Chapter 2
Statistical Aspects

2.1 Introduction

Practical signal processing frequently involves statistical aspects. If you are using a sensor to measure say temperature, or light, or pressure, or anything else, you usually get a signal with noise in it, and then there is a problem of noise removal. The almost immediate idea could be to apply averaging; however our advice is first try to know better about the noise you have. There are many other contexts where the data you get suffer from interference, lack of precision, variations along time, etc. For example, suppose you want to measure the period of a pendulum using a watch: the scientific procedure is to repeat the measurements (the values obtained will be different for each measurement), get a data set, and then statistically process this set.

Let us imagine an example that encloses the main sources of randomness in the signals to be processed. Suppose that with a radar on the sea coast you want to determine the position of a floating body you just detected. There will be three main problems:

- The radar signals are contaminated with electromagnetic noise.
- There are resolution limitations in the measurements.
- The floating body is moving because of the waves.

The best you can do in this example is to get a good estimate, in statistical terms.

In this chapter some aspects of probability and statistics, particularly relevant for signal processing, are selected. First several kinds of probability density distributions are considered, and then parameters to characterize random signals are introduced, [101]. The last sections are devoted to matters that, nowadays, are subject of increasing attention, like for instance Bayes' rule and Markov processes.

In view of these topics it is opportune to consider two random events A and B. They occur with probabilities $P(A)$ and $P(B)$ respectively. The two random events are *independent* if the probability of having A and B is $P(A, B) = P(A)P(B)$. A typical example is playing with two dice. Another important concept is *conditional*

J.M. Giron-Sierra, *Digital Signal Processing with Matlab Examples, Volume 1*,
Signals and Communication Technology, DOI 10.1007/978-981-10-2534-1_2

probability. The expression *P(A|C)* reads as the probability of *A* given *C*. For instance, the probability of raining in July.

Some of the functions used in this chapter belong to the MATLAB Statistics Toolbox. This will be indicated with (*ST); for example *weibpdf()* (*ST) says the function *weibpdf()* belongs to the MATLAB Statistics Toolbox.

2.2 Random Signals and Probability Density Distributions

The chief objective of this section is to introduce probability density distributions and functions, selecting three illustrative cases: the uniform, the normal and the log-normal distributions. The normal distribution is, in particular, a very important case.

2.2.1 Basic Concepts

Suppose there is a continuous random variable $y(t)$, the distribution function $F_y(v)$ of this variable is the following:

$$F_y(v) = P\left(y(t) \leq v\right), \; -\infty < v < \infty \tag{2.1}$$

where $P()$ is the probability of.

The probability density function of $y(t)$ is:

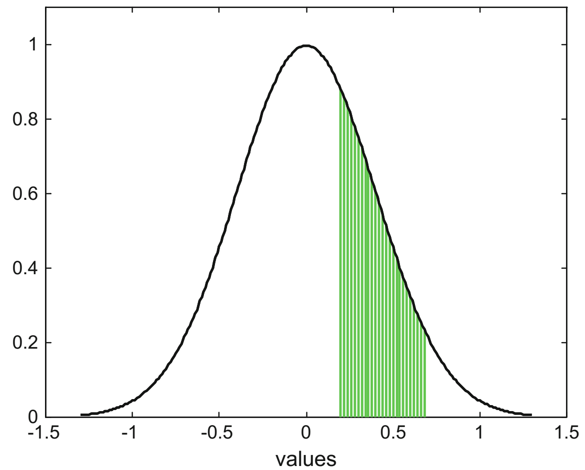


Fig. 2.1 A probability density function

$$f_y(v) = \frac{d\,F_y(v)}{d\,v} \tag{2.2}$$

A well-known example of probability distribution function, the so-called normal distribution, has a bell shaped probability density function as shown in Fig. 2.1. In this figure a shaded zone has been painted corresponding to an interval [a, b] of the values that $y(t)$ can have. The probability of $y(t)$ value to fall into this interval is given by the area of the shaded zone.

The abbreviation "PDF" will be used in this book to denote "Probability Density Function".

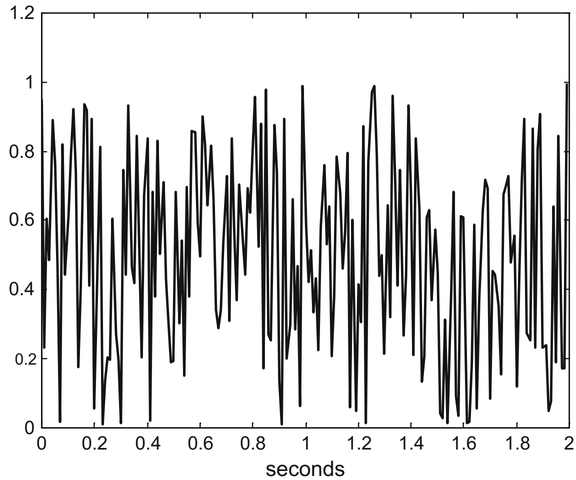
2.2.2 Random Signal with Uniform PD

A random signal taking equiprobable values in successive instants has a uniform PDF. For example, the sequence of values that would be obtained recording the final angles (0°.. 360°) where a roulette wheel stops along several runs in gambling days.

Figure 2.2 shows a random signal with uniform PDF. It has been obtained using the *rand()* function provided by MATLAB. Notice that the values are from 0 to 1. This signal can be easily modified by adding a constant and/or multiplying by a constant: the result will also have a uniform PDF.

The following program has been used to generate Fig. 2.2.

Fig. 2.2 A random signal with uniform PDF



Program 2.1 Random signal with uniform PDF

```
% Random signal with uniform PDF
fs=100; %sampling frequency in Hz
tiv=1/fs; %time interval between samples;
t=0:tiv:(2-tiv); %time intervals set (200 values)
N=length(t); %number of data points
y=rand(N,1); %random signal data set
plot(t,y,'-k'); %plots figure
axis([0 2 0 1.2]);
xlabel('seconds');
title('random signal with uniform PDF');
```

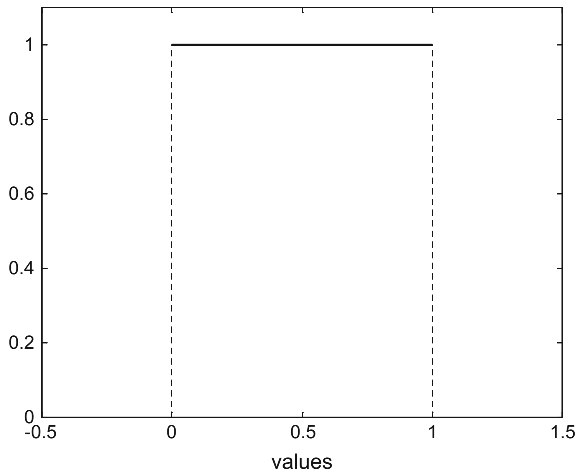
The uniform PDF graphical representation is just a horizontal line between two limits, as shown in Fig. 2.3. This figure has been generated by Program 2.2, which uses *unifpdf()* (*ST) for a uniform PDF between values 0 and 1 (other values can be specified in the *unifpdf()* function parenthesis).

Program 2.2 Uniform PDF

```
% Uniform PDF
v=0:0.01:1; %values set
ypdf=unifpdf(v,0,1); %uniform PDF
plot(v,ypdf,'k'); hold on; %plots figure
axis([-0.5 1.5 0 1.1]);
xlabel('values'); title('uniform PDF');
plot([0 0],[0 1],'--k');
plot([1 1],[0 1],'--k');
```

An interesting way to check the quality of the MATLAB random variable generation functions is by plotting a histogram of the signal values along time. For this

Fig. 2.3 Uniform PDF



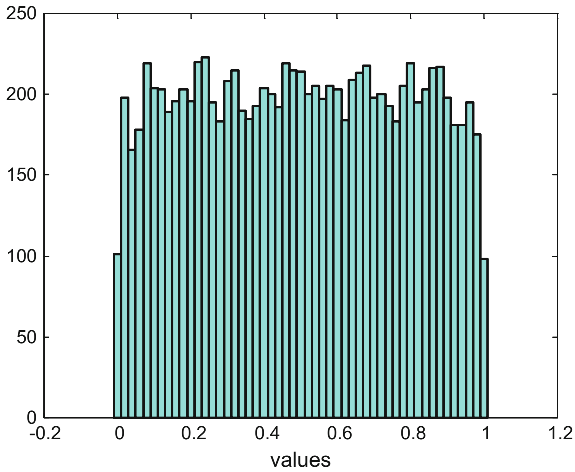


Fig. 2.4 Histogram of a random signal with uniform PDF

purpose relatively large signal data sets should be generated. MATLAB provides the *hist()* function to obtain the histogram.

Figure 2.4 shows the result for the *rand()* function. As it can be seen in Program 2.3, which has been used to generate the figure, a signal data set of 10,000 values has been generated and then classified into data bins 0.02 wide, from 0 to 1 signal values. The colour of the bars has been chosen to be cyan for a better view in press. In general, Fig. 2.4 shows a passable approximation to a uniform PDF.

Program 2.3 Histogram of a random signal with uniform PDF

```
% Histogram of a random signal with uniform PDF
fs=100; %sampling frequency in Hz
tiv=1/fs; %time interval between samples;
t=0:tiv:(100-tiv); %time intervals set (10000 values)
N=length(t); %number of data points
y=rand(N,1); %random signal data set
v=0:0.02:1; %value intervals set
hist(y,v); colormap(cool); %plots histogram
xlabel('values');
title('Histogram of random signal with uniform PDF');
```

2.2.3 Random Signal with Normal (Gaussian) PDF

The normal PDF has the following mathematical expression:

$$f_y(\upsilon) = \frac{e^{-(\upsilon-\mu)^2/2\sigma^2}}{\sigma\sqrt{2\pi}}, \quad \sigma > 0, \; -\infty < \mu < \infty, \; -\infty < \upsilon < \infty \tag{2.3}$$

where μ is the mean and σ is the standard deviation of the random variable.

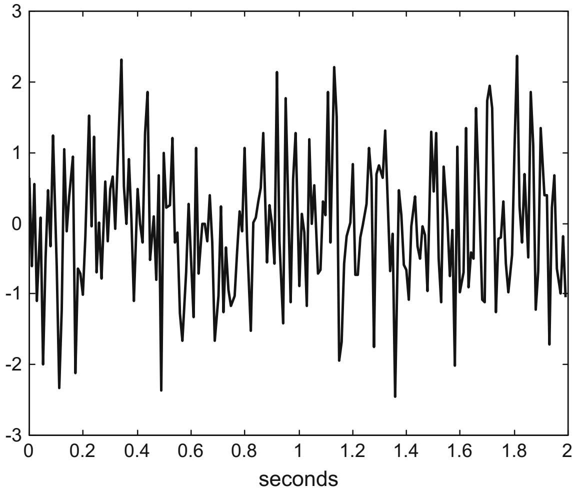


Fig. 2.5 A random signal with normal PDF

As said before, the normal distribution is very important, both from theoretical and practical points of view (see [78] for historical details). Most noise and perturbation models employed in systems or automatic control theory are of Gaussian nature. The practical reason is provided by the central limit theorem, which in words says that if a phenomenon is the accumulation of many small additive random effects, it tends to a normal distribution. For instance, the number of travels per day of an elevator.

Figure 2.5 shows a random signal with normal PDF. It has been obtained using the *randn()* function provided by MATLAB (notice the slight name difference compared to *rand()*, which corresponds to uniform PDF).

The values of the signal in Fig. 2.5 have positive and negative values. The figure has been obtained with Program 2.4, using the *randn()* function, which generates a signal with mean zero and variance one.

Program 2.4 Random signal with normal PDF

```
% Random signal with normal PDF
fs=100; %sampling frequency in Hz
tiv=1/fs; %time interval between samples;
t=0:tiv:(2-tiv); %time intervals set (200 values)
N=length(t); %number of data points
y=randn(N,1); %random signal data set
plot(t,y,'-k'); %plots figure
axis([0 2 -3 3]);
xlabel('seconds');
title('random signal with normal PDF');
```

The normal PDF has the shape of a bell. The larger the standard deviation, the wider is the bell. Figure 2.6 shows a PDF example, obtained with Program 2.5, for a mean zero and a standard deviation one. The program uses the *normpdf()* (*ST) function to obtain the PDF.

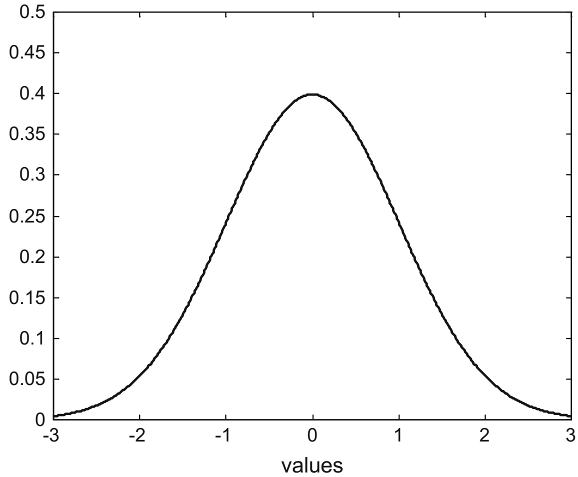


Fig. 2.6 Normal PDF

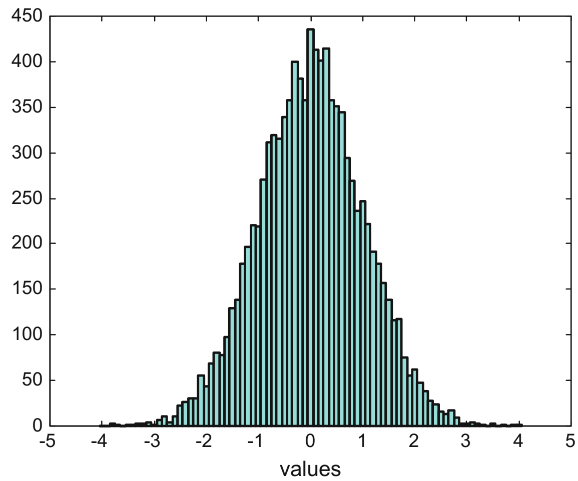


Fig. 2.7 Histogram of a random signal with normal PDF

Program 2.5 Normal PDF

```
% Normal PDF
v=-3:0.01:3; %values set
mu=0; sigma=1; %random variable parameters
ypdf=normpdf(v,mu,sigma); %normal PDF
plot(v,ypdf,'k'); hold on; %plots figure
axis([-3 3 0 0.5]);
xlabel('values'); title('normal PDF');
```

Like in the previous case—the uniform PDF—a histogram of the signal generated by the *randn()* function has been obtained, with 10,000 signal data values, data bins 0.1 wide. Figure 2.7 shows the result: a fairly good approximation to the normal PDF.

Program 2.6 Histogram of a random signal with normal PDF

```
% Histogram of a random signal with normal PDF
fs=100; %sampling frequency in Hz
tiv=1/fs; %time interval between samples;
t=0:tiv:(100-tiv); %time intervals set (10000 values)
N=length(t); %number of data points
y=randn(N,1); %random signal data set
v=-4:0.1:4; %value intervals set
hist(y,v); colormap(cool); %plots histogram
xlabel('values');
title('Histogram of random signal with normal PDF');
```

2.2.4 Random Signal with Log-Normal PDF

A random variable y is log-normally distributed if *log(y)* has a normal distribution. The log-normal PDF has the following mathematical expression:

$$f_y(v) = \frac{e^{-(\log(v)-\mu)^2/2\sigma^2}}{v\,\sigma\,\sqrt{2\pi}}, \quad \sigma > 0, \ -\infty < \mu < \infty, \ -\infty < v < \infty \tag{2.4}$$

where μ is the mean and σ is the standard deviation of the random variable.

The log-normal distribution is related the multiplicative product of many small independent factors. It is observed for instance in environment, microbiology, human medicine, social sciences, or economics contexts, [49]. For example, the case of latent periods (time from infection to first symptoms) of infectious diseases.

Figure 2.8 shows a random signal with log-normal PDF. It has been obtained, using the *lognrnd()* (*ST) function, with the Program 2.7. A mean zero and a standard deviation one has been specified inside the parenthesis of *lognrnd()*; other mean and standard deviation values can be explored. Notice that signal values are always positive.

Program 2.7 Random signal with log-normal PDF

```
% Random signal with log-normal PDF
fs=100; %sampling frequency in Hz
tiv=1/fs; %time interval between samples;
t=0:tiv:(2-tiv); %time intervals set (200 values)
N=length(t); %number of data points
mu=0; sigma=1; %random signal parameters
y=lognrnd(mu,sigma,N,1); %random signal data set
plot(t,y,'-k'); %plots figure
axis([0 2 0 12]);
xlabel('seconds'); title('random signal with log-normal PDF');
```

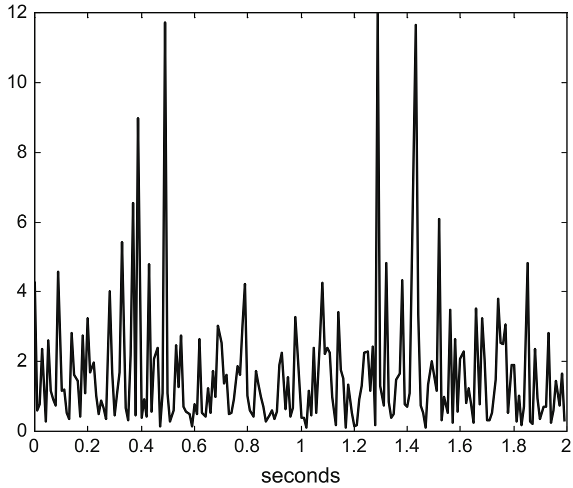


Fig. 2.8 A random signal with log-normal PDF

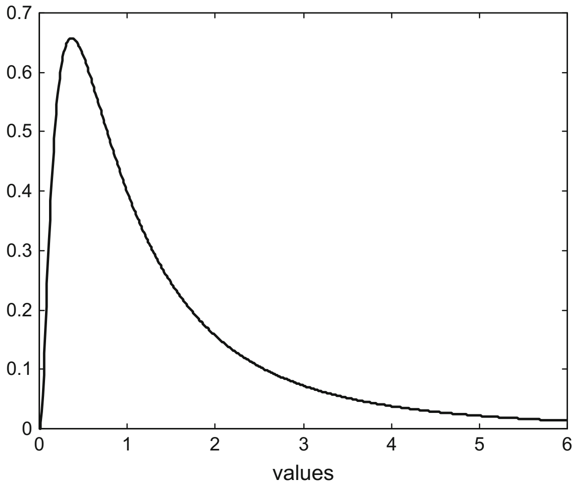


Fig. 2.9 Log-normal PDF

Figure 2.9 shows a log-normal PDF, as obtained by Program 2.8 using *lognpdf()* (*ST). Notice that the PDF is skewed, corresponding to the fact that the signal exhibit large peaks, as can be seen in Fig. 2.8.

Program 2.8 Log-normal PDF

```
% Log-normal PDF
v=-3:0.01:6; %values set
mu=0; sigma=1; %random variable parameters
ypdf=lognpdf(v,mu,sigma); %log-normal PDF
plot(v,ypdf,'k'); hold on; %plots figure
axis([0 6 0 0.7]);
xlabel('values'); title('log-normal PDF');
```

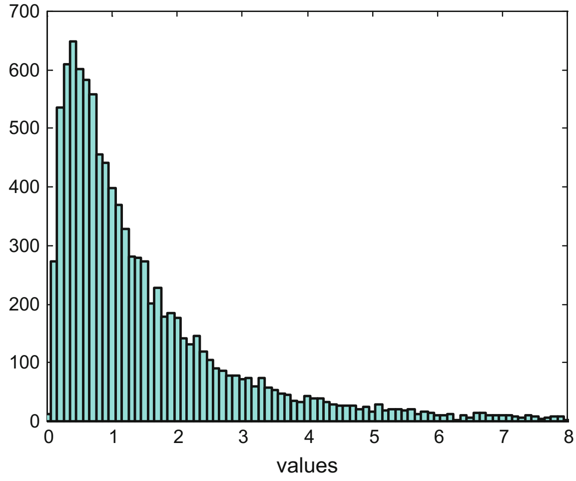


Fig. 2.10 Histogram of a random signal with log-normal PDF

Again a histogram of the signal has been obtained, for the case of log-normal PDF, with 10,000 signal data values and data bins 0.1 wide. Figure 2.10 shows the result, as obtained by Program 2.9.

Program 2.9 Histogram of a random signal with log-normal PDF

```
% Histogram of a random signal with log-normal PDF
fs=100; %sampling frequency in Hz
tiv=1/fs; %time interval between samples;
t=0:tiv:(100-tiv); %time intervals set (10000 values)
N=length(t); %number of data points
mu=0; sigma=1; %random signal parameters
y=lognrnd(mu,sigma,N,1); %random signal data set
v=0:0.1:12; %value intervals set
hist(y,v); colormap(cool); %plots histogram
axis([0 8 0 700]);
xlabel('values');
title('Histogram of random signal with log-normal PDF');
```

2.3 Expectations and Moments

Let us take from descriptive statistics some important concepts concerning the characterization of random signals. Some comments and examples are added for a better insight.

2.3.1 Expected Values, and Moments

Consider the random variable y, with $f_y(v)$ as PDF. The expected value of y is:

$$E(y) = \int_{-\infty}^{\infty} v\, f_y(v)\ dv \tag{2.5}$$

$E(y)$ is said to exist if the integral converges absolutely.

Let $g(y)$ be a function of y, then the expected value of $g(y)$ is:

$$E(g(y)) = \int_{-\infty}^{\infty} g(v) f_y(v)\ dv \tag{2.6}$$

The moments about the origin for the variable y are given by:

$$\mu'_k \;=\; E(y^k),\;\; k = 1,\, 2,\, 3\ldots \tag{2.7}$$

For $k = 1$: $\mu'_1 \;=\; \mu$ (μ denotes the mean of y)

The moments about the mean, or central moments, for the variable y are given by:

$$\mu_k \;=\; E((y - \mu)^k),\;\; k = 1,\, 2,\, 3\ldots \tag{2.8}$$

2.3.2 Mean, Variance, Etc.

Figure 2.11 shows a skewed PDF where the mean, the median and the mode values are marked (Program 2.10). In symmetrical PDFs these three values would be coincident.

The **mean** μ of the variable y is the expected value of y (Eq. 2.5). It is also called the average value of y. Using a mass analogy, it may be regarded as the center of mass of the distribution.

A **median** y_0 of the variable y is any point that divides the mass of the distribution into two equal parts, that is:

$$P\,(y \,\le\, y_0) \;=\; \frac{1}{2} \tag{2.9}$$

A point v_i such that:

$$f_y(v_i) \;>\; f_y(v_i \,+\, \varepsilon)\ \mathit{and}\ f_y(v_i) \;>\; f_y(v_i \,-\, \varepsilon) \tag{2.10}$$

(where ε is an arbitrarily small positive quantity) is called a **mode** of y.

A mode is a value of y corresponding to a peak of the PDF. When the PDF has only one peak, the distribution is said to be *unimodal*.

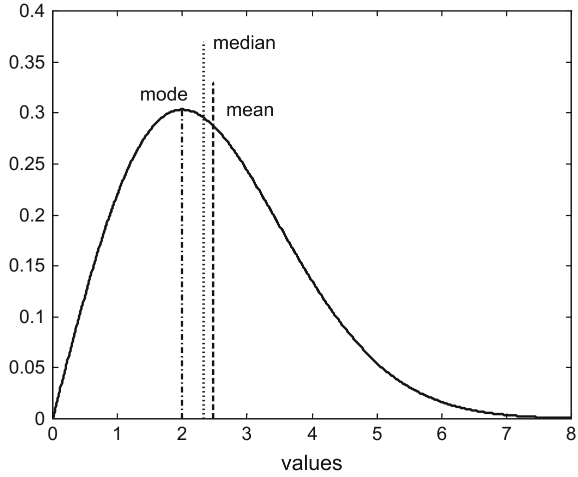


Fig. 2.11 Mean, median and mode marked on a PDF

In measurement tasks, depending on the PDF of the signal being obtained it would be advisable to consider the mean, or the median, or the mode or modes, as the value of interest. In particular, while the mean of a variable y may not exist, the median will exist.

Program 2.10 A skewed PDF with mean, median and mode

```
% A skewed PDF with mean, median and mode
v=0:0.01:8; %values set
alpha=2;; %random variable parameter
ypdf=raylpdf(v,alpha); %Rayleigh PDF
plot(v,ypdf,'k'); hold on; %plots figure
axis([0 8 0 0.4]);
xlabel('values'); title('a skewed PDF');
fs=100; %sampling frequency in Hz
tiv=1/fs; %time interval between samples;
t=0:tiv:(20-tiv); %time intervals set (2000 values)
N=length(t); %number of data points
y=raylrnd(alpha,N,1); %random signal data set
mu=mean(y); %mean of y
vo=median(y); %median of y
[pky,pki]=max(ypdf); %peak of the PDF
plot([mu mu],[0 0.33],'--k'); %mean
plot([vo vo],[0 0.37],':k'); %median
plot([v(pki) v(pki)],[0 pky],'-.k'); %mode
```

The **autocorrelation** of y(t) is defined as:

$$R(t_1\, ,\, t_2) \;=\; E(y(t_1)\, y(t_2)) \tag{2.11}$$

The autocorrelation is related with the extent of predictability about the future behaviour of $y(t)$ taking into account its past.

The value of $R(t_1, t_2)$ for $t_1 = t_2$ is the **average power** of the signal $y(t)$. It is also the second moment of $y(t)$ about the origin.

The **autocovariance** of $y(t)$ is defined as:

$$C(t_1\,,\,t_2) \;=\; E((y(t_1) - \mu)\,(y(t_2) - \mu)) \tag{2.12}$$

The value of $C(t_1, t_2)$ for $t_1 = t_2$ is the **variance** of the signal $y(t)$. It is also the second moment of $y(t)$ about the mean. The positive square root of the variance is the **standard deviation** σ.

The variance is related with how large is the range of values of $y(t)$.

The random signal $y(t)$ is called *strict-sense stationary* if all its statistical properties are invariant to a shift of the time origin.

MATLAB offers the following functions: *mean(), median(), var(), std()* (for standard deviation).

2.3.3 Transforms

There are several transforms that help to beautifully deduce important results. For instance the following generating functions:

- *Generating function*:

$$g(\upsilon) \;=\; E(\upsilon^y) \tag{2.13}$$

- *Moment generating function*:

$$\Gamma(\upsilon) \;=\; E(e^{\upsilon \cdot y}) \tag{2.14}$$

As in many other contexts, it is convenient to consider the following transforms:

- *Laplace transform*:

$$E(e^{-\,s \cdot y}) \tag{2.15}$$

- *Fourier transform*:

$$E(e^{-j\,\upsilon \cdot y}) \tag{2.16}$$

Finally:

- *Characteristic function*:

$$\varphi_y(\upsilon) = E(e^{j\,\upsilon\cdot y}) = \int_{-\infty}^{\infty} e^{j\,\upsilon\cdot y} f_y(\upsilon)\; d\upsilon \tag{2.17}$$

Notice the relationship of the characteristic function and the Fourier transform of the PDF.

If the characteristic functions of two random variables agree, then the two variables have the same distribution.

Consider the sum of independent random variables:

$$z = \sum y_i \tag{2.18}$$

Then, the PDF of the sum is the convolution (denoted with an asterisk) of the PDFs of each variable:

$$f_z = f_{y1} * f_{y2} * \ldots * f_{yn} \tag{2.19}$$

And the characteristic function is the product:

$$\varphi_z = \varphi_{y1} \cdot \varphi_{y2} \cdots \varphi_{yn} \tag{2.20}$$

2.3.4 White Noise

White noise is a signal $y(t)$ whose autocorrelation is given by:

$$R(t_1\,,\,t_2) = I(t_1)\,\delta(t_1 - t_2) \tag{2.21}$$

where *δ()* is 1 for $t1 = t2$ and zero elsewhere.

The white noise is important for system identification purposes, and for noise modelling.

2.4 Power Spectra

Up to this point only the time and the values domains have been considered. Now let us have a look to the frequency dimension. Power spectra allow us to get an idea of the frequencies contents of random signals.

2.4.1 Basic Concept

The *power spectrum* of a stationary random variable $y(t)$ is the Fourier transform of its autocorrelation:

$$S_y(\omega) = \int_{-\infty}^{\infty} R(\tau)\, e^{-j\omega t}\; d\tau \tag{2.22}$$

The area of $S_y(\omega)$ equals the average power of $y(t)$:

$$E(y^2) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_y(\omega)\, d\omega \tag{2.23}$$

2.4.2 Example of Power Spectral Density of a Random Variable

Figure 2.12 shows the power spectral density (PSD for short) of a random signal $y(t)$ with log-normal PDF. The PSD has units of power per unit frequency interval (for example, if y(t) is in volts, the PSD is in watts per hertz). The PSD curve in Fig. 2.12 is expressed in decibels. This figure has been generated by the Program 2.11, which uses the *pwelch()* function to compute and plot the PSD. The name of the function refers to the method of Welch to obtain the PSD by repeated application of the Fourier Transform.

The PSD curve in Fig. 2.12 is in general flat, except for a clear peak at 0 Hz. This peak is due to the non-zero DC level of the signal $y(t)$. In order to suppress this peak, the mean of $y(t)$ should be obtained and then subtracted to $y(t)$.

Program 2.11 Power spectral density (PSD) of random signal with log-normal PDF

```
% Power spectral density (PSD) of random signal
% with log-normal PDF
fs=100; %sampling frequency in Hz
tiv=1/fs; %time interval between samples;
t=0:tiv:(40.96-tiv); %time intervals set (4096 values)
N=length(t); %number of data points
mu=0; sigma=1; %random signal parameters
y=lognrnd(mu,sigma,N,1); %random signal data set
nfft=256; %length of FFT
window=hanning(256); %window function
numoverlap=128; %number of samples overlap
pwelch(y,window,numoverlap,nfft,fs);
title('PSD of random signal with log-normal PDF');
```

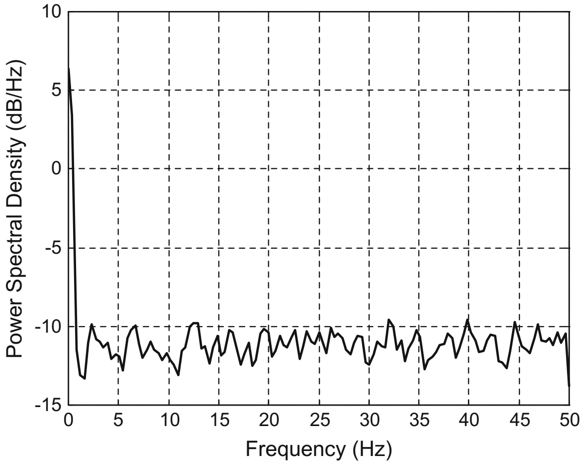


Fig. 2.12 PSD of a random signal with log-normal PDF

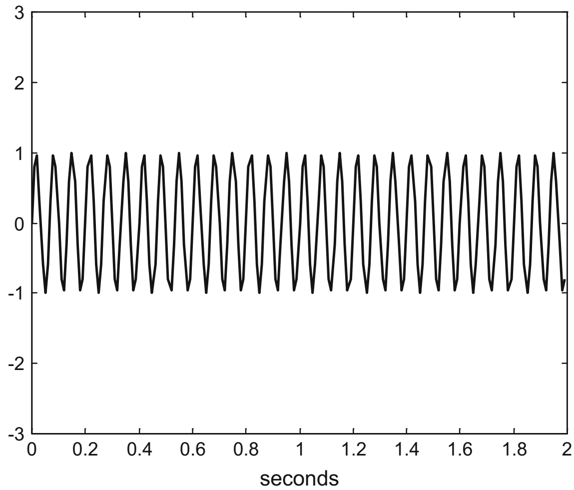


Fig. 2.13 The buried sinusoidal signal

2.4.3 Detecting a Sinusoidal Signal Buried in Noise

Let us consider the following signal:

$$y(t) = \sin(15 \cdot 2 \cdot \pi) + y_n(t) \tag{2.24}$$

where $y_n(t)$ is a zero-mean random signal with normal PDF.

Figure 2.13 shows the 15 Hz sinusoidal signal, and Fig. 2.14 shows the signal $y(t)$. Program 2.12 has been used to generate Fig. 2.14.

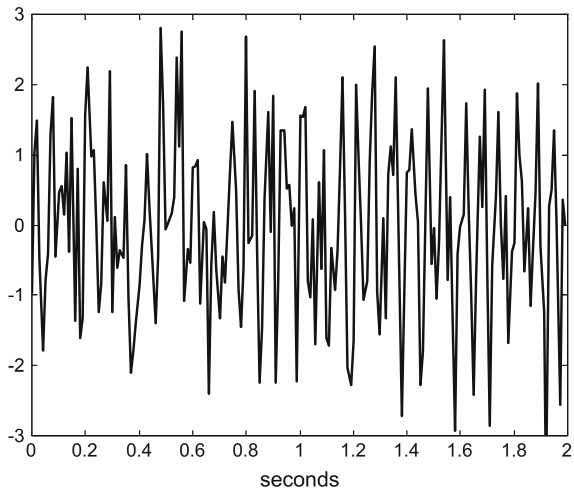


Fig. 2.14 The sine+noise signal

Program 2.12 The sine+noise signal

```
% The sine+noise signal
fs=100; %sampling frequency in Hz
tiv=1/fs; %time interval between samples;
t=0:tiv:(2-tiv); %time intervals set (200 values)
N=length(t); %number of data points
yr=randn(N,1); %random signal data set
ys=sin(15*2*pi*t); %sinusoidal signal (15 Hz)
y=ys+yr'; %the signal+noise
plot(t,y,'k'); %plots sine+noise
axis([0 2 -3 3]);
xlabel('seconds'); title('sine+noise signal');
```

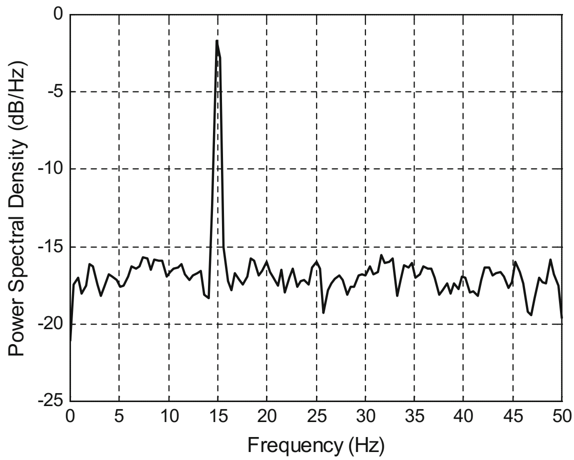
Since we fabricated the signal $y(t)$ we already know there is a sinusoidal signal buried into $y(t)$. However it seems difficult to notice this in Fig. 2.14.

We can use the PSD to detect the sinusoidal signal. Figure 2.15 shows the result. There is a peak on 15 Hz that reveals the existence of a buried 15 Hz signal. Figure 2.15 has been generated with Program 2.13.

Program 2.13 Power spectral density (PSD) of a signal+noise

```
% Power spectral density (PSD) of a signal+noise
fs=100; %sampling frequency in Hz
tiv=1/fs; %time interval between samples;
t=0:tiv:(40.96-tiv); %time intervals set (4096 values)
N=length(t); %number of data points
yr=randn(N,1); %random signal data set
ys=sin(15*2*pi*t); %sinusoidal signal (15 Hz)
y=ys+yr'; %the signal+noise
nfft=256; %length of FFT
window=hanning(256); %window function
```

Fig. 2.15 PSD of the sine+noise signal




```
numoverlap=128; %number of samples overlap
pwelch(y,window,numoverlap,nfft,fs);
title('PSD of a sine+noise signal');
```

2.4.4 *Hearing Random Signals*

Like in Sect. 1.4, we can use MATLAB to hear random signals. Humans are usually good to distinguish subtle details in the sounds.

Figure 2.16 shows 1 s of a random signal with normal PDF, generated by Program 2.14. This program also includes some lines to let us hear 5 s of the same signal.

In case you wished to hear any other of the random signals considered in this chapter, the advice is to confine the signal into an amplitude range $-1 < y < 1$. It is good to plot the signal to see what to do: for instance compute its mean and subtract it to the signal, and then multiply by an opportune constant.

Program 2.14 See and hear a random signal with normal PDF

```
% See and hear a random signal with normal PDF
fs=100; %sampling frequency in Hz
tiv=1/fs; %time interval between samples;
t=0:tiv:(2-tiv); %time intervals set (200 values)
N=length(t); %number of data points
y=randn(N,1); %random signal data set
plot(t,y,'-k'); %plots figure
axis([0 2 -3 3]);
xlabel('seconds');
title('random signal with normal PDF');
fs=6000; %sampling frequency in Hz
```

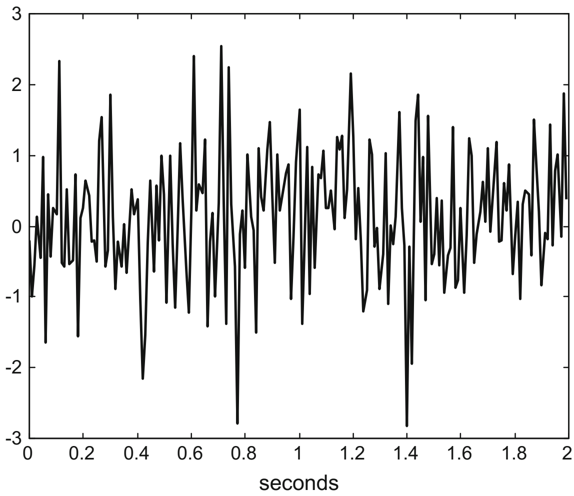


Fig. 2.16 The random signal with normal PDF to be heared

```
tiv=1/fs; %time interval between samples;
t=0:tiv:(5-tiv); %time intervals set (5 seconds)
N=length(t);
y=randn(N,1); %random signal data set
sound(y,fs); %sound
```

2.5 More Types of PDFs

There are many types of PDFs. This section is devoted to add some significant types of PDFs to the three types already presented in Sect. 2.2, [40, 71, 96].

2.5.1 Distributions Related with the Gamma Function

The gamma function has the following expression:

$$\Gamma(\alpha) = \int_0^\infty \upsilon^{\alpha-1}\, e^{-\upsilon}\; d\upsilon \tag{2.25}$$

Figure 2.17, obtained with the Program 2.15, depicts the gamma function.

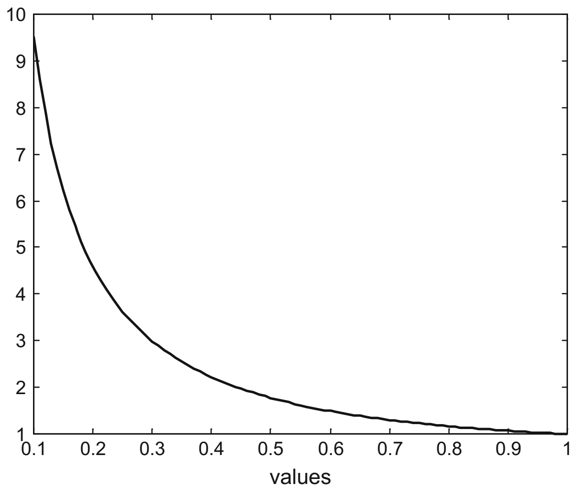


Fig. 2.17 The gamma function

Program 2.15 Gamma function

```
% Gamma function
v=0.1:0.01:1;
ygam=gamma(v);
plot(v,ygam,'k'); hold on;
xlabel('values'); title('gamma function');
```

2.5.1.1 The Gamma PDF

The gamma PDF has the following mathematical expression:

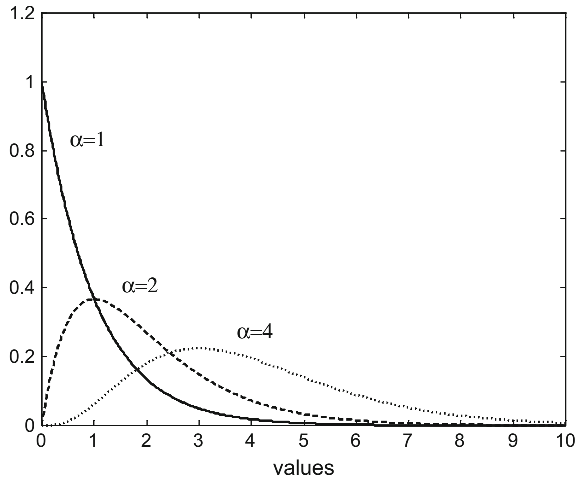
$$f_y(v) = \begin{cases} \frac{v^{\alpha-1}\, e^{-v/\beta}}{\beta^{\alpha}\, \Gamma(\alpha)} & \alpha,\ \beta > 0,\ \ 0 \le v \le \infty \\ 0 & elsewhere \end{cases} \tag{2.26}$$

The gamma distribution corresponds to positively skewed data, such as movement data and electrical measurements. The parameter α is called the rate parameter, and β is called the scale parameter. Figure 2.18 shows three gamma PDFs, corresponding to $\beta = 1$ and three different values of α. The figure has been generated with Program 2.16, which uses the *gampdf()* (*ST) function.

For large values of α the gamma distribution closely approximates a normal PDF.

If y^2 has gamma PDF with $\alpha = 3/2$ and $\beta = 2\alpha$, then y has a Maxwell–Boltzmann PDF.

Fig. 2.18 Gamma-type PDFs



Program 2.16 Gamma-type PDFs

```
% Gamma-type PDFs
v=0:0.01:10; %values set
alpha=1; beta=1; %random variable parameters
ypdf=gampdf(v,alpha,beta); %gamma-type PDF
plot(v,ypdf,'k'); hold on; %plots figure
axis([0 10 0 1.2]);
alpha=2; beta=1; %random variable parameters
ypdf=gampdf(v,alpha,beta); %gamma-type PDF
plot(v,ypdf,'--k'); hold on; %plots figure
alpha=4; beta=1; %random variable parameters
ypdf=gampdf(v,alpha,beta); %gamma-type PDF
plot(v,ypdf,':k'); hold on; %plots figure
xlabel('values'); title('gamma-type PDFs');
```

2.5.1.2 The Exponential PDF

Taking as reference the gamma PDF, the density function for the special case $\alpha = 1$ is called the exponential PDF, thus having the following expression:

$$f_y(v) = \frac{e^{-v/\beta}}{\beta} \quad \beta > 0,\ v \geq 0 \tag{2.27}$$

This PDF is one of the curves in Fig. 2.18. The time intervals between successive random events follow an exponential distribution; this is the case, for example, of life-times of electronic devices.

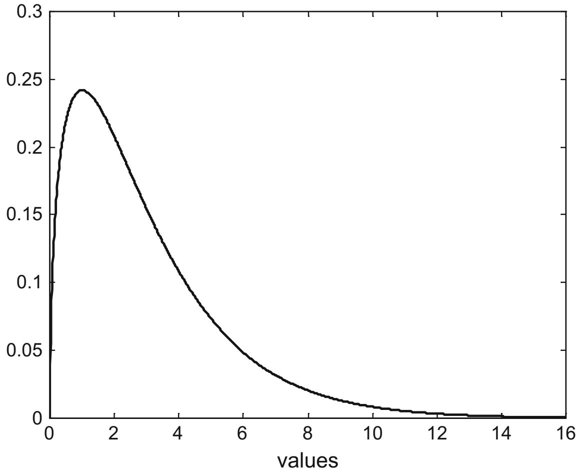


Fig. 2.19 Example of chi-square PDF

2.5.1.3 The Chi-Square PDF

The gamma PDF with parameters $\alpha = \upsilon/2$ and $\beta = 2$ is called a chi-square PDF. The parameter υ is called the "number of degrees of freedom" associated with the chi-square random variable. Figure 2.19 shows an example of chi-square PDF, for $\upsilon = 3$. This figure has been generated with Program 2.17, which uses the *chi2pdf()* (*ST) function.

The sum of υ independent y^2 variables with y having normal PDF is a chi-square signal with υ degrees of freedom.

Program 2.17 Chi-square PDF

```
% Chi-square PDF
v=-3:0.01:16; %values set
nu=3; %random variable parameter ("degrees of freedom")
ypdf=chi2pdf(v,nu); %chi-square PDF
plot(v,ypdf,'k'); hold on; %plots figure
axis([0 16 0 0.3]);
xlabel('values'); title('chi-square PDF');
```

2.5.1.4 The Beta PDF

The beta PDF is defined on a [0..1] interval, according with the following mathematical expression:

$$f_y(\upsilon) = \begin{cases} \frac{\upsilon^{\alpha-1}\,(1-\upsilon)^{\beta-1}}{B(\alpha,\beta)} & \alpha,\ \beta > 0,\ \ 0 \le \upsilon \le 1 \\ 0 & elsewhere \end{cases} \tag{2.28}$$

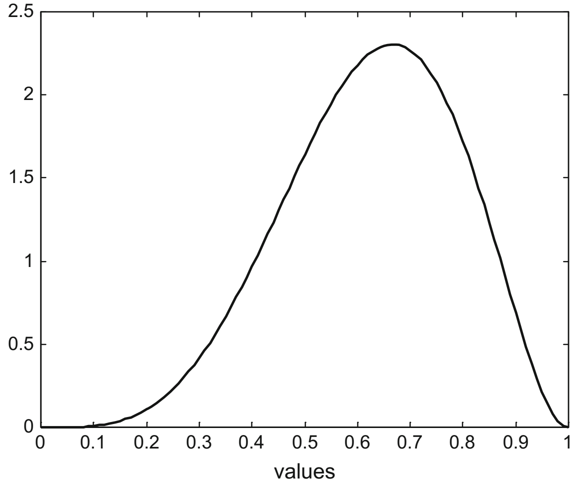


Fig. 2.20 Example of beta PDF

where:

$$B(\alpha, \beta) = \frac{\Gamma(\alpha)\,\Gamma(\beta)}{\Gamma(\alpha + \beta)} \tag{2.29}$$

The parameter α is the first shape parameter and β is the second shape parameter. Figure 2.20 shows an example of beta PDF, corresponding to $\alpha = 5$ and $\beta = 3$. The figure has been generated with Program 2.18, which uses the *betapdf()* (*ST) function.

The beta distribution is used in Bayesian statistics. Events which are constrained to be within an interval defined by a minimum and a maximum correspond to beta distributions; for instance time to completion of a task in project management or in control systems.

For $\alpha = \beta = 1$ the beta distribution is identical to the uniform distribution.

$y_1/(y_1 + y_2)$ has a beta PDF if y_1 and y_2 are independent and have gamma PDF.

Program 2.18 beta PDF

```
% beta PDF
v=0:0.01:1; %values set
alpha=5; beta=3; %random variable parameters
ypdf=betapdf(v,alpha,beta); %beta PDF
plot(v,ypdf,'k'); %plots figure
axis([0 1 0 2.5]);
xlabel('values'); title('beta PDF');
```

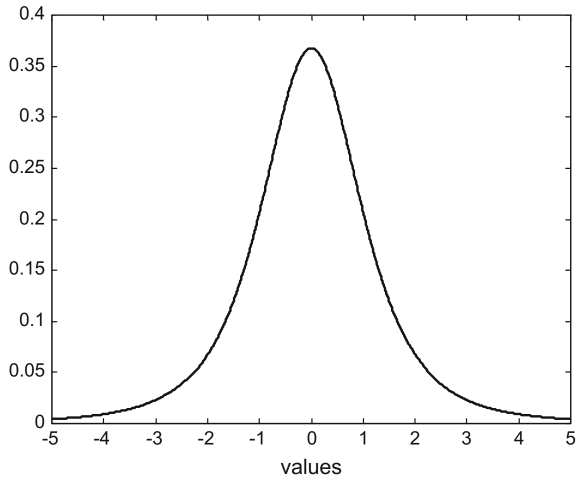


Fig. 2.21 Example of Student's t PDF

2.5.1.5 The Student's t PDF

The Student's t PDF has the following mathematical expression:

$$f_y(\upsilon) = \frac{\Gamma((\upsilon+1)/2)}{\sqrt{\upsilon\,\pi}\,(\upsilon/2)\,(1 + \frac{\upsilon^2}{\upsilon})^{(\upsilon+1)/2}} \tag{2.30}$$

The parameter υ is the number of "degrees of freedom" of the distribution. Figure 2.21 shows an example of Student's t PDF, corresponding to $\upsilon = 3$. The figure has been generated with Program 2.19, which uses the *tpdf()* (*ST) function.

The parameter υ is also the size of random samples of a normal variable; the larger the degrees of freedom, the closer is the PDF to the normal PDF.

Program 2.19 Student's PDF

```
% Student's PDF
v=-5:0.01:5; %values set
nu=3; %random variable parameter ("degrees of freedom")
ypdf=tpdf(v,nu); %Student's PDF
plot(v,ypdf,'k'); hold on; %plots figure
axis([-5 5 0 0.4]);
xlabel('values'); title('Student's PDF');
```

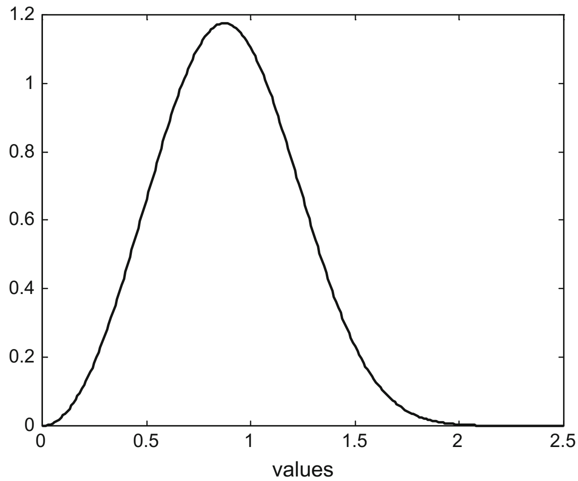


Fig. 2.22 Example of Weibull PDF

2.5.2 Weibull and Rayleigh PDFs

2.5.2.1 The Weibull PDF

The Weibull PDF has the following mathematical expression:

$$f_y(v) = \frac{m\, v^{m-1} e^{-v^m/\alpha}}{\alpha} \qquad \alpha,\ m > 0,\ 0 \leq v < \infty \tag{2.31}$$

The parameter α is the scale parameter and m is the shape parameter. The Weibull distribution is frequently used in reliability studies for time to failure modelling, [2, 103]. If the failure rate decreases over time then $m < 1$, if it is constant $m = 1$, and if it increases $m > 1$. When $m < 1$ it suggests that defective items fail early; when $m = 1$ the failing comes from random events; when $m > 1$ there is "wear out". Figure 2.22 shows an example of Weibull PDF, corresponding to $\alpha = 1$ and $m = 3$ The figure has been generated with Program 2.20, which uses the *weibpdf()* (*ST) function.

For $\alpha = 1$ and $m = 1$ the Weibull distribution is identical to the exponential distribution. For m = 3 the Weibull distribution is similar to the normal distribution.

Program 2.20 Weibull PDF

```
% Weibull PDF
v=0:0.01:2.5; %values set
alpha=1; m=3; %random variable parameters
ypdf=weibpdf(v,alpha,m); %Weibull PDF
plot(v,ypdf,'k'); hold on; %plots figure
axis([0 2.5 0 1.2]);
xlabel('values'); title('Weibull PDF');
```

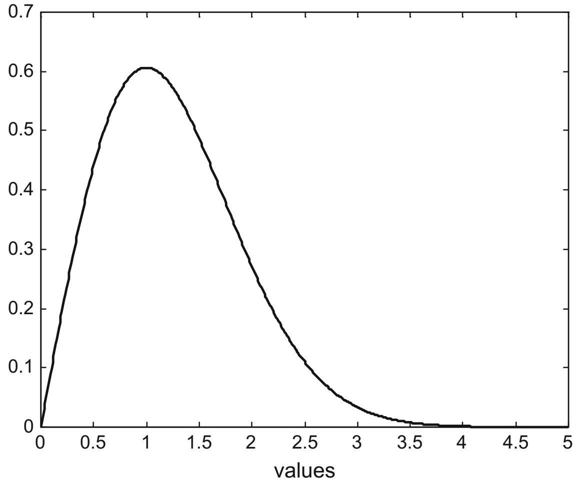


Fig. 2.23 Example of Rayleigh PDF

2.5.2.2 The Rayleigh PDF

The Rayleigh PDF is a particular case of the Weibull PDF, having the following mathematical expression:

$$f_y(v) = \frac{v e^{-v^2/2\beta^2}}{\beta^2} \qquad \beta > 0,\ 0 \le v < \infty \tag{2.32}$$

If y_1 and y_2 are independent random signals with normal PDF and equal variance, then

$$\sqrt{y_1^2 + y_2^2} \tag{2.33}$$

has Rayleigh PDF.

For example the distance of darts from the target in a dart-throwing game has a Rayleigh distribution. Complex numbers with real and imaginary parts being independent random numbers with normal distribution are also examples of Rayleigh distributions. Figure 2.23 shows an example of Rayleigh PDF, corresponding to $\beta = 1$. The figure has been generated with Program 2.21, which uses the *raylpdf()* (*ST) function.

If y has Rayleigh distribution then y^2 is chi-square with 2 degrees of freedom.

Rayleigh distributions are considered in image noise modelling and restoration, [39, 53, 97], wind energy forecasting, [18, 56], reliability studies, [38], etc.

Program 2.21 Rayleigh PDF

```
% Rayleigh PDF
v=0:0.01:5; %values set
beta=1; %random variable parameter
```

```
ypdf=raylpdf(v,beta); %Rayleigh PDF
plot(v,ypdf,'k'); hold on; %plots figure
axis([0 5 0 0.7]);
xlabel('values'); title('Rayleigh PDF');
```

2.5.3 Multivariate Gaussian PDFs

The multidimensional version of the Gaussian PDF is the following:

$$f(\vec{x}) = \frac{1}{(2\,\pi)^{n/2}\,\sqrt{|S|}} \cdot \exp\left(-\frac{1}{2}\,(\vec{x} - \vec{\mu}_x)^T \cdot S \cdot\, (\vec{x} - \vec{\mu}_x) \right) \tag{2.34}$$

where n is the dimension, and S is the covariance matrix:

$$S = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & & \sigma_{2n} \\ - & - & - & - \\ \sigma_{n1} & \sigma_{n2} & & \sigma_n^2 \end{pmatrix} \tag{2.35}$$

In case of two dimensions, the covariance matrix is:

$$S = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{pmatrix} \tag{2.36}$$

Hence, the bivariate Gaussian PDF is:

$$f(\vec{x}) = \frac{1}{(2\,\pi)\,\sqrt{|S|}} \cdot \exp\left(-\frac{1}{2}\,\frac{\sigma_1^2\,\sigma_2^2}{|S|} \cdot Q \right) \tag{2.37}$$

where Q:

$$Q = \left\{ \frac{(x_1 - \mu_1)^2}{\sigma_1^2} - 2\,\sigma_{12}\frac{(x_1 - \mu_1)(x_2 - \mu_2)}{\sigma_1^2\,\sigma_2^2} + \frac{(x_2 - \mu_2)^2}{\sigma_2^2} \right\} \tag{2.38}$$

Figure 2.24 depicts in 3D an example of bivariate Gaussian PDF. The figure has been generated with the Program 2.22, which also generates the Fig. 2.25.

Figure 2.25 shows the probability density information (the same information given by Fig. 2.24) via contour plot, and it clearly highlights that the contours are inclined ellipses: the inclination is due to the cross terms in the covariance matrix.

Fig. 2.24 Example of bivariate Gaussian PDF

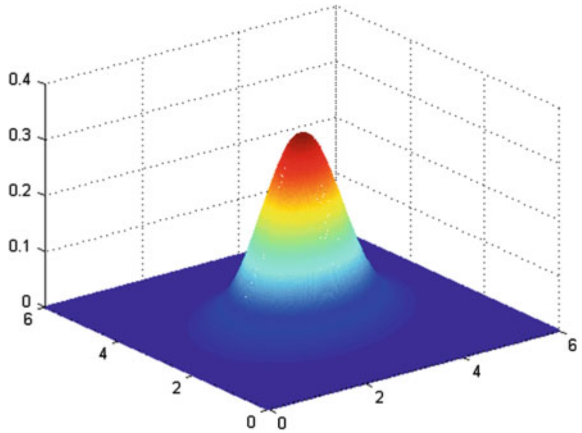
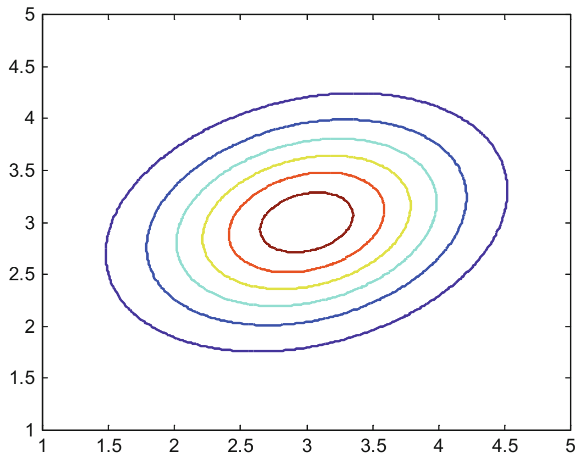


Fig. 2.25 Example of bivariate Gaussian PDF



Program 2.22 Bivariate normal PDF

```
% Bivariate normal PDF
x1=0:0.02:6;
x2=0:0.02:6;
N=length(x1);
%the PDF
mu1=3; mu2=3;
C=[0.4 0.1;
0.1 0.6];
D=det(C);
K=1/(2*pi*sqrt(D)); Q=(C(1,1)*C(2,2))/(2*D);
ypdf=zeros(N,N); %space for the PDF
for ni=1:N,
  for nj=1:N,
    aux1=(((x1(ni)-mu1)^2)/C(1,1))+...
    +(((x2(nj)-mu2)^2)/C(2,2))...
```

```
    -(((x1(ni)-mu1).*(x2(nj)-mu2)/C(1,2)*C(2,1)));
    ypdf(ni,nj)= K*exp(-Q*aux1);
  end;
end;
%display
figure(1)
mesh(x1,x2,ypdf);
title('Bivariate Gaussian: 3D view');
figure(2)
contour(x1,x2,ypdf);
axis([1 5 1 5]);
title('Bivariate Gaussian PDF: top view');
```

2.5.4 Discrete Distributions

Discrete distributions, [46], are related to counting discrete events. Although we continue using the term PDF, it should be considered as a discrete version.

2.5.4.1 The Binomial PDF

If an event occurs with probability q, and we make n trials, then the number of times m that it occurs is:

$$m = \binom{n}{j} q^j (1-q)^{n-j} \tag{2.39}$$

An example of binomial PDF is given in Fig. 2.26 generated with Program 2.23, which uses the *binomial()* (*ST) function.

Program 2.23 Binomial PDF

```
% Binomial PDF
n=20;
ypdf=zeros(1,n);
for k=1:n,
  ypdf(k)=binomial(n,k);
end;
stem(ypdf,'k'); %plots figure
xlabel('values'); title('Binomial PDF');
```

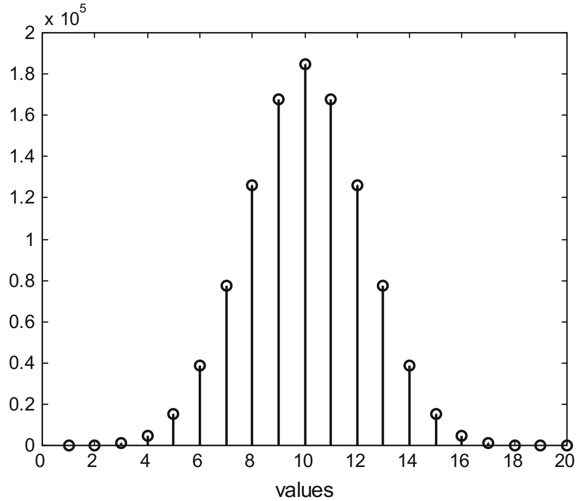


Fig. 2.26 Example of binomial PDF

2.5.4.2 The Poisson PDF

Suppose there is a particular event you are counting along a given time interval T. It is known that the expected count is λ. For instance you know that on average the event occurs 5 times every minute, and $T = 20$ min; then $\lambda = 5 \times 20 = 100$.

The Poisson PDF has the following mathematical expression:

$$f(k) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{2.40}$$

This expression gives the probability that the actual count is k (integer values).

The Poisson distribution is of practical importance. It is used to predict the number of telephone calls, access to a web page, failures of a production chain, performance of a communication channel or a computer network, etc.

Figure 2.27 shows an example of Poisson PDF. It has been generated with the Program 2.24, which uses the *poisspdf()* (*ST) function.

Program 2.24 Poisson PDF

```
% Poisson PDF
lambda=20;
N=50;
ypdf=zeros(1,N);
for nn=1:N,
  ypdf(nn)=poisspdf(nn,lambda);
end;
stem(ypdf,'k'); %plots figure
axis([0 N 0 0.1]);
xlabel('values'); title('Poisson PDF');
```

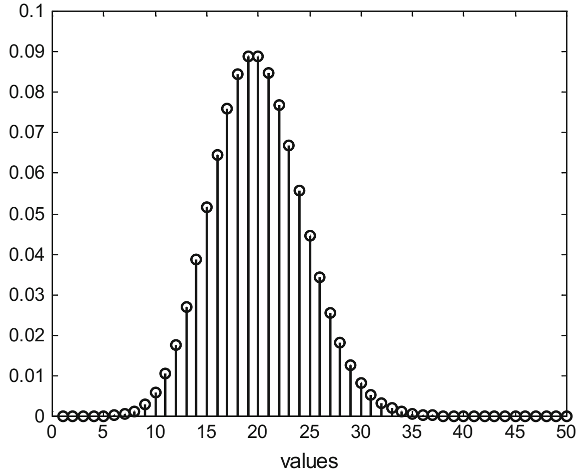


Fig. 2.27 Example of Poisson PDF

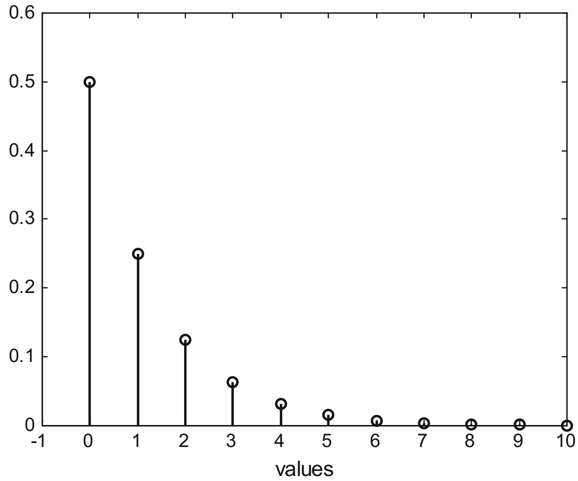


Fig. 2.28 Example of geometric PDF

2.5.4.3 The Geometric PDF

The geometric distribution is a discrete analog of the exponential distribution. As an example, k could be the number of consecutive heads when repeatedly flipping a coin. The probability of heads in each attempt is p (might be 0.5).

The geometric PDF has the following mathematical expression:

$$f(k) = (1-p)^k \cdot p \qquad 0 < p \leq 1 \tag{2.41}$$

Figure 2.28, which has been generated with the Program 2.25, shows an example of geometric PDF. The program uses the *geopdf()* (*ST) function.

Program 2.25 Geometric PDF

```
% Geometric PDF
P=0.5;
N=10;
ypdf=zeros(1,N);
for nn=0:N,
  ypdf(nn+1)=geopdf(nn,P);
end;
stem(0:N,ypdf,'k'); %plots figure
axis([-1 10 0 0.6]);
xlabel('values'); title('Geometric PDF');
```

2.6 Distribution Estimation

Given a data set, we would like to estimate its probability distribution. This section presents some of the available methods for this purpose, [6, 55, 83] (see also [1, 62] for the Weibull distribution). In general, one tries reasonable distribution alternatives (hypotheses), until getting a satisfactory solution.

2.6.1 Probability Plots

There are some graphical representation methods that help to determine an appropriate distribution fitting for a given random signal.

2.6.1.1 Normal Probability

Let us generate with the simple Program 2.26, a random signal with normal PDF, and let us use the function *normplot()* (*ST) to plot the signal data set in a special way. Figure 2.29 shows the result. The signal data, represented with plus signs, look grouped along a straight line: this fact confirms that the signal approximately has a normal PDF.

Program 2.26 Normal probability plot

```
% Normal probability plot
N=200; %200 values
y=randn(N,1); %random signal with normal PDF
normplot(y); % the normal probability plot
```

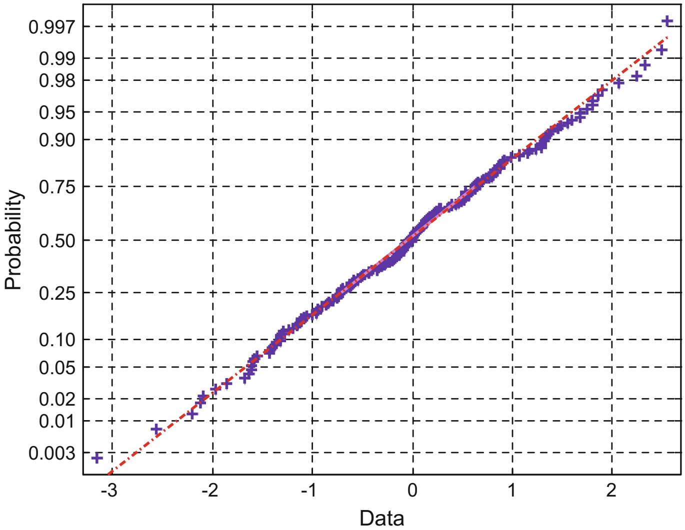


Fig. 2.29 Normal probability plot

2.6.1.2 Weibull Probability

Like in the case of the normal PDF, let us now generate with a few MALAB lines, the Program 2.27, a random signal with Weibull PDF and then plot in a special way the signal data using the function *weibplot()* (*ST). Figure 2.30 shows the result. Again the signal data, represented with plus signs, look grouped along a straight line, confirming that the signal has a Weibull PDF.

Program 2.27 Weibull probability plot

```
% Weibull probability plot
N=200; %200 values
y=weibrnd(2,0.5,N,1); %random signal with Weibull PDF
weibplot(y); % the Weibull probability plot
```

2.6.2 Histogram

Most times, the first thing to do is to look at the histogram of the random data, since it gives a lot of fundamental information.

Supposing there seems to be a good distribution PDF candidate to fit the data, it is convenient first to normalize the histogram. Recall that the area covered by a PDF is one, and so the area of the normalized histogram must be one.

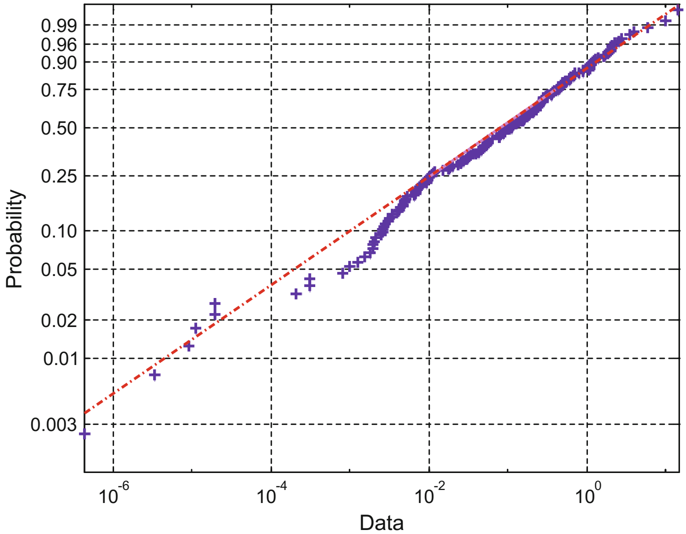


Fig. 2.30 Weibull probability plot

In order to normalize the histogram, it must be divided by the following factor:

$$r = N h \tag{2.42}$$

where N is the number of data, and h is the width of each histogram bin.

The normalized histogram is called the *density histogram*.

A typical problem is to decide how many bins to use for the histogram. There are several published rules. One of them, the *Normal Reference Rule* is the following:

$$h = \left(\frac{24\,\sigma^3 \sqrt{3}}{n} \right)^{1/3} \approx 3.5 \cdot \sigma \cdot N^{-1/3} \tag{2.43}$$

For skewed distributions, Scott proposed the following correction factor:

$$h \approx 3.5 \cdot s \cdot N^{-1/3} \tag{2.44}$$

$$s = \frac{2^{1/3}\,\sigma}{\exp(\frac{5\sigma^2}{4})\,(\sigma^2 + 2)\sqrt{(\exp(\sigma^2) - 1)}} \tag{2.45}$$

2.6.3 Likelihood

Suppose you have a set of data $\vec{x} = (x_1, x_2, \ldots, x_n)$ with a certain PDF. This PDF is characterized by a parameter set $\vec{\theta} = (\theta_1, \theta_2, \ldots, \theta_k)$. For instance, in the case of a Gaussian PDF, the parameters are μ and σ.

Let us express the PDF as $f(\vec{x}|\vec{\theta})$.

We are interested in finding the PDF that is most likely to have produced the data. We define the *'likelihood function'* by reversing the roles of $\vec{x}$ and $\vec{\theta}$:

$$L(\vec{\theta}) = f(\vec{x}\,|\,\vec{\theta}) \tag{2.46}$$

The problem is: given the data, find the PDF parameters.

The maximum likelihood estimate (MLE) of $\vec{\theta}$ is that value of $\vec{\theta}$ that maximises $L(\vec{\theta})$, [57, 98].

Supposing the random data are mutually independent, the likelihood function can be expressed as a product:

$$L(\vec{\theta}) = f(x_1\,|\,\vec{\theta}) \cdot f(x_2\,|\,\vec{\theta}) \cdot \cdot \cdot f(x_n\,|\,\vec{\theta}) \tag{2.47}$$

It is usual, in this context, to use natural logarithms. The *log-likelihood function* is:

$$l(\vec{\theta}) = \sum_i \log\,(f(x_i\,|\,\vec{\theta})) \tag{2.48}$$

For instance, the log-likelihood function corresponding to the Gaussian distribution is:

$$l(\mu, \sigma) = -\frac{n}{2}\,\log(2\pi\sigma^2) - \frac{1}{2\,\sigma^2}\sum_i (x_i - \mu)^2 \tag{2.49}$$

Figure 2.31 shows examples of the log-likelihood function of the Gaussian distribution. We supposed a constant value of the variance, equal to one. The figure has been generated with the Program 2.28, which explores 100 different values of the PDF parameter μ (the mean). The function values have been computed using N random data generated with the *randn()* MATLAB function, using $\mu = 5$. Four values of N have been chosen. Notice that as the number of data increases the curve is sharper. The peak of the curve, the maximum, corresponds to the mean equal to 5.

Program 2.28 Likelihood example

```
% Likelihood example
sig2=1;
%constant
r=1/(2*sig2);
Lh=zeros(4,101); %reserve space
```

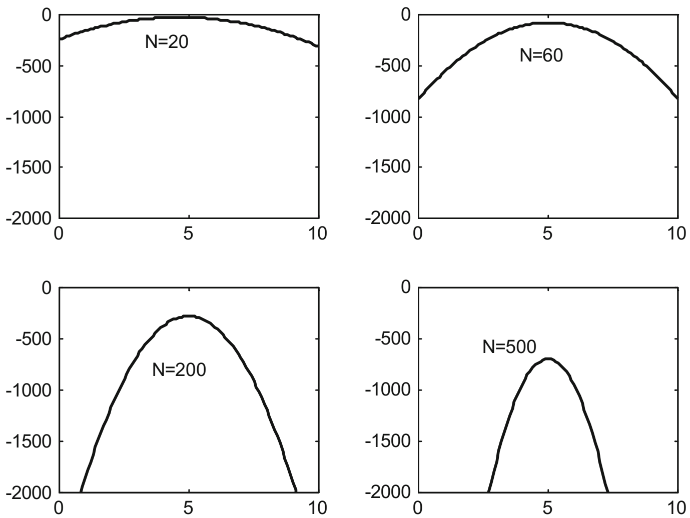


Fig. 2.31 The log-likelihood of the Gaussian distribution, using 20, or 60, or 200, or 500 data values

```
for ni=1:4,
  switch ni
    case 1, N=20;
    case 2, N=60;
    case 3, N=200;
    case 4, N=500;
  end;
  %N is number of data
  %data generation with normal distribution,
  % mean=5, sigma=1
  x=5+randn(1,N);
  K=(-N/2)*log(2*pi*sig2);
  aux=0;
  for nm=1:101,
    mu=(nm-1)/10; %mean
    aux=(x-mu).^2;
    Lh(ni,nm)=K-(r*sum(aux)); %Log-Likelihood
  end;
end;
%display
ex=0:0.1:10;
figure(1)
for ni=1:4,
subplot(2,2,ni),
plot(ex,Lh(ni,:),'k');
axis([0 10 -2000 0]);
end;
```

The maximum of the log-likelihood can be analytically determined, using derivatives, [36]. For instance, in the case of the Gaussian PDF:

$$\frac{\partial l(\vec{\theta})}{\partial \mu} = 0 \rightarrow \mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{2.50}$$

$$\frac{\partial l(\vec{\theta})}{\partial \sigma} = 0 \rightarrow \sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)^2 \tag{2.51}$$

The log-likelihood of the **gamma** PDF is:

$$l(\alpha, \beta) = \sum_i \left((\alpha - 1) \log x_i - \frac{x_i}{\beta} - \alpha \log \beta - \log \Gamma(\alpha) \right) \tag{2.52}$$

The log-likelihood of the **exponential** PDF is:

$$l(\alpha, \beta) = \sum_i \left(-\frac{x_i}{\beta} - \log \beta \right) \tag{2.53}$$

The log-likelihood of the **Weibull** PDF is:

$$l(\alpha, m) = \sum_i \left(m + (m - 1) \log x_i - \frac{x_i^m}{\alpha} - \log \alpha \right) \tag{2.54}$$

The log-likelihood of the **Poisson** PDF is:

$$l(\lambda) = \sum_i (k_i \log \lambda - \lambda - \log (k_i!)) \tag{2.55}$$

The log-likelihood of the **geometric** PDF is:

$$l(p) = \sum_i (k_i \log (1 - p) + \log p) \tag{2.56}$$

2.6.4 The Method of Moments

Let us recall from (2.7) the definition of moment:

$$\mu'_k = E(y^k), \ k = 1, 2, 3 \ldots \tag{2.57}$$

As in the last sub-section, suppose you have a set of data $\vec{y} = (y_1, y_2, \ldots, y_n)$ with a certain PDF. Based on these data, an estimate of the moments can be obtained:

$$\hat{\mu}\prime_k = \frac{1}{n} \sum_i (y_i^k) \tag{2.58}$$

Assume that the PDF parameters, $\vec{\theta} = (\theta_1, \theta_2, \ldots, \theta_k)$, can be written as functions of the moments. For instance, $\theta_1 = h(\mu_1, \mu_2, \mu_3)$.

Now, the idea for the estimation of the parameters is just to use the estimated moments. Continuing with the example: $\hat{\theta}_1 = h(\hat{\mu}_1, \hat{\mu}_2, \hat{\mu}_3)$.

Honouring its name, the moment generating function can be used to actually generate moments, [37]:

$$\frac{d\Gamma}{d\upsilon}(0) = E(y); \quad \frac{d^2\Gamma}{d\upsilon^2}(0) = E(y^2); \ldots; \quad \frac{d^n\Gamma}{d\upsilon^n}(0) = E(y^n) \tag{2.59}$$

For instance, in the case of the Poisson distribution the moment generating function is:

$$\Gamma(\upsilon) = \sum_k e^{\upsilon k} \frac{\lambda^k}{k!} e^{-\lambda} = e^{-\lambda} e^{\lambda \exp(\upsilon)} = Q \tag{2.60}$$

Taking derivatives:

$$\frac{d\Gamma}{d\upsilon} = \lambda e^{\upsilon} Q \tag{2.61}$$

$$\frac{d^2\Gamma}{d\upsilon^2} = \lambda e^{\upsilon} Q + \lambda^2 e^{2\upsilon} Q \tag{2.62}$$

The evaluation at 0 gives:

$$E(y) = \lambda \tag{2.63}$$

$$E(y^2) = \lambda + \lambda^2 \tag{2.64}$$

Clearly, the estimation of the first moment, using the data, is enough for the estimation of λ.

Consider another example: the gamma distribution. The moment generating function is:

$$\Gamma(\upsilon) = \left(\frac{1/\beta}{(1/\beta) - 1} \right)^{\alpha} \tag{2.65}$$

Taking derivatives and evaluating at 0:

$$\frac{d\Gamma}{d\upsilon}(0) = E(y) = \alpha\,\beta \tag{2.66}$$

$$\frac{d^2 \Gamma}{dv^2}(0) = E(y^2) = \alpha\,(\alpha + 1)\,\beta^2 \tag{2.67}$$

Using the estimated first and second moments, we have two equations and the values of α and β can be obtained.

2.6.5 Mixture of Gaussians

A popular way to approximate the PDF of a given random data set, is by using a mixture of well-know PDFs. It can be written as follows:

$$\hat{f}(x) = \sum_k p_k f_k(x) \tag{2.68}$$

where $\hat{f}(x)$ is the estimated PDF, $f_k(x)$ are PDFs (the components of the mixture), and p_k sets the proportions of the mixture (the sum of the p_k is one).

Depending on the *a priori* knowledge on the data, different types of PDFs could be combined: Weibull, beta, Rayleigh, etc.

Nowadays, the use of Gaussians is predominant for many applications, [92]. They are universal approximators of continuous densities given enough Gaussian components.

The use of mixtures is most appropriate for multi-modal PDFs. This is the case chosen for the next example, treated with the Program 2.29. It is a simple example with a bimodal PDF (two peaks).

As it can be seen in the Program 2.29, the random data have been generated by interleaving data from two Gaussian distribution, according with the proportions defined by p.

Figure 2.32 shows the density histogram (the normalized histogram) of the generated data. The figure also shows the shape of the estimated PDF, which is a mixture of two Gaussian PDFs.

Program 2.29 Mixture of 2 Gaussians

```
% Mixture of 2 Gaussians
v=-6:0.02:10; %value set
mu1=0; sigma1=1.5; %parameters of Gaussian 1
mu2=5; sigma2=1; %"""Gaussian 2
ypdf1=normpdf(v,mu1,sigma1); %PDF1
ypdf2=normpdf(v,mu2,sigma2); %PDF2
p=0.4; %mix parameter
%mixed Gaussian PDF
ypdf=(p*ypdf1)+((1-p)*ypdf2);
%random data generation
N=5000;
y=zeros(1,N); %reserve space
for nn=1:N,
  r=rand(1); %uniform PDF
```

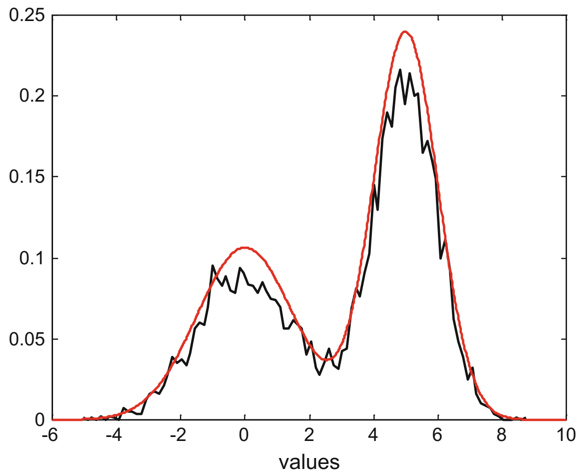


Fig. 2.32 Bimodal distribution and mixture of Gaussians

```
  if r<p,
    y(nn)=mu1+(sigma1*randn(1)); %PDF1
  else
    y(nn)=mu2+(sigma2*randn(1)); %PDF2
  end;
end;
%histogram normalization
nB=100; %number of bins
h=16/100; %bin width
k=N*h;
%display
figure(1)
[nh,xh]=hist(y,100);
plot(xh,nh/k,'k'); hold on; %density histogram
plot(v,ypdf,'r'); %multi-modal PDF
xlabel('values');
title('Mix of 2 Gaussians: histogram and PDF');
```

An example of bimodal distribution is shown in Fig. 2.33. It corresponds to waiting times (minutes) between successive eruptions of the Old Faithful geyser at Yellowstone National Park. See the Resources section for the web address of data. The figure with the histogram has been generated using the Program 2.30.

Program 2.30 Histogram of Bimodal distribution

```
%Histogram of Bimodal distribution
% Geyser eruption data (time between eruptions)
%read data
fer=0;
while fer==0,
fid2=fopen('Geyser1.txt','r');
if fid2==-1, disp('read error')
```

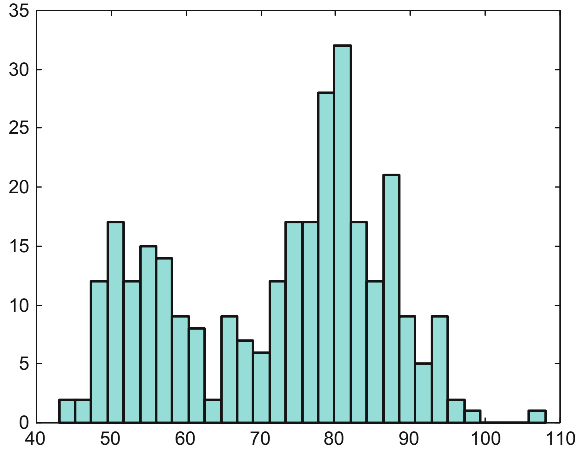


Fig. 2.33 Bimodal distribution example

```
else
y1=fscanf(fid2,'%f \r\n'); fer=1;
end;
end;
fclose('all');
%display
hist(y1,30); colormap('cool');
title('Time between Geyser eruptions');
```

2.6.6 *Kernel Methods*

Again, suppose you have a set of data $\vec{x} = (x_1, x_2, \ldots, x_n)$ with a certain PDF. It was suggested by Parzen (1962) to use the following estimation of the PDF:

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} K(x - x_i) \tag{2.69}$$

where $K()$ is the Parzen window, which is a rectangular window:

$$K(u) = \begin{cases} \frac{1}{2h} \; for \;\; |u| < h \\ 0 \quad otherwise \end{cases} \tag{2.70}$$

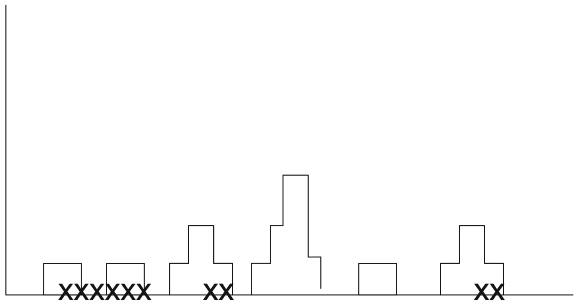


Fig. 2.34 Parzen estimation of PDF

The idea of the Parzen estimation is represented in the Fig. 2.34 for an example having only a few data. It is similar to the histogram. A rectangle of height $1/2h$ and width $2h$ is placed over each datum; heights are added in overlapping zones.

The idea has been extended and refined, choosing other functions—kernel functions—for $K()$, [81, 102].

A popular choice is the Gaussian PDF:

$$K(u) = \frac{1}{\sqrt{2\pi h}} \exp(-u^2/2h^2) \tag{2.71}$$

Some other choices of kernels are the following:

- *Triangular*:

$$K(u) = 1 - \left|\frac{u}{h}\right| \; for \; \left|\frac{u}{h}\right| < 1; \;\; 0 \; otherwise \tag{2.72}$$

- *Biweight*:

$$K(u) = \frac{15}{16}\,(1 - (u/h)^2)^2 \; for \; \left|\frac{u}{h}\right| < 1; \;\; 0 \; otherwise \tag{2.73}$$

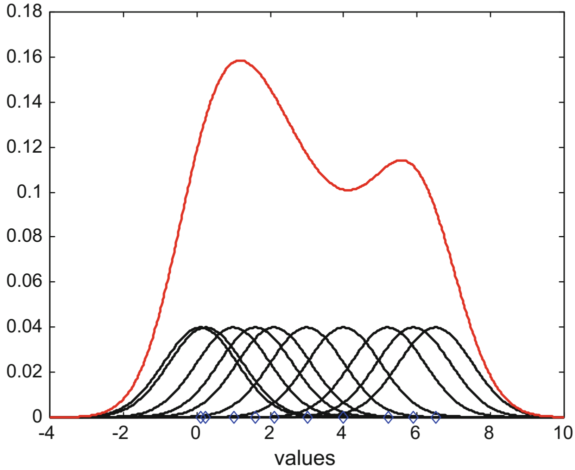
- *Epanechnikov*:

$$K(u) = \frac{0.75 \cdot (1 - 0.2 \cdot (u/h)^2)}{\sqrt{5}} \; for \; \left|\frac{u}{h}\right| < \sqrt{5}; \;\; 0 \; otherwise \tag{2.74}$$

The main problem with the kernel methods is to choose an adequate value for the bandwidth h.

Figure 2.35 shows an example of PDF estimation using a series of Gaussian PDFs with the same bandwidth. The number of Gaussians is given by the number of data points.

Fig. 2.35 Kernel-based estimation of PDF



Program 2.31 Kernel method example

```
% Kernel method example, using Gaussian kernel
v=-4:0.02:10; %set of values
L=length(v); %number of values
% random data:
X=[0.1, 0.25, 1, 1.6, 2.1, 3, 4, 5.2, 5.9, 6.5];
N=length(X); %number of data points
Kpdf=zeros(N,L); % reserve space
h=1; %bandwidth
q=1/(sqrt(2*pi)*h); %constant
for np=1:N,
  for nv=1:L,
    Kpdf(np,nv)=(q/N)*exp((-(v(nv)-X(np))^2)/(2*(h)^2));
  end;
end;
%total PDF
ypdf=sum(Kpdf);
%display
figure(1)
for np=1:N,
  plot(v,Kpdf(np,:),'k'); hold on; %PDF components
end;
plot(v,ypdf,'r'); %total PDF
plot(X,zeros(1,N),'bd'); %the data
axis([-4 10 0 0.18]);
xlabel('values');title('PDF estimation with Kernel method');
```

2.7 Monte Carlo Methods

Random variables could conveniently be used for several computation and evaluation purposes. An illustrative example is given in the next subsection about Monte Carlo integration. The other subsections extend and apply the basic ideas.

Before going into next topics it is convenient to rewrite a small modification of Eq. (2.6), to obtain the expected value of a function $g(x)$:

$$E(g(x)) = \int_{-\infty}^{\infty} g(v) f_x(v)\ dv \tag{2.75}$$

Although the Monte Carlo methods will be introduced here using one-dimensional examples, the real advantage of the methods take place in multi-dimensional problems where deterministic numerical approximations stumble upon combinatorial explosion, [47].

The name *Monte Carlo* was suggested by Nicolas Metropolis in 1949. This name is linked with gambling, Monaco and all that. Statistical simulation has some similarity with it. A little more history on Monte Carlo methods, together with an illustrative tutorial, is given by [50]; a frequently cited introduction is [52].

2.7.1 Monte Carlo Integration

The Monte Carlo integration methodology has proved to be effective in difficult or complicated cases. Let us introduce a series of approaches in this context, [3, 65, 73].

2.7.1.1 A Basic Method

Consider the following example, as represented in Fig. 2.36. There is a curve, given by a certain function $g(x)$, with x between 0 and 10. It is asked to determine the area A covered by the curve.

Program 2.32 Curve and area

```
% Curve and area
%the curve
x=0:0.1:10;
y=0.5+(0.3*sin(0.8*x));
plot(x,y,'k'); hold on;
axis([0 10 0 1]);
for vx=1:1:10,
  nx=vx*10;
  plot([vx vx],[0 y(nx+1)],'g','linewidth',2);
```

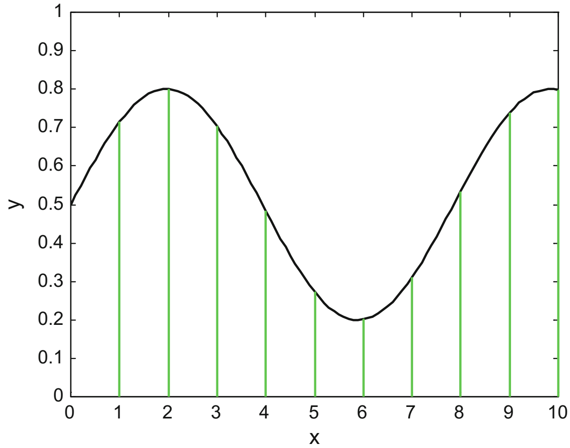


Fig. 2.36 Area covered by a curve

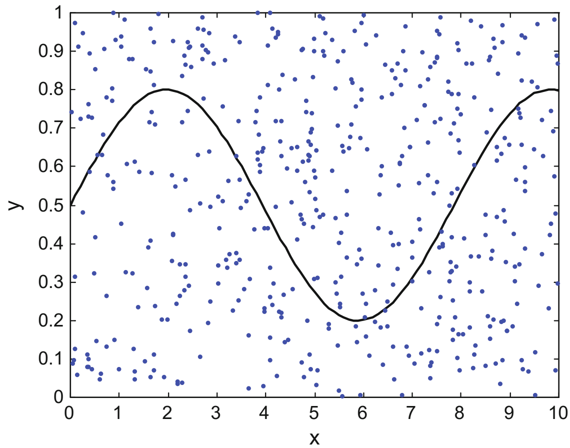


Fig. 2.37 Same as previous figure, but with random points

```
end;
xlabel('x'); ylabel('y');
title('area covered by a curve');
```

Let us generate with the simple Program 2.33 a series of random points on the $x-y$ plane, with x uniformly random between 0 and 10, and y uniformly random between 0 and 1. Figure 2.37 shows these points on the same plane as Fig. 2.36.

Program 2.33 Monte Carlo points, and area approximation

```
% Monte Carlo points, and area approximation
%the curve
x=0:0.1:10;
y=0.5+(0.3*sin(0.8*x));
%the random points
```

```
N=500; %number of points
px=10*rand(1,N); %uniforma distribution
py=rand(1,N); %"""
plot(x,y,'k'); hold on;
plot(px,py,'b.');
axis([0 10 0 1]);
xlabel('x');ylabel('y');
title('curve and random points');
%area calculation
na=0; %counter of accepted points
for nn=1:N,
  xnn=px(nn); ynn=0.5+(0.3*sin(0.8*xnn));
  if py(nn)<ynn, na=na+1; end; %point accepted
end;
%print computed area
%the plot rectangle area is 10
A=(10*na)/N
```

Denote the area of the plane ($10 \times 1 = 10$) as S. The total number of random points is N. Count the *na* points inside A.

Then, one can approximate the area A as follows:

$$\frac{A}{S} \cong \frac{na}{N} \rightarrow A \cong \frac{na}{N} \cdot S \tag{2.76}$$

This is an example of Monte Carlo integration. Notice that we have accepted *nb* points, and *rejected* the rest of the points. Notice that the last part of Program 2.33 provides an implementation of Monte Carlo integration. The last sentence prints the area computation result.

2.7.1.2 Using Expected Values

Suppose one has a certain function $q(x)$ such that:

$$\begin{array}{l} q(x) \geq 0, \quad x \in (a, b) \\ \int_a^b q(x)\, dx = M < \infty \end{array} \tag{2.77}$$

Now, let:

$$p(x) = \frac{q(x)}{M} \tag{2.78}$$

Then $p(x)$ satisfies the conditions for being a PDF. The value M could be obtained using the basic integration method just explained.

If we have to integrate the following:

$$y = \int_a^b g(x)\, q(x)\, dx \tag{2.79}$$

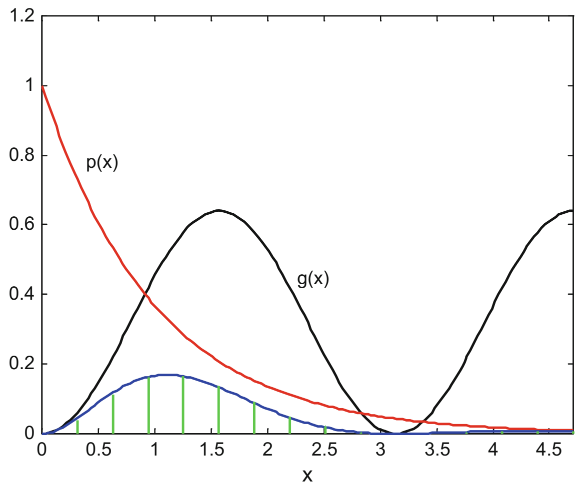


Fig. 2.38 Example of integral of the product $g(x) \cdot p(x)$

This is equivalent to:

$$y = \int_a^b M \cdot g(x)\, p(x)\, dx = M \cdot E(g(x)) \tag{2.80}$$

(recall expression (2.26) with the expected value)

According to the common practice for the computation of expected values, the integral can be approximated with:

$$y \approx M \cdot \frac{1}{n} \sum_{i=1}^{n} g(x_i) \tag{2.81}$$

One draws a set of x_i samples from the $p(x)$ PDF, and then computes the sum of $g(x_i)$.

An example of the integration technique is presented in the Fig. 2.38. To simplify the example, a $p(x)$ has been chosen that directly can be represented with the *exp-pdf()* (*ST) function; that is, $p(x)$ is an exponential function. Likewise, the function *random('exp',..)* has been used to generate samples from $p(x)$ as PDF. This can be seen in the Program 2.34, which generates the figure. The other function $g(x)$ has been chosen as a fragment of sinusoidal signal. The figure includes a plot of the product *p(x) g(x)*, adding some vertical lines to visualize the area which should be the result of the integral.

Program 2.34 Integration as expected value

```
% Integration as expected value
% integral of g(x)*p(x), where p(x) can be taken as a PDF
% the integrand functions
x=0:(pi/100):(1.5*pi); %domain of the integral
g=(0.8*sin(x)).^2; % the function g(x)
mu=1; %parameter of the exponential distribution
p=exppdf(x,mu); %the function p(x) (exponential PDF)
%Deterministic approximation of the integral
aux=abs(g.*p);
disp('deterministic integral result:');
DS=sum(aux)*(pi/100) %print result
%display of the integrand functions
figure(1)
plot(x,g,'k'); hold on;
plot(x,p,'r');
plot(x,aux,'b');
for vx=10:10:151, %mark the integral area
  l=(vx*pi)/100;
  plot([l l],[0 aux(vx)],'g','linewidth',2);
end;
axis([0 1.5*pi 0 1.2]);
title('Integral of the product g(x)p(x)');
xlabel('x');
%Monte Carlo Integration------------------------------------
%draw N samples from p(x) as PDF
N=3000; %number of samples
x=random('exp',mu,1,N); %the samples
%evaluate g(x) at the samples
nv=0; %counter of valid data points
L=1.5*pi; %limit of the integral
for nn=1:N,
  if x(nn)<=L, g(nn)=(0.8*sin(x(nn)))^2; nv=nv+1;
  else
  g(nn)=0; %the value of x is outside integral domain
  end;
end;
%integral
disp('Monte Carlo integral result:');
S=(sum(g)/nv) %print result
```

The Program 2.34 also computes with a deterministic simple approach the integral. For comparison purposes, both the deterministic and the Monte Carlo results are printed when executing the program. Notice that the program includes a protection against trying to operate outside the integral domain.

Coming now to a simpler case:

$$y = \int_a^b f(x)\, dx \tag{2.82}$$

Let us take:

$$g(x) = \frac{f(x)}{p(x)} \tag{2.83}$$

Therefore:

$$y = \int_a^b g(x)\, p(x)\, dx \tag{2.84}$$

Which can be approximated as follows:

$$y \approx \cdot \frac{1}{n} \sum_{i=1}^{n} g(x_i) \tag{2.85}$$

2.7.1.3 Importance Sampling

In the previous approximations a certain $p(x)$ has been used. It is an arbitrary PDF. Several alternatives have been proposed for choosing a $p(x)$ in order to speed up the convergence (some literature refers to it as variance reduction).

A key observation is that in:

$$y \approx \cdot \frac{1}{n} \sum_{i=1}^{n} \frac{f(x_i)}{p(x_i)} \tag{2.86}$$

it is convenient that $p(x) \approx f(x)$ in order to avoid negligible terms. If this is done, most samples of $f(x)$ will be taken where $f(x)$ is larger, and so it is termed as *importance sampling,* [19].

Let us put an example of *not* using importance sampling. The example is represented in Fig. 2.39. A uniform PDF is chosen for $p(x)$ along a wide range. What we see on this figure is that a large part of $p(x)$ is of no use since there are many samples with $f(x_i) = 0$.

Figure 2.39 is useful also for noticing a possible problem. If $p(x)$was narrowed so it fits inside $f(x)$ there would be samples where:

$$\frac{f(x_i)}{p(x_i)} \approx \infty \tag{2.87}$$

This should be avoided. In general, the advice is to use $p(x)$ with long tails.

A better option for $p(x)$ is shown in Fig. 2.40. It is clear that $p(x)$ is similar to the $f(x)$ to be integrated, and that it covers the tails of $f(x)$. The case has been treated with the Program 2.35. Notice that the program includes a protection against division by zero. In this example, the function $p(x)$ corresponds to a beta PDF, so one can use MATLAB (*ST) functions. The program also prints, for comparison, the results of the deterministic and the importance sampling integrations.

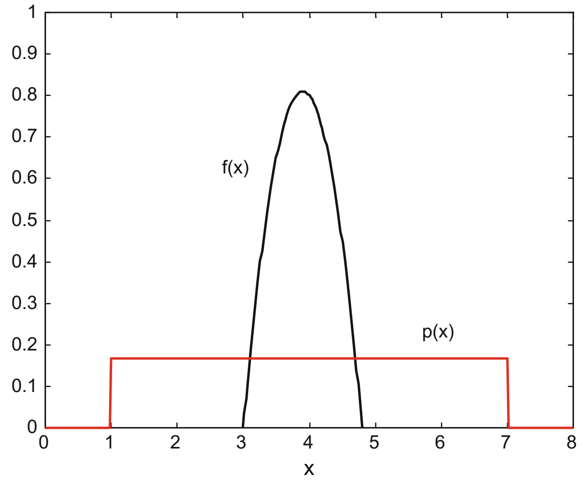


Fig. 2.39 Example of un-importance sampling

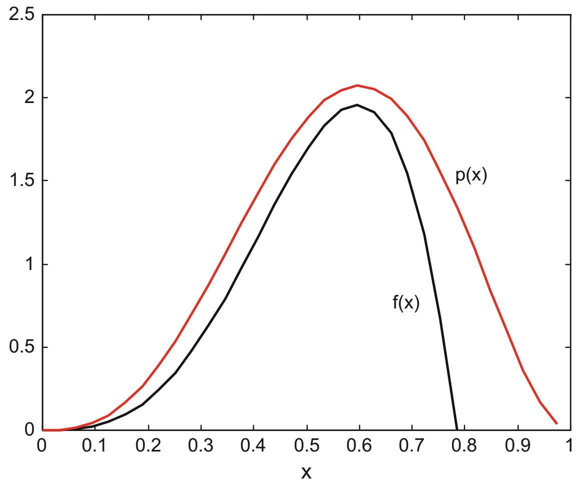


Fig. 2.40 Example of f(x), and p(x) for importance sampling

Program 2.35 Integration as expected value: Importance sampling

```
% Integration as expected value: Importance sampling
% integral of f(x), an appropriate p(x) PDF is taken
% the integrand functions
x=0:(pi/100):(0.25*pi); %domain of f(x)
xx=0:(pi/100):1; %domain of p(x)
f=25*(x.^3).*cos(2*x); %the function f(x)
alpha=4; beta=3; %parameters of the PDF
p=betapdf(xx,alpha,beta); %the function p(x) (beta PDF)
%Deterministic approximation of the integral
disp('deterministic integral result:');
DS=sum(f)*(pi/100) %print result
```

```
%display of f(x) and p(x)
figure(1)
plot(x,f,'k'); hold on;
plot(xx,p,'r');
axis([0 1 0 2.5]);
title('importance sampling: f(x) and p(x)');
xlabel('x');
%Monte Carlo Integration------------------------------------
%draw N samples from p(x) PDF
N=2000; %number of samples
x=random('beta',alpha,beta,1,N); %the samples
%evaluate g(x) at the samples
nv=0; %counter of valid data points
g=0; %initial value
for nn=1:N,
  if x(nn)>0, %avoid division by zero
  if x(nn)<=(0.25*pi), %values inside f() domain
    f=25*(x(nn).^3).*cos(2*x(nn)); %evaluate f() at xi
  else
    f=0;
end;
p=betapdf(x(nn),alpha,beta);
g=g+(f/p); %adding
nv=nv+1;
end;
end;
%integral
disp('Monte Carlo integral result:');
S=(g/nv) %print result
```

Let us consider again the integration of:

$$y = \int_a^b g(x)\, p(x)\, dx \tag{2.88}$$

It can be written as:

$$y = \int_a^b g(x)\, \frac{p(x)}{h(x)} h(x)\, dx \tag{2.89}$$

Denote:

$$w(x) = \frac{p(x)}{h(x)} \tag{2.90}$$

as ‘weight function’.

Then, the approximation is:

$$y \approx \cdot \frac{1}{n} \sum_{i=1}^{n} (g(x_i) \cdot w(x_i)) \tag{2.91}$$

The function $h(x)$ is a *proposed* PDF, as close as possible to $p(x)$, and the samples x_i are drawn from the $h(x)$ PDF.

2.7.2 Generation of Random Data with a Desired PDF

Several PDFs, provided by MATLAB, have been presented in this chapter. With the function *random()* (*ST) it is possible to select a PDF among a set of alternatives, and then use the function to generate random numbers from the selected PDF.

In order to be open for more options, it is convenient to study how to generate random numbers from any desired PDF, [27, 91].

In this subsection, two methods will be introduced: the first is based on inversion of the distribution function, [66]; the second is based on rejection. Other methods will be described later on, in the section on Markov processes.

2.7.2.1 Inversion Sampling

For easier description, let us denote distribution functions as $F(x)$ (recall Sect. 2.2.1). The value of a distribution function is in the range 0..1 and increases or keep cosntant as x increases.

In our case, we wish to obtain a set of samples obeying to a desired distribution $F(x)$. Denote as F^{-1} the inverse of F.

Let us draw a set of samples U with values between 0 and 1 from a uniform PDF. Then, the set of samples:

$$Z = F^{-1}(U) \tag{2.92}$$

obeys to the desired distribution

Figure 2.41 depicts the idea of sample generation.

Therefore the procedure is: (a) draw a sample y_i from uniform PDF; (b) compute $z_i = F^{-1}(y_i)$; and go back to (a), until sufficient data have been obtained.

For example, it is desired to generate a set of samples with a sinusoidal distribution function as depicted on the left part of Fig. 2.42. The corresponding PDF (the derivative) is depicted on the right part of this figure.

In this example, it is easy to analytically obtain the inverse of the distribution function: the inverse of *sin(x)* is *arc sin(x)*. Moreover, MATLAB actually provides the *asin()* function.

The inversion procedure is implemented with the Program 2.36. It generates the set of samples with the desired distribution, and displays Figs. 2.42 and 2.43. This last figure is an histogram of the generated data.

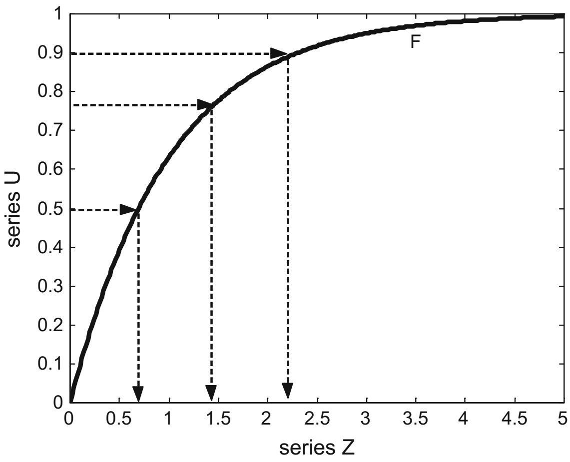


Fig. 2.41 The inversion procedure

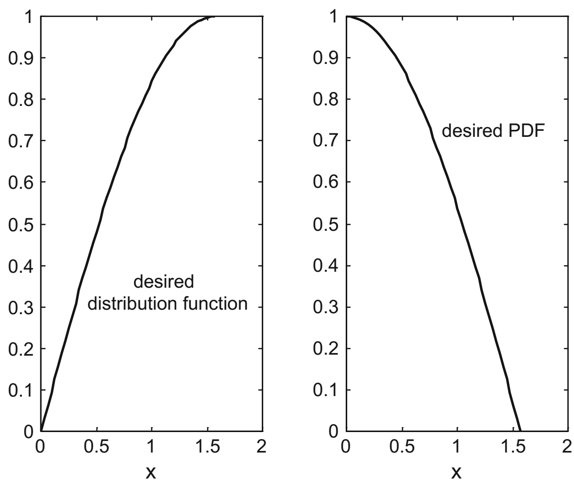


Fig. 2.42 Example of desired distribution function and PDF

Program 2.36 Generation of random data with a desired PDF

```
% Generation of random data with a desired PDF
% using analytic inversion
% example of desired distribution function
x=0:(pi/100):(pi/2);
F=sin(x); %an always growing curve
pf=cos(x); %PDF=derivative of F
% generation of random data
N=2000; %number of data
y=rand(1,N); %uniform distribution
% random data generation:
z=asin(y); %the inverse of F
figure(1)
```

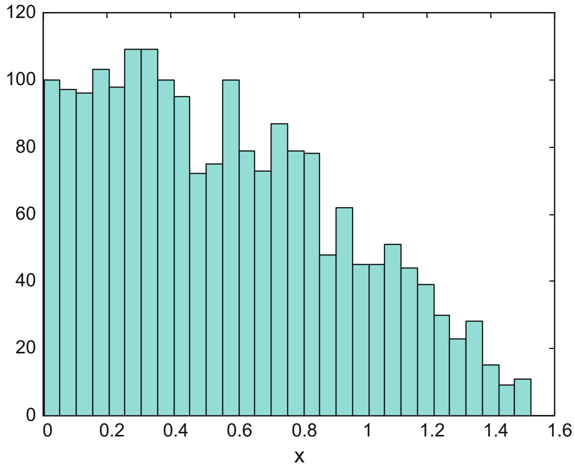


Fig. 2.43 Histogram of random data generated using analytical inversion

```
subplot(1,2,1)
plot(x,F,'k');
xlabel('x'); title('desired distribution function');
subplot(1,2,2)
plot(x,pf,'k');
xlabel('x'); title('desired PDF');
figure(2)
hist(z,30); colormap('cool');
xlabel('x');title('histogram of the generated data');
```

Here is a set of analytical inverses of distribution functions.

Exponential:

PDF	F	F^{-1}
e^{-x}, $x > 0$	$(1 - e^{-x})$	$\log(\frac{1}{U})$

Weibull (simple version):

PDF	F	F^{-1}
$m\,x^{m-1} \cdot e^{-x^m}$, $x > 0$	$(1 - e^{-x^m})$	$\left(\log(\frac{1}{U})\right)^{1/m}$

Cauchy:

PDF	F	F^{-1}
$\frac{1}{\pi(1+x^2)}$	$(\frac{1}{2} + \frac{1}{\pi}$ arctan $x)$	$\tan(\pi\,U)$

Pareto:

PDF	F	F^{-1}
$\frac{a}{x^{a+1}}$, $a > 0,\ x > 1$	$(1 - \frac{a}{x^a})$	$(\frac{1}{U^{1/a}})$

In case of difficulty with the analytical inversion, it is still possible to numerically obtain the inversion. Program 2.37 gives an example of it, continuing with the previous example. Figure 2.44 compares the numerical result with the analytical result: they are essentially the same.

Program 2.37 Numerical inversion of a function

```
% Numerical inversion of a function
% example of F=sin(x), in the growing interval
M=1001;
y=0:0.001:1; %F between 0 and 1
x=zeros(1,M);
%incremental inversion
aux=0; dax=0.001*pi;
for ni=1:M,
  while y(ni)>sin(aux),
    aux=aux+dax;
  end;
  x(ni)=aux;
end;
plot(y,asin(y),'gx'); hold on; %analytical inversion
plot(y,x,'k'); %result of numerical inversion
xlabel('y'); ylabel('x');
title('numerical and analytical inversion of F');
```

To complete the example, the Program 2.38 obtains a set of samples with the desired PDF, using numerical inversion. Figure 2.45 shows the histogram.

Fig. 2.44 Generation of random data with a certain PDF

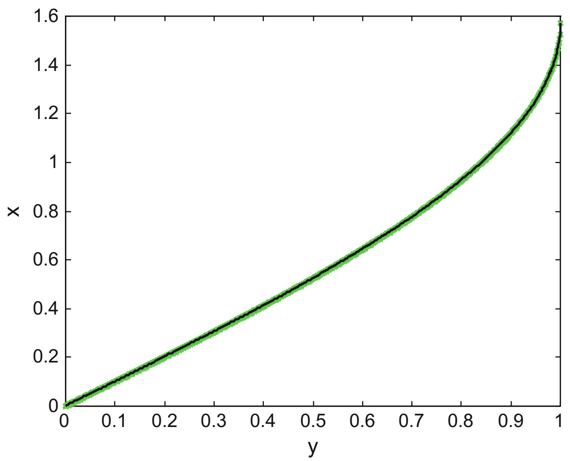
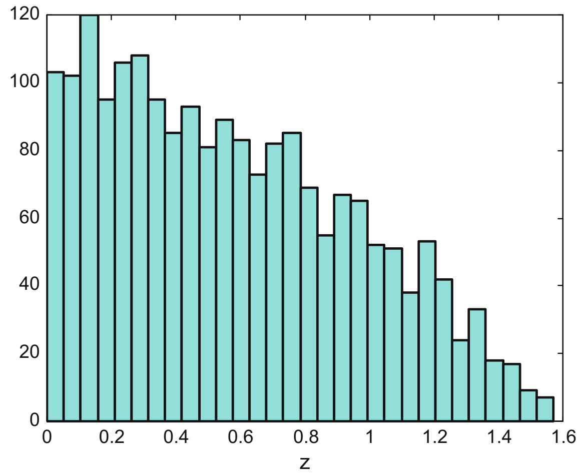


Fig. 2.45 Histogram of random data generated using numerical inversion



Program 2.38 Generation of random data with a desired PDF

```
% Generation of random data with a desired PDF
% using numerical inversion
%first: a table with the inversion of F
M=1001;
y=0:0.001:1; %F between 0 and 1
x=zeros(1,M);
%incremental inversion
aux=0; dax=0.001*pi;
for ni=1:M,
  while y(ni)>sin(aux),
    aux=aux+dax;
  end;
  x(ni)=aux;
end;
%second: generate uniform random data
N=2000; %number of data
ur=rand(1,N); %uniform distribution
%third: use inversion table
z=zeros(1,N);
for nn=1:N,
  %compute position in the table:
  pr=1+round(ur(nn)*1000);
  z(nn)=x(pr); %read output table
end;
%display histogram of generated data
hist(z,30); colormap('cool');
xlabel('z');
title('histogram of the generated data');
```

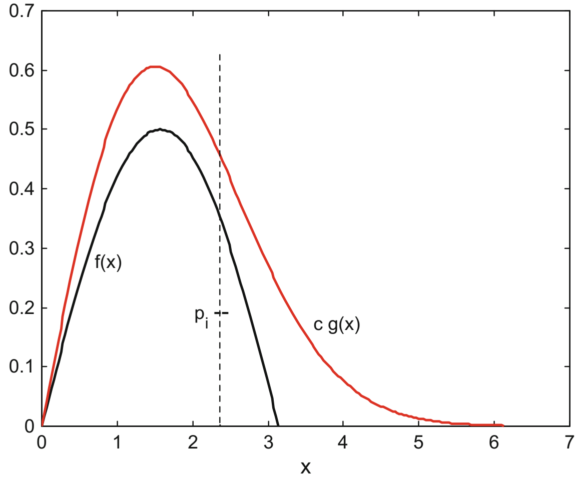


Fig. 2.46 Example of desired f(x) PDF and proposal g(x) PDF

2.7.2.2 Rejection Sampling

The desired PDF is $f(x)$. A proposal $g(x)$ PDF is chosen, such that:

$$\frac{f(x)}{g(x)} \leq c \; for \; all \, x \tag{2.93}$$

where c is a positive constant. Figure 2.46 shows an example, and illustrates the procedure explained below.

The procedure is: (a) generate a sample v_i from the $g(x)$ PDF; (b) generate a sample u_i from uniform PDF on (0, 1); (c) if:

$$p_i = u_i \cdot c \cdot g(v_i) \; < f(v_i) \tag{2.94}$$

then accept v_i, else reject; (d) back to (a) until sufficient data have been obtained, [82].

Program 2.39 provides an implementation of the rejection procedure, with the same desired *sin()* PDF as before. Figure 2.47 shows an histogram of the generated data.

Program 2.39 Generation of random data with a desired PDF

```
% Generation of random data with a desired PDF
% Using rejection method
% example of desired PDF
x=0:(pi/100):pi;
dpf=0.5*sin(x); % desired PDF
% example of proposal PDF
xp=0:(pi/100):pi+3;
ppf=raylpdf(xp,1.5);
```

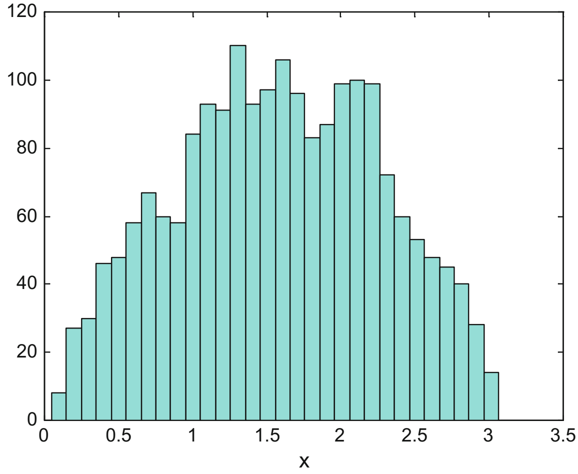


Fig. 2.47 Histogram of random data generated using the rejection method

```
%factor
c=1.5;
% generation of random data
N=2000; %number of data
z=zeros(1,N); %space for data to be generated
for nn=1:N,
  accept=0;
  while accept==0,
    v=raylrnd(1.5,1,1); %Rayleigh distribution
    u=rand(1,1); %uniform distribution
    if v<=pi, %v must be inside dpf domain
      P=u*c*raylpdf(v,1.5);
      L=0.5*sin(v);
      if P<L, z(nn)=v; accept=1; end; %accept
    end;
  end;
end;
figure(1)
plot(x,dpf,'k'); hold on;
plot(xp,c*ppf,'r');
xlabel('x'); title('desired PDF and proposal PDF');
figure(2)
hist(z,30); colormap('cool');
xlabel('x');title('histogram of the generated data');
```

2.7.2.3 Other Methods

There are a number of transformations that can be used to generate random variables with a desired PDF.

For instance, the method of Box and Müller obtains a pair of Gaussian variables as follows, [13]:

$$X = \sqrt{\log\left(\frac{1}{U_1}\right)} \cdot \cos(2\pi\, U_2) \tag{2.95}$$

$$Y = \sqrt{\log\left(\frac{1}{U_1}\right)} \cdot \sin(2\pi\, U_2) \tag{2.96}$$

where U_1 and U_2 are independent uniform [0, 1] random variables.

The random cosine is also used by other methods, which are called *polar methods*, [33, 80]. A symmetric beta distribution (with $\alpha = \beta$) is obtained using:

$$X = \frac{1}{2}\,(1 + \sqrt{1 - U_1^V} \cdot \cos(2\pi\, U_2)) \tag{2.97}$$

where: $V = \frac{2}{2\alpha - 1}$.

The Student's t distribution can be obtained using:

$$X = \sqrt{a(U_1^{-2/a} - 1)} \cdot \cos(2\pi\, U_2) \tag{2.98}$$

2.8 Central Limit

There are two main alternative formulations of the Central Limit Theorem (CLT). The first alternative is related to the distribution of means; while the second alternative is related to sums of random data sets. In both cases, one has several random data sets of size n: $X_1, X_2, \ldots, X_K$. The data sets are independent with equal distribution. The variances of the data set are finite. Suppose K tends to infinity, then:

1. Take the means $\mu_1, \mu_2, \ldots, \mu_K$ of each data set. CLT establishes that these means form a random data set with normal (Gaussian) distribution.
2. The sum of $X_1, X_2, \ldots, X_K$ is also a random data set with normal (Gaussian) distribution.

It does not matter what the distribution of the data sets $X_1, X_2, \ldots, X_K$ is.

The reader is invited to repeatedly convolve any PDF with itself (recall 2.3.3, about the characteristic function), the result always tend to a Gaussian PDF. Perhaps the most dramatic example is when you use a uniform PDF for this exercise.

In the case of products of positive random data sets, the logarithm will tend to a normal distribution, and the product itself will tend to a log-normal distribution.

For the interested reader it is recommended to examine the topic of stable distributions, [12, 61]. Particular cases of stable distributions are the normal distribution, the Cauchy distribution and the Lévy distribution. If the random data sets have not finite variance (this can be observed on the PDF tails), the sum may still tend to a stable distribution.

There are some variants of the CLT, [79]. In particular, the Lyapunov CLT and the Lindeberg CLT require the random data sets to be independent but not necessarily to have the same PDF.

Consider again $X_1, X_2, \ldots, X_K$; each data set has a mean $\mu_i$and variance σ_i^2. Define $s_n^2 = \sum_i \sigma_i^2$ and $Y_i = X_i - \mu_i$. If there exists $\delta > 0$ such that:

$$\lim_{n \to \infty} \frac{1}{s_n^{2+\delta}} \sum_{i=1}^{n} E(|Y_i|^{2+\delta}) = 0 \tag{2.99}$$

then the sum of Y_i / s_n tends to a normal distribution. This is the Lyapunov CLT.

The Lindenberg CLT is similar, [41], but using as condition that for every $\varepsilon > 0$:

$$\lim_{n \to \infty} \frac{1}{s_n^2} \sum_{i=1}^{n} E(|Y_i|^2 \, I(|\, Y_i| \geq \varepsilon \, s_n)) = 0 \tag{2.100}$$

where $I()$ is the indicator function.

Both are sufficient conditions. The Lyapunov condition is stronger than the Lindenberg condition.

In the next chapters some sound files will be used for several purposes. These sounds are quite different: music, animal sounds, sirens… The Program 2.40 just reads a set of 8 sounds, and adds the corresponding data. Figure 2.48 shows a histogram of the result: it exhibits a Gaussian shape, as predicted by the central limit theorem. The final sentence of the Program let you hear the accumulated signal.

Program 2.40 Central limit of wav sounds

```
%Central limit of wav sounds
%read a set of sound files
[y1,fs]=wavread('srn01.wav'); %read wav file
[y2,fs]=wavread('srn02.wav'); %read wav file
[y3,fs]=wavread('srn04.wav'); %read wav file
[y4,fs]=wavread('srn06.wav'); %read wav file
[y5,fs]=wavread('log35.wav'); %read wav file
[y6,fs]=wavread('ORIENT.wav'); %read wav file
[y7,fs]=wavread('elephant1.wav'); %read wav file
[y8,fs]=wavread('harp1.wav'); %read wav file
%Note: all signals have in this example fs=16000
N=25000; %clip signals to this length
y=zeros(8,N); %signal set
y(1,:)=y1(1:N)'; y(2,:)=y2(1:N)';
```

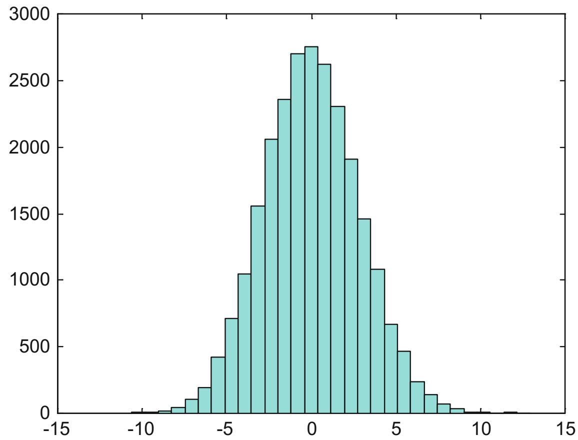


Fig. 2.48 Histogram of sum of signals

```
y(3,:)=y3(1:N)'; y(4,:)=y4(1:N)';
y(5,:)=y5(1:N)'; y(6,:)=y6(1:N)';
y(7,:)=y7(1:N)'; y(8,:)=y8(1:N)';
%normalization
for nn=1:8,
  s=y(nn,:); s=s-mean(s); %zero mean
  vr=var(s); s=s/sqrt(vr); %variance=1
  y(nn,:)=s;
end;
%sum of signals
S=sum(y);
%histogram
figure(1)
hist(S,30); colormap('cool');
title('histogram of the sum of signals');
%sound of the sum
soundsc(S,fs);
```

2.9 Bayes' Rule

According with the Stanford Encyclopedia of Philosophy (web site cited in the Resources section), "the most important fact about conditional probabilities is undoubtedly Bayes' Theorem, whose significance was first appreciated by the British cleric Thomas Bayes (1764)".

Nowadays, the recognition given to the Bayes approach is rapidly extending in several methodologies and fields of activity, like estimation, modelling, decision taking, etc.

A reference book on Bayesian Theory is [10]. In addition, [30] provides a detailed history of how the Bayesian methodology has evolved.

This section has three parts, following a logical order. First, the concept of conditional probability is introduced. Then, the Bayes' rule is enounced, and illustrated with the help of some figures. Finally, a brief introduction of Bayesian networks is made.

2.9.1 Conditional Probability

Let us introduce the concept of conditional probability using an example.

There was a factory producing hundreds of a certain device. The products were tested before going to the market.

Each device could be 'good' or 'bad' (it works well, or not). The situation is that 2 % of the devices are bad. Then, there are two probabilities:

$$P(good) = 0.98\,; \quad P(bad) = 0.02 \tag{2.101}$$

The test says 'accept' or 'reject'. Sometimes the test is erroneous:

- In the case of good devices there are two probabilities:

$$P(accept\,|good) = 0.99\,; \quad P(reject\,|good) = 0.01 \tag{2.102}$$

- In the case of bad devices there are two probabilities:

$$P(accept\,|bad) = 0.03\,; \quad P(reject\,|bad) = 0.97 \tag{2.103}$$

Conditional probability is expressed as $P\,(A|B)$: the conditional probability of A given B.

Unconditional probability $P(A)$ of the event A, is the probability of A regardless of what happens with B. The unconditional probability is also denoted as 'prior' or *'marginal'* probability, and also 'a priori'.

The conditional probability $P\,(A|B)$ could also be denoted as 'posterior', or 'a posteriori' probability.

Notice in the example of the device that good or bad refers to the state of the device, and accept or reject pertains to measurements. This point of view is important in the context of Kalman filters.

The *joint probability* is the probability of having both A and B events together. The joint probability is denoted as $P\ (A \cap B)$ (or $P\ (AB)$, or $P(A, B)$). Recall that two events A and B are independent if:

$$P(A \cap B) = P(A)\,P(B) \tag{2.104}$$

2.9.2 Bayes' Rule

The conditional probability and the joint probability are related by the formula:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{2.105}$$

Likewise:

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \tag{2.106}$$

Combining the two equations:

$$P(A|B)\ P(B) = P(A \cap B) = P(B|A)\,P(A) \tag{2.107}$$

Therefore:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \tag{2.108}$$

This is the famous Bayes' rule.

Let us return to the devices example. There will be four cases:

(a) $P(accept\ \cap\ good) = P(accept|\,good)\,P(good) = 0.9702$

(b) $P(accept\ \cap\ bad) = P(accept|\,bad)\,P(bad) = 0.0006$

(c) $P(reject\ \cap\ good) = P(reject|\,good)\,P(good) = 0.0098$

(d) $P(reject\ \cap\ bad) = P(reject|\,bad)\,P(bad) = 0.0194$

The rejected devices are:

$$P(reject) = P(reject \cap good) + P(reject \cap bad) = 0.0292 \tag{2.109}$$

And the probability of a rejected device to be a good device:

$$P(good|reject) = \frac{P(good \cap reject)}{P(reject)} = \frac{0.0098}{0.0292} = 0.335 \tag{2.110}$$

Now, let us introduce conditional PDFs. Given two random variables x_1 and x_2, the conditional PDF of x_1 given x_2 is:

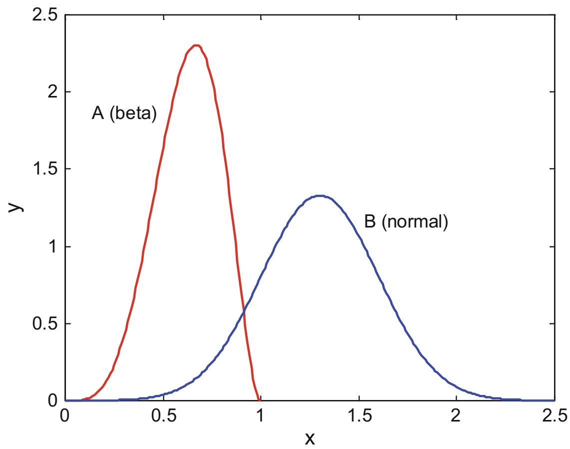


Fig. 2.49 The PDFs of two variables A and B

$$f(x_1|x_2) = \frac{f(x_1 \cap x_2)}{f(x_2)} \tag{2.111}$$

The Bayes' rule can be generalized to conditional PDFs.

$$f(x_1|x_2) = \frac{f(x_2|x_1)f(x_1)}{f(x_2)} \tag{2.112}$$

Here is the Chapman-Kolmogorov equation about the product of two conditional PDFs:

$$f(x_1|(x_2 \cap x_3 \cap x_4))f((x_2 \cap x_3)\,|x_4) = f((x_1 \cap x_2 \cap x_3)|x_4) \tag{2.113}$$

In order to illustrate the Bayes' rule in the PDF context, a simple example is now presented. Figure 2.49 shows the case: two random variables A and B. The variable A has a beta PDF; the variable B a normal PDF. There is a region where both PDFs overlap.

The two products of interest for the Bayes' rule are shown in Fig. 2.50. In both cases, the product is plotted: it is a low hill at the bottom. Clearly, the result of the two products is the same. Notice that the conditional probability $f(A|B)$ is zero outside the domain of B, and similarly with $f(B|A)$ and A.

It is also clear that the result of the products considered in Fig. 2.50 can also be obtained by simple product of the two PDFs, as it is shown in Fig. 2.51. The joint PDF, $f(A, B)$ is also equal to this product.

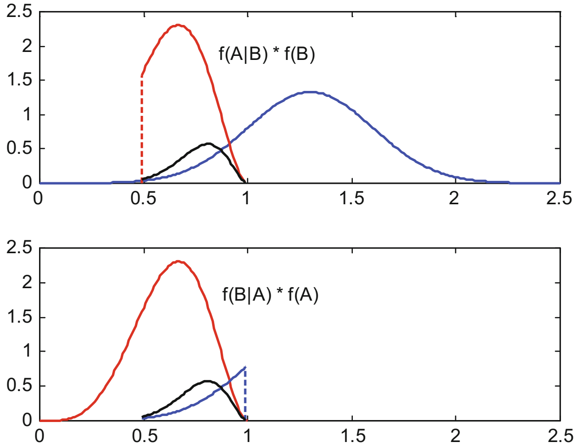


Fig. 2.50 The two products of interest

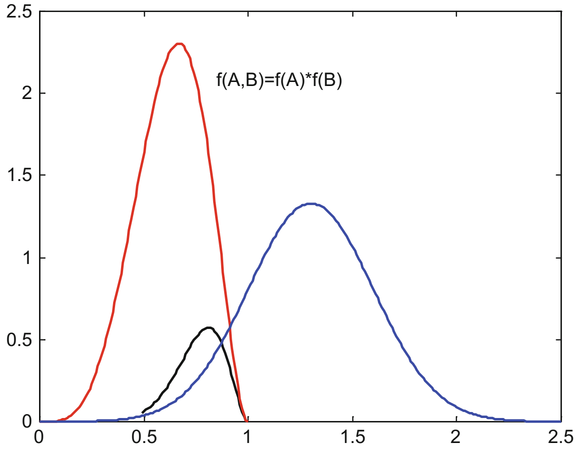


Fig. 2.51 The product of the two PDFs

Program 2.41 Two overlapped PDFs

```
% Two overlapped PDFs
x=0:0.01:2.5;
%densities
alpha=5; beta=3;
Apdf=betapdf(x,alpha, beta); %beta PDF
mu=1.3; sigma=0.3;
Bpdf=normpdf(x,mu,sigma); %normal pdf
%product of PDFs at intersection zone
piz=Apdf(50:100).*Bpdf(50:100);
%display
figure(1)
plot(x,Apdf,'r'); hold on;
plot(x,Bpdf,'b');
title('Two random variables A and B: their PDFs')
```

```
xlabel('x'); ylabel('y');
figure(2)
subplot(2,1,1)
plot(x,Bpdf,'b'); hold on;
plot(x(50:100),Apdf(50:100),'r')
plot([x(50) x(50)],[0 Apdf(50)],'r--');
plot(x(50:100),piz,'k');
title('f(A|B) * f(B)');
subplot(2,1,2)
plot(x,Apdf,'r'); hold on;
plot(x(50:100),Bpdf(50:100),'b')
plot([x(100) x(100)],[0 Bpdf(100)],'b--');
plot(x(50:100),piz,'k');
title('f(B|A) * f(A)');
figure(3)
join=Apdf.*Bpdf;
plot(x(50:100),join(50:100),'k'); hold on;
plot(x,Apdf,'r'); hold on;
plot(x,Bpdf,'b');
title('f(A,B)=f(A)*f(B)')
```

More details on the Bayes' rule can be found in [64]. Some examples of applications are given in [11]. A more extensive exposition on Bayesian probability topics is [16].

2.9.3 Bayesian Networks. Graphical Models

One convenient way for the study of probabilistic situations is offered by the Bayesian networks, [8, 21, 89]. These are graphical models that represent probabilistic relationships among a set of variables.

A very simple model is shown in Fig. 2.52. It refers to two random variables A and B. We would say that A is a parent of B, B is a child of A.

According with Fig. 2.52 one has:

$$P(A \cap B) \;=\; P(A)\, P(B|A) \tag{2.114}$$

Both the Bayes network of Fig. 2.52 and Eq. (2.35) can represent the four cases (a, b, c, d) of the devices example. For instance, the case a) is:

Fig. 2.52 A simple Bayes network

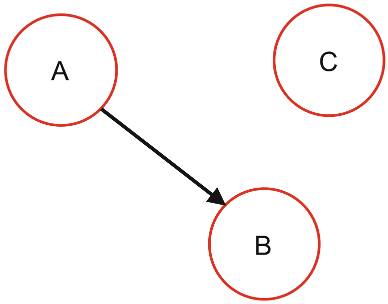


Fig. 2.53 Another example of Bayes network

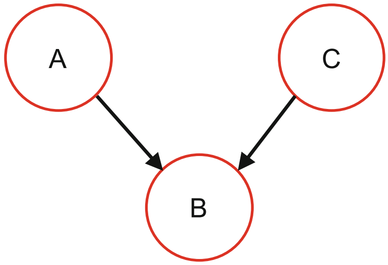


Fig. 2.54 Two parents in a Bayes network

$$P(A = good \cap B = accept) = P(A = good)\, P(B = accept|A = good) \quad (2.115)$$

In some cases, the variables A and B can take only two values: true or false (good or bad, etc.). In other cases, the variables can take any value.

Figure 2.53 represents another situation. In this figure, C is independent of A and B; while B depends on A.

Concerning Fig. 2.53, one has:

$$P(A \cap B \cap C) = P(A)\, P(B|A)\, P(C) \quad (2.116)$$

A typical example of the situation depicted in Fig. 2.54 is that A and C are the results of flips of two coins (two possible values: T or H). B is true if the values of A and C coincide.

Now:

$$P(A \cap B \cap C) = P(B|A \cap C)\, P(A)\, P(C) \quad (2.117)$$

Suppose that, in the example of the coins, we know that B is true (we got *evidence* on this), then:

$$P((A = H \cap C = H) \mid B = true) = \frac{1}{2} \quad (2.118)$$

On the other hand:

$$P(A = H \mid B = true)\, P(C = H \mid B = true) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4} \quad (2.119)$$

Therefore, once we know B is true, A and C are not independent (Eqs. (2.38) and (2.39) give different results).

The literature related with Bayesian networks (BN) is becoming particularly extensive. In general, BN are used for the study of scenarios with several alternatives, like medical diagnosis, planning and decision making, etc. The description of a situation in terms of a BN would be quite useful for forward or backward inference; and also for illustrating the complexity and the internal structure of the problem at hand.

Most cited books are [59] on BN and decision graphs, [43] on BN and Bayesian Artificial Intelligence, or [58] on learning BN. This last subject, learning, has deep interest for a number of reasons, being one of them the possibility of automatic construction of BN by learning mechanisms, instead of direct human work.

With respect to learning BN, [34] offers a tutorial, [26] presents some BN learning approaches, and [54] treats in academic detail learning from data.

A related topic is '*belief networks*'. Representative references are [22] for classifier systems, and the tutorial of [45].

There are many published applications, like some papers connecting BN and GIS. The acronym GIS means Geographic Information System, which, for instance, could be related with the prediction of flooding or avalanches, etc. Examples of this kind of applications are [86] on BN and GIS based decision systems, and [87] on BN, GIS and planning in marine pollution scenarios.

Other illustrative applications are, [17] for meteorology, [5, 51, 60] for medical diagnosis and prediction, [99] for risk analysis and maintenance, [68] for natural resources management, and [32] for financial analysis. See the book [70] for more types of applications.

2.10 Markov Process

A *stochastic process* (or random process) with state space S, is a collection of indexed random variables. Usually the index is time. The state space could be discrete or continuous. Likewise, the index (time) could be discrete or continuous. Hence, there are four general types of stochastic processes. There are many books on stochastic processes, like [63, 67]. In addition, there are also brief academic introductions, like [14, 44].

Consider any state S_i of the stochastic process. The next state could be any of the states belonging to S. There are state transition probabilities. The Markov property is that these probabilities only depend on the present state, and not on past states.

A Markov chain is a discrete-state random process with the Markov property. The chain could be discrete-time or continuous time.

The first part of this section focuses on Markov chains, [42]. The section then continues with the generation of random data using Markov chain Monte Carlo (MCMC).

2.10.1 Markov Chain

In a Markov chain the transition probability from S_n to S_{n+1} is:

$$P(S_{n+1} \mid S_n,\ S_{n-1}, \ldots, S_{n-m}) = P(S_{n+1} \mid S_n) \tag{2.120}$$

which is the Markov property.

The transition probabilities could be written as a table. For instance, in a process with three states A, B and C:

			after	
		A	B	C
	A	0.65	0.20	0.15
before	B	0.3	0.24	0.46
	C	0.52	0.12	0.36

Notice that row sums are equal to 1.

Also, the transition probabilities could be written in matrix form:

$$T = \begin{bmatrix} 0.65 & 0.20 & 0.15 \\ 0.30 & 0.24 & 0.46 \\ 0.52 & 0.12 & 0.36 \end{bmatrix} \tag{2.121}$$

In writing these numbers we are supposing the Markov chain is time-homogeneous, that is: the probabilities keep constant along time.

A graphical expression of the Markov chain could be done as a stochastic finite state machine (FSM). For instance, continuing with the example (Fig. 2.55):

The process starts with an initial probability vector:

$\vec{X}_0 = [x_1(0),\ x_2(0), , x_k(0)]$

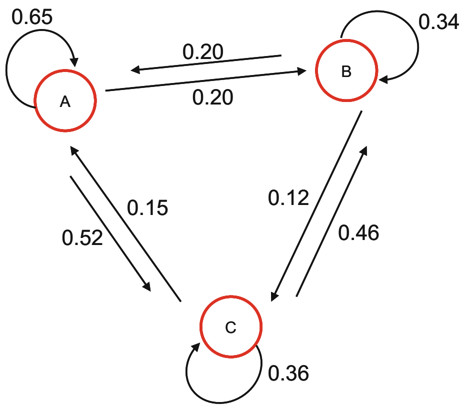


Fig. 2.55 An example of Markov chain FSM

For instance, it could be $[0.25, 0.40, 0.35]$ for the three states example we are considering (therefore, the initial probabilities are: $P(A) = 0.25, P(B) = 0.40, P(C) = 0.35$).

The probability vector after one transition is:

$$\vec{X}_1 \;=\; \vec{X}_0\, T \tag{2.122}$$

And, after n transitions:

$$\vec{X}_1 \;=\; \vec{X}_0\, T^{\,n} \tag{2.123}$$

There are many applications of this framework. Like for instance the study of certain system evolutions: market preferences, voting, population components (the case of several types of trees in a forest), transportation options, etc. We recommend [94], as it describes the five greatest applications of Markov Chains, including Shannon's information theory, web searching (Google), computer performance evaluation, etc. Another interesting text is [35], with a connection between Markov chains and game theory.

A transition matrix is *regular* if for some k, all the entries $t_{ij} \in T^k$ are positive; that is: $0 < t_{ij} < 1$. For example:

$$T \;=\; \begin{bmatrix} 0.35 & 0.65 & 0 \\ 0 & 0.30 & 0.7 \\ 0.75 & 0.25 & 0 \end{bmatrix} \tag{2.124}$$

$$T^2 \;=\; \begin{bmatrix} 0.1225 & 0.4225 & 0.4550 \\ 0.5250 & 0.2650 & 0.2100 \\ 0.2625 & 0.5625 & 0.1750 \end{bmatrix} \tag{2.125}$$

For $k = 2$, all entries are positive; the matrix is regular.

If the matrix T is regular, then for any initial probability vector $\vec{z}$, it happens that as the number of transitions increases, the probability vector tends to a unique vector $\vec{V}$:

$$\vec{z}\, T^n \;=\; \vec{V} \tag{2.126}$$

This vector $\vec{V}$ is denoted as the *equilibrium vector*, or the *fixed vector*, or the *steady state* vector. This last name refers to the fact that:

$$\vec{V}\, T \;=\; \vec{V} \tag{2.127}$$

Therefore, the studies on system evolutions may well end with a constant, equilibrium population.

It is important to study the eigenvalues of the transition matrix. If the transition matrix is regular, the largest eigenvalue is 1, and the rest of eigenvalues are $|\lambda_1| < 1$. Let us express a 3×3 transition matrix in diagonal form:

$$T = \vec{v} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \vec{v}^{-1} \tag{2.128}$$

Then, as n increases:

$$T^n = \vec{v} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \lambda_2^n & 0 \\ 0 & 0 & \lambda_3^n \end{bmatrix} \vec{v}^{-1} \rightarrow \vec{v} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \vec{v}^{-1} \tag{2.129}$$

Therefore, the matrix T^n converges to a constant matrix we shall denote as T_e.

Notice that Eq. (2.127) gives the left eigenvector corresponding to an eigenvalue equal to 1. This eigenvector is $\vec{V}$. All the rows of matrix T_e are equal to $\vec{V}$. For instance, if $\vec{V} = [0.13,\ 0.42,\ 0.45]$, then:

$$T_e = \begin{bmatrix} 0.13 & 0.42 & 0.45 \\ 0.13 & 0.42 & 0.45 \\ 0.13 & 0.42 & 0.45 \end{bmatrix} \tag{2.130}$$

Another type of Markov chain is the *absorbing Markov chain*. One or more of the diagonal entries t_{ii} of T is equal to 1, so the transition matrix is not regular. The states corresponding to such entries are *absorbing states*. Once the process enters in an absorbing state, it is not possible to leave.

Program B.1, which has been included in the Appendix for long programs, considers a simple weather prediction model with three states: Clouds ('C'), Rain ('R'), or Sun ('S'). We take the same values depicted in Fig. 2.55. The program departs from a vector of initial probabilities, and depicts in Fig. 2.56 the transitions between states.

The Program B.1 also prints the series of consecutive states as a string of characters, like: CCSRCRR…

2.10.2 Markov Chain Monte Carlo (MCMC)

Let us consider again the generation of random data with a desired PDF, $p(x)$. The MCMC methods do use a Markov chain that converges to a stationary distribution with the desired $p(x)$. Therefore, once the chain has converged, the chain is used to get draws from $p(x)$, although they would be correlated.

This convergence occurs regardless of the starting point. Usually, one throws out a certain number of initial draws. This is known as the *'burn-in'* of the algorithm

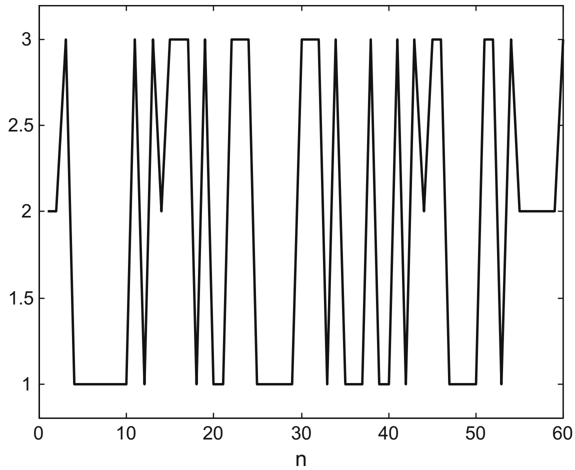


Fig. 2.56 Example of Markov Chain result

The research has already provided a number of methods for driving the chain to the desired PDF. See for instance the handbook [15]. The key contribution to start all this activity was the Metropolis algorithm, which is recognized as one of the ten most influential algorithms proposed in the 20th century [7]. According with [28], one could tell of a MCMC revolution. A brief history of this revolution is reported in [77], and with more extension in [75].

As background literature on MCMC, a brief introduction is [95], while a more extended introduction is [4]. Details of the rationale behind MCMC can be found in [23]. More extended texts are [9, 84, 93]. In addition, the academic literature from [76, 90] includes MATLAB programs.

2.10.2.1 Metropolis Algorithm

The goal is to draw samples from some $p(x)$ PDF, where $p(x) = f(x) / K$, and K is not known.

According with the algorithm introduced by Nicholas Metropolis in 1953, a proposal distribution (also called jumping distribution) $q(y|x)$ is chosen. This distribution corresponds to the transition probabilities of a Markov chain.

The generation of samples starts from an initial value x_0,

(a) Draw a sample from $q(y|x) \rightarrow x_1$
(b) Compute:

$$\alpha = \frac{p(x_1)}{p(x_0)} = \frac{f(x_1)}{f(x_0)} \tag{2.131}$$

(the constant K cancels out)

(c) If $\alpha > 1$ accept x_1 as new sample;
else, with probability α accept x_1,
else reject it and take $x_1 = x_0$ as new sample
(d) Back to (a) until sufficient number of samples has been obtained.

Unlike rejection sampling, when a sample is rejected we do not try again until one is accepted, we just let $x_1 = x_0$ and continue with the next time step.

The Metropolis algorithm uses a symmetric proposal distribution:

$$q(y|x) = q(x|y) \tag{2.132}$$

2.10.2.2 Metropolis–Hastings Algorithm

Hastings generalized in 1970 the Metropolis algorithm, taking an arbitrary $q(y|x)$, possibly non-symmetric, and using the following acceptance probability:

$$\alpha = \min\left(\frac{f(x_1)\, q(x_1|x_0)}{f(x_0)\, q(x_0|x_1)},\ 1\right) \tag{2.133}$$

(the Metropolis algorithm takes: $\alpha = \min(\frac{f(x_1)}{f(x_0)}\ ,\ 1)$)

See [23], and references therein, for different implementation strategies for the Metropolis–Hastings algorithm.

2.10.2.3 Example

Let us consider for example a desired PDF with a half-sine shape. We choose a Gaussian PDF for the proposal distribution. Figure 2.57 depicts the scenario with the desired (D) PDF and the proposal (P) PDF.

Program 2.42 provides an implementation of the Metropolis algorithm for the example just described. Two figures are generated. The first is Fig. 2.57 with the desired and the proposal PDFs. The second is Fig. 2.58 that shows the histogram of draws obtained by the Metropolis algorithm, which agrees with the desired PDF.

Program 2.42 Generation of random data with a desired PDF

```
% Generation of random data with a desired PDF
% Using MCMC
% Metropolis algorithm
% example of desired PDF
x=0:(pi/100):pi;
dpf=0.5*sin(x); % desired PDF
% example of proposal PDF (normal)
xp=-4:(pi/100):4;
sigma=0.6; %deviation
q=normpdf(xp,0,sigma);
```

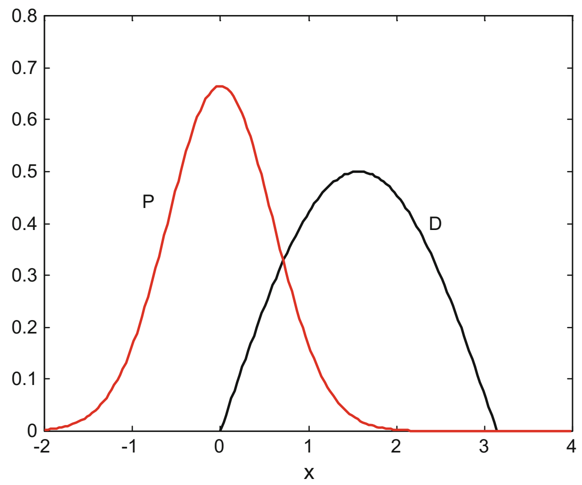


Fig. 2.57 The case considered in the Metropolis example

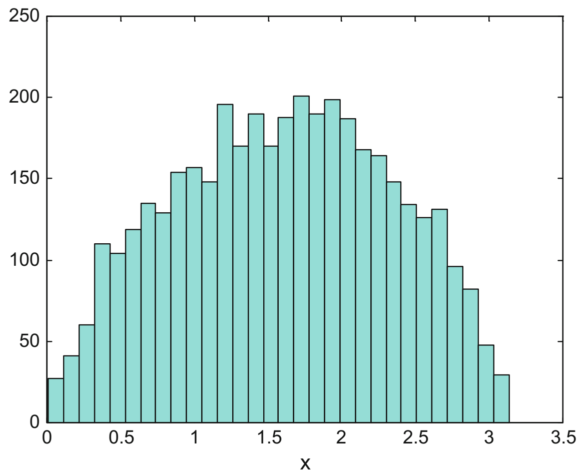


Fig. 2.58 Histogram of random data generated by the Metropolis algorithm

```
% generation of random data
N=5000; %number of data
z=zeros(1,N); %space for data to be generated
x0=pi/2; %initial value
for nn=1:N,
  inr=0;
  while inr==0, %new value proposal (Markovian transition)
    x1=x0+(sigma*randn(1)); %normal distribution (symmetric)
    if (x1<pi) & (x1>0),
      inr=1; %x1 is valid (is inside dpf domain)
    end;
  end;
  f1=0.5*sin(x1);
```

```
  f0=0.5*sin(x0);
  alpha=f1/f0;
  if alpha>=1
    z(nn)=x1; %accept
  else
  aux=rand(1);
  if aux<alpha,
    z(nn)=x1; %accept
  else
  z(nn)=x0;
  end;
end;
x0=x1;
end;
nz=z(1000:5000); %eliminate initial data
figure(1)
plot(x,dpf,'k'); hold on;
plot(xp,q,'r');
axis([-2 4, 0 0.8]);
xlabel('x'); title('desired PDF and proposal PDF');
figure(2)
hist(nz,30); colrmap('cool');
xlabel('x');title('histogram of the generated data');
```

The MATLAB Statistics Toolbox includes the *mhsample()* and *slicesample()* functions for using MCMC.

2.10.3 Hidden Markov Chain (HMM)

Consider the case of a Markov chain where states emit certain observable variables. Figure 2.59 shows a simplistic model of speech, the Markov chain has two states: (1) wovel and (2) consonant. When the process is in state 1, it could emit one of the three consonants W, H, or T, or a spacing _. When the process is in state 2, it could emit one of the four wovels A, E, O, U. The emission of any observable is made with an assigned probability.

Hence, in this example, there is a *'hidden Markov chain'* (HMM) with two states, and eight observables. The fundamental reference on HMM is [72]. Brief introductions are given in [29, 85].

This kind of process is being useful for the study of languages, genetics (bioinformatics), and other important fields, [24, 31, 48, 100].

Program B.2 implements the HMM process depicted in the previous diagram (Fig. 2.59). The results of one experiment running this program is shown in Fig. 2.60, with two subplots. The subplot on top corresponds to the hidden Markov chain transitions. The other subplot goes through four possible states (1, 2, 3, 4), according with the obervables emitted during the experiment.

The Program B.2 has been included in the Appendix for long programs. The program also prints the 'synthetic speech' generated by the HMM. It is possible, from public domain, to obtain data on transition probabilities of real human languages.

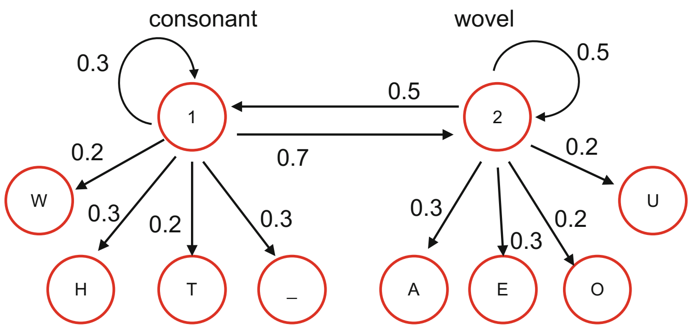


Fig. 2.59 An example of HMM (speech generator)

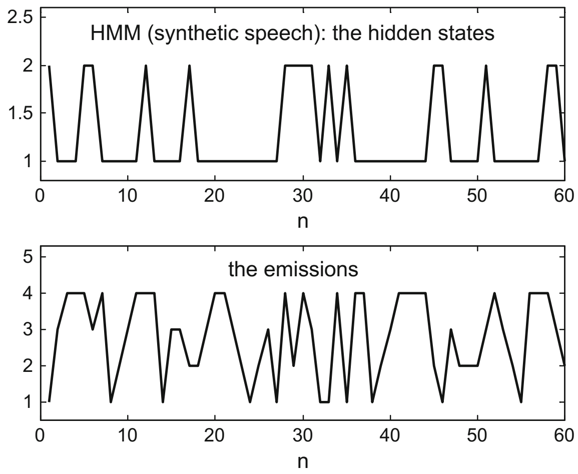


Fig. 2.60 Results of HMM example

In order to give a more complete idea of HMM, Fig. 2.61 shows another diagram. It is about the habits of someone, as the day is sunny or there is rain. Notice that for instance, this person like to walk under the Sun, and also (not so much) to walk in the rain.

The description of an HMM can be made with the transition matrix of the hidden Markov chain and with a matrix of observation probabilities. Continuing with the example, this last matrix corresponds to the following table:

			Observations		
		(1)Swim	(2)Walk	(3)Shop	(4)TV
States	1	0.2	0.5	0.3	0
	2	0	0.2	0.4	0.4

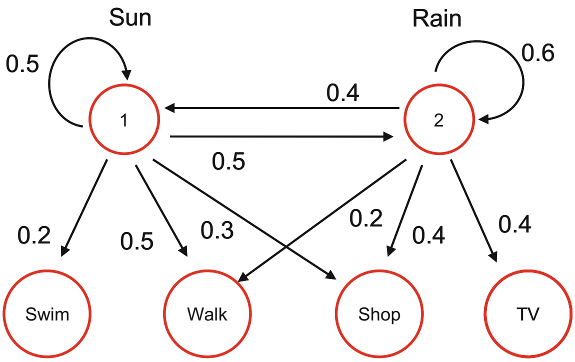


Fig. 2.61 Another HMM example

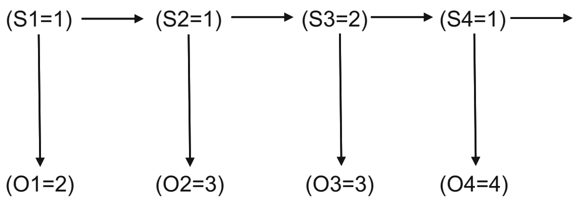


Fig. 2.62 An experiment with the HMM example

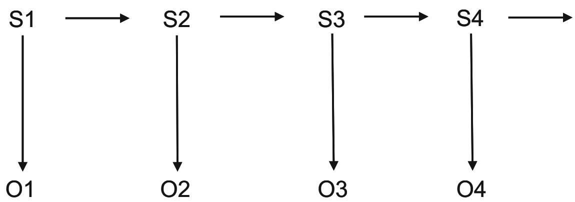


Fig. 2.63 A generic HMM path

Figure 2.62 shows an example of experiment running the HMM. The process advances through the Markov chain states, and it is observed by a sequence of emissions.

For example, suppose you are studying a series of archaeological strata, from a series of observables you might be interested in guessing if there were climate changes along certain epochs.

Figure 2.63 depicts a more abstract diagram showing the general behavior of the HMM along time. When you give values to states and observables, you describe a particular path of the process.

An interesting set of HMM application examples is given in [74]. Other published applications are, [88] on video background modeling, [20] on folk music classification, and [25] on classification of continuous heart sound signals.

2.11 MATLAB Tools for Distributions

The MATLAB Statistics Toolbox provides an interactive graph of PDF for many probability distributions. In response to the MATLAB prompt, the user writes:

disttool

And the screen shown in Fig. 2.64 will appear.

The user may select one of the many types of distributions included in the tool, and choose the visualization of the CDF or the PDF. Distribution parameters can be changed in order to observe their effects.

Another tool of interest is:

randtool

In response to this, the screen shown in Fig. 2.65 will appear.

The *randtool* will obtain samples of the PDF selected by the user, and visualize the corresponding histogram.

In other order of things, it is convenient to mention the interest of the MATLAB function *boxplot( )* for the display of statistical box plots. See the web site (https://plot.ly/matlab/box-plots/) for interesting examples.

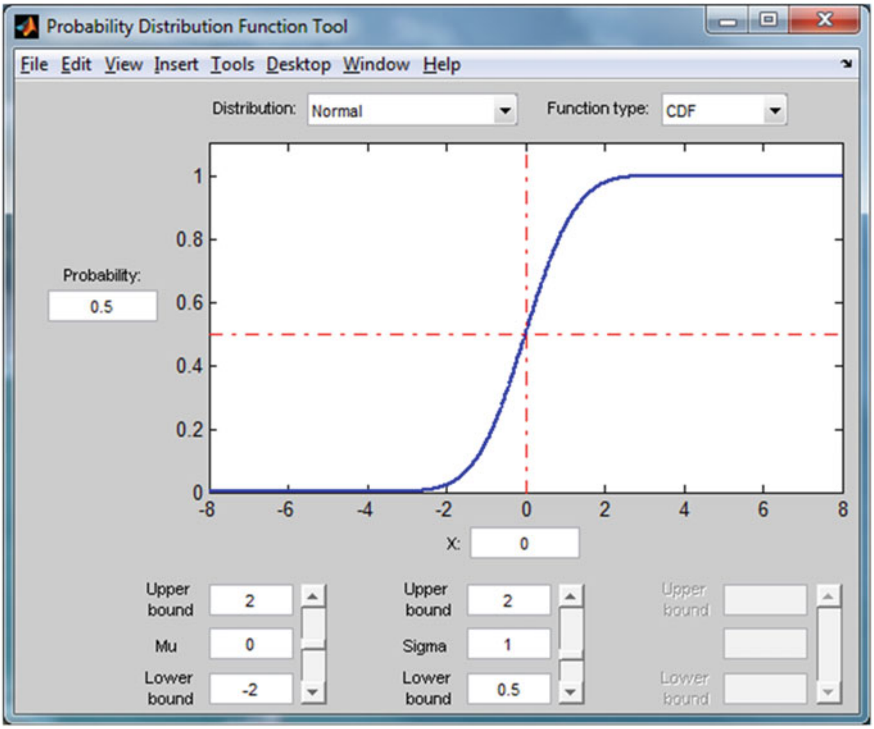


Fig. 2.64 Initial disttool screen

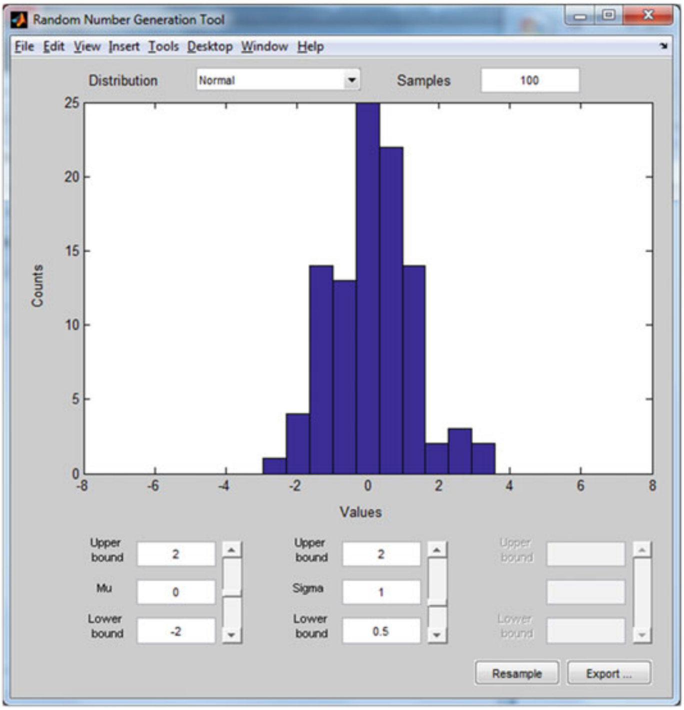


Fig. 2.65 Initial randtool screen

Figure 2.66 shows an example of box plot. The figure has been generated with the Program 2.43, which also contains the data being visualized.

Each box is used to indicate the position of the upper and lower quartiles. There is a crossbar inside the box that indicates the median. The extrema of the distribution are indicated with dashed lines and markers. See [69] for more details.

Program 2.43 Example of box plots

```
%Example of box plots
data=[1 5 8 3;
3 2 1 5;
5 4 8 1;
9 12 1 3;
14 0 2 2;
7 9 1 3];
median(data) %median of each column
mean(data) %mean of each column
std (data) % standard deviation of each column
figure(1)
boxplot(data)
title('box plot example')
```

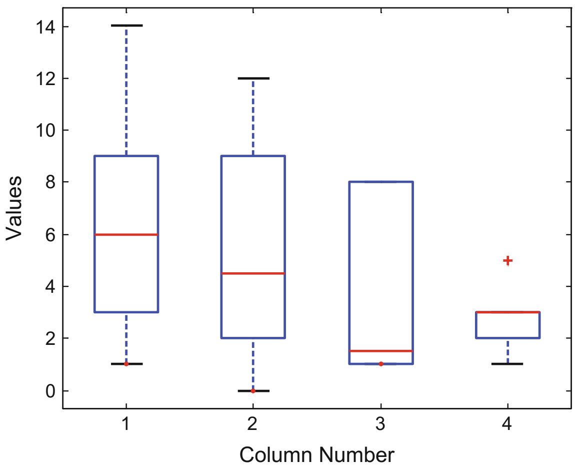


Fig. 2.66 Example of box plot

2.12 Resources

2.12.1 MATLAB

2.12.1.1 Toolboxes

- Exploratory Data Analysis Toolbox (EDA):
  http://cda.psych.uiuc.edu/martinez/edatoolbox/Docs/Contents.htm
- Bayes Net Toolbox:
  https://code.google.com/p/bnt/
- Bayes Net Toolbox for Student Modeling:
  http://www.cs.cmu.edu/~listen/BNT-SM/
- Markov Decision Processes (MDP) Toolbox:
  http://www7.inra.fr/mia/T/MDPtoolbox/
- MCMC Toolbox for Matlab:
  http://helios.fmi.fi/~lainema/mcmc/
- MCMC Methods for MLP and GP and Stuff (Aalto Univ.):
  http://becs.aalto.fi/en/research/bayes/mcmcstuff/
- Hidden Markov Model (HMM) Matlab Toolbox:
  http://nuweb.neu.edu/bbarbiellini/CBIO3580/HW7.html
- Mendel HMM Toolbox for Matlab:
  http://www.math.uit.no/bi/hmm/
- Stochastic Processes Toolkit for Risk Magement:
  http://www.damianobrigo.it/toolboxweb.pdf
- CompEcon Toolbox for Matlab (economics and finance):
  http://www4.ncsu.edu/~pfackler/compecon/toolbox.html

2.12.1.2 Links to Toolboxes

- Van Horn (Bayesian statistical inference):
  http://ksvanhorn.com/bayes/free-bayes-software.html
- Graphical Models (Bayesian):
  http://fuzzy.cs.uni-magdeburg.de/books/gm/tools.html
- Tools (Bayesian matters):
  http://www.cs.iit.edu/~mbilgic/classes/fall10/cs595/tools.html

2.12.1.3 Matlab Code

- Educational MATLAB GUIs (demos):
  http://users.ece.gatech.edu/mcclella/matlabGUIs/
- Advanced Box Plot for Matlab (Alex Bikfalvi):
  http://alex.bikfalvi.com/research/advanced_matlab_boxplot/
- Bayesian Statistics:
  http://www2.isye.gatech.edu/~brani/isyebayes/programs.html
- Matlab listings for Markov chains (Renato Feres):
  http://www.math.wustl.edu/~feres/Math450Lect04.pdf
- Matlab Code: Tutorial 1: Creating a Bayesian Network:
  https://dslpitt.org/genie/wiki/Matlab_Code:_Tutorial_1:_Creating_a_Bayesian_Network
- CGBayesNets (Gaussian Bayesian networks):
  http://www.cgbayesnets.com/
- Monte Carlo Methods (G. Gordon):
  http://www.cs.cmu.edu/~ggordon/MCMC/
- MCMC (Kevin Murphy):
  http://www.cs.ubc.ca/~murphyk/Teaching/CS340-Fall06/reading/mcmc.pdf
- Handbook of Monte Carlo Methods (D.P. Kroese et al.):
  http://www.maths.uq.edu.au/~kroese/montecarlohandbook/

2.12.2 Web Sites

- Stanford Encyclopedia of Philosophy (Bayes' Theorem):
  http://plato.stanford.edu/entries/bayes-theorem/
- STAT 504 PennState Univ.:
  https://onlinecourses.science.psu.edu/stat504
- scikit-learn:
  http://scikit-learn.org/dev/index.html
- Graphics_Examples (sample data for graphics demonstrations):
  https://people.sc.fsu.edu/~jburkardt/m_src/graphics_examples/graphics_examples.html

- Probvis; K. Potter (visualization of distribution functions.): https://people.sc.fsu.edu/~jburkardt/
- John Burkardt (Matlab codes, examples, etc.): https://people.sc.fsu.edu/~jburkardt/
- Bayes Nets: http://www.bayesnets.com/
- The Gaussian Processes Web Site: http://www.gaussianprocess.org/
- Belief Networks: https://www.cis.upenn.edu/~ungar/KDD/belief-nets.html

References

1. M.A. Al-Fawzan, *Methods for Estimating the Parameters of TheWeibull Distribution* (King Abdulaziz City for Science and Technology, Saudi Arabia, 2000). http://interstat.statjournals.net/YEAR/2000/articles/0010001.pdf
2. S.J. Almalki, S. Nadarajah, Modifications of the Weibull distribution: A review. Reliab. Eng. Syst. Saf. **124**, 32–55 (2014)
3. E. Anderson, *Monte Carlo Methods and Importance Sampling*. Lecture Notes, UC Berkeley (1999). http://ib.berkeley.edu/labs/slatkin/eriq/classes/guest_lect/mc_lecture_notes.pdf
4. C. Andrieu, N. De Freitas, A. Doucet, M.I. Jordan, An introduction to MCMC for machine learning. Mach. Learn. **50**(1–2), 5–43 (2003)
5. G. Arroyo-Figueroa, L.E. Suear, A temporal Bayesian network for diagnosis and prediction, in *Proceedings 15th Conference Uncertainty in Artificial Intelligence* (Morgan Kaufmann Publishers Inc, 1999), pp. 13–20
6. A. Assenza, M. Valle, M. Verleysen, A comparative study of various probability density estimation methods for data analysis. Int. J. Comput. Intell. Syst. **1**(2), 188–201 (2008)
7. I. Beichl, F. Sullivan, The Metropolis algorithm. Comput. Sci. Eng. **2**(1), 65–69 (2000)
8. I. Ben-Gal, Bayesian networks, in *Encyclopedia of Statistics in Quality & Reliability*, ed. by F. Faltin, R. Kenett, F. Ruggeri (Wiley, Chichester, 2007)
9. M. Bergomi, C. Pedrazzoli, *Bayesian Statistics: Computational Aspects*. Lecture Notes, ETH Zurich (2008). http://www.rw.ethz.ch/dmath/research/groups/sfs-old/teaching/lectures/FS_/seminar/8.pdf
10. J.M. Bernardo, A.F.M. Smith, *Bayesian Theory* (Wiley, New York, 2000)
11. G. Bohling, *Applications of Bayes' Theorem*. Lecture Notes, Kansas Geological Surveys (2005). http://people.ku.edu/~gbohling/cpe940/BayesOverheads.pdf
12. S. Borak, W. Härdle, R. Weron, Stable distributions, in *Statistical Tools for Finance and Insurance* (2005), pp. 21–44
13. G.E. Box, M.E. Muller, A note on the generation of random normal deviates. Ann. Math. Stat. **29**, 610–611 (1958)
14. L. Breuer. *Introduction to Stochastic Processes*. Lecture Notes, Univ. Kent (2014). https://www.kent.ac.uk/smsas/personal/lb209/files/notes1.pdf
15. S. Brooks, A. Gelman, G.L. Jones, X.-L. Meng, *Handbook of Markov Chain Mote Carlo* (Chapman and Hall/CRC, Boca Raton, 2011)
16. H. Bruyninckx, *Bayesian Probability*. Lecture Notes, KU Leuven, Belgium (2002). http://www.stats.org.uk/bayesian/Bruyninckx.pdf
17. R. Cano, C. Sordo, J.M. Gutiérrez, Applications of Bayesian networks in meteorology, in *Advances in Bayesian Networks*, ed. by J.A. Gamez, et al. (Springer, Berlin, 2004), pp. 309–328

18. A.N. Celik, A statistical analysis of wind power density based on the Weibull and Rayleigh models at the southern region of Turkey. Renew. Energy **29**(4), 593–604 (2004)
19. V. Cevher, *Importance Sampling*. Lecture Notes, Rice University (2008). http://www.ece.rice.edu/~vc3/elec633/ImportanceSampling.pdf
20. W. Chai, B. Vercoe, Folk music classification using hidden Markov models, in *Proceedings of International Conference on Artificial Intelligence*, vol. 6, no. 4 (2001)
21. E. Charniak, Bayesian networks without tears. AI Mag. **12**(4), 50–63 (1991)
22. J. Cheng, R. Greiner, Learning bayesian belief network classifiers: algorithms and system, in *Advances in Artificial Intelligence*, ed. by M. Stroulia (Springer, Berlin, 2001), pp. 141–151
23. S. Chib, E. Greenberg, Understanding the Metropolis-Hastings algorithm. Am. Stat. **49**(4), 327–335 (1995)
24. K.H. Choo, J.C. Tong, L. Zhang, Recent applications of hidden Markov models in computational biology. Genomics Proteomics Bioinf. **2**(2), 84–96 (2004)
25. Y.J. Chung, Classification of continuous heart sound signals using the ergodic hidden Markov model. Pattern Recogn. Image Anal. 563–570 (2007)
26. R. Daly, Q. Shen, S. Aitken, Learning Bayesian networks: approaches and issues. Knowl. Eng. Rev. **26**(2), 99–157 (2011)
27. L. Devroye, Sample-based non-uniform random variate generation, in *Proceedings ACM 18th Winter Conference on Simulation* (1986), pp. 260–265
28. P. Diaconis, The Markov chain Monte Carlo revolution. Bull. Am. Math. Soc. **46**(2), 179–205 (2009)
29. S.R. Eddy, What is a hidden Markov model? Nat. Biotechnol. **22**(10), 1315–1316 (2004)
30. S.E. Fienberg, When did Bayesian inference become "Bayesian"? Bayesian Anal. **1**(1), 1–40 (2006)
31. M. Gales, S. Young, The application of hidden Markov models in speech recognition. Found. Trends Sig. Process. **1**(3), 195–304 (2008)
32. J. Gemela, Financial analysis using Bayesian networks. Appl. Stoch. Models Bus. Ind. **17**(1), 57–67 (2001)
33. M. Haugh, *Generating Random Variables and Stochastic Processes*. Lecture Notes, Columbia Univ (2010). http://www.columbia.edu/~mh2078/MCS_Generate_RVars.pdf
34. D. Heckerman, *A Tutorial on Learning with Bayesian Networks* (Springer, Netherlands, 1998)
35. C.C. Heckman, *Matrix Applications: Markov Chains and Game Theory*. Lecture Notes, Arizona State Univ (2015). https://math.la.asu.edu/~checkman/MatrixApps.pdf
36. S. Holmes, *Maximum Likelihood Estimation*. Lecture Notes, Stanford Univ. (2001). http://statweb.stanford.edu/~susan/courses/s200/lect11.pdf
37. S. Holmes, *The Methods of Moments*. Lecture Notes, Stanford Univ. (2001). http://statweb.stanford.edu/~susan/courses/s200/lect8.pdf
38. R.J. Hoppenstein, *Statistical Reliability Analysis on Rayleigh Probability Distributions* (2000). www.rfdesign.com
39. S. Intajag, S. Chitwong, Speckle noise estimation with generalized gamma distribution, in *Proceedings of IEEE International Joint Conference SICE-ICASE* (2006), pp. 1164–1167
40. D. Joyce, *Common Probability Distributions*. Lecture Notes, Clark University (2006)
41. K. Knight. *Central Limit Theorems*. Lecture Notes, Univ. Toronto (2010). www.utstat.toronto.edu/keith/eco2402/clt.pdf
42. T. Konstantopoulos, *Markov Chains and Random Walks*. Lecture Notes (2009). http://159.226.43.108/~wangchao/maa/mcrw.pdf
43. K.B. Korb, A.E. Nicholson, *Bayesian Artificial Intelligence* (CRC Press, Boca Raton, 2010)
44. M. Kozdron, *The Definition of a Stochastic Process*. Lecture Notes, Univ. Regina (2006). http://stat.math.uregina.ca/~kozdron/Teaching/Regina/862Winter06/Handouts/revised_lecture1.pdf
45. M.L. Krieg, A tutorial on Bayesian belief networks. Technical Report DSTO-TN-0403 (2001). http://dspace.dsto.defence.gov.au/dspace/handle/1947/3537
46. D.P. Kroese, *A Short Introduction to Probability*. Lecture Notes, Univ. Queensland (2009). http://www.maths.uq.edu.au/~kroese/asitp.pdf

47. D.P. Kroese, T. Brereton, T. Taimre, Z.I. Botev, Why the Monte Carlo method is so important today. Wiley Interdisciplinary Reviews: Computational Statistics **6**(6), 386–392 (2014)
48. A. Krogh, M. Brown, I.S. Mian, K. Sjölander, D. Haussler, Hidden Markov models in computational biology: applications to protein modeling. J. Mol. Biol. **235**(5), 1501–1531 (1994)
49. E. Limpert, W.A. Stahel, M. Abbt, Log-normal distributions across the sciences: keys and clues. Bioscience **51**(5), 341–352 (2001)
50. R. Linna, *Monte Carlo Methods I*. Lecture Notes, Aalto University (2012). http://www.lce.hut.fi/teaching/S-114.1100/lect_9.pdf
51. P. Lucas, *Bayesian Networks in Medicine: A Model-based Approach to Medical Decision Making*. Lecture Notes, Univ. Aberdeen (2001). http://cs.ru.nl/~peterl/eunite.pdf
52. D.J. MacKay, Introduction to Monte Carlo methods, in *Learning in Graphical Models*, ed. by M.I. Jordan (Springer, Berlin, 1998), pp. 175–204
53. V. Manian, *Image Processing: Image Restoration*. Lecture Notes, Univ. Puerto Rico (2009). www.ece.uprm.edu/~manian/chapter5IP.pdf
54. D. Margaritis, Learning Bayesian Network Model Structure from Data. Ph.D. thesis, US Army (2003)
55. W.L. Martinez, A.R. Martines, *Computational Statistics Handbook with MATLAB* (Chapman & Hall/CRC, Boca Raton, 2007)
56. G.M. Masters, *Renewable and Efficient Electric Power Systems* (Wiley, New York, 2013)
57. I.J. Myung, Tutorial on maximum likelihood estimation. J. Math. Psychol. **47**(1), 90–100 (2003)
58. R.E. Neapolitan, *Learning Bayesian Networks*, vol. 38 (Prentice Hall, Upper Saddle River, 2004)
59. T.D. Nielsen, F.V. Jensen, *Bayesian Networks and Decision Graphs* (Springer Science & Business Media, New York, 2009)
60. D. Nikovski, Constructing bayesian networks for medical diagnosis from incomplete and partially correct statistics. IEEE T. Knowl. Data Eng. **12**(4), 509–516 (2000)
61. J.P. Nolan, *Stable Distributions*, Chap1. Lecture Notes, American University (2014). http://academic2.american.edu/~jpnolan/stable/chap1.pdf
62. F.N. Nwobi, C.A. Ugomma, A comparison of methods for the estimation of Weibull distribution parameters. Adv. Method. Stat./Metodoloski zvezki **11**(1), 65–78 (2014)
63. P. Olofsson, M. Andersson, *Probability, Statistics, and Stochastic Processes* (Wiley, Chichester, 2012)
64. B.A. Olshausen, *Bayesian Probability Theory*. Lecture Notes, UC. Berkeley (2004). http://redwood.berkeley.edu/bruno/npb163/bayes.pdf
65. A.B. Owen, *Monte Carlo Theory, Methods and Examples*. Lecture Notes, Stanford Univ. (2013). Book in progress. http://statweb.stanford.edu/~owen/mc/
66. C. Pacati, *General Sampling Methods*. Lecture Notes, Univ. Siena (2014). http://www.econ-pol.unisi.it/fineng/gensampl_doc.pdf
67. M. Pinsky, S. Karlin, *An Introduction to Stochastic Modeling* (Academic Press, Cambridge, 2010)
68. C.A. Pollino, C. Henderson, Bayesian Networks: A Guide for Their Application in Natural Resource Management and Policy. Technical Report 14, Landscape Logicpp. (2010)
69. K. Potter, *Methods for Presenting Statistical Information: The Box Plot*. Lecture Notes, Univ. Utah (2006). http://www.kristipotter.com/publications/potter-MPSI.pdf
70. O. Pourret, P. Naïm, B. Marcot (eds.), *Bayesian Networks: A Practical Guide to Applications*, vol. 73 (Wiley, Chichester, 2008)
71. R forge distributions Core Team. A guide on probability distributions. Technical report (2009). http://dutangc.free.fr/pub/prob/probdistr-main.pdf
72. L. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
73. R. Ramamoorthi, *Monte Carlo Integration*. Lecture Notes, U.C. Berkeley (2009). https://inst.eecs.berkeley.edu/~cs/fa09/lectures/scribe-lecture4.pdf

74. N. Ramanathan, *Applications of Hidden Markov Models*. Lecture Notes, University of Maryland (2006). http://www.cs.umd.edu/~djacobs/CMSC828/ApplicationsHMMs.pdf
75. M. Richey, The evolution of Markov chain Monte Carlo methods. Am. Math. Mon. **117**(5), 383–413 (2010)
76. B.D. Ripley, *Computer-Intensive Statistics*. Lecture Notes, Oxford Univ (2008). http://www.stats.ox.ac.uk/~ripley/APTS2012/APTS-CIS-lects.pdf
77. C. Robert, G. Casella, A short history of Markov chain Monte Carlo: Subjective recollections from incomplete data. Stat. Sci. **26**(1), 102–115 (2011)
78. S. Stahl, The evolution of the normal distribution. Math. Mag. **79**(2), 96–113 (2006)
79. F.W. Scholz, *Central Limit Theorems and Proofs*. Lecture Notes, Univ. Washington (2011). http://www.stat.washington.edu/fritz/DATAFILES394_/CLT.pdf
80. R. Seydel, *Tools for Computational Finance* (Springer, Berlin, 2012)
81. S.J. Sheather, Density estimation. Stat. Sci. **19**(4), 588–597 (2004)
82. K. Sigman, *Acceptance-Rejection Method*. Lecture Notes, Columbia University (2007). www.columbia.edu/~ks20/4703-Sigman-Notes-ARM.pdf
83. B.W. Silverman, *Density Estimation for Statistics and Data Analysis* (2002). https://ned.ipac.caltech.edu/level5/March02/Silverman/paper.pdf
84. M. Sköld, *Computer Intensive Statistical Methods*. Lecture Notes, Lund University (2006)
85. M. Stamp, *A Revealing Introduction to Hidden Markov Models*. Lecture Notes, San Jose State University (2012). http://gcat.davidson.edu/mediawiki-1.19.1/images/2/23/HiddenMarkovModels.pdf
86. A. Stassopoulou, M. Petrou, J. Kittler, Application of a Bayesian network in a GIS based decision making system. Int. J. Geogr. Inf. Sci. **12**(1), 23–46 (1998)
87. V. Stelzenmüller, J. Lee, E. Garnacho, S.I. Rogers, Assessment of a Bayesian belief network-GIS framework as a practical tool to support marine planning. Mar. Pollut. Bull. **60**(10), 1743–1754 (2010)
88. B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, J.M. Buhmann, Topology free hidden Markov models: Application to background modeling, in *Proceedings of IEEE International Conference on Computer Vision*, vol. 1 (2001), pp. 294–301
89. T.A. Stephenson, An introduction to bayesian network theory and usage. Technical report, IDIAP Research, Switzerland (2000). http://publications.idiap.ch/downloads/reports/2000/rr00-03df
90. M. Steyvers, *Computational Statistics with MATLAB*. Lecture Notes, UC Irvine (2011). http://www.cidlab.com/205c/205C_v4.pdf
91. D.B. Thomas, W. Luk, P.H. Leong, J.D. Villasenor, Gaussian random number generators. ACM Comput. Surv. (CSUR) **39**(4), 11 (2007)
92. C. Tomasi, *Estimating Gaussian Mixture Models with EM*. Lecture Notes, Duke University (2004). https://www.cs.duke.edu/courses/spring04/cps196.1/handouts/EM/tomasiEM.pdf
93. B. Vidakovic, *MCMC Methodology*. Lecture Notes, Georgia Tech (2014). http://www2.isye.gatech.edu/~brani/isyebayes/bank/handout10.pdf
94. P. Von Hilgers, A.N. Langville, The five greatest applications of Markov chains, in *Proceedings of the Markov Anniversary Meeting* (Boston Press, Boston, 2006)
95. R. Waagepetersen, *A quick introduction to Markov chains and Markov chain Monte Carlo (revised version)*. Aalborg Univ. (2007). http://people.math.aau.dk/~rw/Papers/mcmc_intro.pdf
96. C. Walk, Hand-book of statistical distributions for experimentalists. Technical report, University of Stockholm (2007). Internal Report
97. G. Wang, Q. Dong, Z. Pan, X. Zhao, J. Yang, C. Liu, Active contour model for ultrasound images with Rayleigh distribution. Mathematical Problems in Engineering, ID 295320 (2014)
98. J.C. Watkins, *Maximum Likelihood Estimation*. Lecture Notes, University of Arizona (2011). http://math.arizona.edu/~jwatkins/o-mle.pdf
99. P. Weber, G. Medina-Oliva, C. Simon, B. Iung, Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas. Eng. Appl. Artif. Intell. **25**(4), 671–682 (2012)

100. B.J. Yoon, Hidden Markov models and their applications in biological sequence analysis. Curr. Genomics **10**(6), 402–415 (2009)
101. G.A. Young, *M2S1 Lecture Notes*. Lecture Notes, Imperial College London (2011). www2.imperial.ac.uk/~ayoung/m2s1/M2S1.PDF
102. A.Z. Zambom, R. Dias, *A Review of Kernel Density Estimation with Applications to Econometrics* (2012). arXiv preprint arXiv:1212.2812
103. L. Zhang, *Applied Statistics I*. Lecture Notes, University of Utah (2008). http://www.math.utah.edu/~lzhang/teaching/3070summer/DailyUpdates/jul1/lecture_jul1.pdf

Springer

http://www.springer.com/978-981-10-2533-4

Digital Signal Processing with Matlab Examples, Volume 1
Signals and Data, Filtering, Non-stationary Signals, Modulation
Giron-Sierra, J.M.
2017, XXXVII, 622 p. 467 illus., 186 illus. in color., Hardcover
ISBN: 978-981-10-2533-4