Probabilistic Matching Pursuit with Gabor Dictionaries

S. E. Ferrando [1]

Department of Mathematics, Physics and Computer Science, Ryerson Polytechnic University, Toronto, Ontario M5B 2K3, Canada

E. J. Doolittle

Department of Mathematics, University of Toronto, Toronto, Ontario M5S 3G3, Canada

A. J. Bernal, L. J. Bernal

Department of Physics, Mar del Plata National University, Funes 3350, Mar del Plata 7600, Argentina

Abstract

We propose a probabilistic extension of the matching pursuit adaptive signal processing algorithm introduced by Mallat and others. In adaptive signal processing, signals are expanded in terms of a large linearly dependent "dictionary" of functions rather than in terms of an orthonormal basis. Matching pursuit is a simple greedy algorithm for generating an expansion of a given signal. In probabilistic matching pursuit multiple random expansions are obtained as estimates for a given signal. The new algorithm is illustrated in the context of signal denoising. Although most of the random expansions generated by probabilistic matching pursuit are poorer estimates for the signal than those obtained by matching pursuit, our final estimate, obtained as an expected value computed by means of an ergodic average, can improve the result obtained by MP in some denoising situations. One of the major underlying ideas is a novel notion of coherence between a signal and the dictionary. Several simulated examples are presented.

Key words: Matching pursuit, Gabor function dictionary, denoising, rejection sampling, Bernoulli shift.

[1] The research of S. E. Ferrando was supported by NSERC grant.

0 Notation

The following standard notation will be used: $\mathbb{N} = \{1, 2, 3, \dots\}$ is the set of natural numbers, $\mathbb{R}$ is the set of real numbers, and $\mathbb{C}$ is the set of complex numbers. $\mathbb{R}^+ = (0, \infty)$ is the set of positive real numbers.

Given a set S, we denote the countable Cartesian product of S with itself by $S^{\mathbb{N}}$ and we denote the cardinality of S by $|S|$.

We use the following standard functions: Given a real number x, $\lfloor x \rfloor$ represents the floor of x, i.e., the greatest integer less than or equal to x. We denote the normal distribution with mean μ and standard deviation σ by $\mathcal{N}(\mu, \sigma)$.

The symbol $\propto$ is used to indicate that two density functions are equal up to a normalization constant.

We use the following notation from linear algebra: Given a vector v, we denote its components by $[v]_i$ or v_i. An inner product between two members f and g of a vector space is denoted by $\langle f, g \rangle$.

Given a probability space (Ω, μ), the support of μ is the largest measurable subset $A \subseteq \Omega$ such that $\mu(A) = 1$.

Given a noise signal $f = z + \sigma w$, the *signal to noise ratio* is defined as the ratio of signal energy to noise energy:

$$\text{SNR} = \frac{\|z\|^2}{\|\sigma w\|^2}.$$

For the simulated examples, f_i, $i = 1, \dots, 5$ are signals defined in Section 4.4 and each $f_{i,j}$, $i = 1, \dots, 5$, $j = 1, \dots, 4$ represent the signal f_i with added noise of "level" j. $f_{i,j}^M$ and $f_{i,j}^P$ are the reconstructions of f_i by matching pursuit and probabilistic matching pursuit respectively.

1 Introduction

Probabilistic matching pursuit is a randomization of the matching pursuit algorithm for signal processing introduced by Mallat and others, which in turn is an adaptive alternative to traditional signal processing methods. We begin with a brief overview of the chain of ideas leading from traditional signal processing to probabilistic matching pursuit, first in the context of signal compression, and then in the context of denoising. We then provide a description of the main idea of our paper. We close the introduction with a detailed overview of the rest of the paper.

1.1 Signal Compression with Adaptive Expansions

Signal compression is generally performed by expanding a given signal in a series and then throwing away terms of the series which may be neglected. The traditional approach uses an orthonormal basis such as a Fourier or wavelet basis to develop unique series expansions. In order to further improve signal compression it is natural to search for adaptive decompositions in which series are expanded in terms of a *linearly dependent* family of unit vectors (here called a *dictionary*) which is in general much larger than a basis. In this case expansions are no longer unique, so they may be adapted in a manner which depends on the given signal [19,14]. Such adaptive expansions will fit the underlying signal better, but there is the new difficulty of determining which of the many different expansions to use.

In the case of compression, we wish to get the best approximation by using as few terms as possible. Therefore, the basic optimization problem we must solve is the following: given a dictionary $\mathcal{D} = \{g_\gamma : \gamma \in \Gamma\}$, an integer M and a vector f in the space spanned by $\mathcal{D}$, find a sub-collection $I \subseteq \Gamma$ with cardinality M and numbers β_γ such that $\|f - \sum_{\gamma \in I} \beta_\gamma g_\gamma\|$ is minimized. Given the generality of the collection $\mathcal{D}$, globally optimizing the search of vectors in the expansion is an NP-complete search problem [6]. Various efficient suboptimal solutions to this optimization problem have been proposed. The *best wavelet basis* algorithm [22] gives optimal solutions for special collections of functions. The *basis pursuit* algorithm [5] poses the problem as a linear optimization problem. The *matching pursuit* (MP) algorithm, introduced in [6] and [13] and described below, is perhaps the simplest general-purpose adaptive signal processing algorithm. Numerical comparisons for those three approaches are presented in [5] and [14].

In matching pursuit the single dictionary element which best matches the signal is removed from the signal, and the process is repeated with the signal

residue from the previous step until a *stopping rule* is satisfied. If by "best dictionary element" we mean the one with maximum inner product with the signal, the residue obtained at each step has squared norm as small as possible for that step. An algorithm that operates in this way with minimal "look ahead" is known as a *greedy algorithm*. The greedy MP algorithm is simple, fast, and general, with many interesting applications [12,15,18]. Its generality is due to the fact that it requires minimal assumptions on the dictionary vectors, which must only belong to a Hilbert space. Rates of convergence and other mathematical questions related to MP are investigated in [7] and [13].

However, the fact that multiple expansions are possible with redundant dictionaries gives us the opportunity of selecting expansions at random from many different possibilities, an option which is not available in the case of orthonormal bases. Random selection of expansions offers the following advantages over deterministic selection:

- Random expansions may be generated more quickly than deterministic expansions.
- Random expansions do not force us to restrict searches to a discrete subset (a search grid) of a dictionary.
- The availability of multiple random expansions permits us to improve results by averaging.
- The probability distributions employed in random selection are far more flexible than deterministic selection procedures, and may be adjusted to take into account empirical observations or additional knowledge in special-case situations.

Furthermore, the deterministic choice of expansions is a special case of probabilistic choice, which means that carefully chosen probabilistic expansions should be at least as good as deterministic expansions.

The purpose of this paper is to present a probabilistic extension of the MP algorithm which offers some advantages over MP. We have modified matching pursuit to select dictionary elements at random from a list of promising candidates (those above a certain threshold inner product). Such selection rule gives an algorithm which is easy to understand and analyze relative to matching pursuit. This new algorithm was designed as a denoising algorithm and is poor as a compression tool.

1.2 Signal Denoising with Adaptive Expansions

Signal denoising is generally performed in a manner similar to that of compression, by expanding a given signal in a series and throwing away the terms that "look like" noise. The correct selection of stopping rule is critical in denoising:

if we terminate our series too early we may miss crucial signal features, and if we terminate our series too late we will include noise in our estimate of the signal.

For example, signals embedded in Gaussian noise are well estimated by non-linear shrinkage of wavelet coefficients from an orthonormal wavelet basis [1]. It has been established in [10] that this type of denoising outperforms traditional estimators. The heuristic behind this approach relies on the fact that all empirical wavelet coefficients contribute noise proportional to the variance but only a few wavelet coefficients contribute to the signal. Therefore it is the property of wavelet basis to compress the signal information (few coefficients are needed to reproduce a large percentage of the energy of the signal) and the ability to recognize noise in the wavelet domain that leads to a powerful technique for denoising.

Adaptive approximations necessarily offer a better compression of a signal than expansions in terms of an orthonormal basis. However, in the presence of noise, estimations by thresholding may not be improved by an adaptive expansion because the flexibility of the search may result in a dictionary element that correlates well with the noise. A practical solution to this problem is to define the noisiness of a signal *relative* to the given dictionary. In the context of an orthonormal basis this is known as *coherent basis thresholding* (see [14, page 465]). In the MP context this idea implies a practical stopping rule, namely comparing the normalized inner products of the residues with the averages of the normalized inner products of noise. Then, given the iterative nature of MP, denoising is performed by stopping the search for new components once the next residue of the algorithm is recognized as noise. The denoised function is then the expansion of the original noisy function in terms of the components found (which are called *coherent* components). See Section 2.2 for details.

1.3 Probabilistic Matching Pursuit

We now describe the main idea of our paper. Let $f_i = z_i + \sigma w_i$ where the vector z is a given signal and w_i are a finite set of samples from an i.i.d. sequence of random variables W_i with distribution $\mathcal{N}(0,1)$. Our intention is to take advantage of the many representations for z available in the redundant dictionary $\mathcal{D}$. Usually z is approximated by a single (coherent) vector c, but we will introduce a probability space (Ω, μ), the elements of which are denoted by x, and a family of functionals C_i on this probability space. The denoised approximation to each component z_i will be given by the expected value

$$\mathbf{E}_\mu(C_i) = \int C_i(x)\, d\mu(x).$$

The measure μ will be supported in a subset of vectors of $\mathcal{D}$ that resemble characteristics of the signal and hence offer the opportunity to reinforce a good reconstruction. The main ingredient in our approach is a randomization of the set of labels of the dictionary which gives a probabilistic way to distinguish between coherent structure and noise. These notions are relative to the given dictionary. The construction of (Ω, μ) is done is Section 3.

Next, we give an argument to explain why an expected value offers a better reconstruction in certain situations. The reconstruction error is measured in terms of the relative mean squared error

$$\mathrm{RMSE}(z) = \frac{\|z - \mathbf{E}_\mu(C)\|}{\|z\|}$$

where $\|\cdot\|$ is the norm induced by the given inner product. From convexity of the norm functional and Jensen's ([16]) inequality we see that

$$\frac{\|z - \mathbf{E}_\mu(C)\|}{\|z\|} = \left\| \mathbf{E}_\mu \left(\frac{(z - C)}{\|z\|} \right) \right\| \leq \mathbf{E}_\mu \left(\frac{\|z - C\|}{\|z\|} \right). \tag{1}$$

When the MP expansion is not optimal (see the discussion in Section 4) many good quality expansions are available. These expansions are included in the support of μ and the average of their RMSE appear in the right hand side of (1). In practical situations the left hand side of (1) is considerably smaller than the right hand side because the norm functional is strictly convex and the samples tend to surround the correct value with similar errors, so we are averaging approximations which lie roughly on a sphere surrounding the correct value. This improvement will be demonstrated with numerical examples throughout the paper. We report improved performance *relative* to the performance of MP denoising; comparison with respect to other denoising techniques is outside the scope of the paper. However, note that adaptive denoising improves over denoising with an orthonormal basis (which in turn improves over traditional estimators) if the signal to be analyzed can be more efficiently compressed in the dictionary that in the given orthonormal basis (see the discussion in [14, page 464]).

1.4 Organization of the Paper

The matching pursuit algorithm with Gabor dictionaries as discussed in [13] and [14] is reviewed in Section 2, where we also indicate the specific Hilbert spaces which will be used in our implementations, present important definitions, and introduce denoising using the MP algorithm. We introduce our probabilistic extension of MP and comment on its main features in Section 3. We describe the computational details of PMP and analyze its computational

cost in Sections 4.1 and 4.2. We discuss potential improvements of the new method over the MP method for denoising and study performance in several simulated denoising tasks in Sections 4.3 and 4.4. Finally, we draw conclusions on the degree of success of our method and its future prospects in Section 5.

The appendices provide technical details on three aspects of the implementation of our algorithm. The rejection method for sampling is presented in Appendix A, and two methods for speeding up the algorithm, the Bernoulli shift and fast formulas for the inner products of Gabor functions, are discussed in Appendices B and C.

2 The MP Algorithm

In this section we review the essential aspects of the matching pursuit algorithm as discussed in [13]. Let $\mathbf{H}$ be a Hilbert space, we define a dictionary as a family $\mathcal{D} = \{g_\gamma : \gamma \in \Gamma\}$ of vectors in $\mathbf{H}$ such that $\|g_\gamma\| = 1$. Let $\mathbf{V}$ be the closed linear span of the dictionary vectors. We say that the dictionary is complete if $\mathbf{V} = \mathbf{H}$. MP approximates f by orthogonal projections on elements of $\mathcal{D}$; i.e., given $g_{\gamma_0} \in \mathcal{D}$ the vector f can be written as

$$f = \langle f, g_{\gamma_0} \rangle g_{\gamma_0} + Rf \tag{2}$$

where Rf is the residual vector left after approximating f in the direction of g_{γ_0}. Clearly g_{γ_0} is orthogonal to Rf, so

$$\|f\|^2 = |\langle f, g_{\gamma_0} \rangle|^2 + \|Rf\|^2. \tag{3}$$

To minimize $\|Rf\|$ we must maximize $|\langle f, g_{\gamma_0} \rangle|$ over $g_{\gamma_0} \in \mathcal{D}$. In general, it is only computationally feasible to find an "almost optimal" vector g_{γ_0} in the sense that

$$|\langle f, g_{\gamma_0} \rangle| = \max_{\gamma \in \Gamma_\alpha} |\langle f, g_\gamma \rangle| \geq \alpha \sup_{\gamma \in \Gamma} |\langle f, g_\gamma \rangle|, \tag{4}$$

where $\Gamma_\alpha \subseteq \Gamma$ and α is an optimality factor which satisfies $0 < \alpha \leq 1$. The construction of Γ_α depends on the dictionary; typically, if the dictionary is indexed by a set of continuous parameters Γ, then Γ_α will be a discrete grid of some sort in Γ. For details we refer to Section 2.1 and [13].

We then continue the matching pursuit by induction. Let $R^0 f = f$. Suppose that we have computed $R^n f$, the residue of order n, for some $n \geq 0$. We then choose an element $g_{\gamma_n} \in \mathcal{D}_\alpha = \{g_\gamma : \gamma \in \Gamma_\alpha\}$ which closely matches the residue $R^n f$:

$$|\langle R^n f, g_{\gamma_n} \rangle| \geq \alpha \sup_{\gamma \in \Gamma} |\langle R^n f, g_\gamma \rangle|. \tag{5}$$

The residue $R^n f$ is decomposed as

$$R^n f = \langle R^n f, g_{\gamma_n} \rangle g_{\gamma_n} + R^{n+1} f \tag{6}$$

which defines $R^{n+1} f$, the residue of order $n+1$. Let us repeat this decomposition m times. Writing f in terms of the residues $R^n f$, $n = 0, 1, \ldots, m$ and applying (6) yields

$$f = \sum_{n=0}^{m-1} \langle R^n f, g_{\gamma_n} \rangle g_{\gamma_n} + R^m f. \tag{7}$$

The following theorem is fundamental to the MP algorithm [13].

Theorem 1 *If $\mathcal{D}$ is a complete dictionary and if $f \in \mathbf{H}$ then*

$$f = \sum_{k=0}^{\infty} \left\langle R^k f, g_{\gamma_k} \right\rangle g_{\gamma_k} \tag{8}$$

and

$$\|f\|^2 = \sum_{k=0}^{\infty} \left| \left\langle R^k f, g_{\gamma_k} \right\rangle \right|^2. \tag{9}$$

2.1 Gabor Dictionaries

Gabor functions are "windowed" trigonometric functions with infinite exponentially decreasing tails. It is useful to consider two kinds of Gabor functions: functions which are either continuous (defined on $\mathbb{R}$) or discrete (defined on the discrete subset $\mathbf{S}$ of $\mathbb{R}$).

As the window function $g(t)$ we use the normalized Gaussian given by

$$g(t) = 2^{1/4} e^{-\pi t^2}. \tag{10}$$

For any $\gamma = (s, u, \xi) \in \mathbb{R}^+ \times \mathbb{R}^2 = \Gamma$, let the Gabor function g_γ be given by

$$g_\gamma(t) = \frac{1}{\sqrt{s}} g\left(\frac{t-u}{s}\right) e^{i\xi t}. \tag{11}$$

The factor $1/\sqrt{s}$ normalizes $g_\gamma(t)$. Here $s > 0$ is called the *scale* of the function, u its *translation* and ξ its *frequency modulation*. $g_\gamma(t)$ is centered at the abscissa u and its energy is mostly concentrated in a neighborhood of u of size proportional to s. When $f(t)$ is real we use dictionaries of real time-frequency functions. For any $\gamma = (s, u, \xi)$ any phase $\phi \in [0, 2\pi)$, define

$$g_{(\gamma,\phi)}(t) = \frac{K_{(\gamma,\phi)}}{\sqrt{s}} g\left(\frac{t-u}{s}\right) \cos(\xi t + \phi) \tag{12}$$

where the positive constant $K_{(\gamma,\phi)}$ is determined by the condition $\|g_{(\gamma,\phi)}\| = 1$. For convenience we use the notation $\beta = (\gamma, \phi)$ and $K_\beta = K_{(\gamma,\phi)}$.

We now define the discrete Gabor functions which are used in the formalism of Section 3 and in the software implementation used for the numerical experiments. In the discrete case f is assumed to be a signal $f(t)$ supported on a discrete set $\mathbf{S} = \{t_i\}$, $i = 1, \ldots, |\mathbf{S}|$ where $|\mathbf{S}|$ may be finite or infinite. We equip $\mathbf{S}$ with the Dirac discrete measure and we will consider the space $L^2(\mathbf{S})$. If $|\mathbf{S}| = N$ we use the notation $f_i = [f]_i = f(t_i)$, $f = (f_1, \ldots, f_N)$. The Gabor functions are discretized (see (**??**) in Appendix C) and considered as elements of $L^2(S)$. Inner products are given by

$$\langle f, g \rangle = \sum_{i \in \mathbf{S}} f(t_i)\, \bar{g}_\beta(t_i). \tag{13}$$

In Appendix C equations are presented which are only valid when the points in $\mathbf{S}$ are uniformly spaced and $|\mathbf{S}|$ equals infinity. To make use of these formulas in practice, f is assumed to be zero outside a given interval. The dictionary of real time-frequency vectors is defined by $\mathcal{D}_R = \{g_{(\gamma,\phi)} : (\gamma, \phi) \in \Lambda = \Gamma \times [0, 2\pi)\}$. Matching pursuit performed with this dictionary decomposes any real signal $f(t)$ into the sum

$$f(t) = \sum_{n=0}^{\infty} \langle R^n f, g_{\beta_n} \rangle g_{\beta_n}(t) \tag{14}$$

where the indices $\beta_n = (s_n, u_n, \xi_n, \phi_n)$ are chosen by maximizing $|\langle R^n f, g_{\beta_n} \rangle|$ over Λ. In practice this maximization is not feasible and an approximation scheme as indicated in equations (4) and (5) has to be used. Define the discretized complex dictionary by $\mathcal{D}_\alpha = \{g_\gamma : \gamma \in \Gamma_\alpha\}$, a subset of the complex Gabor dictionary where the index set Γ_α is composed of all $\gamma = (a^j, pa^j \Delta u, ka^{-j} \Delta\xi)$, with $a = 2$, $\Delta u = \frac{1}{2}$, $\Delta\xi = \pi$, $0 < j < \log_2 N$, $0 \le p < N 2^{-j+1}$ and $0 \le k < 2^{j+1}$. In [13] it is proven (in the continuous case) that if the parameters (s, u, ξ) are discretized in this way there exists an $\alpha > 0$ such that the MP algorithm is sub-optimal with respect to α, i.e., (5) holds. The reader is referred to [13] for a thorough presentation of this discretization. It is numerically convenient to perform most of the computations with complex Gabor vectors. We will work only with the three parameters $\gamma = (s, u, \xi)$; explicit use of ϕ can be avoided by making use of the following definition: given $h \in \mathbf{H}$, $\gamma = (s, u, \xi)$ and $\langle h, g_\gamma \rangle = a + ib$ and $\tan\phi = b/a$ we set $\beta = (s, u, \xi, \phi)$. Therefore we have

$$\langle h, g_\beta \rangle = K_{(s,u,\xi,\phi)} \langle h, g_\gamma \rangle; \tag{15}$$

Hence, we choose β_n in (14) in such a way that the following holds:

$$|\langle R^n f, g_{\beta_n} \rangle| = \max_{\gamma \in \Gamma_\alpha} \left(K_{(s,u,\xi,\psi)} |\langle R^n f, g_\gamma \rangle| \right) \tag{16}$$

where $\tan(\psi) = b/a$ and $\langle R^n f, g_\gamma \rangle = a + ib$.

2.2 MP Denoising

In this section we follow [13]. Let f be a vector in a finite dimensional Hilbert space $\mathbf{H}$. Let us denote

$$\lambda_n(R^n f) = \frac{|\langle R^n f, g_{\beta_n} \rangle|}{\|R^n f\|}. \tag{17}$$

Note that $\lambda_n(h)$ only depends upon the position of $\frac{h}{\|h\|}$ on the unit sphere of the space $\mathbf{H}$. Let W be a discrete Gaussian white noise. For any $n \geq 0$, the average value of $\lambda_n(R^n W)$ measured with a uniform probability distribution over the unit sphere, is equal to the expected value $\mathbf{E}(\lambda_n(R^n W))$. Indeed, after normalization, the realizations of a discrete Gaussian white noise have a uniform probability distribution over the unit sphere of $\mathbf{H}$. We define the *coherent structures* of a signal f as the first m vectors $(g_{\beta_n})_{0 \leq n < m}$ that have a higher than average correlation with $R^n f$. In other words, f has m coherent structures if and only if for $0 \leq n < m$

$$\lambda_n(R^n f) > \mathbf{E}(\lambda_n(R^n W)), \tag{18}$$

and

$$\lambda_m(R^m f) \leq \mathbf{E}(\lambda_m(R^m W)). \tag{19}$$

Empirical evidence that this is a well defined rule (i.e. that (19) actually holds for a finite m) is given in [6]. To summarize: MP denoising is performed by stopping the algorithm when the correlations between the residuals and dictionary elements are comparable to the average noise correlations.

The problem of optimally stopping the algorithm is fundamental. The above rule is an extrapolation of coherent basis thresholding on orthonormal basis ([14, page 465]) to general dictionaries. This stopping rule is useful in practical situations but it is not optimal. At present, to the best of our knowledge, there are no results in the theory of adaptive representations that can be used to support a better choice of stopping rule (or choice of thresholding in other adaptive algorithms). Our probabilistic version of the MP uses a probabilistic generalization of the above rule and will be discussed in Section 3.

3 Probabilistic MP Denoising

In this section a probabilistic extension of MP is introduced. Notation and definitions are taken from Section 2.1. The goal is to randomize the parameters β in order to have a probabilistic extension of the notion of coherence introduced in Section 2.2. This is achieved by means of the notion of *probabilistic coherent structure*, (28). The fact that MP is a recursive algorithm is reflected by defining a measure in an infinite product space through *conditional probabilities*. In practice the number of components is finite due to the presence of a stopping time.

3.1 Main Definitions

For technical reasons we consider the components of the γ parameters to be limited to bounded intervals of the real line and the space $\mathbf{H}$ will be finite dimensional . We denote these intervals by I_i, $i = 1, 2, 3$, and $I = I_1 \times I_2 \times I_3$. Explicitly the intervals are $I_1 = [\delta, b-a]$, $I_2 = [a, b]$, $I_3 = [0, c]$, where $a < b$ are real numbers and δ and c are positive real numbers. In applications δ will be the smallest resolution in scale, $[a, b]$ will be the support of the sampled signal and c the maximum expected frequency. Given that the number of coherent components in one expansion is unbounded, the correct space for our approach is $I^{\mathbb{N}}$. Hence a point $x \in I^{\mathbb{N}}$ is a vector of parameters given by

$$x = (\gamma_0, \gamma_1, \dots). \tag{20}$$

For consistency with the notation of Appendix B we set $x_0 = \gamma_0$, $x_1 = \gamma_1$, $\dots$. For a given $f \in \mathbf{H}$ we next define a probability measure on $I^{\mathbb{N}}$. The construction of this measure relies on the general construction of measures on infinite product spaces ([3, page108]) by means of conditional densities which we will assume factorize as the product of two nonnegative functions

$$p_n(x_n|x_0, \dots, x_{n-1}, f) \propto \ell_n(x_n|x_0, \dots, x_{n-1}, f)\ \pi(x_n|x_0, \dots, x_{n-1}) \tag{21}$$

where the constant of proportionality is in general a function of $f, x_0, \dots, x_{n-1}$. For clarity of exposition we present the construction of p_n in two stages.

Stage I (Construction of π): We introduce a distribution $\pi(x_n|x_0, \dots, x_{n-1})$ on the set I. Define

$$\pi(x_n|x_0, \dots, x_{n-1}) = \pi(x_n) = \pi_s(s_n)\pi_u(u_n|s_n)\pi_\xi(\xi_n|s_n) \tag{22}$$

where $\pi_s(s_n)$ is the uniform distribution on $[\delta, (b-a)]$, $\pi_u(u_n|s_n)$ is the uniform distribution on the discrete set $\{a, a+s_n/2, \dots, a+Ks_n/2\}$ with $K = \lfloor \frac{(b-a)\ 2}{s_n} \rfloor$

and $\pi_\xi(\xi_n|s_n)$ is the uniform distribution on the discrete set $\{0, \frac{\pi}{s_n}, \frac{2\,\pi}{s_n}, \ldots, \frac{J\,\pi}{s_n}\}$ where $J = \lfloor \frac{2\,s_n}{\delta} \rfloor$. This distribution is clearly motivated by the finite subdictionary described in Section 2.1. Once the scale s is sampled we have a uniform distribution over the discrete grid. Without loss of generality we consider the density $\pi(x_n)$ to be properly normalized and will also use π to denote the induced measure on I. We use the label γ when referring to a generic coordinate $x_n \in I$.

Stage II (Construction of ℓ_n):

Let $w \in \mathbf{H}$ be a sample from (discrete) Gaussian white noise and define the random variables on the probability space (I, π)

$$Y_w(\gamma) = \frac{\left|\left\langle w, g_{(\gamma,\phi)} \right\rangle\right|}{\|w\|} \tag{23}$$

and

$$Y(\gamma) = \mathbf{E}\left(\frac{\left|\left\langle w, g_{(\gamma,\phi)} \right\rangle\right|}{\|w\|}\right) \tag{24}$$

where the expectation is taken with respect to the underlying measure of the white noise process. Thus, the idea is to study the probability distribution of inner products between Gaussian white noise and the dictionary elements. Similarly, given coordinates $x_0, \ldots, x_{n-1}$, define the following random variables on (I, π) for the signal f and its residuals $R^n f$:

$$Y_n(\gamma) = \frac{\left|\left\langle R^n f, g_{(\gamma,\phi)} \right\rangle\right|}{\|R^n f\|}. \tag{25}$$

In Figure E.1 we plotted the densities of Y and Y_w, in Figure E.2 we plotted the densities Y and Y_0 for a given signal f. These definitions provide the concepts to decide whether the dictionary can be used to denoise the given signal. We want to isolate a subset of I where the corresponding Gabor functions have high probability of correlating well with the function and low probability of correlating well with the noise. To achieve this define

$$H(s) = \left(\pi\left(\{\gamma | Y_0(\gamma) \geq s\}\right) - \pi\left(\{\gamma | Y(\gamma) \geq s\}\right)\right)$$

and the *coherence threshold parameter* ρ by

$$\rho = \max\left(\arg\left(\max_{s\in(0,1)} H(s)\right)\right) \tag{26}$$

If such a ρ does not exist, we consider this fact an indication that the dictionary is not well suited to denoise the given signal. It is easy to see that the

possible values of $(\arg(\max_{s\in(0,1)} H(s)))$ are given by points of intersection of the densities of Y and Y_0. This value is not enough to be able to distinguish signal from noise with high probability. To see this we introduce the notation

$$r_n(\rho,\delta) = \frac{\pi\left(\{\gamma | Y(\gamma) \in [\rho, \rho+\delta]\}\right)}{\pi\left(\{\gamma | Y_n(\gamma) \in [\rho, \rho+\delta]\}\right)}. \tag{27}$$

The ratio $r_0(\rho,\delta)$ can be close to one for some $\delta > 0$. This is the case in Figure E.2 where the densities of Y and Y_0 (Y_0 computed with an example signal from Section 4.4) are displayed. A confidence level $\eta \in [0,1]$ has to be introduced to guarantee that r_0 remains small. Given η, define the *noise threshold parameter* $\tau \in [\rho, 1]$ as the smallest number such that $r_0(\tau,\delta) \leq \eta$ for all $\delta > 0$. Both ρ and τ are displayed in Figure E.2 for $\eta = 0.25$. A description of how to compute this value of η is postponed until Section 4.4. We remark that when the signal to noise ratio is large, τ is close to ρ.

For a given residual $R^n f$, *probabilistic coherent structure* is then given by those labels γ that satisfy

$$\frac{\left|\left\langle R^n f, g_{(\gamma,\phi)}\right\rangle\right|}{\|R^n f\|} \geq \tau. \tag{28}$$

Inequality (28) suggests the following definition for $\ell_n(x_n | x_0, \ldots, x_{n-1}, f)$. Let

$$\ell_n(x_n | x_0, \ldots, x_{n-1}, f) = \frac{\left|\left\langle R^n f, g_{(x_n,\phi_n)}\right\rangle\right| \ \chi_{\{Y_n \geq \tau\}}(x_n)}{\|R^n f\|} \tag{29}$$

where χ_A is the characteristic function of the set A.

Having completed the construction of ℓ_n, combining equations (29) and (22) and (21) we have constructed a density $p_n(x_n | x_0, \ldots, x_n, f)$ (up to a constant) which puts more weight on values of x_n which correspond to large inner products.

The densities p_n define a probability measure μ on $I^{\mathbb{N}}$; the probability space (Ω, μ) mentioned in the introduction is then $(I^{\mathbb{N}}, \mu)$. The new random variables to be introduced below are defined on this probability space. Our goal is to compute expected values of certain functionals defined on $I^{\mathbb{N}}$ which will depend on a stopping time T also defined on $I^{\mathbb{N}}$, so $T(x)$ will be a nonnegative integer when $x \in I^{\mathbb{N}}$. Given a stopping time $T(x)$ (to be defined shortly) define the coherent component functionals $C_t : I^{\mathbb{N}} \longrightarrow \mathbb{R}, \quad (t \in \{t_i\})$, by

$$C_t(x) = \sum_{k=0}^{T(x)} \left\langle R^k f, g_{(x_k,\phi_k)}\right\rangle g_{(x_k,\phi_k)}(t) \tag{30}$$

where

$$R^{n+1}f = f - \sum_{k=0}^{n} \left\langle R^k f, g_{(x_k,\phi_k)} \right\rangle \, g_{(x_k,\phi_k)} \tag{31}$$

and ϕ_k satisfies $\tan\phi_k = b/a$ with $\left\langle R^k f, g_{x_k} \right\rangle = a + ib$; hence ϕ_k becomes a function of f and $x_0, \ldots, x_k$. The pointwise estimates are given by expected values

$$\mathbf{E}_\mu(C_t) = \int C_t(x) \, d\mu(x). \tag{32}$$

The vector approximation is denoted by $\mathbf{E}_\mu(C)$.

We now define the stopping time $T : I^{\mathbb{N}} \to \mathbb{N}$. To simplify, we use the notation introduced in (27), define

$$T(x) = \inf \{n \,|\, r_{n+1}(\tau, 1) \geq 1\} \tag{33}$$

In other words, $T(x)$ is the smallest integer such that the probability of finding probabilistic coherent structures at iteration $T(x) + 1$ is smaller than the probability of finding noise.

We have found in our numerical experiments that this stopping time is integrable, hence finite. This is the natural stopping rule given the above formalism. For completeness, we argue now that the rule can be easily augmented to make it finite by simple inspection. To see this, note that without loss of generality we can take $\tau > 0$ and $\tau \geq \alpha$, where α is the parameter in equation (4), it follows from the definition of l_n in (29) that

$$|\left\langle R^n f, g_{(x_n,\phi_n)} \right\rangle| \geq \|R^n f\| \; \tau \geq \alpha \sup_{\gamma \in \Gamma} |\langle R^n f, g_{\gamma,\phi} \rangle|. \tag{34}$$

Then Theorem 1 is applicable, in particular equation (9) holds. From this equation it follows that $\pi\left(\{\gamma | Y_n(\gamma) \in [\tau, 1]\}\right)$ is zero for some integer n or $\|R^n f\|$ converges to zero. Hence we can augment the rule (33) by requiring to stop as soon as $\|R^n f\|^2$ is smaller than the variance of the noise process.

The above framework will be called *probabilistic matching pursuit* (PMP).

3.2 Extensions and Remarks

Next we comment briefly on the formalism just introduced. Specific instances of the formal integrals appearing in (32) will be computed by producing samples from μ and then computing an ergodic average. Sample points $x^k \in I^{\mathbb{N}}$ are generated by sampling each component x_n^k given the previous components

$x_0^k, \ldots, x_{n-1}^k$. From (30) it follows that only $T(x^k)$ components are needed. To sample from the densities given by (21) we have used the rejection method (Appendix A). It is essential to use a sampling method that does not require knowledge of the value of the normalization constants involved.

Given prior information on the frequency and/or spatial content of the given signal it is possible to modify the density $\pi(\cdot)$. It is important to notice that the PMP framework contains (formally) the MP algorithm as a special case. To see this let π be a uniform prior over the whole dictionary or finite sub-dictionary and let $\ell_n(x_n)$ be a Dirac density supported in the label that gives the maximum inner product.

It can be seen that even when the signal-to-noise ratio is relatively high, the MP stopping rule may be far from optimal. In practical terms, this means that the MP algorithm may overfit the underlying signal. The process of computing averages has the positive effect of suppressing undesirable components which were not detected by the stopping rule. At the same time, noise recognition (the stopping rule) becomes a more pressing issue in our approach given that we have to stop the iterations many times.

4 Performance Analysis of PMP

We next describe in detail the basic algorithms underlying a software implementation of PMP. We also analyze the expected running time. Then with simple examples we analyze why and when PMP improves the estimates given by MP. Numerical examples of typical denoising tasks are also provided.

4.1 Description of the Algorithm

In this section we provide details on computing the parameters that specify a model of PMP. Our implementation of MP differs from that of [13] in that we do not periodize the functions nor do we add the Dirac and Fourier dictionaries. Moreover, the implementation described in [13] maximizes only over $|\langle R^n f, g_\gamma \rangle|$ in (16) and *then* performs a local optimization in order to find ϕ.

Our estimates are given by

$$\begin{aligned} \mathbf{E}_\mu(C_{t_i}) &= \int C_{t_i}(x)\, d\mu(x) = \int \sum_{k=0}^{T(x)} \left\langle R^k f, g_{x_k} \right\rangle g_{x_k}(t_i)\, d\mu(x) \\ &= \lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} C_{t_i}(x^k). \end{aligned} \tag{35}$$

The points x^k are sampled from the joint densities

$$p(x_0, \ldots, x_n | f) = \prod_{k=0}^{n} p_k(x_k | x_0, \ldots, x_{k-1}, f). \tag{36}$$

We need only sample the coordinates $x_0^k, \ldots, x_{T(x^k)}^k$ for each x^k. We do this by means of the conditional distribution method in conjunction with the rejection method (see [8] and Appendix A). We say that the k^{th} iteration of PMP has ended when the coordinates $x_0^k, \ldots, x_{T(x^k)}^k$ of the point $x^k \in I^{\mathbb{N}}$ have been sampled.

To use the rejection algorithm we need to bound p_n from above with a density from which we can sample. Let d_n be the constant of normalization for p_n. We have the obvious inequality

$$p_n(x_n) = d_n \ell_n(x_n) \pi(x_n) \leq d_n \pi(x_n) \tag{37}$$

so $d_n \geq 1$. We are now in a position to describe the sampling of the coordinate x_n^k given the previous $x_0^k, \ldots, x_{n-1}^k$ coordinates. We sample x_n^k by sampling $x' = (s', u', \xi')$ from $\pi(x)$ and then we accept $\gamma' = (s', u', \xi')$ with probability

$$\gamma_A = \frac{\left| \left\langle R^n f, g_{(s', u', \xi', \phi')} \right\rangle \right| \chi_{\{Y_n \geq \tau\}}(x')}{\|R^n f\|}. \tag{38}$$

We continue until acceptance of a proposed coordinate $x_n^k = \gamma'$ and end the k^{th} iteration if $T(x^k) = n$, where $T(x^k)$ is given by (33). Otherwise we repeat the above procedure to sample x_{n+1}^k.

4.2 Expected Running Time of the Algorithm

Let K be the number of proposals before x_n is accepted. It follows from [8, page 42] that $\mathbf{E}(K) = d_n$. Given that the number of computations for each proposed component is bounded by N, an iteration of PMP for $T(x) = P$ components requires a number of operations of the order $O(\sum_{i=0}^{P} d_n N)$ (making use of the fast expansions of Appendix C). It is easy to check that

$$d_n \leq \frac{1}{\tau \; \pi(Y \geq \tau).} \tag{39}$$

So in fact we can see that if $\mathbf{E}(T)$ denotes the expected stopping time, one iteration of PMP takes of the order $O(N\mathbf{E}(T))$ operations.

The closer the index n of x_n is to P the slower the acceptance step becomes because d_n becomes larger. This fact is independent of N and can be used to improve the stopping rule in practical situations. As described in Appendix B,

many of the computations performed in previous iterations can be reused in later iterations of PMP. Moreover, there are several methods to accelerate the rejection algorithm, some of which are mentioned in Appendix A.

4.3 Convexity Improvements of PMP

In this section we show with simple examples the main reason for the improvement in the estimates given by PMP. This remark also indicates under which conditions one can expect an improvement of PMP relative to MP. To better illustrate the phenomena we consider the functions f_i $i = 1, 2, 3$ defined in the next section. These functions have no Gaussian noise added. This fact does not prevent the application of the PMP algorithm given that we can take the variance of the Gaussian noise, which is needed to apply the formalism, as small as we want because the distribution of Y in (24) is independent of the variance. The letter f will stand for any of the three functions f_i. Given x^k, sampled from μ, we denote with $f^P(x^k)$ the approximation to f given by the k^{th} iteration of PMP. We perform the following numerical experiment: given $\epsilon > 0$ we will stop the approximation of iteration k when

$$\text{RMSE}(f^P(x^k)) = \frac{\|f - f^P(x^k)\|}{\|f\|} \leq \epsilon \tag{40}$$

Therefore we have exchanged (33) with the stopping rule (40). Denote the approximation of PMP by $f^P = \int f^P(x)\, d\mu(x)$. Similarly, let f^M denote the approximation of MP obtained by the same stopping rule. In general, if ϵ is small, we have $\frac{\|f-f^P(x^k)\|}{\|f\|} \approx \text{RMSE}(f^M) = \frac{\|f-f^M\|}{\|f\|}$ for each x^k. As mentioned in the introduction from Jensen's inequality we have:

$$\frac{\|f - f^P\|}{\|f\|} \leq \int \frac{\|f - f^P(x)\|}{\|f\|} d\mu(x) \tag{41}$$

and the right hand side is approximately equal to $\frac{\|f-f^M\|}{\|f\|}$. Tables D.1. and D.2. report the values of the left hand side (see the column titled "PMP RMSE") and right hand side (see the column titled "Avg RMSE") of (41). We also report the RMSE of MP (see the column titled "MP RMSE"). These values give an idea of the percentage of improvement in the RMSE to be expected from PMP and, most importantly, it indicates that PMP can generate *different* expansions that complement each other to reinforce a better reconstruction.

Table D.1
Table D.2

Now consider a noisy function $f = z + \sigma w$. When can we expect the above percentage improvements to be reflected in the denoising task? To answer this question consider the situation where MP delivers optimal denoised approximations. In particular, this is the case when z is an element in the dictionary or a linear expansion of dictionary elements $z = \sum_{\beta \in B} c_\beta g_\beta$ with $\langle g_\beta, g_{\beta'} \rangle \approx 0$,

$\beta, \beta' \in B$. In this situation one expects that $\frac{\|f-f^P(x)\|}{\|f\|} > \frac{\|f-f^M\|}{\|f\|}$ for most x in the support of μ and hence the PMP denoised approximation may not offer an improvement. A numerical example is presented in Section 4.4. Redundant dictionaries are not orthogonal and the ideal z considered above will not be typical in practice. In general we have observed that PMP improves over MP when $\frac{\|f-f^P(x)\|}{\|f\|} \approx \frac{\|f-f^M\|}{\|f\|}$ for most x in the support of μ. The simplest example of this situation is when $z = g_{\beta_1} + g_{\beta_2}$ and $|\langle g_{\beta_1}, g_{\beta_2}\rangle| > 0$. Let us introduce the following convenient notation for the residuals $R_i^1 f = f - \langle f, g_{\beta_i}\rangle g_{\beta_i}$. MP can give one of the following two denoised approximations $f_1^P = \langle f, g_{\beta_1}\rangle g_{\beta_1} + \langle R_1^1 f, g_{\beta_2}\rangle g_{\beta_2}$ or $f_2^P = \langle f, g_{\beta_2}\rangle g_{\beta_2} + \langle R_2^1 f, g_{\beta_1}\rangle g_{\beta_1}$. If $\langle f, g_{\beta_1}\rangle \approx \langle f, g_{\beta_2}\rangle$ both expansions a priori offer the same quality of reconstruction, and both will be included in the PMP reconstruction but only one in the MP reconstruction. A numerical result for an instance of this situation is reported in Section 4.4. Notice that these multiple expansions will not be available in a dictionary consisting of a single orthonormal basis.

4.4 Simulated Examples

In our numerical experiments we set $a = 0$, $\delta = 1$, $b = 127$ and $c = 2\pi$, i.e., we consider signals sampled uniformly one unit apart in the interval $[0, 127]$. The expectation in (24) was computed with 30 averages of samples of Gaussian white noise w. The value of the parameter $\eta = 0.25$ (introduced in Section 3.1) was obtained by studying the effect on the value of ρ when changing Y by Y_w (for many samples w) in equation (26). The underlying functions which we study in the simulated examples are

$$f_1(t) = \begin{cases} t^2, & t \in [0, b/3), \\ t^2(\sin(t/12) + \sin(t/4), & t \in [b/3, 2b/3) \\ -t^2 + 100t, & t \in [2b/3, b] \end{cases} \tag{42}$$

$$f_2(t) = g_{(30,52,\pi/5,0)}(t) + g_{(60,30,\pi/15,0)}(t) + g_{(50,90,\pi/25,0)}(t) \tag{43}$$

$$f_3(t) = \sin(t/2) + \sin(t/8) \tag{44}$$

The signals $f_{i,j}$ associated with f_i were obtained by adding Gaussian white noise with variance such that the signal to noise ratio is l_j, $j = 1, \ldots, 4$; see Table D.3. The graphs of some of the functions $f_{i,j}$ are given in Figures E.3 through E.8; the original signals f_i are shown by the solid line in each of the figures while the signals with added noise are shown by the dotted lines. The density of $Y_0(\gamma) = \left\langle f_{1,3}, g_{(\gamma,\phi)} \right\rangle / \|f_{3,1}\|$ is plotted alongside the density of $Y(\gamma)$ and $Y_w(\gamma)$ in Figure E.1. (see equations (24) and (23)). The values of ρ are obtained as crossing points in these type of graphs. It turns out that the values of τ are similar in each of our examples; see Table D.4 below.

Table D.3

Figure E.3–E.8

Figure E.1

The reconstructed functions will be denoted by $f_{i,j}^P$ and $f_{i,j}^M$ for methods PMP and M respectively. The Relative Mean Squared Error (RMSE) of the reconstruction of signal f_i at noise level j by method X is defined by

$$\mathrm{RMSE}(f_{i,j}^X) = \frac{\|f_i - f_{i,j}^X\|}{\|f_i\|} \tag{45}$$

where the norms are induced by the inner products. A detailed list of the RMSEs in each of the cases is given in Table D.5. Table D.5

It is important to note that the reported relative mean squared errors for the MP algorithm are hand-picked to be the best possible (for our implementation). That is, we have stopped the MP expansions when it gives the best possible expansion. This is done to better highlight the improvements of PMP without having to worry about problems in the stopping rule of MP.

If, more realistically, MP is stopped automatically as suggested in Section 2.2, then the improvements of PMP over MP are more dramatic. For example, Figure E.9 illustrates the reconstruction of $f_{3,5}$ by matching pursuit, while Figure E.10 shows the reconstruction by method A with our automatic stopping rule. A detailed picture of the progress of MP in this case is given in Table D.6. An improved version of the automatic stopping rule described in Section 2.2 stops MP at an RMSE of 0.746 which is clearly not the best. The automatic stopping rule for method PMP gives an RMSE of 0.477. Figure E.9 Figure E.10

We now give an example of the situation described at the end of Section 4.3. Consider the functions

$$f_4(t) = g_{(64,64,\pi/8,0)}(t) + g_{(16,120,\pi/8,\pi)}(t) \tag{46}$$

$$f_5(t) = g_{(64,64,\pi/8,0)}(t) + g_{(16,56,\pi/8,\pi)}(t) \quad . \tag{47}$$

The dictionary functions that make up f_4 are nearly orthogonal. One of these functions is translated and used to define f_5, so the quasi orthogonality is then lost and multiple expansions, of a priori good reconstruction quality, become available. The graphs of f_4 and f_5 and its noisy versions are shown in Figures E.11 and E.12. The RMSE obtained from MP for the noisy functions $f_{4,1}$ and $f_{5,1}$ are 0.089 and 0.231 respectively. The RMSE obtained from PMP are 0.170 0.198 respectively. Figure E.11 Figure E.12

5 Conclusion

We have introduced an extension of the matching pursuit algorithm which randomly generates multiple expansions of a signal with respect to a redun-

dant dictionary. Probability is used as a technique to identify a subset of the dictionary vectors which potentially offer a good reconstruction of the given signal.

Probabilistic matching pursuit really is an extension of MP in the sense that MP is a special case of PMP. We have selected a simple method of deciding how to choose the next term of an expansion by placing a uniform probability distribution on dictionary elements that match the signal above a certain threshold value. With that simple method, we have found that each random expansion obtained has more components than an MP expansion but the average time for the computation of an expansion is smaller.

Random selection of expansions also provides us with the opportunity of improving results by averaging. Some conditions under which this is actually the case are discussed and studied numerically. We conclude that our particular implementation of PMP is better than MP when MP is not optimal, which is generally the case when a signal does not have an efficient representation in terms of a small number of dictionary elements.

In summary, adaptive signal analysis offers a collection of good quality expansions of a signal, and we show that better estimates are obtained under certain conditions by averaging over a reasonable random selection of those expansions.

5.1 Future Prospects

We suspect that better selection methods can give clear improvements over MP under all conditions. In particular, we could make better use of a priori information obtained from initial analysis of the signal, for example to always select an overwhelmingly good match from the dictionary, which would ensure that PMP outperforms MP in all cases we have examined.

The PMP formalism introduced here gives strong indication that it is worth studying the degeneracy implicit in adaptive methods on redundant dictionaries. In particular, the coherent subset of vectors introduced in the paper can be studied by means other than the densities introduced here. For example, simulated annealing could be used to explore the region.

A Appendix: Rejection Method

In our implementations we have used the rejection method [8, page 42] to generate samples. Here we briefly describe the algorithm.

Suppose we are given densities f, g with associated random variables X_f, X_g and a constant $d \geq 1$ such that

$$f(x) \leq dg(x) \tag{A.1}$$

for all $x \in \mathbb{R}^d$. Samples x_f from X_f can be obtained as follows: sample, independently, x_g from X_g and u from a random variable uniformly distributed on $[0, 1]$. Let $x_f = x_g$ if

$$\frac{f(x_g)}{dg(x_g)} \geq u$$

otherwise repeat the above. The sequence generated in this way are then independent samples from X_f. To connect this notation with that of (37) we take $p_n = f$, $d_n = d$ and $\pi = g$.

Unless sharp bounds for the densities are available, the method could be inefficient. There are methods of accelerating the algorithm; for instance [20] proposes combining the usual rejection method with the Metropolis algorithm. A simple way to accelerate the rejection algorithm for our model is to use an empirical (approximate) bound for $\frac{|\langle R^n f, g_\beta \rangle|}{\|R^n f\|}$ in combination with the methods described in [20]. Faster dynamical sampling methods, such us Markov Chain Monte Carlo [11], [17], are available but are less reliable.

B Appendix: The Bernoulli Shift

Next we describe another method, based on the Bernoulli shift, which can be used to reduce the overall running time of the algorithm. The Bernoulli shift enables one to make use of partial computations when running the rejection algorithm. Our discussion is applicable to models of PMP which use the rejection method for sampling. We begin by indicating why MP can be efficiently implemented [14, page 415]. Given

$$R^{n+1} f = R^n f - \langle R^n f, g_{\beta_n} \rangle \, g_{\beta_n} \tag{B.1}$$

the MP algorithm maximizes $|\langle R^{n+1} f, g_\beta \rangle|$ in a sub-optimal way. From (16) and by unraveling (B.1) we see that for doing this computation we only need to compute $\langle f, g_{\gamma_i} \rangle$, $\gamma_i \in \Gamma_\alpha$ and $\langle g_{\beta_k}, g_{\gamma_i} \rangle$, $\gamma_i \in \Gamma_\alpha$ for $k = 0, \ldots, n$. These computations can be arranged in a way that partial computations (namely $\langle R^{n+1} f, g_{\gamma_i} \rangle$) can be reused. Details are given in [13].

Equation (B.1) can be used in the probabilistic setting by keeping the values of $\langle f, g_{\beta_k} \rangle$ and $\left\langle g_{\beta_i}, g_{\beta_j} \right\rangle$ in memory where the parameters $\gamma_p = (s_p, u_p, \xi_p)$ have been sampled from $\pi(\gamma)$ given by (22). Below we show how this idea can

be formalized. We restrict the presentation to our specialized setting; more general and extensive discussions are given in [4] and [2].

Let (Ω, P) be a probability space and $X : \Omega \longrightarrow I^{\mathbb{N}}$ be a stochastic process with coordinates (X_i), $i \in \mathbb{N}$ where $I = I_1 \times \cdots \times I_3$, and the intervals I_i were introduced at the beginning of Section 3. We denote the law of X by $\mu = P \circ X^{-1}$. We are interested in evaluating integrals of functionals along paths of this stochastic process, i.e., given a real valued functional C on $I^{\mathbb{N}}$ we want to evaluate

$$\mathbf{E}(C) = \int C(X(\omega))\, dP(\omega). \tag{B.2}$$

We assume that C depends only on a finite number of components of $X(\omega)$ for each ω; actually, these components are $0, \ldots, T(X(\omega))$. $T \circ X$ is a finite stopping time on Ω. At this point we indicate how this notation relates to that of Section 3. The space (Ω, P) is the (formal) probability space underlying the sampling process by means of the rejection method. The law of X, i.e., $\mu = P \circ X^{-1}$, corresponds to the measure on $I^{\mathbb{N}}$ mentioned below equation (21).

The process of producing uniformly distributed random numbers and sampling with the rejection algorithm is formalized as follows. There exists a probability space $(\Omega', d\lambda)$, which in our applications will be $[0,1]^4$ with the Lebesgue product measure and Borel sets, and a stochastic process $V = (V_i)$ with $V : \Omega \longrightarrow {\Omega'}^{\mathbb{N}}$. We assume that the coordinates of V are random variables V_i which are independent and identically distributed with measure $d\lambda = P \circ V_i^{-1}$. The space ${\Omega'}^{\mathbb{N}}$ is considered as a probability space with the product measure $d\lambda^{\otimes_{\mathbb{N}}} = P \circ V^{-1}$. The following notation is used below: $V(\omega) = v = (v_0, v_1, \ldots)$, where $v_i = ([v]_1, \ldots, [v]_4) \in [0,1]^4$ and $X(\omega) = x = (x_0, x_1, \ldots)$ where $x_i \in I$. Finally, the sampling process (implicitly) defines a function $F : {\Omega'}^{\mathbb{N}} \longrightarrow I^{\mathbb{N}}$

$$X = F(V) \tag{B.3}$$

Equation (B.3) amounts to a representation of X on $({\Omega'}^{\mathbb{N}}, d\lambda^{\otimes_{\mathbb{N}}})$. It is this fact that allows the use of the Bernoulli shift. The left shift is defined by $\Theta v = z$ for $v, z \in {\Omega'}^{\mathbb{N}}$ and $[z]_i = [v]_{i+1}$. The left shift is an ergodic transformation; this fact justifies the replacement of integrals by limits below.

$$\begin{aligned} \mathbf{E}(C) &= \int C(z)\, d\mu(z) = \int C(X(\omega))\, dP(\omega) \\ &= \int C(F(V(\omega)\, dP(\omega) = \int C(F(v)\, d\lambda^{\otimes_{\mathbb{N}}}(v) \\ &= \lim_{K \to \infty} \frac{1}{K} \sum_{k=0}^{K-1} C(F(\Theta^k v)). \end{aligned} \tag{B.4}$$

Therefore the computation of (B.4) amounts to evaluating C at a point $F(v) = x^0 = (x_0^0, \ldots, x_{T(x^0)}^0, \ldots)$ and its translates by the shift, i.e., $x^k = F(\Theta^k v)$.

Now the key idea is that in order to compute $F(\Theta^k v)$ we may reuse previously computed quantities which were needed to obtain x^j, $j = 1, \ldots, k-1$. For example, let us see how x_0^0 is constructed. If

$$v_0 = ([v_0]_1, [v_0]_2, [v_0]_3, [v_0]_4) \in [0,1]^4 \tag{B.5}$$

are four random numbers and g_0 represents the associated random Gabor function. Explicitly, $g_0 = g_{\gamma_0}$, $\gamma_0 = (s_0, u_0, \xi_0)$ and, as described below (22) $s_0 = ([v_0]_1 (b - a - \delta))$, $u_0 = \left(a + \frac{K\, s_0\, [v_0]_2}{2}\right)$, $\xi_0 = \frac{J\, \pi\, [v_0]_3}{s_0}$, with $K = \lfloor \frac{(b-a)\ 2}{s_0} \rfloor$ and $J = \lfloor \frac{2 s_0}{\delta} \rfloor$. Then if $\frac{|\langle f, g_0 \rangle|}{\|f\|} \geq \tau$ and

$$\frac{|\langle f, g_0 \rangle|}{\|f\|} \geq [v_0]_4 \tag{B.6}$$

we let

$$x_0^0 = (s_0, u_0, \xi_0), \tag{B.7}$$

otherwise we perform the same computations with

$$g_1 = g_{\gamma_1}. \tag{B.8}$$

In general, there exists an integer k_0 such that

$$x_0^0 = (s_{k_0}, u_{k_0}, \xi_{k_0}). \tag{B.9}$$

Similar computations will give $x_1^0, \ldots, x_{T(x^0)}^0$ for a given stopping time T. While performing these computations, the following numbers must be computed:

$$\langle f, g_i \rangle \quad \langle g_i, g_j \rangle \quad \text{for } i, j = 0, \ldots, k_{T(x^1)},\ i \leq j, \tag{B.10}$$

where in general $k_{T(x^1)} \gg T(x^1)$. Regarding the generation of random numbers, we only need to generate $k_{T(x^1)} + 1$ batches (of size four). Therefore $v = (v_0, \ldots, v_{k_{T(x^1)}}, \ldots)$ where the second set of dots indicate that those coordinates have not yet been generated. The next point, x^1, is computed by using $\Theta v = (v_1, \ldots)$. The quantities in (B.10) are saved in memory; we may have to generate v_n for $n > k_{T(x^0)}$ only if $T(x^1) > T(x^0)$. This scheme allows for a systematic reuse of partial computations involving expensive function evaluations.

C Appendix: Inner Products of Gabor Functions

In this appendix we consider the problem of efficiently evaluating inner products of Gabor functions. Four different kinds of Gabor functions are considered:

real and complex valued functions which are either continuous (defined on the real line) or discrete (defined on the discrete subgroup $\delta\mathbb{Z}$ of the real line). In the continuous case exact formulas in terms of elementary functions are possible, while in the discrete case only approximations are possible, in which case we find efficient series approximations.

C.1 Complex Gabor functions on $\mathbb{R}$

The complex Gabor functions on the real line are

$$g_{(s,u,\xi)}(t) = \frac{2^{1/4} K_{(s,u,\xi)}}{\sqrt{s}} e^{-\pi(t-u)^2/s^2} e^{i\xi t}, \quad s \in \mathbb{R}^+, u, \xi \in \mathbb{R} \tag{C.1}$$

where $K_{(s,u,\xi)}$ is a normalization constant; it is not difficult to see that $K_{(s,u,\xi)} = 1$ in this case. The inner product of two such functions is

$$\langle g_0, g_1 \rangle = \frac{2^{1/2} K_0 K_1}{\sqrt{s_0 s_1}} \int_{\mathbb{R}} e^{-\pi(t-u_0)^2/s_0^2} e^{-\pi(t-u_1)^2/s_1^2} e^{-i(\xi_1-\xi_0)t}\, dt. \tag{C.2}$$

The above integral may be evaluated explicitly by completing the square in the exponent to obtain

$$-\pi \left(\frac{t-u_0}{s_0}\right)^2 - \pi \left(\frac{t-u_1}{s_1}\right)^2 = -A(t-B)^2 + C \tag{C.3}$$

where

$$A = \pi \left(\frac{1}{s_0^2} + \frac{1}{s_1^2}\right), \quad B = \frac{\pi}{A}\left(\frac{u_0}{s_0^2} + \frac{u_1}{s_1^2}\right), \quad C = -\pi \left(\frac{u_0^2}{s_0^2} + \frac{u_1^2}{s_1^2}\right) + AB^2, \tag{C.4}$$

and then evaluating the Fourier transform of a Gaussian to obtain

$$\langle g_0, g_1 \rangle = \left(\frac{2\pi}{s_0 s_1 A}\right)^{1/2} e^{-(\xi_1-\xi_0)^2/4A - i(\xi_1-\xi_0)B + C} \tag{C.5}$$

where A, B, C are given by (C.4).

C.2 Real Gabor functions on $\mathbb{R}$

Real Gabor functions on $\mathbb{R}$ are formed by replacing $e^{i\xi t}$ with $\cos(\xi t + \phi)$:

$$g_{(s,u,\xi,\phi)}(t) = \frac{2^{1/4} K_{(s,u,\xi,\phi)}}{\sqrt{s}} e^{-\pi(t-u)^2/s^2} \cos(\xi t + \phi), \quad s \in \mathbb{R}^+, u, \xi \in \mathbb{R}, \phi \in [0, 2\pi), \tag{C.6}$$

where $K_{(s,u,\xi,\phi)}$ is a normalization constant to be found (it is generally not 1 in this case). The inner product of two such functions is

$$\langle g_0, g_1 \rangle = \frac{2^{1/2} K_0 K_1}{\sqrt{s_0\, s_1}} \int_{\mathbb{R}} e^{-\pi(t-u_0)^2/s_0^2} e^{-\pi(t-u_1)^2/s_1^2} \cos(\xi_0 t + \phi_0) \cos(\xi_1 t + \phi_1)\, dt. \quad \text{(C.7)}$$

The above integral may be reduced to the complex case by the trigonometric identity

$$4 \cos X \cos Y = e^{i(X+Y)} + e^{i(X-Y)} + e^{i(-X+Y)} + e^{i(-X-Y)} \quad \text{(C.8)}$$

to obtain

$$K_{(s,u,\xi,\phi)} = 2^{1/2} \left[\cos(2(\xi u + \phi)) e^{-(s\xi)^2/2\pi} + 1 \right]^{-1/2} \quad \text{(C.9)}$$

and

$$\langle g_0, g_1 \rangle = \frac{2^{1/2} K_0 K_1}{\sqrt{s_0\, s_1}} \left(\frac{\pi}{4A} \right)^{1/2} e^C \left[\cos((\xi_0 + \xi_1)B + (\phi_0 + \phi_1)) e^{-(\xi_0+\xi_1)^2/4A} + \cos((\xi_0 - \xi_1)B + (\phi_0 - \phi_1)) e^{-(\xi_0-\xi_1)^2/4A} \right] \quad \text{(C.10)}$$

where A, B, C are again given by (C.4).

The above formulas may be used as approximations in the discrete case by considering the discrete sums as Riemann sum approximations for corresponding integrals; however, more accurate evaluations methods using theta functions are available, and are explored in the following two sections.

C.3 Complex Gabor functions on $\delta\mathbb{Z}$

The discrete complex Gabor functions are just samples of the continuous Gabor functions at the points $\delta\mathbb{Z}$:

$$g_{(s,u,\xi)}(n) = \frac{2^{1/4} K_{(s,u,\xi)}}{\sqrt{s}} e^{-\pi(\delta n - u)^2/s^2} e^{i\xi\delta n}, \quad s \in \mathbb{R}^+, u, \xi \in \mathbb{R}. \quad \text{(C.11)}$$

The inner product of two such functions is just (C.2) with the integral replaced by a sum:

$$\langle g_0, g_1 \rangle = \frac{2^{1/2} K_0 K_1}{\sqrt{s_0 s_1}} \sum_{n=-\infty}^{\infty} e^{-\pi(\delta n - u_0)^2/s_0^2 - \pi(\delta n - u_1)^2/s_1^2} e^{-i(\xi_1 - \xi_0)\delta n} \tag{C.12}$$

$$= \frac{2^{1/2} K_0 K_1}{\sqrt{s_0 s_1}} \sum_{n=-\infty}^{\infty} e^{-\delta^2 A n^2 + \delta(2AB - i(\xi_1 - \xi_0))n - AB^2 + C} \tag{C.13}$$

$$= \frac{2^{1/2} K_0 K_1}{\sqrt{s_0 s_1}} e^{C - AB^2} \sum_{n=-\infty}^{\infty} e^{-\delta^2 A n^2} e^{\delta(2AB - i(\xi_1 - \xi_0))n}. \tag{C.14}$$

The above expression may be written in terms of the theta function [21, page 464]

$$\theta_3(z;t) = \sum_{n=-\infty}^{\infty} e^{\pi i n^2 t} e^{2inz}. \tag{C.15}$$

Clearly

$$\langle g_0, g_1 \rangle = \frac{2^{1/2} K_0 K_1}{\sqrt{s_0 s_1}} e^{C - AB^2} \theta_3 \left(\delta \left(-iAB - (\xi_1 - \xi_0)/2 \right); \frac{i\delta^2 A}{\pi} \right). \tag{C.16}$$

In general, theta functions cannot be evaluated in terms of more elementary functions. We may approximate the value using the series (C.15), but for small values of $\delta^2 A$ the series converges rather slowly.

Fortunately, the theory of theta functions provides a way of quickly evaluating such expressions to a high degree of accuracy. Using the Poisson identity [21, page 475]

$$\theta_3(z;t) = (-it)^{-1/2} e^{z^2/\pi i t} \theta_3 \left(\frac{z}{t}; -\frac{1}{t} \right) \tag{C.17}$$

our theta function can be rearranged in a manner which facilitates rapid calculation:

$$\langle g_0, g_1 \rangle = \frac{2^{1/2} K_0 K_1}{\sqrt{s_0 s_1}} e^{C - AB^2} \left(\frac{\pi}{\delta^2 A} \right)^{1/2} e^{-(iAB + (\xi_1 - \xi_0)/2)^2/A}$$
$$\theta_3 \left(\frac{\pi(-iAB - (\xi_1 - \xi_0)/2)}{i\delta A}; \frac{i\pi}{\delta^2 A} \right). \tag{C.18}$$

The series for the latter theta function converges more rapidly the smaller the value of $\delta^2 A$. This series may be evaluated to accuracy ϵ in time $O(\sqrt{\log \epsilon})$. For many cases, only the leading term of the series for the theta function is required, in which case the formulas in the discrete case are equal to the formulas in the continuous case. As the variance of the Gaussian envelope of the Gabor function tends tends to the mesh width δ, more terms of the theta

series are required for accurate approximation, but this still allows us (for s bounded away from zero and for fixed δ) to evaluate inner products of Gabor functions in constant time.

C.4 Real Gabor functions on $\delta\mathbb{Z}$

These functions are just samples of the real continuous Gabor functions at the points $\delta\mathbb{Z}$:

$$g_{(s,u,\xi,\phi)}(n) = \frac{2^{1/4} K_{(s,u,\xi,\phi)}}{\sqrt{s}} e^{-\pi(\delta n - u)^2/s^2} \cos(\xi \delta n + \phi),$$
$$s \in \mathbb{R}^+, u, \xi \in \mathbb{R}, \phi \in [0, 2\pi). \quad \text{(C.19)}$$

The inner product of two such functions is just (C.7) with the integral replaced by a sum:

$$\langle g_0, g_1 \rangle = \frac{2^{1/2} K_0 K_1}{\sqrt{s_0 s_1}} \sum_{n=-\infty}^{\infty} e^{-\pi(\delta n - u_0)^2/s_0^2 - \pi(\delta n - u_1)^2/s_1^2}$$
$$\cos(\xi_0 \delta n + \phi_0) \cos(\xi_1 \delta n + \phi_1). \quad \text{(C.20)}$$

The above expression may again be reduced to the complex case by the trigonometric identity (C.8) to obtain

$$\langle g_0, g_1 \rangle = \frac{2^{1/2} K_0 K_1}{2\sqrt{s_0 s_1}} e^{C - AB^2} \left(\frac{\pi}{\delta^2 A}\right)^{1/2}$$
$$\mathrm{Re}\left(e^{i(\phi_1 + \phi_0)} e^{(iAB + (\xi_1 + \xi_0)/2)^2/A} \theta_3 \left(\frac{\pi(-iAB - (\xi_1 + \xi_0)/2)}{i\delta A}; \frac{i\pi}{\delta^2 A} \right) \right.$$
$$\left. + e^{i(\phi_1 - \phi_0)} e^{(iAB + (\xi_1 - \xi_0)/2)^2/A} \theta_3 \left(\frac{\pi(-iAB - (\xi_1 - \xi_0)/2)}{i\delta A}; \frac{i\pi}{\delta^2 A} \right) \right) \quad \text{(C.21)}$$

which again converges more rapidly the smaller the value of $\delta^2 A$ and may be evaluated to accuracy ϵ in time $O(\sqrt{\log \epsilon})$.

References

[1] B. Abramovich, Statistical Modeling by Wavelets, J. Wiley, 1999.

[2] M. B. Alaya, "On the simulation of expectations of random variables depending on a stopping time", Stochastic Analysis and Applications, **11**:2 (1993), 133–153.

[3] R. B. Ash, Real Analysis and Probability, Academic Press, New York, 1972.

[4] N. Bouleau and D. Lepingle, Numerical Methods for Stochastic Processes, Wiley and Sons, New York, 1994.

[5] S. Chen, D. Donoho, "Atomic decomposition by basis pursuit", SIAM Journal on Scientific Computing, To appear.

[6] G. Davis, S. Mallat and M. Avellaneda, "Greedy adaptive approximations", J. of Constr. Approx. **13** (1997), 57–98.

[7] , R.A. DeVore and V.N. Temlyakov "Some remarks on greedy algorithms", Advances in Computational Mathematics. **5** (1996), 173–187.

[8] L. Devroye, Non-Uniform Random Variate Generation, Springer Verlag, 1986.

[9] D.L. Donoho and I.M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage", Biometrika, **81** (1994), 425–455.

[10] D.L. Donoho and I.M. Johnstone, "Asymptotic minimaxity of wavelet estimators with sampled data", Statistics Sinica, **9**:1 (1999), 1–32.

[11] J. Besag, P.J. Green, D. Higdon and K. Mengersen, "Bayesian computation and stochastic systems" (with discussion), Statistical Science **10** (1995), 3–66).

[12] S. Jaggi, W. Karl, S. Mallat and A. Willsky "High resolution pursuit for feature extraction", Applied and Computational Harmonic Analysis, 1998.

[13] S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries", IEEE Transactions on Signal Processing **41**:12 (1993), 3397–3415.

[14] S. Mallat, A Wavelet Tour of Signal Processing, Academic Press, Boston, MA, 1998.

[15] R.Neff and A. Zakhor "Matching pursuit video coding at very low bit rates". In Proc. Data Compression Conf. 1995.

[16] A.W. Roberts, D.L. Varberg, Convex Functions, Academic Press, New York, 1973.

[17] J. K. Ruanaidh and W. J. Fitzgerald, Numerical Bayesian Methods Applied to Signal Processing, Springer Verlag, New York 1996.

[18] S. Tompaidis "Portfolio compression and projection pursuit", Preprint.

[19] A.Theolis, Computational Signal Processing with Wavelets, Birkhauser, Boston, MA, 1998.

[20] L.Tierney, "Markov chains for exploring posterior distributions", The Annals of Statistics **22**:4, (1994), 1701–1762.

[21] E. T. Whittaker and G. N. Watson, A Course of Modern Analysis, Cambridge University Press, 1927.

[22] M.V. Wickerhauser, Adapted Wavelet Analysis from Theory to Software, A.K. Peters, Wellesley, MA, 1994.

List of Tables

D Tables

Table D.1
Jensen's Inequality with Stopping at RMSE=0.2

Signal	PMP RMSE	Avg RMSE	MP RMSE
f_1	0.109	0.192	0.197
f_2	0.052	0.190	0.181
f_3	0.059	0.190	0.169

Table D.2
Jensen's Inequality with Stopping at RMSE=0.4

Signal	PMP RMSE	Ave. RMSE	MP RMSE
f_1	0.235	0378	0.358
f_2	0.136	0.375	0.281
f_3	0.164	0.379	0.339

Table D.3
Signal to Noise Ratio for Noise Levels $j = 1, \ldots, 4$

NL j	1	2	3	4
SNR l_j	5	2	1.25	1

Table D.4
τ for Various Signals

	Noise Level				
Signal	0	1	2	3	4
f_1	0.31	0.31	0.31	0.32	0.32
f_2	0.31	0.28	0.30	0.27	0.26
f_3	0.28	0.29	0.29	0.27	0.25

Table D.5
Relative Mean Squared Errors in All Cases

Signal	Method P	Method M
$f_{1,1}$	0.231	0.314
$f_{1,2}$	0.320	0.444
$f_{1,3}$	0.366	0.542
$f_{1,4}$	0.462	0.629
$f_{2,1}$	0.215	0.347
$f_{2,2}$	0.343	0.438
$f_{2,3}$	0.381	0.522
$f_{2,4}$	0.434	0.567
$f_{3,1}$	0.245	0.312
$f_{3,2}$	0.350	0.437
$f_{3,3}$	0.408	0.504
$f_{3,4}$	0.478	0.660

*

Table D.6
MP Iterations for $f_{3,4}$

Iteration	IP MP	IP Noise	RMSE	Comment
1	0.455	0.320	0.823	
2	0.427	0.313	0.672	
3	0.347	0.311	0.609	BEST
4	0.358	0.300	0.704	
5	0.333	0.274	0.714	
6	0.340	0.270	0.745	STOP
7	0.293	0.277	0.793	

List of Figures

E Figures

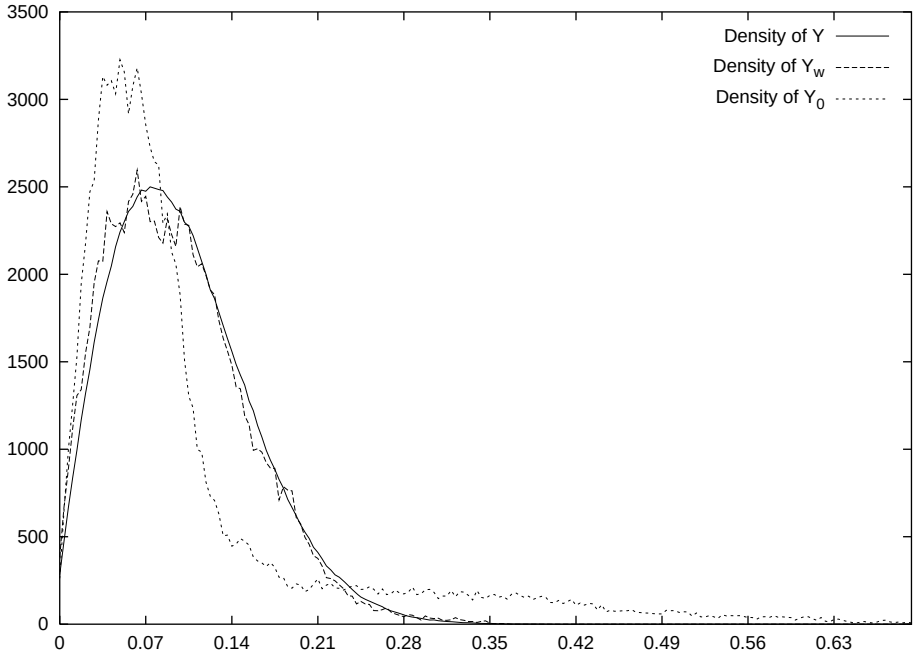

Fig. E.1. Densities of $Y(\gamma)$, $Y_w(\gamma)$, and $Y_0(\gamma) = \left\langle f_{3,1}, g_{(\gamma,\phi)} \right\rangle / \|f_{3,1}\|$

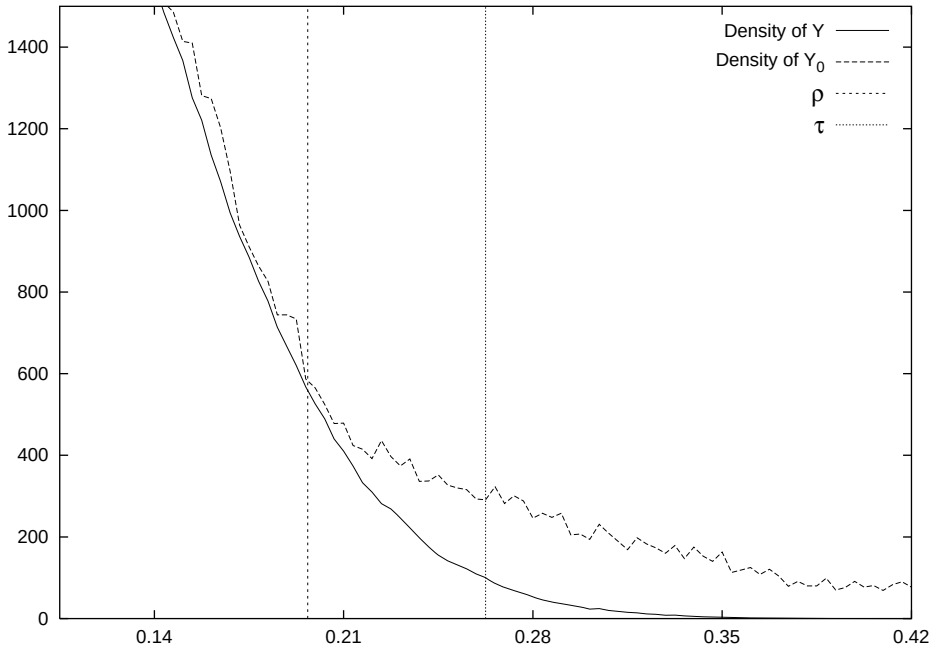

Fig. E.2. Densities of $Y(\gamma)$, $Y_0(\gamma) = \left\langle f_{3,3}, g_{(\gamma,\phi)} \right\rangle / \|f_{3,3}\|$, and ρ and τ

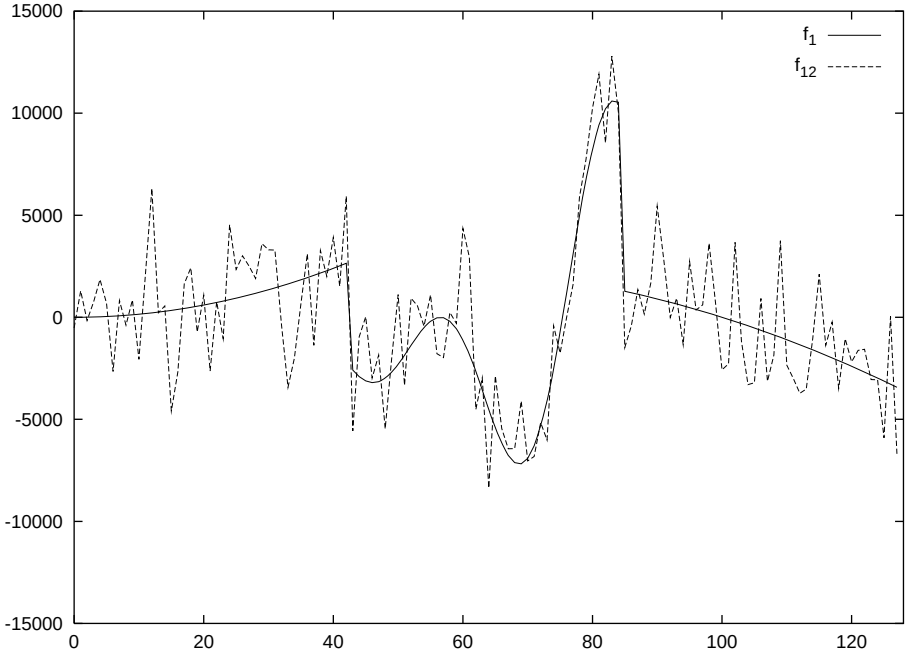

Fig. E.3. f_1 and $f_{1,2}$

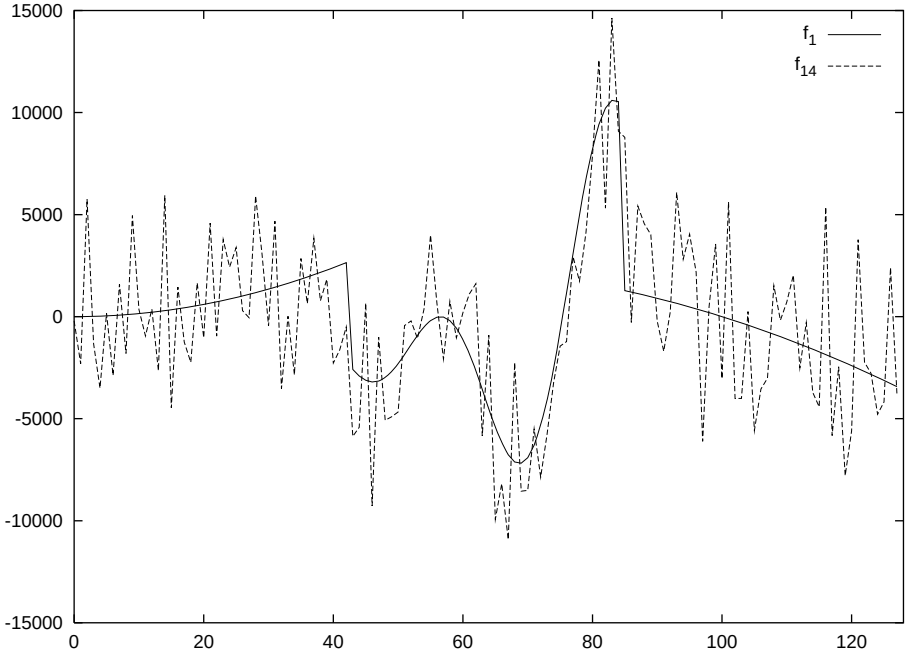

Fig. E.4. f_1 and $f_{1,4}$

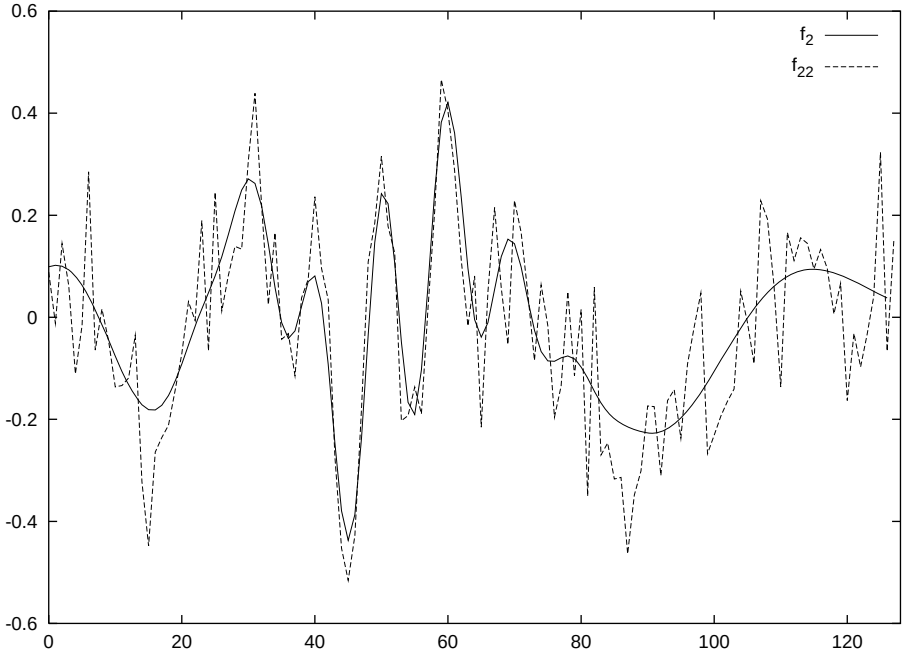

Fig. E.5. f_2 and $f_{2,2}$

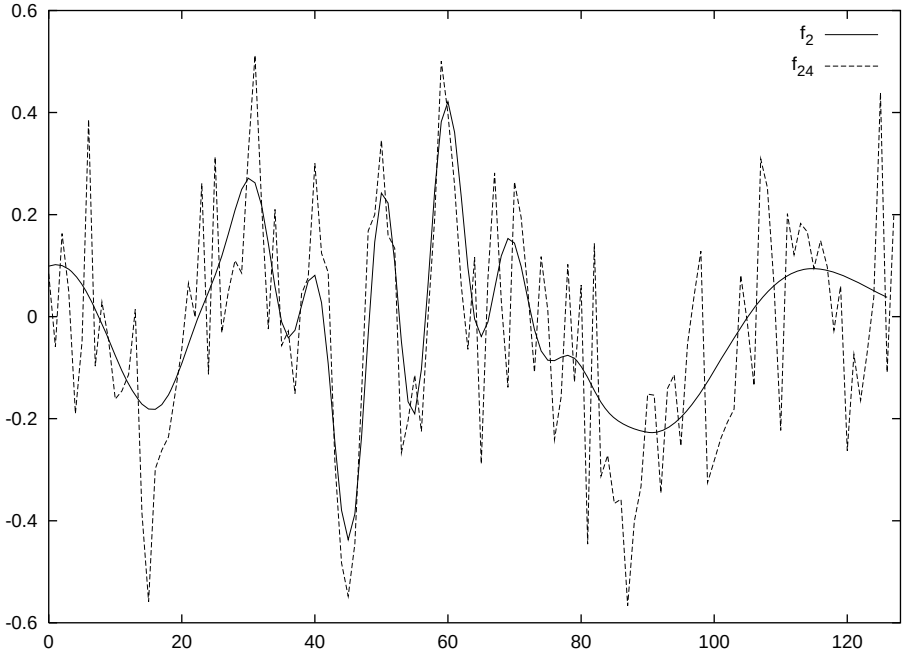

Fig. E.6. f_2 and $f_{2,4}$

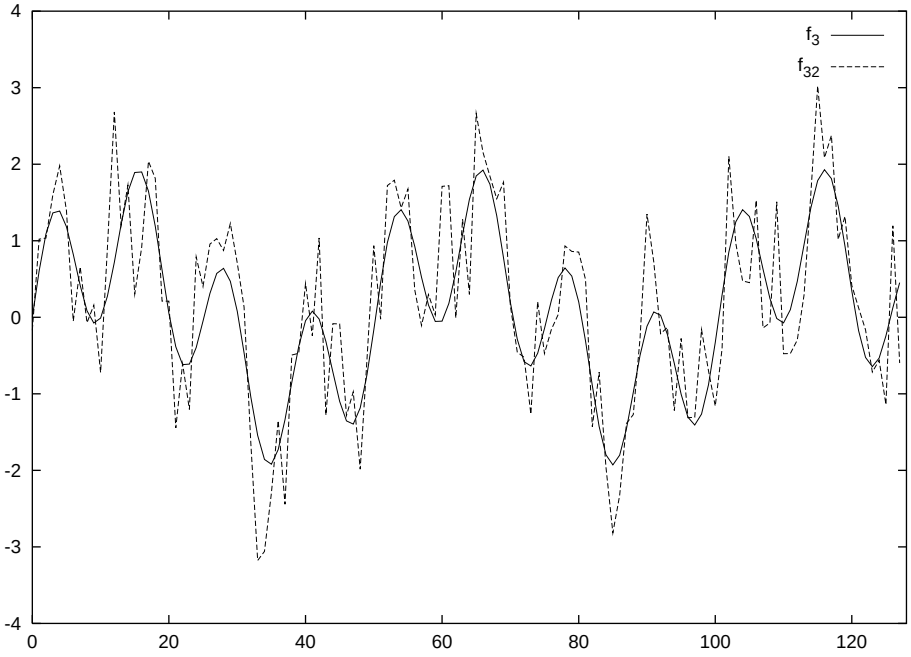

Fig. E.7. f_3 and $f_{3,2}$

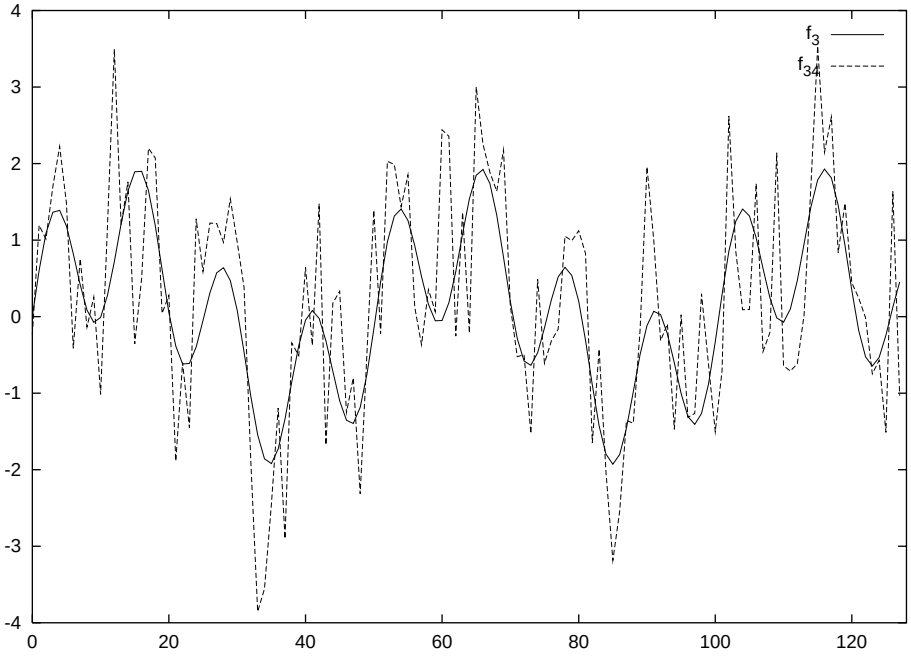

Fig. E.8. f_3 and $f_{3,4}$

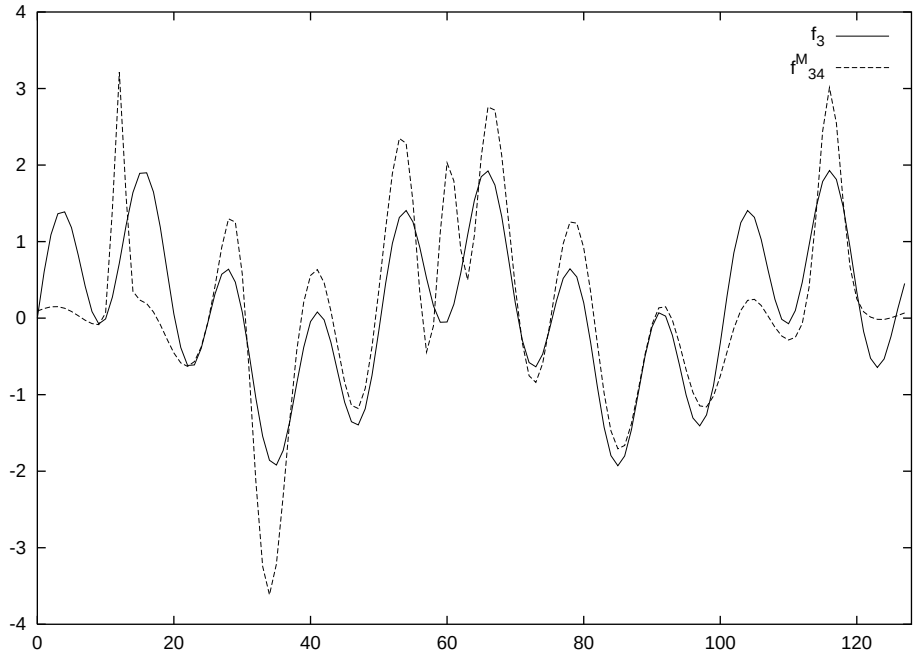

Fig. E.9. f_3 and $f^M_{3,4}$: MP with Automatic Stopping Rule

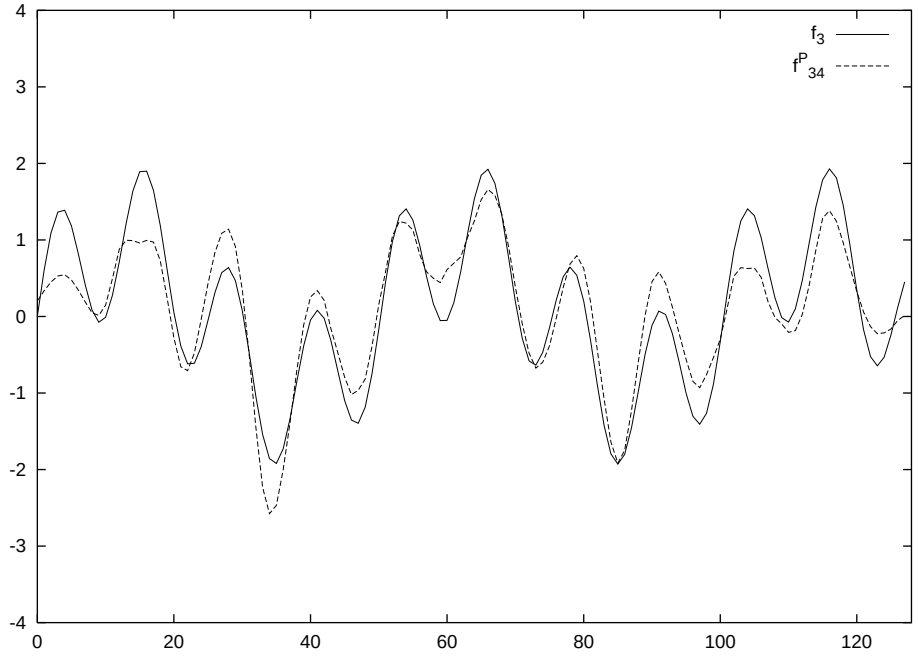

Fig. E.10. f_3 and $f^P_{3,4}$

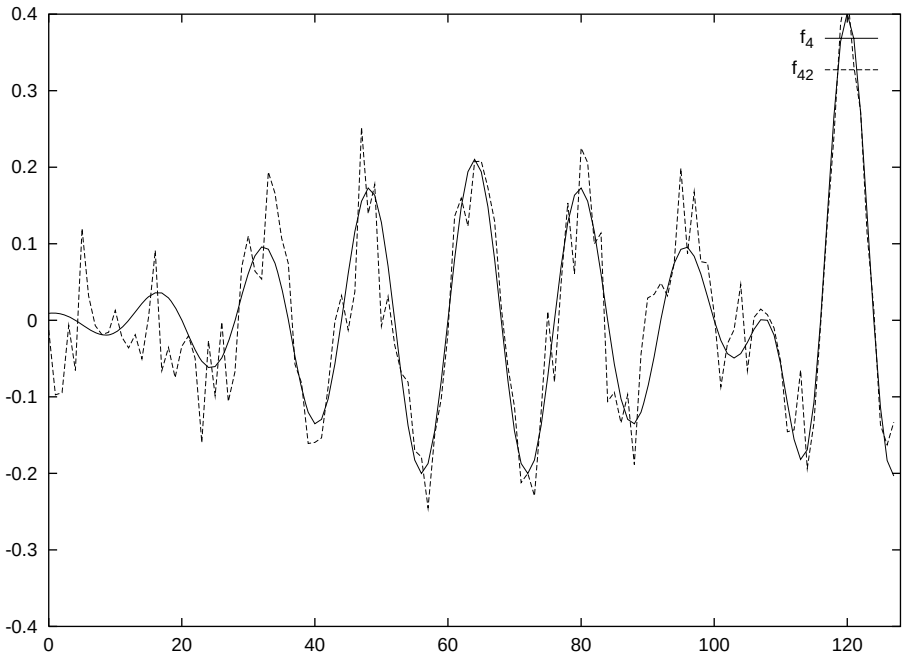

Fig. E.11. f_4 and $f_{4,2}$

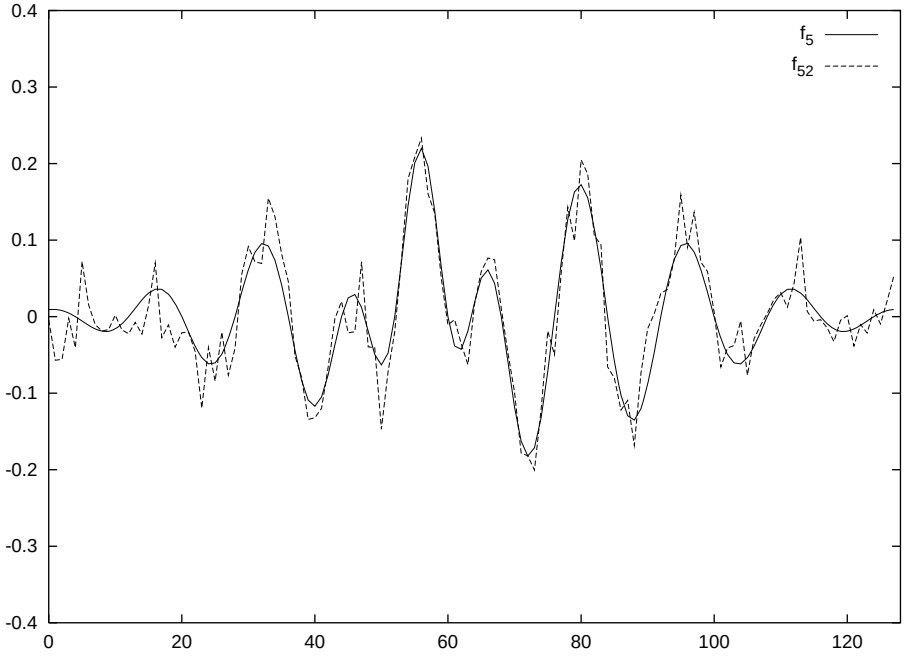

Fig. E.12. f_5 and $f_{5,2}$