

List of Figures

1.1	A schematic representation of the classification of the data mining techniques discussed in this book.	5
1.2	The codes that can be used for representing a DNA sequence.	8
1.3	Three representations for protein molecules. From left to right: the full-atom representation of the whole protein, the representation of the atoms of the backbone only, and the representation through the torsion angles Φ and Ψ	10
1.4	The simulated annealing algorithm.	19
2.1	A possible transformation on aligned points: (a) the points are in their original locations; (b) the points are rotated so that the variability of their y component is zero.	25
2.2	A possible transformation on quasi-aligned points: (a) the points are in their original locations; (b) the points after the transformation.	26
2.3	A transformation on a set of points obtained by applying PCA. The circles indicate the original set of points.	29
2.4	Interpolation of 10 points by a join-the-dots function.	31
2.5	Interpolation of 10 points by the Newton polynomial.	33
2.6	Interpolation of 10 points by a cubic spline.	34
2.7	Linear regression of 10 points on a plane.	35
2.8	Quadratic regression of 10 points on a plane.	36
2.9	Average and standard deviations for all the parameters used for evaluating the chicken breast quality. Data from [156].	39
2.10	The PCA method applied in MATLAB [®] to a random set of points lying on the line $y = x$	41
2.11	The figure generated if the MATLAB instructions in Figure 2.10 are executed.	42
2.12	A sequence of instructions for drawing interpolating functions in MATLAB.	42

2.13	Two figures generated by MATLAB: (a) the instructions in Figure 2.12 are executed; (b) the instructions in Figure 2.14 are executed.	43
2.14	A sequence of instructions for drawing interpolating and regression functions in MATLAB.	44
3.1	A partition in clusters of a set of points. Points are marked by the same symbol if they belong to the same cluster. The two big circles represent the centers of the two clusters.	49
3.2	The Lloyd's or k -means algorithm.	50
3.3	Two possible partitions in clusters considered by the k -means algorithm. (a) The first partition is randomly generated; (b) the second partition is obtained after one iteration of the algorithm.	51
3.4	Two Voronoi diagrams in two easy cases: (a) the set contains only 2 points; (b) the set contains aligned points.	53
3.5	A simple procedure for drawing a Voronoi diagram.	53
3.6	The Voronoi diagram of a random set of points on a plane.	54
3.7	The k -means algorithm presented in terms of Voronoi diagram.	54
3.8	Two partitions of a set of points in 5 clusters and Voronoi diagrams of the centers of the clusters: (a) clusters and cells differ; (b) clusters and cells provide the same partition.	55
3.9	The h -means algorithm.	56
3.10	The h -means algorithm presented in terms of Voronoi diagram.	57
3.11	(a) A partition in 4 clusters in which one cluster is empty (and therefore there is no cell for representing it); (b) a new cluster is generated as the algorithm in Figure 3.12 describes.	59
3.12	The k -means+ algorithm.	60
3.13	The h -means+ algorithm.	60
3.14	A graphic representation of the compounds considered in datasets A , B , E and F . A and E are related to data measured within the three days that the fermentation started; B and F are related to data measured during the whole fermentation process.	69
3.15	Classification of wine fermentations by using the k -means algorithm with $k = 5$ and by grouping the clusters in 13 groups. In this analysis the dataset A is used.	71
3.16	The MATLAB function <code>generate</code>	74
3.17	Points generated by the MATLAB function <code>generate</code>	74
3.18	The MATLAB function <code>centers</code>	75
3.19	The center (marked by a circle) of the set of points generated by <code>generate</code> and computed by <code>centers</code>	76
3.20	The MATLAB function <code>kmeans</code>	77
3.21	The MATLAB function <code>plotp</code>	79
3.22	The partition in clusters obtained by the function <code>kmeans</code> and displayed by the function <code>plotp</code>	79

3.23	Different partitions in clusters obtained by the function <code>kmeans</code> . The set of points is generated with different <code>eps</code> values. (a) <code>eps</code> = 0.10, (b) <code>eps</code> = 0.05.	80
3.24	Different partitions in clusters obtained by the function <code>kmeans</code> . The set of points is generated with different <code>eps</code> values. (a) <code>eps</code> = 0.02, (b) <code>eps</code> = 0.	81
4.1	(a) The 1-NN decision rule: the point ? is assigned to the class on the left; (b) the k -NN decision rule, with $k = 4$: the point ? is assigned to the class on the left as well.	84
4.2	The k -NN algorithm.	84
4.3	An algorithm for finding a consistent subset T_{CNN} of T_{NN}	86
4.4	Examples of correct and incorrect classification.	86
4.5	An algorithm for finding a reduced subset T_{RNN} of T_{NN}	87
4.6	The study area of the application of k -NN presented in [97]. The image is taken from the quoted paper.	90
4.7	The 10 validation sites in Florida and Georgia used to develop the raw climate model forecasts using statistical correction methods.	92
4.8	The 10 target combinations of the outputs of FSU-GSM and FSU-RSM climate models.	92
4.9	Graphical representation of k -NN for finding the “best” match for a target soil. Image from [118].	95
4.10	The MATLAB function <code>knn</code>	97
4.11	The training set used with the function <code>knn</code>	98
4.12	The classification of unknown samples performed by the function <code>knn</code>	99
4.13	The MATLAB function <code>condense</code> : first part.	100
4.14	The MATLAB function <code>condense</code> : second part.	101
4.15	(a) The original training set; (b) the corresponding condensed subset T_{CNN0} obtained by the function <code>condense</code>	102
4.16	The classification of a random set of points performed by <code>knn</code> . The training set which is actually used is the one in Figure 4.15(b).	103
4.17	The MATLAB function <code>reduce</code>	104
4.18	(a) The reduced subset T_{RNN} obtained by the function <code>reduce</code> ; (b) the classification of points performed by <code>knn</code> using the reduced subset T_{RNN} obtained by the function <code>reduce</code>	105
5.1	Multilayer perceptron general scheme.	109
5.2	The face and the smile of Mona Lisa recognized by a neural network system. Image from [200].	115
5.3	A schematic representation of the test procedure for recording the sounds issued by pigs. Image from [45].	117
5.4	The time signal of a pig cough. Image from [45].	118
5.5	The confusion matrix for a 4-class multilayer perceptron trained for recognizing pig sounds.	119

5.6	X-ray and classic view of an apple. X-ray can be useful for detecting internal defects without slicing the fruit.	120
6.1	Apples with a short or long stem on a Cartesian system.	124
6.2	(a) Examples of linear classifiers for the apples; (b) the classifier obtained by applying a SVM.	124
6.3	An example in which samples cannot be classified by a linear classifier.	127
6.4	Example of a set of data which is not linearly classifiable in its original space. It becomes such in a two-dimensional space.	128
6.5	Chinese characters recognized by SVMs. Symbols from [63].	132
6.6	The hooked crow (lat. ab.: cornix) can be recognized by an SVM based on the sounds of birds.	133
6.7	The structure of the SVM decision tree used for recognizing bird species. Image from [71].	135
6.8	The MATLAB function <code>generate4libsvm</code>	138
6.9	The first rows of file <code>trainset.txt</code> generated by <code>generate4libsvm</code>	139
6.10	The DOS commands for training and testing an SVM by SVMLIB.	139
7.1	A microarray.	154
7.2	The partition found in biclusters separating the ALL samples and the AML samples.	156
7.3	Tissues from the HuGE Index set of data.	157
7.4	The partition found in biclusters of the tissues in the HuGE Index set of data.	158
8.1	The test set method for validating a linear regression model.	165
8.2	The test set method for validating a linear regression model. In this case, a validation set different from the one in Figure 8.1 is used.	166
8.3	The leave-one-out method for validation. (a) The point $(x(1), y(1))$ is left out; (b) the point $(x(4), y(4))$ is left out.	168
8.4	The leave-one-out method for validation. (a) The point $(x(7), y(7))$ is left out; (b) the point $(x(10), y(10))$ is left out.	169
8.5	A set of points partitioned in two classes.	171
8.6	The results obtained applying the k -fold method. (a) Half set is considered as a training set and the other half as a validation set; (b) training and validation sets are inverted.	172
9.1	A graphic scheme of the MIMD computers with distributed and shared memory.	174
9.2	A parallel algorithm for computing the minimum distance between one sample and a set of samples in parallel.	178
9.3	A parallel algorithm for computing the centers of clusters in parallel.	179
9.4	A parallel version of the h -means algorithm.	180
9.5	A parallel version of the k -NN algorithm.	180

9.6	A parallel version of the training phase of a neural network.	182
9.7	The tree scheme used in the parallel training of a SVM.	183
9.8	A parallel version of the training phase of a SVM.	183
10.1	A set of points before and after the application of the principal component analysis.	186
10.2	The line which is the solution of Exercise 4.	187
10.3	The solution of Exercise 7.	189
10.4	The solution of Exercise 8.	190
10.5	The solution of Exercise 9.	190
10.6	The set of points of Exercise 1 plotted with the MATLAB function <code>plotp</code> . Note that 3 of these points lie on the x or y axis of the Cartesian system.	198
10.7	The training set and the unknown point that represents a possible solution to Exercise 4.	202
10.8	A random set of 200 points partitioned in two clusters.	204
10.9	The condensed and reduced set obtained in Exercise 7: (a) the condensed set corresponding to the set in Figure 10.8; (b) the reduced set corresponding to the set in Figure 10.8.	205
10.10	The classification of a random set of points by using a training set of 200 points.	206
10.11	The classification of a random set of points by using (a) the condensed set of the set in Figure 10.8; (b) the reduced set of the set in Figure 10.8.	207
10.12	The structure of the network considered in Exercise 1.	208
10.13	The structure of the network considered in Exercise 3.	209
10.14	The structure of the network considered in Exercise 7.	211
10.15	The structure of the network required in Exercise 8.	212
10.16	The classes C^+ and C^- in Exercise 3.	213
A.1	Points drawn by the MATLAB function <code>plot</code>	225
A.2	The sine and cosine functions drawn with MATLAB.	227
A.3	The function <code>fun</code>	228
A.4	The graphic of the MATLAB function <code>fun</code>	229
B.1	The function <code>hmeans</code>	232
B.2	The prototypes of the functions called by <code>hmeans</code>	234
B.3	The function <code>rand_clust</code>	235
B.4	The function <code>compute_centers</code>	236
B.5	The function <code>find_closest</code>	237
B.6	The function <code>isStable</code>	237
B.7	The function <code>copy_centers</code>	238
B.8	An example of input text file.	239
B.9	The function <code>dimfile</code>	239
B.10	The function <code>readfile</code>	241

B.11	The function <code>main</code>	242
B.12	The function <code>main</code> of the application for generating random sets of data. Part 1.	246
B.13	The function <code>main</code> of the application for generating random sets of data. Part 2.	247
B.14	An example of input text file for the application <code>hmeans</code>	248
B.15	The output file provided by the application <code>hmeans</code> when the input is the file in Figure B.14 and $k = 2$	248
B.16	An output file containing a set of data generated by the application <code>generate</code>	249
B.17	The partition provided by the application <code>generate</code> (column A), the partition found by <code>hmeans</code> (column B) and the components of the samples (following columns) in an Excel spreadsheet.	250