Monica Borda · Romulus Terebes ·
Raul Malutan · Ioana Ilea · Mihaela Cislariu ·
Andreia Miclea · Stefania Barburiceanu

Randomness and Elements of Decision Theory Applied to Signals

Springer

Randomness and Elements of Decision Theory Applied to Signals

Monica Borda · Romulus Terebes · Raul Malutan ·
Ioana Ilea · Mihaela Cislariu · Andreia Miclea ·
Stefania Barburiceanu

Randomness and Elements of Decision Theory Applied to Signals

Springer

Monica Borda
Technical University of Cluj Napoca
Cluj-Napoca, Romania

Romulus Terebes
Technical University of Cluj-Napoca
Cluj-Napoca, Romania

Raul Malutan
Technical University of Cluj-Napoca
Cluj-Napoca, Romania

Ioana Ilea
Technical University of Cluj-Napoca
Cluj-Napoca, Romania

Mihaela Cislariu
Technical University of Cluj-Napoca
Cluj-Napoca, Romania

Andreia Miclea
Technical University of Cluj-Napoca
Cluj-Napoca, Romania

Stefania Barburiceanu
Technical University of Cluj-Napoca
Cluj-Napoca, Romania

ISBN 978-3-030-90313-8 ISBN 978-3-030-90314-5 (eBook)
https://doi.org/10.1007/978-3-030-90314-5

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The work, organized in 15 chapters and 3 Appendices, presents the basics of random variables/processes and decision theory, all these massively illustrated with examples, mainly from telecommunications and image processing.

The addressability of the book is principally to students in electronics, communications and computer sciences branches, but we think it is very useful to Ph.D. students dealing with applications requiring randomness in many other areas.

The whole material has an educational presentation, starting with theoretical backgrounds, followed by simple examples and MATLAB implementation and ending with tasks given to the reader as homework. A short bibliography is available for each chapter.

The difficulty of the material is put in a progressive way, starting with simple examples, and continuing with applications taken from signal and image processing.

Chapter 1, *Random variables*, introduces random variables and the study of their characteristic functions and values, from both theoretical and practical (MATLAB simulations) points of view.

Chapter 2, *Probability distributions*, aims to study the probability distributions and the central limit theorem, both theoretically and practically using MATLAB.

Chapter 3, *Joint random variables*, describes joint random variables and their characteristic functions and values. These elements are theoretically introduced and a MATLAB implementation is available.

Chapter 4, Random processes, describes the random processes theoretically but also practically using MATLAB.

Chapter 5, Pseudo-noise sequences. The purpose of this work is the study of the properties and applications of the Pseudo-Noise Sequences (PNS). MATLAB implementation of a PNS generator is also proposed in the chapter.

Chapter 6, Markov systems, shows several problems and exercises to illustrate the theory of memory sources and Markov chains theory.

Chapter 7, Noise in telecommunication systems. The purpose of this chapter is the study of the noise that appears in communication systems.

Chapter 8, Decision systems in noisy transmission channels, aims the study of decision systems in noisy transmissions and the implementation of an application with a graphical user interface in the LabVIEW environment.

Chapter 9, Comparison of ICA algorithms for oligonucleotide microarray data, presents a study and comparison of Independent Component Analysis algorithms on oligonucleotide microarray data, with application in MATLAB.

Chapter 10, Image classification based on statistical modeling of textural information, introduces an application of statistics, for image processing. More precisely, a classification algorithm based on the statistical modeling of textural information is presented.

Chapter 11, Histogram equalization, aims to study the histogram of an image and the process of histogram equalization of an image.

Chapter 12, PCM and DPCM—applications in image processing, describes pulse code modulation (PCM) and differential pulse code modulation (differential PCM or DPCM) with applications in image processing.

Chapter 13, Nearest neighbor (NN) and k-nearest neighbor (kNN) supervised classification algorithms, presents the theoretical foundations associated with the supervised classification algorithms of NN (*Nearest Neighbor*) and kNN (*k-Nearest Neighbors*) type and exemplifies their use in data classification applications using MATLAB, operating on simplified representations of forms and patterns in terms of feature vectors having the most representative characteristics (descriptors).

Chapter 14, Texture feature extraction and classification using the Local Binary Patterns operator, shows the theoretical fundamentals regarding texture feature extraction using the Local Binary Patterns operator as well as its use in a MATLAB application that performs image classification.

Chapter 15, Supervised deep learning classification algorithms, aims to present the theoretical foundations associated with supervised classification networks of deep learning (CNNs), as well as to exemplify their use in data classification applications in MATLAB.

The examples were selected to be as simple as possible, but pointing out the essential aspects of the processing. Some of them are classical, others are taken from the literature, but the majority are original.

The understanding of the phenomenon, of the aim of processing, in its generality, not necessarily linked to a specific application, the development of the "technical good sense" is the logic thread guiding the whole work.

Cluj-Napoca, Romania

Monica Borda
Romulus Terebes
Raul Malutan
Ioana Ilea
Mihaela Cislariu
Andreia Miclea
Stefania Barburiceanu

Contents

1 Random Variables . . . 1
1.1 Definitions . . . 1
1.2 Classification . . . 2
1.2.1 Discrete Random Variables . . . 2
1.2.2 Continuous Random Variables . . . 2
1.3 Characteristic Functions . . . 2
1.3.1 Probability Distribution Function . . . 2
1.3.2 Probability Density Function . . . 3
1.4 Characteristic Values . . . 4
1.4.1 ***n***th Order Average Operator . . . 4
1.4.2 ***n***th Order Centered Moment . . . 5
1.5 Other Statistical Means . . . 6
1.6 Applications and Matlab Examples . . . 9
1.6.1 Computation of Characteristic Functions and Values . . . 9
1.6.2 Influence of Outlier Values on the Data's Centroid . . . 12
1.7 Tasks . . . 14
1.8 Conclusions . . . 15
References . . . 15

2 Probability Distributions . . . 17
2.1 Discrete Distributions . . . 17
2.1.1 Bernoulli Distribution . . . 17
2.1.2 Binomial Distribution . . . 18
2.1.3 Poisson Distribution . . . 18
2.1.4 Geometric Distribution . . . 19
2.2 Continuous Distributions . . . 20
2.2.1 Uniform Distribution . . . 20
2.2.2 Normal Distribution (Gaussian) . . . 20
2.2.3 Rayleigh Distribution . . . 21
2.3 The Central Limit Theorem . . . 22

2.4 Probability Distributions in Matlab 22
2.5 Applications and Matlab Examples 23
2.5.1 Probability Distribution Tool 23
2.5.2 Generating Bernoulli Experiments 23
2.5.3 The Connection Between Bernoulli Distribution and Binomial Distribution 25
2.5.4 The Connection Between Bernoulli Distribution and the Geometric Distribution 26
2.5.5 The Central Limit Theorem 28
2.5.6 Problem 30
2.6 Tasks 32
2.7 Conclusions 32
References 33

3 Joint Random Variables 35
3.1 Definition and Characteristics 35
3.2 Characteristic Functions 36
3.2.1 Joint Probability Distribution Function 36
3.2.2 Joint Probability Density Function 36
3.2.3 Marginal Density Functions 37
3.3 Characteristic Values 37
3.3.1 Joint Moments of Order ij 37
3.3.2 Joint Centered Moments of Order ij 38
3.3.3 Correlation Coefficient Between X and Y 39
3.3.4 Correlation Matrix 39
3.4 Sample Covariance and Correlation 40
3.4.1 Definition 40
3.4.2 Matlab Functions 40
3.5 Applications and Matlab Examples 41
3.5.1 Joint Random Variables 41
3.5.2 Sample Correlation 46
3.6 Tasks 48
3.7 Conclusions 48
References 48

4 Random Processes 49
4.1 Definition and Classification 49
4.2 Stationary Process 54
4.2.1 Statistical Mean for a Random Process 55
4.3 Ergodic Processes 56
4.4 Autocorrelation Function of a Sum of 2 Independent Processes 57
4.5 Applications and Matlab Examples 58
4.5.1 Generation of a Random Process 58
4.5.2 Statistical Means for a Random Process 60
4.5.3 Ergodic Processes 62

4.5.4 Autocorrelation Function of a Sum of 2 Independent Processes 62
4.6 Tasks 65
4.7 Conclusions 66
References 66

5 Pseudo-Noise Sequences 67
5.1 Theoretical Introduction 67
5.2 PNS Generator 67
5.3 Identification of the Pseudo-Noise Sequences 69
5.3.1 Autocorrelation Function 69
5.3.2 Mutual Correlation Function 71
5.4 Applications and MATLAB Examples 72
5.4.1 PNS Generator Using MATLAB Functions 72
5.4.2 Graphs of Autocorrelation and Mutual Correlation Functions 73
5.5 Tasks 76
5.6 Conclusions 77
References 77

6 Markov Systems 79
6.1 Solved Problems 79
6.2 Applications and MATLAB Examples 85
6.2.1 Functions to Create, Determine and Visualize the Markov Chain Structure 85
6.2.2 Create, Modify and Visualize the Markov Chain Model 85
6.3 Proposed Problems [2] 87
6.4 Conclusions 88
References 88

7 Noise in Telecommunication Systems 89
7.1 Generalities 89
7.1.1 Definition and Classification 89
7.1.2 Internal Noise 89
7.1.3 External Noise 90
7.2 Thermal Noise 90
7.3 Models for the Internal Noise 93
7.3.1 Effective Noise Temperature 93
7.3.2 Noise Figure 94
7.4 Noise in Cascaded Systems 95
7.5 Effective Noise Temperature of an Attenuator 96
7.6 Noise in PCM Systems 97
7.6.1 Quantization Noise 97
7.6.2 Decision Noise 98
7.6.3 The Total Signal-to-Noise Ratio in a PCM System 99

7.7 Solved Problems 99
7.8 Tasks 102
7.8.1 Problems to be Solved 102
7.8.2 Practical Part in Matlab 103
7.9 Conclusions 109
References 109

8 Decision Systems in Noisy Transmission Channels 111
8.1 Introduction 111
8.2 Bayes Criterion 114
8.3 Discrete Detection of a Unipolar Signal 117
8.4 Discrete Detection of Polar Signal 121
8.5 Continuous Detection of Known Signal 122
8.6 Transmission Systems with Binary Decision 127
8.7 Applications and LabView Examples 131
8.7.1 LabView Application for the Transmission System with Binary Decision 131
8.7.2 Tasks 133
8.7.3 Problems Proposed 134
8.8 Conclusions 135
References 135

9 Comparison of ICA Algorithms for Oligonucleotide Microarray Data 137
9.1 Introduction 137
9.2 Blind Source Separation 137
9.3 Theoretical Aspects of ICA 139
9.4 ICA Algorithms 140
9.4.1 FastICA Algorithm 140
9.4.2 JADE Algorithm 141
9.4.3 EGLD Algorithm 143
9.4.4 Gene Expression by MAS 5.0 Algorithm 144
9.5 Robust Estimation of Microarray Hybridization 145
9.6 Reasons for Inconsistent Test Pair Hybridization Patterns 147
9.7 Reliably and Unreliably Expressed Test Probes 149
9.8 Applications and Matlab Implementation 151
9.8.1 Collecting the Initial Data 151
9.8.2 Data After ICA Processing 153
9.9 Conclusions 153
References 155

10 Image Classification Based on Statistical Modeling of Textural Information 157
10.1 Texture in Image Processing 157
10.2 Texture Classification 157
10.2.1 Feature Extraction 158

10.2.2 Classification Based on the Kullback–Leibler Divergence 160
10.3 Matlab Implementation 160
10.3.1 Wavelet Decomposition 161
10.3.2 Covariance Matrix Estimation 161
10.3.3 Kullback–Leibler Divergence 163
10.4 Tasks 163
10.5 Conclusions 164
References 164

11 Histogram Equalization 165
11.1 Histogram 165
11.2 Histogram Equalization 165
11.3 Histogram of an Image and Histogram Equalization in Matlab 167
11.4 Tasks 170
11.5 Conclusions 170
References 171

12 PCM and DPCM–Applications in Image Processing 173
12.1 Pulse Code Modulation 173
12.2 Differential Pulse Code Modulation 175
12.3 Applications and Matlab Examples 178
12.3.1 Study of a PCM System 178
12.3.2 Study of a DPCM system 183
12.4 Conclusions 183
References 184

13 Nearest Neighbour (NN) and k-Nearest Neighbour (kNN) Supervised Classification Algorithms 185
13.1 Unsupervised and Supervised Classification of Data 185
13.2 NN and kNN Classification Algorithms 185
13.3 Support for NN and kNN Algorithms in Matlab 187
13.4 Tasks 189
13.5 Conclusions 190
References 190

14 Texture Feature Extraction and Classification Using the Local Binary Patterns Operator 191
14.1 Automatic Image Classification System 191
14.2 Feature Extraction 192
14.2.1 Generalities 192
14.2.2 The Initial Local Binary Patterns Operator 193
14.2.3 The Improved Version of the Local Binary Patterns 195
14.3 Image Classification 196
14.3.1 Introduction 196
14.3.2 The Support Vector Machine Classifier 196

14.4 Work Description 197
14.5 Tasks 203
14.6 Conclusions 204
References 204

15 Supervised Deep Learning Classification Algorithms 205
15.1 Convolutional Neural Network (CNN) 205
15.1.1 Introduction 205
15.1.2 Layers in a CNN Network 205
15.2 Applications and MATLAB Examples 208
15.2.1 Deep Learning MATLAB Toolbox 208
15.2.2 Define, Train and Test a CNN Architecture 209
15.3 Conclusions 215
References 215

Appendix A 217

Appendix B 219

Appendix C 225

List of Figures

Fig. 1.1 Plot of the probability mass function 11
Fig. 1.2 Plot of the probability distribution function 12
Fig. 1.3 Behavior of the arithmetic mean and median **a** for an outlier free dataset and **b** in the presence of outliers 13
Fig. 1.4 Behavior of the trimmed average for different percentages of discarded outliers **a** $= 0\%$, **b** $\alpha = 2\%$, **c** $\alpha = 5\%$ and **d** $\alpha = 10\%$ 14
Fig. 2.1 The graphical interface of the application "*Probability Distribution Tool*" 24
Fig. 2.2 The connection between Bernoulli distribution and binomial distribution 27
Fig. 2.3 The connection between the Bernoulli distribution and the geometric distribution 28
Fig. 2.4 Illustration of the central limit theorem for the sum of n random variables with uniform distribution in the range [0, 1] 30
Fig. 4.1 Example of processes 50
Fig. 4.2 Random process continuous in time with continuous values 51
Fig. 4.3 Random process continuous in time with discrete values—random sequences 52
Fig. 4.4 Random process continuous in time with discrete values—random binomial process 52
Fig. 4.5 Random process discrete in time with continuous values—Gaussian random process 53
Fig. 4.6 Random process discrete in time with discrete values—Bernoulli random process 53
Fig. 4.7 Quantization noise in PCM (from [5]) 54
Fig. 4.8 Random process continuous in time with discrete values 59
Fig. 4.9 Time discrete random process 60
Fig. 4.10 Statistical mean 61

Fig. 4.11 Representation of an ergodic process $X(t)$ and the autocorrelation function 65
Fig. 4.12 Autocorrelation function of two independent random processes 65
Fig. 5.1 Block scheme of a PNS generator with LFSR and XOR gates 68
Fig. 5.2 Block scheme of the PNS generator using the polynomial $g(x) = x^3 + x^2 + 1$ 69
Fig. 5.3 Graph of the autocorrelation function 70
Fig. 5.4 The graph of the autocorrelation function for a PNS of length 7 75
Fig. 5.5 The graph of the mutual correlation function for a PNS of length 7, when 1 error is present 75
Fig. 5.6 The graph of the mutual correlation function for a PNS of length 7, when 2 errors are present 76
Fig. 6.1 Graph of the Markov chain 87
Fig. 7.1 Thermal noise generation 90
Fig. 7.2 Computation of the available power of the thermal noise 91
Fig. 7.3 Power spectral density of the thermal noise in **a** one-sided and **b** two-sided representation 92
Fig. 7.4 Ideal and real transfer function 92
Fig. 7.5 Computation of the equivalent temperature 93
Fig. 7.6 The computation of the noise factor 94
Fig. 7.7 Cascade of amplifiers 95
Fig. 7.8 The model of an attenuator 96
Fig. 7.9 Cascade of two amplifiers 100
Fig. 7.10 The block scheme of a transmission system having three amplifiers 100
Fig. 7.11 The block scheme of a transmission system having two amplifiers 103
Fig. 7.12 The block scheme of the cascade of three amplifiers 103
Fig. 7.13 The block scheme of the cascade of four amplifiers 105
Fig. 7.14 The input sinusoidal signal 106
Fig. 7.15 The input sinusoidal signal, the quantized signal and the PCM encoded signal 108
Fig. 8.1 Block scheme of a transmission system using signal detection. S–source, N–noise generator, SD–signal detection, U–user, $s_i(t)$–transmitted signal, $r(t)$–received signal, $n(t)$–noise voltage, $\hat{s}_i(t)$–estimated signal [1] 112
Fig. 8.2 Binary decision splits observation space Δ into two disjoint spaces Δ_0 and Δ_1 [1] 112
Fig. 8.3 Block scheme of an optimal receiver (operating according to Bayes criterion, of minimum risk) [1] 114

Fig. 8.4 Binary detection parameters: P_m—probability of miss, P_D—probability of detection, P_f—probability of false detection [1] . . . 115
Fig. 8.5 Graphical representation of function Q(y) [1] . . . 116
Fig. 8.6 Block-scheme of the optimal receiver for unipolar signal and discrete observation [1] . . . 118
Fig. 8.7 Graphical representation of pdfs of decision variables for unipolar decision and discrete observation [1] . . . 120
Fig. 8.8 Block Scheme of an optimal receiver with continuous observation decision for one known signal s(t): (top) correlator based implementation; (bottom) matched filter implementation [1] . . . 125
Fig. 8.9 Graphical representation of $p(z/s_0)$ and $p(z/s_1)$ [1] . . . 126
Fig. 8.10 Structure of optimal receiver with continuous detection of two signals . . . 130
Fig. 8.11 Structure of optimal receiver with continuous detection of two signals in phase opposition . . . 130
Fig. 8.12 Front panel of the transmission system with binary decision . . . 131
Fig. 8.13 Controls section. **a** Noise standard deviation control; **b** Binary sequence control; **c** Sample and hold amplification control; **d** Threshold control . . . 132
Fig. 8.14 **a** Secondary panel; **b** LED arrays; **c** simulation time; **d** error counter . . . 133
Fig. 9.1 **a** The signals corresponding to four speakers recorded by the microphone; **b** The original sources signals after ICA was applied [2] . . . 138
Fig. 9.2 PM and MM sequences [6] . . . 146
Fig. 9.3 The geometrical relation between PM-MM vectors . . . 149
Fig. 9.4 The Affymetrix data flow [8] . . . 151
Fig. 9.5 Results for a very unreliable probe set. An γ value improvement, after EGLD computing, from 0.945 to 0.085 can be observed . . . 155
Fig. 10.1 VisTex texture database [1] . . . 158
Fig. 10.2 Texture analyzed at two different scales . . . 158
Fig. 10.3 Image wavelet transform with two levels and three orientations . . . 159
Fig. 10.4 Extraction of the spatial information . . . 162
Fig. 11.1 Image and image histogram . . . 166
Fig. 11.2 (top-left) The original image, (top-right) the image obtained after applying histogram equalization and (bottom) the table with each sample in the histogram and the value that becomes after applying histogram equalization . . . 169

Fig. 11.3 (top-left) The original image; (top-right) original image histogram, (bottom-left) the image after applying histogram equalization; (bottom-right) modified image histogram ... 171
Fig. 12.1 Pulse code modulation (PCM) block scheme ... 174
Fig. 12.2 Example of PCM generation ... 174
Fig. 12.3 Example of uniform quantization using PCM: **a** original image: **b** image quantized with 2 bits/pixel; **c** image quantized with 3 bits/pixel; **d** image quantized with 5 bits/pixel ... 175
Fig. 12.4 Illustration of DPCM principle ... 176
Fig. 12.5 DPCM decoder ... 177
Fig. 12.6 Example of a predictive spatial compression scheme: **a** the original image; **b** the prediction error image for a 6 bit/pixel representation using only the previous neighbour; **c** the decoded image corresponding to the error image in **b**; **d** the prediction error image for a 6 bit/pixel representation using the previous neighbour and 2 neighbours on the previously decoded top line; **e** the decoded image corresponding to the error image in **d**; **f** the prediction error image for a 1 bit/pixel representation using only the previous neighbour; **g** the decoded image corresponding to the error image in **f**; **h** the prediction error image for a 6 bit/pixel representation using the previous neighbour and 2 neighbours on the previous decoded top line; **i** the decoded image corresponding to the error image in **h** ... 179
Fig. 12.7 Illustrating the analogue signal encoding process using PCM ... 182
Fig. 12.8 Results. **a** the original image; **b** the image obtained after decoding; **c** the representation of the original signal and the decoded signal ... 184
Fig. 13.1 Prediction of a categorical label using the kNN algorithm (k = 3) ... 187
Fig. 13.2 Graphical visualization of vectors in the training set ... 188
Fig. 13.3 Visualisation of the created data structure for kNN classification in Matlab ... 188
Fig. 13.4 Result of the categorical prediction for the observation y = (2.5,−1) ... 189
Fig. 14.1 Schematic diagram of a supervised texture classification system ... 192
Fig. 14.2 Thresholding principle in LBP ... 193
Fig. 14.3 Conversion to decimal ... 194
Fig. 14.4 Storing the decimal number in the output matrix ... 194
Fig. 14.5 Example of circular neighborhood ... 195

Fig. 14.6 Example of maximum margin linear SVM binary classification 197
Fig. 14.7 Database folder structure 198
Fig. 14.8 Example of matrix that stores the train feature vectors 201
Fig. 14.9 The confusion matrix computed for the particular partition of the training and test sets 203
Fig. 15.1 An example of a neural network used to identify animals belonging to two classes, cats and dogs 206
Fig. 15.2 The architecture of the ConvNet with alternative convolutional filters, pooling operations and fully connected layers are applied on the original data given as input 206
Fig. 15.3 The concept of convolutional filters with the corresponding parameters 207
Fig. 15.4 The concept of pooling filters with the corresponding generated dimensions 208
Fig. C.1 Example of a graph in Matlab 235

Chapter 1
Random Variables

1.1 Definitions

Definition: *A signal, a phenomenon, or an experiment is random* if its development is, at least partially, ruled by probabilistic laws, meaning that the result (also called outcome, sample, or event) of such experiment is confirmed only by accomplishment.

Definition: *A random variable* (r.v.) X is a function defined on the sample space S (the set of all possible distinct outcomes, samples, or events $s_i \in S$) and taking real values:

$$X : S \to \mathbb{R}$$

$$X(s_i) = x_i$$

where: $x_i \in \mathbb{R}$ is the value of the random variable $X(s_i)$.

Remark

The random variables are denoted by upper case letters (e.g. X), while their values are denoted by the corresponding lower-case letters (e.g. x).

M. Borda et al., *Randomness and Elements of Decision Theory Applied to Signals*,
https://doi.org/10.1007/978-3-030-90314-5_1

1.2 Classification

1.2.1 Discrete Random Variables

Definition: *A random variable X is discrete* if the number of possible values is finite (for instance $x_i \in \{0, 1\}$), or countably infinite (such as $x_i \in \{0, 1, 2, \cdots\}$, $x_i \in \mathbb{Z}$) [1].

Example

Tossing a die, flipping a coin, or symbol generation by a binary source are some examples of events that can be modeled by discrete random variables [2].

1.2.2 Continuous Random Variables

Definition: *A random variable X is continuous* if it can take an infinite number of values from an interval of real numbers [1].

Example

The noise voltage in a resistor or the lifetime of electronic devices are some examples of phenomena modeled by continuous random variables.

1.3 Characteristic Functions

1.3.1 Probability Distribution Function

Abbreviation: PDF, Or cdf (*Cumulative Distribution Function*)

Definition: Let X be a random variable. The *probability distribution function* of X is denoted by $F_X(x)$ and it represents the function $F_X : \mathbb{R} \to [0, 1]$, defined as:

$$F_X(x) = P\{X \leq x\}, \forall x \in \mathbb{R}$$

Properties:

- $F_X(x) \geq 0, \forall x \in \mathbb{R}$;
- $F_X(-\infty) = 0$;
- $F_X(\infty) = 1$;
- $F_X(x)$ is not a decreasing function : $x_1 < x_2 \Rightarrow F_X(x_1) \leq F_X(x_2)$.

Remark

If X is a discrete random variable, then a probability p_i is associated to each sample s_i: $p_i = p(s_i) = P\{X(s_i) = x_i\} = p(x_i)$. In this case, the *probability mass function* (PMF) is defined as: $X : \begin{pmatrix} x_i \\ p_i \end{pmatrix}, \sum_i p_i = 1$

1.3.2 Probability Density Function

Abbreviation: pdf.

Definition: Let X be a random variable. The *probability density function* of X is denoted by $f_X(x)$ and it represents the derivative of the probability distribution function:

$$f_X(x) = \frac{dF_X(x)}{dx}$$

Properties:

- $f_X(x) \geq 0, \forall x \in \mathbb{R}$;
- $F_X(x) = \int\limits_{-\infty}^{x} f_X(t)dt$
- $\int\limits_{-\infty}^{\infty} f_X(x)dx = F_X(\infty) - F_X(-\infty) = 1$;
- $$\int\limits_{x_1}^{x_2} f_X(x)dx = \int\limits_{-\infty}^{x_2} f_X(x)dx - \int\limits_{-\infty}^{x_1} f_X(x)dx = F_X(x_2) - F_X(x_1) = \\ = P(x_1 < X \leq x_2)$$

Remark

The probability density function can be defined only if the probability distribution function is continuous, and it has a derivative

1.4 Characteristic Values

1.4.1 nth Order Average Operator

Definition: Let X be a random variable. The nth *order average operator* is denoted by $E\{X^n\}$ and its formula depends on the variable's type:

- if X is a discrete random variable:

$$E\left\{X^n\right\} = \sum_i x_i^n p(x_i)$$

- if X is a continuous random variable:

$$E\left\{X^n\right\} = \int_{-\infty}^{\infty} x^n f_X(x)dx$$

The most used average operators:

- If $n = 1$, the *expected value*, or the *expectation*, or the *mean* of the random variable is obtained, and it is denoted by $\mu_X = E\{X\}$.

Definition: The *expected value* represents the mean of all possible values taken by the random variable, weighted by their probabilities.

Properties:

- $E\{a\} = a$, if $a \in \mathbb{R}$ is a constant;
- $E\{ax\} = aE\{X\}$, if $a \in \mathbb{R}$ is a constant;
- $E\{ax + b\} = aE\{X\} + b$, if $a, b \in \mathbb{R}$ are constants;
- The operator can be applied also to functions depending on X. Let consider the function $g(X)$. In this case, the expectation is given by:

$$E\{g(X)\} = \begin{cases} \sum_i g(x_i)p(x_i), & \text{X discrete r.v.} \\ \int_{-\infty}^{\infty} g(x) f_X(x)dx, & \text{X continuous r.v.} \end{cases}$$

Remark

The mean can be interpreted as the DC component of a signal.

- If $n = 2$, the second order average operator is obtained, denoted by $E\{X^2\}$.

Remark

The significance of this operator is that of the power of a signal.

1.4.2 *nth Order Centered Moment*

Definition: Let X be a random variable. The nth *order centered moment* of the random variable is denoted by $E\{(X - E\{X\})^n\}$ and its formula depends on the variable's type:

- if X is a discrete random variable:

$$E\{(X - E\{X\})^n\} = \sum_i (x_i - E\{X\})^n p(x_i);$$

- if X is a continuous random variable:

$$E\{(X - E\{X\})^n\} = \int_{-\infty}^{\infty} (x - E\{X\})^n f_X(x)dx,$$

where $E\{X\}$ is the expectation.

Remark

- $X - E\{X\}$ represents the amplitude of the random variable's oscillation around the mean.
- $E\{X - E\{X\}\} = 0$.

The most used centered moment: If $n = 2$, the second order centered moment, also called *variance*, or *dispersion* is obtained, and it is denoted by $\sigma_X^2 = E\left\{(X - E\{X\})^2\right\}$.

Remark

- The significance of this operator is that of the variable's power of the oscillation around the mean.
- A small value of the dispersion implies that all the variable's values are close to their mean $E\{X\}$.
- The dispersion is not a linear operator.
- The *standard deviation* is defined as: $\sigma = \sqrt{\sigma_X^2}$

Properties:

- $\sigma_X^2\{a\} = 0$, if $a \in \mathbb{R}$ is a constant;
- $\sigma_X^2\{ax\} = a^2\sigma_X^2(x)$, if $a \in \mathbb{R}$ is a constant;
- $\sigma_X^2\{ax + b\} = a^2\sigma_X^2(x)$, if $a, b \in \mathbb{R}$ are constants.

1.5 Other Statistical Means

When studying samples, some other statistical means can be used, including the arithmetic mean, the geometric mean, the harmonic mean, the median, the trimmed average, or the Fréchet mean.

Let X be a random variable and $x = \{x_1, x_2, \ldots, x_N\}$ a sample of size N. In this case, the following measures can be defined:

(a) **Arithmetic mean:**

$$\overline{x}_{MA} = \frac{1}{N}\sum_{i=1}^{N} x_i.$$

Remark

- The arithmetic mean can be viewed as the sample's *center of mass* (or *centroid*), and it is obtained by minimizing the Euclidean distance between the elements $x_1, x_2, \ldots, x_N$.

(b) **Geometric mean:**

$$\overline{x}_{MG} = \sqrt[N]{\prod\nolimits_{i=1}^{N} x_i}.$$

Example

Let consider $x = \{1, 9, 7, 3, 5\}$. Knowing that $N = 5$, the geometric mean is: $\overline{x}_{MG} = \sqrt[5]{1 \times 9 \times 7 \times 3 \times 5} = 3.93$.

(c) **Harmonic mean:**

$$\overline{x}_{MH} = N\left(\sum_{i=1}^{N} \frac{1}{x_i}\right)^{-1}.$$

Example

Let consider $x = \{1, 9, 7, 3, 5\}$. Knowing that $N = 5$, the harmonic mean is:
$\overline{x}_{MH} = 5\left(\frac{1}{1} + \frac{1}{9} + \frac{1}{7} + \frac{1}{3} + \frac{1}{5}\right)^{-1} = 2.79$.

Remark

- Between the previously defined means (arithmetic, geometric and harmonic) the following relation can be established: $\overline{x}_{MA} \geq \overline{x}_{MG} \geq \overline{x}_{MH}$.

(d) **Median:** This value is obtained by sorting the elements $x_1, x_2, \ldots, x_N$ in ascending order and by taking the central value that divides the initial set into two subsets. Both the median and the arithmetic mean are centroid computation methods.

Compared to the arithmetic mean, the median is less influenced by the *outlier values* (values strongly different from the majority), giving a much better estimate of the central value.

Example

Let consider $x = \{1, 9, 7, 3, 5\}$. To compute the median, the vector is first sorted in ascending order, giving the set $\{1, 3, 5, 7, 9\}$. Next, the median is obtained by choosing the central value, which for this example is 5.

Remark

- If the sample x contains an even number of elements, then the median is computed as the arithmetic mean of the two central values.

(e) **Trimmed average:** This method is used when the data set $\{x_1, x_2, \ldots, x_N\}$ contains outlier values. In order to reduce their impact on the centroid estimation, the trimmed average can be employed. This approach deals with outliers by eliminating them from the dataset. Next, it computes the arithmetic mean or the median of the remaining elements.

(f) **Fréchet mean (or Karcher mean):** Represents a method to compute the center of mass, when the samples are defined on a surface.

Example

The covariance matrices can be modeled as realizations of a random variable defined on a surface. In this case, the center of mass can be computed by using the Fréchet mean.

Remark

- The arithmetic mean, the geometric mean and the harmonic mean can be viewed as Fréchet means for real-valued data.

1.6 Applications and Matlab Examples

1.6.1 Computation of Characteristic Functions and Values

Exercise: Let consider the experiment consisting in tossing a fair die. The result is observed, and the number of obtained dots is reported. Compute the probability mass function and the probability distribution function characterizing the random variable that models the experiment. What is the expectation of this variable? What is its dispersion?

Solution:

Let X be the discrete random variable modeling the experiment. The sample space S contains the six dice faces $S = \{f_1, \ldots, f_6\}$. The values taken by the variable X are given by the number of dots on each face: $x = \{1, \ldots, 6\}$. Supposing that a fair die is used, all the possible outcomes have the same probability: $p_i = \frac{1}{6}, \forall i = \overline{1, 6}$ [3].

Therefore, the probability mass function describing the random variable X is:

$$X : \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}$$

The probability distribution function must be computed for each interval:

$$\forall x \in (-\infty, 1) : F_X(x) = P\{X \le x\} = 0;$$

$$\forall x \in [1, 2) : F_X(x) = P\{X \le x\} = \sum\nolimits_{x_i \le x} p_i = \frac{1}{6};$$

$$\forall x \in [2, 3) : F_X(x) = P\{X \le x\} = \sum\nolimits_{x_i \le x} p_i = \frac{1}{6} + \frac{1}{6} = \frac{2}{6};$$

$$\forall x \in [3, 4) : F_X(x) = P\{X \le x\} = \sum\nolimits_{x_i \le x} p_i = \frac{1}{6} + \frac{1}{6} + \frac{1}{6} = \frac{3}{6};$$

$$\forall x \in [4, 5) : F_X(x) = P\{X \le x\} = \sum\nolimits_{x_i \le x} p_i = \frac{1}{6} + \frac{1}{6} + \frac{1}{6} + \frac{1}{6} = \frac{4}{6};$$