10 Time Series Analysis

"It is my feeling that Time ripens all things; with Time all things are revealed; Time is the father of truth." (François Rabelais)

This chapter summarizes the fundamental concepts and tools for analyzing time series data. Time series analysis is a branch of applied mathematics developed mostly in the fields of signal processing and statistics. Contributions to this field, from an astronomical perspective, have predominantly focused on unevenly sampled data, low signal-to-noise data, and heteroscedastic errors. There are more books written about time series analysis than pages in this book and, by necessity, we can only address a few common use cases from contemporary astronomy. Even when limited to astronomical data sets, the diversity of potential applications is enormous. The most common applications range from the detection of variability and periodicity to the treatment of nonperiodic variability and searches for localized events.

Within time-domain data, measurement errors can range from as small as one part in 100,000 (e.g., the photometry from the Kepler mission [32]), to potential events buried in noise with a signal-to-noise ratio per data point of, at best, a few (e.g., searches for gravitational waves using the Laser Interferometric Gravitational Observatory (LIGO) data[1] [1]). Data sets can include many billions of data points, and sample sizes can be in the millions (e.g., the LINEAR data set with 20 million light curves, each with a few hundred measurements [50]). Upcoming surveys, such as Gaia and LSST, will increase existing data sets by large factors; the Gaia satellite will measure about a billion sources about 70 times each during its five-year mission, and the ground-based LSST will obtain about 800 measurements each for about 20 billion sources over its ten years of operation. Scientific utilization of such data sets will include searches for extrasolar planets; tests of stellar astrophysics through studies of variable stars and supernova explosions; distance determination (e.g., using standard candles such as Cepheids, RR Lyrae, and supernovas); and fundamental physics such

[1] LIGO aims to detect gravitational waves, the ripples in the fabric of space and time produced by astrophysical events such as black hole and neutron star collisions, predicted by the general theory of relativity. The two 4-km large detectors in Hanford, Washington and Livingston, Louisiana utilize laser interferometers to measure a fractional change of distance between two mirrors with a precision of about 1 part in 10^{18}.

as tests of general relativity with radio pulsars, cosmological studies with supernovas, and searches for gravitational wave events.

We start with a brief introduction to the main concepts in time series analysis, and then discuss the main tools from the modeling toolkit for time series analysis. Despite being set in the context of time series, many tools and results are readily applicable in other domains, and for this reason our examples will not be strictly limited to time-domain data. Armed with the modeling toolkit, we will then discuss the analysis of periodic time series, search for temporally localized signals, and conclude with a brief discussion of stochastic processes. The main data sets used in this chapter include light curves obtained by the LINEAR survey (§1.5.9) and a variety of simulated examples.

10.1. Main Concepts for Time Series Analysis

The time series discussed here will be limited to two-dimensional scalar data sets: pairs of random variables, $(t_1, y_1), \ldots, (t_N, y_N)$, with no assumptions about the sampling of the time coordinate t (with the exception of the so-called arrival time data that consists of detection times for individual photons, discussed in §10.3.5). In many ways, analysis methods discussed here are closely related to the parameter estimation and model selection problems discussed in the context of regression in chapter 8; when the temporal variable t is replaced by x, this connection becomes more obvious. Nevertheless, there are some important differences encountered in time series analysis, such as models that have a sense of the directionality of time in them (discussed in §10.5.3). Unlike regression problems where different y measurements are typically treated as independent random variables, in such models the value of y_{i+1} directly depends on the preceding value y_i.

The main tasks of time series analysis are (1) to characterize the presumed temporal correlation between different values of y, including its significance, and (2) to forecast (predict) future values of y. In many astronomical cases, the characterization of the underlying physical processes that produced the data, which is typically addressed by learning parameters for a model, is the key goal. For example, analysis of a light curve can readily differentiate between pulsating and eclipsing variable stars. Good examples of the second task are solar activity forecasting, or prediction of the time and place of a potential asteroid impact on Earth.

10.1.1. Is My Time Series Just Noise?

Given a time series, we often first want to determine whether we have detected variability, irrespective of the details of the underlying process. This is equivalent to asking whether the data are consistent with the null hypothesis described by a model consisting of a constant signal plus measurement noise.

From the viewpoint of classical (frequentist) statistics, this question can be treated as a case of hypothesis testing: What is the probability that we would obtain our data by chance if the null hypothesis of no variability were correct? If the errors are known and Gaussian, we can simply compute χ^2 and the corresponding p values. If the errors are unknown, or non-Gaussian, the modeling and model selection tools, such as those introduced in chapter 5 for treating exponential noise or outliers, can be used instead.

Consider a simple example of $y(t) = A\sin(\omega t)$ sampled by $N \sim 100$ data points with homoscedastic Gaussian errors with standard deviation σ. The variance of a well-sampled time series given by this model is $V = \sigma^2 + A^2/2$. For a model with $A = 0$, $\chi^2_{\rm dof} = N^{-1}\sum_j (y_j/\sigma)^2 \sim V/\sigma^2$. When $A = 0$ is true, the $\chi^2_{\rm dof}$ has an expectation value of 1 and a standard deviation of $\sqrt{2/N}$. Therefore, if variability is present (i.e., $|A| > 0$), the computed $\chi^2_{\rm dof}$ will be larger than its expected value of 1. The probability that $\chi^2_{\rm dof} > 1 + 3\sqrt{2/N}$ is about 1 in 1000. If this false-positive rate is acceptable (recall §4.6; for example, if the expected fraction of variable stars in a sample is 1%, this false-positive rate will result in a sample contamination rate of ∼10%), then the minimum detectable amplitude is $A > 2.9\sigma/N^{1/4}$ (derived from $V/\sigma^2 = 1 + 3\sqrt{2/N}$). For example, for $N = 100$ data points, the minimum detectable amplitude is $A = 0.92\sigma$, and $A = 0.52\sigma$ for $N = 1000$. However, we will see that in all cases of specific models, our ability to discover variability is *greatly improved* compared to this simple $\chi^2_{\rm dof}$ selection. For illustration, for the single harmonic model, the minimum detectable variability levels for the false-positive rate of 1 in 1000 are $A = 0.42\sigma$ for $N = 100$ and $A = 0.13\sigma$ for $N = 1000$ (derived using $\sigma_A = \sigma\sqrt{2/N}$; see eq. 10.39). We will also see, in the case of periodic models, that such a simple harmonic fit performs even better than what we might expect a priori (i.e., even in cases of much more complex underlying variations).

This improvement in ability to detect a signal using a model is not limited to periodic variability—this is a general feature of model fitting (sometimes called *matched filter* extraction). Within the Bayesian framework, we cannot even begin our analysis without specifying an alternative model to the constant signal model. If underlying variability is not periodic, it can be roughly divided into two other families: stochastic variability, where variability is always there but the changes are not predictable for an indefinite period (e.g., quasar variability), and temporally localized events such as bursts (e.g., flares from stars, supernova explosions, gamma-ray bursts, or gravitational wave events). The various tools and methods to perform such time series analysis are discussed in the next section.

10.2. Modeling Toolkit for Time Series Analysis

The main tools for time series analysis belong to either the time domain or the frequency domain. Many of the tools and methods discussed in earlier chapters play a prominent role in the analysis of time series data. In this section, we first revisit methods introduced earlier (mostly applicable to the time-domain analysis) and discuss parameter estimation, model selection, and classification in the context of time series analysis. We then extend this toolkit by introducing tools for analysis in the frequency domain, such as Fourier analysis, discrete Fourier transform, wavelet analysis, and digital filtering. Nondeterministic (stochastic) time series are briefly discussed in §10.5.

10.2.1. Parameter Estimation, Model Selection, and Classification for Time Series Data

Detection of a signal, whatever it may be, is essentially a hypothesis testing or model selection problem. The quantitative description of a signal belongs to parameter estimation and regression problems. Once such a description is available for a set of time

series data (e.g., astronomical sources from families with distinctive light curves), their classification utilizes essentially the same methods as discussed in the preceding chapter.

In general, we will fit a model to a set of N data points $(t_1, y_1), \ldots, (t_N, y_N)$, $j = 1, \ldots, N$ with known errors for y,

$$y_j(t_j) = \sum_{m=1}^{M} \beta_m \, T_m(t_j|\boldsymbol{\theta}_m) + \epsilon_j, \tag{10.1}$$

where the functions $T_m(t|\boldsymbol{\theta}_m)$ need not be periodic, nor do the times t_j need to be evenly sampled. As before, the vector $\boldsymbol{\theta}_m$ contains model parameters that describe each $T_m(t)$ (here we use the symbol "|" to mean "given parameters $\boldsymbol{\theta}_m$," and not in the sense of a conditional pdf). Common deterministic models for the underlying process that generates data include $T(t) = \sin(\omega t)$ and $T(t) = \exp(-\alpha t)$, where the frequency ω and decay rate α are model parameters to be estimated from data. Another important model is the so-called "chirp signal," $T(t) = \sin(\phi + \omega t + \alpha t^2)$. In eq. 10.1, ϵ stands for noise, which is typically described by heteroscedastic Gaussian errors with zero mean and parametrized by known σ_j. Note that in this chapter we have changed the index for data values from i to j because we will frequently encounter the imaginary unit $i = \sqrt{-1}$.

Finding whether data favor such a model over the simplest possibility of no variability ($y(t)$=constant plus noise) is no different from model selection problems discussed earlier, and can be addressed via the Bayesian model odds ratio, or approximately using AIC and BIC criteria (see §5.4). Given a quantitative description of time series $y(t)$, the best-fit estimates of model parameters $\boldsymbol{\theta}_m$ can then be used as attributes for various supervised and unsupervised classification methods (possibly with additional attributes that are not extracted from the analyzed time series).

Depending on the amount of data, the noise behavior (and our understanding of it), sampling, and the complexity of a specific model, such analyses can range from nearly trivial to quite complex and computationally intensive. Despite this diversity, there are only a few new concepts needed for the analysis that were not introduced in earlier chapters.

10.2.2. Fourier Analysis

Fourier analysis plays a major role in the analysis of time series data. In Fourier analysis, general functions are represented or approximated by integrals or sums of simpler trigonometric functions. As first shown in 1822 by Fourier himself in his analysis of heat transfer, this representation often greatly simplifies analysis. Figure 10.1 illustrates how an RR Lyrae light curve can be approximated by a sum of sinusoids (details are discussed in §10.2.3). The more terms that are included in the sum, the better is the resulting approximation. For periodic functions, such as periodic light curves in astronomy, it is often true that a relatively small number of terms (less than 10) suffices to reach an approximation precision level similar to the measurement precision.

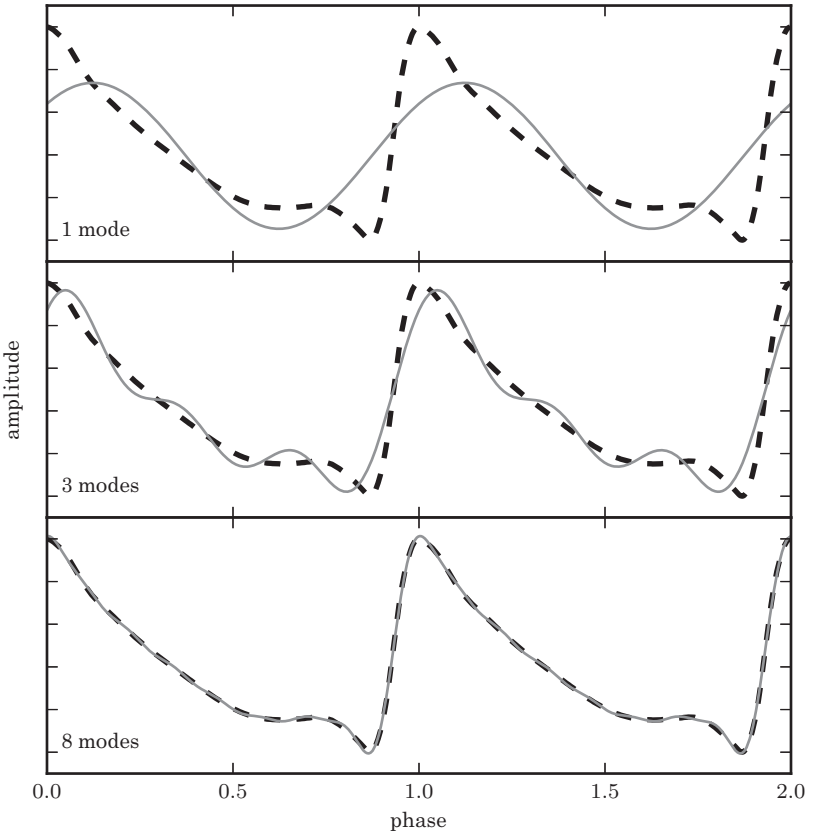

Figure 10.1. An example of a truncated Fourier representation of an RR Lyrae light curve. The thick dashed line shows the true curve; the gray lines show the approximation based on 1, 3, and 8 Fourier modes (sinusoids).

The most useful applications of Fourier analysis include convolution and deconvolution, filtering, correlation and autocorrelation, and power spectrum estimation (practical examples are interspersed throughout this chapter). The use of these methods is by no means limited to time series data; for example, they are often used to analyze spectral data or in characterizing the distributions of points. When the data are evenly sampled and the signal-to-noise ratio is high, Fourier analysis can be a powerful tool. When the noise is high compared to the signal, or the signal has a complex shape (i.e., it is not a simple harmonic function), a probabilistic treatment (e.g., Bayesian analysis) offers substantial improvements, and for irregularly (unevenly) sampled data probabilistic treatment becomes essential. For these reasons, in the analysis of astronomical time series, which are often irregularly sampled with heteroscedastic errors, Fourier analysis is often replaced by other methods (such as the periodogram analysis discussed in §10.3.1). Nevertheless, most of the main concepts introduced in Fourier analysis carry over to those other methods and thus Fourier analysis is an indispensable tool when analyzing time series.

A periodic signal such as the one in figure 10.1 can be decomposed into Fourier modes using the fast Fourier transform algorithm available in `scipy.fftpack`:

```
>>> from scipy import fftpack
>>> from astroML.datasets import fetch_rrlyrae_templates

>>> templates = fetch_rrlyrae_templates()
>>> x, y = templates['115r'].T

>>> k = 3  # reconstruct using 3 frequencies
>>> y_fft = fftpack.fft(y)  # compute the Fourier transform
>>> y_fft[k + 1:-k] = 0  # zero-out frequencies higher than k
>>> y_fit = fftpack.ifft(y_fft).real  # reconstruct using k modes
```

The resulting array is the reconstruction with `k` modes: this procedure was used to generate figure 10.1. For more information on the fast Fourier transform, see §10.2.3 and appendix E.

Numerous books about Fourier analysis are readily available. An excellent concise summary of the elementary properties of the Fourier transform is available in NumRec (see also the appendix of Greg05 for a very illustrative summary). Here, we will briefly summarize the main features of Fourier analysis and limit our discussion to the concepts used in the rest of this chapter.

The Fourier transform of function $h(t)$ is defined as

$$H(f) = \int_{-\infty}^{\infty} h(t)\ \exp(-i2\pi ft)\, dt, \tag{10.2}$$

with inverse transformation

$$h(t) = \int_{-\infty}^{\infty} H(f)\ \exp(i2\pi ft)\, df, \tag{10.3}$$

where t is time and f is frequency (for time in seconds, the unit for frequency is hertz, or Hz; the units for $H(f)$ are the product of the units for $h(t)$ and inverse hertz; note that in this chapter f is *not* a symbol for the empirical pdf as in the preceding chapters). We note that NumRec and most physics textbooks define the argument of the exponential function in the inverse transform with the minus sign; the above definitions are consistent with SciPy convention and most engineering literature. Another notational detail is that angular frequency, $\omega = 2\pi f$, is often used instead of frequency (the unit for ω is radians per second) and the extra factor of 2π due to the change of variables is absorbed into either $h(t)$ or $H(f)$, depending on convention.

For a real function $h(t)$, $H(f)$ is in general a complex function.[2] In the special case when $h(t)$ is an even function such that $h(-t) = h(t)$, $H(f)$ is real and even as well. For example, the Fourier transform of a pdf of a zero-mean Gaussian $\mathcal{N}(0, \sigma)$ in the time domain is a Gaussian $H(f) = \exp(-2\pi^2\sigma^2 f^2)$ in the frequency domain. When the time axis of an arbitrary function $h(t)$ is shifted by Δt, then the Fourier transform of $h(t + \Delta t)$ is

$$\int_{-\infty}^{\infty} h(t + \Delta t)\ \exp(-i2\pi ft)\, dt = H(f)\ \exp(i2\pi f \Delta t), \tag{10.4}$$

[2] Recall Euler's formula, $\exp(ix) = \cos x + i \sin x$.

where $H(f)$ is given by eq. 10.2. Therefore, the Fourier transform of a Gaussian $\mathcal{N}(\mu, \sigma)$ is

$$H_{\rm Gauss}(f) = \exp(-2\pi^2\sigma^2 f^2)\,\left[\cos(2\pi f\mu) + i\,\sin(2\pi f\mu)\right]. \tag{10.5}$$

This result should not be confused with a Fourier transform of Gaussian noise with time-independent variance σ^2, which is simply a constant. This is known as *white noise* as there is no frequency dependence (also known as *thermal noise* or *Johnson's noise*). The cases known as *pink noise* and *red noise* are discussed in §10.5.

An important quantity in time series analysis is the one-sided *power spectral density* (PSD) function (or *power spectrum*) defined for $0 \le f < \infty$ as

$$\mathrm{PSD}(f) \equiv |H(f)|^2 + |H(-f)|^2. \tag{10.6}$$

The PSD gives the amount of power contained in the frequency interval between f and $f + df$ (i.e., the PSD is a quantitative statement about the "importance" of each frequency mode). For example, when $h(t) = \sin(2\pi\, t/T)$, $P(f)$ is a δ function centered on $f = 1/T$.

The total power is the same whether computed in the frequency or in the time domain:

$$P_{\rm tot} \equiv \int_0^\infty \mathrm{PSD}(f)\, df = \int_{-\infty}^{\infty} |h(t)|^2\, dt. \tag{10.7}$$

This result is known as *Parseval's theorem*.

Convolution Theorem

Another important result is the convolution theorem: A convolution of two functions $a(t)$ and $b(t)$ is given by (we already introduced it as eq. 3.44)

$$(a \star b)(t) \equiv \int_{-\infty}^{\infty} a(t')b(t - t')\, dt'. \tag{10.8}$$

Convolution is an unavoidable result of the measurement process because the measurement resolution, whether in time, spectral, spatial, or any other domain, is never infinite. For example, in astronomical imaging the true intensity distribution on the sky is convolved with the atmospheric seeing for ground-based imaging, or the telescope diffraction pattern for space-based imaging (radio astronomers use the term *beam convolution*). In the above equation, the function a can be thought of as the "convolving pattern" of the measuring apparatus, and the function b is the signal. In practice, we measure the convolved (or smoothed) version of our signal, $[a * b](t)$, and seek to uncover the original signal b using the presumably known a.

The convolution theorem states that if $h = a \star b$, then the Fourier transforms of h, a, and b are related by their pointwise products:

$$H(f) = A(f)B(f). \tag{10.9}$$

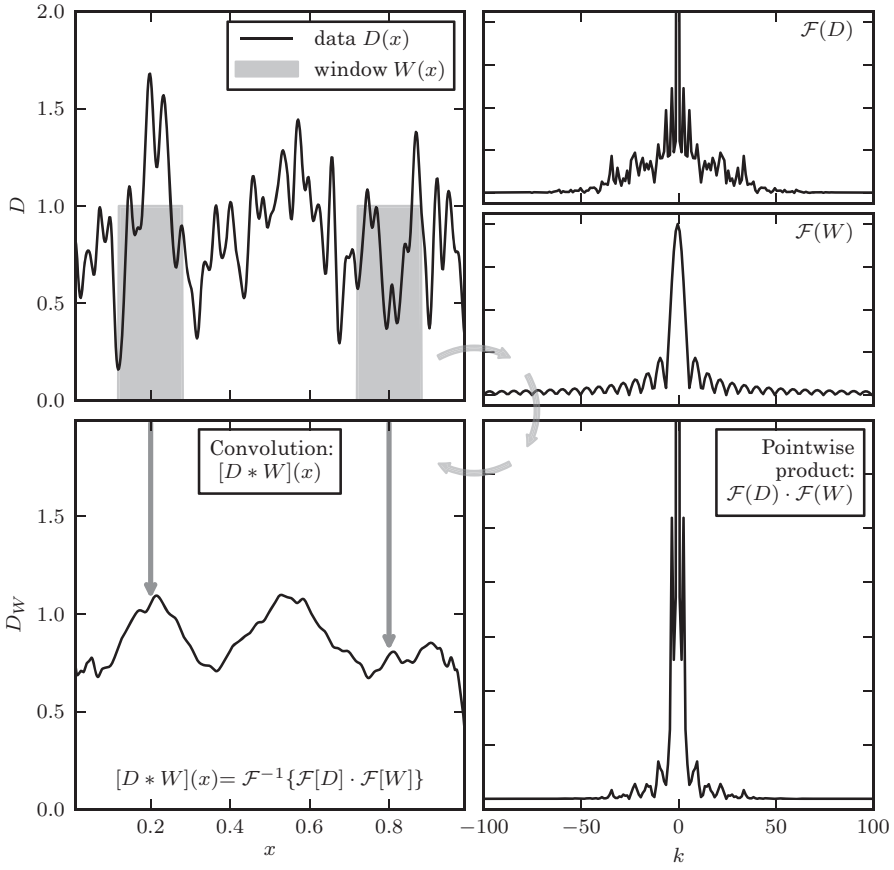

Figure 10.2. A schematic of how the convolution of two functions works. The top-left panel shows simulated data (black line); this time series is convolved with a top-hat function (gray boxes); see eq. 10.8. The top-right panels show the Fourier transform of the data and the window function. These can be multiplied together (bottom-right panel) and inverse transformed to find the convolution (bottom-left panel), which amounts to integrating the data over copies of the window at all locations. The result in the bottom-left panel can be viewed as the signal shown in the top-left panel smoothed with the window (top-hat) function.

Thus a convolution of two functions is transformed into a simple multiplication of the associated Fourier representations. Therefore, to obtain b, we can simply take the inverse Fourier transform of the ratio $H(f)/A(f)$. In the absence of noise, this operation is exact. The convolution theorem is a very practical result; we shall consider further examples of its usefulness below.

A schematic representation of the convolution theorem is shown in figure 10.2. Note that we could have started from the convolved function shown in the bottom-left panel and uncovered the underlying signal shown in the top-left panel. When noise is present we can, however, never fully recover all the detail in the signal shape. The methods for the deconvolution of noisy data are many and we shall review a few of them in §10.2.5.

10.2.3. Discrete Fourier Transform

In practice, data are always discretely sampled. When the spacing of the time interval is constant, the discrete Fourier transform is a powerful tool. In astronomy, temporal data are rarely sampled with uniform spacing, though we note that LIGO data are a good counterexample (an example of LIGO data is shown and discussed in figure 10.6 further below). Nevertheless, uniformly sampled data are a good place to start, because of the very fast algorithms available for this situation, and because the primary concepts also extend to unevenly sampled data.

When computing the Fourier transform for discretely and uniformly sampled data, the Fourier integrals from eqs. 10.2 and 10.3 are translated to sums. Let us assume that we have a continuous real function $h(t)$ which is sampled at N equal intervals $h_j = h(t_j)$ with $t_j \equiv t_0 + j\Delta t$, $j = 0, \ldots, (N-1)$, where the sampling interval Δt and the duration of data taking T are related via $T = N\Delta t$ (the binning could have been done by the measuring apparatus, e.g., CCD imaging, or during the data analysis).

The discrete Fourier transform of the vector of values h_j is a complex vector of length N defined by

$$H_k = \sum_{j=0}^{N-1} h_j \exp[-i2\pi jk/N], \tag{10.10}$$

where $k = 0, \ldots, (N-1)$. The corresponding inverse discrete Fourier transform is defined by

$$h_j = \frac{1}{N} \sum_{k=0}^{N-1} H_k \exp[i2\pi jk/N], \tag{10.11}$$

where $j = 0, \ldots, (N-1)$. Unlike the continuous transforms, here the units for H_k are the same as the units for h_j. Given H_k, we can represent the function described by h_j as a sum of sinusoids, as was done in figure 10.1.

The Nyquist Sampling Theorem

What is the relationship between the transforms defined by eqs. 10.2 and 10.3, where integration limits extend to infinity, and the discrete transforms given by eqs. 10.10 and 10.11, where sums extend over sampled data? For example, can we estimate the PSD given by eq. 10.6 using a discrete Fourier transform? The answer to these questions is provided by the *Nyquist sampling theorem* (also known as the *Nyquist–Shannon theorem*, and as the *cardinal theorem of interpolation theory*), an important result developed within the context of signal processing.

Let us define $h(t)$ to be *band limited* if $H(f) = 0$ for $|f| > f_c$, where f_c is the band limit, or the Nyquist critical frequency. If $h(t)$ is band limited, then there is some "resolution" limit in t space, $t_c = 1/(2f_c)$ below which $h(t)$ appears "smooth." When $h(t)$ is band limited, then according to the Nyquist sampling theorem we can *exactly reconstruct* $h(t)$ *from evenly sampled data when* $\Delta t \leq t_c$, as

$$h(t) = \frac{\Delta t}{t_c} \sum_{k=-\infty}^{k=\infty} h_k \frac{\sin[2\pi f_c(t - k\Delta t)]}{2\pi f_c(t - k\Delta t)}. \tag{10.12}$$

This result is known as the *Whittaker–Shannon*, or often just *Shannon*, interpolation formula (or *sinc-shifting* formula). Note that the summation goes to infinity, but also that the term multiplying h_k vanishes for large values of $|t - k\Delta t|$. For example, $h(t) = \sin(2\pi t/P)$; has a period P and is band limited with $f_c = 1/P$. If it is sampled with Δt not larger than $P/2$, it can be fully reconstructed at any t (it is important to note that this entire discussion assumes that there is no noise associated with sampled values h_j). On the other hand, when the sampled function $h(t)$ is not band limited, or when the sampling rate is not sufficient (i.e., $\Delta t > t_c$), an effect called *aliasing* prevents us from exactly reconstructing $h(t)$ (see figure 10.3). In such a case, all of the power spectral density from frequencies $|f| > f_c$ is aliased (falsely transferred) into the $-f_c < f < f_c$ range. The aliasing can be thought of as inability to resolve details in a time series at a finer detail than that set by f_c. The aliasing effect can be recognized if the Fourier transform is nonzero at $|f| = 1/(2\Delta t)$, as is shown in the lower panels of figure 10.3.

Therefore, the discrete Fourier transform is a good estimate of the true Fourier transform for properly sampled band-limited functions. Eqs. 10.10 and 10.11 can be related to eqs. 10.2 and 10.3 by approximating $h(t)$ as constant outside the sampled range of t, and assuming $H(f) = 0$ for $|f| > 1/(2\Delta t)$. In particular,

$$|H(f_k)| \approx \Delta t\, |H_k|, \tag{10.13}$$

where $f_k = k/(N\Delta t)$ for $k \leq N/2$ and $f_k = (k - N)/(N\Delta t)$ for $k \geq N/2$ (see appendix E for a more detailed discussion of this result). The discrete analog of eq. 10.6 can now be written as

$$\mathrm{PSD}(f_k) = (\Delta t)^2 \left(|H_k|^2 + |H_{N-k}|^2\right), \tag{10.14}$$

and explicitly

$$\mathrm{PSD}(f_k) = 2\left(\frac{T}{N}\right)^2 \left[\left(\sum_{j=0}^{N-1} h_j \cos(2\pi f_k t_j)\right)^2 + \left(\sum_{j=0}^{N-1} h_j \sin(2\pi f_k t_j)\right)^2\right]. \tag{10.15}$$

Using these results, we can estimate the Fourier transform and PSD of any discretely and evenly sampled function. As discussed in §10.3.1, these results are strictly true only for noiseless data (although in practice they are often applied, sometimes incorrectly, to noisy data).

The Window Function

Figure 10.3 shows the relationship between sampling and the *window function*: the sampling window function in the time domain can be expressed as the sum of delta functions placed at sampled observation times. In this case the observations are regularly spaced. The Fourier transform of a set of delta functions with spacing Δt is another set of delta functions with spacing $1/\Delta t$; this result is at the core of the Nyquist sampling theorem. By the convolution theorem, pointwise multiplication of this sampling window with the data is equivalent to the convolution of their Fourier representations, as seen in the right-hand panels.

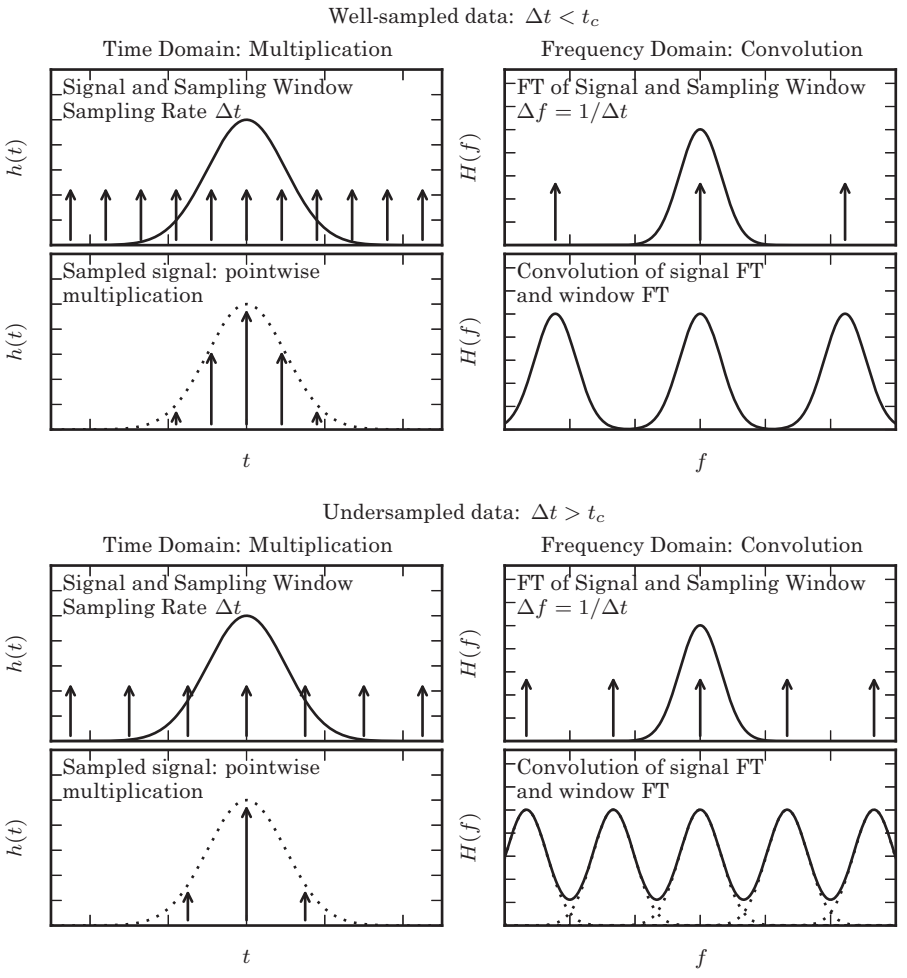

Figure 10.3. A visualization of aliasing in the Fourier transform. In each set of four panels, the top-left panel shows a signal and a regular sampling function, the top-right panel shows the Fourier transform of the signal and sampling function, the bottom-left panel shows the sampled data, and the bottom-right panel shows the convolution of the Fourier-space representations (cf. figure 10.2). In the top four panels, the data are well sampled, and there is little to no aliasing. In the bottom panels, the data are not well sampled (the spacing between two data points is larger) which leads to aliasing, as seen in the overlap of the convolved Fourier transforms (figure adapted from Greg05).

When data are nonuniformly sampled, the impact of sampling can be understood using the same framework. The sampling window is the sum of delta functions, but because the delta functions are not regularly spaced, the Fourier transform is a more complicated, and in general complex, function of f. The PSD can be computed using the discrete Fourier transform by constructing a fine grid of times and setting the window function to one at the sampled times and zero otherwise. The resulting PSD is called the *spectral window function*, and models how the Fourier-space signal is affected by the sampling. As discussed in detail in [19], the observed PSD is a convolution of the true underlying PSD and this spectral window function.

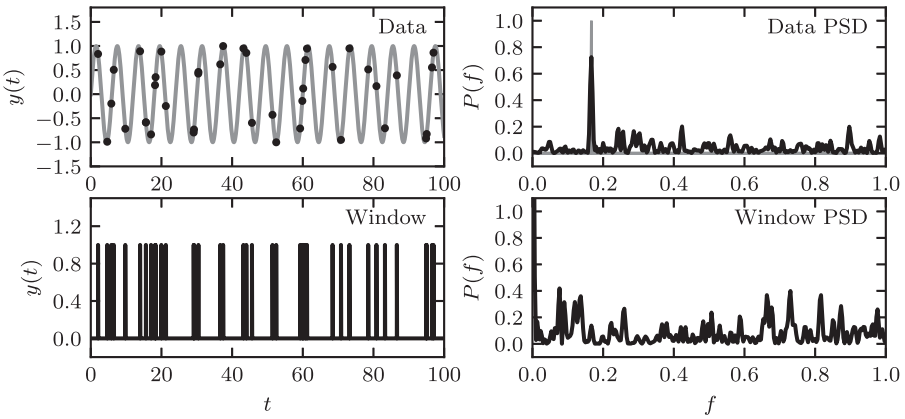

Figure 10.4. An illustration of the impact of a sampling window function of resulting PSD. The top-left panel shows a simulated data set with 40 points drawn from the function $y(t|P) = \sin t$ (i.e., $f = 1/(2\pi) \sim 0.16$). The sampling is random, and illustrated by the vertical lines in the bottom-left panel. The PSD of sampling times, or spectral window, is shown in the bottom-right panel. The PSD computed for the data set from the top-left panel is shown in the top-right panel; it is equal to a convolution of the single peak (shaded in gray) with the window PSD shown in the bottom-right panel (e.g., the peak at $f \sim 0.42$ in the top-right panel can be traced to a peak at $f \sim 0.26$ in the bottom-right panel).

An example of an irregular sampling window is shown in figure 10.4: here the true Fourier transform of the sinusoidal data is a localized spike. The Fourier transform of the function viewed through the sampling window is a convolution of the true FT and the FT of the window function. This type of analysis of the spectral window function can be a convenient way to summarize the sampling properties of a given data set, and can be used to understand aliasing properties as well; see [23].

The Fast Fourier Transform

The *fast Fourier transform* (FFT) is an algorithm for computing discrete Fourier transforms in $\mathcal{O}(N \log N)$ time, rather than $\mathcal{O}(N^2)$, using a naive implementation. The algorithmic details for the FFT can be found in NumRec. The speed of FFT makes it a widespread tool in the analysis of evenly sampled, high signal-to-noise ratio, time series data.

The FFT and various related tools are available in Python through the submodules `numpy.fft` and `scipy.fftpack`:

```
>>> import numpy as np
>>> from scipy import fftpack

>>> x = np.random.normal(size=1000)  # white noise
>>> x_fft = fftpack.fft(x)  # Fourier transform
>>> x2 = fftpack.ifft(x_fft)  # inverse: x2=x to numerical precision
```

For more detailed examples of using the FFT in practice, see appendix E or the source code of many of the figures in this chapter.

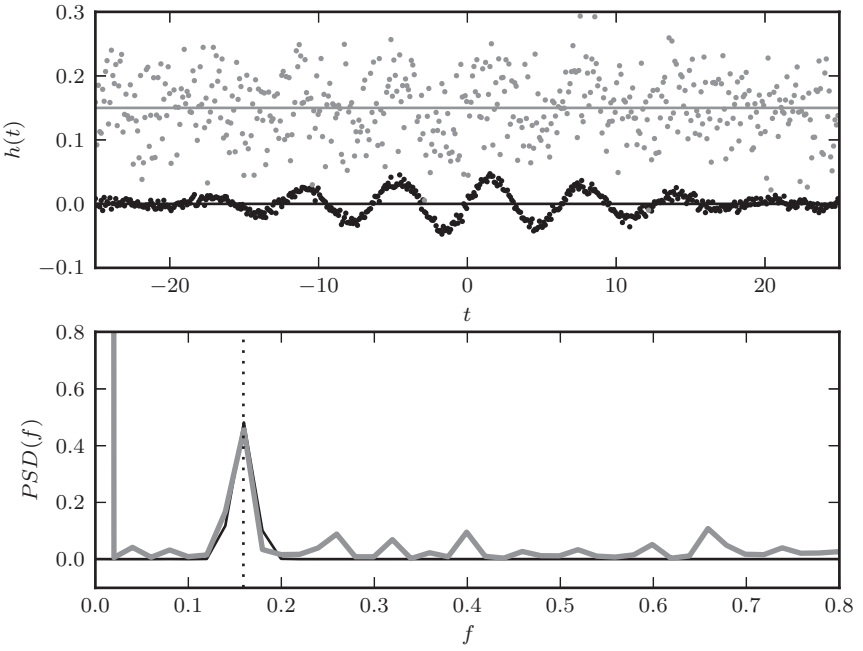

Figure 10.5. The discrete Fourier transform (bottom panel) for two noisy data sets shown in the top panel. For 512 evenly sampled times t (Δt=0.977), points are drawn from $h(t) = a + \sin(t)\, G(t)$, where $G(t)$ is a Gaussian $\mathcal{N}(\mu = 0, \sigma = 10)$. Gaussian noise with $\sigma = 0.05$ (top data set) and 0.005 (bottom data set) is added to signal $h(t)$. The value of the offset a is 0.15 and 0, respectively. The discrete Fourier transform is computed as described in §10.2.3. For both noise realizations, the correct frequency $f = (2\pi)^{-1} \approx 0.159$ is easily discernible in the bottom panel. Note that the height of peaks is the same for both noise realizations. The large value of $|H(f = 0)|$ for data with larger noise is due to the vertical offset.

An example of such analysis is shown in figure 10.5 for a function with a single dominant frequency: a sine wave whose amplitude is modulated by a Gaussian. The figure shows the results in the presence of noise, for two different noise levels. For the high noise level, the periodic signal is hard to recognize in the time domain. Nevertheless, the dominant frequency is easily discernible in the bottom panel for both noise realizations. One curious property is that the expected value of the peak heights are the same for both noise realizations. Another curious feature of the discrete PSD given by eq. 10.14 is that its precision as an estimator of the PSD given by eq. 10.6 does not depend on the number of data values, N (i.e., the discrete PSD is an inconsistent estimator of the true PSD). For example, if N is doubled by doubling the data-taking interval T, then the resulting discrete PSD is defined at twice as many frequencies, but the value of PSD at a given frequency does not change. Alternatively, if N is doubled by doubling the sampling rate such that $\Delta t \to \Delta t/2$, then the Nyquist frequency increases by a factor of 2 to accommodate twice as many points, again without a change in PSD at a given frequency. We shall discuss PSD peaks in more detail in §10.3.1, when we generalize the concept of the PSD to unevenly sampled data.

The discrete Fourier transform can be a powerful tool even when data are not periodic. A good example is estimating power spectrum for noise that is not white.

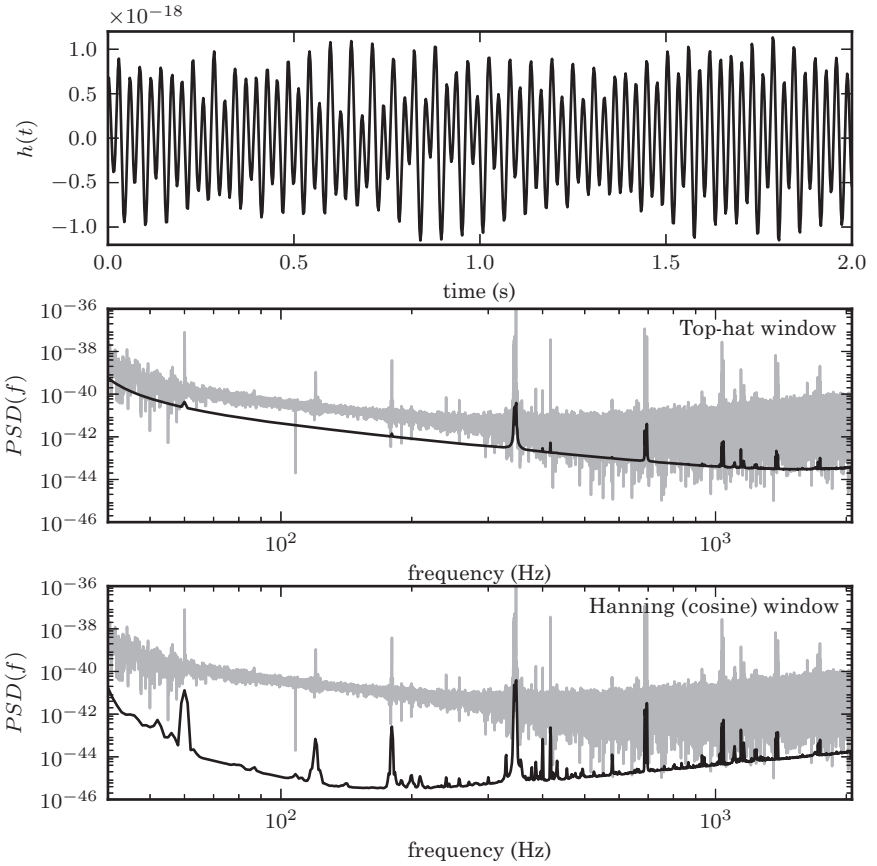

Figure 10.6. LIGO data and its noise power spectrum. The upper panel shows a 2-second-long stretch of data (∼8000 points; essentially noise without signal) from LIGO Hanford. The middle and bottom panels show the power spectral density computed for 2048 seconds of data, sampled at 4096 Hz (∼8 million data values). The gray line shows the PSD computed using a naive FFT approach; the dark line uses Welch's method of overlapping windows to smooth noise [62]; the middle panel uses a 1-second-wide top-hat window and the bottom panel the so-called Hanning (cosine) window with the same width.

In figure 10.6 we compute the noise power spectrum for a stream of time series data from LIGO. The measurement noise is far from white: it has a minimum at frequencies of a few hundred hertz (the minimum level is related to the number of photons traveling through the interferometers), and increases rapidly at smaller frequencies due to seismic effects, and at higher frequencies due to a number of instrumental effects. The predicted signal strengths are at best a few times stronger than the noise level and thus precise noise characterization is a prerequisite for robust detection of gravitational waves.

For noisy data with many samples, more sophisticated FFT-based methods can be used to improve the signal-to-noise ratio of the resulting PSD, at the expense of frequency resolution. One well-known method is *Welch's method* [62], which computes multiple Fourier transforms over overlapping windows of the data to smooth noise

effects in the resulting spectrum; we used this method and two window functions (*top-hat* and the *Hanning*, or *cosine* window) to compute PSDs shown in figure 10.6. The Hanning window suppresses noise and better picks up features at high frequencies, at the expense of affecting the shape of the continuum (note that computations are done in linear frequency space, while the figure shows a logarithmic frequency axis).

For detailed discussion of these effects and other methods to analyze gravitational wave data, see the literature provided at the LIGO website.

10.2.4. Wavelets

The trigonometric basis functions used in the Fourier transform have an infinite extent and for this reason the Fourier transform may not be the best method to analyze nonperiodic time series data, such as the case of a localized event (e.g., a burst that decays over some timescale so that the PSD is also varying with time). Although we can evaluate the PSD for finite stretches of time series and thus hope to detect its eventual changes, this approach (called *spectrogram*, or *dynamical power spectra analysis*) suffers from degraded spectral resolution and is sensitive to the specific choice of time series segmentation length. With basis functions that are localized themselves, this downside of the Fourier transform can be avoided and the ability to identify signal, filter, or compress data, significantly improved.

An increasingly popular family of basis functions is called *wavelets*. A good introduction is available in NumRec.[3] By construction, wavelets are localized in *both* frequency and time domains. Individual wavelets are specified by a set of *wavelet filter coefficients*. Given a wavelet, a complete orthonormal set of basis functions can be constructed by scalings and translations. Different wavelet families trade the localization of a wavelet with its smoothness. For example, in the frequently used *Daubechies wavelets* [13], members of a family range from highly localized to highly smooth. Other popular wavelets include *Mexican hat* and *Haar* wavelets. A famous application of wavelet-based compression is the FBI's 200 TB database, containing 30 million fingerprints.

The *discrete wavelet transform* (DWT) can be used to analyze the power spectrum of a time series as a function of time. While similar analysis could be performed using the Fourier transform evaluated in short sliding windows, the DWT is superior. If a time series contains a localized event in time and frequency, DWT may be used to discover the event and characterize its power spectrum. A toolkit with wavelet analysis implemented in Python, PyWavelets, is publicly available.[4] A well-written guide to the use of wavelet transforms in practice can be found in [57].

Figures 10.8 and 10.9 show examples of using a particular wavelet to compute a wavelet PSD as a function of time t_0 and frequency f_0. The wavelet used is of the form

$$w(t|t_0, f_0, Q) = A\ \exp[i2\pi f_0(t-t_0)]\ \exp[-f_0^2(t-t_0)^2/Q^2], \tag{10.16}$$

where t_0 is the central time, f_0 is the central frequency, and the dimensionless parameter Q is a model parameter which controls the width of the frequency window. Several examples of this wavelet are shown in figure 10.7. The Fourier transform of eq. 10.16

[3] A compendium of materials on wavelet analysis can be found at http://www.wavelet.org/

[4] https://pywavelets.readthedocs.io/

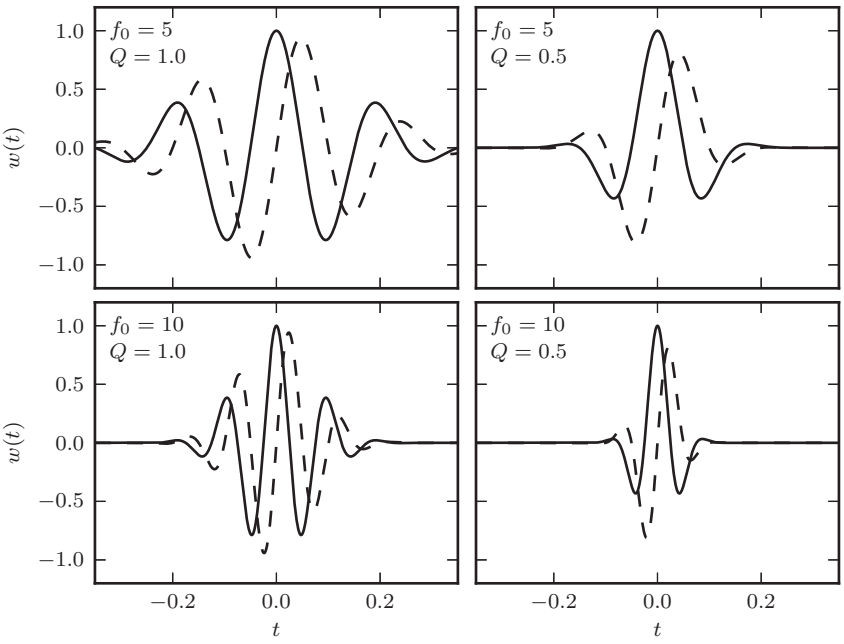

Figure 10.7. Wavelets for several values of wavelet parameters Q and f_0. Solid lines show the real part and dashed lines show the imaginary part (see eq. 10.16).

is given by

$$W(f|t_0, f_0, Q) = \left(\frac{\pi}{f_0^2/Q^2}\right)^{1/2} \exp(-i2\pi f t_0)\ \exp\left[\frac{-\pi^2 Q^2 (f - f_0)^2}{Q f_0^2}\right]. \qquad (10.17)$$

We should be clear here: the form given by eqs. 10.16–10.17 is not technically a wavelet because it does not meet the *admissibility* criterion (the equivalent of orthogonality in Fourier transforms). This form is closely related to a true wavelet, the *Morlet wavelet*, through a simple scaling and offset. Because of this, eqs. 10.16–10.17 should probably be referred to as "matched filters" rather than "wavelets." Orthonormality considerations aside, however, these functions display quite nicely one main property of wavelets: the localization of power in both time and frequency. For this reason, we will refer to these functions as "wavelets," and explore their ability to localize frequency signals, all the while keeping in mind the caveat about their true nature.

The wavelet transform applied to data $h(t)$ is given by

$$H_w(t_0; f_0, Q) = \int_{-\infty}^{\infty} h(t)\, w(t|t_0, f_0, Q) dt. \qquad (10.18)$$

This is a convolution; by the convolution theorem (eq. 10.9), we can write the Fourier transform of H_w as the pointwise product of the Fourier transforms of $h(t)$ and $w^*(t; t_0, f_0, Q)$. The first can be approximated using the discrete Fourier transform as shown in appendix E; the second can be found using the analytic formula for $W(f)$ (eq. 10.17). This allows us to quickly evaluate H_w as a function of t_0 and f_0, using two $\mathcal{O}(N \log N)$ fast Fourier transforms.

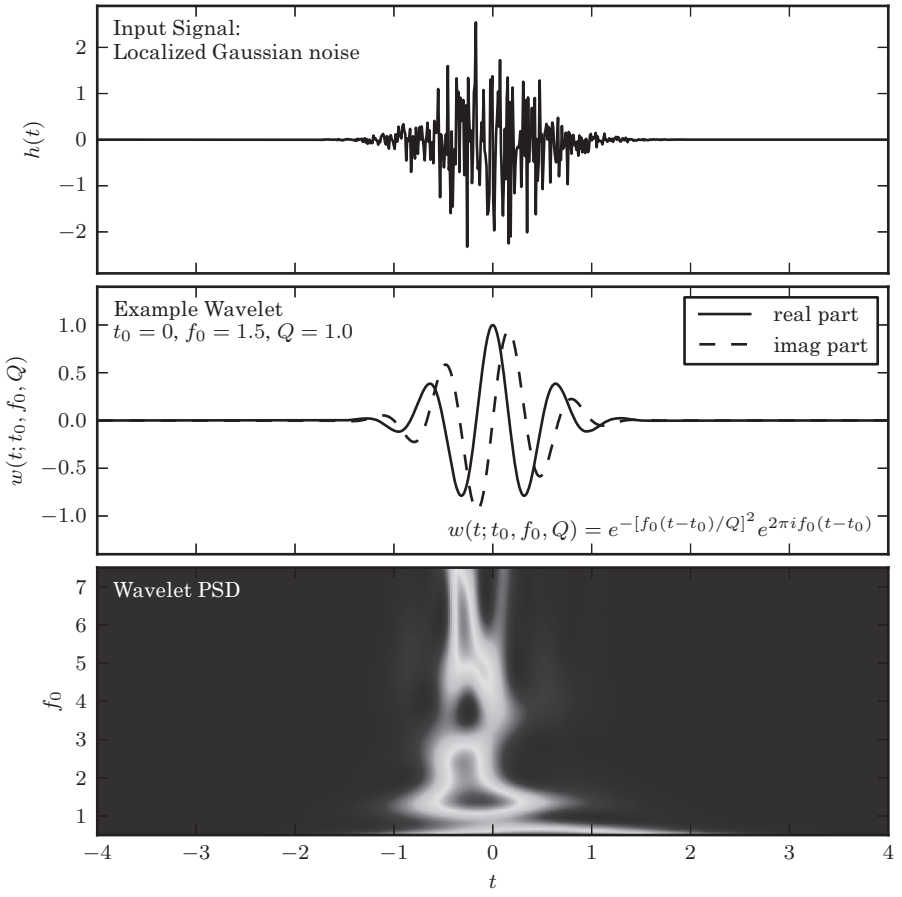

Figure 10.8. Localized frequency analysis using the wavelet transform. The upper panel shows the input signal, which consists of localized Gaussian noise. The middle panel shows an example wavelet. The lower panel shows the power spectral density as a function of the frequency f_0 and the time t_0, for $Q = 1.0$.

Figures 10.8 and 10.9 show the wavelet PSD, defined by $\mathrm{PSD}_w(f_0, t_0; Q) = |H_w(t_0; f_0, Q)|^2$. Unlike the typical Fourier-transform PSD, the wavelet PSD allows detection of frequency information which is localized in time. This is one approach used in the LIGO project to detect gravitational wave events. Because of the noise level in the LIGO measurements (see figure 10.6), rather than a standard wavelet like that seen in eq. 10.16, LIGO instead uses functions which are tuned to the expected form of the signal (i.e., matched filters). Another example of wavelet application is discussed below in figure 10.28.

A related method for time-frequency analysis when PSD is not constant, called *matching pursuit*, utilizes a large redundant set of nonorthogonal functions; see [38]. Unlike wavelet analysis, which assumes a fixed set of basis functions, in this method the data themselves are used to derive an appropriate large set of basis functions (called a *dictionary*). The matching pursuit algorithm has been successful in sound analysis, and recently in astronomy; see [34].

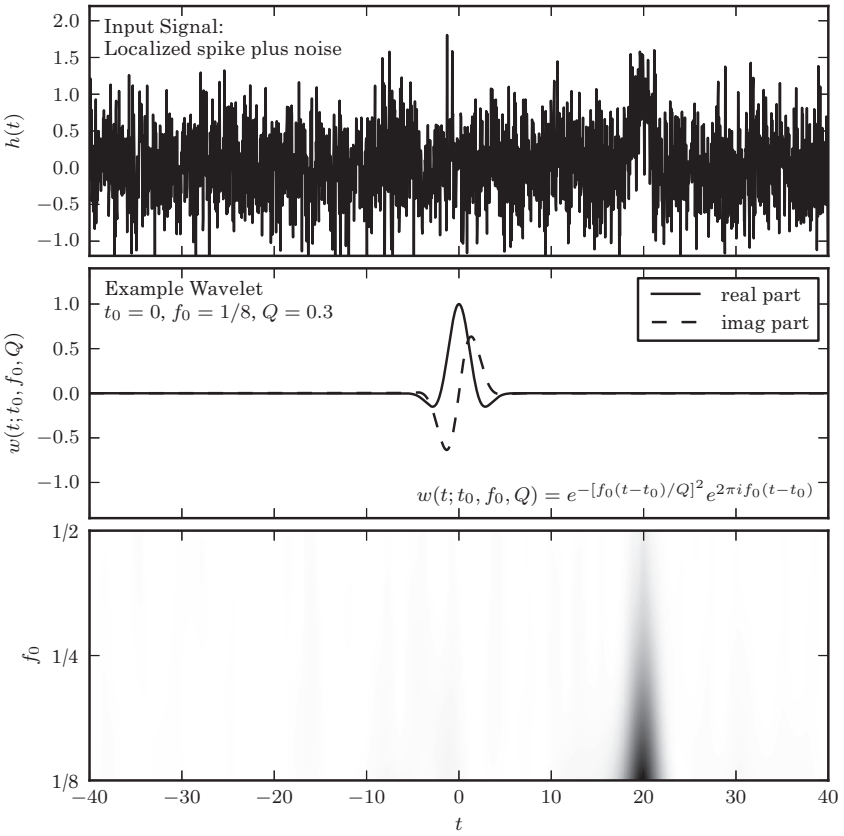

Figure 10.9. Localized frequency analysis using the wavelet transform. The upper panel shows the input signal, which consists of a Gaussian spike in the presence of white (Gaussian) noise (see figure 10.10). The middle panel shows an example wavelet. The lower panel shows the power spectral density as a function of the frequency f_0 and the time t_0, for $Q = 0.3$.

For exploring signals suspected to have time-dependent and frequency-dependent power, there are several tools available. Matplotlib implements a basic sliding-window spectrogram, using the function `matplotlib.mlab.specgram`. Alternatively, AstroML implements the wavelet PSD described above, which can be used as follows:

```
>>> import numpy as np
>>> from astroML.fourier import wavelet_PSD

>>> t = np.linspace(0, 1, 1000)  # times of signal
>>> x = np.random.normal(size=1000)  # white noise
>>> f0 = np.linspace(0.01, 1, 100)  # candidate frequencies

>>> WPSD = wavelet_PSD(t, x, f0, Q=1)  # 100 x 1000 PSD
```

For more detailed examples, see the source code used to generate the figures in this chapter.

10.2.5. Digital Filtering

Digital filtering aims to reduce noise in time series data, or to compress data. Common examples include *low-pass filtering*, where high frequencies are suppressed, *high-pass filtering*, where low frequencies are suppressed, *passband filtering*, where only a finite range of frequencies is admitted, and a *notch filter*, where a finite range of frequencies is blocked. Fourier analysis is one of the most useful tools for performing filtering. We will use a few examples to illustrate the most common applications of filtering. Numerous other techniques can be found in signal processing literature, including approaches based on the wavelets discussed above.

We emphasize that filtering always decreases the information content of data (despite making it appear less noisy). As we have already learned throughout previous chapters, when model parameters are estimated from data, raw (unfiltered) data should be used. In some sense, this is an analogous situation to binning data to produce a histogram—while very useful for visualization, estimates of model parameters can become biased if one is not careful. This connection will be made explicit below for the *Wiener filter*, where we show its equivalence to kernel density estimation (§6.1.1), the generalization of histogram binning.

Low-pass filters

The power spectrum for common Gaussian noise is flat and will extend to frequencies as high as the Nyquist limit, $f_N = 1/(2\Delta t)$. If the data are band-limited to a lower frequency, $f_c < f_N$, then they can be smoothed without much impact by suppressing frequencies $|f| > f_c$. Given a filter in frequency space, $\Phi(f)$, we can obtain a smoothed version of data by taking the inverse Fourier transform of

$$\hat{Y}(f) = Y(f)\,\Phi(f), \tag{10.19}$$

where $Y(f)$ is the discrete Fourier transform of data. At least in principle, we could simply set $\Phi(f)$ to zero for $|f| > f_c$, but this approach would result in ringing (i.e., unwanted oscillations) in the signal. Instead, the optimal filter for this purpose is constructed by minimizing the MISE between $\hat{Y}(f)$ and $Y(f)$ (for detailed derivation see NumRec) and is called the *Wiener filter*:

$$\Phi(f) = \frac{P_S(f)}{P_S(f) + P_N(f)}. \tag{10.20}$$

Here $P_S(f)$ and $P_N(f)$ represent components of a two-component (signal and noise) fit to the PSD of input data, $\mathrm{PSD}_Y(f) = P_S(f) + P_N(f)$, which holds as long as the signal and noise are uncorrelated. Given some assumed form of signal and noise, these terms can be determined from a fit to the observed PSD, as illustrated by the example shown in figure 10.10. Even when the fidelity of the PSD fit is not high, the resulting filter performs well in practice (the key features are that $\Phi(f) \sim 1$ at small frequencies and that it drops to zero at high frequencies for a band-limited signal).

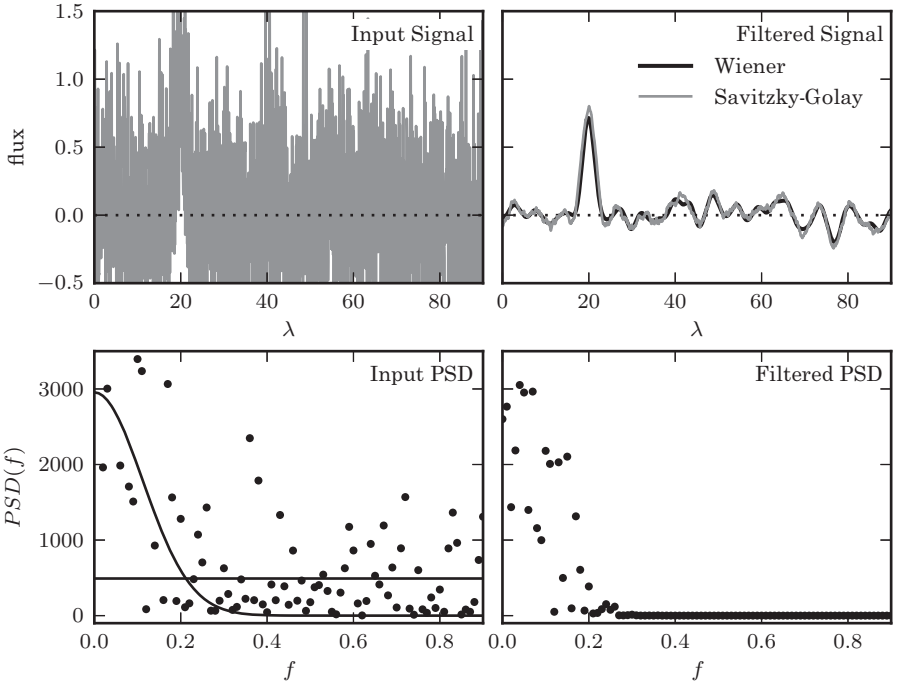

Figure 10.10. An example of data filtering using a Wiener filter. The upper-left panel shows noisy input data (200 evenly spaced points) with a narrow Gaussian peak centered at $x = 20$. The bottom panels show the input (left) and Wiener-filtered (right) power spectral density (PSD) distributions. The two curves in the bottom-left panel represent two-component fit to PSD given by eq. 10.20. The upper-right panel shows the result of the Wiener filtering on the input: the Gaussian peak is clearly seen. For comparison, we also plot the result of a fourth-order Savitzky–Golay filter with a window size of $\Delta\lambda = 10$.

There is a basic Wiener filter implementation in `scipy.signal.wiener`, based on assumptions of the local data mean and variance. AstroML implements a Wiener filter based on the more sophisticated procedure outlined above, using user-defined priors regarding the signal and noise:

```
>>> import numpy as np
>>> from astroML.filters import wiener_filter

>>> t = np.linspace(0, 1, 1000)
>>> y = np.random.normal(size=1000)  # white noise
>>> y_smooth = wiener_filter(t, y, signal='gaussian', noise='flat')
```

For a more detailed example, see the source code of figure 10.10.

There is an interesting connection between the kernel density estimation method discussed in §6.1.1 and Wiener filtering. By the convolution theorem, the Wiener-filtered result is equivalent to the convolution of the unfiltered signal with the inverse

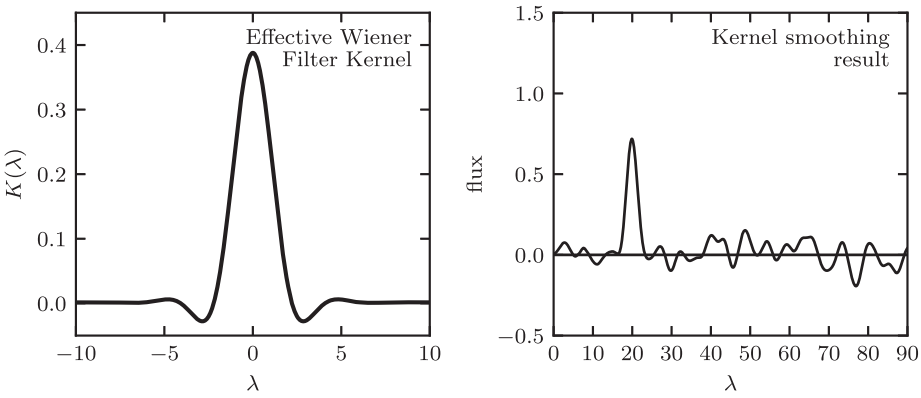

Figure 10.11. The left panel shows the inverse Fourier transform of the Wiener filter $\Phi(f)$ applied in figure 10.10. By the convolution theorem, the Wiener-filtered result is equivalent to the convolution of the unfiltered signal with the kernel shown above, and thus Wiener filtering and kernel density estimation (KDE) are directly related. The right panel shows the data smoothed by this kernel, which is equivalent to the Wiener filter smoothing in figure 10.10.

Fourier transform of $\Phi(f)$: this is the kernel shown in figure 10.11. This convolution is equivalent to kernel density estimation. When Wiener filtering is viewed in this way, it effectively says that we believe the signal is as wide as the central peak shown in figure 10.11, and the statistics of the noise are such that the minor peaks in the wings work to cancel out noise in the major peak. Hence, the modeling of the PSD in the frequency domain via eq. 10.20 corresponds to choosing the optimal kernel width. Just as detailed modeling of the Wiener filter is not of paramount importance, the choice of kernel is not either.

When data are not evenly sampled, the above Fourier techniques cannot be used. There are numerous alternatives discussed in NumRec and digital signal processing literature. As a low-pass filter, a very simple but powerful method is the *Savitzky–Golay filter*. It fits low-order polynomials to data (in the time domain) using sliding windows (it is also known as the *least-squares filter*). For a detailed discussion, see NumRec. The results of a fourth-order Savitzky–Golay filter with a window function of size $\Delta\lambda = 10$ is shown beside the Wiener filter result in figure 10.10.

High-pass filters

The most common example of high-pass filtering in astronomy is baseline estimation in spectral data. Unlike the case of low-pass filtering, here there is no universal filter recipe. Baseline estimation is usually the first step toward the estimation of model parameters (e.g., location, width, and strength of spectral lines). In such cases, the best approach might be full modeling and marginalization of baseline parameters as nuisance parameters at the end of analysis.

AstroML contains an implementation of the minimum component filter described above:

```
>>> import numpy as np
>>> from astroML.filters import min_component_filter

>>> t = np.linspace(0, 1, 1000)
>>> x = np.exp(-500 * (t - 0.5) ** 2)  # a spike at 0.5
>>> x += np.random.random(size=1000)  # white noise
>>> mask = (t > 0.4) & (t < 0.6)  # mask the signal

>>> x_smooth = min_component_filter(t, x, mask)
```

For a more detailed example, see the source code for figures 10.12 and 10.13.

A simple iterative technique for high-pass filtering, called *minimum component filtering*, is discussed in detail in WJ03. These are the main steps:

1. Determine baseline: exclude or mask regions where signal is clearly evident and fit a baseline model (e.g., a low-order polynomial) to the unmasked regions.
2. Get FT for the signal: after subtracting the baseline fit in the unmasked regions (i.e., a linear regression fit), apply the discrete Fourier transform.
3. Filter the signal: remove high frequencies using a low-pass filter (e.g., Wiener filter), and inverse Fourier transform the result.
4. Recombine the baseline and the filtered signal: add the baseline fit subtracted in step 2 to the result from step 3. This is the minimum component filtering estimate of baseline.

A minimum component filter applied to the spectrum of a white dwarf from the SDSS is shown in figures 10.12 and 10.13.

10.3. Analysis of Periodic Time Series

We shall now focus on characterization of periodic time series. Many types of variable stars show periodic flux variability; analysis of such stars is important both for understanding stellar evolution and for using such stars as distance indicators (e.g., Cepheids and RR Lyrae stars); for a good summary of the variable star zoo, see [24]. The main goal of the analysis is to detect variability and to estimate the period and its uncertainty.

A periodic time series satisfies $y(t+P) = y(t)$, where P is the period (assuming no noise). In the context of periodic variability, a convenient concept is the so-called phased light curve, where the data (and models) are plotted as a function of phase,

$$\phi = \frac{t}{P} - \text{int}\left(\frac{t}{P}\right), \tag{10.21}$$

where the function int(x) returns the integer part of x.

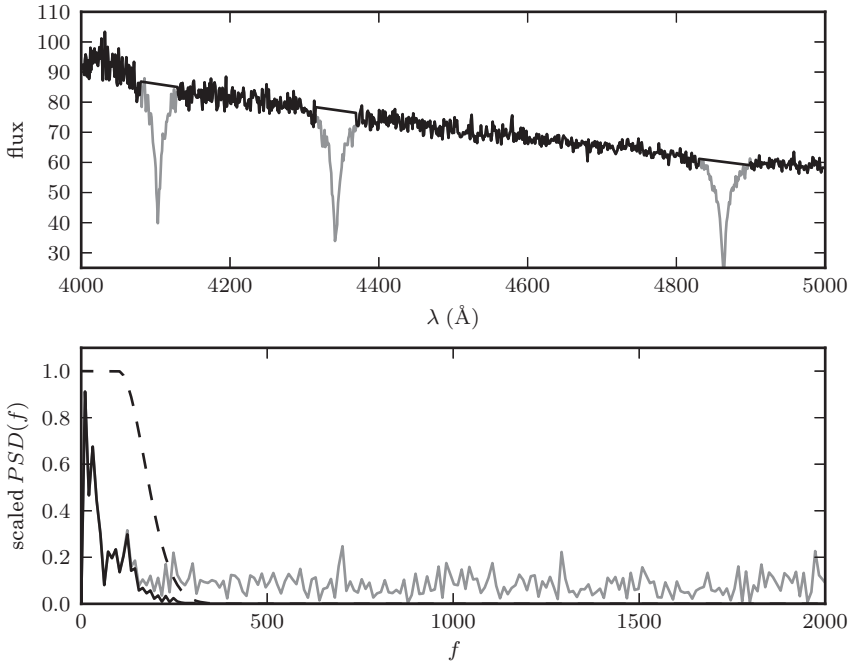

Figure 10.12. The intermediate steps of the minimum component filter procedure applied to the spectrum of a white dwarf from the SDSS data set (mjd = 52199, plate = 659, fiber = 381). The top panel shows the input spectrum; the masked sections of the input spectrum are shown by thin lines (i.e., step 1 of the process in §10.2.5). The bottom panel shows the PSD of the masked spectrum, after the linear fit has been subtracted (gray line). A simple low-pass filter (dashed line) is applied, and the resulting filtered spectrum (dark line) is used to construct the result shown in figure 10.13.

We begin discussion with analysis of a simple single harmonic model, including its relationship to the discrete Fourier transform and the Lomb–Scargle periodogram. We then extend discussion to analysis of truncated Fourier series and provide an example of classification of periodic light curves. We conclude with methods for analysis of arrival time data.

10.3.1. A Single Sinusoid Model

Given time series data $(t_1, y_1), \ldots, (t_N, y_N)$, we want to test whether it is consistent with periodic variability and, if so, to estimate the period. In order to compute the posterior pdf for the frequency (or period) of a periodic variability sought in data, we need to adopt a specific model. We will first consider a simple model based on a single harmonic with angular frequency ω $(= 2\pi f = 2\pi P^{-1})$,

$$y(t) = A\ \sin(\omega t + \phi) + \epsilon, \tag{10.22}$$

where the first term models the underlying process that generated the data and ϵ is measurement noise. Instead of using the phase ϕ, it is possible to shift the time axis and write the argument as $\omega\,(t - t_o)$. In the context of subsequent analysis, it is

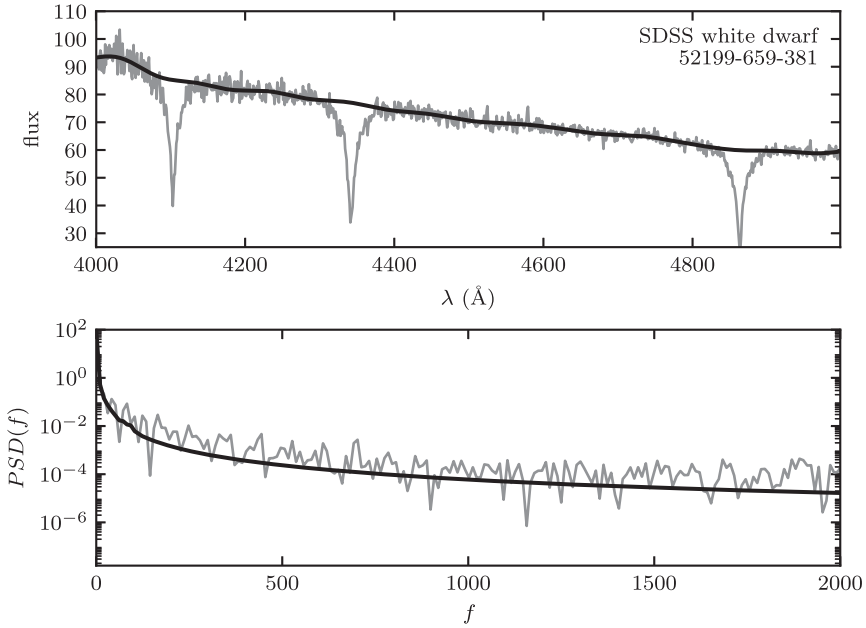

Figure 10.13. A minimum component filter applied to the spectrum of a white dwarf from SDSS data set (mjd = 52199, plate = 659, fiber = 381). The upper panel shows a portion of the input spectrum, along with the continuum computed via the minimum component filtering procedure described in §10.2.5 (see figure 10.12). The lower panel shows the PSD for both the input spectrum and the filtered result.

practical to use trigonometric identities to rewrite this model as

$$y(t) = a\ \sin(\omega t) + b\ \cos(\omega t), \tag{10.23}$$

where $A = (a^2 + b^2)^{1/2}$ and $\phi = \tan^{-1}(b/a)$. The model is now linear with respect to coefficients a and b, and nonlinear only with respect to frequency ω. Determination of these three parameters from the data is the main goal of the following derivation.

We fit this model to a set of data points $(t_1, y_1), \ldots, (t_N, y_N)$, $j = 1, \ldots, N$ with noise ϵ described by homoscedastic Gaussian errors parametrized by σ. We will consider cases of both known and unknown σ. Note that there is no assumption that the times t_j are evenly sampled. Below, we will generalize this model to a case with heteroscedastic errors and an additional constant term in the assumed model (here, we will assume that the mean value was subtracted from the raw data values to obtain y_j, that is, $\overline{y} = 0$; this may not work well in practice, as discussed further below). We begin with this simplified case for pedagogical reasons, to better elucidate choices to be made in Bayesian analysis and its connections to classical power spectrum analysis. For the same reasons, we provide a detailed derivation.

Following the methodology from chapters 4 and 5, we can write the data likelihood as

$$L \equiv p(\{t, y\}|\omega, a, b, \sigma) = \prod_{j=1}^{N} \frac{1}{\sqrt{2\pi}\sigma}\ \exp\left(\frac{-[y_j - a\sin(\omega t_j) - b\cos(\omega t_j)]^2}{2\sigma^2}\right). \tag{10.24}$$

Although we *assumed* a Gaussian error distribution, if the only information about noise was a known value for the variance of its probability distribution, we would still end up with a Gaussian distribution via the principle of maximum entropy (see §5.2.2).

We shall retrace the essential steps of a detailed analysis developed by Bretthorst [4, 6, 7]. We shall assume uniform priors for a, b, ω, and σ. Note that this choice of priors leads to nonuniform priors on A and ϕ if we choose to parametrize the model via eq. 10.22. Nevertheless, the resulting pdfs are practically equal when data overwhelms the prior information; for a more detailed discussion see [3]. We will also assume that ω and σ must be positive. The posterior pdf is

$$p(\omega, a, b, \sigma | \{t, y\}) \propto \sigma^{-N} \exp\left(\frac{-NQ}{2\sigma^2}\right), \tag{10.25}$$

where

$$Q = V - \frac{2}{N}\left[a\, I(\omega) + b\, R(\omega) - a\, b\, M(\omega) - \frac{1}{2}a^2\, S(\omega) - \frac{1}{2}b^2\, C(\omega)\right]. \tag{10.26}$$

The following terms depend only on data and frequency ω:

$$V = \frac{1}{N}\sum_{j=1}^{N} y_j^2, \tag{10.27}$$

$$I(\omega) = \sum_{j=1}^{N} y_j \sin(\omega t_j), \quad R(\omega) = \sum_{j=1}^{N} y_j \cos(\omega t_j), \tag{10.28}$$

$$M(\omega) = \sum_{j=1}^{N} \sin(\omega t_j)\ \cos(\omega t_j), \tag{10.29}$$

and

$$S(\omega) = \sum_{j=1}^{N} \sin^2(\omega t_j), \quad C(\omega) = \sum_{j=1}^{N} \cos^2(\omega t_j). \tag{10.30}$$

The expression for Q can be considerably simplified. When $N \gg 1$ (and unless $\omega t_N \ll 1$, which is a low-frequency case corresponding to a period of oscillation longer than the data-taking interval and will be considered further below) we have that $S(\omega) \approx C(\omega) \approx N/2$ and $M(\omega) \ll N/2$ (using identities $\sin^2(\omega t_j) = [1 - \cos(2\omega t_j)]/2$, $\cos^2(\omega t_j) = [1 + \cos(2\omega t_j)]/2$ and $\sin(\omega t_j)\cos(\omega t_j) = \sin(2\omega t_j)/2$), and thus

$$Q \approx V - \frac{2}{N}\left[a\, I(\omega) + b\, R(\omega)\right] + \frac{1}{2}(a^2 + b^2). \tag{10.31}$$

When quantifying the evidence for periodicity, we are not interested in specific values of a and b. To obtain the two-dimensional posterior pdf for ω and σ, we marginalize over the four-dimensional pdf given by eq. 10.25,

$$p(\omega, \sigma | \{t, y\}) \propto \int p(\omega, a, b, \sigma | \{t, y\})\, da\, db, \tag{10.32}$$

where the integration limits for a and b are sufficiently large for the integration to be effectively limited by the exponential (and not by the adopted limits for a and b, whose absolute values should be at least several times larger than σ/N). It is easy to derive (by completing the square of the arguments in the exponential)

$$p(\omega, \sigma | \{t, y\}) \propto \sigma^{-(N-2)} \exp\left(\frac{-NV}{2\sigma^2} + \frac{P(\omega)}{\sigma^2}\right), \tag{10.33}$$

where the periodogram $P(\omega)$ is given by

$$P(\omega) = \frac{1}{N}[I^2(\omega) + R^2(\omega)]. \tag{10.34}$$

In the case when the noise level σ is known, this result further simplifies to

$$p(\omega | \{t, y\}, \sigma) \propto \exp\left(\frac{P(\omega)}{\sigma^2}\right). \tag{10.35}$$

Alternatively, when σ is unknown, $p(\omega, \sigma | \{t, y\})$ can be marginalized over σ to obtain (see [3])

$$p(\omega | \{t, y\}) \propto \left[1 - \frac{2P(\omega)}{N\,V}\right]^{1-N/2}. \tag{10.36}$$

The best-fit amplitudes

Marginalizing over amplitudes a and b is distinctively Bayesian. We now determine MAP estimates for a and b (which are identical to maximum likelihood estimates because we assumed uniform priors) using

$$\left.\frac{d\, p(\omega, a, b, \sigma | \{t, y\})}{da}\right|_{a=a_0} = 0, \tag{10.37}$$

and analogously for b, yielding

$$a_0 = \frac{2I(\omega)}{N}, \quad b_0 = \frac{2R(\omega)}{N}. \tag{10.38}$$

By taking second derivatives of $p(\omega, a, b, \sigma | \{t, y\})$ with respect to a and b, it is easy to show that uncertainties for MAP estimates of amplitudes, a_0 and b_0, in the case of known σ are

$$\sigma_a = \sigma_b = \sigma\sqrt{2/N}. \tag{10.39}$$

Therefore, for a given value of ω, the best-fit amplitudes (a and b) from eq. 10.23 are given by eqs. 10.38 and 10.39 (in case of known σ).

The Meaning of Periodogram

We have not yet answered what is the best value of ω supported by the data, and whether the implied periodic variability is statistically significant. We can compute $\chi^2(\omega)$ for a fit with $a = a_0$ and $b = b_0$ as

$$\chi^2(\omega) \equiv \frac{1}{\sigma^2}\sum_{j=1}^{N}[y_j - y(t_j)]^2 = \frac{1}{\sigma^2}\sum_{j=1}^{N}[y_j - a_0\ \sin(\omega t_j) - b_0\ \cos(\omega t_j)]^2. \quad (10.40)$$

It can be easily shown that

$$\chi^2(\omega) = \chi_0^2\left[1 - \frac{2}{N\,V}\,P(\omega)\right], \quad (10.41)$$

where $P(\omega)$ is the periodogram given by eq. 10.34, and χ_0^2 corresponds to a model $y(t)$=constant (recall that here we assumed $\bar{y} = 0$),

$$\chi_0^2 = \frac{1}{\sigma^2}\sum_{j=1}^{N} y_j^2 = \frac{N\,V}{\sigma^2}. \quad (10.42)$$

This result motivates a renormalized definition of the periodogram as

$$P_{\rm LS}(\omega) = \frac{2}{N\,V}\,P(\omega), \quad (10.43)$$

where index LS stands for *Lomb–Scargle periodogram*, introduced and discussed further below. With this renormalization, $0 \le P_{\rm LS}(\omega) \le 1$, and thus the reduction in χ^2 for the harmonic model, relative to χ^2 for the pure noise model, χ_0^2, is

$$\frac{\chi^2(\omega)}{\chi_0^2} = 1 - P_{\rm LS}(\omega). \quad (10.44)$$

The relationship between $\chi^2(\omega)$ and $P(\omega)$ can be used to assess how well $P(\omega)$ estimates the true power spectrum. If the model is correct, then we expect that χ^2 corresponding to the peak with maximum height, at $\omega = \omega_0$, is N, with a standard deviation of $\sqrt{2N}$ (assuming that N is sufficiently large so that this Gaussian approximation is valid). It is easy to show that the expected height of the peak is

$$P(\omega_0) = \frac{N}{4}(a_0^2 + b_0^2), \quad (10.45)$$

with a standard deviation

$$\sigma_P(\omega_0) = \frac{\sqrt{2}}{2}\,\sigma^2, \quad (10.46)$$

where a_0 and b_0 are evaluated using eq. 10.38 and $\omega = \omega_0$.

As is evident from eq. 10.45, the expected height of the peaks in a periodogram does not depend on σ, as we already observed in figure 10.5. On the other hand, its

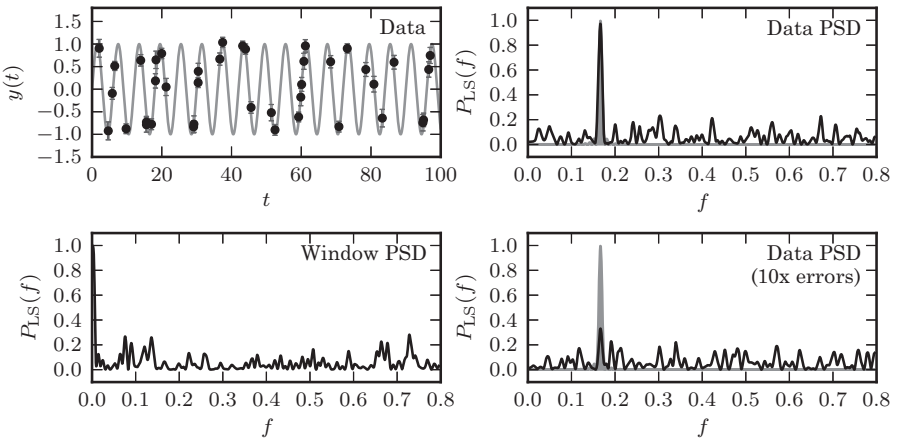

Figure 10.14. An illustration of the impact of measurement errors on $P_{\rm LS}$ (cf. figure 10.4). The top-left panel shows a simulated data set with 40 points drawn from the function $y(t|P) = \sin t$ (i.e., $f = 1/(2\pi) \sim 0.16$) with random sampling. Heteroscedastic Gaussian noise is added to the observations, with a width drawn from a uniform distribution with $0.1 \leq \sigma \leq 0.2$ (this error level is negligible compared to the amplitude of variation). The spectral window function (PSD of sampling times) is shown in the bottom-left panel. The PSD ($P_{\rm LS}$) computed for the data set from the top-left panel is shown in the top-right panel; it is equal to a convolution of the single peak (shaded in gray) with the window PSD shown in the bottom-left panel (e.g., the peak at $f \sim 0.42$ in the top-right panel can be traced to a peak at $f \sim 0.26$ in the bottom-left panel). The bottom-right panel shows the PSD for a data set with errors increased by a factor of 10. Note that the peak $f \sim 0.16$ is now much shorter, in agreement with eq. 10.47. In addition, errors now exceed the amplitude of variation and the data PSD is no longer a simple convolution of a single peak and the spectral window.

variation from the expected height depends only on noise σ, and not on the sample size N. Alternatively, the expected height of $P_{\rm LS}$, which is bound to the [0–1] range, is

$$P_{\rm LS}(\omega_0) = 1 - \frac{\sigma^2}{V}. \tag{10.47}$$

As noise becomes negligible, $P_{\rm LS}(\omega_0)$ approaches its maximum value of 1. As noise increases, $P_{\rm LS}(\omega_0)$ decreases and eventually the peak becomes too small and effectively buried in the background periodogram noise. Of course, these results are only correct if the model is correct; if it is not, the PSD peaks are shorter (because χ^2 is larger; see eq. 10.44).

An illustration of the impact of measurement errors σ on $P_{\rm LS}$ is shown in figure 10.14. The measured PSD is a convolution of the true underlying PSD and the spectral window (the PSD of the sampling window function; recall §10.2.3). As the measurement noise increases, the peak corresponding to the underlying frequency in the data can become as small as the peaks in the spectral window; in this case, the underlying periodic variability becomes hard to detect.

Finally, we can use the results of this section to quantify the detailed behavior of frequency peaks around their maximum, and to estimate the uncertainty in ω of the highest peak. When the single harmonic model is appropriate and well constrained by data, the posterior pdf for ω given by eq. 10.35 can be approximated as a Gaussian

$\mathcal{N}(\omega_0, \sigma_\omega)$. The uncertainty σ_ω can be obtained by taking the second derivative of $P(\omega)$,

$$\sigma_\omega = \left| \frac{d^2 P(\omega)}{d\omega^2} \right|_{\omega=\omega_0}^{-1/2}. \tag{10.48}$$

The Gaussian approximation implies that $P_{\rm LS}$ can be approximated by a parabola around its maximum,

$$P_{\rm LS}(\omega) \approx 1 - \frac{\sigma^2}{V} - \frac{(\omega-\omega_0)^2}{NV\sigma_\omega^2}. \tag{10.49}$$

Note that the height of the peak, $P_{\rm LS}(\omega_0)$, does *not* signify the precision with which ω_0 is estimated; instead, σ_ω is related to the peak width. It can be easily shown that the full width at half maximum of the peak, $\omega_{1/2}$, is related to σ_ω as

$$\sigma_\omega = \omega_{1/2} \left[2N(V-\sigma^2) \right]^{-1/2}. \tag{10.50}$$

For a fixed length of time series, T, $\omega_{1/2} \propto T^{-1}$, and $\omega_{1/2}$ does not depend on the number of data points N when there are on average at least a few points per cycle. Therefore, for a fixed T, $\sigma_\omega \propto N^{-1/2}$ (note that fractional errors in ω_0 and the period are equal).

We can compute σ_ω, the uncertainty of ω_0, from data using eq. 10.48 and

$$\left| \frac{d^2 P(\omega)}{d\omega^2} \right|_{\omega=\omega_0} = \frac{2}{N} \left[\left(R'(\omega_0)\right)^2 + R(\omega_0)\, R''(\omega_0) + \left(I'(\omega_0)\right)^2 + I(\omega_0)\, I''(\omega_0) \right], \tag{10.51}$$

where

$$R'(\omega) = -\sum_{j=1}^{N} y_j\, t_j \sin(\omega t_j), \quad R''(\omega) = -\sum_{j=1}^{N} y_j\, t_j^2 \cos(\omega t_j), \tag{10.52}$$

and

$$I'(\omega) = \sum_{j=1}^{N} y_j\, t_j \cos(\omega t_j), \quad I''(\omega) = -\sum_{j=1}^{N} y_j\, t_j^2 \sin(\omega t_j). \tag{10.53}$$

The Significance of Periodogram Peaks

For a given ω, the peak height, as shown by eq. 10.44, is a measure of the reduction in χ^2 achieved by the model, compared to χ^2 for a pure noise model. We can use BIC and AIC information criteria to compare these two models (see eqs. 4.18 and 5.35). The difference in BIC is

$$\Delta{\rm BIC} = \chi_0^2 - \chi^2(\omega_0) - (k_0 - k_\omega)\, \ln N, \tag{10.54}$$

where the number of free parameters is $k_0 = 1$ for the no-variability model (the mean value was subtracted) and $k_\omega = 4$ for a single harmonic model (it is assumed

that the uncertainty for all free parameters decreases proportionally to $N^{-1/2}$). For homoscedastic errors,

$$\Delta\text{BIC} = \frac{NV}{\sigma^2} P_{\text{LS}}(\omega_0) - 3 \ln N, \tag{10.55}$$

and similarly

$$\Delta\text{AIC} = \frac{NV}{\sigma^2} P_{\text{LS}}(\omega_0) - 6. \tag{10.56}$$

There is an important caveat here: it was assumed that ω_0 was given (i.e., known). When we need to find ω_0 using data, we evaluate $P_{\text{LS}}(\omega)$ for many ω and thus we have the case of multiple hypothesis testing (recall §4.6). We return to this point further below (§10.3.2).

When errors are heteroscedastic, the term NV/σ^2 is replaced by $\chi_0^2 = \sum_j (y_j/\sigma_j)^2$. Using the approximation given by eq. 10.47, and assuming a single harmonic with amplitude A ($V = \sigma^2 + A^2/2$), the first term becomes $N(A/\sigma)^2/2$. If we adopt a difference of ten as a threshold for evidence in favor of harmonic behavior for both information criteria, the minimum A/σ ratio to detect periodicity is approximately

$$\frac{A}{\sigma} > \left(\frac{20 + 6 \ln N}{N} \right)^{1/2} \tag{10.57}$$

using BIC, and with $\ln N$ replaced by 2 for AIC. For example, with $N = 100$, periodicity can be found for $A \sim 0.7\sigma$, and when $N = 1000$ even for $A \sim 0.2\sigma$. At the same time, the fractional accuracy of estimated A is about 20%–25% (i.e., the signal-to-noise ratio for measuring A is $A/\sigma_A \sim 4$–5).

Therefore, to answer the question "did my data come from a periodic process?", we need to compute $P_{\text{LS}}(\omega)$ first, and then the model odds ratio for a single sinusoid model vs. no-variability model via eq. 10.55. These results represent the foundations for analysis of unevenly periodic time series. Practical examples of this analysis are discussed in the next section. For more practical details, please see VanderPlas (2018).

Bayesian View of Fourier Analysis

Now we can understand the results of Fourier analysis from a Bayesian viewpoint. The discrete Fourier PSD given by eq. 10.15 corresponds to the periodogram $P(\omega)$ from eq. 10.34, and *the highest peak in the discrete Fourier PSD is an optimal frequency estimator for the case of a single harmonic model and homoscedastic Gaussian noise*. As discussed in more detail in [3], the discrete PSD gives optimal results if the following conditions are met:

1. The underlying variation is a single harmonic with constant amplitude and phase.
2. The data are evenly sampled and N is large.
3. Noise is Gaussian and homoscedastic.

The performance of the discrete PSD when these conditions are not met varies from suboptimal to simply impossible to use, as in cases of unevenly sampled data. In the rest of this chapter, we will consider examples that violate all three of these conditions.

10.3.2. The Lomb–Scargle periodogram

As we already discussed, one of the most popular tools for analysis of regularly (evenly) sampled time series is the discrete Fourier transform (§10.2.3). However, it cannot be used when data are unevenly (irregularly) sampled (as is often the case in astronomy). The *Lomb–Scargle periodogram* [35, 45] is a standard method to search for periodicity in unevenly sampled time series data. A normalized Lomb–Scargle periodogram,[5] with heteroscedastic errors, is defined as

$$P_{\rm LS}(\omega) = \frac{1}{V}\left[\frac{R^2(\omega)}{C(\omega)} + \frac{I^2(\omega)}{S(\omega)}\right], \tag{10.58}$$

where data-based quantities independent of ω are

$$\overline{y} = \sum_{j=1}^{N} w_j\, y_j, \tag{10.59}$$

and

$$V = \sum_{j=1}^{N} w_j\, (y_j - \overline{y})^2, \tag{10.60}$$

with weights (for homoscedastic errors $w_j = 1/N$)

$$w_j = \frac{1}{W}\,\frac{1}{\sigma_j^2}, \quad W = \sum_{j=1}^{N} \frac{1}{\sigma_j^2}. \tag{10.61}$$

Quantities which depend on ω are defined as

$$R(\omega) = \sum_{j=1}^{N} w_j\, (y_j - \overline{y})\, \cos[\omega(t_j - \tau)], \quad I(\omega) = \sum_{j=1}^{N} w_j\, (y_j - \overline{y})\, \sin[\omega(t_j - \tau)], \tag{10.62}$$

$$C(\omega) = \sum_{j=1}^{N} w_j\, \cos^2[\omega(t_j - \tau)], \quad S(\omega) = \sum_{j=1}^{N} w_j\, \sin^2[\omega(t_j - \tau)]. \tag{10.63}$$

The offset τ makes $P_{\rm LS}(\omega)$ invariant to translations of the t axis, and is defined by

[5] An analogous periodogram in the case of uniformly sampled data was introduced in 1898 by Arthur Schuster with largely intuitive justification. Parts of the method attributed to Lomb and Scargle were also used previously by Gottlieb et al. [27].

$$\tan(2\omega\tau) = \frac{\sum_{j=1}^{N} w_j \sin(2\omega t_j)}{\sum_{j=1}^{N} w_j \cos(2\omega t_j)}. \tag{10.64}$$

For the purposes of notational simplicity in the derivations below, we will also define the quantity

$$M(\omega) = \sum_{j=1}^{N} w_j \sin[\omega(t_j - \tau)] \cos[\omega(t_j - \tau)]. \tag{10.65}$$

If τ is instead set to zero, then eq. 10.58 becomes slightly more involved, though still based only on the sums defined above; see [63]. We note that the definition of the Lomb–Scargle periodogram in NumRec contains an additional factor of 2 before V, and does not account for heteroscedastic errors. The above normalization follows Lomb [35], and produces $0 \leq P_{\rm LS}(\omega) < 1$.

The Meaning of the Lomb–Scargle Periodogram

The close similarity of the Lomb–Scargle periodogram and the results obtained for a single harmonic model in the previous section is evident. The main differences are inclusion of heteroscedastic (but still Gaussian!) errors in the Lomb–Scargle periodogram and slightly different expressions for the periodograms. When terms $C(\omega)$ and $S(\omega)$ in eq. 10.58 are approximated as 1/2, eq. 10.43 follows from eq. 10.58. Without these approximations, the exact solutions for MAP estimates of a and b are (cf. approximations from eq. 10.38)

$$a(\omega) = \frac{I(\omega)\, C(\omega) - R(\omega)\, M(\omega)}{C(\omega)\, S(\omega) - M^2(\omega)} \tag{10.66}$$

and

$$b(\omega) = \frac{R(\omega)\, S(\omega) - I(\omega)\, M(\omega)}{C(\omega)\, S(\omega) - M^2(\omega)}. \tag{10.67}$$

Therefore, the Lomb–Scargle periodogram corresponds to a single sinusoid model, and it is directly related to the χ^2 of this model (via eq. 10.44) evaluated with MAP estimates for a and b; see [12, 63]. It can be thought of as an "inverted" plot of the $\chi^2(\omega)$ normalized by the no-variation χ_0^2.

It is often misunderstood that the Lomb–Scargle periodogram somehow saves computational effort because it purportedly avoids explicit model fitting. However, the coefficients a and b can be computed using eqs. 10.66 and 10.67 with little extra effort. Instead, the key point of using the periodogram is that the significance of each peak can be assessed, as discussed in the previous section.

Practical Application of the Lomb–Scargle Periodogram

The underlying model of the Lomb–Scargle periodogram is nonlinear in frequency and basis functions at different frequencies are not orthogonal. As a result, the periodogram has many local maxima and thus in practice the global maximum of the periodogram is found by grid search. The searched frequency range can be bounded by $\omega_{\rm min} = 2\pi/T_{\rm data}$, where $T_{\rm data} = t_{\rm max} - t_{\rm min}$ is the interval sampled by the data, and

by $\omega_{\max}$. As a good choice for the maximum search frequency, a pseudo-Nyquist frequency $\omega_{\max} = \overline{\pi/\Delta t}$, where $\overline{1/\Delta t}$ is the median of the inverse time interval between data points, was proposed by [18] (in the case of even sampling, $\omega_{\max}$ is equal to the Nyquist frequency). In practice, this choice may be a gross underestimate because unevenly sampled data can detect periodicity with frequencies even higher than $2\pi/(\Delta t)_{\min}$ (see [23]). An appropriate choice of $\omega_{\max}$ thus depends on sampling (the phase coverage at a given frequency is the relevant quantity) and needs to be carefully chosen: a hard limit on maximum detectable frequency is of course given by the time interval over which individual measurements are performed, such as imaging exposure time.

The frequency step can be taken as proportional to $\omega_{\min}$, $\Delta\omega = \eta\omega_{\min}$, with $\eta \sim 0.1$ (see [18]). A linear regular grid for ω is a good choice because the width of peaks in $P_{\rm LS}(\omega)$ does not depend on ω_0. Note that in practice the ratio $\omega_{\max}/\omega_{\min}$ can be very large (often exceeding 10^5) and thus lead to many trial frequencies (the grid step must be sufficiently small to resolve the peak; that is, $\Delta\omega$ should not be larger than σ_ω). The use of trigonometric identities can speed up computations, as implemented in the `astropy` code used in the following example. Another approach to speeding up the evaluation for a large number of frequencies is based on Fourier transforms, and is described in NumRec.

SciPy contains a fast Lomb–Scargle implementation, which works only for homoscedastic errors: `scipy.signal.lombscargle`. On the other hand, the implementation in Astropy implements both the standard and generalized Lomb–Scargle periodograms, correctly accounting for heteroscedastic errors. Note that all frequencies in the Astropy implementation are not angular frequencies, but rather frequencies of oscillation, i.e., number of cycles per unit time:

```
>>> import numpy as np
>>> try:
...     from astropy.timeseries import LombScargle
... except ImportError:
...     # astropy.timeseries subpackage was added in v3.2
...     from astropy.stats import LombScargle

>>> t = 100 * np.random.random(1000)  # irregular observations
>>> dy = 1 + np.random.random(1000)  # heteroscedastic errors
>>> y = np.sin(2 * np.pi * t) + np.random.normal(0, dy)
>>> frequencies = np.linspace(0.01, 2, 1000)

>>> P_LS = LombScargle(t, y).power(frequencies)
```

For more details, see the Astropy documentation and references therein, and the online source code of the figures in this chapter.

Figure 10.15 shows the Lomb–Scargle periodogram for a relatively small sample with $N = 30$ and $\sigma \sim 0.8A$, where σ is the typical noise level and A is the amplitude of a single sinusoid model. The data are sampled over ∼300 cycles. Due to large noise and poor sampling, the data do not reveal any obvious pattern of periodic variation. Nevertheless, the correct period is easily discernible in the periodogram, and corresponds to $\Delta BIC = 26.1$.

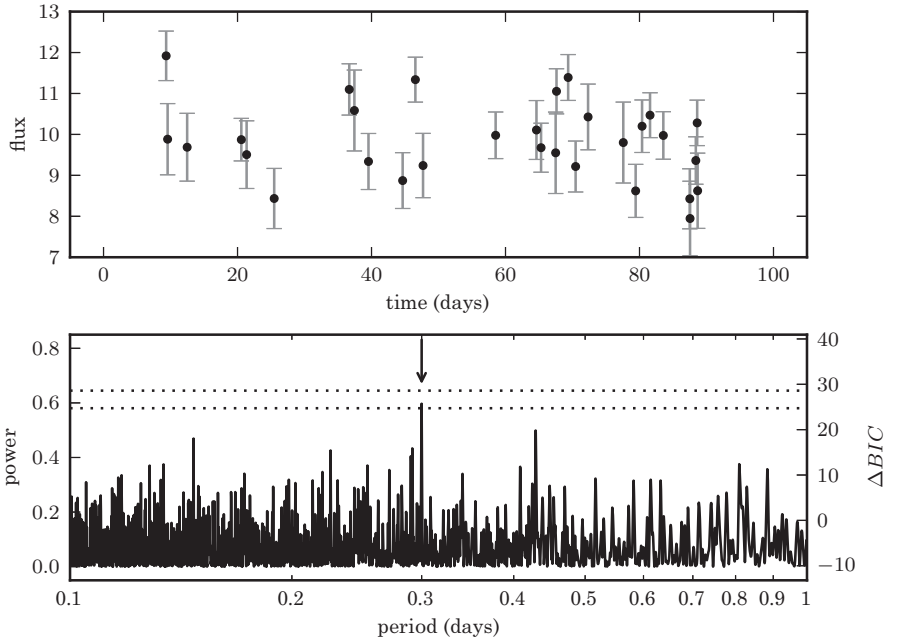

Figure 10.15. Example of a Lomb–Scargle periodogram. The data include 30 points drawn from the function $y(t|P) = 10 + \sin(2\pi t/P)$ with $P = 0.3$. Heteroscedastic Gaussian noise is added to the observations, with a width drawn from a uniform distribution with $0.5 \leq \sigma \leq 1.0$. Data are shown in the top panel and the resulting Lomb–Scargle periodogram is shown in the bottom panel. The arrow marks the location of the true period. The dotted lines show the 1% and 5% significance levels for the highest peak, determined by 1000 bootstrap resamplings (see §10.3.2). The change in BIC compared to a nonvarying source (eq. 10.55) is shown on the right y-axis. The maximum power corresponds to a $\Delta BIC = 26.1$, indicating the presence of a periodic signal. Bootstrapping indicates the period is detected at ∼5% significance.

False alarm probability

The derivation of eq. 10.54 assumed that ω_0 was given (i.e., known). However, to find ω_0 using data, $P_{\rm LS}(\omega)$ is evaluated for many different values of ω and thus the false alarm probability (FAP, the probability that $P_{\rm LS}(\omega_0)$ is due to chance) will reflect the multiple hypothesis testing discussed in §4.6. Even when the noise in the data is homoscedastic and Gaussian, an analytic estimator for the FAP for general uneven sampling does not exist (a detailed discussion and references can be found in FB2012; see also [25] and [49]).

A straightforward method for computing the FAP that relies on nonparametric bootstrap resampling was recently discussed in [54]. The times of observations are kept fixed and the values of y are drawn B times from observed values with replacement. The periodogram is computed for each resample and the maximum value found. The distribution of B maxima is then used to quantify the FAP. This method was used to estimate the 1% and 5% significance levels for the highest peak shown in figure 10.15.

Generalized Lomb–Scargle Periodogram

There is an important practical deficiency in the original Lomb–Scargle method described above: it is implicitly assumed that the mean of data values, $\bar{y}$, is a good estimator of the mean of $y(t)$. In practice, the data often do not sample all the phases equally, the data set may be small, or it may not extend over the whole duration of a cycle: the resulting error in mean can cause problems such as aliasing; see [12]. A simple remedy proposed in [12] is to add a constant offset term to the model from eq. 10.22. Zechmeister and Kürster [63] have derived an analytic treatment of this approach, dubbed the *generalized* Lomb–Scargle periodogram (it may be confusing that the same terminology was used by Bretthorst for a very different model [5]). The resulting expressions have a similar structure to the equations corresponding to the standard Lomb–Scargle approach listed above and are not reproduced here. Zechmeister and Kürster also discuss other methods, such as the floating-mean method and the date-compensated discrete Fourier transform, and show that they are by and large equivalent to the generalized Lomb–Scargle method.

Both the standard and generalized Lomb–Scargle methods are implemented in Astropy. Figure 10.16 compares the two in a nearly worst-case scenario where the data sampling is such that the standard method grossly overestimates the mean. While the standard approach fails to detect the periodicity due to the unlucky data sampling, the generalized Lomb–Scargle approach still recovers the expected signal. Though this example is quite contrived, it is not entirely artificial: in practice one could easily end up in such a situation if the period of the object in question were on the order of one day, such that minima occur only during daylight hours during the period of observation.

10.3.3. Truncated Fourier Series Model

What happens if data have an underlying variability that is more complex than a single sinusoid? Is the Lomb–Scargle periodogram still an appropriate model to search for periodicity? We address these questions by considering a multiple harmonic model.

Figure 10.17 shows phased (recall eq. 10.21) light curves for six stars from the LINEAR data set, with periods estimated using the Lomb–Scargle periodogram. In most cases the phased light curves are smooth and indicate that a correct period has been found, despite significant deviation from a single sinusoid shape. A puzzling case can be seen in the top-left panel where something is clearly wrong: at $\phi \sim 0.6$ the phased light curve has two branches! We will first introduce a tool to treat such cases, and then discuss it in more detail.

The single sinusoid model can be extended to include M Fourier terms,

$$y(t) = b_0 + \sum_{m=1}^{M} a_m \sin(m\omega t) + b_m \cos(m\omega t). \tag{10.68}$$

Following the steps from the single harmonic case, it can be easily shown that in this case the periodogram is (normalized to the 0–1 range)

$$P_M(\omega) = \frac{2}{V}\left[\sum_{m=1}^{M} R_m^2(\omega) + I_m^2(\omega)\right], \tag{10.69}$$

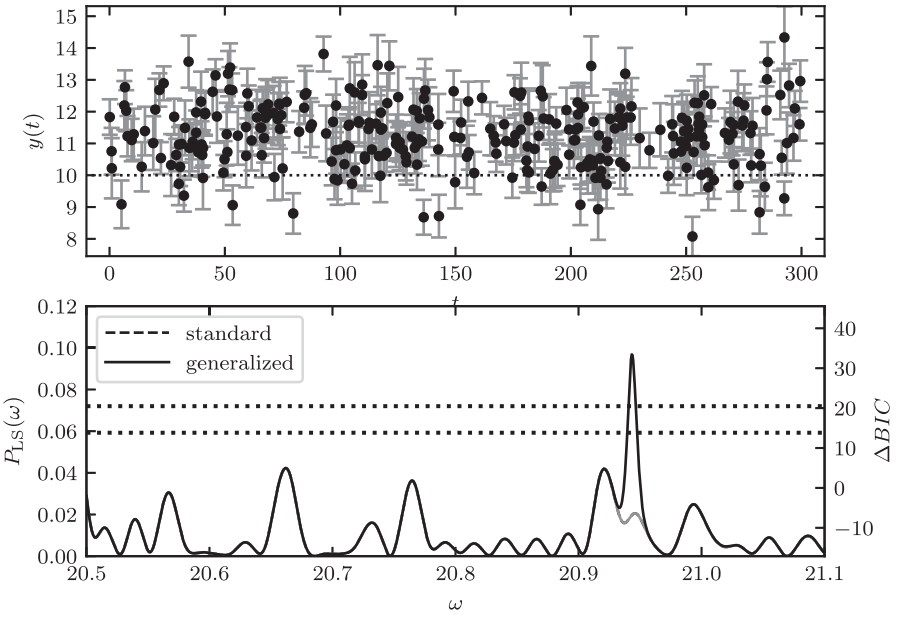

Figure 10.16. A comparison of the standard and generalized Lomb–Scargle periodograms for a signal $y(t) = 10 + \sin(2\pi t/P)$ with $P = 0.3$, corresponding to $\omega_0 \approx 21$. This example is, in some sense, a worst-case scenario for the standard Lomb–Scargle algorithm because the fraction of sampled points during the times when $y_{\rm true} < 10$ is very small, which leads to a gross overestimation of the mean. The bottom panel shows the Lomb–Scargle and generalized Lomb–Scargle periodograms for these data; the generalized method recovers the expected peak as the highest peak, while the standard method incorrectly chooses the peak at $\omega \approx 17.6$ (because it is higher than the true peak at $\omega_0 \approx 21$). The dotted lines show the 1% and 5% significance levels for the highest peak in the generalized periodogram, determined by 1000 bootstrap resamplings (see §10.3.2). (The version of this figure published in the first edition of this book had an error; for more details please see http://www.astroml.org/book_figures/chapter10/fig_LS_sg_comparison.html.)

where

$$I_m(\omega) = \sum_{j=1}^{N} w_j\, y_j \sin(m\omega t_j) \tag{10.70}$$

and

$$R_m(\omega) = \sum_{j=1}^{N} w_j\, y_j \cos(m\omega t_j), \tag{10.71}$$

where weights w_j are given by eq. 10.61 and V by eq. 10.60. Trigonometric functions with argument $m\omega t_j$ can be expressed in terms of functions with argument ωt_j so fitting M harmonics is not M times the computational cost of fitting a single harmonic (for detailed discussion see [40]). If M harmonics are indeed a better fit to data than a single harmonic, the peak of $P(\omega)$ around the true frequency will be enhanced relative to the peak for $M = 1$.

In the limit of large N, the MAP values of amplitudes can be estimated from

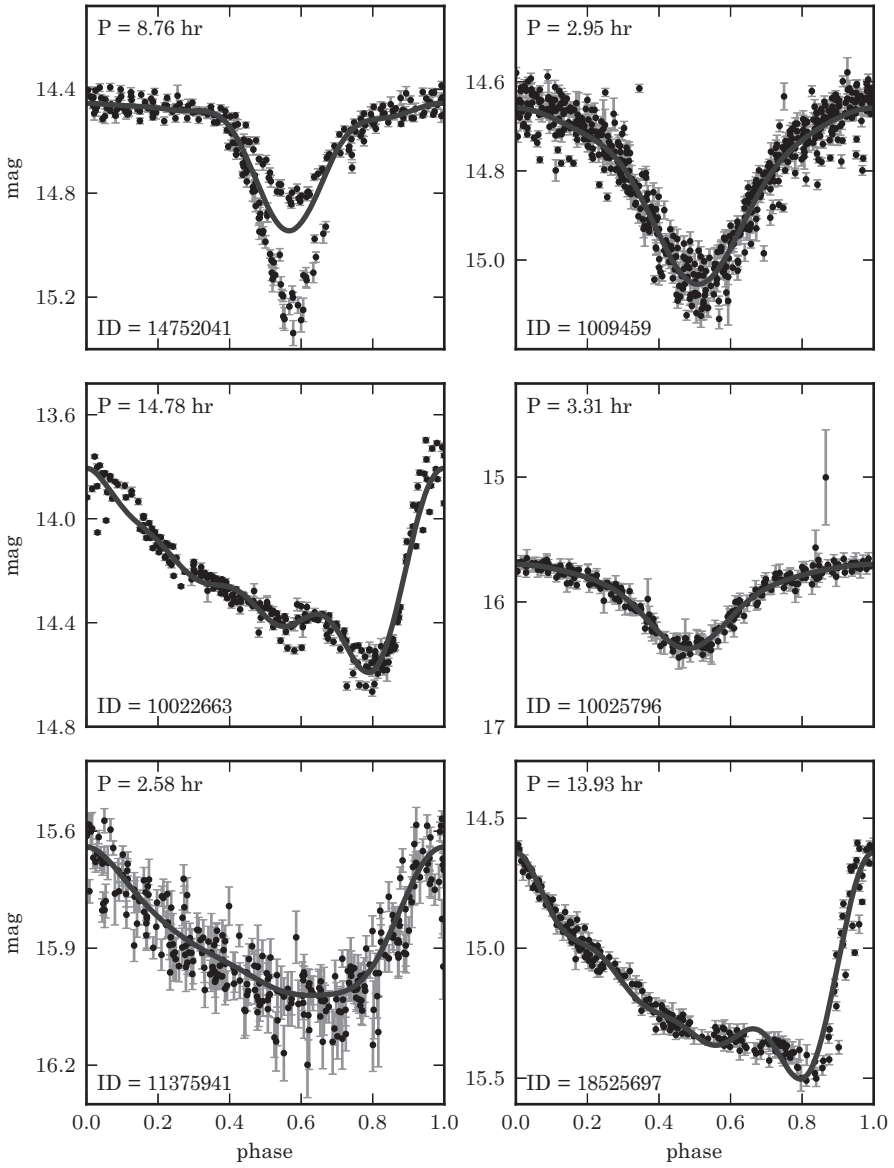

Figure 10.17. Phased light curves for six of the periodic objects from the LINEAR data set. The lines show the best fit to the phased light curve using the first four terms of the Fourier expansion (eq. 10.68), with the ω_0 selected using the Lomb–Scargle periodogram.

$$a_m = \frac{2I_m(\omega)}{N}, \quad b_m = \frac{2R_m(\omega)}{N}. \tag{10.72}$$

These expressions are only approximately correct (see discussion after eq. 10.30). The errors for coefficients a_m and b_m for $m > 0$ remain $\sigma\sqrt{2/N}$ as in the case of a single harmonic. The MAP value for b_0 is simply $\bar{y}$.

It is clear from eq. 10.69 that the periodogram $P_M(\omega)$ increases with M at all frequencies ω. The reason for this increase is that more terms allow for more fidelity and thus produce a smaller χ^2. Indeed, the input data could be exactly reproduced with $M = N/2 - 1$.

The Astropy implementation of the Lomb–Scargle periodogram includes an option to control the number of Fourier terms via the `nterms` argument to expand the single-term sinusoidal fit to truncated Fourier seried with multiple frequencies.

```
>>> import numpy as np
>>> try:
...     from astropy.timeseries import LombScargle
... except ImportError:
...     # astropy.timeseries subpackage was added in v3.2
...     from astropy.stats import LombScargle

>>> np.random.seed(42)
>>> t = 100 * np.random.random(1000)  # irregular observations
>>> dy = 1 + np.random.random(1000)  # heteroscedastic errors
>>> y = np.sin(2 * np.pi * t) + np.random.normal(0, dy)
>>> frequencies = np.linspace(0.01, 2, 1000)

>>> P_M = LombScargle(t, y, nterms=3).power(frequencies)
```

For more details, see the online source code of the figures in this chapter.

Figure 10.18 compares the periodograms and phased light curves for the problematic case from the top-left panel in figure 10.17 using $M = 1$ and $M = 6$. The single sinusoid model ($M = 1$) is so different from the true signal shape that it results in an incorrect period equal to 1/2 of the true period. The reason is that the underlying light curve has two minima (this star is an Algol-type eclipsing binary star) and a single sinusoid model produces a smaller χ^2 than for the pure noise model when the two minima are aligned, despite the fact that they have different depths. The $M = 6$ model is capable of modeling the two different minima, as well as flat parts of the light curve, and achieves a lower χ^2 for the correct period than for its alias favored by the $M = 1$ model. Indeed, the correct period is essentially unrecognizable in the power spectrum of the $M = 1$ model. Therefore, *when the signal shape significantly differs from a single sinusoid, the Lomb–Scargle periodogram may easily fail* (this is true both for the original and generalized implementations).

As this example shows, a good method for recognizing that there might be a problem with the best period is to require the phased light curve to be smooth. This requirement forms the basis for the so-called *minimum string length* (MSL) method (see [21]) and the *phase dispersion minimization* (PDM) method (see [53]). Both methods are based on analysis of the phased light curve: the MSL measures the length of the line connecting the points, and the PDM compares the interbin variance to the sample variance. Both metrics are minimized for smooth phased light curves.

The key to avoiding such pitfalls is to use a more complex model, such as a truncated Fourier series (or a template, if known in advance, or a nonparametric model, such as discussed in the following section). How do we choose an appropriate M for a truncated Fourier series? We can extend the analysis from the previous section and compare the BIC and AIC values for the model M to those for the no-variability model

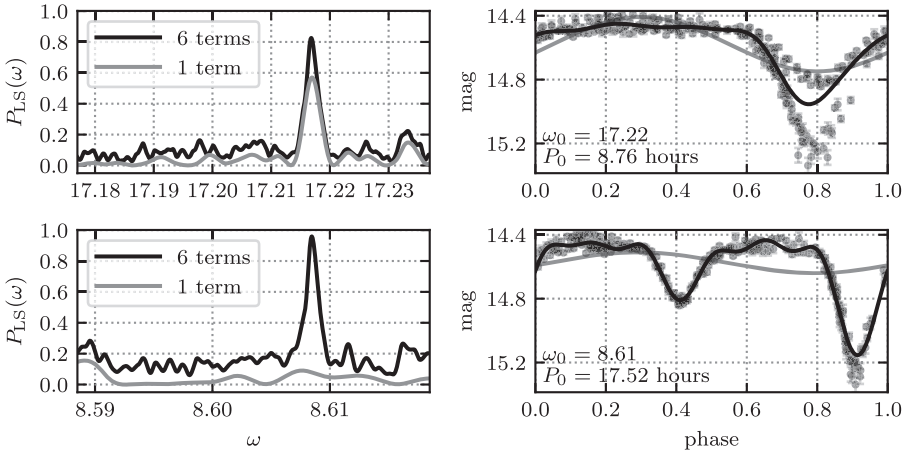

Figure 10.18. Analysis of a light curve where the standard Lomb–Scargle periodogram fails to find the correct period (the same star as in the top-left panel in figure 10.17). The two top panels show the periodograms (left) and phased light curves (right) for the truncated Fourier series model with $M = 1$ and $M = 6$ terms. Phased light curves are computed using the incorrect aliased period favored by the $M = 1$ model. The correct period is favored by the $M = 6$ model but unrecognized by the $M = 1$ model (bottom-left panel). The phased light curve constructed with the correct period is shown in the bottom-right panel. This case demonstrates that the Lomb–Scargle periodogram may easily fail when the signal shape significantly differs from a single sinusoid.

$y(t) = b_0$. The difference in BIC is

$$\Delta\text{BIC}_M = \chi_0^2\, P_M(\omega_0^M) - (2M+1)\, \ln N, \tag{10.73}$$

where $\chi_0^2 = \sum_j (y_j/\sigma_j)^2$ ($\chi_0^2 = NV/\sigma^2$ in the homoscedastic case) and similarly for AIC (with $\ln N$ replaced by 2). Figure 10.19 shows the value of ΔBIC as a function of the number of frequency components, using the same two peaks as shown in figure 10.18. With many Fourier terms in the fit, the BIC strongly favors the lower-frequency peak, which agrees with our intuition based on figure 10.18.

Finally, we note that while the Lomb–Scargle periodogram is perhaps the most popular method of finding periodicity in unevenly sampled time series data, it is not the only option. For example, nonparametric Bayesian estimation based on Gaussian processes (see §8.10) has recently been proposed in [61]. The MSL and PDM methods introduced above, as well as the Bayesian blocks algorithm (§5.7.2), are good choices when the shape of the underlying light curve cannot be approximated with a small number of Fourier terms.

10.3.4. Classification of Periodic Light Curves

As illustrated in Fig 10.17, stellar light curves often have distinctive shapes (e.g., such as skewed light curves of RR Lyrae type ab stars, or eclipsing binary stars). In addition to shapes, the period and amplitude of the light curve also represent distinguishing characteristics. With large data sets, it is desirable and often unavoidable

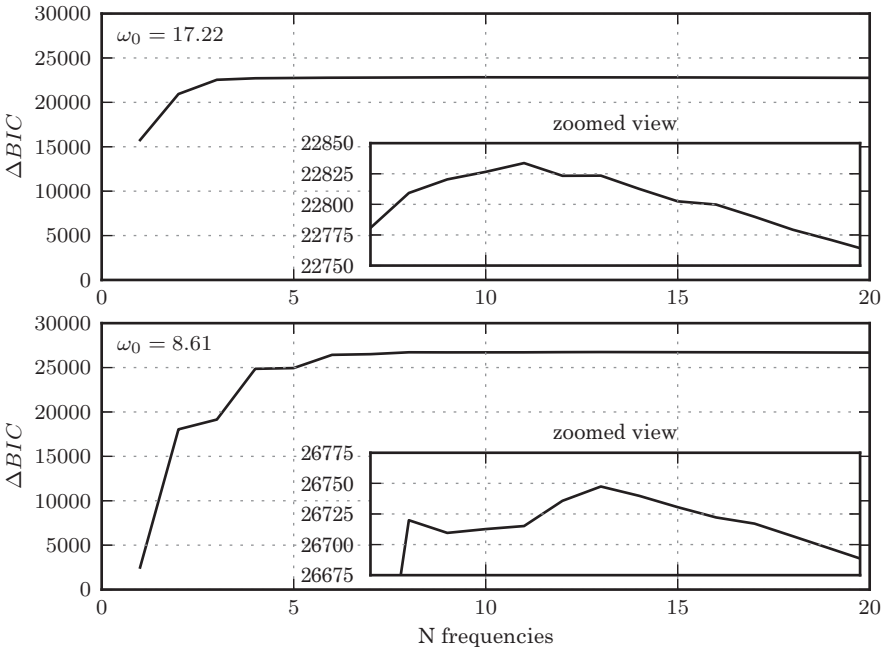

Figure 10.19. BIC as a function of the number of frequency components for the light curve shown in figure 10.18. BIC for the two prominent frequency peaks is shown. The inset panel details the area near the maximum. For both frequencies, the BIC peaks at between 10 and 15 terms; note that a high value of BIC is achieved already with 6 components. Comparing the two, the longer-period model (bottom panel) is much more significant.

to use machine learning methods for classification (as opposed to manual/visual classification). In addition to light curves, other data such as colors are also used in classification.

As discussed in chapters 6 and 9, classification methods can be divided into supervised and unsupervised. With supervised methods we provide a training sample, with labels such as "RR Lyrae," "Algol type," "Cepheid" for each light curve, and then seek to assign these labels to another data set (essentially, we request, "find me more light curves such as this one in the new sample."). With unsupervised methods, we provide a set of attributes and ask if the data set displays clustering in the multidimensional space spanned by these attributes. As practical examples, below we discuss unsupervised clustering and classification of variable stars with light curves found in the LINEAR data set, augmented with photometric (color) data from the SDSS and 2MASS surveys.

The Lomb–Scargle periodogram fits a single harmonic (eq. 10.23). If the underlying time series includes higher harmonics, a more general model than a single sinusoid should be used to better describe the data and obtain a more robust period, as discussed in the preceding section. As an added benefit of the improved modeling, the amplitudes of Fourier terms can be used to efficiently classify light curves; for example, see [29, 41]. In some sense, fitting a low-M Fourier series to data represents an example of the dimensionality reduction techniques discussed in chapter 7. Of course, it is not necessary to use Fourier series and other methods have been proposed, such

as direct analysis of folded light curves using PCA; see [17]. For an application of PCA to analyze light curves measured in several passbands simultaneously, see [55].

Given the best period, $P = 2\pi/\omega_0$, determined from the M-term periodogram $P_M(\omega)$ given by eq. 10.69 (with M either fixed a priori, or determined in each case using BIC/AIC criteria), a model based on the first M Fourier harmonics can be fit to the data,

$$y(t) = b_0 + \sum_{m=1}^{M} a_m \sin(m\omega_0 t) + b_m \cos(m\omega_0 t). \tag{10.74}$$

Since ω_0 is assumed known, this model is linear in terms of $(2M + 1)$ unknown coefficients a_j and b_j and thus the fitting can be performed rapidly (approximate solutions given by eq. 10.72 are typically accurate enough for classification purposes).

Given a_m and b_m, useful attributes for the classification of light curves are the amplitudes of each harmonic

$$A_m = (a_m^2 + b_m^2)^{1/2}, \tag{10.75}$$

and phases

$$\phi_m = \tan^{-1}(b_m, a_m), \tag{10.76}$$

with $-\pi < \phi_m \leq \pi$. It is customary to define the zero phase to correspond to the maximum, or the minimum, of a periodic light curve. This convention can be accomplished by setting ϕ_1 to the desired value (0 or $\pi/2$), and redefining phases of other harmonics as

$$\phi_m^0 = \phi_m - m\,\phi_1. \tag{10.77}$$

It is possible to extend this model to more than one fundamental period, for example, as done by Debosscher et al. in analysis of variable stars [18]. They subtract the best-fit model given by eq. 10.74 from data and recompute the periodogram to obtain next best period, find the best-fit model again, and then once again repeat all the steps to obtain three best periods. Their final model for a light curve is thus

$$y(t) = b_0 + \sum_{k=1}^{3} \sum_{m=1}^{M} a_{km} \left[\sin(m\omega_k t) + b_{km} \cos(m\omega_k t)\right], \tag{10.78}$$

where $\omega_k = 2\pi/P_k$. With three fixed periods, there are $6M + 1$ free parameters to be fit. Again, finding the best-fit parameters is a relatively easy linear regression problem when period(s) are assumed known. This and similar approaches to the classification of variable stars are becoming a standard in the field [2, 44]. A multistaged treelike classification scheme, with explicit treatment of outliers, seems to be an exceptionally powerful and efficient approach, even in the case of sparse data [2, 43].

We now return to the specific example of the LINEAR data (see §1.5.9). Figures 10.20 and 10.21 show the results of a Gaussian mixture clustering analysis which attempts to find self-similar (or compact) classes of about 6,000 objects without using any training sample. The main idea is that different physical classes of objects (different types of variable stars) might be clustered in the multidimensional attribute space. If we indeed identify such clusters, then we can attempt to assign them a physical meaning.

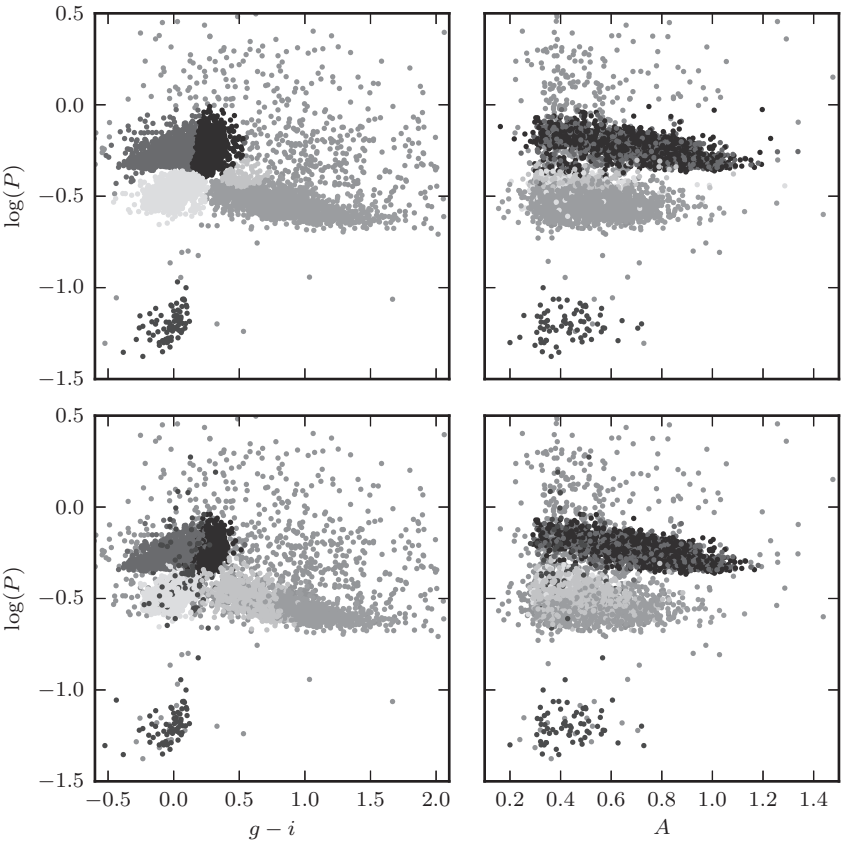

Figure 10.20. Unsupervised clustering analysis of periodic variable stars from the LINEAR data set. The top row shows clusters derived using two attributes ($g-i$ and $\log P$) and a mixture of 12 Gaussians. The colorized symbols mark the six most significant clusters. The bottom row shows analogous diagrams for clustering based on seven attributes (colors $u-g$, $g-i$, $i-K$, and $J-K$; $\log P$, light-curve amplitude, and light-curve skewness), and a mixture of 15 Gaussians. See figure 10.21 for data projections in the space of other attributes for the latter case.

The top panels of figure 10.20 show a 12-component Gaussian mixture fit to only two data attributes, the $g-i$ color and $\log P$, the base 10 logarithm of the best Lomb–Scargle period in days. The number of mixture components is determined by minimizing the BIC. The six most compact clusters are color coded; other clusters attempt to describe the background. Three clusters can be identified with ab- and c-type RR Lyrae stars. Interestingly, ab-type RR Lyrae stars are separated into two clusters. The reason is that the $g-i$ color is a single-epoch color from SDSS that corresponds to a random phase. Since ab-type RR Lyrae stars spend more time close to minimum than to maximum light, when their colors are red compared to colors at maximum light, their color distribution deviates strongly from a Gaussian. The elongated sequence populated by various types of eclipsing binary stars is also split into two clusters because its shape cannot be described by a single Gaussian either.

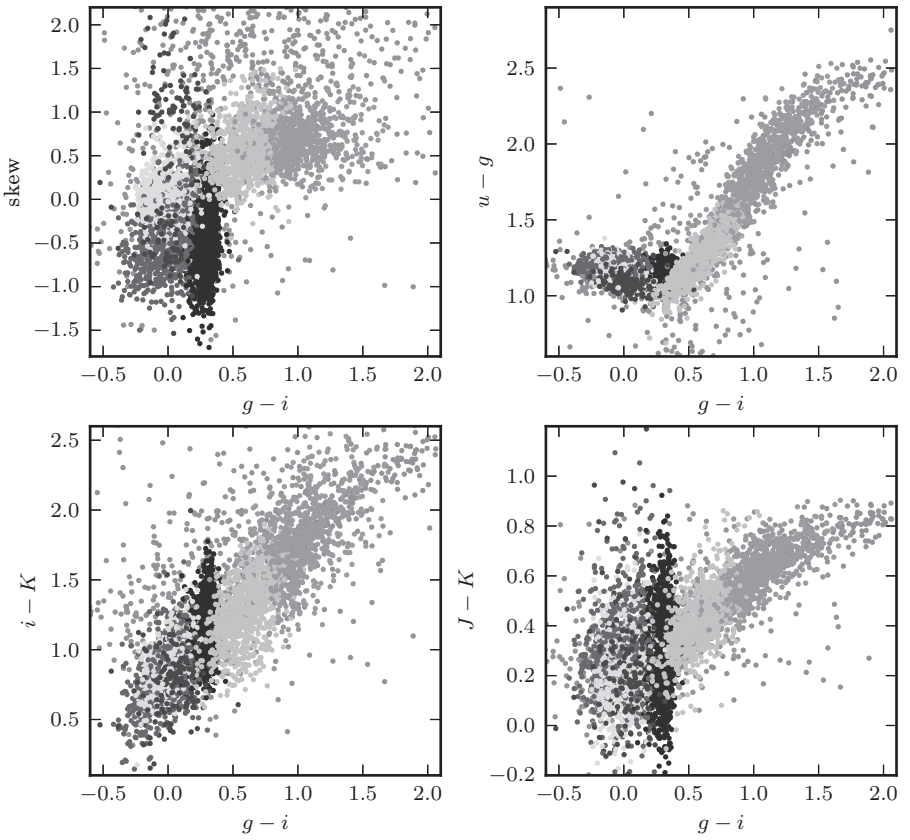

Figure 10.21. Unsupervised clustering analysis of periodic variable stars from the LINEAR data set. Clusters are derived using seven attributes (colors $u-g$, $g-i$, $i-K$, and $J-K$; $\log P$, light-curve amplitude, and light-curve skewness), and a mixture of 15 Gaussians. The $\log P$ vs. $g-i$ diagram and $\log P$ vs. light-curve amplitude diagram for the same clusters are shown in the lower panels of figure 10.20.

The cluster with $\log P < -1$ is dominated by the so-called δ Scu and SX Phe variable stars; see [24]. The upper-right panel shows the clusters in a different projection, $\log P$ vs. light-curve amplitude. The top four clusters are still fairly well localized in this projection due to $\log P$ carrying significant discriminative power, but there is some mixing between the background and the clusters.

The bottom panels of figure 10.20 show a 15-component Gaussian mixture fit to seven data attributes. The number of mixture components is again determined by minimizing BIC. The clustering attributes include four photometric colors based on SDSS and 2MASS measurements ($u-g$, $g-i$, $i-K$, $J-K$) and three parameters determined from the LINEAR light curve data ($\log P$, amplitude, and light-curve skewness). The clusters derived from all seven attributes are remarkably similar to the clusters derived from just two attributes: this shows that the additional data adds very little new information. Figure 10.21 shows the locations of these clusters in the space of other attributes. The means and standard deviations of the distribution

TABLE 10.1.
The means and standard deviations for each cluster and each attribute.

	$u-g$	$g-i$	$i-K$	$J-K$	$\log(P)$	amplitude	skew
1	1.19 ± 0.04	-0.00 ± 0.11	0.87 ± 0.18	0.22 ± 0.13	-0.47 ± 0.06	0.43 ± 0.06	0.00 ± 0.26
2	1.16 ± 0.04	0.31 ± 0.04	1.16 ± 0.17	0.29 ± 0.15	-0.24 ± 0.05	0.70 ± 0.17	-0.45 ± 0.38
3	1.19 ± 0.13	0.52 ± 0.12	1.22 ± 0.22	0.39 ± 0.11	-0.47 ± 0.06	0.41 ± 0.07	0.51 ± 0.30
4	1.20 ± 0.04	0.05 ± 0.16	0.88 ± 0.22	0.30 ± 0.17	-0.24 ± 0.05	0.70 ± 0.18	-0.44 ± 0.36
5	1.63 ± 0.29	0.88 ± 0.20	1.58 ± 0.28	0.55 ± 0.11	-0.56 ± 0.05	0.54 ± 0.13	0.68 ± 0.25
6	1.11 ± 0.06	-0.01 ± 0.12	0.88 ± 0.28	0.33 ± 0.31	-0.99 ± 0.34	0.47 ± 0.12	-0.30 ± 0.44

of points assigned to each cluster for the seven-attribute clustering are shown in table 10.1.

As is evident from visual inspection of figures 10.20 and 10.21, the most discriminative attribute is the period. Clusters 2 and 3, which have very similar period distributions, are separated by the $g-i$ and $i-K$ colors, which are a measure of the star's effective temperature; see [11].

To contrast this unsupervised clustering with classification based on a training sample, light-curve types for the LINEAR data set based on visual (manual) classification by domain experts are utilized with two machine learning methods: a Gaussian mixture model Bayes classifier (GMMB, see §9.3.5) and support vector machines (SVM, see §9.6). As above, both two-attribute and seven-attribute cases are considered. Figure 10.22 shows GMMB results and figure 10.23 shows SVM results. The training sample of 2,000 objects includes five input classes and the methods assign the most probable classification, among these five classes, to each object not in the training sample (about 4,000). Since the input expert classification is known for all 6,000 objects, it can be used to compute completeness and contamination for automated methods, as summarized in tables 10.2 and 10.3.

The performances of GMMB and SVM methods are very similar and only a quantitative analysis of completeness and contamination listed in these tables reveals some differences. Both methods assign a large fraction of the EA class (Algol-type eclipsing binaries) to the EB/EW class (contact binaries). This is not necessarily a problem with automated methods because these two classes are hard to distinguish even by experts. While GMMB achieves a higher completeness for c-type RR Lyrae and EB/EW classes, SVM achieves a much higher completeness for SX Phe class. Therefore, neither method is clearly better, but both achieve reasonably satisfactory performance level relative to domain expert classification.

10.3.5. Analysis of Arrival Time Data

Discussion of periodic signals in the preceding sections assumed that the data included a set of N data points $(t_1, y_1), \ldots, (t_N, y_N)$, $j = 1, \ldots, N$ with known errors for y. An example of such a data set is the optical light curve of an astronomical source where many photons are detected and the measurement error is typically dominated by photon counting statistics and background noise. Very different data sets are collected at X-ray and shorter wavelengths where individual photons are detected and background contamination is often negligible. In such cases, the data

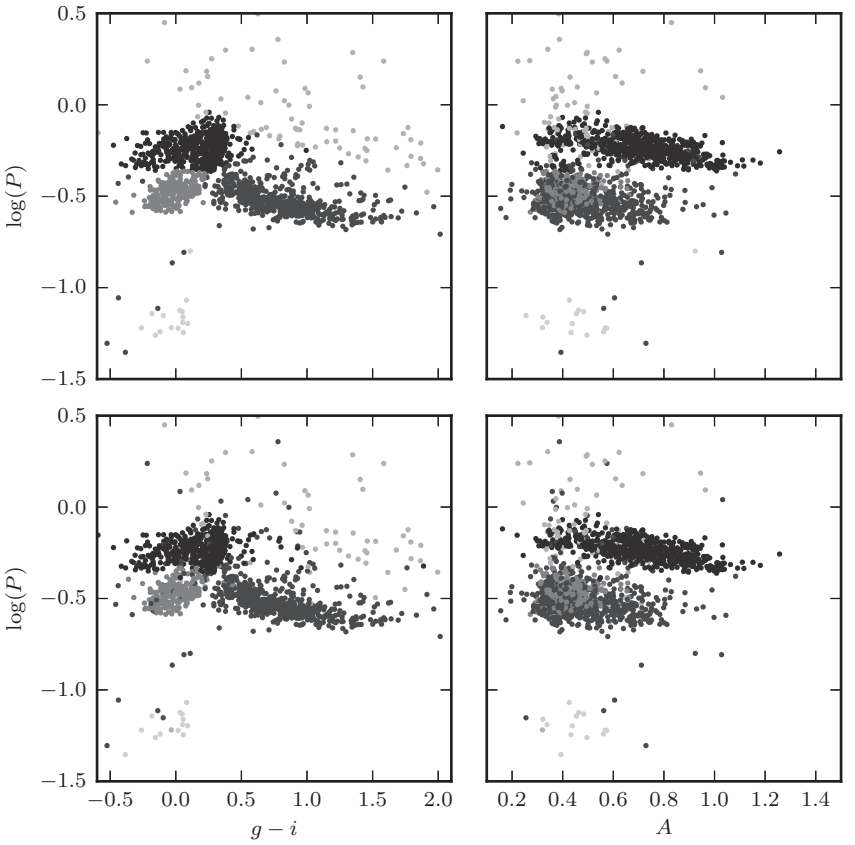

Figure 10.22. Supervised classification of periodic variable stars from the LINEAR data set using Gaussian mixture model Bayes classifier. The training sample includes five input classes. The top row shows clusters derived using two attributes ($g-i$ and $\log P$) and the bottom row shows analogous diagrams for classification based on seven attributes (colors $u-g$, $g-i$, $i-K$, and $J-K$; $\log P$; light-curve amplitude; and light-curve skewness). See table 10.2 for the classification performance.

set consists of the arrival times of individual photons, $t_1, \ldots, t_N$, $j = 1, \ldots, N$, where it can be typically assumed that errors are negligible. Given such a data set, how do we search for a periodic signal, and more generally, how do we test for any type of variability?

The best known classical test for variability in arrival time data is the Rayleigh test, and it bears some similarity to the analysis of periodograms (its applicability goes far beyond this context). Given a trial period, the phase ϕ_j corresponding to each datum is evaluated using eq. 10.21 and the following statistic is formed:

$$R^2 = \left(\sum_{j=1}^{N} \cos(2\pi\phi_j) \right)^2 + \left(\sum_{j=1}^{N} \sin(2\pi\phi_j) \right)^2 . \tag{10.79}$$

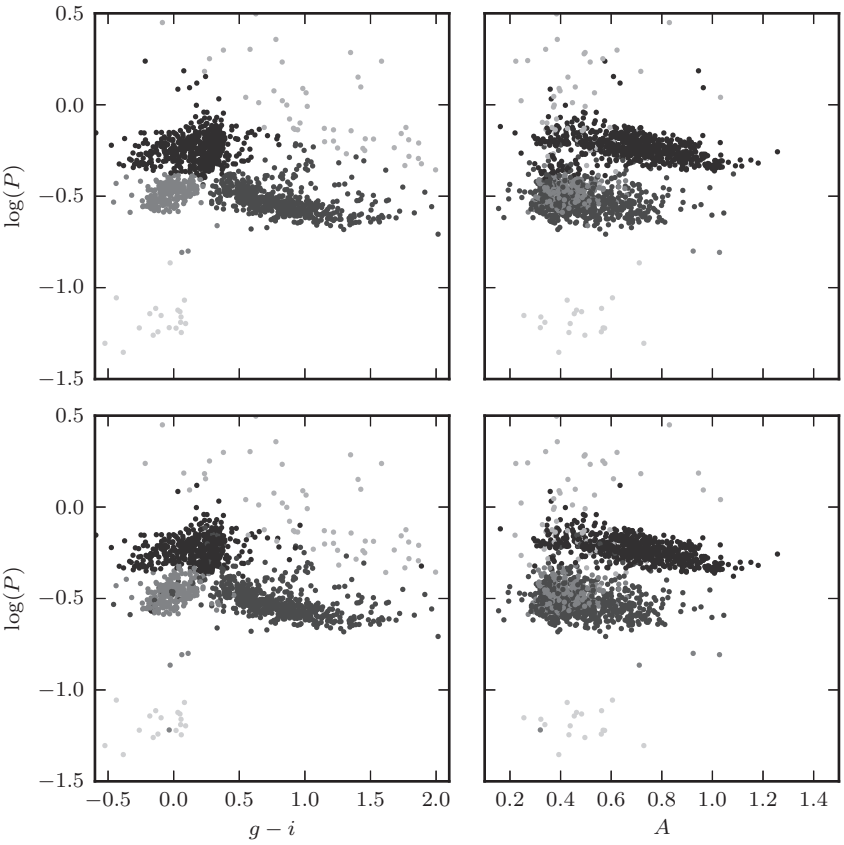

Figure 10.23. Supervised classification of periodic variable stars from the LINEAR data set using support vector machines method. The training sample includes five input classes. The top row shows clusters derived using two attributes ($g-i$ and $\log P$) and the bottom row shows analogous diagrams for classification based on seven attributes (colors $u-g$, $g-i$, $i-K$, and $J-K$; $\log P$; light-curve amplitude; and light-curve skewness). See table 10.3 for the classification performance.

This expression can be understood in terms of a random walk, where each angle ϕ_j defines a unit vector, and R is the length of the resulting vector. For random data, R^2 is small, and for periodic data, R^2 is large, when the correct period is chosen. Similarly to the analysis of the Lomb–Scargle periodogram, R^2 is evaluated for a grid of P, and the best period is chosen as the value that maximizes R^2. For $N > 10$, $2R^2/N$ is distributed as χ^2 with two degrees of freedom (this easily follows from the random walk interpretation), and this fact can be used to assess the significance of the best-fit period (i.e., the probability that a value that large would happen by chance when the signal is stationary). A more detailed discussion of classical tests can be found in [14].

An alternative solution to this problem was derived by Gregory and Loredo [28] and here we will retrace their analysis. First, we divide the time interval $T = t_N - t_1$ into many arbitrarily small steps, Δt, so that each interval contains either 1 or 0 detections. If the event rate (e.g., the number of photons per unit time) is $r(t)$, then the

TABLE 10.2.
The performance of supervised classification using Gaussian mixture model Bayes classifier. Each row corresponds to an input class listed in the first column (ab RRL: ab-type RR Lyrae; c RRL: c-type RR Lyrae; EA: Algol-type eclipsing binaries; EB/EW: contact eclipsing binaries; SX Phe: SX Phe and δ Scu candidates). The second column lists the number of objects in each input class, and the remaining columns list the percentage of sources classified into classes listed in the top row. The bottom row lists classification contamination in percent for each class listed in the top row.

class	N	ab RRL	c RRL	EA	EB/EW	SX Phe
ab RRL	1772	96.4	0.5	1.2	1.9	0.0
c RRL	583	0.2	95.3	0.2	4.3	0.0
EA	228	1.3	0.0	63.2	35.5	0.0
EB/EW	1507	0.7	0.8	2.9	95.6	0.0
SX Phe	56	0.0	0.0	0.0	26.8	73.2
contam.	—	0.9	3.6	31.4	9.7	0.0

TABLE 10.3.
The performance of supervised classification using support vector machines method. Each row corresponds to an input class listed in the first column (ab RRL: ab-type RR Lyrae; c RRL: c-type RR Lyrae; EA: Algol-type eclipsing binaries; EB/EW: contact eclipsing binaries; SX Phe: SX Phe and δ Scu candidates). The second column lists the number of objects in each input class, and the remaining columns list the percentage of sources classified into classes listed in the top row. The bottom row lists classification contamination in percent for each class listed in the top row.

class	N	ab RRL	c RRL	EA	EB/EW	SX Phe
ab RRL	1772	95.9	0.3	1.4	2.4	0.0
c RRL	583	1.5	91.3	0.2	7.0	0.0
EA	228	5.3	1.3	67.5	25.9	0.0
EB/EW	1507	2.1	4.0	3.1	90.7	0.1
SX Phe	56	0.0	1.8	0.0	1.8	96.4
contam.	—	3.0	11.6	31.6	9.5	1.8

expectation value for the number of events during Δt is

$$\mu(t) = r(t)\,\Delta t. \tag{10.80}$$

Following the Poisson statistics, the probability of detecting no events during Δt is

$$p(0) = e^{-r(t)\,\Delta t}, \tag{10.81}$$

and the probability of detecting a single event is

$$p(1) = r(t)\,\Delta t\, e^{-r(t)\,\Delta t}. \tag{10.82}$$

The data likelihood becomes

$$p(D|r,I) = (\Delta t)^N\, e^{-\int_{(T)} r(t)\,dt} \prod_{j=1}^{N} r(t_j). \tag{10.83}$$

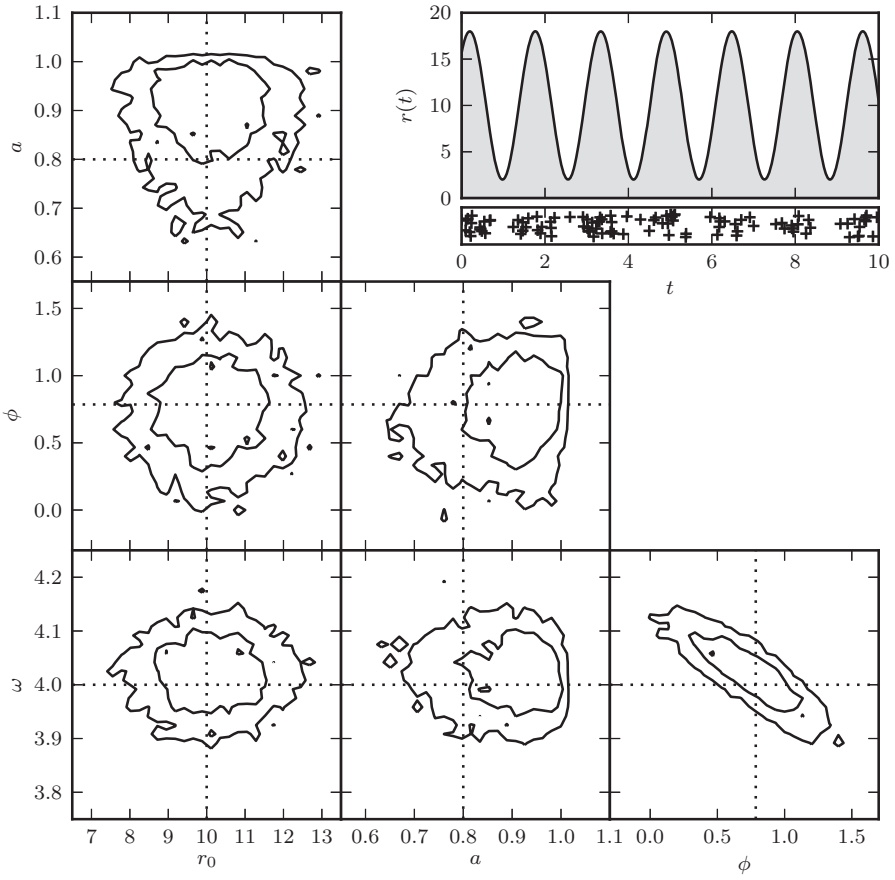

Figure 10.24. Modeling time-dependent flux based on arrival time data. The top-right panel shows the rate $r(t) = r_0[1 + a\sin(\omega t + \phi)]$, along with the locations of the 104 detected photons. The remaining panels show the model contours calculated via MCMC; dotted lines indicate the input parameters. The likelihood used is from eq. 10.83. Note the strong covariance between ϕ and ω in the bottom-right panel.

The integral of $r(t)$ over time should be performed only over the intervals when the data were collected. For simplicity, hereafter we assume that the data were collected in a single stretch of time with no gaps.

With an appropriate choice of model, and priors for the model parameters, analysis of arrival time data is no different than any other model selection and parameter estimation problem. For example, figure 10.24 shows the posterior pdf for a model based on periodic $r(t)$ and arrival times for 104 photons. Input model parameters and their uncertainties are easily evaluated using MCMC and the data likelihood from eq. 10.83 (though MCMC may not be necessary in such a low-dimensional problem).

Instead of fitting a parametrized model, such as a Fourier series, Gregory and Loredo used a nonparametric description of the rate function $r(t)$. They described the shape of the phased light curve using a piecewise constant function, f_j, with M

steps of the same width, and $\sum_j f_j = 1$. The rate is described as

$$r(t_j) \equiv r_j = M A f_j, \tag{10.84}$$

where A is the average rate, and bin j corresponding to t_j is determined from the phase corresponding to t_j and the trial period. In addition to the frequency ω (or period), phase offset, and the average rate A, their model includes $M - 1$ parameters f_j (not M, because of the normalization constraint). They marginalize the resulting pdf to produce an analog of the periodogram, expressions for computing the model odds ratio for signal detection, and for estimating the light-curve shape. In the case when little is known about the signal shape, this method is superior to the more popular Fourier series expansion.

Bayesian blocks

Scargle has developed a nonparametric Bayesian method similar in spirit to the Gregory–Loredo method for treating arrival time data from periodic time series [47, 48]. Scargle's method works with both arrival time and binned data, and can be used to detect bursts and characterize their shapes. The algorithm produces the most probable segmentation of the observation into time intervals during which the signal has no statistically significant variations, dubbed *Bayesian blocks* (we previously discussed Bayesian blocks in the context of histograms; see §5.7.2). In this case, the underlying model is a piecewise constant variation, and the position and number of blocks is determined by data (blocks are not of uniform duration). In some sense, Bayesian blocks improves on the simple idea of the phase dispersion minimization method by using nonuniform adaptive bins.

10.4. Temporally Localized Signals

A case frequently encountered in practice is a stationary signal with an event localized in time. Astronomical examples include the magnification due to gravitational microlensing, bursts of emission (where the source brightness increases and then decreases to the original level over a finite time interval), and the signature of a gravitational wave in data from LIGO (and other gravitational wave detectors).

When the noise properties are understood, and the expected shape of the signal is known, a tool of choice is *full forward* modeling. That is, here too the analysis includes model selection and parameter estimation steps, and is often called a *matched filter* search. Even when the shape of the matched filter is not known, it can be treated in a nonparametrized form as was discussed in the context of arrival time data. Similarly, even when a full understanding of the noise is missing, it is possible to marginalize over unknown noise when some weak assumptions are made about its properties (recall the example from §5.8.5).

We will discuss two simple parametric models here: a burst signal and a chirp signal. In both examples we assume Gaussian known errors. The generalization to nonparametric models and more complex models can be relatively easily implemented by modifying the code developed for these two examples.

10.4.1. Searching for a Burst Signal

Consider a model where the signal is stationary, $y(t) = b_0 + \epsilon$, and at some unknown time, T, it suddenly increases, followed by a decay to the original level b_0 over some unknown time period. Let us describe such a burst by

$$y_B(t|T, A, \boldsymbol{\theta}) = A\, g_B(t - T|\boldsymbol{\theta}), \tag{10.85}$$

where the function g_B describes the shape of the burst signal ($g_B(t < 0) = 0$). This function is specified by a vector of parameters $\boldsymbol{\theta}$ and can be analytic, tabulated in the form of a template, or treated in a nonparametric form. Typically, MCMC methods are used to estimate model parameters.

For illustration, we consider here a case with $g_B(t|\alpha) = \exp(-\alpha t)$. Figure 10.25 shows the simulated data and projections of posterior pdf for the four model parameters (b_0, T, A, and α). Other models for the burst shape can be readily analyzed using the same code with minor modifications.

Alternatively, the burst signal could be treated in the case of arrival time data, using the approach outlined in §10.3.5. Here, the rate function is not periodic, and can be obtained as $r(t) = (\Delta t)^{-1} y(t)$, where $y(t)$ is the sum of the stationary signal and the burst model (eq. 10.85).

10.4.2. Searching for a Chirp Signal

Here we consider a chirp signal, added to a stationary signal b_0,

$$y(t) = b_0 + A \sin[\omega t + \beta t^2], \tag{10.86}$$

and analyze it using essentially the same code as for the burst signal. Figure 10.26 shows the simulated data and projections of posterior pdf for the four model parameters (b_0, A, ω, and β). Note that here the second term in the argument of the sine function above (βt^2) produces the effect of increasing frequency in the signal seen in the top-right panel. The resulting fit shows a strong inverse correlation between β and ω. This is expected because they both act to increase the frequency: starting from a given model, slightly increasing one while slightly decreasing the other leads to a very similar prediction.

Figure 10.27 illustrates a more complex ten-parameter case of chirp modeling. The chirp signal is temporally localized and it decays exponentially for $t > T$:

$$y_C(t|T, A, \phi, \omega, \beta) = A\ \sin[\phi + \omega(t - T) + \beta(t - T)^2]\ \exp[-\alpha(t - T)]. \tag{10.87}$$

The signal in the absence of chirp is taken as

$$y(t) = b_0 + b_1 \sin(\Omega_1 t) \sin(\Omega_2 t). \tag{10.88}$$

Here, we can consider parameters A, ω, β, and α as "interesting," and other parameters can be treated as "nuisance." Despite the model complexity, the MCMC-based analysis is not much harder than in the first simpler case, as illustrated in figure 10.28.

In both examples of a matched filter search for a signal, we assumed white Gaussian noise. When noise power spectrum is not flat (e.g., in the case of LIGO data; see figure 10.6), the analysis becomes more involved. For signals that are localized

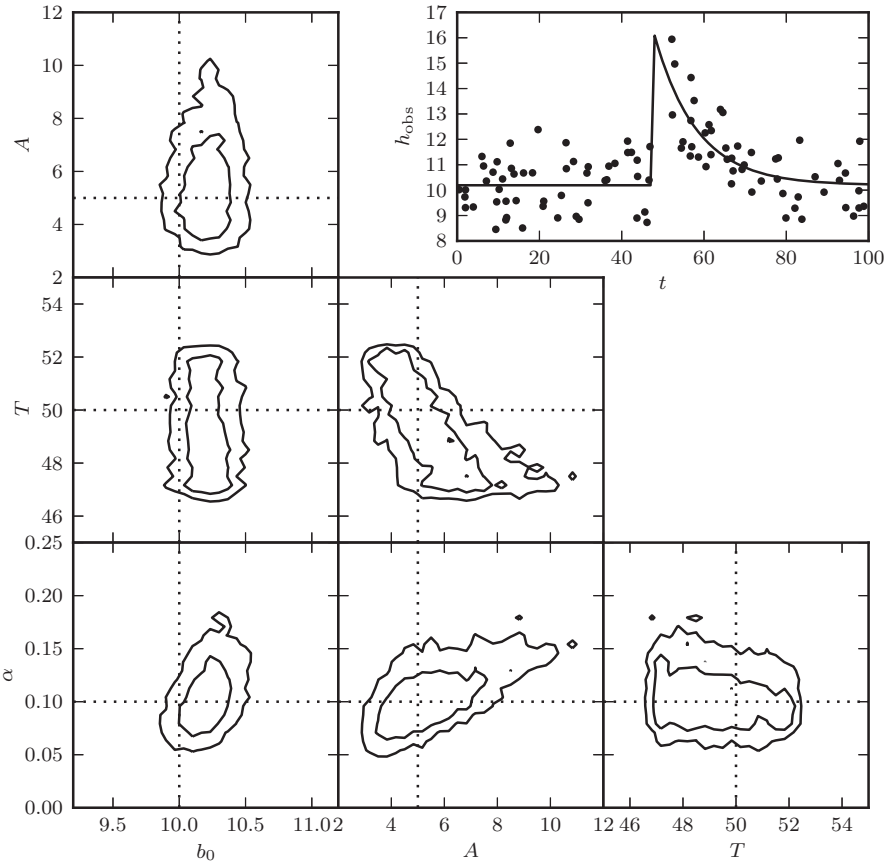

Figure 10.25. A matched filter search for a burst signal in time series data. A simulated data set generated from a model of the form $y(t) = b_0$ for $t < T$ and $y = b_0 + A \exp[-\alpha(t - T)]$ for $t > T$, with homoscedastic Gaussian errors with $\sigma = 1$, is shown in the top-right panel. The posterior pdf for the four model parameters is determined using MCMC and shown in the other panels.

not only in time, but in frequency as well, the wavelet-based analysis discussed in §10.2.4 is a good choice. A simple example of such an analysis is shown in figure 10.28. The two-dimensional wavelet-based PSD easily recovers the increase of characteristic chirp frequency with time. To learn more about such types of analysis, we refer the reader to the rapidly growing body of tools and publications developed in the context of gravitational wave analysis.[6]

10.5. Analysis of Stochastic Processes

Stochastic variability includes behavior that is not predictable forever as in the periodic case, but unlike temporally localized events, variability is always there. Typically, the underlying physics is so complex that we cannot deterministically predict future

[6]See, for example, http://www.ligo.org/

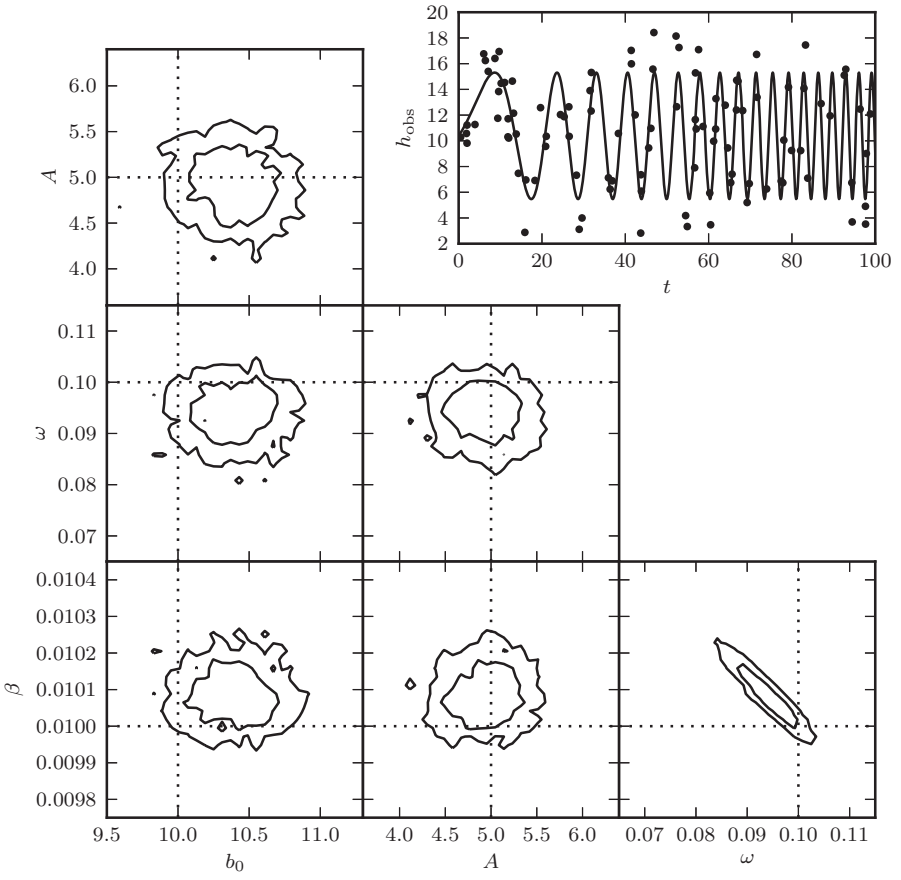

Figure 10.26. A matched filter search for a chirp signal in time series data. A simulated data set generated from a model of the form $y = b_0 + A\sin[\omega t + \beta t^2]$, with homoscedastic Gaussian errors with $\sigma = 2$, is shown in the top-right panel. The posterior pdf for the four model parameters is determined using MCMC and shown in the other panels.

values (i.e., the stochasticity is inherent in the process, rather than due to measurement noise). Despite their seemingly irregular behavior, stochastic processes can be quantified, too, as briefly discussed in this section. References to more in-depth literature on stochastic processes are listed in the final section.

10.5.1. The Autocorrelation and Structure Functions

One of the main statistical tools for the analysis of stochastic variability is the autocorrelation function. It represents a specialized case of the correlation function of two functions, $f(t)$ and $g(t)$, scaled by their standard deviations, and defined at time lag Δt as

$$\mathrm{CF}(\Delta t) = \frac{\lim_{T\to\infty} \frac{1}{T} \int_{(T)} f(t)\, g(t+\Delta t)\, dt}{\sigma_f\, \sigma_g}, \tag{10.89}$$

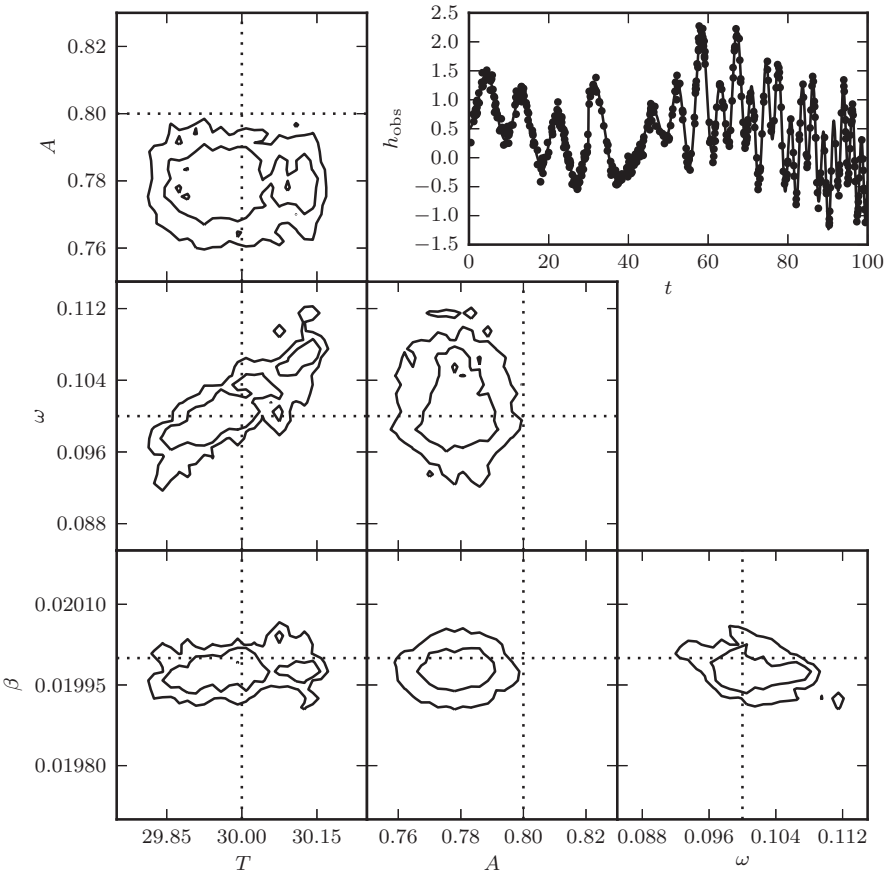

Figure 10.27. A ten-parameter chirp model (see eq. 10.87) fit to a time series. Seven of the parameters can be considered nuisance parameters, and we marginalize over them in the likelihood contours shown here.

where σ_f and σ_g are standard deviations of $f(t)$ and $g(t)$, respectively. With this normalization, the correlation function is unity for $\Delta t = 0$ (without normalization by standard deviation, the above expression is equal to the covariance function). It is assumed that both f and g are statistically weakly stationary functions, which means that their mean and autocorrelation function (see below) do not depend on time (i.e., they are statistically the same irrespective of the time interval over which they are evaluated). The correlation function yields information about the time delay between two processes. If one time series is produced from another one by simply shifting the time axis by $t_{\rm lag}$, their correlation function has a peak at $\Delta t = t_{\rm lag}$.

With $f(t) = g(t) = y(t)$, the autocorrelation of $y(t)$ defined at time lag Δt is

$$\mathrm{ACF}(\Delta t) = \frac{\lim_{T\to\infty} \frac{1}{T} \int_{(T)} y(t)\, y(t+\Delta t)\, dt}{\sigma_y^2}. \tag{10.90}$$

The autocorrelation function yields information about the variable timescales present in a process. When y values are uncorrelated (e.g., due to white noise without any

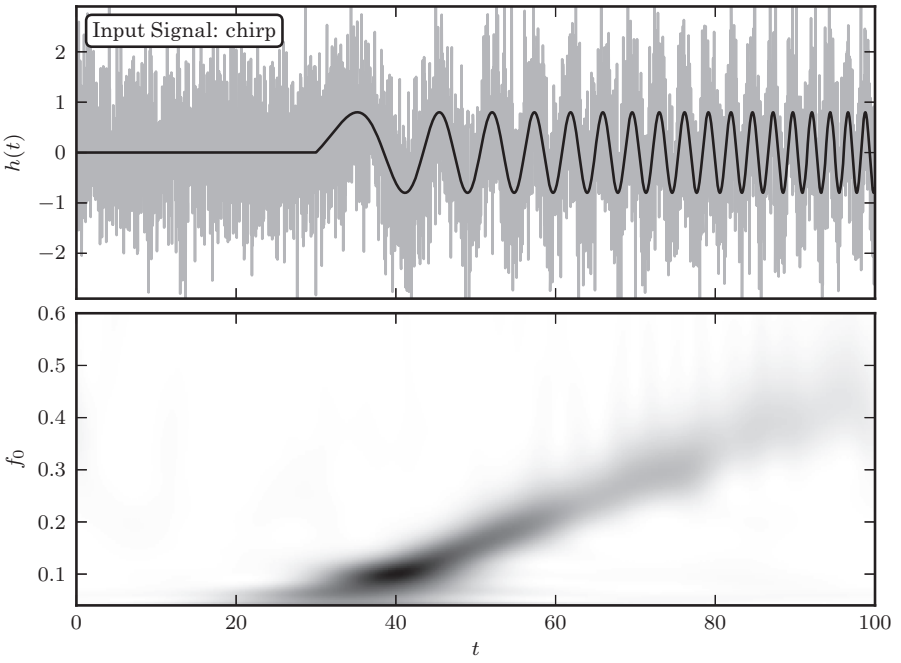

Figure 10.28. A wavelet PSD of the ten-parameter chirp signal similar to that analyzed in figure 10.27. Here, the signal with an amplitude of $A = 0.8$ is sampled in 4096 evenly spaced bins, and with Gaussian noise with $\sigma = 1$. The two-dimensional wavelet PSD easily recovers the increase of characteristic chirp frequency with time.

signal), $\mathrm{ACF}(\Delta t) = 0$, except for ACF(0) =1. For processes that "retain memory" of previous states only for some characteristic time τ, the autocorrelation function vanishes for $\Delta t \gg \tau$. In other words, the predictability of future behavior for such a process is limited to times up to $\sim\tau$. One such process is *damped random walk*, discussed in more detail in §10.5.4.

The autocorrelation function and the PSD of function $y(t)$ (see eq. 10.6) are Fourier pairs; this fact is known as the Wiener–Khinchin theorem and applies to stationary random processes. The former represents an analysis method in the time domain, and the latter in the frequency domain. For example, for a periodic process with a period P, the autocorrelation function oscillates with the same period, while for processes that retain memory of previous states for some characteristic time τ, ACF drops to zero for $t \sim \tau$.

The structure function is another quantity closely related to the autocorrelation function,

$$\mathrm{SF}(\Delta t) = \mathrm{SF}_\infty \left[1 - \mathrm{ACF}(\Delta t)\right]^{1/2}, \tag{10.91}$$

where SF_∞ is the standard deviation of the time series evaluated over an infinitely large time interval (or at least much longer than any characteristic timescale τ). The structure function, as defined by eq. 10.91, is equal to the standard deviation of the distribution of the difference of $y(t_2) - y(t_1)$ evaluated at many different t_1 and t_2 such that time lag $\Delta t = t_2 - t_1$, and divided by $\sqrt{2}$ (because of differencing). When the structure function $\mathrm{SF} \propto t^\alpha$, then $\mathrm{PSD} \propto 1/f^{(1+2\alpha)}$. In the statistics literature, the

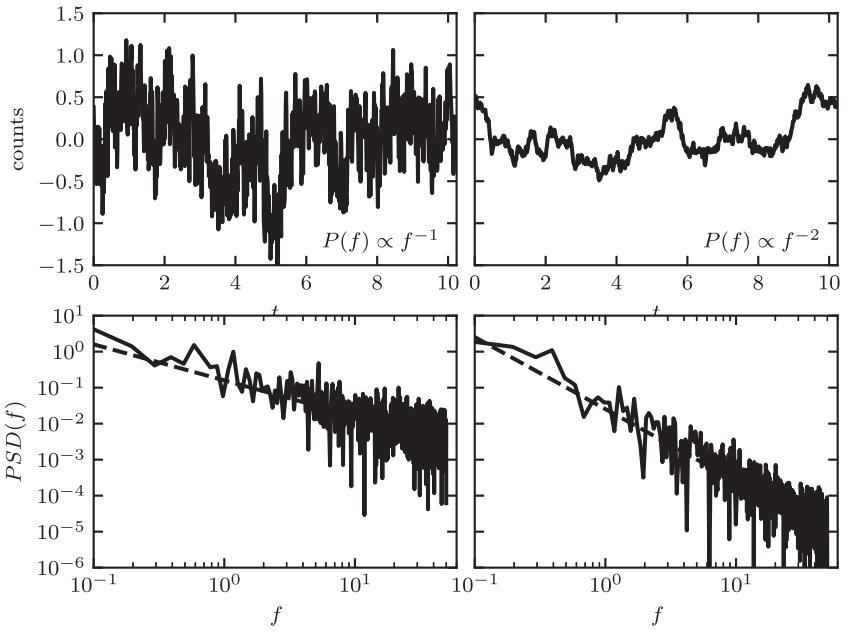

Figure 10.29. Examples of stochastic time series generated from power-law PSDs (left: $1/f$; right: $1/f^2$) using the method from [56]. The top panels show the generated data, while the bottom panels show the corresponding PSD (dashed lines: input PSD; solid lines: determined from time series shown in the top panels).

structure function given by eq. 10.91 is called the *second-order structure function* (or *variogram*) and is defined without the square root (e.g., see FB2012). Although the early use in astronomy followed the statistics literature, for example, [52], we follow here the convention used in recent studies of quasar variability, for example, [58] and [16] (the appeal of taking the square root is that SF then has the same unit as the measured quantity). Note, however, that definitions in the astronomical literature are not consistent regarding the $\sqrt{2}$ factor discussed above.

Therefore, a stochastic time series can be analyzed using the autocorrelation function, the PSD, or the structure function. They can reveal the statistical properties of the underlying process, and distinguish processes such as white noise, random walk (see below), and damped random walk (discussed in §10.5.4). They are mathematically equivalent and all are used in practice; however, due to issues of noise and sampling, they may not always result in equivalent inferences about the data.

Examples of Stochastic Processes: $1/f$ and $1/f^2$ Processes

For a given autocorrelation function or PSD, the corresponding time series can be generated using the algorithm described in [56]. Essentially, the amplitude of the Fourier transform is given by the PSD, and phases are assigned randomly; the inverse Fourier transform then generates time series.

The connection between the PSD and the appearance of time series is illustrated in figure 10.29 for two power-law PSDs: $1/f$ and $1/f^2$. The PSD normalization is such that both cases have similar power at low frequencies. For this reason, the

overall amplitudes (more precisely, the variances) of the two time series are similar. The power at high frequencies is much larger for the $1/f$ case, and this is why the corresponding time series has the appearance of noisy data (the top-left panel in figure 10.29). The structure function for the $1/f$ process is constant, and proportional to $t^{1/2}$ for the $1/f^2$ process (remember that we defined structure function with a square root).

The $1/f^2$ process is also known as *Brownian motion* or *random walk* (or *drunkard's walk*). For an exellent introduction from a physicist's perspective, see [26]. Processes whose PSD is proportional to $1/f$ are sometimes called *long-term memory processes* (mostly in the statistical literature), *flicker noise* and *red noise*. The latter is not unique as sometimes the $1/f^2$ process is called *red noise*, while the $1/f$ process is then called *pink noise*. The $1/f$ processes have infinite variance; the variance of an observed time series of a finite length increases logarithmically with the length (for more details, see [42]). Similarly to the behavior of the mean for Cauchy distribution (see §5.6.3), the variance of the mean for the $1/f$ process does not decrease with the sample size. Another practical problem with the $1/f$ process is that the Fourier transform of its autocovariance function does not produce a reliable estimate of the power spectrum in the distribution's tail. Difficulties with estimating properties of power-law distributions (known as *Pareto distribution* in the statistics literature) in general cases (i.e., not only in the context of time series analysis) are well summarized in [10].

AstroML includes a routine which generates power-law light curves based on the method of [56]. It can be used as follows:

```
>>> import numpy as np
>>> from astroML.time_series import generate_power_law

# beta gives the power-law index: P ~ f^-beta
>>> y = generate_power_law(N=1024, dt=0.01, beta=2)
```

This routine is used to generate the data shown in figure 10.29.

10.5.2. Autocorrelation and Structure Function for Evenly and Unevenly Sampled Data

In the case of evenly sampled data, with $t_i = (i-1)\Delta t$, the autocorrelation function of a discretely sampled $y(t)$ is defined as

$$\mathrm{ACF}(j) = \frac{\sum_{i=1}^{N-j}\left[(y_i - \bar{y})\,(y_{i+j} - \bar{y})\right]}{\sum_{i=1}^{N}(y_i - \bar{y})^2}. \tag{10.92}$$

With this normalization the autocorrelation function is dimensionless and ACF(0) $= 1$. The normalization by variance is sometimes skipped (see [46]), in which case a more appropriate name is the *covariance function*.

When a time series has a nonvanishing ACF, the uncertainty of its mean is larger than for an uncorrelated data set (cf. eq. 3.34),

$$\sigma_{\overline{x}} = \frac{\sigma}{\sqrt{N}} \left[1 + 2 \sum_{j=1}^{N} \left(1 - \frac{j}{N} \right) \mathrm{ACF}(j) \right]^{1/2}, \tag{10.93}$$

where σ is the homoscedastic measurement error. This fact is often unjustifiably neglected in analysis of astronomical data.

When data are unevenly sampled, the ACF cannot be computed using eq. 10.92. For the case of unevenly sampled data, Edelson and Krolik [22] proposed the *discrete correlation function* (DCF) in an astronomical context (called the *slot autocorrelation function* in physics). For discrete unevenly sampled data with homoscedastic errors, they defined a quantity

$$\mathrm{UDCF}_{ij} = \frac{(y_i - \overline{y})\,(g_j - \overline{g})}{\left[(\sigma_y^2 - e_y^2)\,(\sigma_g^2 - e_g^2) \right]^{1/2}}, \tag{10.94}$$

where e_y and e_g are homoscedastic measurement errors for time series y and g. The associated time lag is $\Delta t_{ij} = t_i - t_j$. The discrete correlation function at time lag Δt is then computed by binning and averaging UDCF_{ij} over M pairs of points for which $\Delta t - \delta t/2 \leq \Delta t_{ij} \leq \Delta t + \delta t/2$, where δt is the bin size. The bin size is a trade-off between accuracy of $\mathrm{DCF}(\Delta t)$ and its resolution. Edelson and Krolik showed that even uncorrelated time series will produce values of the cross-correlation $\mathrm{DCF}(\Delta t) \sim \pm 1/\sqrt{M}$.

With its binning, this method is similar to procedures for computing the structure function used in studies of quasar variability [15, 52]. The main downside of the DCF method is the assumption of homoscedastic error. Nevertheless, heteroscedastic errors can be easily incorporated by first computing the structure function, and then obtaining the ACF using eq. 10.91. The structure function is equal to the intrinsic distribution width divided by $\sqrt{2}$ for a bin of Δt_{ij} (just as when computing the DCF above). This width can be estimated for heteroscedastic data using eq. 5.69, or the corresponding exact solution given by eq. 5.64.

Scargle has developed different techniques to evaluate the discrete Fourier transform, correlation function and autocorrelation function of unevenly sampled time series (see [46]). In particular, the discrete Fourier transform for unevenly sampled data and the Wiener–Khinchin theorem are used to estimate the autocorrelation function. His method also includes a prescription for correcting the effects of uneven sampling, which results in leakage of power to nearby frequencies (the so-called *sidelobe effect*). Given an unevenly sampled time series, $y(t)$, the essential steps of Scargle's procedure are as follows:

1. Compute the generalized Lomb–Scargle periodogram for $y(t_i)$, $i = 1, \ldots, N$, namely $P_{\mathrm{LS}}(\omega)$.
2. Compute the sampling window function using the generalized Lomb–Scargle periodogram using $z(t_i) = 1$, $i = 1, \ldots, N$, namely $P_{\mathrm{LS}}^{W}(\omega)$.
3. Compute inverse Fourier transforms for $P_{\mathrm{LS}}(\omega)$ and $P_{\mathrm{LS}}^{W}(\omega)$, namely $\rho(t)$ and $\rho^{W}(t)$, respectively.
4. The autocorrelation function at lag t is $\mathrm{ACF}(t) = \rho(t)/\rho^{W}(t)$.

AstroML includes tools for computing the ACF using both Scargle's method and the Edelson and Krolik method:

```
>>> import numpy as np
>>> from astroML.time_series import generate_damped_RW
>>> from astroML.time_series import ACF_scargle, ACF_EK

>>> t = np.arange(0, 1000)
>>> y = generate_damped_RW(t, tau=300)
>>> dy = 0.1
>>> y = np.random.normal(y, dy)

# Scargle's method
>>> ACF, bins = ACF_scargle(t, y, dy)

# Edelson-Krolik method
>>> ACF, ACF_err, bins = ACF_EK(t, y, dy)
```

For more detail, see the source code of figure 10.30.

Figure 10.30 illustrates the use of Edelson and Krolik's DCF method and the Scargle method. They produce similar results; errors are easier to compute for the DCF method and this advantage is crucial when fitting models to the autocorrelation function.

Another approach to estimating the autocorrelation function is direct modeling of the correlation matrix, as discussed in the next section.

10.5.3. Autoregressive Models

Autocorrelated time series can be analyzed and characterized using stochastic *autoregressive models*. Autoregressive models provide a good general description of processes that "retain memory" of previous states (but are not periodic). An example of such a model is the random walk, where each new value is obtained by adding noise to the preceding value:

$$y_i = y_{i-1} + e_i. \tag{10.95}$$

When y_{i-1} is multiplied by a constant factor greater than 1, the model is known as a *geometric random walk* model (used extensively to model stock market data). The noise need not be Gaussian; white noise consists of uncorrelated random variables with zero mean and constant variance, and Gaussian white noise represents the most common special case of white noise.

The random walk can be generalized to the *linear autoregressive* (linear AR) model with dependencies on k past values (i.e., not just one as in the case of random walk). An autoregressive process of order k, AR(k), for a discrete data set is defined by

$$y_i = \sum_{j=1}^{k} a_j y_{i-j} + e_i. \tag{10.96}$$

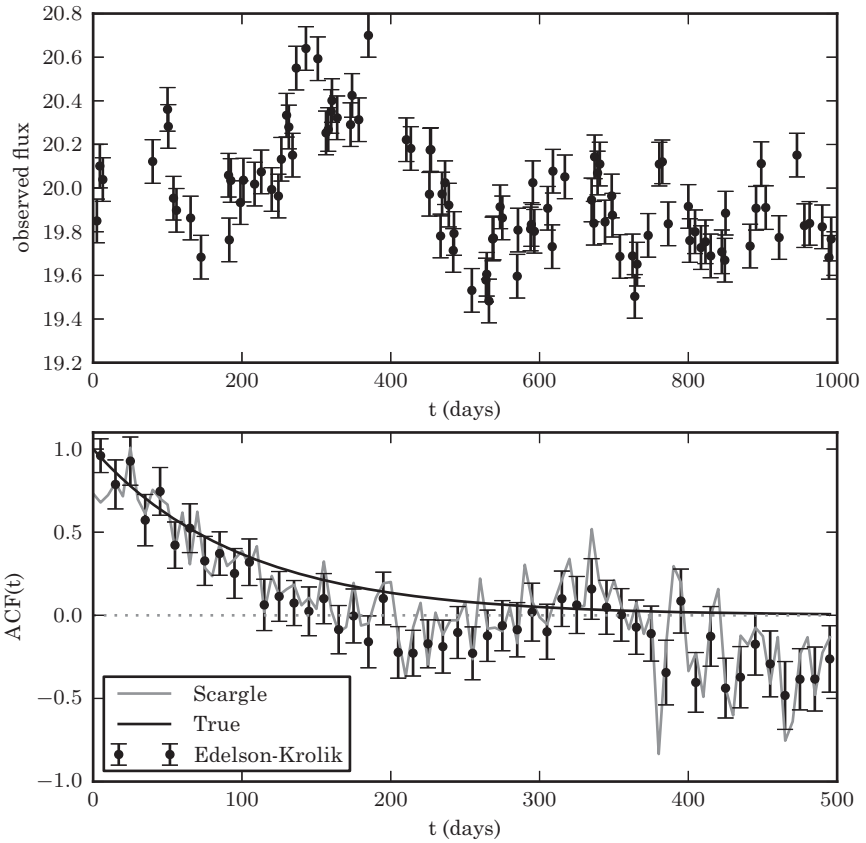

Figure 10.30. Example of the autocorrelation function for a stochastic process. The top panel shows a simulated light curve generated using a damped random walk model (§10.5.4). The bottom panel shows the corresponding autocorrelation function computed using Edelson and Krolik's DCF method and the Scargle method. The solid line shows the input autocorrelation function used to generate the light curve.

That is, the latest value of y is expressed as a linear combination of the k previous values of y, with the addition of noise (for random walk, $k = 1$ and $a_1 = 1$). If the data are drawn from a stationary process, coefficients a_j satisfy certain conditions. The ACF for an AR(k) process is nonzero for all lags, but it decays quickly.

The literature on autoregressive models is abundant because applications vary from signal processing and general engineering to stock-market modeling. Related modeling frameworks include the moving average (MA, where y_i depends only on past values of noise), autoregressive moving average (ARMA, a combination of AR and MA processes), autoregressive integrated moving average (ARIMA, a combination of ARMA and random walk), and state-space or dynamic linear modeling (so-called Kalman filtering). More details and references about these stochastic autoregressive models can be found in FB2012. Alternatively, modeling can be done in the frequency domain (per the Wiener–Khinchin theorem).

For example, a simple but astronomically very relevant problem is distinguishing a random walk from pure noise. That is, given a time series, the question is whether it

better supports the hypothesis that $a_1 = 0$ (noise) or that $a_1 = 1$ (random walk). For comparison, in stock market analysis this pertains to predicting the next data value based on the current data value and the historic mean. If a time series is a random walk, values higher and lower than the current value have equal probabilities. However, if a time series is pure noise, there is a useful asymmetry in probabilities due to regression toward the mean (see §4.7.1). A standard method for answering this question is to compute the Dickey–Fuller statistic; see [20].

An autoregressive process defined by eq. 10.96 applies only to evenly sampled time series. A generalization is called the *continuous autoregressive process*, CAR(k); see [31]. The CAR(1) process has recently received a lot of attention in the context of quasar variability and is discussed in more detail in the next section.

In addition to autoregressive models, data can be modeled using the covariance matrix (e.g., using Gaussian process; see §8.10). For example, for the CAR(1) process,

$$S_{ij} = \sigma^2 \exp(-|t_{ij}|/\tau), \tag{10.97}$$

where σ and τ are model parameters; σ^2 controls the short timescale covariance ($t_{ij} \ll \tau$), which decays exponentially on a timescale given by τ. A number of other convenient models and parametrizations for the covariance matrix are discussed in the context of quasar variability in [64].

10.5.4. Damped Random Walk Model

The CAR(1) process is described by a stochastic differential equation which includes a damping term that pushes $y(t)$ back to its mean (see [31]); hence, it is also known as *damped random walk* (another often-used name is the Ornstein–Uhlenbeck process, especially in the context of Brownian motion, [26]). In analogy with calling random walk "drunkard's walk," damped random walk could be called "married drunkard's walk" (who always comes home instead of drifting away).

Following eq. 10.97, the autocorrelation function for a damped random walk is

$$\mathrm{ACF}(t) = \exp(-t/\tau), \tag{10.98}$$

where τ is the characteristic timescale (relaxation time, or damping timescale). Given the ACF, it is easy to show that the structure function is

$$\mathrm{SF}(t) = \mathrm{SF}_\infty \left[1 - \exp(-t/\tau)\right]^{1/2}, \tag{10.99}$$

where SF_∞ is the asymptotic value of the structure function (equal to $\sqrt{2}\sigma$, where σ is defined in eq. 10.97, when the structure function applies to differences of the analyzed process; for details see [31, 37]) and

$$\mathrm{PSD}(f) = \frac{\tau^2 \, \mathrm{SF}_\infty^2}{1 + (2\pi f \tau)^2}. \tag{10.100}$$

Therefore, the damped random walk is a $1/f^2$ process at high frequencies, just as ordinary random walk. The damped nature is seen as the flat PSD at low frequencies ($f \ll 2\pi/\tau$). An example of a light curve generated using a damped random walk is shown in figure 10.30.

For evenly sampled data, the CAR(1) process is equivalent to the AR(1) process with $a_1 = \exp(-1/\tau)$; that is, the next value of y is the damping factor times the previous value plus noise. The noise for the AR(1) process, $\sigma_{\rm AR}$, is related to SF_∞ via

$$\sigma_{\rm AR} = \frac{\mathrm{SF}_\infty}{\sqrt{2}} \left[1 - \exp(-2/\tau)\right]^{1/2}. \tag{10.101}$$

A damped random walk provides a good description of the optical continuum variability of quasars; see [31, 33, 37]. Indeed, this model is so successful that it has been used to distinguish quasars from stars (both are point sources in optical images, and can have similar colors) based solely on variability behavior; see [9, 36]. Nevertheless, at short timescales of the order a month or less (at high frequencies from 10^{-6} Hz up to 10^{-5} Hz), the PSD is closer to $1/f^3$ behavior than to $1/f^2$ predicted by the damped random walk model; see [39, 64].

Scikit-learn contains a utility which generates damped random walk light curves given a random seed:

```
>>> import numpy as np
>>> from astroML.time_series import generate_damped_RW

>>> t = np.arange(0, 1000)
>>> y = generate_damped_RW(t, tau=300, random_state=0)
```

For a more detailed example, see the source code associated with figure 10.30.

10.6. Which Method Should I Use for Time Series Analysis?

Despite extensive literature developed in the fields of signal processing, statistics, and econometrics, there are no universal methods that always work. This is even more so in astronomy where uneven sampling, low signal-to-noise ratio, and heteroscedastic errors often prevent the use of standard methods drawn from other fields.

The main tools for time series analysis belong to either the time domain or the frequency domain. When searching for periodic variability, tools from the frequency domain are usually better because the signal becomes more "concentrated." This is a general feature of model fitting, where a matched filter approach can greatly improve the ability to detect a signal. For typical astronomical periodic time series, the generalized Lomb–Scargle method is a powerful method; when implemented to model several terms in a truncated Fourier series, instead of a single sinusoid, it works well when analyzing variable stars. It is well suited to unevenly sampled data with low signal-to-noise ratio and heteroscedastic errors. Nevertheless, when the shape of the underlying light curve cannot be approximated with a small number of Fourier terms, nonparametric methods such as the minimum string length method, the phase dispersion minimization method, or the Bayesian blocks algorithm may perform better. Analysis of arrival time data represents different challenges; the Gregory and Loredo algorithm is a good general method in this case.

We have only briefly reviewed methods for the analysis of stochastic time series. Tools such as autocorrelation and structure functions are becoming increasingly used in the context of nonperiodic and stochastic variability. We have not discussed several topics of growing importance, such as state-space models, Kalman filters, Markov chains, and stochastic differential equations (for astronomical discussion of the latter, see [59] and [60]). For a superb book on stochastic dynamical systems written in a style accessible to astronomers, see [30]. Excellent monographs by statisticians that discuss forecasting, ARMA and ARIMA models, state-space models, Kalman filtering, and related time series topics are references [8] and [51].

References

[1] Abbott, B. P., R. Abbott, R. Adhikari, and others (2009). LIGO: the Laser Interferometer Gravitational-Wave Observatory. *Reports on Progress in Physics 72*(7), 076901.

[2] Blomme, J., L. M. Sarro, F. T. O'Donovan, and others (2011). Improved methodology for the automated classification of periodic variable stars. *MNRAS 418*, 96–106.

[3] Bretthorst, G. (1988). *Bayesian Spectrum Analysis and Parameter Estimation*. Lecture Notes in Statistics. Springer.

[4] Bretthorst, G. L. (2001a). Generalizing the Lomb-Scargle periodogram. In A. Mohammad-Djafari (Ed.), *Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, Volume 568 of *American Institute of Physics Conference Series*, pp. 241–245.

[5] Bretthorst, G. L. (2001b). Generalizing the Lomb-Scargle periodogram–the nonsinusoidal case. In A. Mohammad-Djafari (Ed.), *Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, Volume 568 of *American Institute of Physics Conference Series*, pp. 246–251.

[6] Bretthorst, G. L. (2001c). Nonuniform sampling: Bandwidth and aliasing. Volume 567 of *American Institute of Physics Conference Series*, pp. 1–28.

[7] Bretthorst, G. L. (2003). Frequency estimation, multiple stationary nonsinusoidal resonances with trend. In C. J. Williams (Ed.), *Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, Volume 659 of *American Institute of Physics Conference Series*, pp. 3–22.

[8] Brockwell, P. and R. Davis (2006). *Time Series: Theory and Methods*. Springer.

[9] Butler, N. R. and J. S. Bloom (2011). Optimal time-series selection of quasars. *AJ 141*, 93.

[10] Clauset, A., C. R. Shalizi, and M. Newman (2009). Power-law distributions in empirical data. *SIAM Review 51*, 661–703.

[11] Covey, K. R., Ž. Ivezić, D. Schlegel, and others (2007). Stellar SEDs from 0.3 to 2.5 μm: Tracing the stellar locus and searching for color outliers in the SDSS and 2MASS. *AJ 134*, 2398–2417.

[12] Cumming, A., G. W. Marcy, and R. P. Butler (1999). The Lick Planet Search: Detectability and mass thresholds. *ApJ 526*, 890–915.

[13] Daubechies, I. (1992). *Ten Lectures on Wavelets*. SIAM: Society for Industrial and Applied Mathematics.

[14] de Jager, O. C., B. C. Raubenheimer, and J. Swanepoel (1989). A powerful test for weak periodic signals with unknown light curve shape in sparse data. *A&A 221*, 180–190.

[15] de Vries, W. H., R. H. Becker, and R. L. White (2003). Long-term variability of Sloan Digital Sky Survey quasars. *AJ 126*, 1217–1226.

[16] de Vries, W. H., R. H. Becker, R. L. White, and C. Loomis (2005). Structure function analysis of long-term quasar variability. *AJ 129*, 615–629.

[17] Deb, S. and H. P. Singh (2009). Light curve analysis of variable stars using Fourier decomposition and principal component analysis. *A&A 507*, 1729–1737.

[18] Debosscher, J., L. M. Sarro, C. Aerts, and others (2007). Automated supervised classification of variable stars. I. Methodology. *A&A 475*, 1159–1183.

[19] Deeming, T. J. (1975). Fourier analysis with unequally-spaced data. *Ap&SS 36*, 137–158.

[20] Dickey, D. A. and W. A. Fuller (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association 74*, 427–431.

[21] Dworetsky, M. M. (1983). A period-finding method for sparse randomly spaced observations of "How long is a piece of string?". *MNRAS 203*, 917–924.

[22] Edelson, R. A. and J. H. Krolik (1988). The discrete correlation function – A new method for analyzing unevenly sampled variability data. *ApJ 333*, 646–659.

[23] Eyer, L. and P. Bartholdi (1999). Variable stars: Which Nyquist frequency? *A&AS 135*, 1–3.

[24] Eyer, L. and N. Mowlavi (2008). Variable stars across the observational HR diagram. *Journal of Physics Conference Series 118*(1), 012010.

[25] Frescura, F., C. A. Engelbrecht, and B. S. Frank (2008). Significance of periodogram peaks and a pulsation mode analysis of the Beta Cephei star V403Car. *MNRAS 388*, 1693–1707.

[26] Gillespie, D. T. (1996). The mathematics of Brownian motion and Johnson noise. *American Journal of Physics 64*, 225–240.

[27] Gottlieb, E. W., E. L. Wright, and W. Liller (1975). Optical studies of UHURU sources. XI. A probable period for Scorpius X-1 = V818 Scorpii. *ApJL 195*, L33–L35.

[28] Gregory, P. C. and T. J. Loredo (1992). A new method for the detection of a periodic signal of unknown shape and period. *ApJ 398*, 146–168.

[29] Hoffman, D. I., T. E. Harrison, and B. J. McNamara (2009). Automated variable star classification using the Northern Sky Variability Survey. *AJ 138*, 466–477.

[30] Honerkamp, J. (1994). *Stochastic Dynamical Systems*. John Wiley.

[31] Kelly, B. C., J. Bechtold, and A. Siemiginowska (2009). Are the variations in quasar optical flux driven by thermal fluctuations? *ApJ 698*, 895–910.

[32] Koch, D. G., W. J. Borucki, G. Basri, and others (2010). Kepler mission design, realized photometric performance, and early science. *ApJL 713*, L79–L86.

[33] Kozłowski, S., C. S. Kochanek, A. Udalski, and others (2010). Quantifying quasar variability as part of a general approach to classifying continuously varying sources. *ApJ 708*, 927–945.

[34] Lachowicz, P. and C. Done (2010). Quasi-periodic oscillations under wavelet microscope: the application of matching pursuit algorithm. *A&A 515*, A65.

[35] Lomb, N. R. (1976). Least-squares frequency analysis of unequally spaced data. *Ap&SS 39*, 447–462.

[36] MacLeod, C. L., K. Brooks, Ž. Ivezić, and others (2011). Quasar selection based on photometric variability. *ApJ 728*, 26.

[37] MacLeod, C. L., Ž. Ivezić, C. S. Kochanek, and others (2010). Modeling the time variability of SDSS Stripe 82 quasars as a damped random walk. *ApJ 721*, 1014–1033.

[38] Mallat, S. G. and Z. Zhang (1992). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing 41*(12), 3397–3415.

[39] Mushotzky, R. F., R. Edelson, W. Baumgartner, and P. Gandhi (2011). Kepler observations of rapid optical variability in Active Galactic Nuclei. *ApJL 743*, L12.

[40] Palmer, D. M. (2009). A fast chi-squared technique for period search of irregularly sampled data. *ApJ 695*, 496–502.

[41] Pojmanski, G. (2002). The All Sky Automated Survey. Catalog of variable stars. I. 0h - 6h Quarter of the southern hemisphere. *Acta Astronomica 52*, 397–427.

[42] Press, W. H. (1978). Flicker noises in astronomy and elsewhere. *Comments on Astrophysics 7*, 103–119.

[43] Richards, J. W., D. L. Starr, N. R. Butler, and others (2011). On machine-learned classification of variable stars with sparse and noisy time-series data. *ApJ 733*, 10.

[44] Sarro, L. M., J. Debosscher, M. López, and C. Aerts (2009). Automated supervised classification of variable stars. II. Application to the OGLE database. *A&A 494*, 739–768.

[45] Scargle, J. D. (1982). Studies in astronomical time series analysis. II – Statistical aspects of spectral analysis of unevenly spaced data. *ApJ 263*, 835–853.

[46] Scargle, J. D. (1989). Studies in astronomical time series analysis. III – Fourier transforms, autocorrelation functions, and cross-correlation functions of unevenly spaced data. *ApJ 343*, 874–887.

[47] Scargle, J. D. (1998). Studies in astronomical time series analysis. V. Bayesian blocks, a new method to analyze structure in photon counting data. *ApJ 504*, 405.

[48] Scargle, J. D., J. P. Norris, B. Jackson, and J. Chiang (2012). Studies in astronomical time series analysis. VI. Bayesian block representations. *ArXiv:astro-ph/1207.5578*.

[49] Schwarzenberg-Czerny, A. (1998). The distribution of empirical periodograms: Lomb-Scargle and PDM spectra. *MNRAS 301*, 831–840.

[50] Sesar, B., J. S. Stuart, Ž. Ivezić, and others (2011). Exploring the variable sky with LINEAR. I. Photometric recalibration with the Sloan Digital Sky Survey. *AJ 142*, 190.

[51] Shumway, R. and D. Stoffer (2000). *Time Series Analysis and Its Applications*. Springer.

[52] Simonetti, J. H., J. M. Cordes, and D. S. Heeschen (1985). Flicker of extragalactic radio sources at two frequencies. *ApJ 296*, 46–59.

[53] Stellingwerf, R. F. (1978). Period determination using phase dispersion minimization. *ApJ 224*, 953–960.

[54] Süveges, M. (2012). False alarm probability based on bootstrap and extreme-value methods for periodogram peaks. In J.-L. Starck and C. Surace (Eds.), *ADA7 – Seventh Conference on Astronomical Data Analysis*.

[55] Süveges, M., B. Sesar, M. Váradi, and others (2012). Search for high-amplitude Delta Scuti and RR Lyrae stars in Sloan Digital Sky Survey Stripe 82 using principal component analysis. *ArXiv:astro-ph/1203.6196*.

[56] Timmer, J. and M. Koenig (1995). On generating power law noise. *A&A 300*, 707.

[57] Torrence, C. and G. P. Compo (1998). A practical guide to wavelet analysis. *Bull. Amer. Meteor. Soc. 79*(1), 61–78.

[58] Vanden Berk, D. E., B. C. Wilhite, R. G. Kron, and others (2004). The ensemble photometric variability of ∼25,000 quasars in the Sloan Digital Sky Survey. *ApJ 601*, 692–714.

[59] Vio, R., S. Cristiani, O. Lessi, and A. Provenzale (1992). Time series analysis in astronomy - An application to quasar variability studies. *ApJ 391*, 518–530.

[60] Vio, R., N. R. Kristensen, H. Madsen, and W. Wamsteker (2005). Time series analysis in astronomy: Limits and potentialities. *A&A 435*, 773–780.

[61] Wang, Y., R. Khardon, and P. Protopapas (2012). Nonparametric Bayesian estimation of periodic light curves. *ApJ 756*, 67.

[62] Welch, P. (1967). The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics 15*(2), 70–73.

[63] Zechmeister, M. and M. Kürster (2009). The generalised Lomb-Scargle periodogram. A new formalism for the floating-mean and Keplerian periodograms. *A&A 496*, 577–584.

[64] Zu, Y., C. S. Kochanek, S. Kozłowski, and A. Udalski (2012). Is quasar variability a damped random walk? *ArXiv:astro-ph/1202.3783*.