Appendix

This appendix lists the functions (m-files) developed by the authors and used in the examples in this book. Functions used that are part of MATLAB's commercial distribution have been omitted; the reader is referred to the respective MATLAB manuals.

In the following list, functions are ordered alphabetically by chapter. For further function details, including descriptions of input and output arguments, refer to MATLAB's help utility. Also see the complete source code of the listed m-files, provided as part of the software on the companion website.

Chapter 1

bayes_classifier Bayesian classification rule for c classes, modeled by Gaussian distributions (also used in Chapter 2).

comp_gauss_dens_val Computes the value of a Gaussian distribution at a specific point (also used in Chapter 2).

compute_error Computes the error of a classifier based on a data set (also used in Chapter 4).

em_alg_function EM algorithm for estimating the parameters of a mixture of normal distributions, with diagonal covariance matrices.

EM_pdf_est EM estimation of the pdfs of c classes. It is assumed that the pdf of each class is a mixture of Gaussians and that the respective covariance matrices are diagonal.

euclidean_classifier Euclidean classifier for the case of c classes.

Gaussian_ML_estimate Maximum Likelihood parameters estimation of a multivariate Gaussian distribution.

generate_gauss_classes Generates a set of points that stem from c classes, given the corresponding a priori class probabilities and assuming that each class is modeled by a Gaussian distribution (also used in Chapter 2).

k_nn_classifier k-nearest neighbor classifier for c classes (also used in Chapter 4).

knn_density_estimate k-nn-based approximation of a pdf at a given point.

mahalanobis_classifier Mahalanobis classifier for c classes.

mixt_model Generates a set of data vectors that stem from a mixture of normal distributions (also used in Chapter 2).

mixt_value Computes the value of a pdf that is given as a mixture of normal distributions, at a given point.

mixture_Bayes Bayesian classification rule for c classes, whose pdf's are mixtures of normal distributions.

Parzen_gauss_kernel Parzen approximation of a pdf using a Gaussian kernel.

plot_data Plotting utility, capable of visualizing 2-dimensional data sets that consist of, at most, 7 classes.

Auxiliary functions gauss.

Chapter 2

base_clas_coord Implements a specific weak classifier.

***base_clas_coord*_out** Computes the output of the weak classifier implemented by the base_clas_coord function.

boost_clas_coord Generation of a "strong" classifier, using the Adaboost algorithm, that utilizes weak classifiers generated by the base_clas_coord function.

boost_clas_coord_out Computes the output of a "strong" classifier B as a weighted sum of the outputs of the weak classifiers.

CalcKernel Computes the value of a kernel function between two points.

kernel_perce Implements the kernel perceptron algorithm.

NN_evaluation Returns the classification error of a neural network based on a data set.

NN_training Returns a trained multilayer perceptron.

perce Realizes the perceptron learning rule, in a batch mode.

perce_online Realizes the online perceptron learning rule.

plot_kernel_perce_reg Plots the decision boundary that is generated by the kernel perceptron algorithm.

plot_NN_reg Plots the decision boundary that is formed by a neural network.

SMO2 Generates a SVM classifier using either Platt's algorithm or one of its two modifications proposed by Keerthi.

SSErr Generates the linear classifier that optimizes the sum of error squares criterion.

svcplot_book Support Vector Machine plotting utility. It plots the decision regions, the decision surfaces and the margin obtained by a SVM classifier.

Chapter 3

cut_cylinder_3D Generates a cut cylinder in the 3-dimensional space.

im_point Performs the projection of a vector on the subspace spanned by the first m principal components, that result after performing kernel PCA on a data set.

K_fun Computes the value of a kernel function (polynomial or exponential) for two vectors.

kernel_PCA Performs kernel PCA based on a given set of data vectors.

lapl_eig Performs Laplacian eigenmap based on a given data set.

pca_fun Performs Principal Component Analysis (PCA) based on a data set.

plot_orig_trans_kPCA Plots, in different figures, (a) the data points and the classifier in the original (2-dimensional) data space and (b) the projections of the data points and the classifier in the space spanned by the two most significant principal components, as they are computed using the kernel PCA method.

scatter_mat Computes the within scatter matrix, the between scatter matrix and the mixture scatter matrix for a *c*-class classification problem, based on a given data set.

spiral_3D Creates a 3-dimensional Archimedes spiral.

svd_fun Performs Singular Value Decomposition (SVD) of a matrix.

Chapter 4

compositeFeaturesRanking Scalar feature ranking that takes into account the cross-correlation coefficient.

divergence Computes the divergence between two classes.

divergenceBhata Computes the Bhattacharyya distance between two classes.

exhaustiveSearch Exhaustive search for the best feature combination, depending on the adopted class separability measure.

Fisher Computes Fisher's discriminant ratio of a scalar feature in a 2-class problem.

normalizeMnmx Performs MinMax normalization in a given interval [*l r*].

normalizeSoftmax Performs Softmax normalization in the interval [0 1].

normalizeStd Performs data normalization to zero mean and standard deviation equal to 1.

plotData Plotting utility for class data.

plotHist Plots the histograms of two classes for the same feature.

ROC Plots the ROC curve and computes the area under the curve.

ScalarFeatureSelection Ranking Features are treated individually and are ranked according to the adopted class separability criterion.

ScatterMatrices Class separability measure, which is computed using the within-class and mixture scatter matrices.

SequentialBackward Selection Feature vector selection by means of the Sequential Backward Selection technique.

SequentialForward FloatingSelection Feature vector selection by means of the Sequential Forward Floating Selection technique.

SequentialForward Selection Feature vector selection by means of the Sequential Forward Selection technique.

simpleOutlierRemoval Removes outliers from a normally distributed data set by means of the thresholding method.

Chapter 5

BackTracking Performs backtracking on a matrix of node predecessors and returns the best path. This function is also used in Chapter 6.

DTWItakura Computes the Dynamic Time Warping cost between two feature sequences, based on the standard Itakura local constraints.

DTWItakuraEndp Similar to *DTWItakura*, with the addition that endpoints constraints are allowed in the test sequence.

DTWSakoe Computes the Dynamic Time Warping cost between two feature sequences, based on the Sakoe-Chiba local constraints.

DTWSakoeEndp Similar to *DTWSakoe*, with the addition that endpoints constraints are allowed in the test sequence.

editDistance Computes the Edit (Levenstein) distance between two sequences of characters.

Auxiliary functions stEnergy, stZeroCrossingRate, IsoDigitRec.

Chapter 6

BWDoHMMsc Computes the recognition probability of an HMM, given a sequence of discrete observations, by means of the scaled version of the Baum-Welch (any-path) method.

BWDoHMMst Same as *BWDoHMMSc*, except that no scaling is employed.

MultSeqTrainDoHMMBWsc Baum-Welch training (scaled version) of a Discrete Observation HMM, given multiple training sequences.

MultSeqTrain DoHMMVITsc Viterbi training (scaled version) of a Discrete Observation HMM, given multiple training sequences.

MultSeqTrainCoHMMBWsc Baum-Welch training (scaled version) of a Continuous Observation HMM, given multiple training sequences.

VitCoHMMsc Computes the scaled Viterbi score of an HMM, given a sequence of l-dimensional vectors of continuous observations, under the assumption that the pdf of each state is a Gaussian mixture.

VitCoHMMst Same as *VitCoHMMsc* except that no scaling is employed.

VitDoHMMsc Computes the scaled Viterbi score of a Discrete Observation HMM, given a sequence of observations.

VitDoHMMst Same as *VitDoHMMsc*, except that no scaling is employed.

Chapter 7

agglom Generalized Agglomerative Scheme (GAS) for data clustering. It runs, on demand, either the single-link or the complete-link algorithm.

BSAS Basic Sequential Algorithmic Scheme (BSAS algorithm) for data clustering.

CL_step Performs a step of the complete-link algorithm.

dendrogram_cut Determines the clusterings of a hierarchy that best fit the underlying clustering structure of the data set at hand.

fuzzy_c_means FCM algorithm for data clustering.

GMDAS Generalized Mixture Decomposition Algorithmic Scheme (GMDAS algorithm) for data clustering.

k_means k-means clustering algorithm.

k_medoids k-medoids clustering algorithm.
LLA Competitive leaky learning algorithm for data clustering.
possibi Possibilistic clustering algorithm, adopting the squared Euclidean distance.
SL_step Performs a step of the single-link algorithm.
spectral_Ncut2 Spectral clustering based on the normalized cut criterion.
valley_seeking Valley-seeking algorithm for data clustering.
Auxiliary functions cost_comput, distan, distant_init, rand_data_init, rand_init, reassign.