $\ell_1$-regularized least squares cost function to enforce sparsity, we now consider the inversion of (48) from a probabilistic perspective. In particular, we use the assumption that the edge function **g** is sparse as a prior. There has been increased interest in Bayesian probabilistic approaches to sparse signal recovery (SSR) problems [4,21,25,34, 38]. There are two categories of Bayesian probabilistic methods for SSR that encompass many well-known recovery algorithms in practice [21]. The first is type-I, or maximum *a posteriori* (MAP) Bayesian estimation which uses a fixed prior. Type-I includes the most popular methods for SSR from compressed sensing including $\ell_1$ regularization [11,33], iteratively reweighted $\ell_1$ regularization [9], and iteratively reweighted $\ell_2$ regularization [10]. On the other hand, Type-II, or evidence maximization Bayesian estimation employs a flexible parametrized prior that is learned from the given data. In addition to the accuracy advantages mentioned in Sect. 1 [21,25], SBL can also provide the added benefit of a full posterior density function, as well as automatic data-driven estimation of hyper-parameters that correspond to the regularization parameter in the MAP method, (41). Below we describe how the SBL approach can be used to improve edge detection given non-uniform Fourier data.

## 4.1 $\ell_1$ Regularization as a Type-I (MAP) Estimate

We begin by examining $\ell_1$ regularization for SSR from a probabilistic perspective as a type-I (MAP) Bayesian estimate. To develop our technique, as in [25], we will assume that the

entries of $\mathbf{y}$ are independent.[4] The likelihood of the data given an edge function, $\mathbf{g}$, and added noise variance, $\nu^2$, can be written as the Gaussian likelihood model

$$p(\mathbf{y}|\mathbf{g}, \nu^2) = (2\pi\nu^2)^{-(2N+1)/2} \exp\left(-\frac{1}{2\nu^2}||\Theta\mathbf{g} - \mathbf{y}||_2^2\right). \tag{49}$$

We now formulate the assumption that the edge function $\mathbf{g}$ is sparse by using a sparsity-encouraging prior on $\mathbf{g}$. One widely used sparsity-encouraging prior is the Laplace density function, given by

$$p(\mathbf{g}|\mu) = \left(\frac{\mu}{2}\right)^{2J+1} \exp\left(-\mu||\mathbf{g}||_1\right). \tag{50}$$

For given $\mu, \nu^2$, using Bayes' theorem yields

$$\mathbf{g}_{MAP} = \arg\max_{\mathbf{g}} p(\mathbf{g}|\mathbf{y}) = \arg\max_{\mathbf{g}} p(\mathbf{y}|\mathbf{g}, \nu^2) p(\mathbf{g}|\mu) \tag{51}$$

$$= \arg\min_{\mathbf{g}} \left\{||\Theta\mathbf{g} - \mathbf{y}||_2^2 + 2\nu^2\mu||\mathbf{g}||_1\right\}, \tag{52}$$

which is equivalent to (41) where the regularization parameter is $\lambda = 2\nu^2\mu$. Thus we see the connection between the typical inversion process via $\ell_1$ regularization as in [8,14] and a MAP approximation to a Bayesian linear regression analysis with a sparsity-encouraging Laplace prior on $\mathbf{g}$, (51). As it is equivalent to the solution proposed in [18], this particular MAP estimate falls victim to the same problems as Algorithm 2.

In general, there are many sparsity-encouraging priors that can be used to regularize in this context. Functions of this type are sometimes referred to as super-Gaussians as they are characterized by fat tails and a sharp peak at zero. However, as will be seen in what follows, using empirical priors characterized by flexible parameters that are estimated explicitly from the data are more often accurate than type-I estimates.

## 4.2 Sparse Bayesian Learning (SBL)

In the previous section we demonstrated that the $\ell_1$-regularized least squares method of [18] was equivalent to that of a MAP estimate using a fixed sparsity-encouraging Laplace prior, which may not yield satisfactory results, especially when noise is present as is evident in Fig. 6. Further, given the probabilistic formulation in Sect. 4.1, we hope to carry Bayesian analysis further and obtain a full posterior distribution on $\mathbf{g}$. This cannot be done using the Laplace prior because it is not conjugate to the Gaussian likelihood model. Conjugate priors [5], would allow us to maintain the same functional form for the prior and posterior while only updating the parameters [3]. Without conjugacy, the associated Bayesian inference cannot be done in closed form [6,20]. Sparse Bayesian learning (SBL), also referred to as the relevance vector machine (RVM) [34], solves this problem by using a hierarchical parametrized prior with similar properties to the Laplace prior. The SBL method derived below closely follows [25,34].

---

[4] This is not generally the case given the definition of $\mathbf{y}$ in (38). However by construction of (14), if $\lambda_k$ is close to $k$, that is, our samples are nearly uniform in the Fourier domain, $\mathbf{B}$ will be diagonally dominant. Hence adopting the probabilistic approach seems reasonable. We also note that it is discussed in [25] that this assumption may not hold. Future investigations will consider the covariance structure in $\mathbf{y}$.

First, define a zero-mean Gaussian prior on each element of $\mathbf{g}$ as

$$p(\mathbf{g}|\mathbf{a}) = \prod_{i=-J}^{J} \mathcal{N}(\mathbf{g}_i|0, \mathbf{a}_i^{-1}) \tag{53}$$

where the hyper-parameter $\mathbf{a}_i$ is the inverse variance of a zero-mean Gaussian density function for each $i = -J, \ldots, J$. This hyper-parameter will be estimated from the data to determine to the spread of this Gaussian and hence the sparsity of $\mathbf{g}$. We consider a non-informative Gamma prior over each element of $\mathbf{a}$, given by

$$p(\mathbf{a}|a, b) = \prod_{i=-J}^{J} \Gamma(\mathbf{a}_i|a, b).$$

We then marginalize over the hyper-parameters $\mathbf{a}$ to obtain the overall prior on $\mathbf{g}$ as

$$p(\mathbf{g}|a, b) = \prod_{i=-J}^{J} \int_0^\infty \mathcal{N}(\mathbf{g}_i|0, \mathbf{a}_i^{-1})\Gamma(\mathbf{a}_i|a, b)d\mathbf{a}_i. \tag{54}$$

Each integral being multiplied in (54) is distributed via the Student's $t$-distribution, which, for appropriate $a$ and $b$, is strongly peaked at $\mathbf{g}_i = 0$. Therefore this prior favors $\mathbf{g}_i$ being zero, and hence is sparsity-encouraging, similar to the Laplace prior. A Gamma prior $\Gamma(\beta|c, d)$ is also introduced on $\beta = \frac{1}{\nu^2}$.

The posterior for the edge function $\mathbf{g}$ can be solved for analytically as a multivariate Gaussian distribution

$$p(\mathbf{g}|\mathbf{y}, \mathbf{a}, \beta) = \mathcal{N}(\mathbf{g}|\mathbf{m}, \Sigma). \tag{55}$$

with mean and covariance matrix given by

$$\mathbf{m} = \beta \Sigma \Theta^T \mathbf{y}, \tag{56}$$

$$\Sigma = (\beta \Theta^T \Theta + \mathbf{A})^{-1}, \tag{57}$$

where $\mathbf{A} = \mathrm{diag}(\mathbf{a})$ [7]. If $\mathbf{n} = 0$, i.e. there is no added noise, we want to let $\nu^2 \to 0$. In [38], the authors derive the following expressions for $\mathbf{m}$ and $\Sigma$ in this case as

$$\mathbf{m} = \mathbf{A}^{-1/2}(\Theta\mathbf{A}^{-1/2})^\dagger \mathbf{y}, \tag{58}$$

$$\Sigma = (\mathbf{I} - \mathbf{A}^{-1/2}(\Theta\mathbf{A}^{-1/2})^\dagger \Theta)\mathbf{A}^{-1}, \tag{59}$$

where $\dagger$ is Moore–Penrose pseudoinverse. We now have the full posterior, so we just need to learn (estimate) hyper-parameters $\mathbf{a}$ and $\beta$ from the given data. By marginalizing over $\mathbf{g}$, the marginal log-likelihood for $\mathbf{a}$ and $\beta$ is

$$\begin{aligned}
\mathcal{L}(\mathbf{a}, \beta) &= \log p(\mathbf{y}|\mathbf{a}, \beta) \\
&= \log \int p(\mathbf{y}|\mathbf{g}, \beta)p(\mathbf{g}|\mathbf{a})d\mathbf{g} \\
&= -\frac{1}{2}\left((2N+1)\log 2\pi + \log|\mathbf{C}| + \mathbf{y}^t\mathbf{C}^{-1}\mathbf{y}\right),
\end{aligned} \tag{60}$$

with $\mathbf{C} = \beta^{-1}\mathbf{I} + \Theta\mathbf{A}^{-1}\Theta^T$ [7]. In [25,34], a type-II maximum likelihood approximation is used which utilizes the point estimates for $\mathbf{a}$ and $\beta$ to maximize $\mathcal{L}$ in (60), and

can be implemented via the *expectation–maximization* (EM) algorithm [13], to obtain the update

$$\mathbf{a}_i^{(\text{new})} = \frac{\gamma_i}{\mathbf{m}_i^2} \tag{61}$$

for $i = -J, \ldots, J$, where $\mathbf{m}_i$ is the $i$th posterior mean weight from (56) and $\gamma_i = 1 - \mathbf{a}_i \Sigma_{ii}$ with $\Sigma$ from (57). For $\beta$, we obtain the update

$$\beta^{(\text{new})} = \frac{2M + 1 - \sum_i \gamma_i}{||\mathbf{y} - \Theta \mathbf{m}||_2^2}. \tag{62}$$

Derivation details for (61) and (62) can be found in Appendix A of [34]. We have that $\mathbf{a}^{(\text{new})}$ and $\beta^{(\text{new})}$ are functions of $\mathbf{m}$ and $\Sigma$, and vise versa, such that the EM algorithm for recovering the posterior for $\mathbf{g}$ iterates between (56) and (57), and (61) and (62) until a convergence criterion is satisfied. Due to the properties of the EM algorithm, SBL is globally convergent, meaning each iteration is guaranteed to reduce the cost function until a fixed point is achieved [38]. In particular, it has been observed that $\mathbf{a}_i \to \infty$ for $\mathbf{g}_i \approx 0$. Also note that since we seek the point estimates of $\mathbf{a}$ and $\beta$, and not their posterior densities, we do not need to rigorously choose parameters $a$, $b$, $c$, $d$ on the Gamma hyper-priors. Therefore we simply set all of them to 0, implying uniform hyper-priors (over a logarithmic scale) on $\mathbf{a}$ and $\beta$ [34]. The resulting algorithm is provided in Algorithm 3.

---

**Algorithm 3** Reconstruction of $[f](x)$ from non-uniform Fourier data via sparse Bayesian learning

---

1: Given $2M + 1$ Fourier coefficients, $\hat{f}(\lambda_k)$, of a piecewise smooth function, $f$, as in (4).
2: Determine $\boldsymbol{\sigma}$ from Algorithm 1.
3: Construct $\Theta$ and $\mathbf{y}$.
4: Initialize hyper-parameters $\mathbf{a}$ and $\beta$, e.g. $\mathbf{a}_i = 1$ for all $i$ or non-negative random initialization.
5: Compute $\mathbf{m}$ and $\Sigma$ via (56) and (57).
6: Update $\mathbf{a}$ and $\beta$ using (61) and (62).
7: Repeat steps 5 and 6 until convergence to a fixed point $\mathbf{a}^*$.
8: The point estimate for $\mathbf{g}$ is the mean computed with $\mathbf{a}^*$, $\mathbf{m}^* = \mathbf{m}(\mathbf{a}^*)$.

---

Figures 7, 8, and 9 show the results using $M = 100$ for Examples 1–3, with jittered, quadratic, and logarithmic sampling, respectively. In addition to the visual comparison, Table 1 below highlights the dramatic improvements over Algorithm 2 in terms of relative error. When Gaussian noise with mean 0 and standard deviation 0.02 is added to the given Fourier data, we retrieve the results shown in Fig. 10. Notice the significant improvement over Fig. 6 both visually in the ease of identifying true jump locations and in terms of relative error which focuses more on accurately estimating the height of the jump. Figure 11
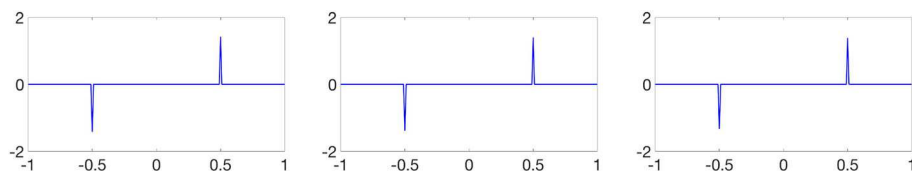


**Fig. 7** $[f_1]$ reconstructed via Algorithm 3 using (left) jittered; (center) quadratic; and (right) logarithmic sampling
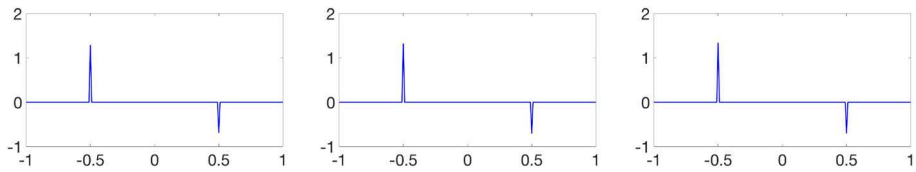
**Fig. 8** $[f_2]$ reconstructed via Algorithm 3 using (left) jittered; (center) quadratic; and (right) logarithmic sampling
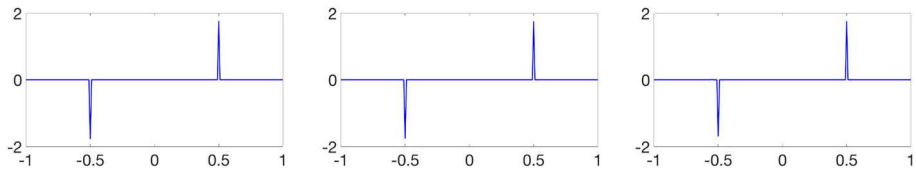


**Fig. 9** $[f_3]$ reconstructed via Algorithm 3 using (left) jittered; (center) quadratic; and (right) logarithmic sampling

**Table 1** Comparison of relative errors from Algorithms 2 and 3 over example functions $f_1$, $f_2$, and $f_3$ using jittered, quadratic, and logarithmic sampling in the noise-free case

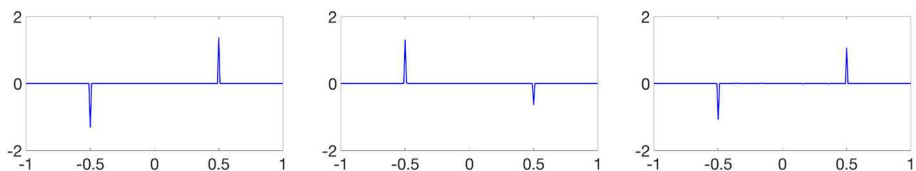|  | Jittered | Quadratic | Logarithmic |
|---|---|---|---|
| $f_1$ with Algorithm 2 | 0.0381 | 0.2475 | 0.3072 |
| $f_1$ with Algorithm 3 | 0.0018 | 0.0229 | 0.0515 |
| $f_2$ with Algorithm 2 | 0.0485 | 0.0856 | 0.2978 |
| $f_2$ with Algorithm 3 | 0.0466 | 0.0277 | 0.0158 |
| $f_3$ with Algorithm 2 | 0.0156 | 0.1513 | 0.2884 |
| $f_3$ with Algorithm 3 | $5.4364 \times 10^{-4}$ | 0.011 | 0.034 |



**Fig. 10** Reconstructions with Algorithm 3 from non-uniform Fourier data with zero-mean Gaussian noise with 0.02 standard deviation. (left) $[f_1]$ from logarithmic sampling; (center) $[f_2]$ from quadratic sampling; and (right) $[f_3]$ from jittered sampling. The relative errors left to right are 0.024, 0.053, and 0.398
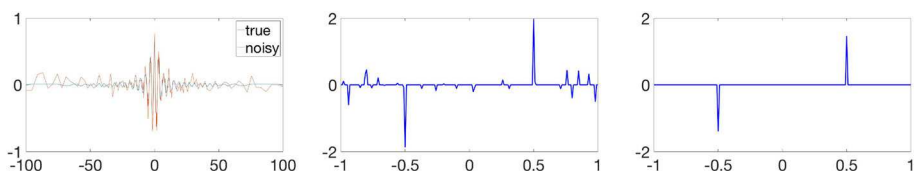


**Fig. 11** Reconstructions of $[f_1]$, given (left) logarithmically sampled non-uniform Fourier data with zero-mean Gaussian noise with 0.08 standard deviation, via (center) Algorithm 2 and (right) Algorithm 3. The relative error in the center is 0.753 and on the right is 0.029
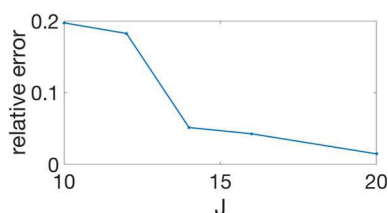
**Fig. 12** Plot of number of grid points $J$ versus relative error. The relative reconstruction error decreases as the number of grid points increases. Each plotted point represents a reconstruction of $[f_3]$ via Algorithm 3 from jittered Fourier data with added zero-mean Gaussian noise with 0.02 standard deviation
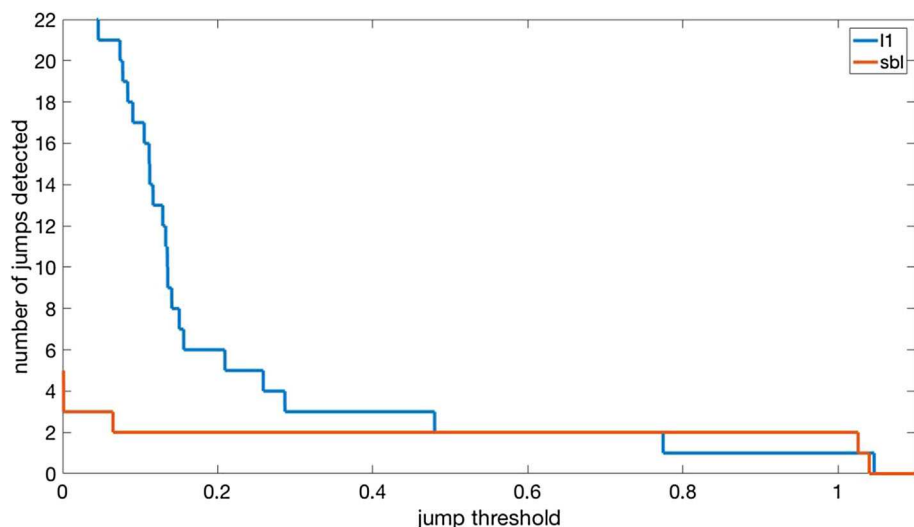


**Fig. 13** Detection comparison of Algorithms 2 and 3 based on number of jumps detected for a given threshold. This plot is derived from the reconstruction of $[f_1]$ from logarithmic sampling with zero-mean Gaussian noise with 0.02 standard deviation

shows another example using even more noise. Finally, Fig. 12 demonstrates that the error depends on the ratio of jumps to grid points in the reconstruction domain. In that example, we see that as the number of grid points increases, the relative reconstruction error decreases as well.

We note that it is possible to further improve these results by non-linearly post-processing the results of Algorithm 3. For example, the algorithm can run multiple times with different concentration factors. Points that differ in sign between the two runs indicate an oscillatory response—not a true edge—and are set to zero. Assuming an appropriate choice for each of the concentration factors, this will only affect falsely identified jumps, further refining the accuracy result in the noisy case.

To summarize the superior performance of Algorithms 2 and 3, we consider a detection scenario where we would like to be able to distinguish true jumps from false jumps that are either artifacts of the edge detection process or misclassified due to noise. In this scenario, a grid point will be classified as a jump if the value at that grid point is above a certain threshold. For a particular example, Fig. 13 shows the number of jumps identified for many thresholds. We see that Algorithm 3 correctly identifies the two correct jumps (we verified they are indeed

at the correct locations) and only those two for a much wider range of threshold values than the $\ell_1$ edge detection method.

### 4.3 Discussion

Our numerical experiments clearly demonstrate that Algorithm 3 yields better accuracy than $\ell_1$ regularization (Algorithm 2) for both the noiseless and noisy cases. The SBL results are better both in terms of damping down false edges as well as more accurately estimating jump height. The Bayesian formulation also provides advantages over other formulations not readily apparent in the figures above. Namely it allows for automatic estimation of model parameters [4]. Choosing parameters in $\ell_1$-regularized formulations is problem dependent, and often by trial and error, sometimes requiring fine tuning. In SBL, the analog of the regularization parameter $\lambda$ as in (41) are the hyper-parameters $\mathbf{a}$ and $\beta$ [38], both of which are estimated directly from the data. This removes user input from the process, which is a common complaint from practitioners regarding $\ell_1$ regularization, since it means that the methods are neither automated nor robust to perturbations in the data. That being said, in very noisy scenarios $\beta$ can still be chosen beforehand instead of estimated, in particular when the point estimate is really the goal. Also not apparent in our numerical experiments is the fact that we have obtained a full density on the edge map $\mathbf{g}$, where only a point estimate is available when using the type-I MAP method. That is, while type-I methods return only a vector estimate of the signal, SBL returns a full multivariate Gaussian density to better quantify the uncertainty. The mean vector of this distribution serves as the de facto point estimate, and the covariance matrix indicates how certain we can be about this estimate. This will be particularly useful in downstream processes where a probability distribution on each gridpoint's value is required. Moreover, because of our accurate estimation of $\Theta$ in (38) with regard to the non-uniform Fourier data acquisition, we are better able to capture different magnitudes of edges, which can be useful in classifying a variety of targets. As was pointed out in [27], determining an edge map directly from the (non-uniform) Fourier data, as opposed to first reconstructing an image (e.g. via non-uniform FFT) and then using an image based edge detection method (such as Canny edge detection), always yields improved fidelity, especially when the data are highly non-uniform or noise is present. This is because important edge information is not lost (filtered) in the image reconstruction process.

SBL is not without its own problems, however. While the iterative algorithm described above has been demonstrated to yield highly accurate sparse solutions [21], with theoretical guarantees [28,38,39], Algorithm 3 requires the inversion of the $(2J+1) \times (2J+1)$ covariance matrix $\Sigma$ in (57), an $\mathcal{O}((2J+1)^3)$ operation, at each step. This can be slow as the signal size grows. To mitigate this problem, a workaround was suggested in [38] using the following simplification:

$$\begin{aligned}
\Sigma &= (\beta \Theta^T \Theta + \mathbf{A})^{-1} \\
&= \mathbf{A}^{-1} - \mathbf{A}^{-1} \Theta^T (\beta^{-1} \mathbf{I} + \Theta \mathbf{A}^{-1} \Theta^T)^{-1} \Theta \mathbf{A}^{-1} \\
&= \mathbf{A}^{-1} - \mathbf{A}^{-1} \Theta^T \mathbf{C}^{-1} \Theta \mathbf{A}^{-1}.
\end{aligned} \tag{63}$$

Now the computation of $\Sigma$ only requires inverting the $(2N+1) \times (2N+1)$ matrix $\mathbf{C}$, reducing the complexity to $\mathcal{O}((2N+1)^3)$, which is much faster for $N < J$. In our experiments this acceleration resulted in a slightly faster solve for Algorithm 3 than for Algorithm 2. Specifically, on average for the experiments shown in Figs. 3, 4, 5 and 6, using CVX [22,23], to solve (41) took 0.65 s. Meanwhile, using the acceleration in (63) with Algorithm 3 for the

analogous Figs. 7, 8, 9 and 10 took on average 0.45 s to converge.[5] Other fast algorithms to consider for optimal implementation include those in [15,25,35].

We also want to address other type-II methods. Another hierarchical parametrized prior-based estimation technique to consider is the so-called type-II $\ell_1$ method [4,21]. Effectively, this prior uses the same hierarchical structure but slightly different update rules than those prescribed in (61) and (62). In [4], empirical evidence supports the idea that using the same number of measurements, type-II $\ell_1$ consistently produced lower reconstruction error than SBL. However, in [21] it was empirically shown that using the same number of measurements, for a signal with the same number of nonzero entries, SBL consistently had a higher probability of successfully reconstructing the signal. Our goal in this current investigation is to develop an algorithm for edge detection using SBL and provide some numerical results. The alternative constructions described here may be effective modifications to our basic algorithm, and will be explored in future work.

## 5 Applications of Edge Detection via SBL

As noted in the introduction, edge detection is not only useful in and of itself, but can be used for downstream processing such as signal recovery and image reconstruction, which can in turn be used for classification, target identification, and change detection. There are two main advantages to employing Algorithm 3 over classical forms of $\ell_1$ regularization. First, due to the construction of $\Theta$ in [18], we achieve better accuracy to the *magnitude* of the edges in the sparse signal recovery. Second, while Algorithm 3 is designed to recover the mean of the solution, it is straightforward to recover the full posterior distribution described in (55)–(57). In what follows we demonstrate how SBL improves image reconstruction. Other applications using our SBL approach will be the subject of future investigations.

Since edges represent sparse features in an image, edge detection methods are often employed to aid in full piecewise-smooth signal reconstruction [1,9,10,12]. For example, the methods in [9,10,12] are designed to apply the regularization more strongly away from the edges. As noted in the introduction, typical $\ell_1$ regularization, which can be considered as a type I MAP estimate, has been enhanced by using iterative reweighting techniques, both for $\ell_1$ and for $\ell_2$ regularization. Such methods are particularly effective when the edges are well separated, that is, when there is sufficient resolution in the acquired data, as well as limited noise. The performance of iterative reweighting schemes deteriorate when this is not the case, mainly because of the propagation and enhancement of false edges caused by the reweighting process. These issues inspired the development of the edge adaptive $\ell_2$ regularization method in [12], which is particularly appealing because it overcomes the magnitude dependence of $\ell_1$ norm regularization by directly identifying edges in one initial step. The procedure begins by creating a binary weighting matrix, or mask, based on the edge locations found using an (any) edge detection algorithm. A weighted $\ell_2$-regularized optimization problem is then solved to achieve the full signal reconstruction. Since the vector inside the regularization term should now be identically zero instead of sparse, it is appropriate and computationally advantageous to regularize using the $\ell_2$ norm. The effect is similar to that of skipping to the final iteration of an iteratively reweighted $\ell_2$ scheme [10]. The full procedure is described in Algorithm 4. In [12], this method was demonstrated to have potential use in SAR image reconstruction. Here we simply demonstrate its compatibility and effectiveness with this SBL edge detection
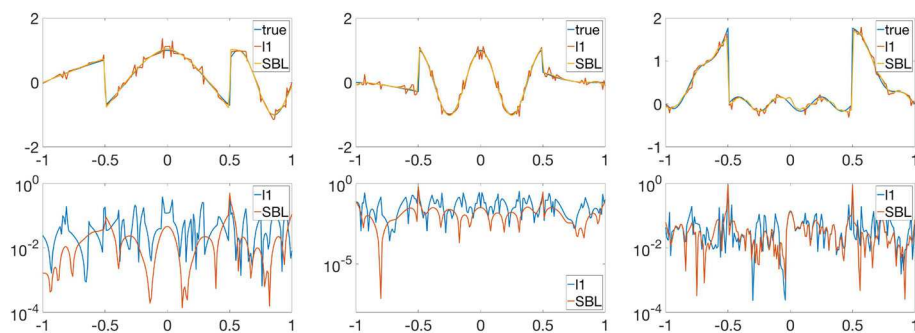
---

**Fig. 14** (top row) Noisy signal reconstructions via Algorithm 4 comparing $\ell_1$ and SBL edge detection for development of the mask with (bottom row) log error for (left) $f_1$ from logarithmic sampling with $m = 2$ with relative error 0.1496 and 0.0823; (center) $f_2$ from quadratic sampling and $m = 3$ with relative error 0.1912 and 0.1015; and (right) $f_3$ from jittered sampling and $m = 1$ with relative error 0.1683 and 0.1527

method starting from non-uniform Fourier data on piecewise smooth functions $f_1$, $f_2$, and $f_3$ given in Examples 1–3 respectively. Figure 14 shows full signal reconstructions of these examples using a variety of sampling patterns when zero-mean Gaussian noise with standard deviation 0.02 is added to the given Fourier data. We use the polynomial annihilation method [1,2], of various orders to ensure sparsity of the corresponding jump function $[f]$. Other operators, e.g. wavelets, may also be used.

Note that in Algorithm 4 we introduce the parameter $\tau$, but in the experiments shown simply set $\tau = \frac{1}{2J+1}$, which is consistent with the grid resolution. Figure 13 also shows that our method performs well for a broad range of $\tau$.

---

**Algorithm 4** Edge-adaptive $\ell_2$-regularized image reconstruction

---

1: Reconstruct the jump function, $[f]$, using Algorithm 2 or 3 as $\mathbf{g}^*$.
2: For each index $j = -J, \ldots, J$ such that $|\mathbf{L}^m \mathbf{g}_j^*| > \tau$, set $\mathbf{y}_j = 0$. Else, $\mathbf{y}_j = 1$. Here $\tau$ is a user-defined thresholding parameter. The mask is $\mathbf{M} = \text{diag}(\mathbf{y})$, and $\mathbf{L}^m$ is the $m$th order polynomial annihilation operator [2], similar to a high order total variation operator. It is necessary to apply $\mathbf{L}^m$ so that the stencil of the sparsity operator is accounted for in the mask.
3: The edge-adaptive $\ell_2$ regularization image reconstruction is the solution to the optimization problem,

$$\mathbf{f}^* = \arg\min_{\mathbf{f}} \left\{ ||\mathbf{F}\mathbf{f} - \hat{\mathbf{f}}||_2^2 + \lambda ||\mathbf{M}\mathbf{L}^m\mathbf{f}||_2^2 \right\}, \tag{64}$$

where $\mathbf{F}$ is the non-uniform Fourier measurement matrix, $\hat{\mathbf{f}}$ is the vector of collected Fourier coefficients, and $\lambda > 0$ is the regularization parameter.

---

## 6 Conclusion

The SBL edge detection method given non-uniform Fourier data developed here compares favorably to typical $\ell_1$-regularized sparse signal recovery methods. This is true even when the data are explicitly considered in the recovery method, as in [18]. Moreover, this type-II approach is more robust than typical type-I MAP estimates, which can be viewed as standard $\ell_1$ regularization methods, since the regularization parameters are estimated directly

from the data and modified accordingly. The lack of automation in standard $\ell_1$ regularization methods is considered highly unfavorable by most practitioners. Our method can also be used to reconstruct a piecewise smooth signal as a pre-processing step for the edge-adaptive reconstruction [12]. Our numerical experiments demonstrate that both the edge detection and resulting reconstruction are robust with respect to noise and signal type, with our method consistently outperforming the one in [18]. Future investigations will include further development of Algorithms 2 and 3 for two-dimensional signals. While theoretically the one-dimensional acceleration in (63) combined with our method expands to multiple dimensions in a dimension-by-dimension manner, our early explorations indicate it is still cost prohibitive for detecting edges in images. Some of the ideas discussed in Sect. 4.3 will be used to improve the efficiency of Algorithm 3, which will be important for higher dimensional reconstructions and larger signals in general.

# References

1. Archibald, R., Gelb, A., Platte, R.B.: Image reconstruction from undersampled Fourier data using the polynomial annihilation transform. J. Sci. Comput. **67**(2), 432–452 (2016)
2. Archibald, R., Gelb, A., Yoon, J.: Polynomial fitting for edge detection in irregularly sampled signals and images. SIAM J. Numer. Anal. **43**(1), 259–279 (2005)
3. Babacan, S.D., Molina, R., Katsaggelos, A.K.: Parameter estimation in TV image restoration using variational distribution approximation. IEEE Trans. Image Process. **17**(3), 326–339 (2008)
4. Babacan, S.D., Molina, R., Katsaggelos, A.K.: Bayesian compressive sensing using laplace priors. IEEE Trans. Image Process. **19**(1), 53–63 (2010)
5. Berger, J.O.: Statistical Decision Theory and Bayesian Analysis. Springer, Berlin (2013)
6. Bernardo, J.M., Smith, A.F.: Bayesian theory (2001)
7. Bishop, C.: Pattern Recognition and Machine Learning. Springer, Berlin (2006)
8. Candès, E.J., Romberg, J., Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory **52**(2), 489–509 (2006)
9. Candes, E.J., Wakin, M.B., Boyd, S.P.: Enhancing sparsity by reweighted $\ell_1$ minimization. J. Fourier Anal. Appl. **14**(5), 877–905 (2008)
10. Chartrand, R., Yin, W.: Iteratively reweighted algorithms for compressive sensing. In: IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008, pp. 3869–3872. IEEE, New York (2008)
11. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM Rev. **43**(1), 129–159 (2001)
12. Churchill, V., Archibald, R., Gelb, A.: Edge-adaptive $\ell_2$ regularization image reconstruction from nonuniform Fourier data. Preprint. arXiv:1811.08487 (2018)
13. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Ser. B (Methodol.) **39**, 1–38 (1977)
14. Donoho, D.L.: Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006)
15. Faul, A.C., Tipping, M.E.: Analysis of sparse Bayesian learning. In: Advances in Neural Information Processing Systems, pp. 383–389 (2002)
16. Gelb, A., Hines, T.: Detection of edges from nonuniform Fourier data. J. Fourier Anal. Appl. **17**(6), 1152–1179 (2011)
17. Gelb, A., Song, G.: A frame theoretic approach to the nonuniform fast Fourier transform. SIAM J. Numer. Anal. **52**(3), 1222–1242 (2014)
18. Gelb, A., Song, G.: Detecting edges from non-uniform Fourier data using Fourier frames. J. Sci. Comput. **71**(2), 737–758 (2017)
19. Gelb, A., Tadmor, E.: Detection of edges in spectral data. Appl. Comput. Harm. Anal. **7**(1), 101–135 (1999)
20. Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., Rubin, D.B.: Bayesian Data Analysis, vol. 2. CRC Press, Boca Raton (2014)
21. Giri, R., Rao, B.D.: Type I and type II Bayesian methods for sparse signal recovery using scale mixtures. IEEE Trans. Signal Process. **64**(13), 3418–3428 (2016)

22. Grant, M., Boyd, S.: Graph implementations for nonsmooth convex programs. In: Blondel, V., Boyd, S., Kimura, H. (eds.) Recent Advances in Learning and Control, Lecture Notes in Control and Information Sciences, pp. 95–110. Springer, Berlin (2008). http://stanford.edu/~boyd/graph_dcp.html
23. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx (2014)
24. Guo, W., Yin, W.: Edge guided reconstruction for compressive imaging. SIAM J. Imaging Sci. **5**(3), 809–834 (2012)
25. Ji, S., Xue, Y., Carin, L.: Bayesian compressive sensing. IEEE Trans. Signal Process. **56**(6), 2346–2356 (2008)
26. Jiménez-Alaniz, J.R., Medina-Bañuelos, V., Yáñez-Suárez, O.: Data-driven brain MRI segmentation supported on edge confidence and a priori tissue information. IEEE Trans. Med. Imaging **25**(1), 74–83 (2006)
27. Martinez, A., Gelb, A., Gutierrez, A.: Edge detection from non-uniform Fourier data using the convolutional gridding algorithm. J. Sci. Comput. **61**(3), 490–512 (2014)
28. Rao, B.D., Wipf, D.P.: Comparing the effects of different weight distributions on finding sparse representations. In: Advances in Neural Information Processing Systems, pp. 1521–1528 (2006)
29. Shattuck, D.W., Sandor-Leahy, S.R., Schaper, K.A., Rottenberg, D.A., Leahy, R.M.: Magnetic resonance image tissue classification using a partial volume model. NeuroImage **13**(5), 856–876 (2001)
30. Song, G., Gelb, A.: Approximating the inverse frame operator from localized frames. Appl. Comput. Harm. Anal. **35**(1), 94–110 (2013)
31. Stefan, W., Viswanathan, A., Gelb, A., Renaut, R.: Sparsity enforcing edge detection method for blurred and noisy Fourier data. J. Sci. Comput. **50**(3), 536–556 (2012)
32. Tadmor, E., Zou, J.: Three novel edge detection methods for incomplete and noisy spectral data. J. Fourier Anal. Appl. **14**(5–6), 744 (2008)
33. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B (Methodol.) **58**, 267–288 (1996)
34. Tipping, M.E.: Sparse Bayesian learning and the relevance vector machine. J. Mach. Learn. Res. **1**(Jun), 211–244 (2001)
35. Tipping, M.E., Faul, A.C., et al.: Fast marginal likelihood maximisation for sparse Bayesian models. In: AISTATS (2003)
36. Viswanathan, A., Gelb, A., Cochran, D.: Iterative design of concentration factors for jump dection. J. Sci. Comput. **51**, 631–649 (2012)
37. Viswanathan, A., Gelb, A., Cochran, D.: Iterative design of concentration factors for jump detection. J. Sci. Comput. **51**(3), 631–649 (2012)
38. Wipf, D.P., Rao, B.D.: Sparse Bayesian learning for basis selection. IEEE Tran. Signal Process. **52**(8), 2153–2164 (2004)
39. Wipf, D.P., Rao, B.D.: $\ell_0$-norm minimization for basis selection. In: Advances in Neural Information Processing Systems, pp. 1513–1520 (2005)