Introduction to Environmental Data Science

Original Figures from the Book

(The printed book may only contain greyscale figures to reduce printing costs. The original figures, most of them in colour, are listed in this file).

William W. Hsieh

2022-07-18

Cambridge University Press

Contents

Chapter 1: Introduction

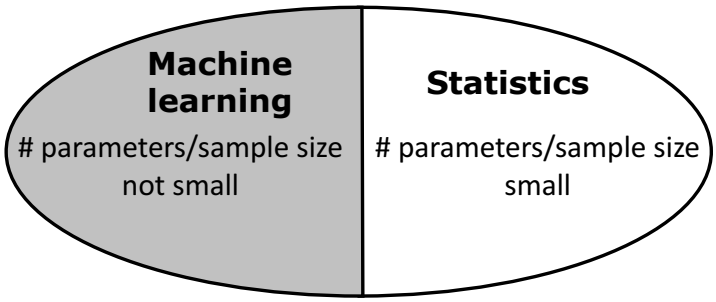


Figure 1.1 ML and statistics tend to occupy different parts of the data science space, as characterized by the number of model parameters to the sample size. In reality, there is overlap and more gradual transition between the two than the sharp boundary shown (see the Venn diagram in Fig. 15.1).

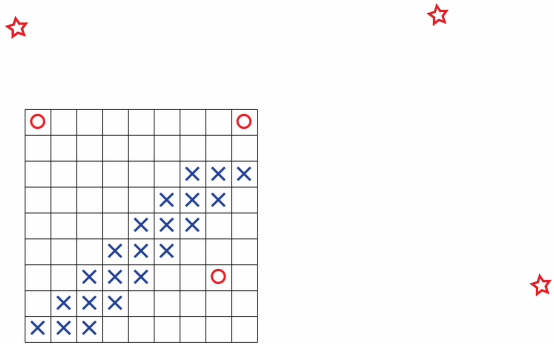


Figure 1.2 Schematic diagram illustrating the problem of outliers in the input data in 2-D. The grid illustrates a finite-domain discrete input data space with crosses indicating training data and circles marking outliers in the test data. For unbounded continuous input variables, the test data can lie well outside the grid and much farther from the training data, as illustrated by the stars.

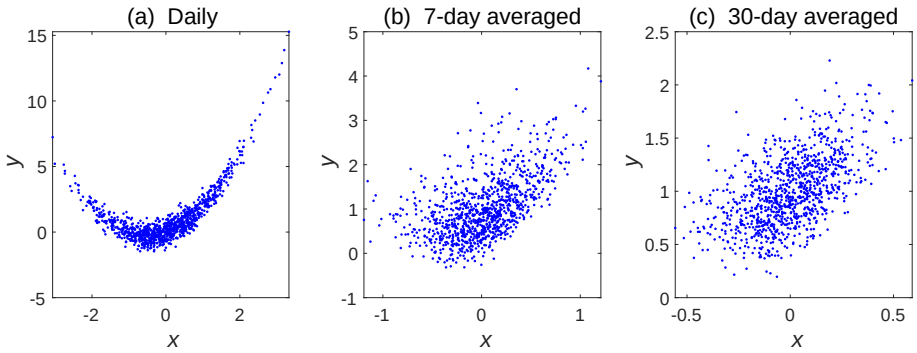


Figure 1.3 Effects of time-averaging on the nonlinear relation (1.1).(a) Synthetic 'daily' data from a quadratic relation between x and y. The data time-averaged over (b) 7 observations and (c) 30 observations. [Follows Hsieh and Cannon (2008).]

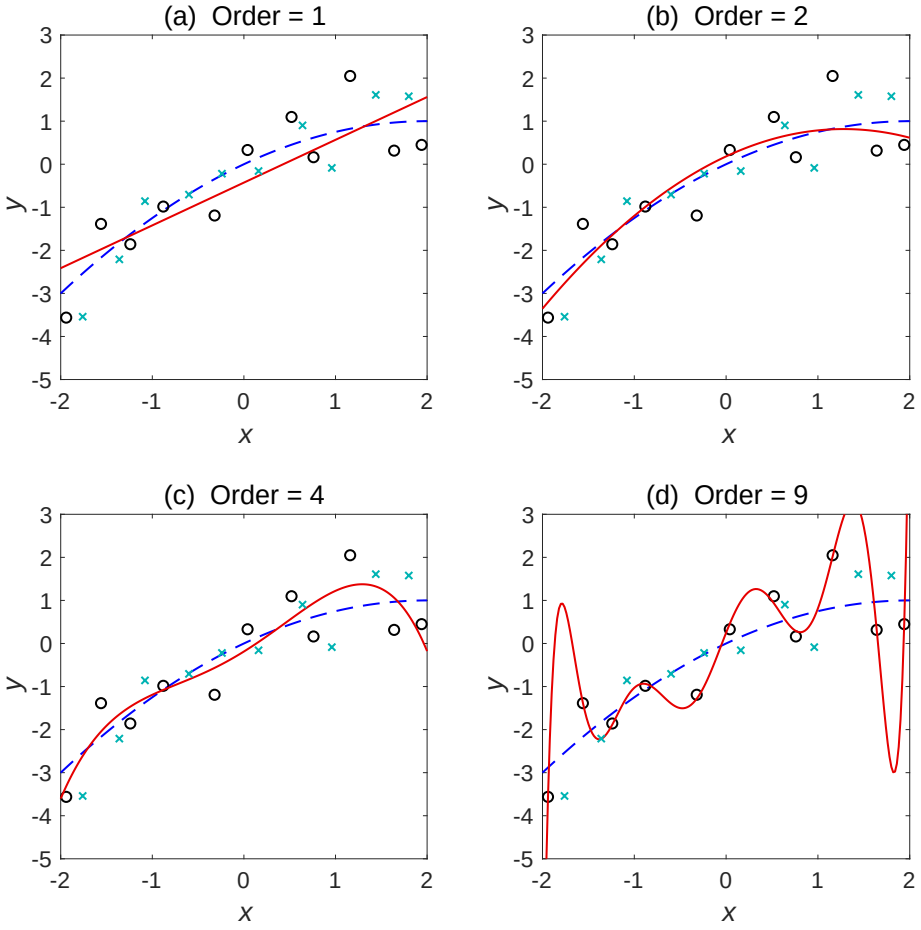


Figure 1.4 Polynomial fit to data using a polynomial of order (a) 1, (b) 2, (c) 4 and (d) 9. The circles indicate the 11 data points used for fitting (i.e. training), the solid curve the polynomial solution $\hat{y}$ and the dashed curve the true signal ($y_{\text{signal}} = x - 0.25\,x^2$). The crosses show 10 new data points used to validate the polynomial fit.

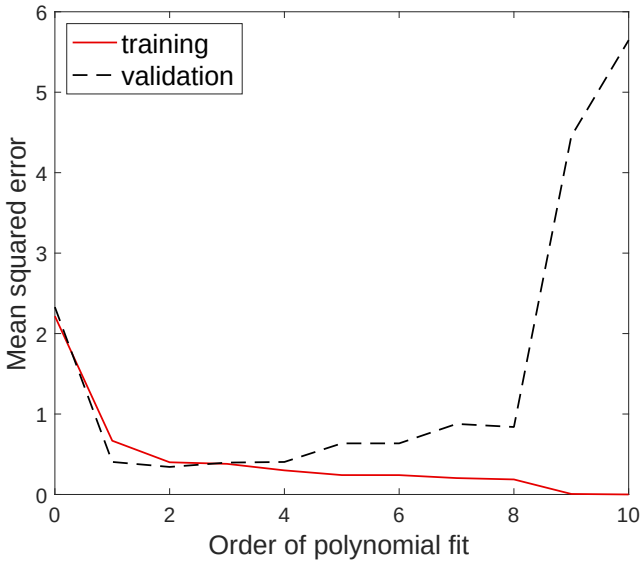


Figure 1.5 Mean squared error of the training and validation data as the order of the polynomial fit varies from 0 to 10.

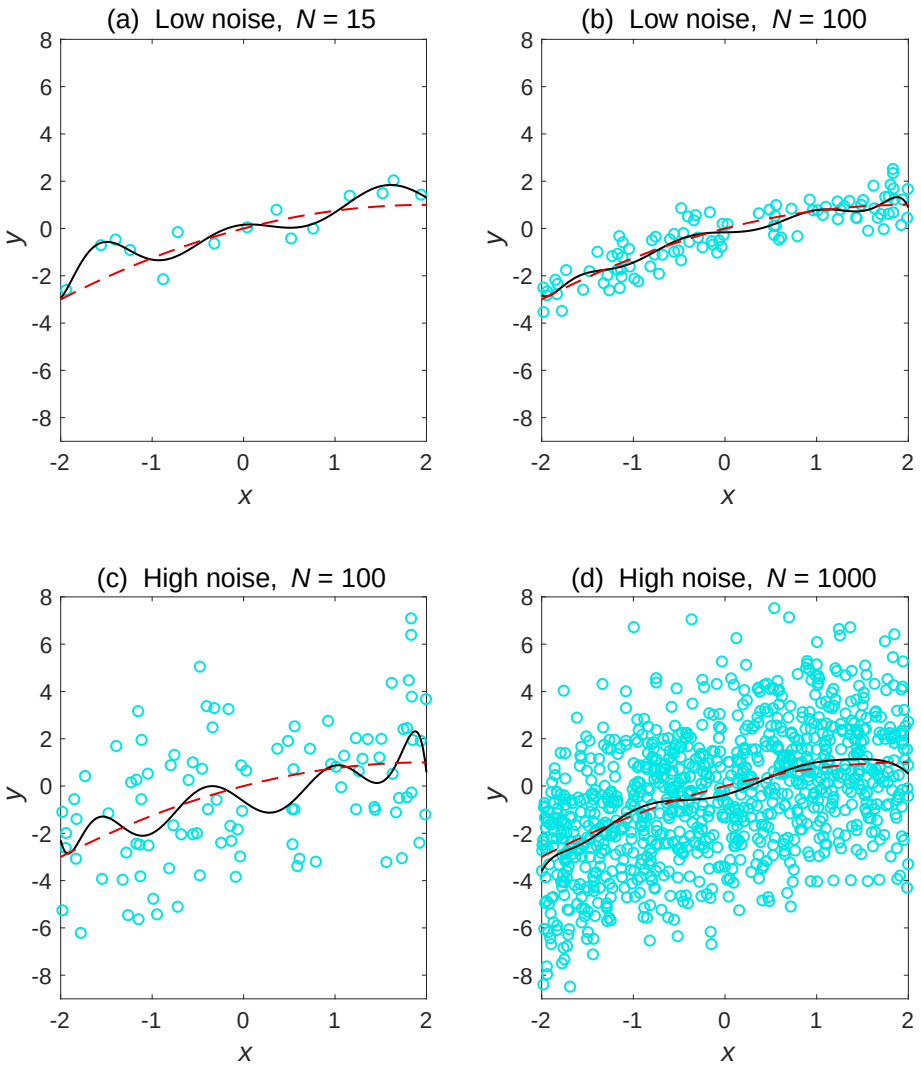


Figure 1.6 The ninth order polynomial fit to data with two noise levels and different numbers of training data points. The circles indicate the data points used for training, the solid curve the polynomial solution $\hat{y}$ and the dashed curve the true signal y_{signal}.

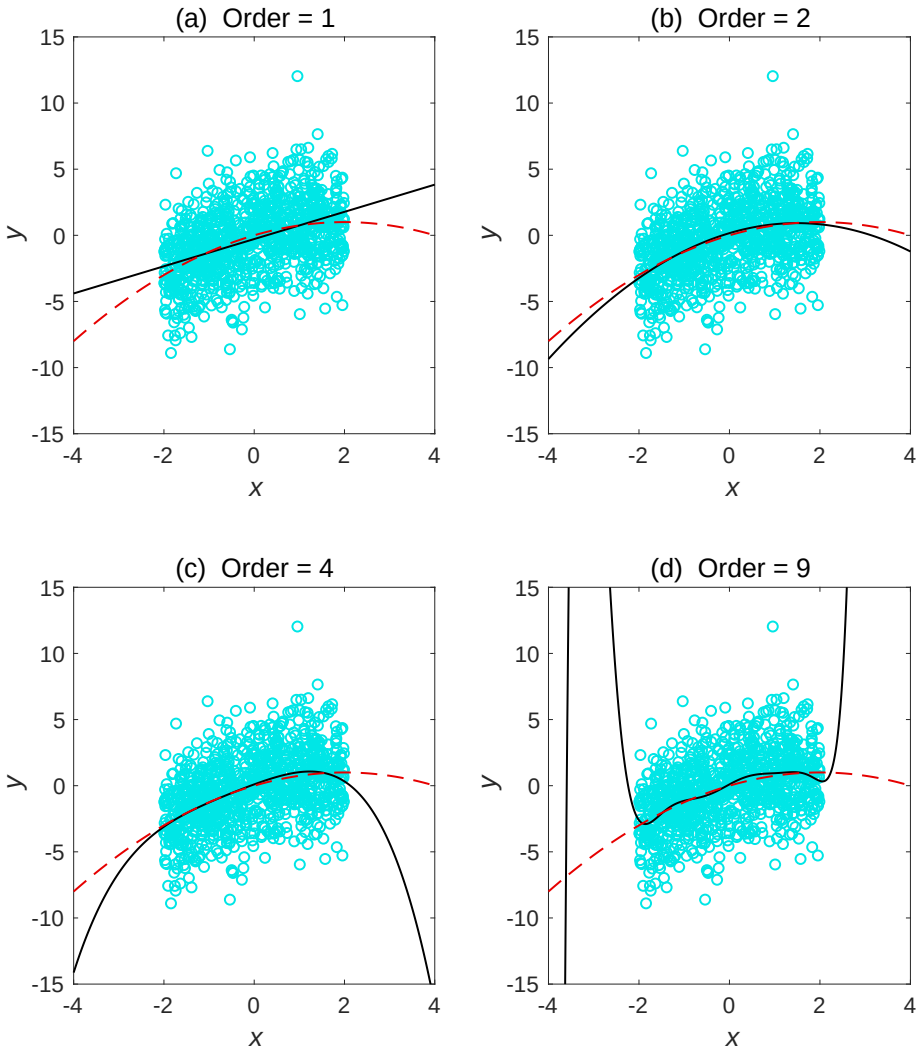


Figure 1.7 Extrapolating the polynomial solution to beyond the training data domain, where 1,000 data points (circles) were used for training and the order of the polynomial was (a) 1, (b) 2, (c) 4 and (d) 9. In (d), extending to the left side, the curve first shot up beyond the top of the plot, then plunged back down.

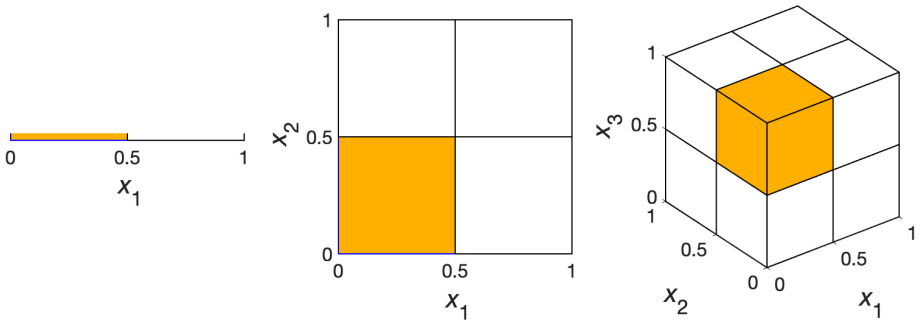


Figure 1.8 The 'curse of dimensionality' effect, as one proceeds from one to three dimensions.

Chapter 2: Basics

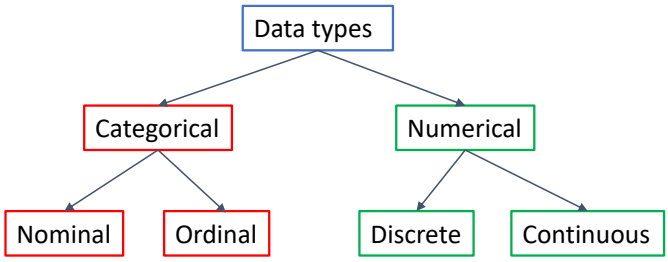


Figure 2.1 Main types of data.

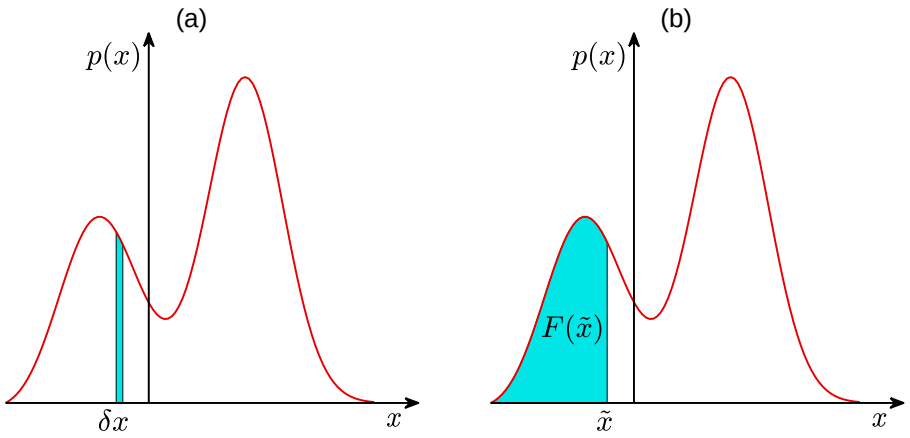


Figure 2.2 (a) The probability of x lying within the interval $(x, x+\delta x)$ is given by the area of the narrow vertical band of height $p(x)$ and width δx. The two peaks in $p(x)$ indicate the two regions of higher probability. (b) The cumulative distribution $F(\tilde{x})$ is given by the shaded area under the curve.

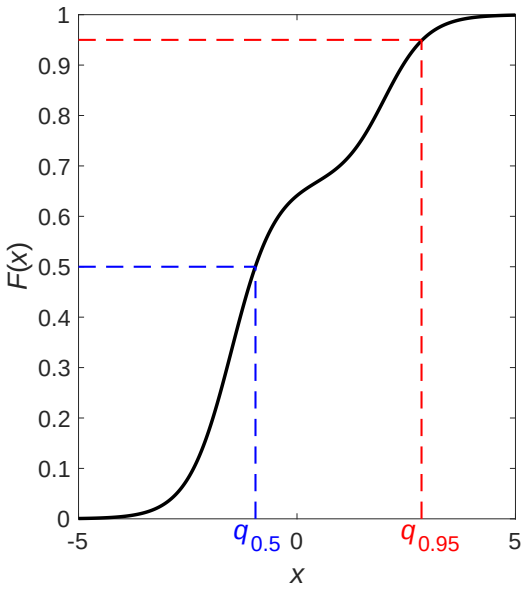


Figure 2.3 A cumulative distribution function $F(x)$. By inverse mapping from the ordinate to the abscissa, one can obtain the quantiles q_α. The 95th percentile $q_{0.95}$ and the median $q_{0.5}$ are shown.

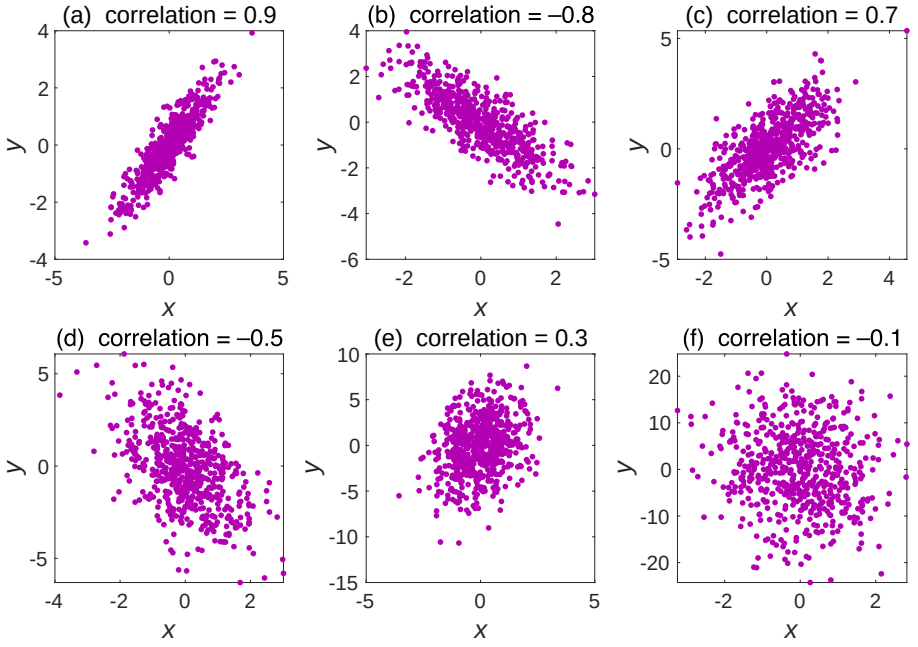


Figure 2.4 Scatterplots showing distribution of (x, y) data and the corresponding Pearson correlation coefficient as the noise level rises from (a) to (f).

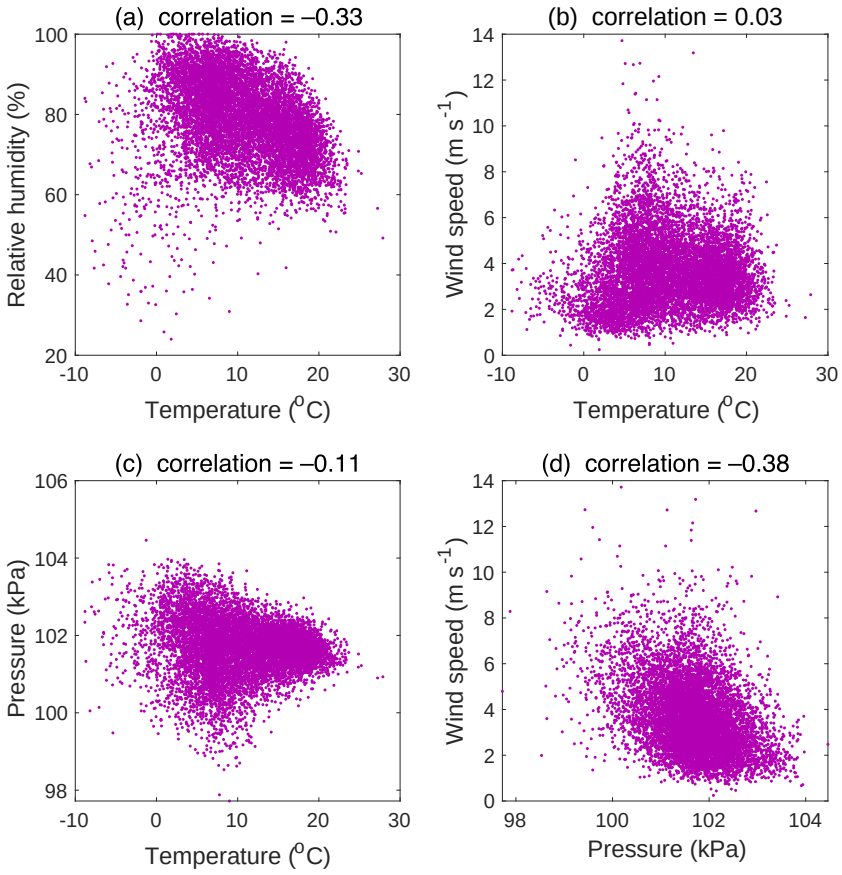


Figure 2.5 Scatterplots and Pearson correlation coefficients of daily weather variables from Vancouver, BC, Canada, with 25 years of data (1993–2017). [Data source: weatherstats.ca based on Environment and Climate Change Canada data.]

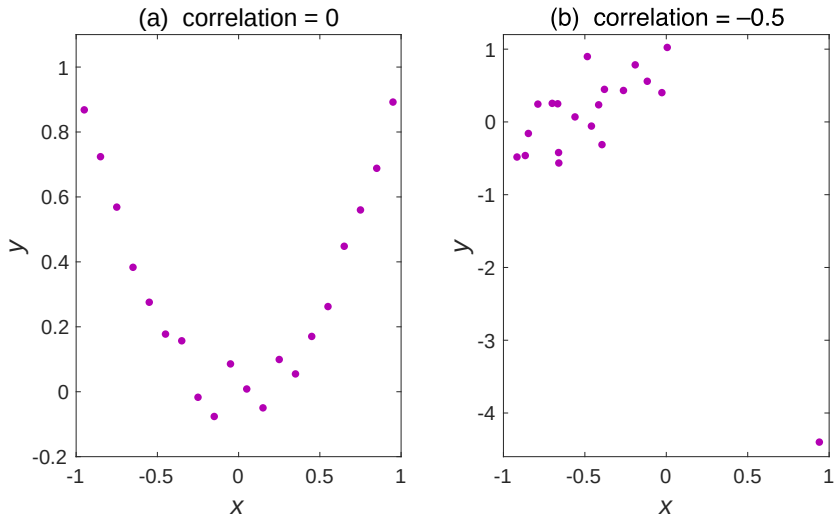


Figure 2.6 (a) An example illustrating that correlation is not robust to deviations from linearity. Here, the strong non-linear relation between x and y is completely missed by the near-zero correlation coefficient. (b) An example showing that correlation is not resistant to outliers. By removing the single outlier on the lower right corner, the correlation coefficient changes from negative to positive.

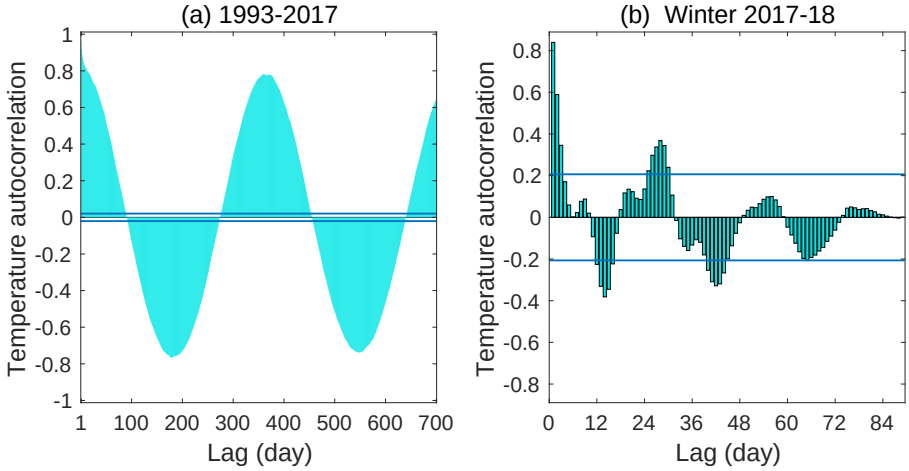


Figure 2.7 Autocorrelation function for the daily temperature at Vancouver, BC during (a) 1993–2017 and (b) winter of 2016–17 (Dec.-Feb.), with the horizontal lines indicating the 95% confidence interval. [Data source: weatherstats.ca based on Environment and Climate Change Canada data.]

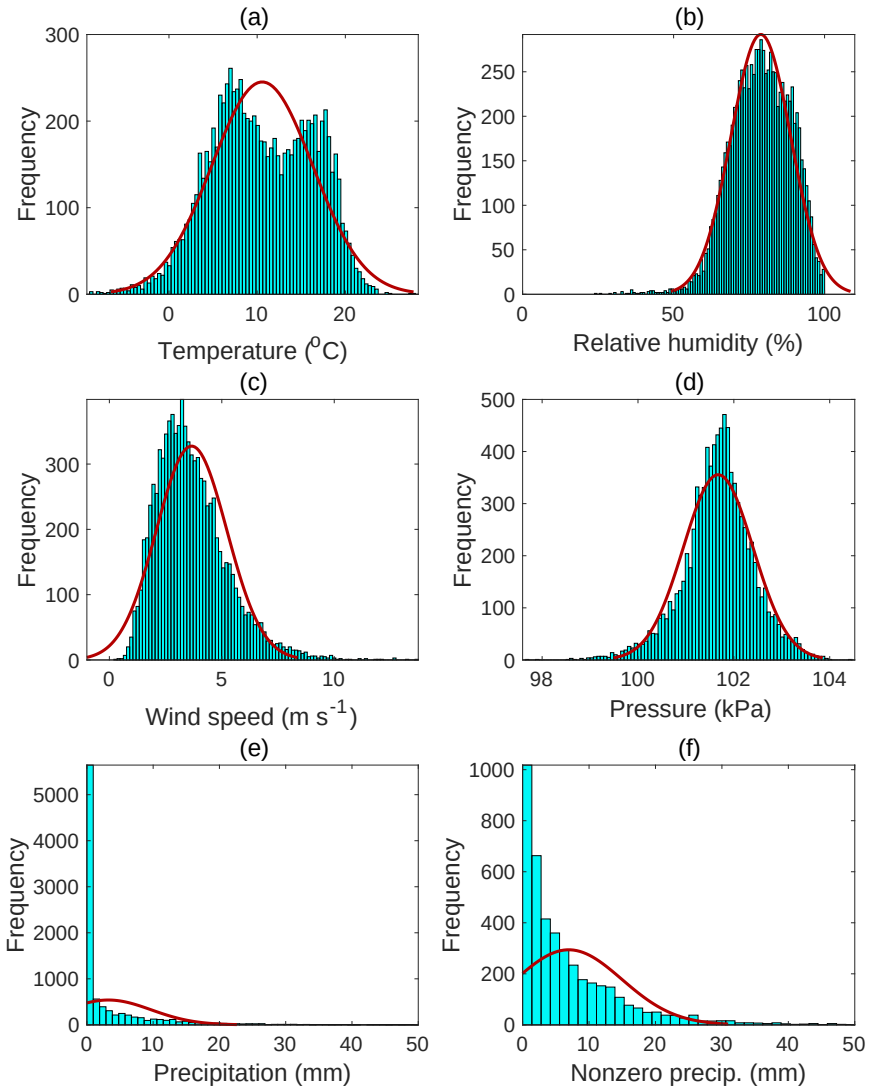


Figure 2.8 Histogram for the distribution of daily (a) temperature, (b) relative humidity, (c) wind speed, (d) sea level pressure, (e) precipitation and (f) non-zero precipitation in Vancouver, BC from 1993–2017. A Gaussian distribution curve has also been fitted to the data. Relative humidity is bounded between 0% and 100%, and wind speed is non-negative. Since 53.4% of the days in (e) have no precipitation, the dry days are omitted in (f). [Data source: weatherstats.ca based on Environment and Climate Change Canada data.]

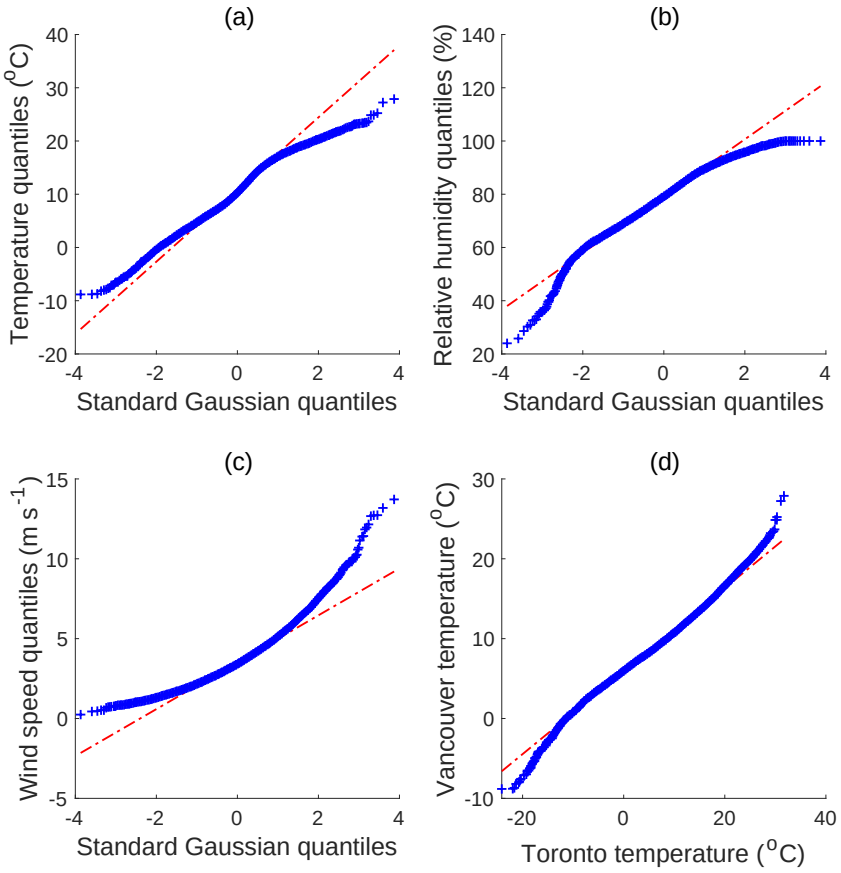


Figure 2.9 Quantile–quantile plots where quantiles of the daily (a) temperature, (b) relative humidity and (c) wind speed in Vancouver, BC from 1993–2017 are plotted against the quantiles of the standard Gaussian distribution as indicated by the '+' symbols. If the observed distribution is a perfect Gaussian, the plot will fall on the straight (dot-dashed) line. In (d), the quantiles of the temperature in Toronto, Ontario are plotted against those from Vancouver. [Data source: weatherstats.ca based on Environment and Climate Change Canada data.]

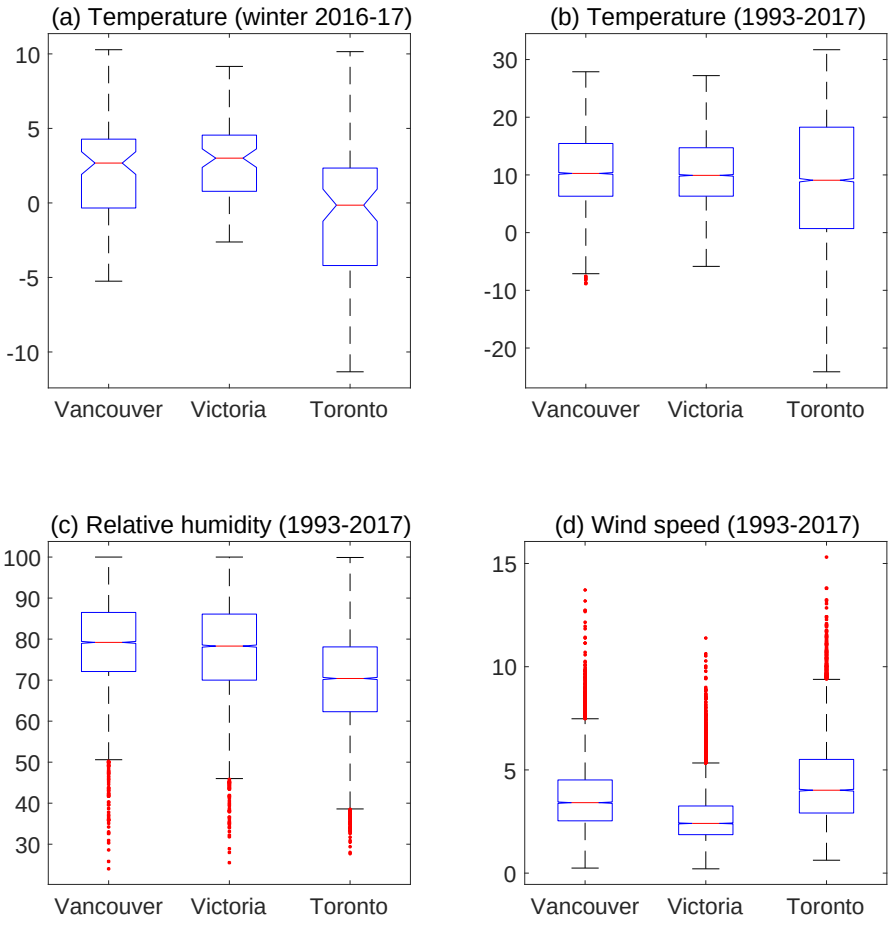


Figure 2.10 Boxplots for the daily weather at three Canadian cities: (a) temperature during the winter of 2016–17, and (b) temperature, (c) relative humidity and (d) wind speed from 1993–2017. [Data source: weatherstats.ca based on Environment and Climate Change Canada data.]

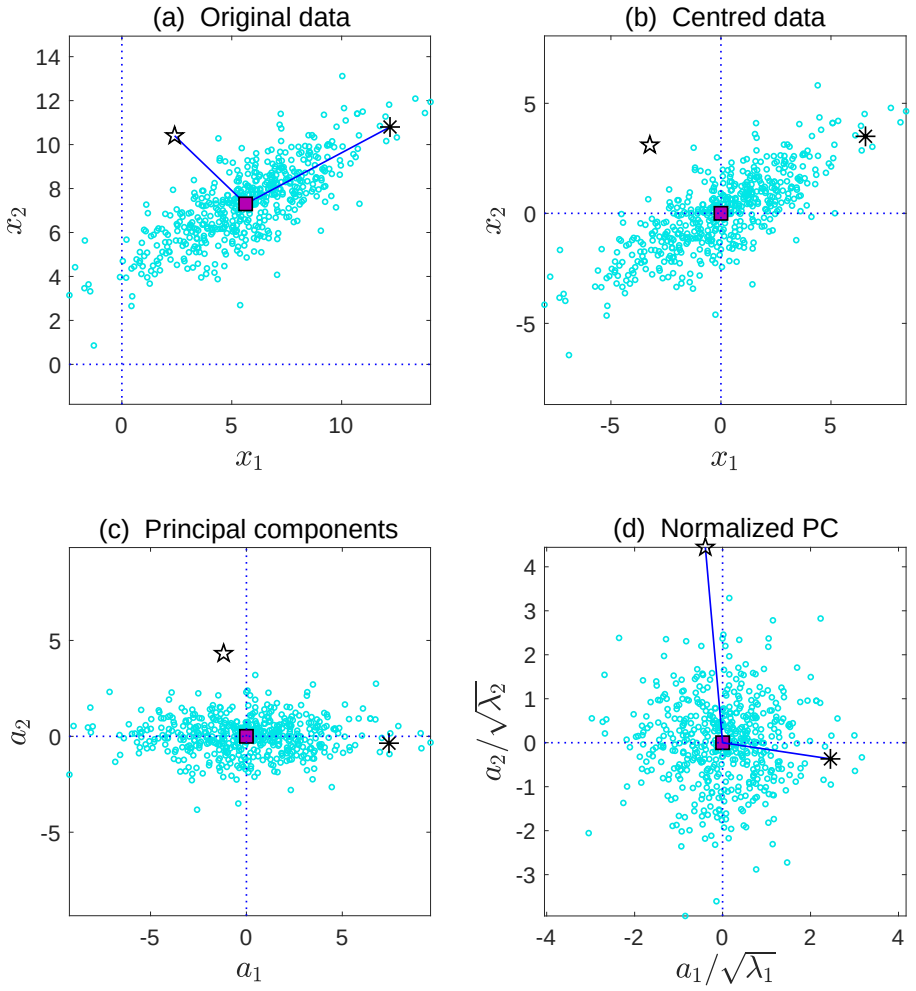


Figure 2.11 Mahalanobis distances versus Euclidean distances, as illustrated by two data points marked by the asterisk and the star. The square marks the centre (i.e. the mean) of the Gaussian dataset containing 500 points. (a) In the original data, the line marking the Euclidean distance from the centre is longer for the asterisk than for the star. (b) Subtracting the mean gives the centred data. (c) Principal components (a_1 and a_2) are obtained by rotating the centred data, so the direction of the maximum variance is along the horizontal axis. (d) Principal components are normalized to have unit variance in each direction. The line connecting the centre and the asterisk/star gives the Mahalanobis distance. Thus in terms of Euclidean distance, the asterisk is further from the centre than the star, but in terms of Mahalanobis distance, the star is further from the centre than the asterisk.

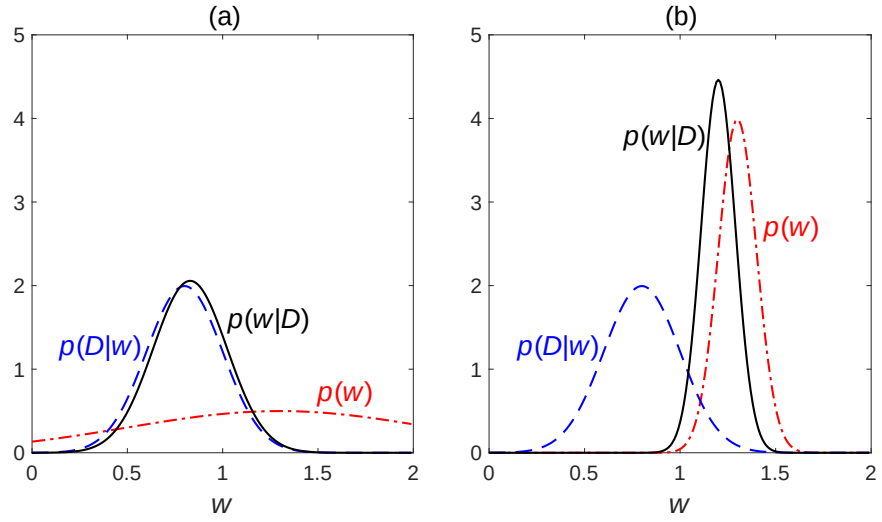


Figure 2.12 Relation between $p(w|D)$, $p(D|w)$ and $p(w)$. (a) A broad and flat distribution of $p(w)$ provides little prior information for estimating w, leading to the posterior distribution $p(w|D)$ being very similar to the likelihood $p(D|w)$. (b) A narrow $p(w)$ distribution leads to a larger difference between $p(w|D)$ and $p(D|w)$. If more data are available, $p(D|w)$ will be narrower and more strongly peaked than that shown in (b), and the $p(w|D)$ distribution will be pulled more towards $p(D|w)$. [Follows Cowan (2007)].

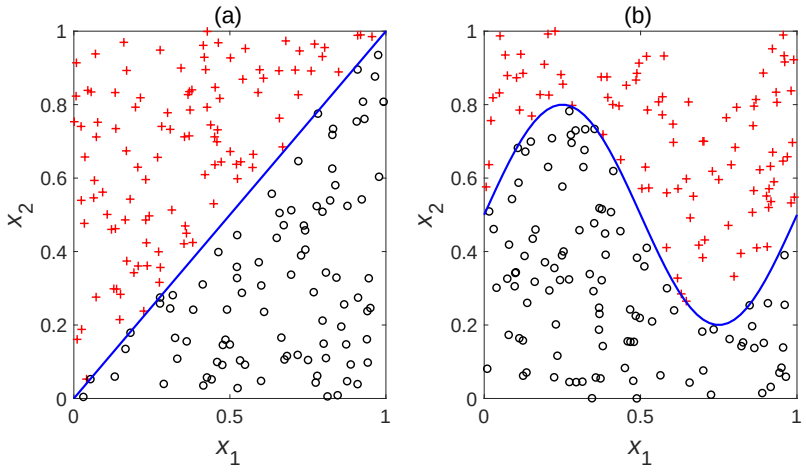


Figure 2.13 (a) A linear decision boundary separating two classes of data denoted by crosses and circles, respectively. (b) A non-linear decision boundary.

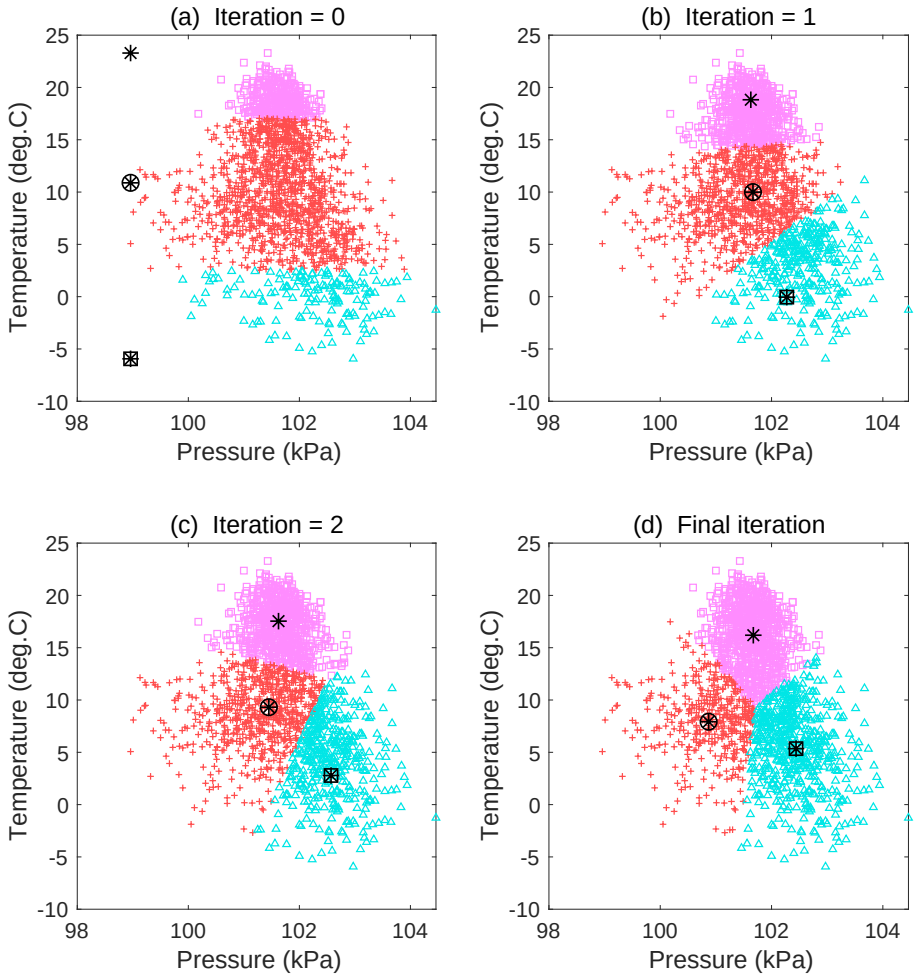


Figure 2.14 (a) The initial guesses for the three centroids are marked by three asterisks. The data points are assigned to clusters based on their nearest centroid. In (b), the centroids have been recalculated based on the mean position of the cluster members in (a), and cluster members in (b) have been reassigned based on their closeness to the centroids in (b). The location of the centroids and their associated cluster members are shown after (c) two iterations and (d) after final convergence of the K-means clustering algorithm.

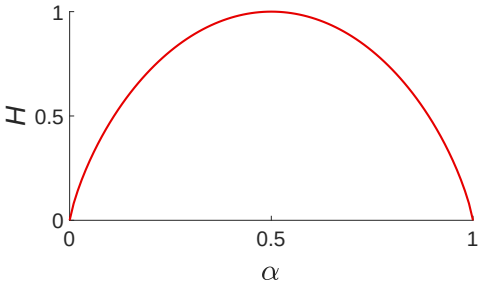


Figure 2.15 Entropy H as a function of α. When $\alpha = 0.5$, the maximum ($H = 1$) is attained.

Chapter 3: Probability distributions

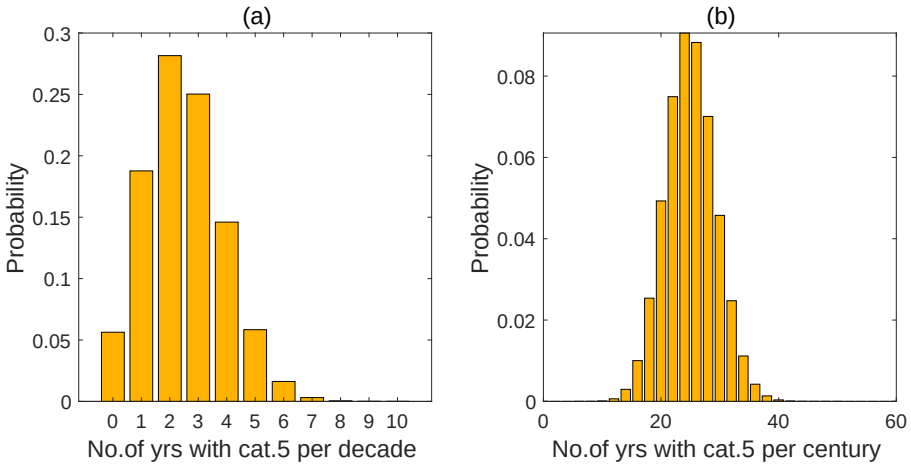


Figure 3.1 Probability distribution of the number of years with category 5 hurricane(s) (a) per decade and (b) per century. The binomial distribution with $p = 0.25$ was used with (a) $N = 10$ and (b) $N = 100$. As N becomes large, the skewness of the distribution disappears, and the binomial distribution can be approximated by the Gaussian (normal) distribution.

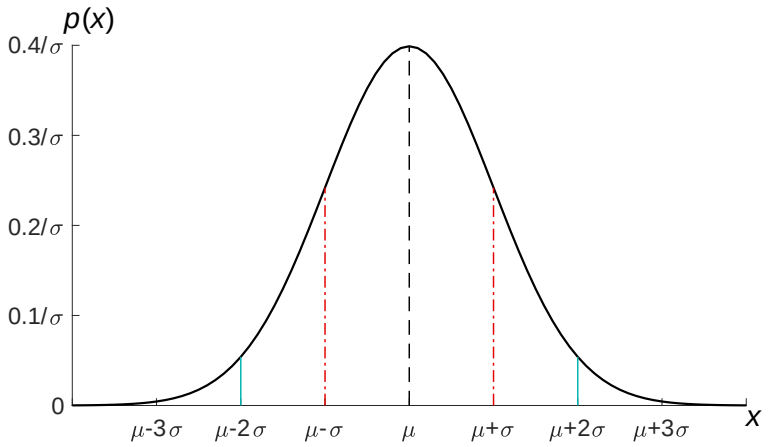


Figure 3.2 Probability density $p(x)$ of a Gaussian distribution with mean μ and standard deviation σ.

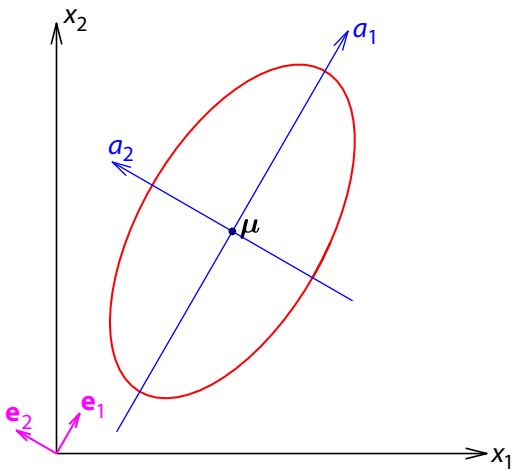


Figure 3.3 The elliptical contour (with semi-major axis of length $\lambda_1^{1/2}$ and semi-minor axis of $\lambda_2^{1/2}$) shows where the probability density is $\exp(-1/2)$ times that at the centre $\boldsymbol{\mu}$ for the Gaussian distribution in a two-dimensional space (x_1, x_2). The semi-major and semi-minor axis are pointed in the directions given by the eigenvectors $\mathbf{e}_1$ and $\mathbf{e}_2$, respectively, while a_1 and a_2 are the distances (measured from $\boldsymbol{\mu}$) along these two directions.

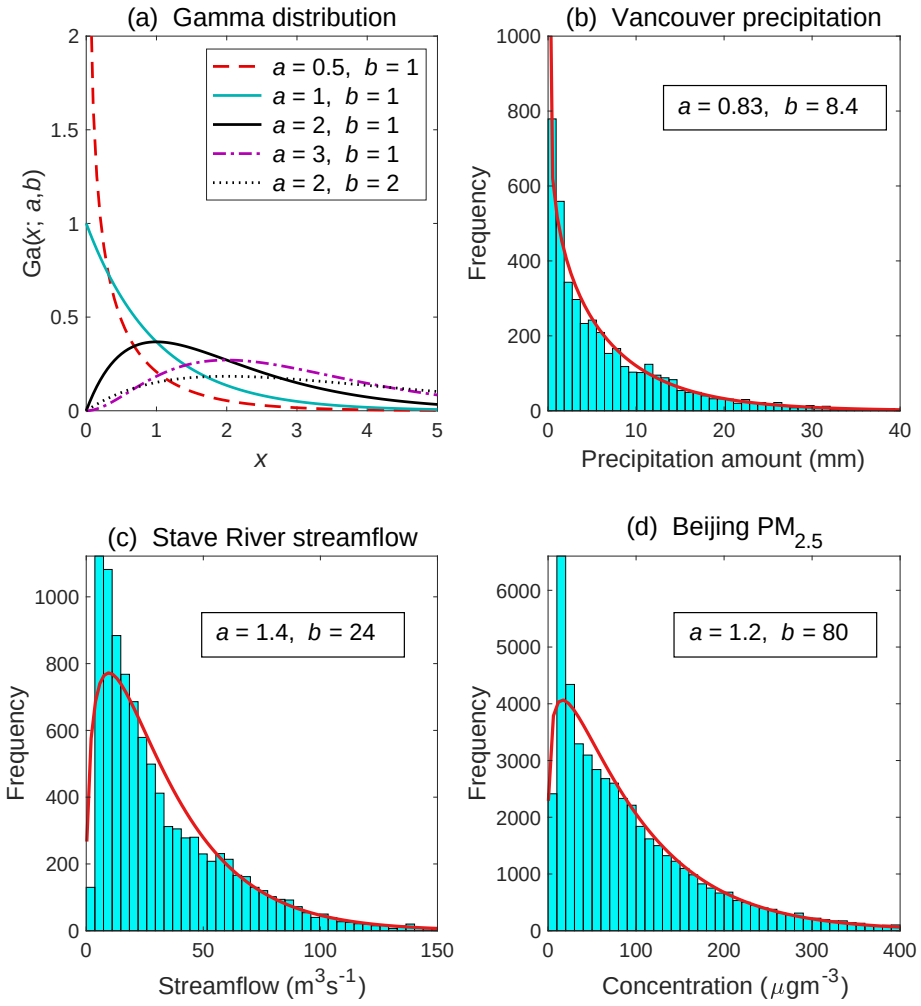


Figure 3.4 (a) Gamma probability density distribution for several values of the shape parameter a and the scale parameter b, and histograms of (b) non-zero daily precipitation amount in Vancouver, BC (1993–2017), (c) daily streamflow of Stave River, BC (1985–2011) and (d) hourly concentration of the atmospheric pollutant $PM_{2.5}$ in Beijing (2010–2015). The gamma distribution is fitted to the histograms, with the values of the parameters a and b given in the legends. [Data source: (b) weatherstats.ca based on Environment and Climate Change Canada data, (c) Water Survey of Canada and (d) Machine Learning Repository, University of California Irvine, with data from X. Liang et al. (2016).]

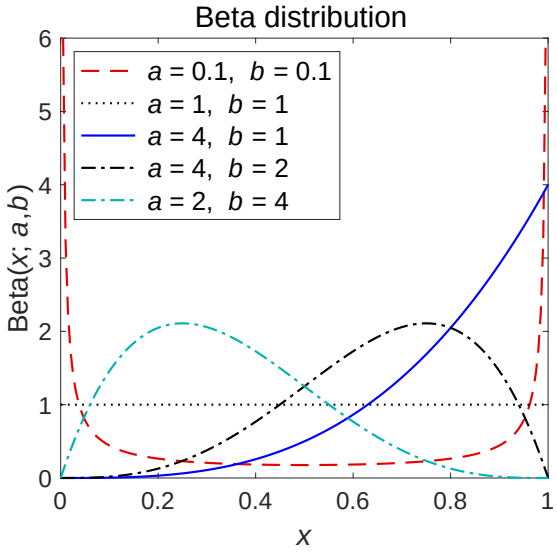


Figure 3.5 Beta probability density distribution for several values of the parameters a and b. When $a = b = 1$, it turns into the uniform distribution. When the parameters are interchanged, as seen between the cases $a = 4$, $b = 2$ and $a = 2$, $b = 4$, the curves are mirror images of each other.

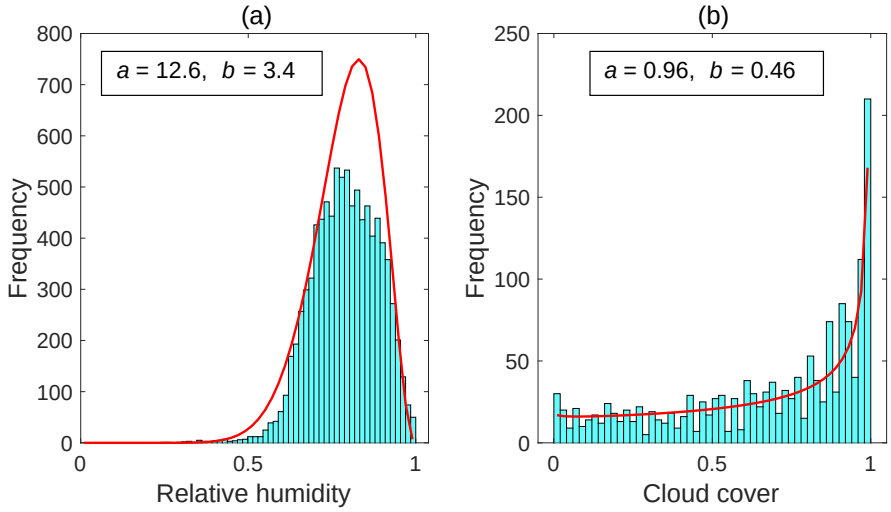


Figure 3.6 Histograms of (a) daily relative humidity and (b) daily cloud cover in Vancouver, BC (1993–2017). The beta distribution is fitted to the histograms, with the values of the parameters a and b given in the legends. [Data source: weatherstats.ca based on Environment and Climate Change Canada data.]

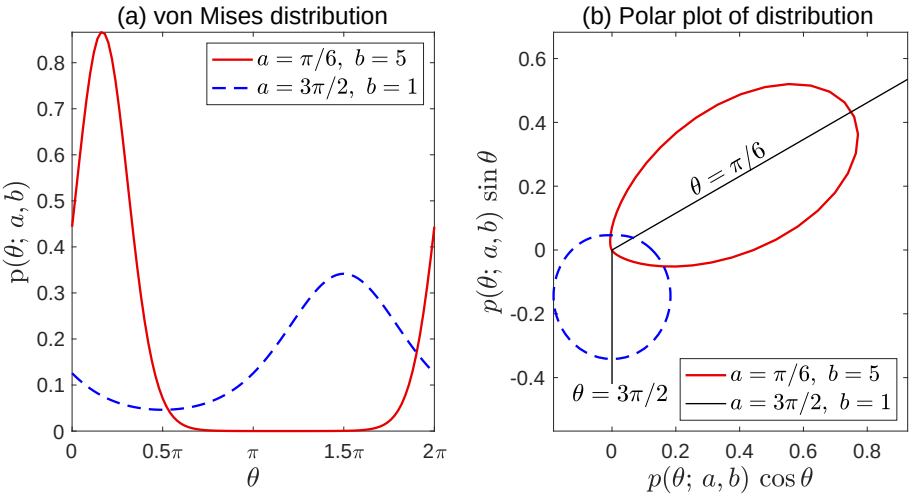


Figure 3.7 (a) von Mises probability density distribution for two sets of the parameters a and b. (b) The same distribution shown in a polar plot $(r\cos\theta, r\sin\theta)$, where r, the radial distance from the origin, is given by $p(\theta|\ a, b)$.

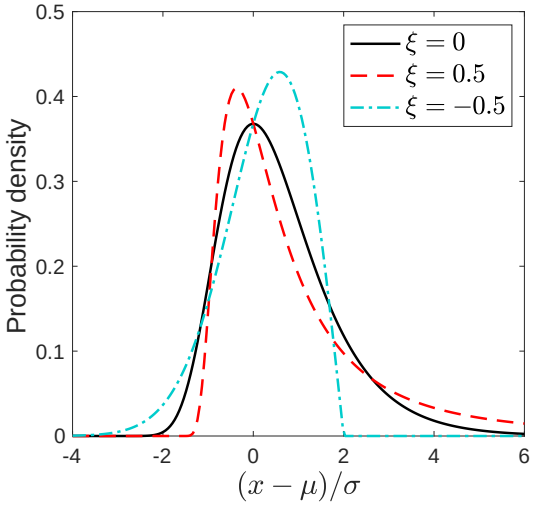


Figure 3.8 GEV probability density curves for the three sub-families: type I ($\xi = 0$), type II ($\xi > 0$) and type III ($\xi < 0$).

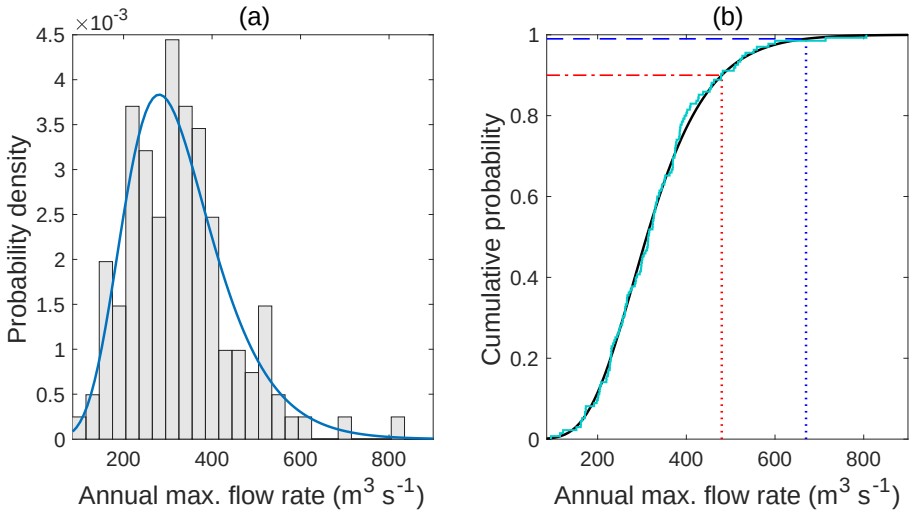


Figure 3.9 GEV fit to the annual maximum daily flow rate of the river Thames at Kingston, UK, with (a) the PDF from the GEV model shown as a curve over the histogram of the observed data, and (b) the CDF from the GEV model shown by the solid black curve, with the empirical CDF shown by the fainter curve. The 0.9 and 0.99 quantiles of the GEV model are indicated by the horizontal dot-dashed and dashed line, respectively, with the corresponding vertical dotted lines indicating, respectively, the 10 year and 100 year return levels along the abscissa. [Data source: National River Flow Archive, UK.]

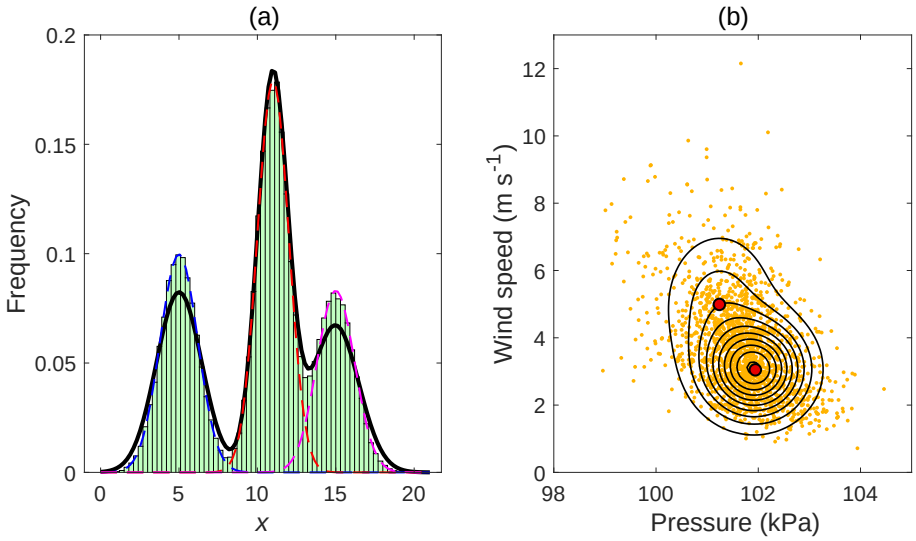


Figure 3.10 (a) Histogram of a synthetic dataset generated from mixing three Gaussian distributions (shown individually by the three dashed curves). Fitting a three-component Gaussian mixture model to the data yields the solid curve with three peaks. (b) A two-component Gaussian mixture model is fitted to the daily pressure and wind speed data (indicated by dots), from Vancouver, BC during 2013–2017. The PDF from the Gaussian mixture model is shown by the contours, with the contour interval being 0.02. The two circles indicate the mean position $\boldsymbol{\mu}_k$ for the two Gaussian components. [Data source: weatherstats.ca based on Environment and Climate Change Canada data.]

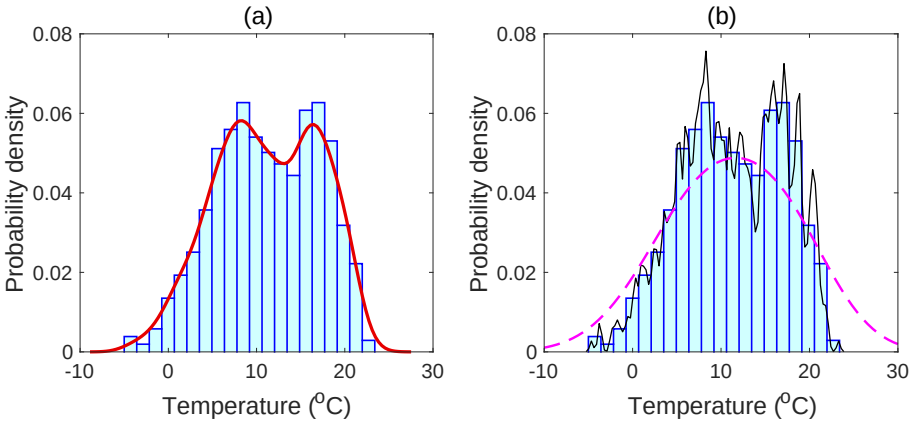


Figure 3.11 (a) Probability density distribution from histogram of daily temperature at Vancouver, BC (2014–2015), and from kernel density estimation (with Gaussian kernel and bandwidth $h = 1.4°C$) as shown by the smooth curve. (b) Kernel density estimation with $h = 0.2°C$ (thin solid curve) and $h = 5°C$ (dashed curve). [Data source: weatherstats.ca based on Environment and Climate Change Canada data.]

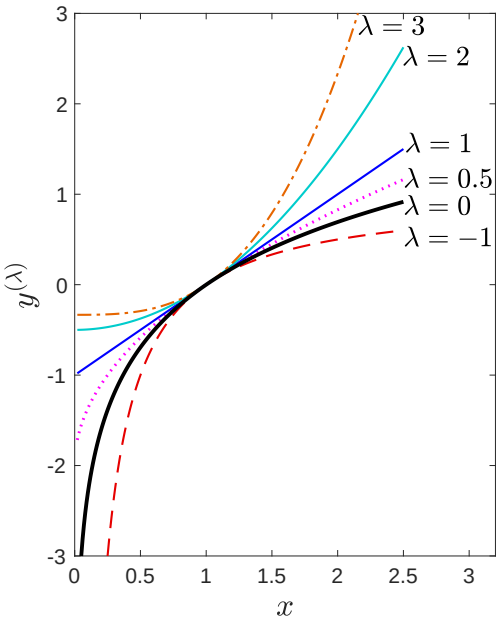


Figure 3.12 Box–Cox transform for various values of the parameter λ. $\lambda = 1$ gives a straight line, while $\lambda > 1$ have transformed variables more positively (i.e. right) skewed than the original variables, and $\lambda < 1$ have transformed variables more negatively (i.e. left) skewed than the original.

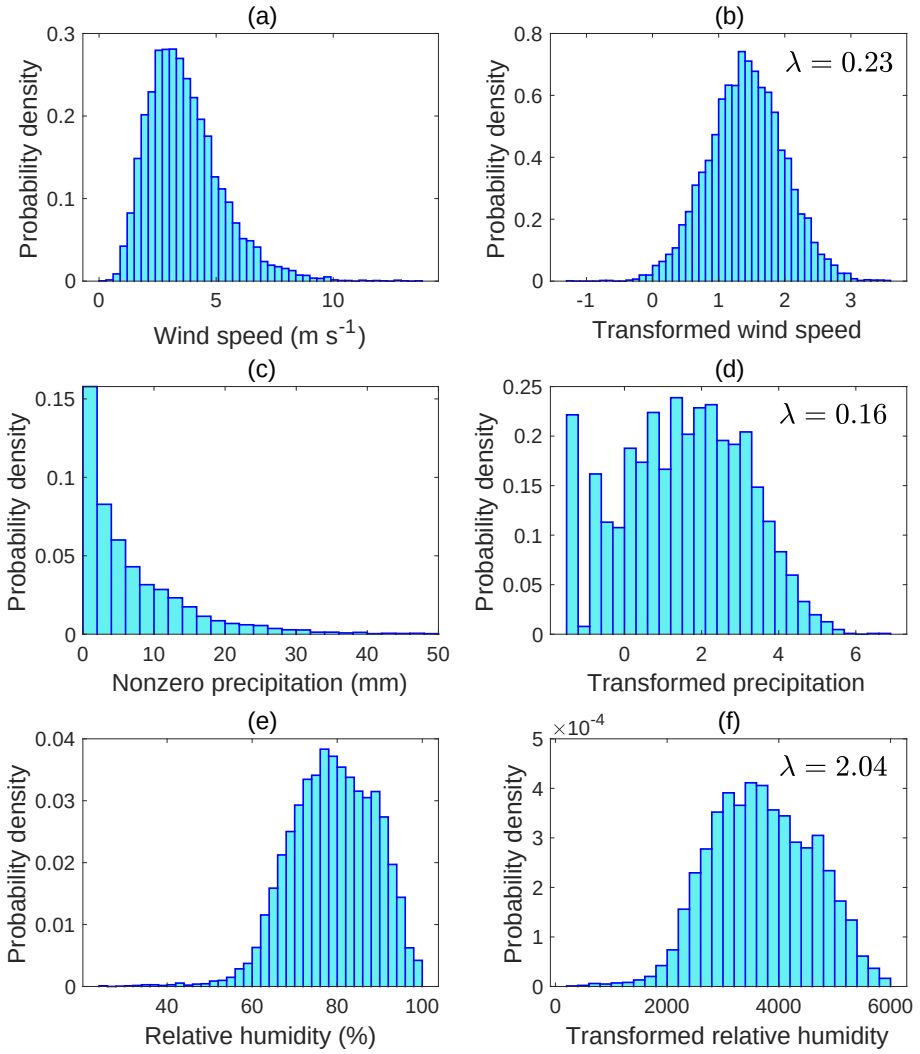


Figure 3.13 Histograms showing original data distribution (left column) and Box–Cox transformed distribution (right column). (a), (c) and (e) are the wind speed, non-zero precipitation and relative humidity in Vancouver, BC during 1993–2017, while (b), (d) and (f) are the corresponding transformed distributions, with $\lambda = 0.23$, 0.16 and 2.04, respectively. [Data source: weatherstats.ca based on Environment and Climate Change Canada data.]

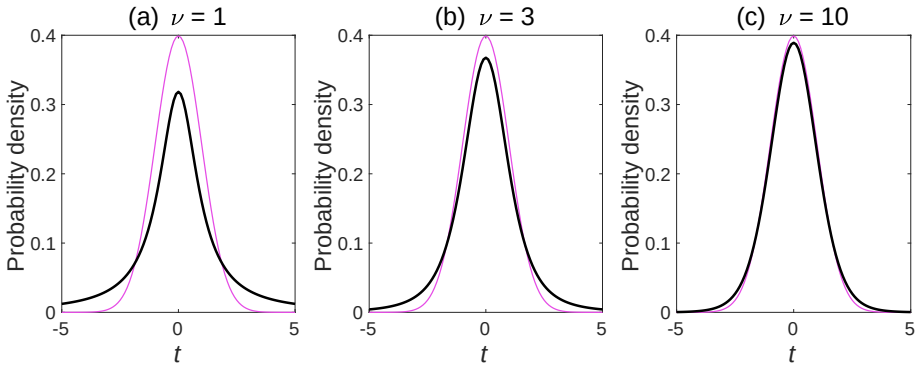


Figure 3.14 The t-distribution with various degrees of freedom: (a) $\nu = 1$, (b) $\nu = 3$ and (c) $\nu = 10$. As ν increases, the t-distribution approaches the standard Gaussian distribution $\mathcal{N}(0, 1)$, shown by a light, thin curve.

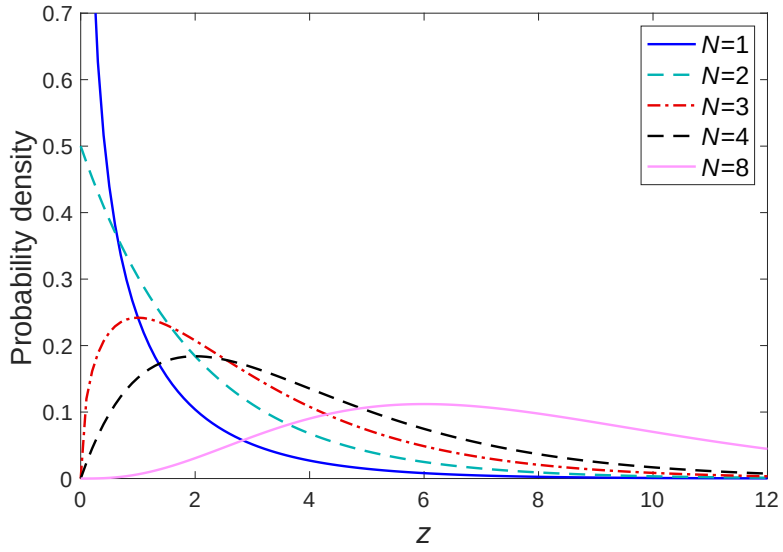


Figure 3.15 The chi-squared distribution with various degrees of freedom (N). The PDF has mean N and variance $2N$. As N increases, the PDF approaches the Gaussian distribution.

Chapter 4: Statistical inference

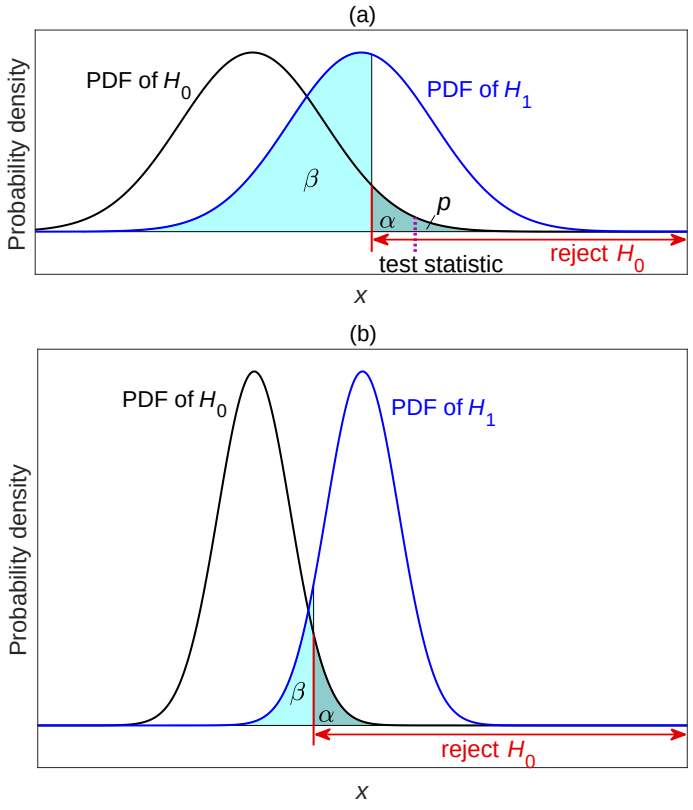


Figure 4.1 (a) Probability density functions of the null hypothesis H_0 and the alternative hypothesis H_1. The vertical solid line marks the critical value for rejecting H_0 at the α level ($\alpha = 0.05$); that is, if the value of the test statistic lies to the right of the critical value, H_0 is rejected. The dark shaded region of area α underneath the PDF of H_0 indicates that there is a probability of α where H_0 is actually true though the test statistic turned up in the region for rejecting H_0 (type I error). The value of the test statistic is marked by the vertical dotted line, and p is the dark shaded area to its right. The light shaded region of area β underneath the PDF of H_1 indicates a probability of β for failing to reject H_0 when H_1 is true (type II error). The *power* of the test is $1 - \beta$, that is, the area under the H_1 PDF to the right of the critical value, and is the probability of correctly accepting H_1. (b) The test statistic is next estimated using many more data points, so the spread of the PDF is much reduced. For the same α value, the vertical line marking the critical value is shifted to the left; the area β is much reduced and the power much enhanced.

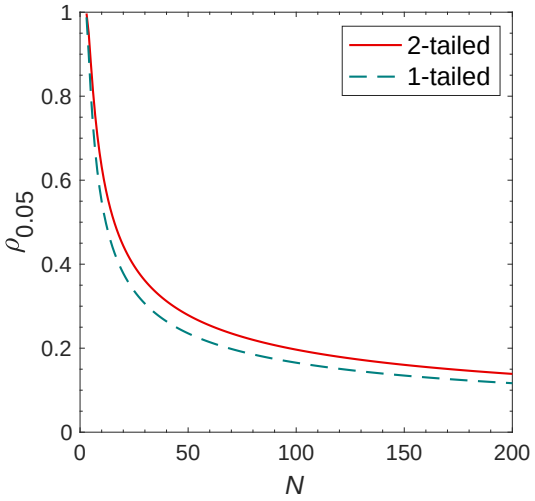


Figure 4.2 The critical correlation value at the 5% level ($\rho_{0.05}$) as a function of the sample size N, for the two-tailed test (solid curve) and the one-tailed test (dashed).

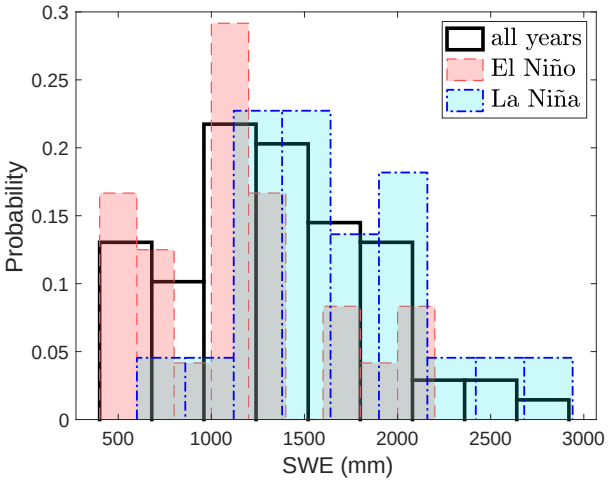


Figure 4.3 Histogram of the winter snow water equivalent distribution at Grouse Mountain, BC for all winters, El Niño winters and La Niña winters. [Data source: River Forecast Centre, British Columbia.]

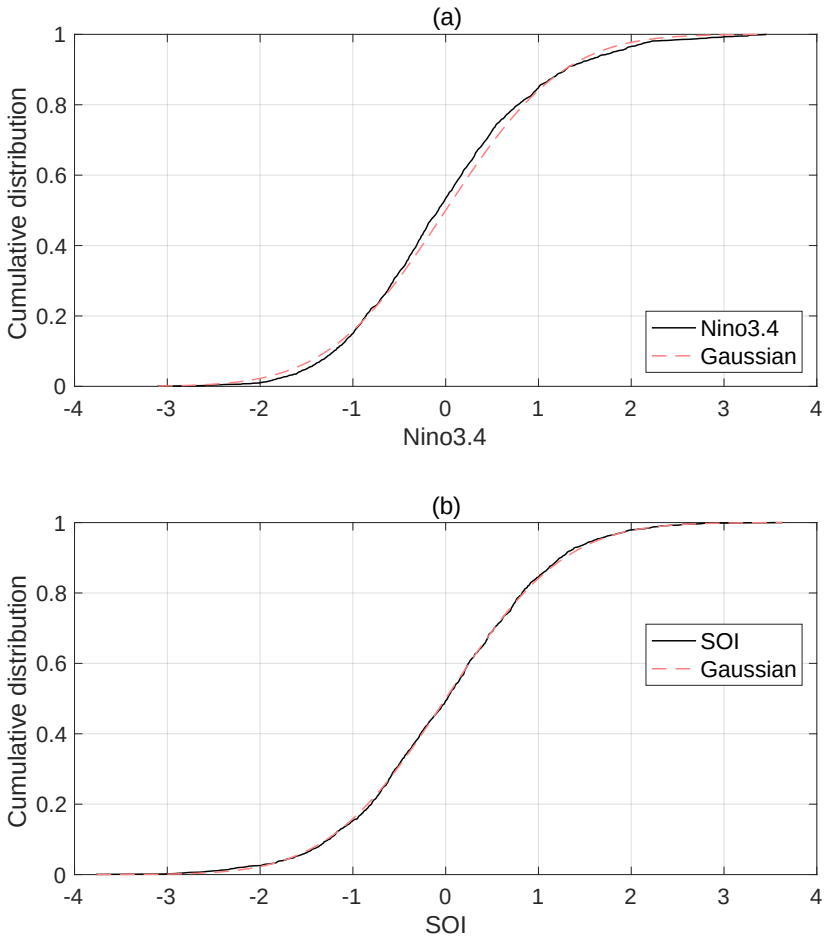


Figure 4.4 Empirical cumulative distribution function (solid curve) for (a) the standardized Niño3.4 index and (b) the standardized SOI index, using monthly data from 1870–2017, with the reference distribution, the standard Gaussian, shown by the dashed curve. Close inspection reveals the empirical distribution curves to be non-smooth, as they vary by steps. D_N, the KS test statistic, is simply the maximum vertical distance between the solid and dashed curves. [Data source: Climatic Research Unit, University of East Anglia.]

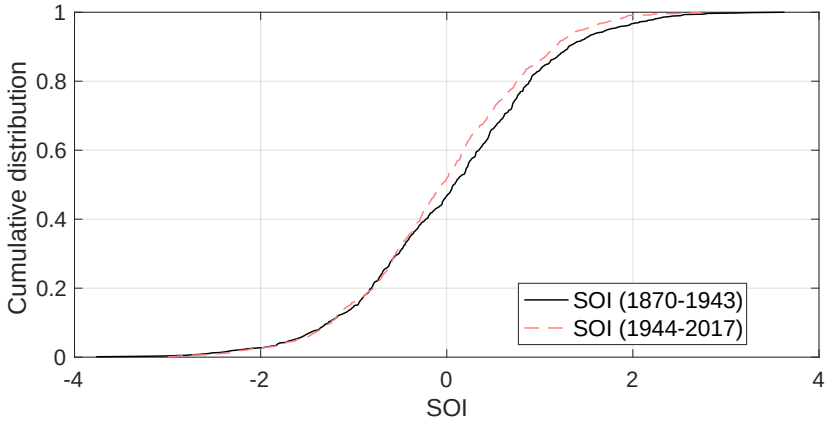


Figure 4.5 The empirical distribution functions for SOI during 1870–1943 and during 1944–2017. The $D_{N,M}$ test statistic from the two-sample KS test is the maximum vertical distance between the two curves. [Data source: Climatic Research Unit, University of East Anglia.]

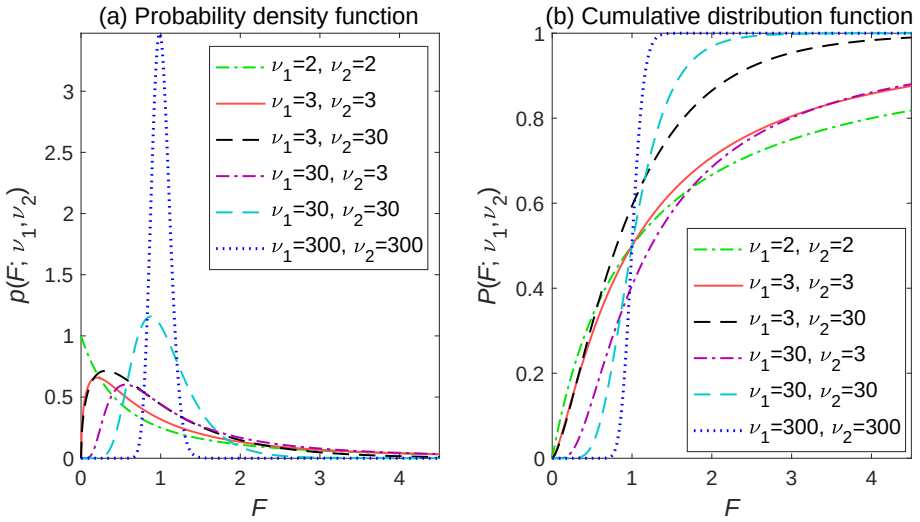


Figure 4.6 (a) PDF and (b) CDF of the F distribution with ν_1 and ν_2 degrees of freedom.

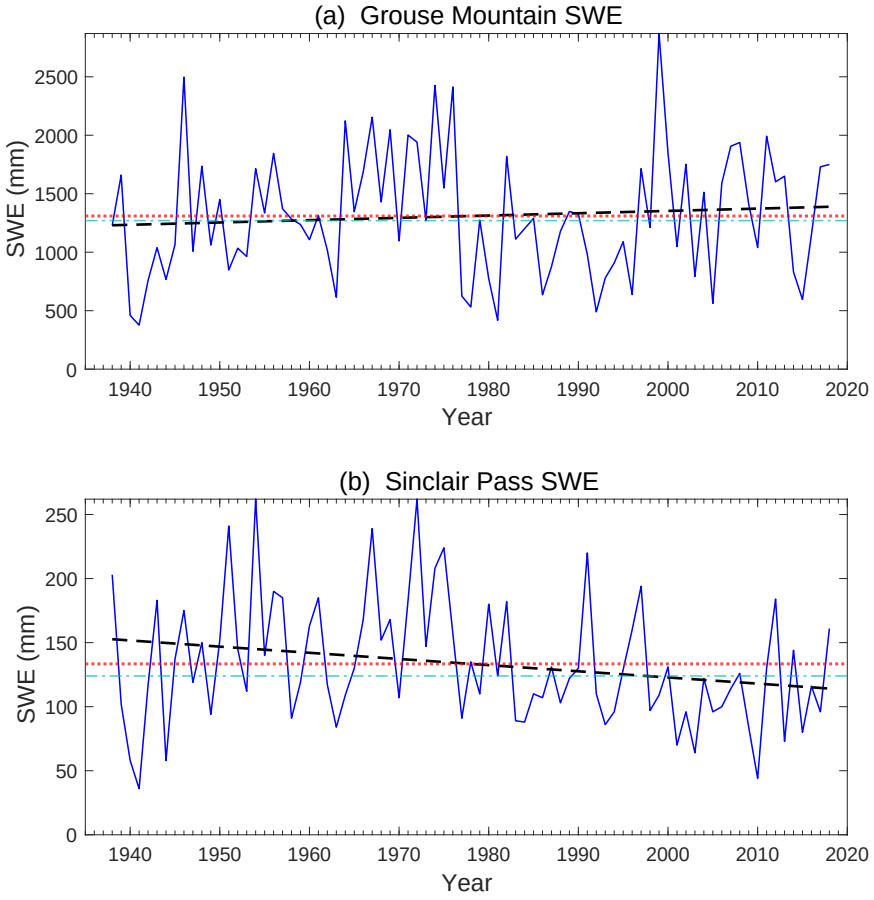


Figure 4.7 Maximum winter snow water equivalent at (a) Grouse Mountain and (b) Sinclair Pass, BC from 1938–2018, with linear trend (dashed line), mean (dotted) and median (dot-dashed). Sinclair Pass (50° 40′ N, 117° 58′ W, 1,370 m elevation) is located just west of the Canadian Rockies, while Grouse Mountain (49° 23′ N, 123° 05′ W, 1,100 m elevation) is located near the west coast (hence the much large winter SWE values). [Data source: River Forecast Centre, British Columbia.]

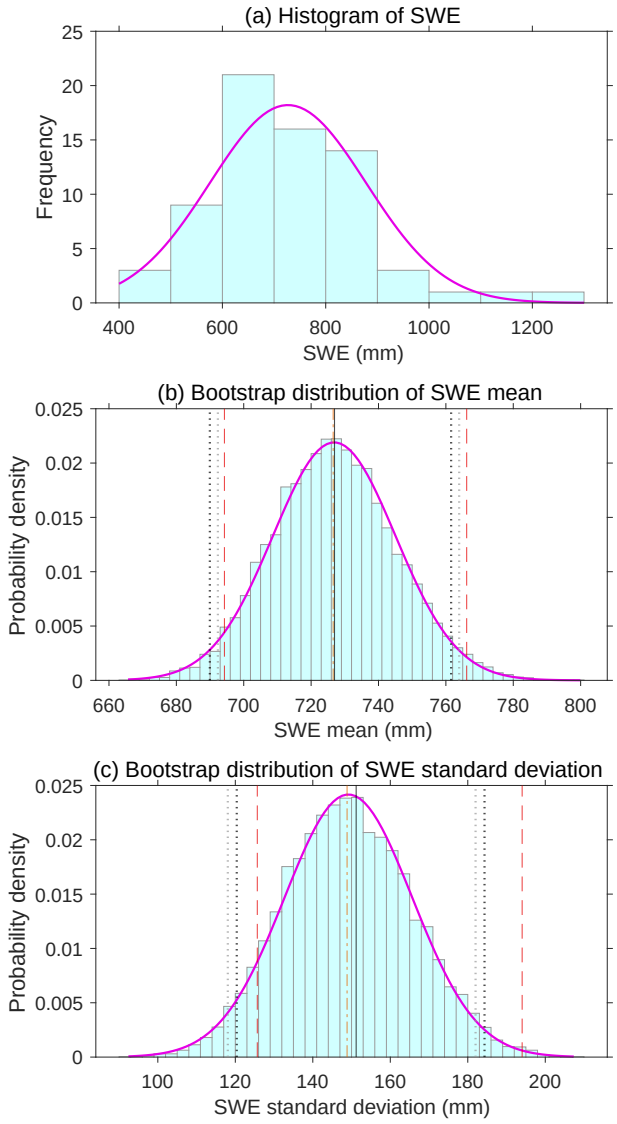


Figure 4.8 (a) Histogram of the maximum winter SWE at Glacier, BC. Distribution of the SWE (b) sample mean and (c) sample standard deviation from 10,000 bootstrap samples. The fitted Gaussian curve is also shown. The vertical lines show the statistic from the original sample (solid), the median of the bootstrap statistic (dot-dashed), the 95% CI from the BCa method (dashed), the basic method (darkly dotted) and the percentile method (lightly dotted). [Data source: River Forecast Centre, British Columbia.]

Chapter 5: Linear regression

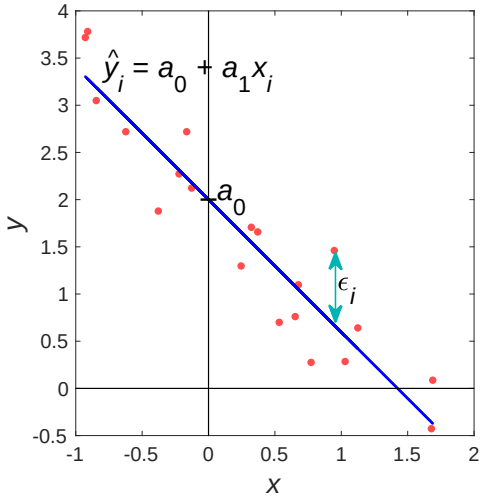


Figure 5.1 Illustrating linear regression. A straight line $\hat{y}_i = a_0 + a_1 x_i$ is fitted to the data, where the parameters a_0 and a_1 are determined from minimizing the sum of the square of the error ϵ_i, which is the vertical distance between the ith data point and the line. The slope of the line is given by a_1 and the y-intercept by a_0.

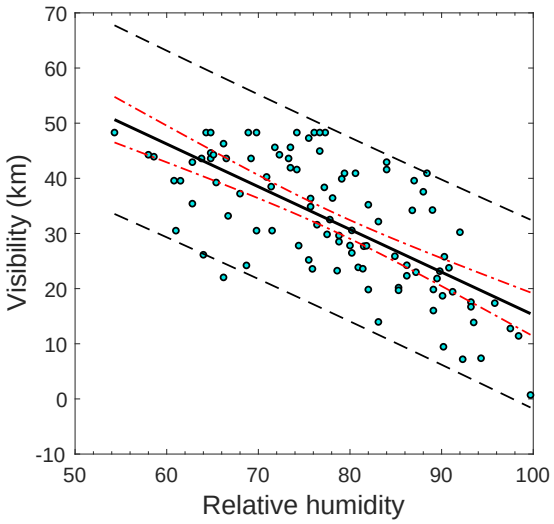


Figure 5.2 Simple linear regression with visibility as the response variable and relative humidity as predictor. The 95% predictor intervals are indicated by the dashed lines and the 95% confidence intervals by the dot-dashed lines. [Data source: weatherstats.ca based on Environment and Climate Change Canada data.]

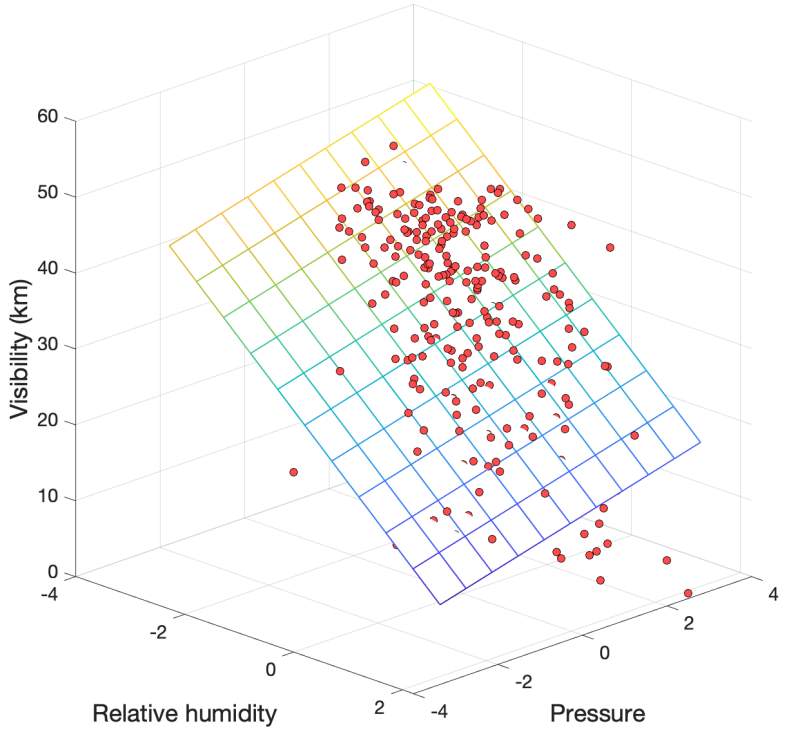


Figure 5.3 Multiple linear regression for daily weather variables in Vancouver, BC, where visibility is the response variable y and relative humility and pressure are the two standardized predictors. The MLR predicted values $\hat{y}_i$ lie on a two-dimensional plane as indicated by the grid, with the observed y_i values indicated by the circles. The grid is tilted downward as relative humidity increases and upward as pressure increases, as expected from the regression parameters $\hat{a}_1 = -7.03$ and $\hat{a}_2 = 1.54$. The vertical distance between a data point $(\mathbf{x}_i, y_i)$ and its projected value $(\mathbf{x}_i, \hat{y}_i)$ on the plane is the error ϵ_i. When there are m predictors in the regression relation, $(\mathbf{x}_i, \hat{y}_i)$ lies on an m-dimensional hyperplane. [Data source: weatherstats.ca based on Environment and Climate Change Canada data.]

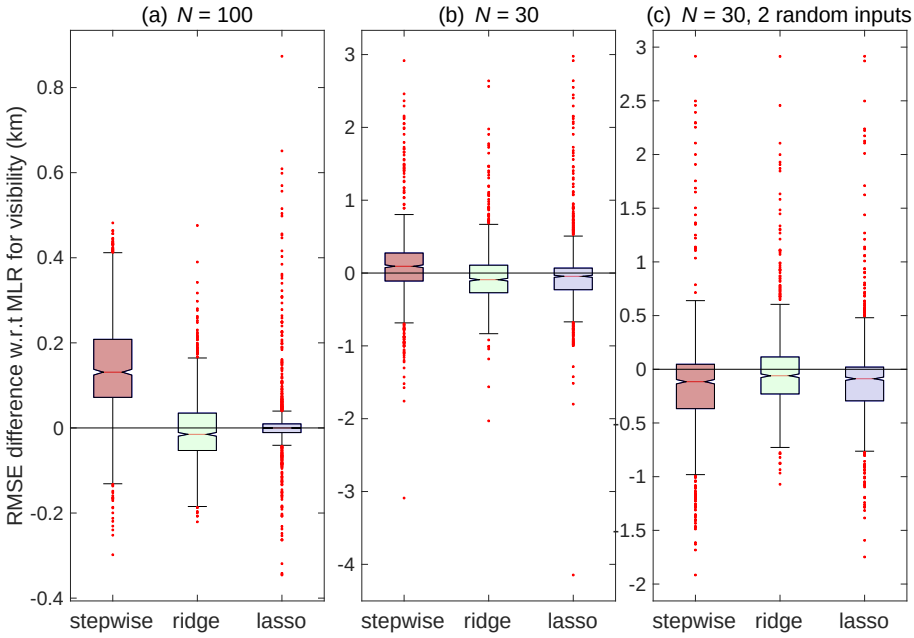


Figure 5.4 Boxplot showing the RMSE difference with respect to the MLR model, that is, the RMSE of stepwise regression, ridge regression and lasso minus the RMSE of the MLR model, for 1,000 trials. Visibility in Vancouver, BC is the response variable, while humidity, pressure, air temperature and wind speed are the four predictors. The sample size of the training data was (a) $N = 100$, (b) $N = 30$ and (c) $N = 30$. In (c), the third and fourth predictors were replaced by random numbers from a standard Gaussian distribution. The RMSE was computed on test data, that is, data not chosen for model training. A positive RMSE difference means the model is underperforming the MLR. See Section 2.12.3 for an explanation of boxplots. [Data source: weatherstats.ca based on Environment and Climate Change Canada data.]

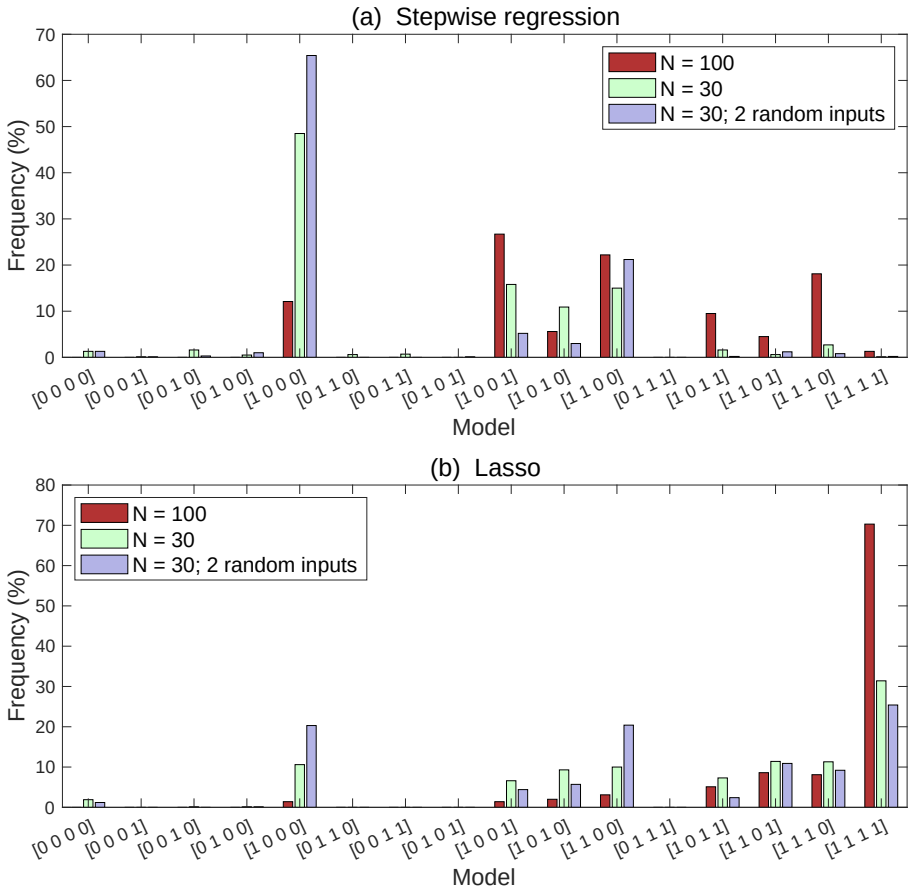


Figure 5.5 Histogram showing the percentage distribution of models selected by (a) stepwise regression and (b) lasso from 1,000 trials with visibility as the response variable. With four predictors, there are 16 possible model architectures, for example, model [1 0 0 0] indicates only the first predictor was used and [1 1 1 1] indicates that all four predictors were used.

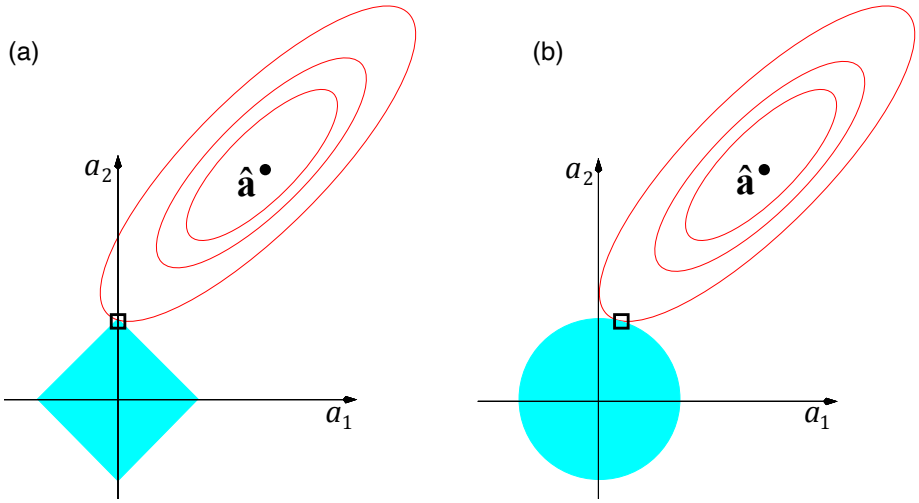


Figure 5.6 Schematic diagram illustrating the SSE contours (ellipses) and the constraint region for (a) lasso (diamond region) and (b) ridge regression (circular region) in a two-dimensional regression parameter space a_1-a_2. The solution is indicated by the small square, marking where the lowest value of the SSE function intercepts the constraint region. The dot in the centre of the ellipses marks $\hat{\mathbf{a}}$, where the minimum of the SSE occurs. [Adapted from Hastie, Tibshirani, et al. (2009, figure 3.11), which was based on Tibshirani (1996, figure 2).]

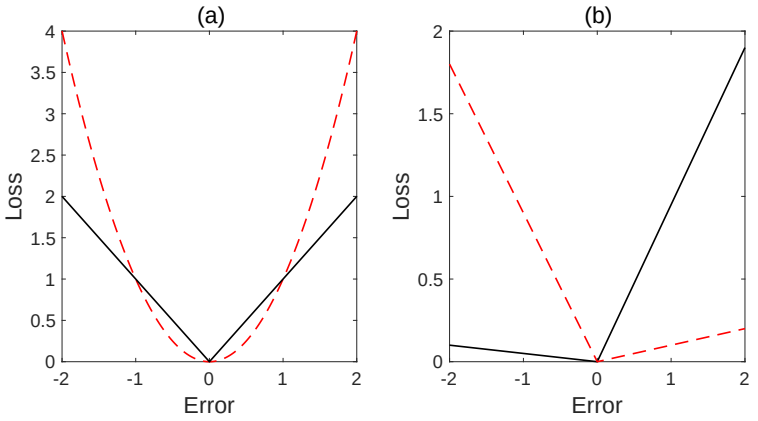


Figure 5.7 The loss L as a function of the error ϵ using (a) absolute errors (solid curve) for finding the conditional median and squared errors (dashed curve) for the conditional mean, and (b) for finding the conditional 0.95 quantile (solid) and the conditional 0.10 quantile (dashed).

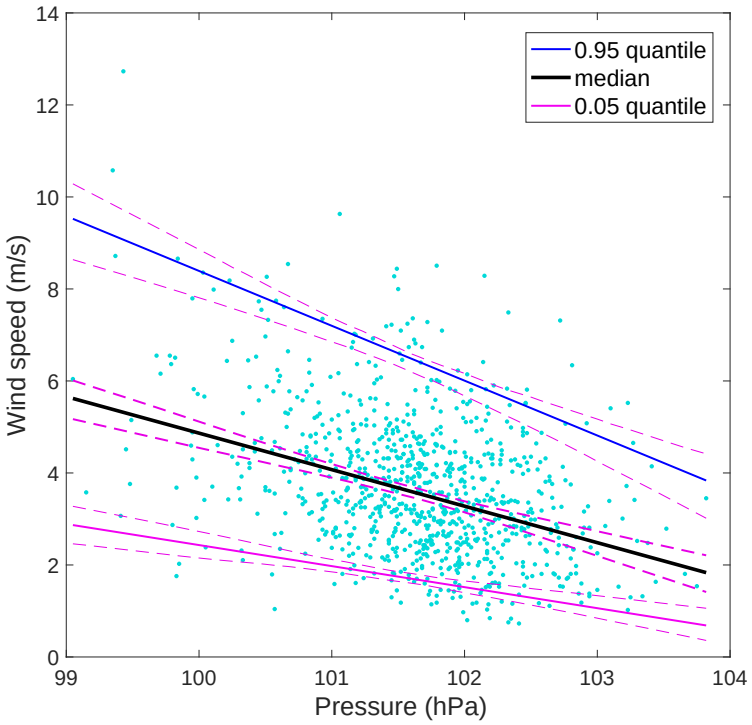


Figure 5.8 Quantile regression with wind speed as response and pressure as predictor, with data from Vancouver, BC. Bootstrap resampling was used to estimate the 95% confidence intervals (dashed lines) around the regression lines. [Data source: weatherstats.ca based on Environment and Climate Change Canada data.]

Chapter 6: Neural networks

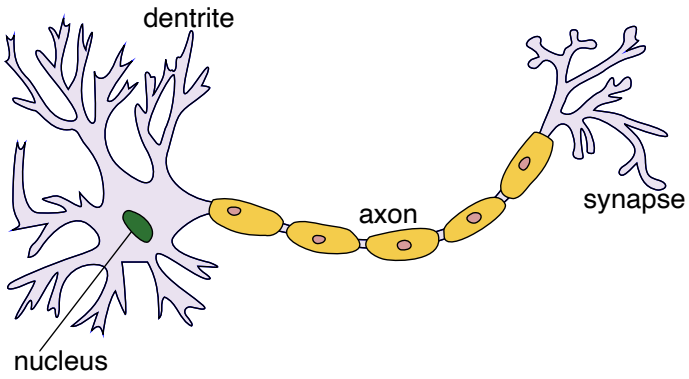


Figure 6.1 In a neuron, dendrites receive signals from other neurons. If the total stimulus exceeds some threshold, the neuron becomes activated, firing a signal down its axon to the synapses at its end. Synapses are sites where neurotransmitting chemicals are released into the space between neurons so the signal can be picked up by neighbouring neurons. [Image source: Quasar Jarosz at English Wikipedia.]

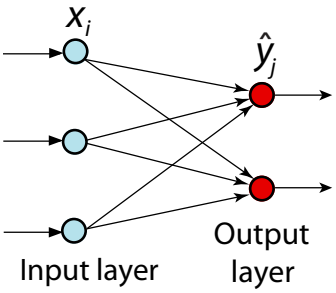


Figure 6.2 The perceptron model consists of a layer of input neurons connected directly to a layer of output neurons.

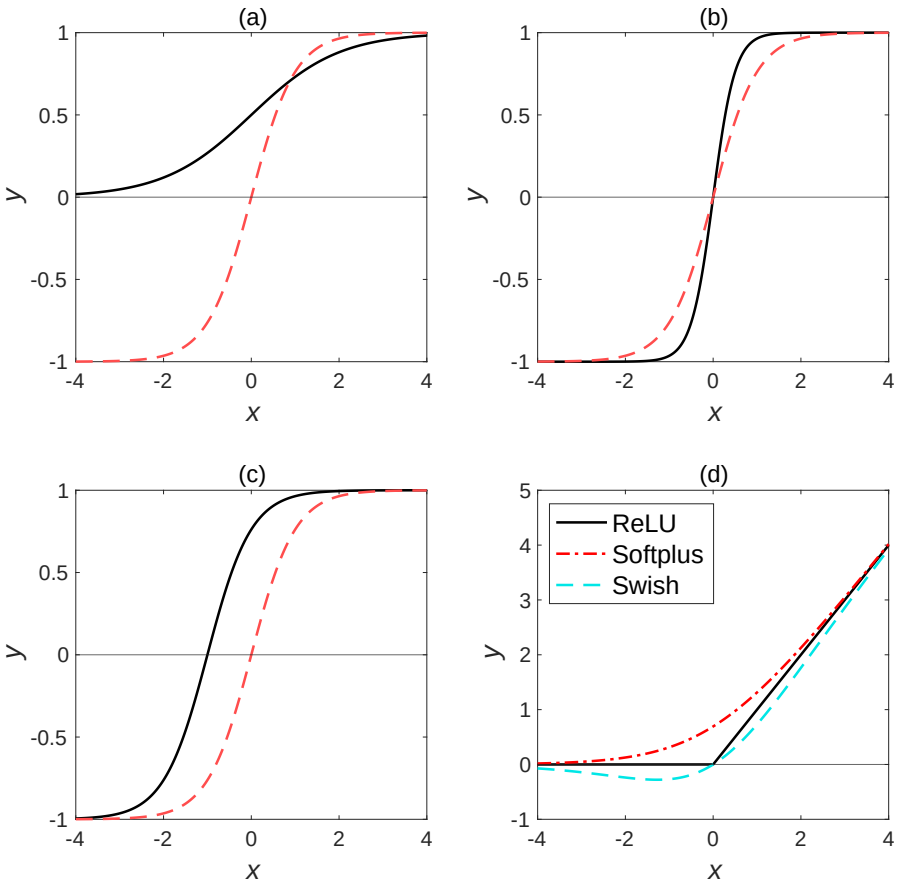
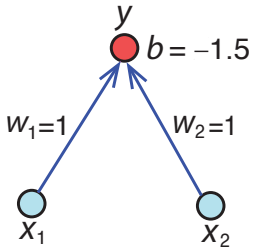


Figure 6.3 (a) The logistic sigmoidal function (solid) and the hyperbolic tangent function tanh (dashed). (b) Effect of the weight w in $f(wx)$ as seen by comparing $f(2x)$ (solid) and $f(x)$ (dashed), with tanh used for f. (c) Effect of the offset parameter b in $f(wx+b)$ by comparing $f(x+1)$ (solid) and $f(x)$ (dashed). (d) Three unbounded activation functions: the rectified linear unit (ReLU) function $f(x) = \max(0, x)$, the softplus function $f(x) = \log(1+\mathrm{e}^x)$ and the swish function given in (15.2).

Figure 6.4 The perceptron model for computing $y = x_1.\mathrm{AND}.x_2$. The activation function f used is the Heaviside step function H in (6.2). The threshold $-b = 1.5$ is exceeded by $w_1x_1 + w_2x_2$ in $H(w_1x_1+w_2x_2+b)$ only when both inputs have the value 1.



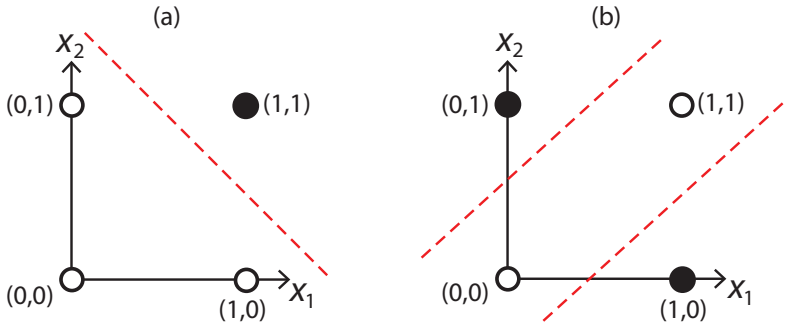


Figure 6.5 The classification of the input data (x_1, x_2) by the Boolean logical operator (a) AND and (b) XOR (exclusive OR). In (a), the decision boundary separating the TRUE domain (black circle) from the FALSE domain (white circles) can be represented by a straight (dashed) line, hence the problem is linearly separable, whereas in (b), two lines are needed, rendering the problem not linearly separable.

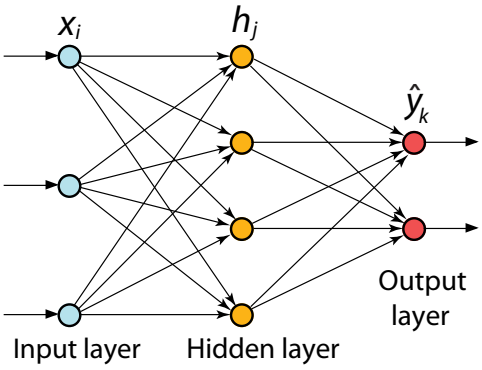


Figure 6.6 The multi-layer perceptron (MLP) or feed-forward neural network (FFNN) model with one 'hidden layer' of neurons or nodes sandwiched between the input layer and the output layer. There are m_1 nodes in the input layer, m_2 nodes in the hidden layer and m_3 nodes in the output layer.

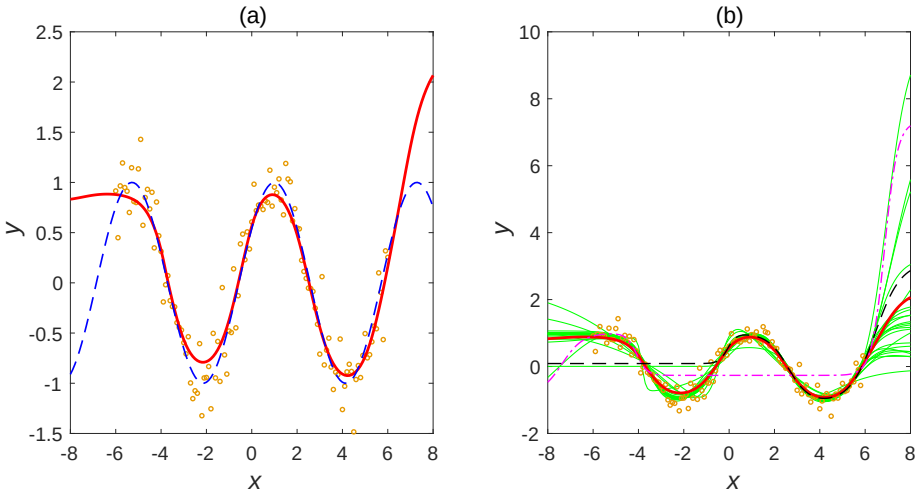


Figure 6.7 (a) The ensemble average (solid curve) of an MLP ensemble with 25 members, the true signal (dashed) and the training data (circles). (b) The 25 individual ensemble members are shown by thin curves in addition to the ensemble average (solid curve), with two of the individual members (corresponding to shallow local minima in the objective function) highlighted by the dashed curve and the dot-dash curve. Different vertical scales are used in (a) and (b).

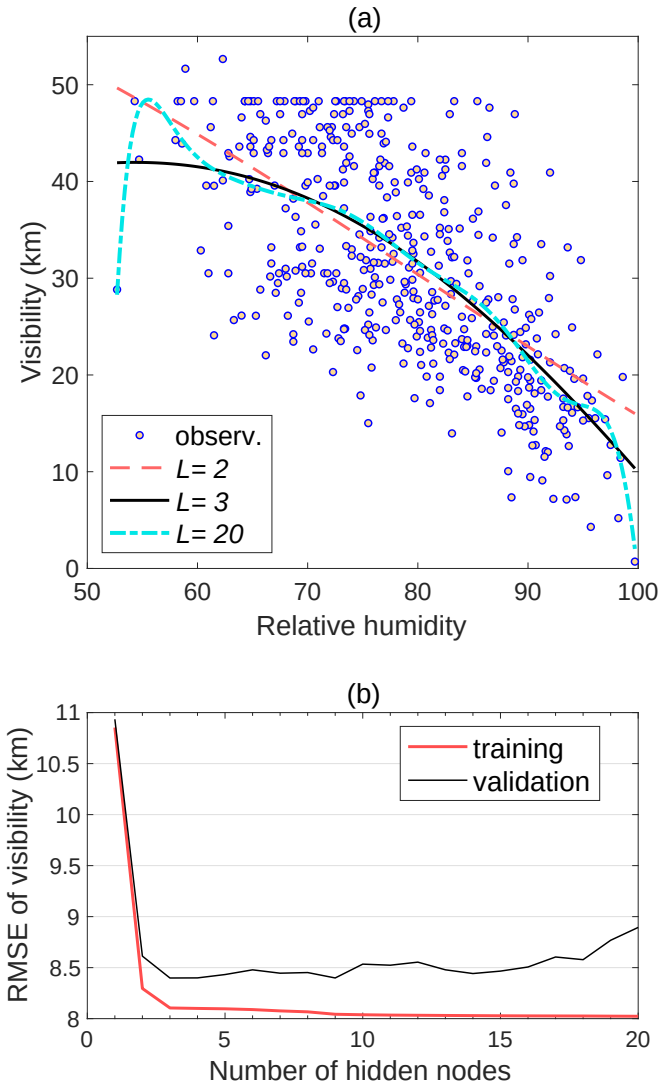


Figure 6.8 The relative humidity–visibility data from Figure 5.2 (but with 457 instead of 100 data points chosen) are used to train an ensemble of 100 ELM models (using the logistic activation function in the hidden layer), with the ensemble averaged output in (a) shown for $L = 2$, 3 and 20 hidden nodes, with underfitting seen when $L = 2$ and overfitting when $L = 20$. (b) With five-fold cross-validation (see Section 8.5), the RMSE of the validation data (dashed curve) bottoms at $L = 3$ hidden nodes. RMSE of the training data (solid curve) keeps on decreasing as L increases, as the model fits closer to the noisy data. [Data source: weatherstats.ca based on Environment and Climate Change Canada data.]

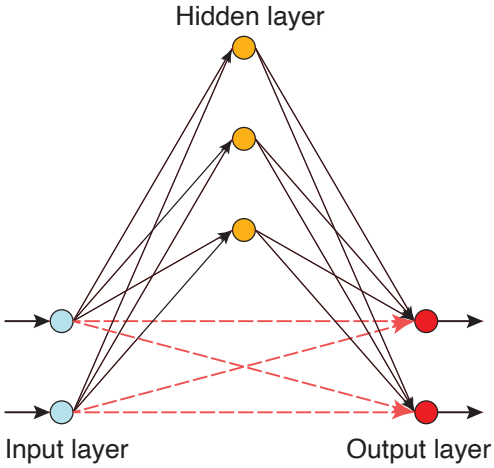


Figure 6.9 The random vector functional link (RVFL) model is similar in architecture to an MLP model except direct linear mapping from the input layer to the output layer is allowed (dashed arrows). In this example, the RVFL model has two inputs, three hidden nodes and two outputs.

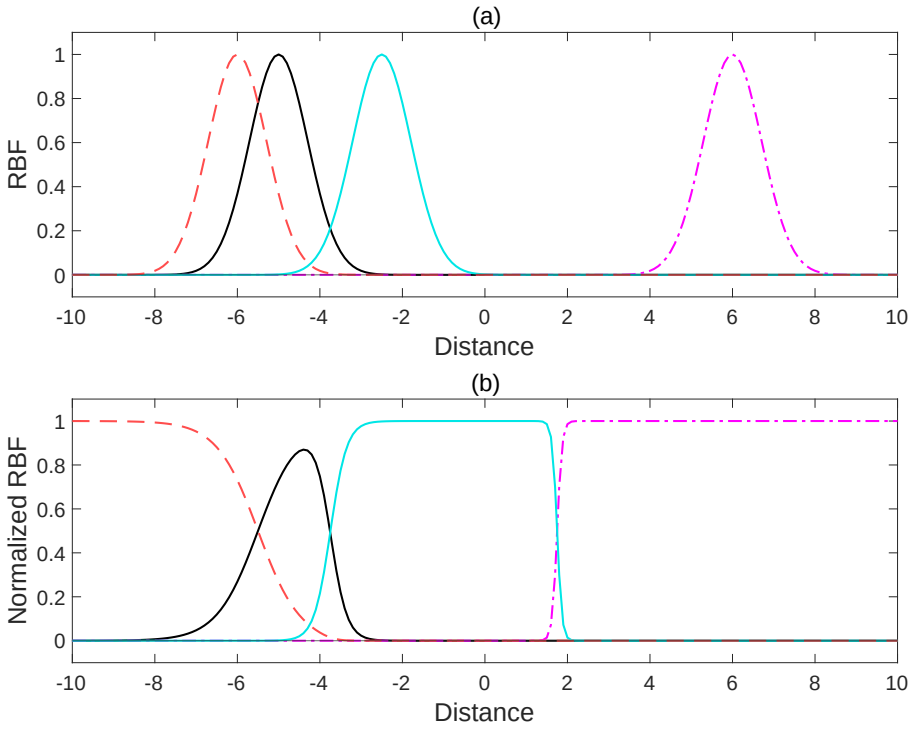


Figure 6.10 (a) Radial basis functions (RBFs) and (b) normalized RBFs. Holes are present in (a), where RBFs with fixed width σ are used. This problem is avoided in (b) with the normalized RBFs.

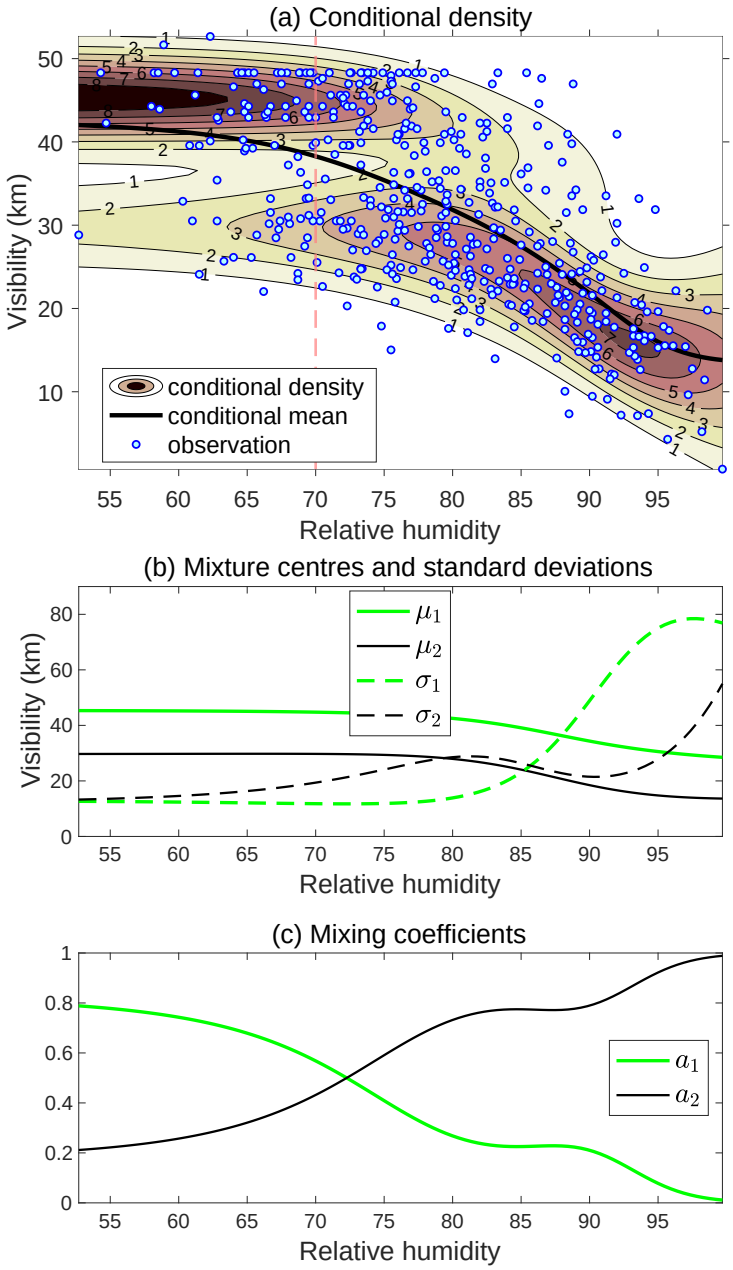


Figure 6.11 (a) Mixing density network (MDN) applied to the same dataset used in Figure 6.8(a), with contours showing the conditional density $p(y|x)$ and the solid curve the conditional mean. The contour labels need to be multiplied by 100 to give the value for $p(y|x)$. Along the vertical dashed line at relative humidity = 70, the conditional density shows two peaks (at visibility around 30 km and 45 km). (b) The centres μ_1 and μ_2 and the standard deviations σ_1 and σ_2 for the two Gaussian functions in the mixture model and (c) the mixing coefficients a_1 and a_2.

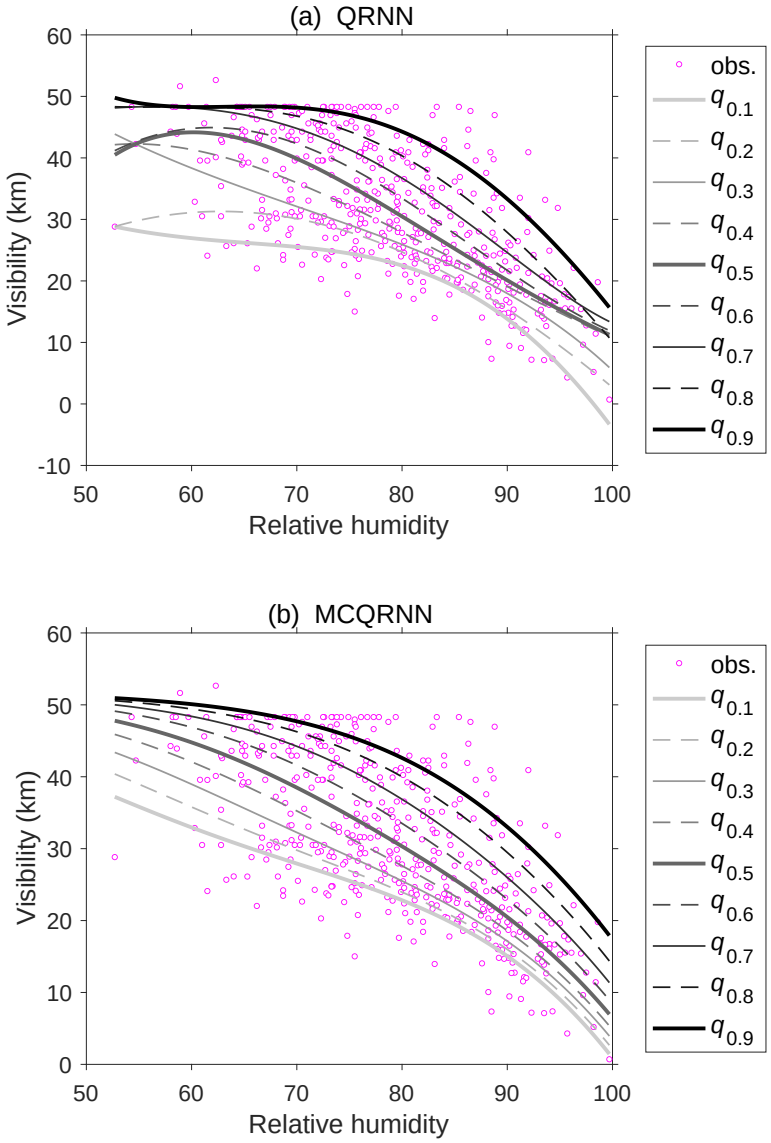


Figure 6.12 Non-linear quantile regression by neural network models as applied to the same dataset as in Figure 6.11, using (a) the basic QRNN model and (b) the MCQRNN model. The observations (circles) and the 0.1, 0.2, ..., 0.9 quantiles are shown. Crossing of quantile curves is seen in (a) but not in (b).

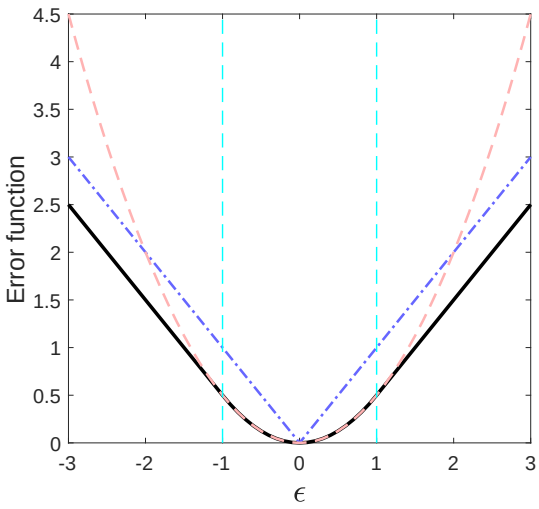


Figure 6.13 The Huber error function $h(\epsilon)$ (with parameter $\delta = 1$) (solid curve) plotted versus the error ϵ. The squared error function (dashed) and the absolute error function (dot-dashed) are also shown for comparison. The vertical dashed lines at $\epsilon = \pm\delta = \pm 1$ indicate where the Huber function changes from behaving like the squared error function to behaving like the absolute error function, which is less sensitive to outliers.

Chapter 7: Non-linear optimization

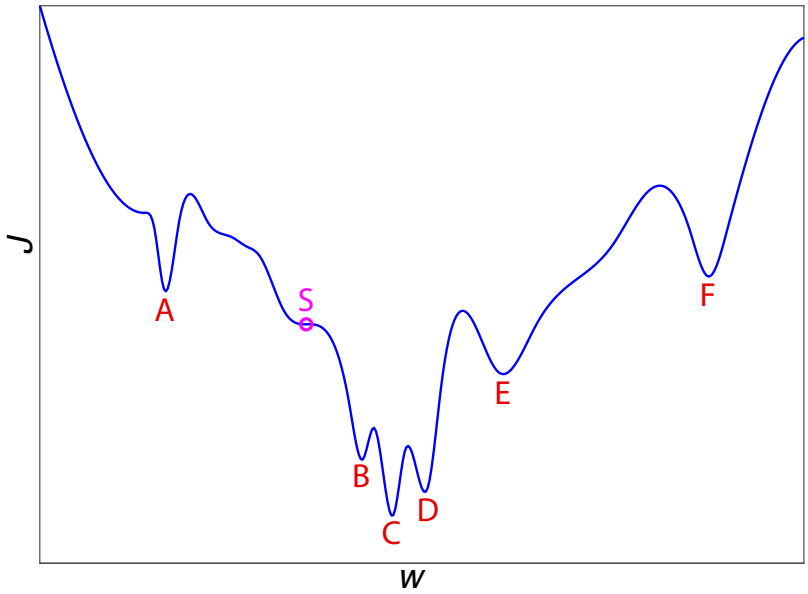


Figure 7.1 A schematic diagram illustrating an objective function $J(w)$. There are multiple local minima labelled A, B, D, E and F, and a global minimum labelled C. Local minima B and D, being close to C, are likely to give good solutions, whereas the shallow minima A and F, poor solutions. The point labelled S is a saddle point – the gradient (i.e. slope) of the curve is zero but the point is neither a maximum nor a minimum.

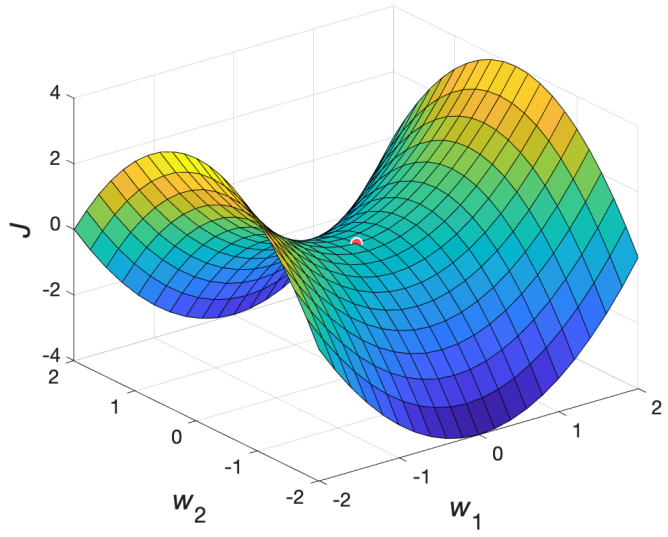


Figure 7.2 A saddle point on the surface $J = w_1^2 - w_2^2$, with the saddle point at $(0, 0)$ marked by a semi-circle. At $(0, 0)$, J concaves up (i.e. positive curvature) along the w_1 dimension and concaves down (negative curvature) along the w_2 dimension.

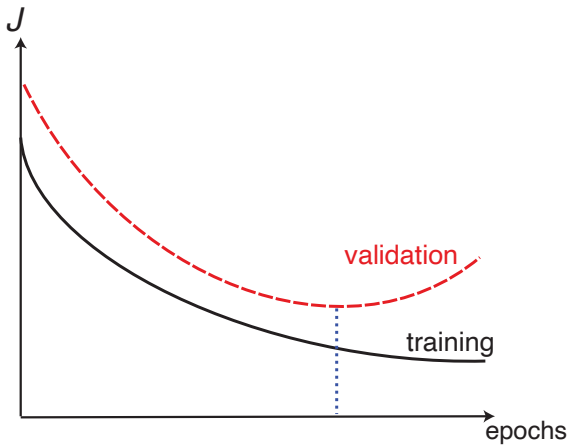
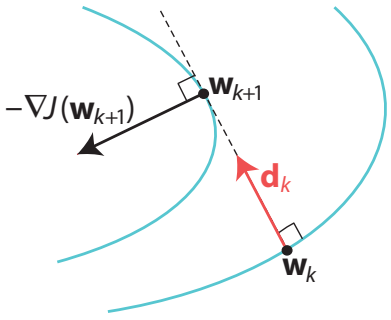


Figure 7.3 Schematic diagram illustrating the behaviour of the objective function J as the number of training epochs increases. Evaluated over the training data, the objective function (solid curve) decreases with increasing number of epochs; however, evaluated over an independent set of validation data, the objective function (dashed curve) initially drops but eventually rises with increasing number of epochs, indicating that overfitting has occurred when a large number of training epochs is used. The minimum in the objective function evaluated over the validation data (as marked by the vertical dotted line) indicates when training should be stopped to avoid overfitting.

Figure 7.4 The gradient descent approach starts from the weights $\mathbf{w}_k$ estimated at step k of an iterative optimization process. The descent path $\mathbf{d}_k$ is chosen along the negative gradient of the objective function J, which is the steepest descent direction. Note that $\mathbf{d}_k$ is perpendicular to the J contour where $\mathbf{w}_k$ lies. The descent along $\mathbf{d}_k$ proceeds until it is tangential to another contour at $\mathbf{w}_{k+1}$, which is the minimum of J along the $\mathbf{d}_k$ direction, thereby giving the optimal step size η in the descend along $\mathbf{d}_k$. At $\mathbf{w}_{k+1}$, the direction of steepest descent is given by $-\nabla J(\mathbf{w}_{k+1})$. The process is iterated.



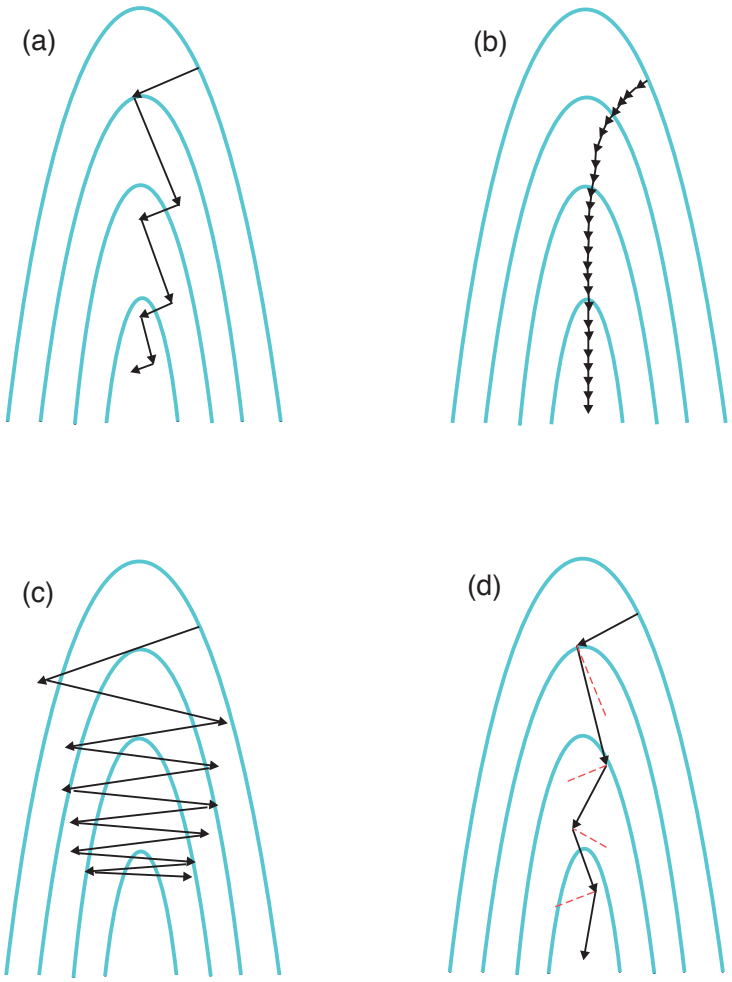


Figure 7.5 The gradient descent method with (a) line minimization (i.e. optimal step size η), (b) a fixed step size that is too small, (c) a fixed step size that is too large and (d) momentum, which reduces the zigzag behaviour during descent. The direction of steepest descent is indicated by dashed lines in (d). [Follows Masters (1995, chapter 1).]

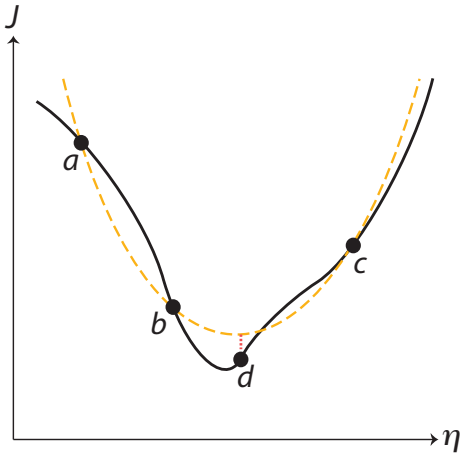


Figure 7.6 Using line search to find the minimum of the function $J(\eta)$. First, three points a, b and c are found with $J(a) > J(b)$ and $J(b) < J(c)$, so that the minimum is bracketed within the interval (a, c). Next a parabola is fitted to pass through the three points (dashed curve). The minimum of the parabola is at $\eta = d$. Next the three points among a, b, c and d with the three lowest values of J are selected, and a new parabola is fitted to the three selected points, with the process iterated until convergence to the minimum of J.

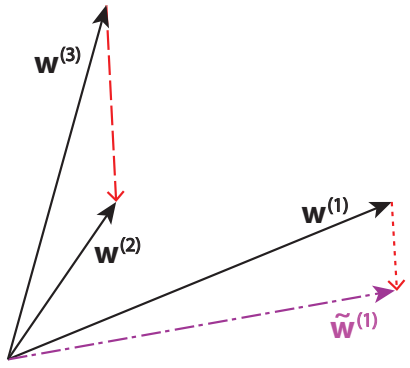


Figure 7.7 Constructing the mutant vector in DE. From three randomly chosen vectors $\mathbf{w}^{(1)}$, $\mathbf{w}^{(2)}$ and $\mathbf{w}^{(3)}$ from the population, the difference vector $\mathbf{w}^{(2)} - \mathbf{w}^{(3)}$ (dashed) is constructed. A scaled version of this difference vector (dotted) is added to $\mathbf{w}^{(1)}$ to give the mutant vector $\tilde{\mathbf{w}}^{(1)}$ (dot-dashed).

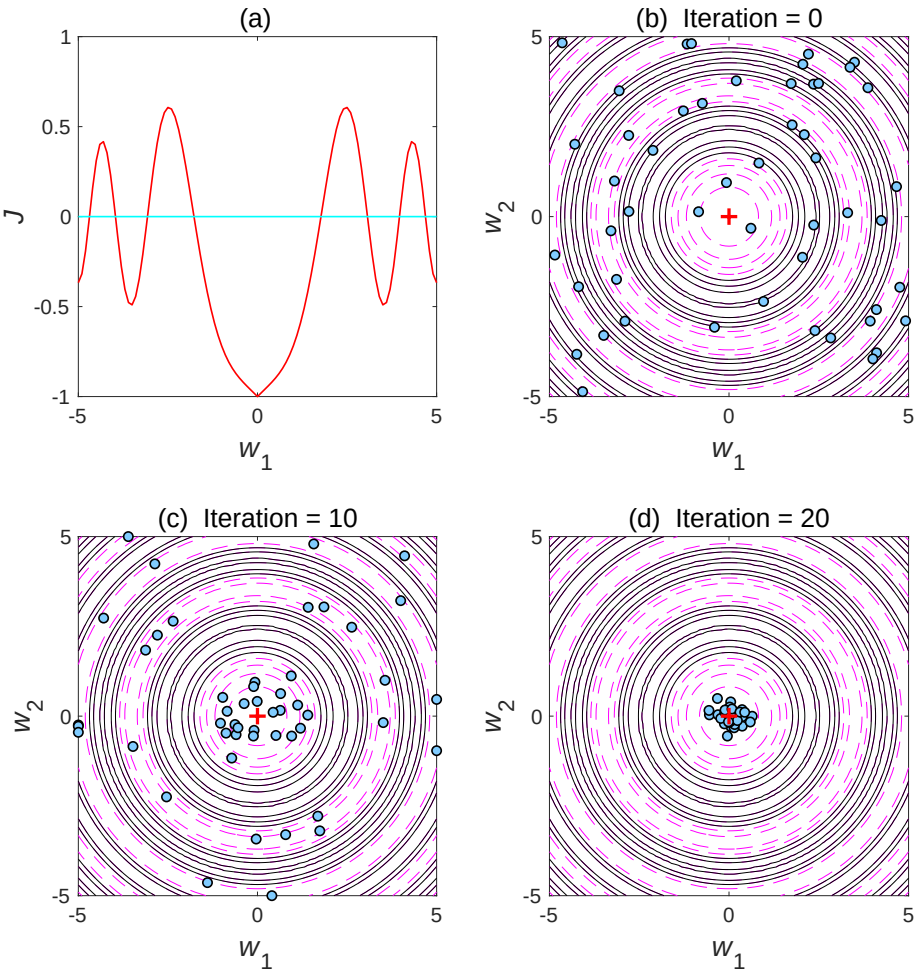


Figure 7.8 (a) The objective function J along the w_1 axis. Position of the 50 candidate solutions in the w_1-w_2 space: (b) at the initial setup and after using the DE algorithm to perform (c) 10 iterations and (d) 20 iterations. If a candidate is perturbed to move beyond the boundary of the interval $[-5, 5]$ in any dimension, it is repositioned to sit right on the boundary. Negative contour values of J are indicated by dashed lines and non-negative contours by solid lines, with the global minimum at (0, 0) marked by the cross.

Chapter 8: Learning and generalization

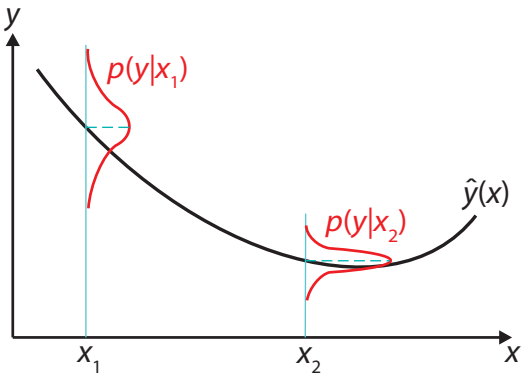


Figure 8.1 The model output $\hat{y}$ is the conditional mean (dashed line) of the target data y, with the conditional probability distribution $p(y|x)$ illustrated at x_1 and at x_2.

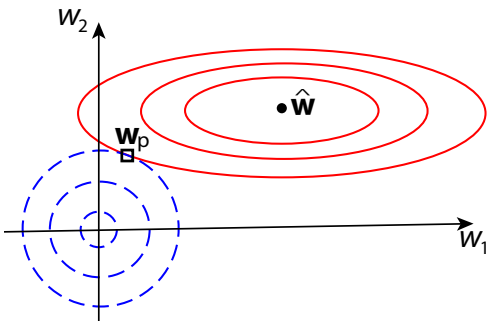


Figure 8.2 The optimal solution $\hat{\mathbf{w}}$ when there is no weight penalty in J and the optimal solution $\mathbf{w}_\mathrm{p}$ when there is weight penalty. The solid contours show J (with only the MSE term), with a minimum at $\hat{\mathbf{w}}$, while the dashed contours show the parabolic contribution from the weight penalty term. Thus, $\mathbf{w}_\mathrm{p}$ is the minimum resulting from adding the weight penalty term to the MSE term.

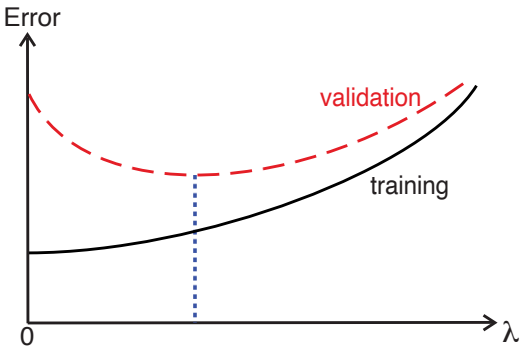


Figure 8.3 Illustrating the model error (e.g. the MSE) for the training data (solid curve) and for the validation data (dashed curve) as a function of the weight penalty parameter λ. The minimum in the dashed curve gives the optimal λ value (as marked by the dotted line).

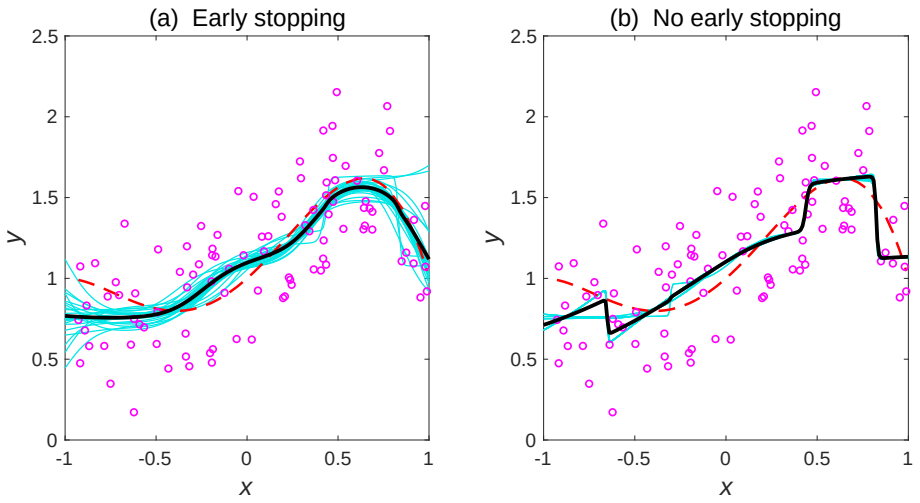
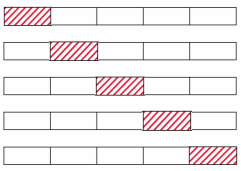


Figure 8.4 (a) The output from 25 runs of an MLP NN model trained using early stopping (thin lines), the ensemble average of the output from the 25 runs (thick line), the signal (dashed line) and the data (circles). (b) Repeat of (a) but without using early stopping during the training of the NN model.

Figure 8.5 In K-fold cross-validation (here $K = 5$), the computational loop begins by withholding the first data segment as validation data (striped pattern), using only the remaining data segments for training the model. The trained model is then used to predict the data over the validation segment. Next, the second segment is withheld as validation data and the other segments used as training data, and so on, until the final segment is used as validation data.



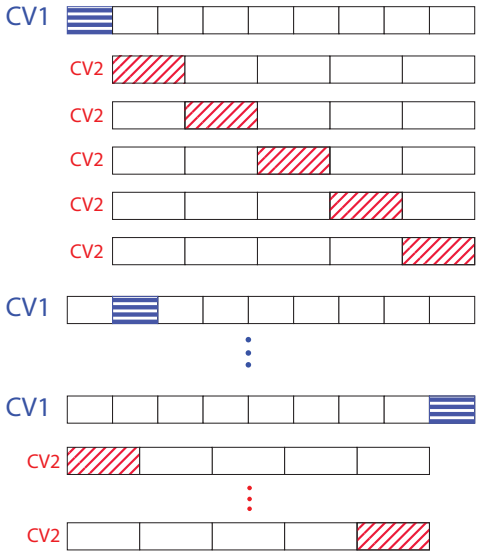


Figure 8.6 Double cross-validation involving an outer cross-validation loop CV1 and an inner loop CV2. In CV1, the data record is divided into a number of segments, and the first segment is withheld as test data (horizontally striped), while the remaining segments are used for model training. In CV2, consider only the data for model training: a standard K-fold cross-validation loop (here $K = 5$) is used where data are withheld for validation (diagonally striped) to determine the optimal model hyperparameters. CV1 continues by withholding the second segment as test data, and so on, until the final segment is used for test data.

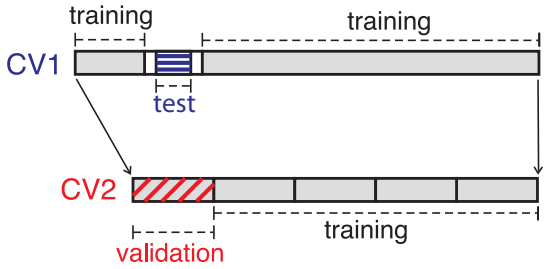


Figure 8.7 A modified double cross-validation scheme to alleviate serial correlation across the boundary between the test data and the training data. In the outer loop (CV1), the training data are shown in grey and the test data are horizontally striped. The data segments (in white) bridging the training data and the test data are not used, to avoid serial correlation leaking information from the training data to the adjacent test data. The test data segment is moved repeatedly from the start of the data record to the end in this cross-validation loop, so forecast performance is tested over the whole record. Meanwhile, in the inner loop (CV2), the training data from CV1 are assembled and divided into training and validation (diagonally striped) data segments, which are rotated under K-fold cross-validation to determine the optimal hyperparameters. The model with the optimal hyperparameters is used to predict over the test data segment in CV1. [Adapted from Zeng et al. (2011, figure A.1).]

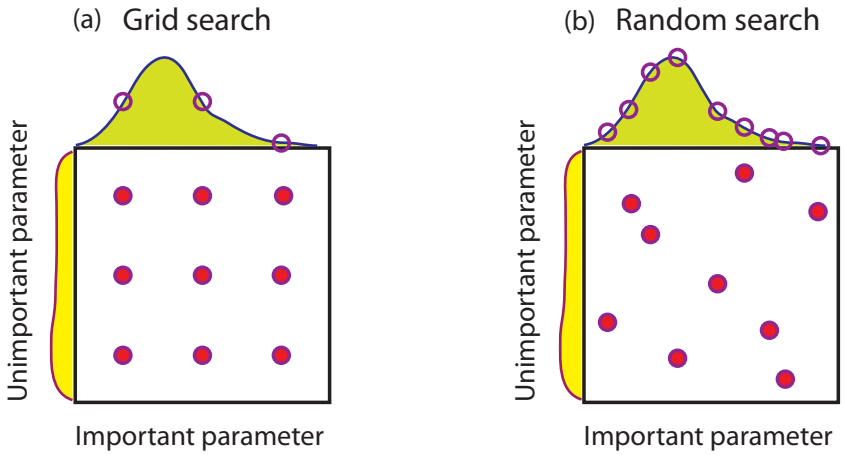


Figure 8.8 Illustrating (a) grid search and (b) random search for hyperparameters. Of the two hyperparameters, one is important and the other unimportant in influencing the objective function. The peaked curve drawn on top of the square illustrates the effectiveness of the important hyperparameter on improving the objective function, while the relatively flat curve on the left side of the square illustrates the ineffectiveness of the unimportant hyperparameter. Both (a) and (b) have nine circles within the square representing nine model runs using different values for the hyperparameters. The peak of the curve for the important hyperparameter was well located by the nine circles projected onto the curve from the nine runs in (b). In contrast, the projected circles from the nine runs in (a) failed to locate the peak of the curve due to multiple projected circles landing on the same spot. [Follows Bergstra and Bengio (2012, figure 1).]

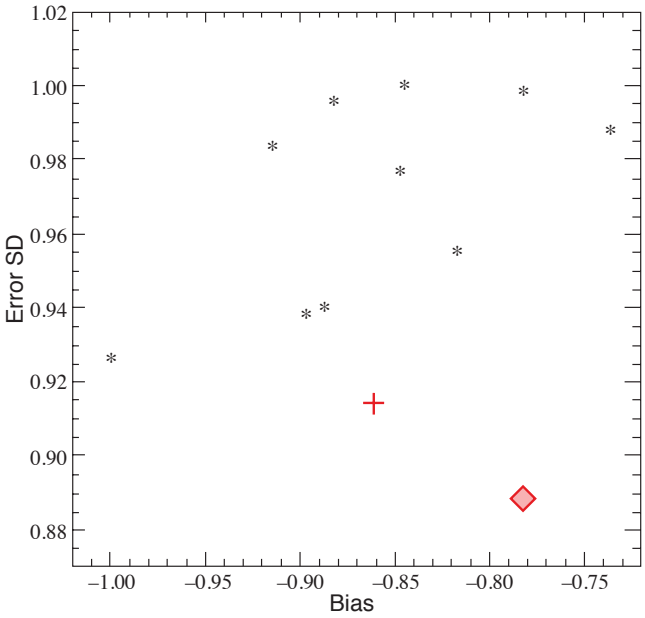


Figure 8.9 Scatter plot of model bias versus standard deviation (SD) of the model error for the 10 individual ensemble members (asterisks), the simple ensemble average (cross) and the non-linear ensemble average by NN (diamond). [Adapted from Krasnopolsky (2007, figure 7).]

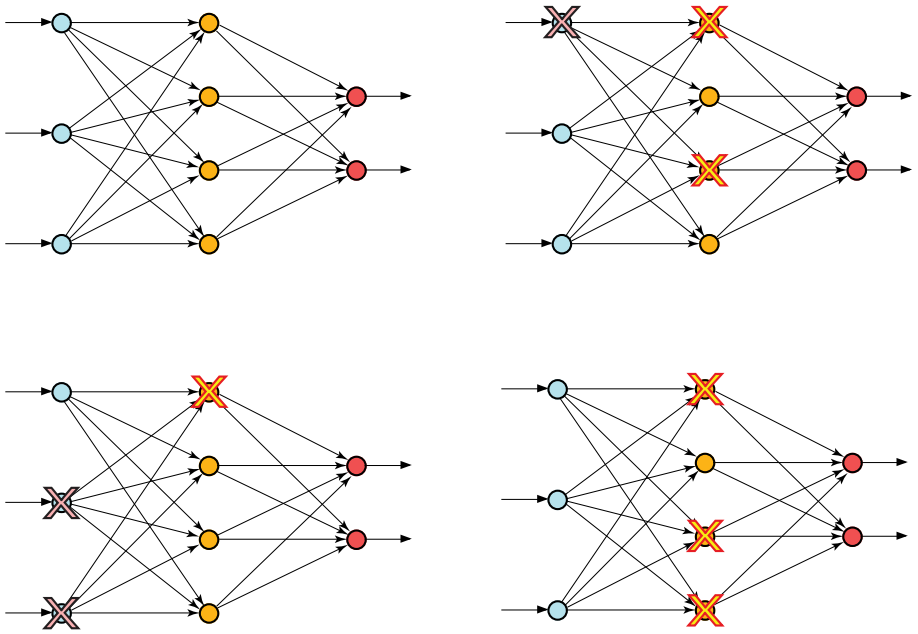
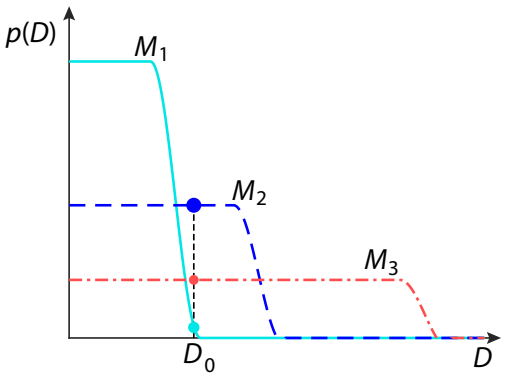


Figure 8.10 The dropout method applied to a neural network model. The original model is shown in the top left quadrant. The other three are versions of the original model but with various input and hidden nodes deleted (as marked by the crosses) during model training. The output nodes are always retained.

Figure 8.11 Schematic diagram of model selection based on the highest model evidence, with $p(D)$ shown for three models – a simple model M_1, an intermediate model M_2 and a complex model M_3. The simple model can only fit a narrow range of observed data D, while the complex model can fit a broad range. For the observed data D_0, the highest $p(D)$ (along the vertical dashed line) is found for model M_2. [Follows Bishop (2006, figure 3.13).]



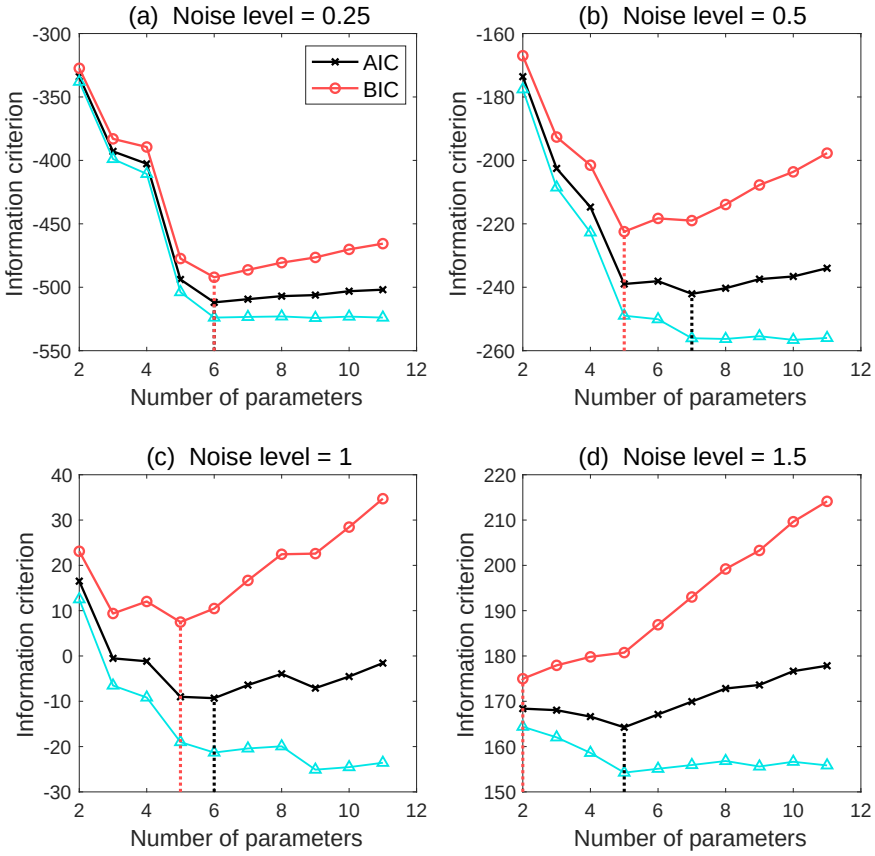


Figure 8.12 AIC (crosses) and BIC (circles) values for polynomial linear regression models of order 1 to 10 (i.e. with corresponding number of parameters from 2 to 11). The signal y_{signal} was generated by a fifth order polynomial (with six parameters). Gaussian noise with standard deviation equal to (a) 0.25, (b) 0.5, (c) 1 and (d) 1.5 times the standard deviation of y_{signal} was added to the signal. Model is selected based on lowest AIC or BIC (as indicated by the vertical dotted lines). In (a), both AIC and BIC selected the correct model with six parameters. The first term in AIC and BIC, that is, $N \log(\hat{\sigma}^2)$, is also shown (triangles).

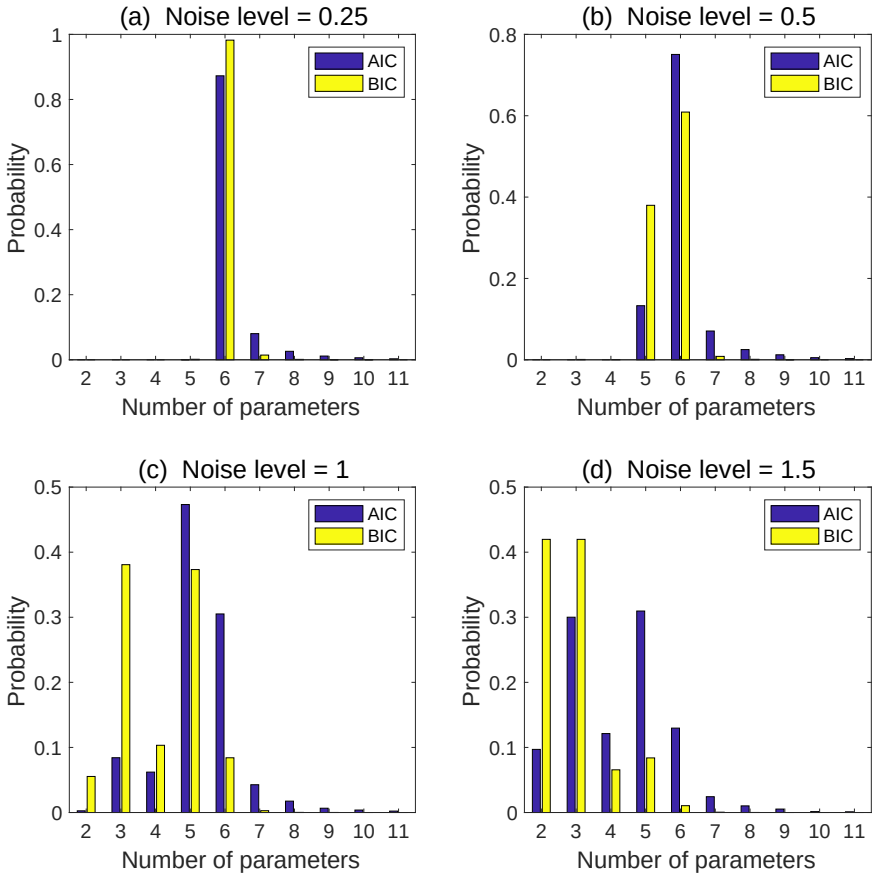


Figure 8.13 The probability of selecting a model with a certain number of parameters using AIC and BIC for four different noise levels, as estimated from 10,000 runs with different random noise. The true model had six parameters.

Chapter 9: Principal components and canonical correlation

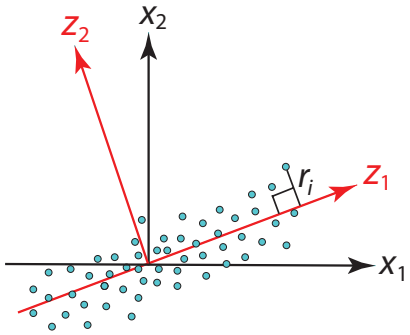


Figure 9.1 The PCA problem formulated as a minimization of the sum of r_i^2, where r_i is the shortest distance from the ith data point to the first PCA axis z_1.

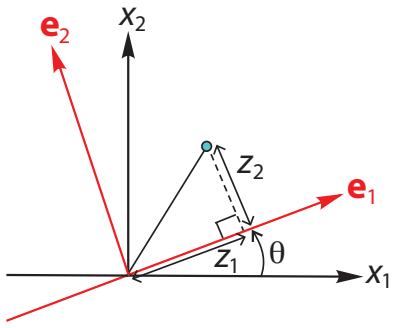


Figure 9.2 Rotation of coordinate axes by an angle θ in a two-dimensional space.

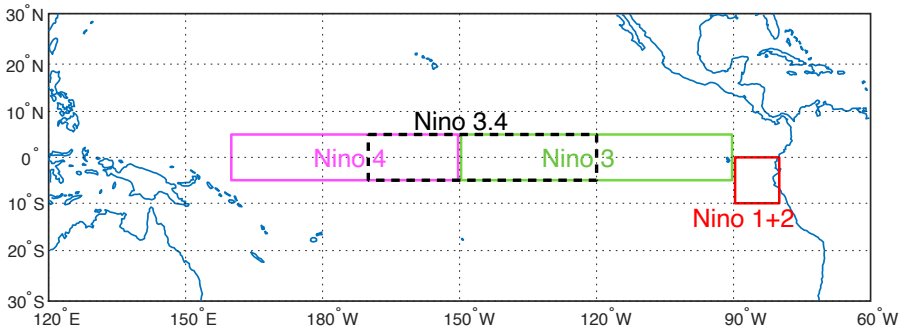


Figure 9.3 Regions of interest in the equatorial Pacific for sea surface temperature anomalies associated with the El Niño/La Niña phenomenon: Niño 1+2 (0°–10°S, 80°W–90°W), Niño 3 (5°S–5°N, 150°W–90°W), and Niño 4 (5°S–5°N, 160°E–150°W). Niño 3.4 (5°S–5°N, 170°W–120°W, marked by dashed box) straddles Niño 3 and Niño 4. SST anomalies averaged over each of these regions are used as climate indices.

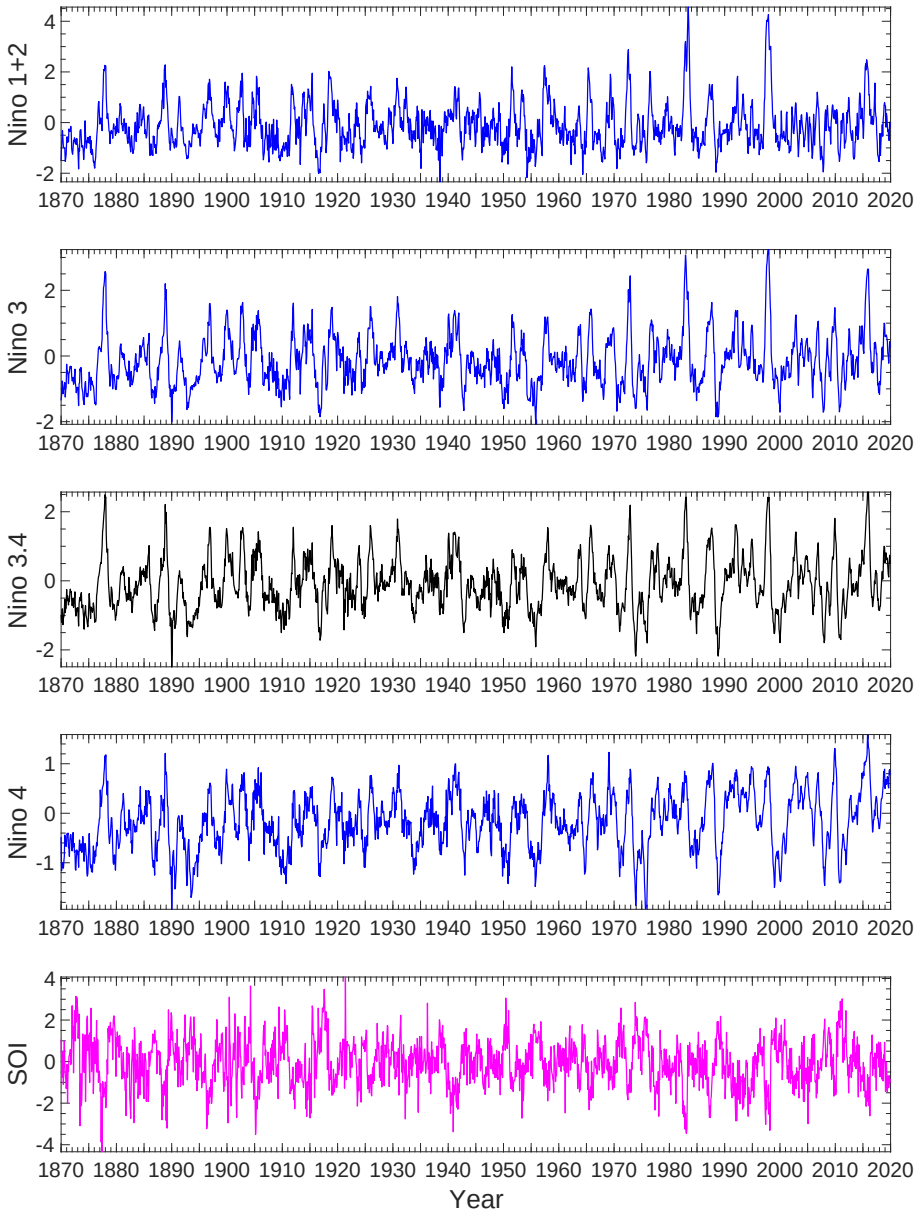


Figure 9.4 The monthly SST anomalies in Niño 1+2, Niño 3, Niño 3.4 and Niño 4 (in °C), and the monthly Southern Oscillation Index, SOI. During El Niño episodes, the SST rises in Niño 3 and Niño 3.4 (and less consistently in Niño 1+2), while the SOI drops. The reverse occurs during a La Niña episode. The grid mark for a year marks the January of that year. [Data source: Climate Research Unit, University of East Anglia.]

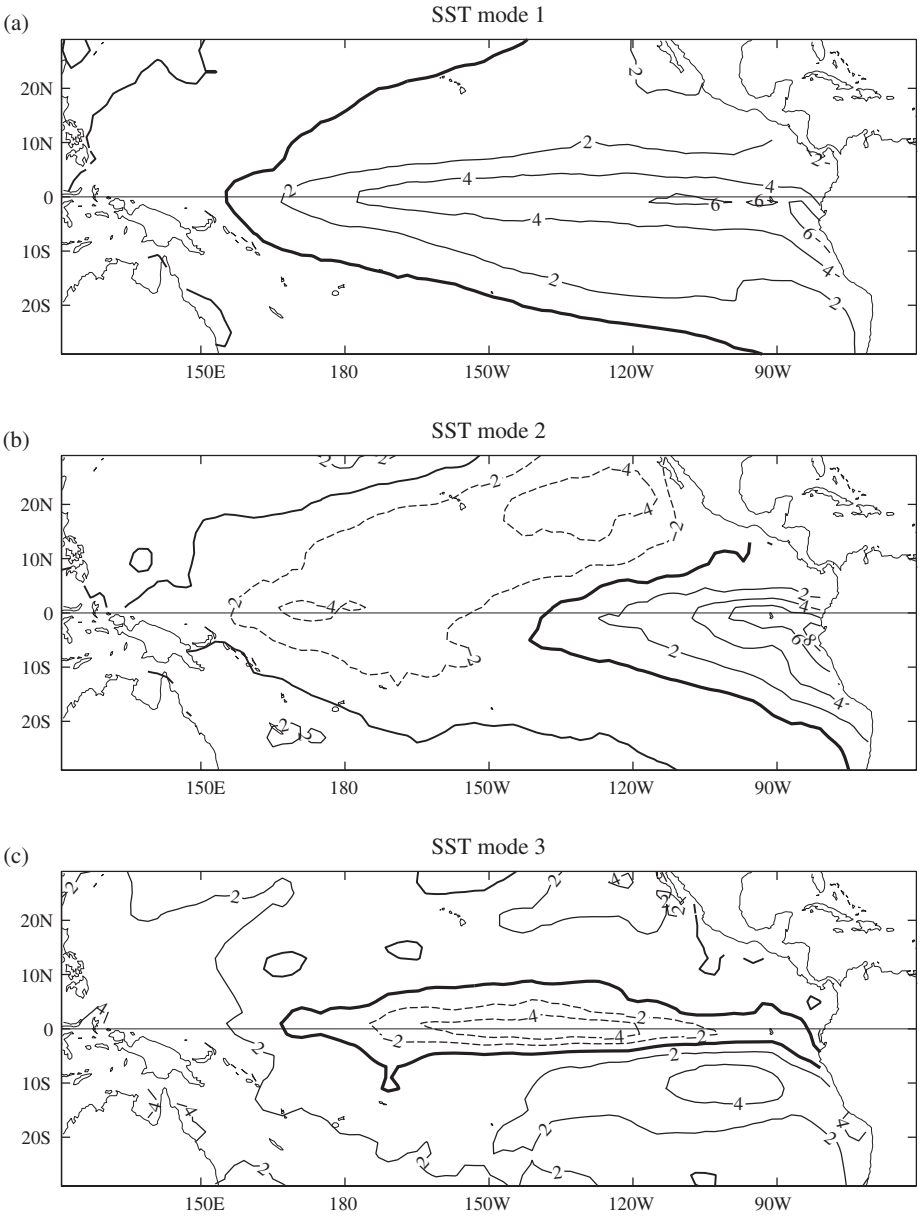


Figure 9.5 The spatial patterns (i.e. eigenvectors or EOFs) of PCA modes (a) 1, (b) 2 and (c) 3 for the SST anomalies. Positive contours are indicated by the solid curves, negative contours by dashed curves and the zero contour by the thick solid curve. The contour unit is 0.01°C. The eigenvectors have been normalized to unit norm. [Reproduced from Hsieh (2001b, figure 7).]

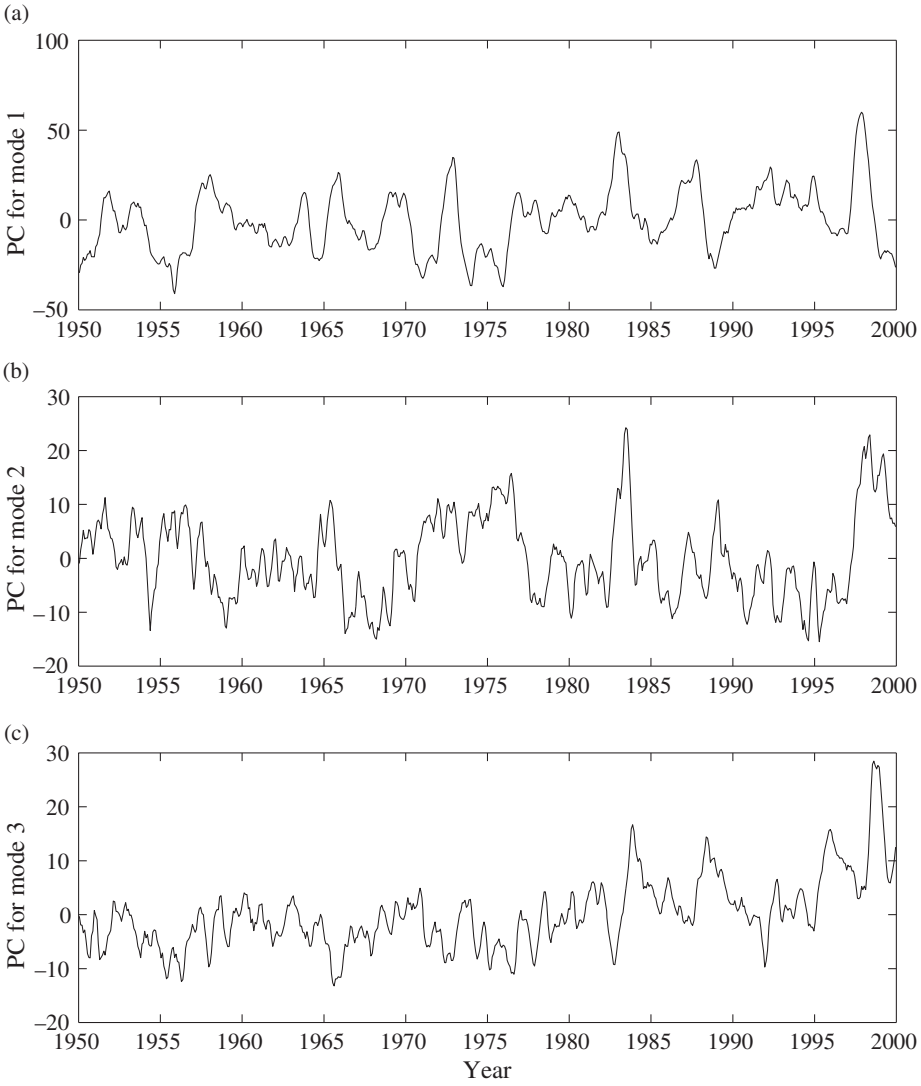


Figure 9.6 The principal component time series for the SST anomaly modes (a) 1, (b) 2 and (c) 3. [Source: Hsieh (2009)]

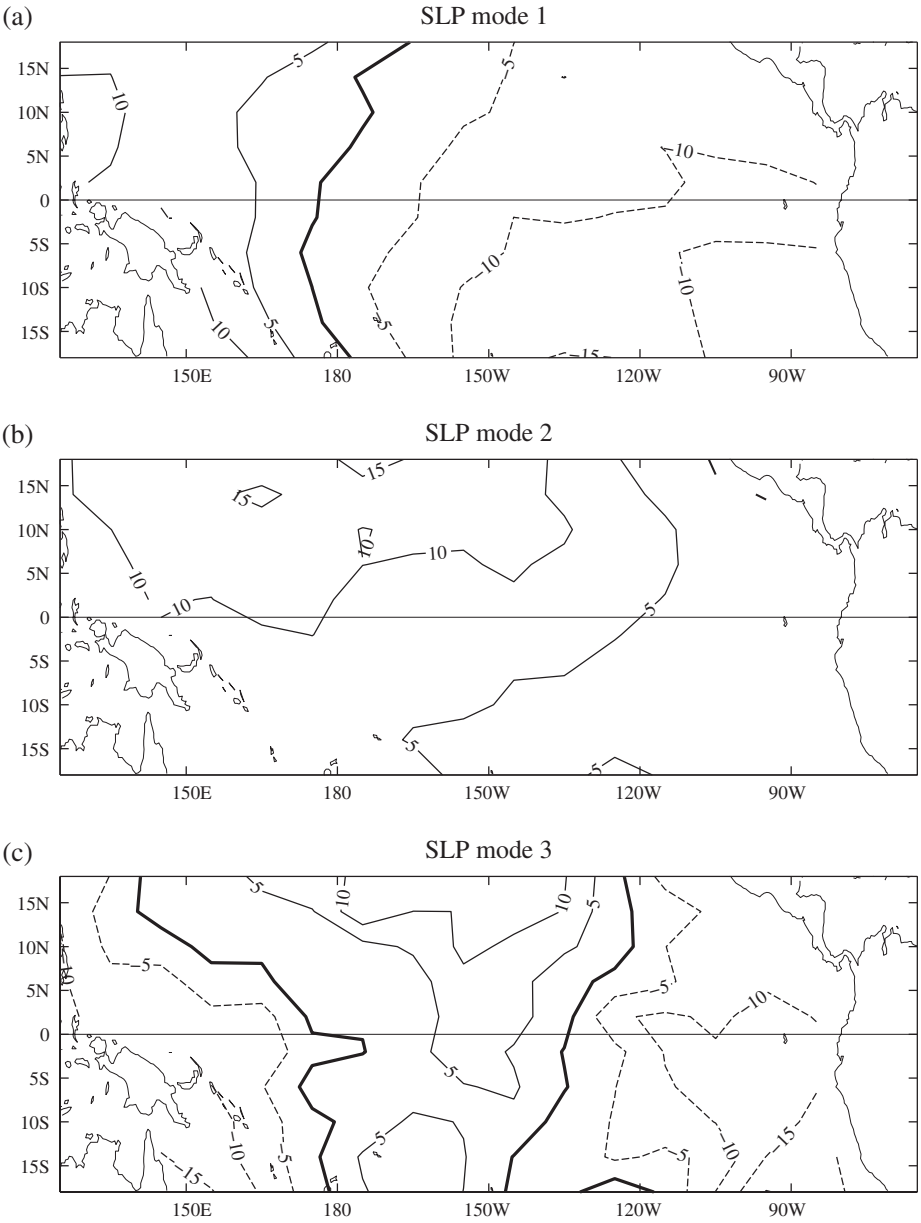


Figure 9.7 The spatial patterns of PCA modes (a) 1, (b) 2 and (c) 3 for the SLP anomalies. The contour unit is 0.01 hPa. Positive contours are indicated by the solid curves, negative contours by dashed curves and the zero contour by the thick solid curve. [Source: Hsieh (2009)]

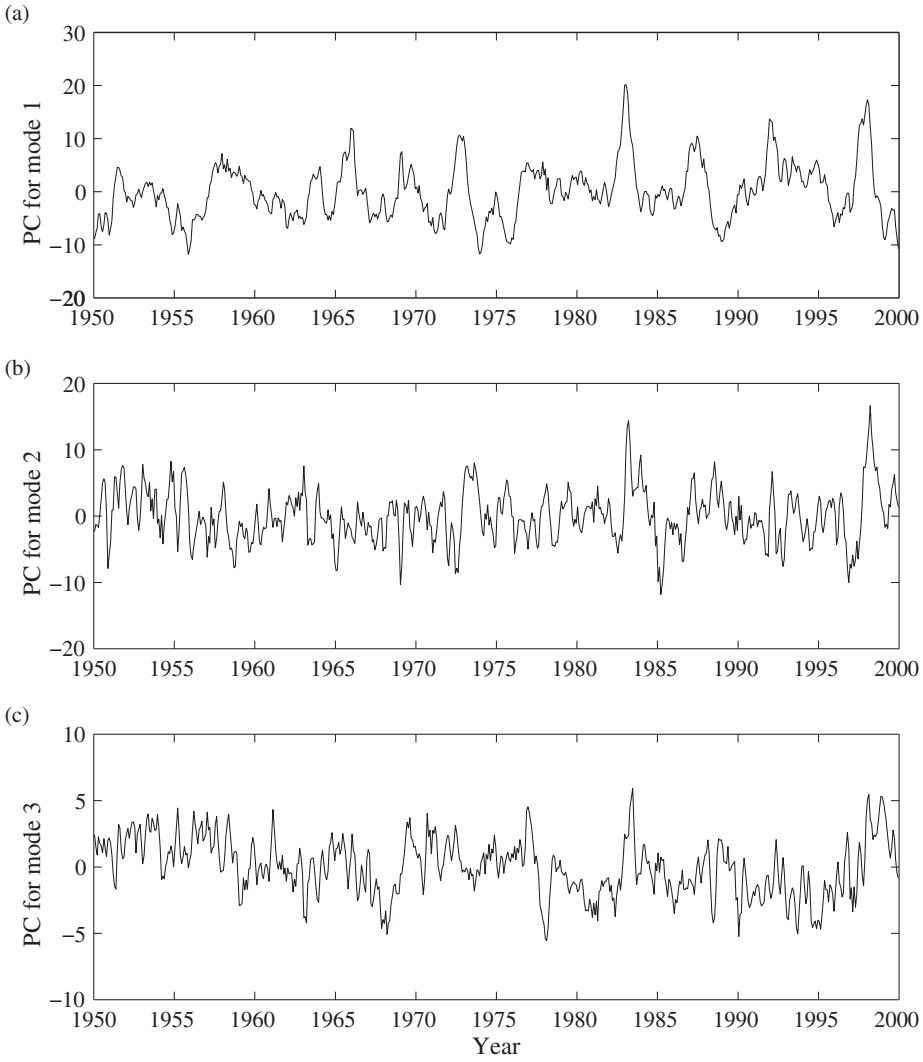


Figure 9.8 The principal component time series for the SLP modes (a) 1, (b) 2 and (c) 3. PC for mode 1 is strongly correlated with the SST mode 1 PC in Fig. 9.6(a). [Source: Hsieh (2009)]

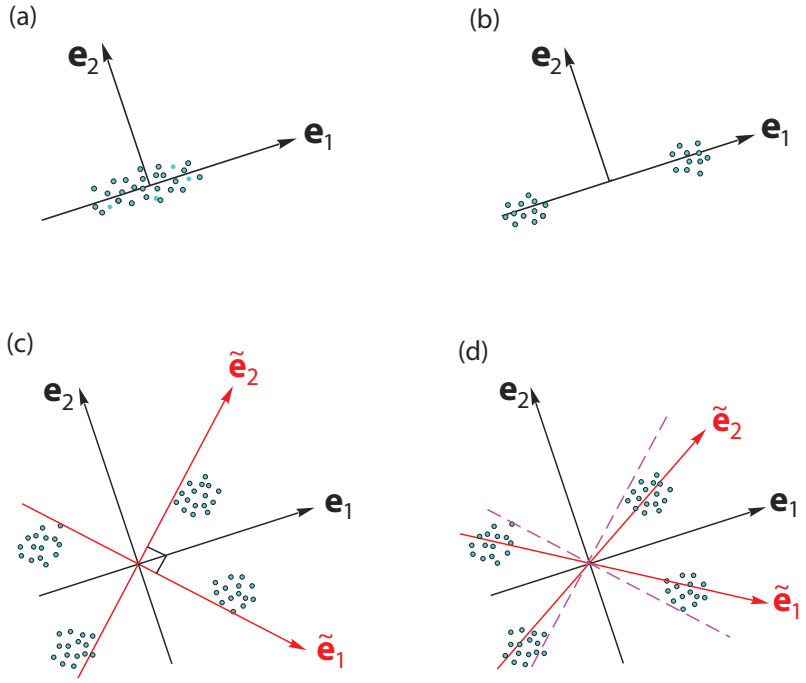


Figure 9.9 The case of PCA applied to a dataset composed of (a) a single cluster, (b) two clusters, (c) three and (d) four clusters. In (c), an *orthonormal rotation* has yielded rotated eigenvectors $\tilde{\mathbf{e}}_j$, $(j = 1, 2)$, which pass much closer to the data clusters than the unrotated eigenvectors $\mathbf{e}_j$. In (d), an *oblique rotation* is used instead of an orthonormal rotation to spear through the data clusters, while the dashed lines indicate the orthonormally rotated eigenvectors. Eigenvectors that failed to approach any data clusters generally bear little resemblance to physical states. [Follows Preisendorfer (1988, figure 7.3).]

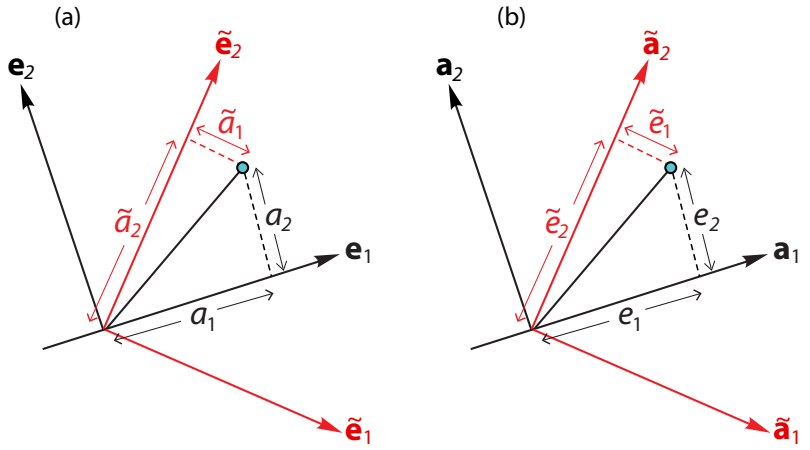


Figure 9.10 (a) In E-frame rotation, the rotated eigenvector $\tilde{\mathbf{e}}_2$ points much closer to the direction of the data point (small circle) than the original eigenvectors $\mathbf{e}_1$ and $\mathbf{e}_2$, with the coordinates of the data point in the original unrotated system being $(a_1,\ a_2)$ and, in the rotated system, $(\tilde{a}_1,\ \tilde{a}_2)$. (b) In A-frame rotation, the rotated PC vector $\tilde{\mathbf{a}}_2$ points much closer to the direction of the data point than the original vectors $\mathbf{a}_1$ and $\mathbf{a}_2$. The roles of $\mathbf{a}$ and $\mathbf{e}$ have been reversed between (a) and (b).

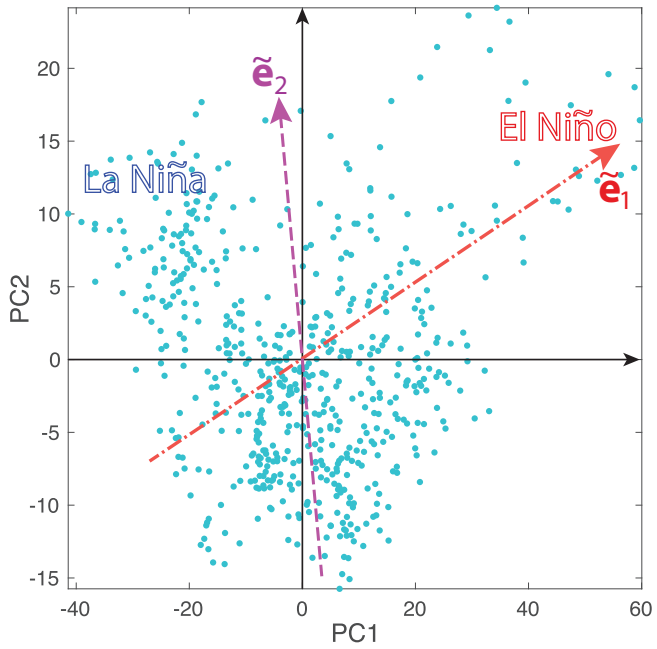


Figure 9.11 PCA with and without rotation for the tropical Pacific SST anomalies. In the PC1–PC2 plane of the scatter plot, where the monthly data are shown as dots, the cool La Niña states lie in the upper left corner, while the warm El Niño states lie in the upper right corner. The first PCA eigenvector would lie along the horizontal direction and the second eigenvector along the vertical direction. A varimax rotation is performed on the first three PCA eigenvectors. The direction of the first RPCA eigenvector $\tilde{\mathbf{e}}_1$ (dot-dashed line) spears through the cluster of El Niño states in the upper right corner, thereby yielding a more accurate description of the SST anomalies during an El Niño. The direction of the second RPCA eigenvector $\tilde{\mathbf{e}}_2$ (dashed line) is orthogonal to the first RPCA eigenvector (though not discernible in this two-dimensional projection of three-dimensional vectors). Note the axes have different scales for clarity. [Adapted from Hsieh (2001b, figure 8).]

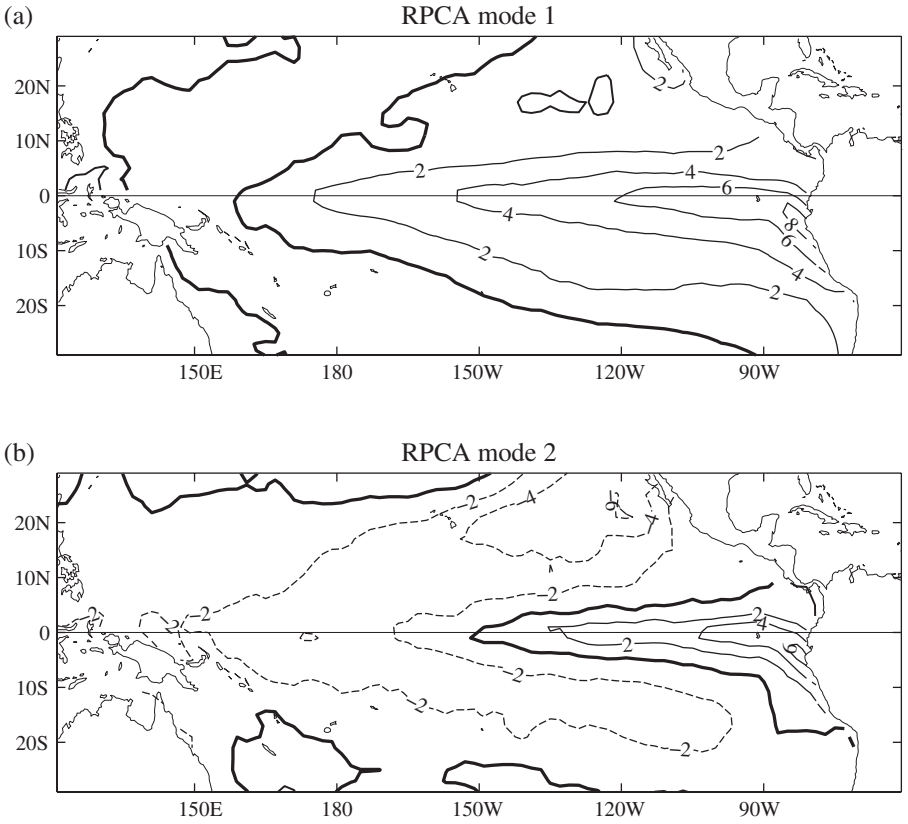


Figure 9.12 The varimax RPCA spatial modes (a) 1 and (b) 2 for the SST. The contour unit is 0.01°C. More intense SST anomalies are found in the eastern equatorial waters off Peru (i.e. just off the west coast of South America) in the RPCA mode 1 than in the PCA mode 1. [Adapted from Hsieh (2001b, figure 9).]

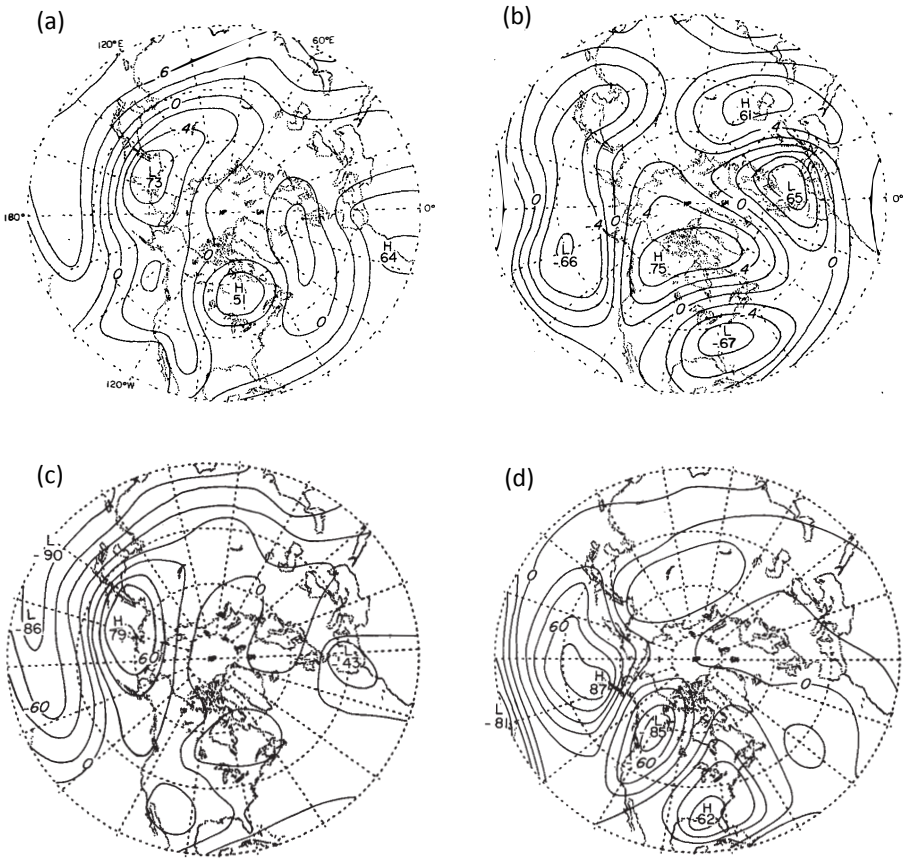


Figure 9.13 Eigenvectors (i.e. EOFs) from PCA on the winter 500 hPa geopotential height anomalies showing the loadings for (a) mode 1 and (b) mode 2. [Reproduced from Wallace and Gutzler (1981, figure 27), ©American Meteorological Society. Used with permission.] Eigenvectors from rotated PCA for (c) mode 1 and (d) mode 2. [Reproduced from Horel (1981, figure 2), ©American Meteorological Society. Used with permission.] The loading at any location is the correlation between the PC and the local 500 hPa height anomaly, as in (9.44). The contour interval is 0.2. The sign of an eigenvector is arbitrary, that is, the entire loading pattern can be multiplied by -1.

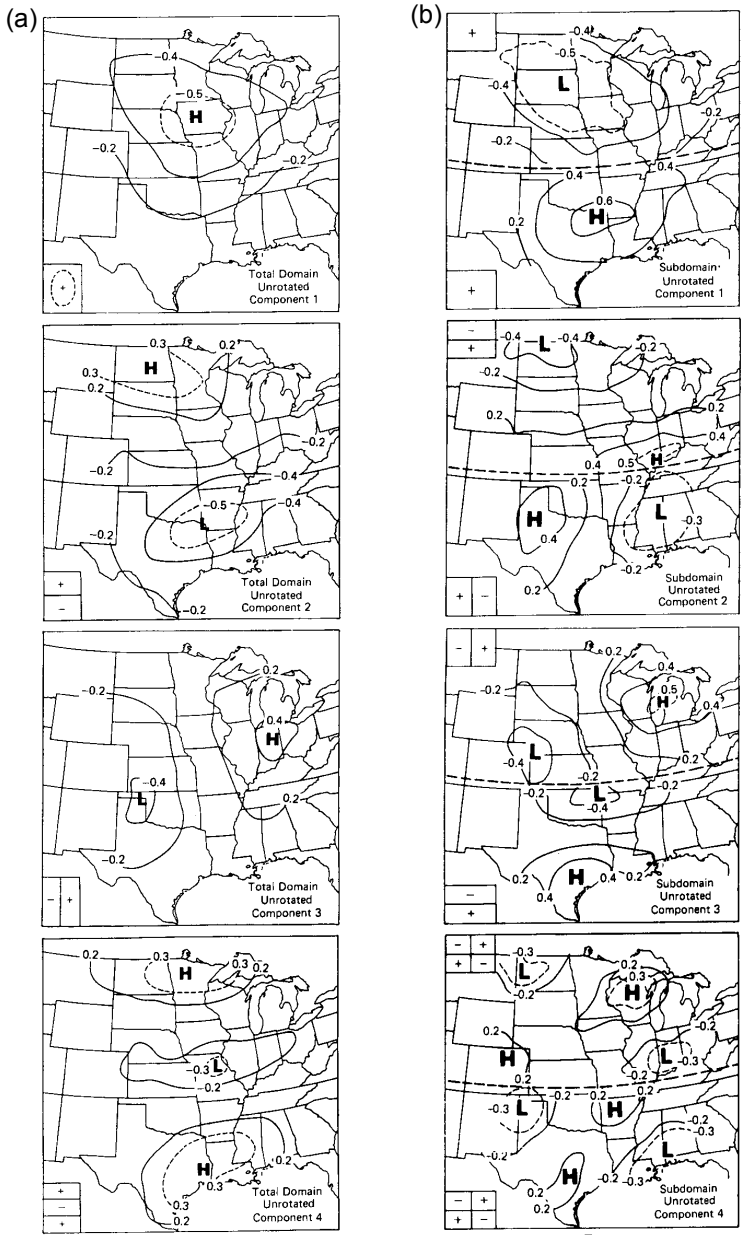


Figure 9.14 First four PCA spatial modes of three-day precipitation (May–August) over central USA. (a) Left panels show the four modes computed for the whole domain. (b) Right panels show the modes computed separately for the northern and southern halves (as separated by the dashed line). Insets show the basic harmonic patterns found by the modes. [Reproduced from Richman (1986, figure 2).]

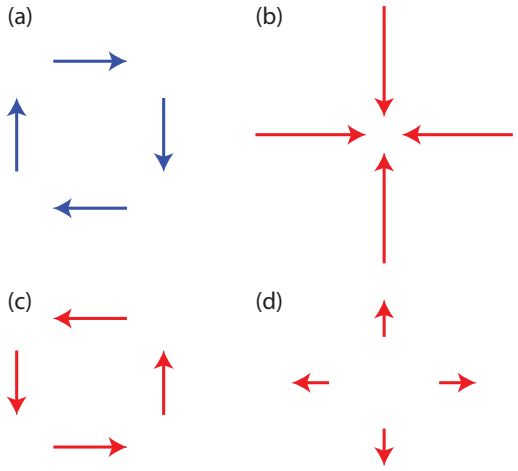


Figure 9.15 A complex PCA mode representing a two-dimensional velocity field: (a) At the first time instance ($l=1$), the PC $=1$ and the eigenvector gives a clockwise circulation pattern. (b) At $l=2$, the PC is $1.5\,e^{-i\pi/2}$ and the mode gives a strong convergent flow pattern. (c) At $l=3$, the PC is $e^{-i\pi}$, with the mode giving a counterclockwise circulation pattern. (d) At $l=4$, the PC is $0.5\,e^{-i3\pi/2}$, with the mode giving a weak divergent flow pattern.

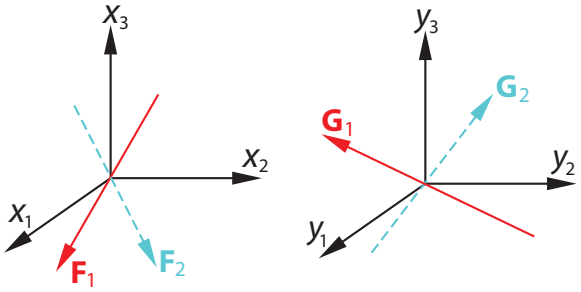


Figure 9.16 Illustrating the CCA solution in the $\mathbf{x}$ and $\mathbf{y}$ spaces. The vectors $\mathbf{F}_1$ and $\mathbf{G}_1$ are the canonical correlation patterns for mode 1, $u_1(t)$ is the amplitude of the fluctuation along $\mathbf{F}_1$ and $v_1(t)$ is the amplitude along $\mathbf{G}_1$. The vectors $\mathbf{F}_1$ and $\mathbf{G}_1$ have been chosen so that the correlation between u_1 and v_1 is maximized. Next, $\mathbf{F}_2$ and $\mathbf{G}_2$ are found, with $u_2(t)$ the amplitude of the fluctuation along $\mathbf{F}_2$, and $v_2(t)$ that along $\mathbf{G}_2$. The correlation between u_2 and v_2 is again maximized, but with $\mathrm{cov}(u_1, u_2) = \mathrm{cov}(v_1, v_2) = \mathrm{cov}(u_1, v_2) = \mathrm{cov}(v_1, u_2) = 0$. In general, $\mathbf{F}_2$ is not orthogonal to $\mathbf{F}_1$, and $\mathbf{G}_2$ is not orthogonal to $\mathbf{G}_1$. Unlike PCA, $\mathbf{F}_1$ and $\mathbf{G}_1$ need not be oriented in the direction of maximum variance. Solving for $\mathbf{F}_1$ and $\mathbf{G}_1$ is analogous to performing rotated PCA in the $\mathbf{x}$ and $\mathbf{y}$ spaces separately, with the rotations determined from maximizing the correlation between u_1 and v_1. [Source: Hsieh (2009)]

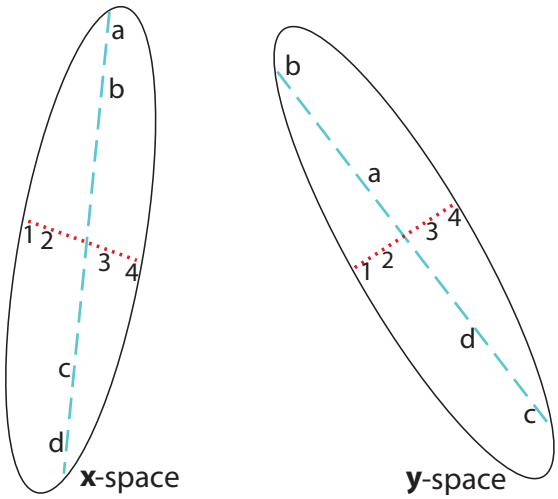


Figure 9.17 Illustrating how CCA may end up extracting a spurious leading mode when working with relatively high-dimensional input spaces. With the ellipses denoting the data clouds in the two input spaces, the dotted lines illustrate directions with little variance but by chance with high correlation (as illustrated by the perfect order in which the data points 1, 2, 3 and 4 are arranged in the **x** and **y** spaces). Since CCA finds the correlation of the data points along the dotted lines to be higher than that along the dashed lines (where the data points a, b, c and d in the **x**-space are ordered as b, a, d and c in the **y**-space), the dotted lines are chosen as the first CCA mode. Maximum covariance analysis (MCA), which looks for modes of maximum covariance instead of maximum correlation, would select the longer dashed lines over the shorter dotted lines, since the lengths of the lines do count in the covariance but not in the correlation; thus, MCA is stable even without pre-filtering by PCA. [Source: Hsieh (2009)]

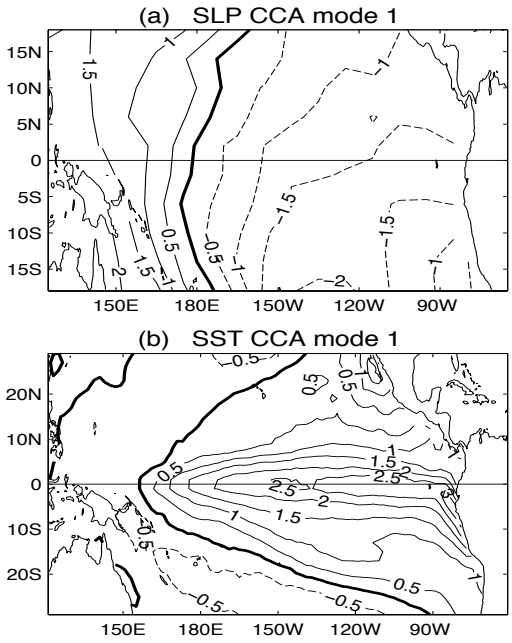


Figure 9.18 The CCA mode 1 for (a) the SLP anomalies and (b) the SST anomalies of the tropical Pacific. As $u_1(t)$ and $v_1(t)$ fluctuate together from one extreme to the other as time progresses, the SLP and SST anomaly fields oscillate as standing wave patterns, changing between an El Niño state and a La Niña state. The pattern in (a) is scaled by $\tilde{u}_1 = [\max(u_1) - \min(u_1)]/2$ and (b) by $\tilde{v}_1 = [\max(v_1) - \min(v_1)]/2$. Contour interval is 0.5 hPa in (a) and 0.5°C in (b). [Reproduced from Hsieh (2001a, figure 6), ©American Meteorological Society. Used with permission.]

Chapter 10: Unsupervised learning

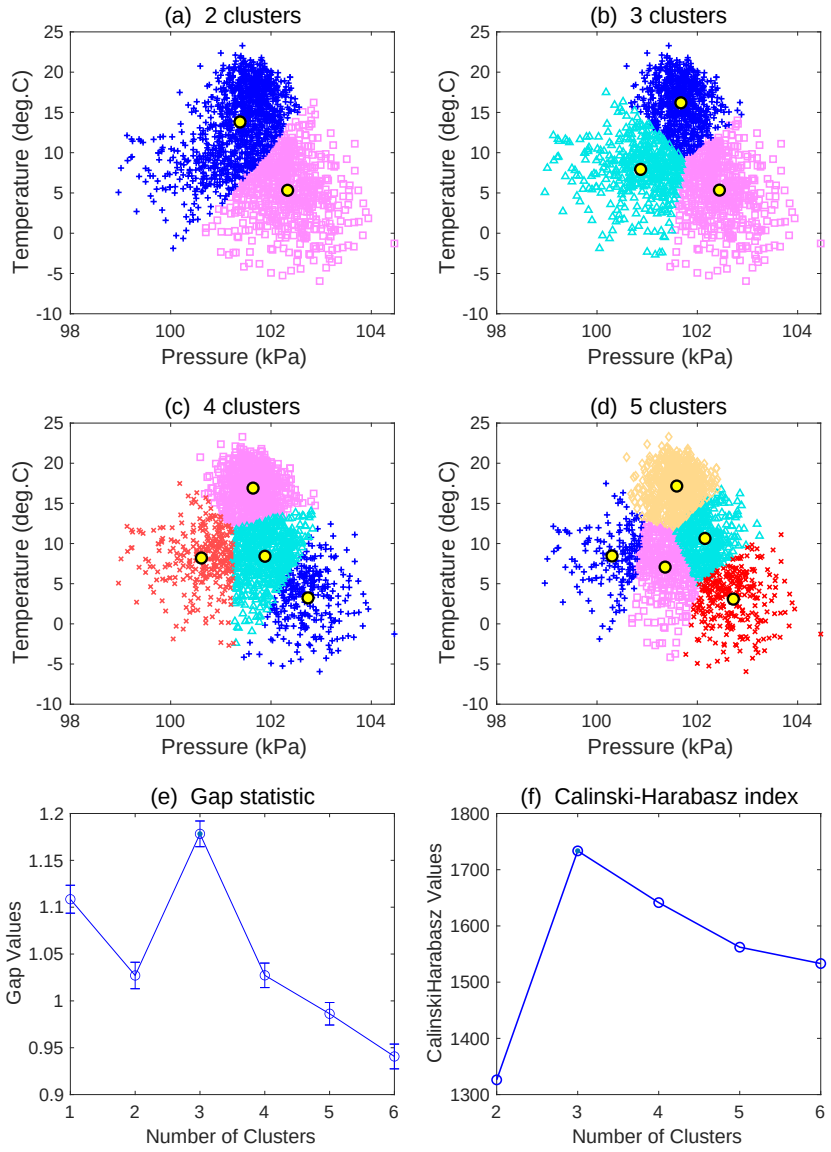


Figure 10.1 K-means clustering of the daily air pressure and temperature data at Vancouver, BC, Canada, using (a) 2, (b) 3, (c) 4 and (d) 5 clusters, with their centroids marked by circles. (e) The gap statistic from (10.15) and (f) the Calinski–Harabasz index, with both choosing $K = 3$ clusters as optimal.

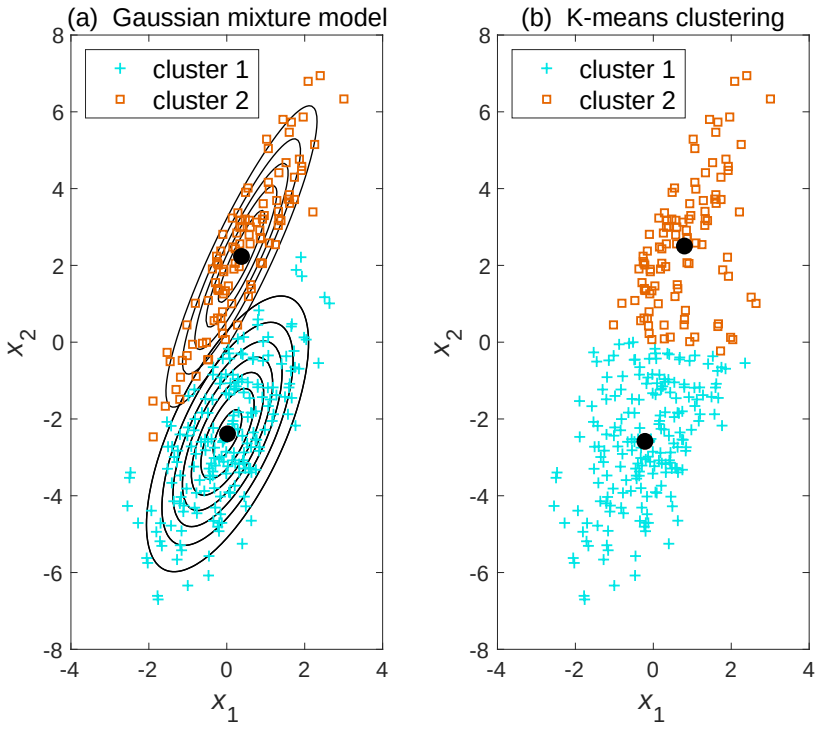


Figure 10.2 2-D data generated from two elliptical Gaussian distributions, with 100 data points in the upper group and 200 in the lower group. (a) Gaussian mixture model clustering of the 300 data points, with $K = 2$, solid circles indicating the centroids and the contours showing the two separate mixture components in (10.24).(b) K-means clustering failing to extract two elliptical-shaped clusters.

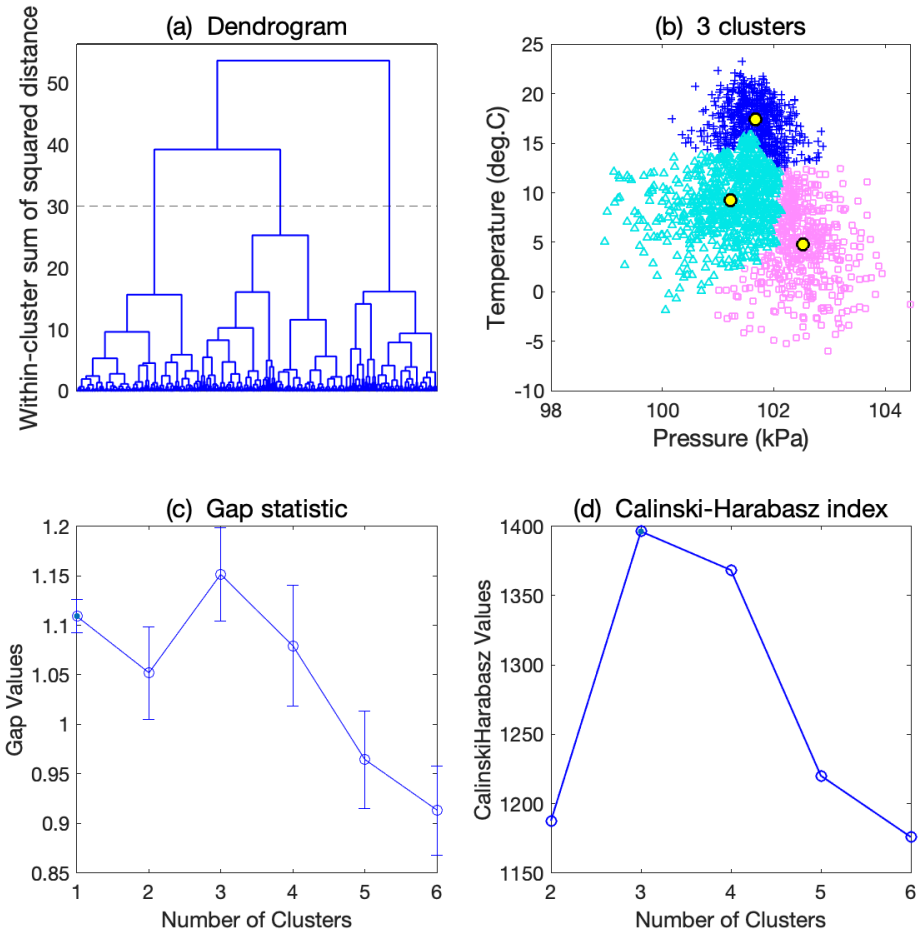


Figure 10.3 Hierarchical clustering of the daily air pressure and temperature data at Vancouver, BC, Canada using Ward's method on standardized data. (a) Dendrogram, where the cut-off level (horizontal dashed line) is set to three clusters, (b) the three clusters, (c) gap statistic and (d) Calinski–Harabasz index.

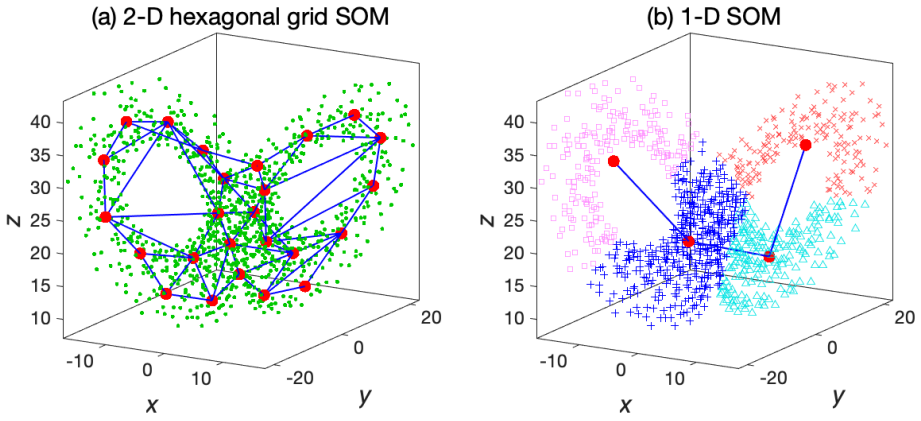


Figure 10.4 (a) A two-dimensional self-organizing map (SOM) where a 5×5 hexagonal mesh is fitted to the Lorenz (1963) attractor data (dots) and (b) a one-dimensional SOM with four units, dividing the data points into four clusters.

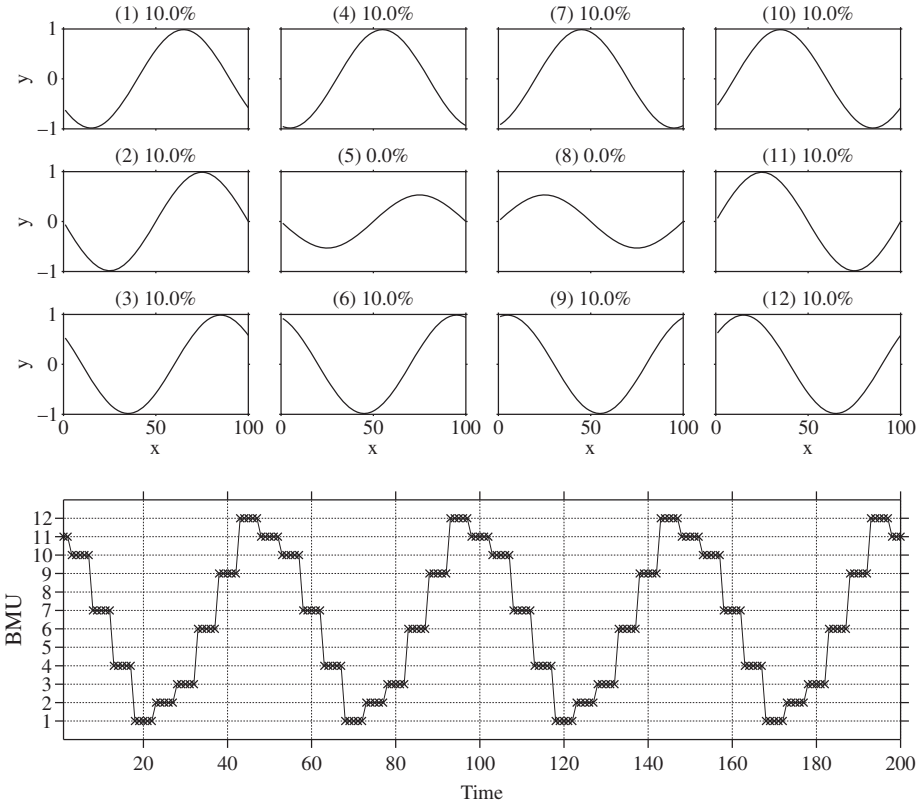


Figure 10.5 A wave propagating to the right is analysed by a 3×4 SOM. The frequency of occurrence of each SOM pattern is given by the percentage on top of each panel. As time progresses, the best matching unit (BMU) rotates counterclockwise around the 3×4 SOM patterns, where the SOM patterns (5) and (8) are bypassed (as indicated by their frequency of occurrence of 0.0%). [Reproduced from Y. Liu, Weisberg, and Mooers (2006, figure 1).]

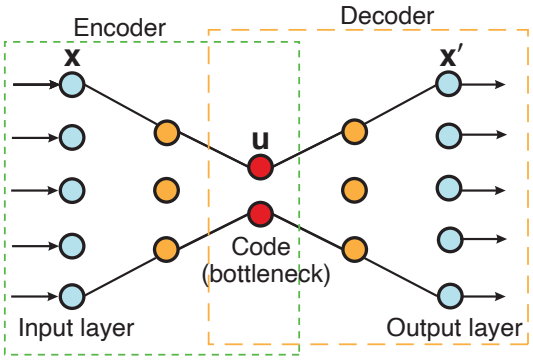


Figure 10.6 The autoencoder or auto-associative neural network is a feedforward MLP NN model, mapping from the input layer to the output layer while passing through several hidden layers, including a bottleneck layer called the 'code', given by the bottleneck nodes $\mathbf{u}$. To reduce clutter, the connecting arrows linking the nodes in the network are not drawn. The model output $\mathbf{x}'$ is trained towards the target data $\mathbf{x}$, the same as the input data $\mathbf{x}$. The first part of the network, called the *encoder*, maps from the input layer to the bottleneck layer, while the second part, the *decoder*, maps from the bottleneck to the output layer.

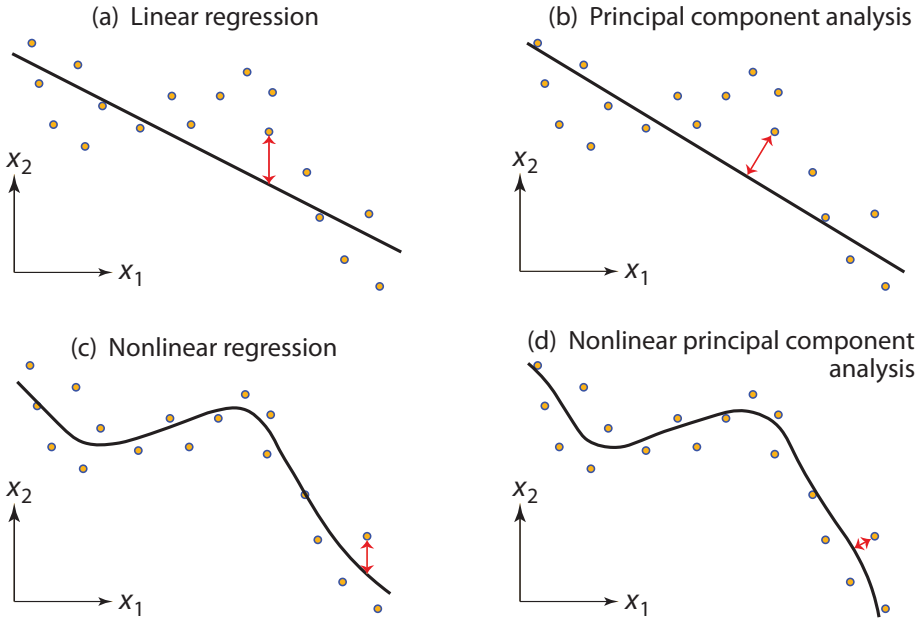


Figure 10.7 Schematic comparison of methods: (a) The linear regression line minimizes the mean squared error (MSE) in the response variable x_2, with the error of a particular data point indicated by the double-headed arrow. (b) Principal component analysis (PCA) minimizes the MSE in all variables, with the double-headed arrow perpendicular to the line. (c) Non-linear regression methods produce a curve minimizing the MSE in the response variable. (d) Non-linear PCA methods use a curve that minimizes the MSE of all variables. In both (c) and (d), the smoothness of the curve can be varied by the method. [Follows Hastie and Stuetzle (1989, figure 1).]

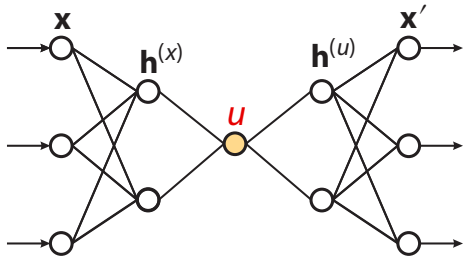


Figure 10.8 An autoencoder for NLPCA, with a feedforward NN architecture. The input layer $\mathbf{x}$ is followed by the encoding layer $\mathbf{h}^{(x)}$, then the code or bottleneck layer (with a single node u for simplicity), the decoding layer $\mathbf{h}^{(u)}$ and finally the output layer $\mathbf{x}'$. Squeezing the input information through a bottleneck accomplishes dimensionality reduction, with u giving the non-linear principal component (NLPC). [Adapted from Hsieh (2001b).]

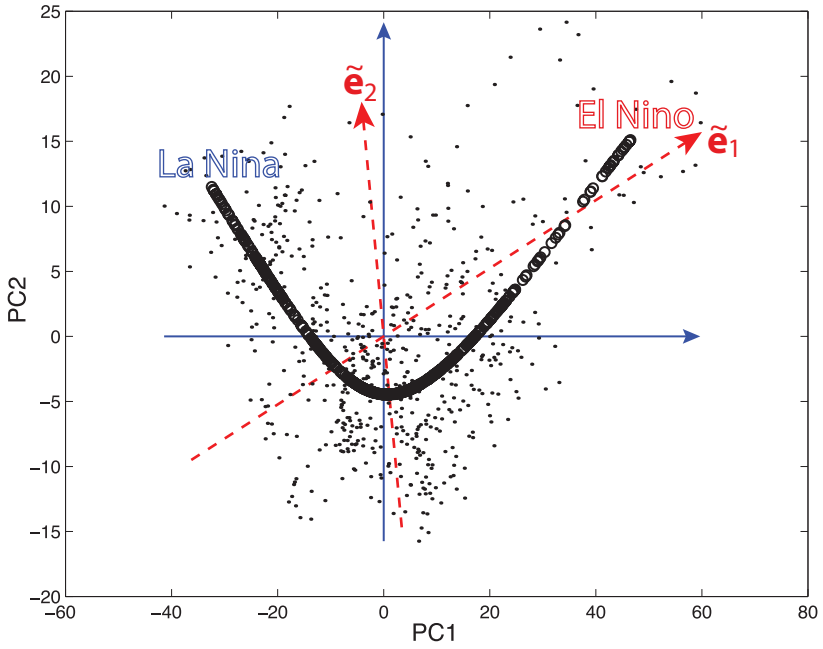


Figure 10.9 Scatter plot of the SST anomaly (SSTA) data (shown as dots) in the PC1–PC2 plane, with the El Niño states lying in the upper right corner and the La Niña states in the upper left corner. The PC2 axis is stretched relative to the PC1 axis for better visualization. The NLPCA first mode approximation to the data is shown by the (overlapping) circles, which trace out a boomerang-shaped curve. The first PCA eigenvector is oriented along the horizontal arrow and the second eigenvector along the vertical arrow. The rotated PCA (RPCA) eigenvectors $\tilde{\mathbf{e}}_1$ and $\tilde{\mathbf{e}}_2$ from a varimax rotation are indicated by the dashed arrows. [Adapted from Hsieh (2001b).]

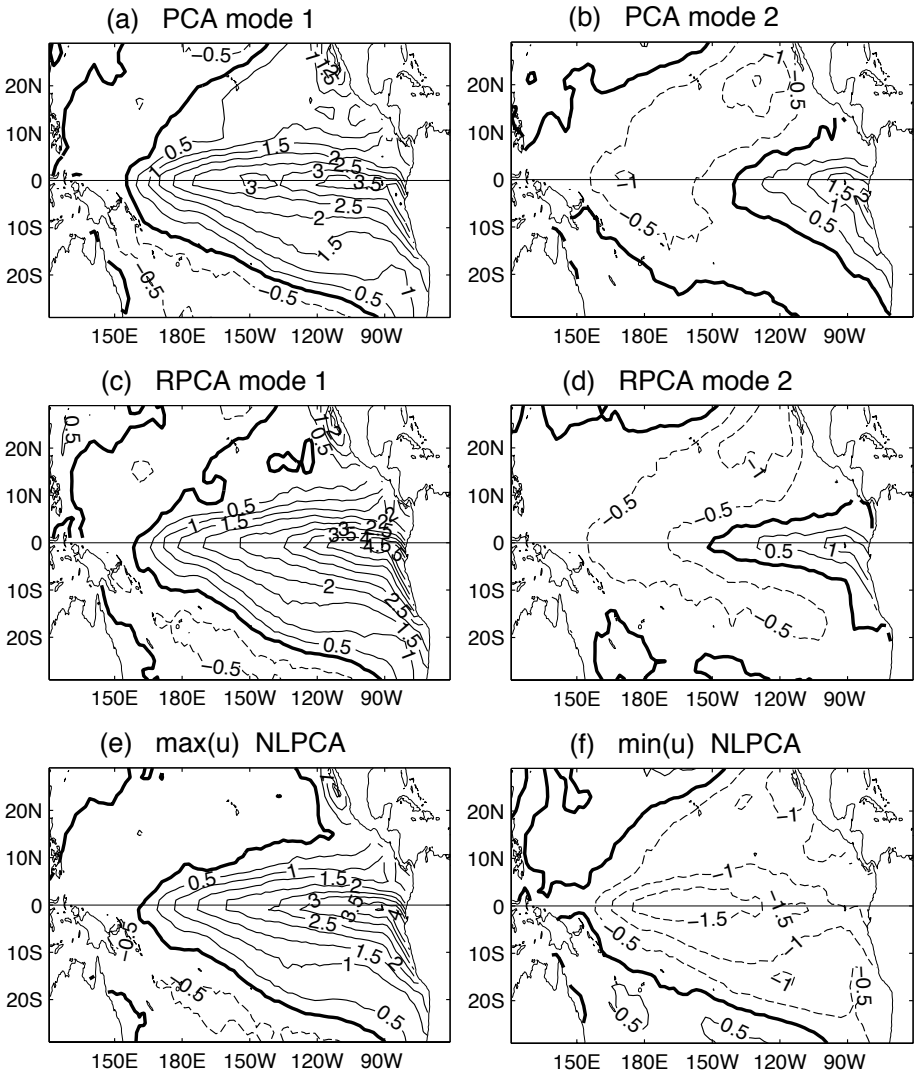


Figure 10.10 The SSTA patterns (in °C) of the PCA, RPCA and the NLPCA. The first and second PCA spatial modes are shown in (a) and (b), respectively (both with their corresponding PCs at maximum value). The first and second varimax RPCA spatial modes are shown in (c) and (d), respectively (both with their corresponding RPCs at maximum value). The anomaly pattern as the NLPC u of the first NLPCA mode varies from (e) its maximum (strong El Niño) to (f) its minimum (strong La Niña). With a contour interval of 0.5°C, the positive contours are shown as solid curves, negative contours as dashed curves and the zero contour as a thick curve. [Reproduced from Hsieh (2004).]

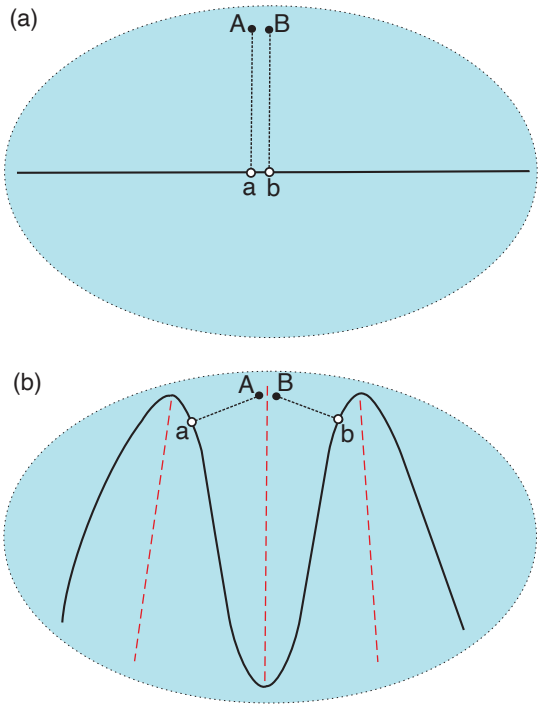


Figure 10.11 Illustrating how overfitting can occur in NLPCA (even in the limit of infinite sample size). (a) PCA solution for a Gaussian data cloud (shaded ellipse), with two neighbouring points A and B shown projecting to the points a and b on the PCA straight line solution. (b) A zigzag NLPCA solution found by a flexible enough non-linear model, with a smaller MSE than that in (a). Dashed lines illustrate 'ambiguity' lines where neighbouring points (e.g. A and B) on opposite sides of these lines are projected to a and b, far apart on the NLPCA curve. [Adapted from Hsieh (2007).]

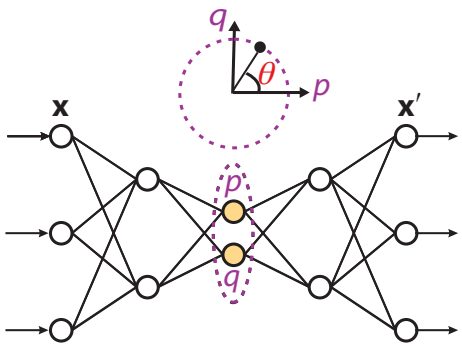


Figure 10.12 NLPCA(cir), the NLPCA model with a circular node at the bottleneck. Instead of having one bottleneck node u, there are now two nodes p and q constrained to lie on a unit circle in the p–q plane, so there is only one free angular variable θ, the NLPC. This network is suited for extracting a closed curve solution. [Adapted from Hsieh (2001b).]

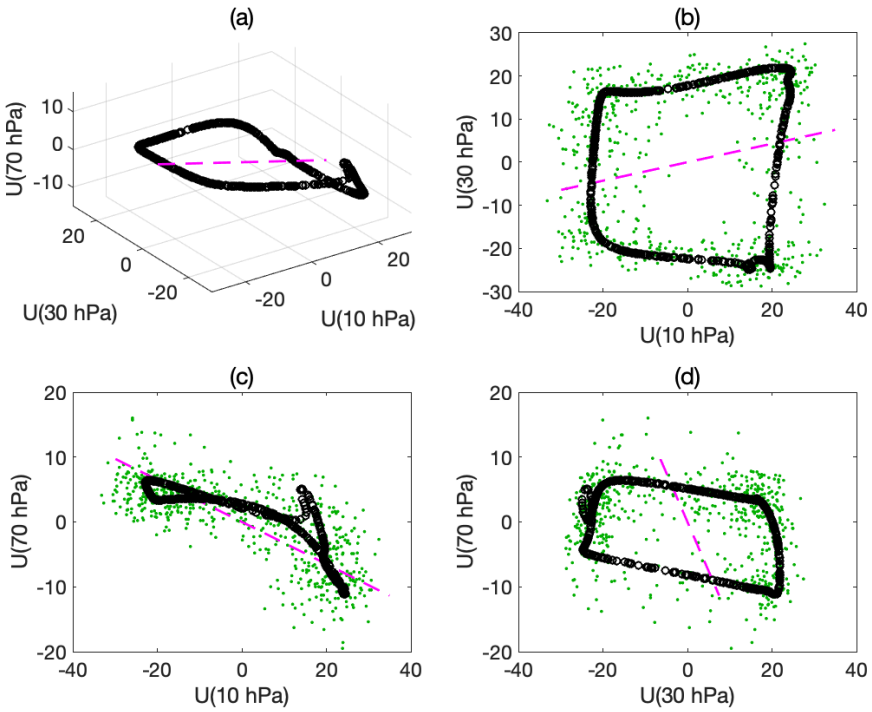


Figure 10.13 The NLPCA(cir) mode 1 solution for the equatorial stratospheric zonal wind anomalies. For comparison, the PCA mode 1 solution is shown by the dashed line. Only three out of seven dimensions are shown, namely the zonal velocity anomaly U at the top, middle and bottom levels (10, 30 and 70 hPa). (a) A 3-D view. (b)–(d) 2-D views. [Reproduced from Hsieh (2007).]

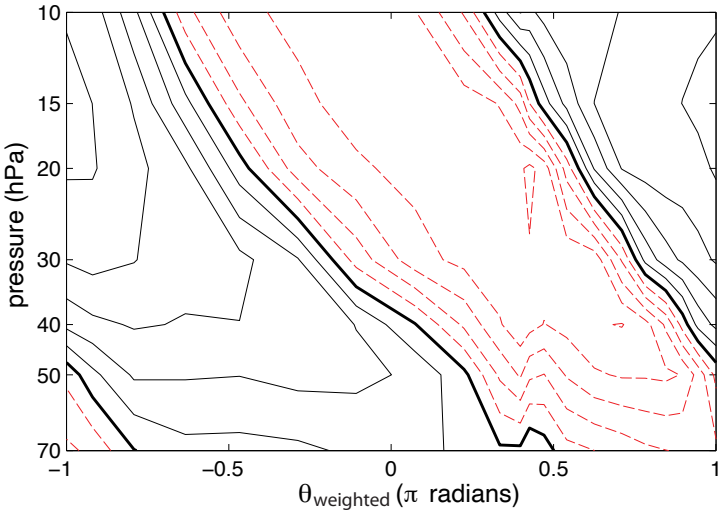


Figure 10.14 Contour plot of the NLPCA(cir) mode 1 zonal wind anomalies as a function of pressure and phase θ_{weighted}, where θ_{weighted} is θ weighted by the histogram distribution of θ (see Hamilton and Hsieh, 2002). Thus, θ_{weighted} is more representative of actual time during a cycle than θ. Contour interval is 5 $\text{m}\,\text{s}^{-1}$, with westerly winds indicated by solid lines, easterlies by dashed lines and zero contours by thick lines. [Reproduced from Hsieh (2007).]

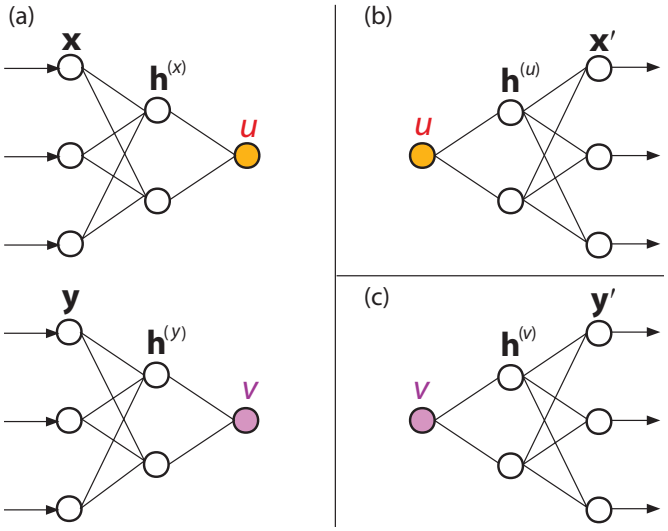


Figure 10.15 The three MLP NNs used to perform NLCCA. (a) The double-barrelled NN maps from the inputs $\mathbf{x}$ and $\mathbf{y}$ to the canonical variates u and v. The objective function J forces the correlation between u and v to be maximized. (b) The NN maps from u to the output layer $\mathbf{x}'$, where the objective function J_1 basically minimizes the MSE of $\mathbf{x}'$ relative to $\mathbf{x}$. (c) The NN maps from v to the output layer $\mathbf{y}'$, where the objective function J_2 basically minimizes the MSE of $\mathbf{y}'$ relative to $\mathbf{y}$. [Reproduced from Hsieh (2001a), ©American Meteorological Society. Used with permission.]

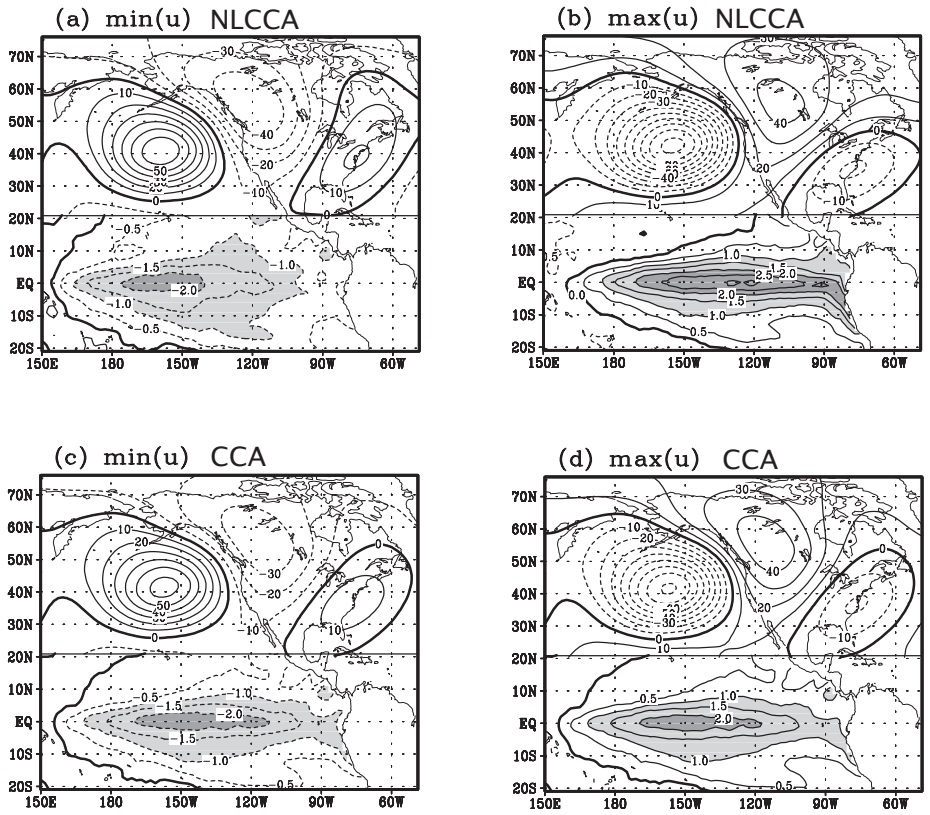


Figure 10.16 The spatial patterns for the first NLCCA mode between the winter Z500A and the tropical Pacific SSTA as the canonical variate u takes its (a) minimum value and (b) maximum value. The Z500A with contour intervals of 10 m are shown north of 20°N. SSTA with contour intervals of 0.5°C are displayed south of 20°N. The SSTA greater than +1°C or less than –1°C are shaded, and more darkly shaded if greater than +2°C or less than –2°C. The linear CCA mode 1 is shown in panels (c) and (d) for comparison. Negative contours are dashed and the zero contour thickened. [Reproduced from Wu, Hsieh, and Zwiers (2003),©American Meteorological Society. Used with permission.]

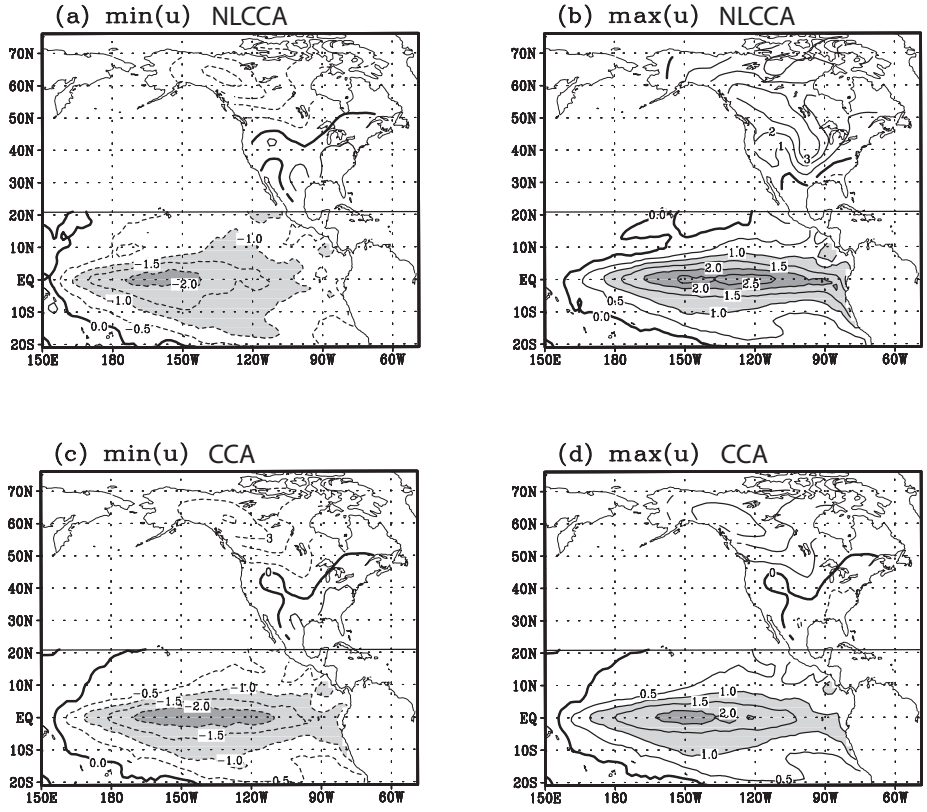


Figure 10.17 Similar to Fig. 10.16, but for the NLCCA mode 1 between the surface air temperature (SAT) anomalies over North America and the tropical SSTA. The contour interval for the SAT anomalies is 1°C. [Reproduced from Wu, Hsieh, and Zwiers (2003), ©American Meteorological Society. Used with permission.]

Chapter 11: Time series

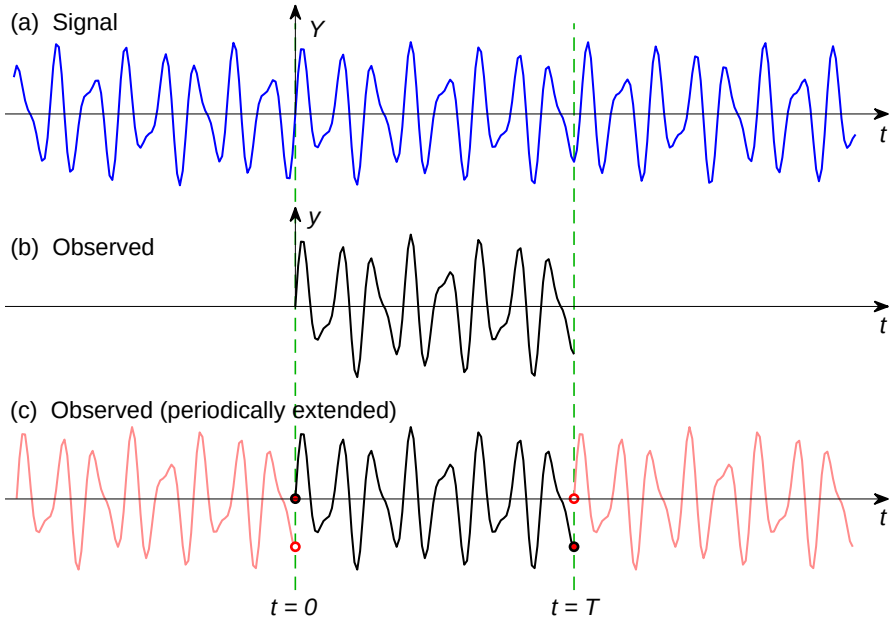


Figure 11.1 A troublesome consequence of the periodicity assumption in Fourier spectral analysis. (a) The true signal Y is over the domain $(-\infty, \infty)$, but (b) the observations y are made during $t = 0$ to T. For Fourier spectral analysis, the observed record is assumed to repeat itself periodically, thereby extending the domain to $(-\infty, \infty)$. (c) The periodicity assumption leads to jump discontinuities at $t = 0$ and $t = T$.

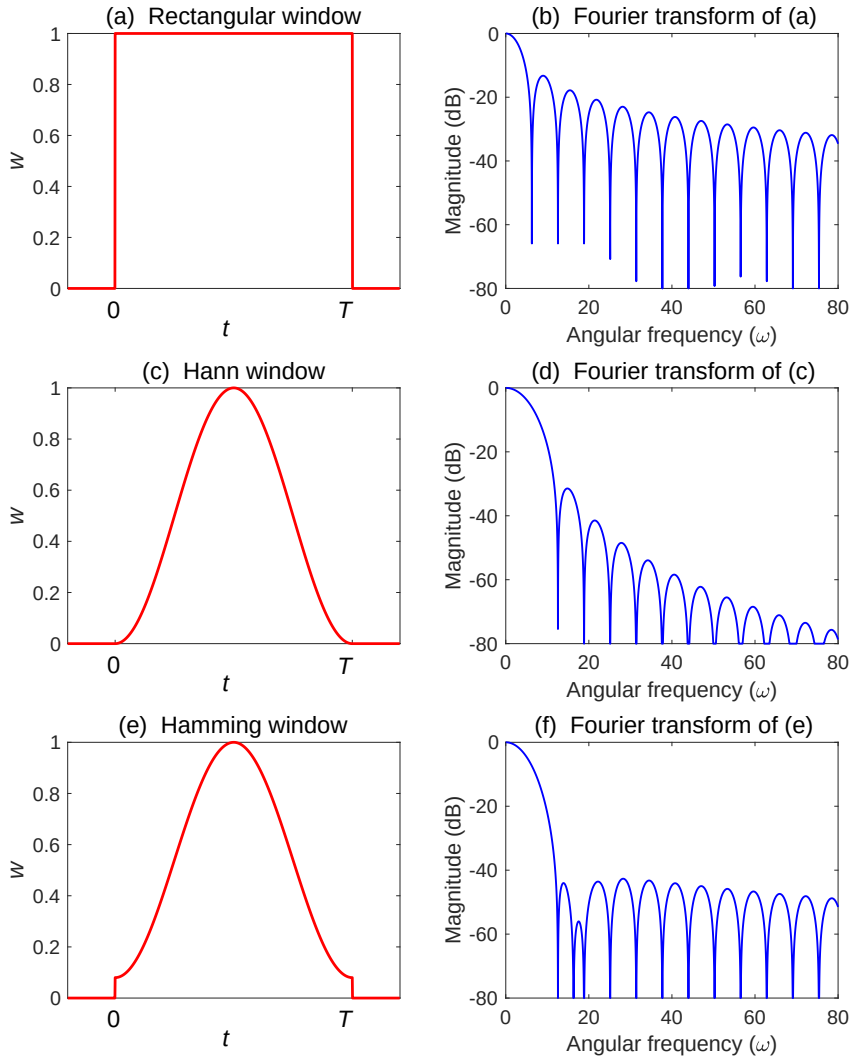


Figure 11.2 Windows and their Fourier transform: (a) and (b) rectangular window, (c) and (d) Hann window, and (e) and (f) Hamming window. The Fourier transforms are plotted only for positive ω, as the function is symmetric about 0, and the magnitude of the Fourier transform is in units of decibel. [The *decibel* (dB) is a common unit for presenting a value (divided by a reference value) on a logarithmic scale. For the magnitude $|y|$ of a variable, the value in dB is given by $20 \log_{10}\left(|y|/|y_0|\right)$, whereas for the power P or variance of y, the value in dB is given by $10 \log_{10}\left(P/P_0\right)$. Here, the maximum value is used as the reference value y_0.]

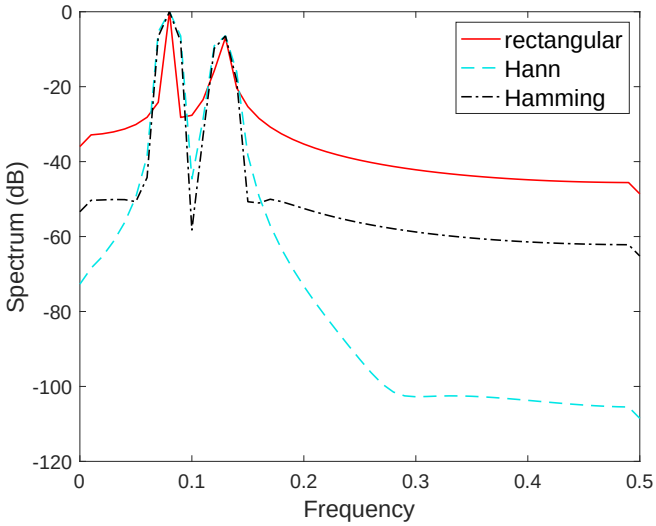
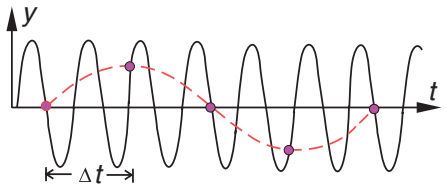


Figure 11.3 The spectrum (in decibels) for the data in Fig. 11.1(b) is computed using the rectangular window (solid curve), the Hann window (dashed) and the Hamming window (dot-dashed). The frequency displayed is $\nu = \omega/(2\pi)$.

Figure 11.4 Illustrating the phenomenon of aliasing. The sampling time interval is Δt, but the signal (solid curve) is oscillating too quickly to be resolved by the sampling. From the observations (dots), an incorrect signal (dashed curve) of much lower frequency is inferred. [Source: Hsieh (2009)]



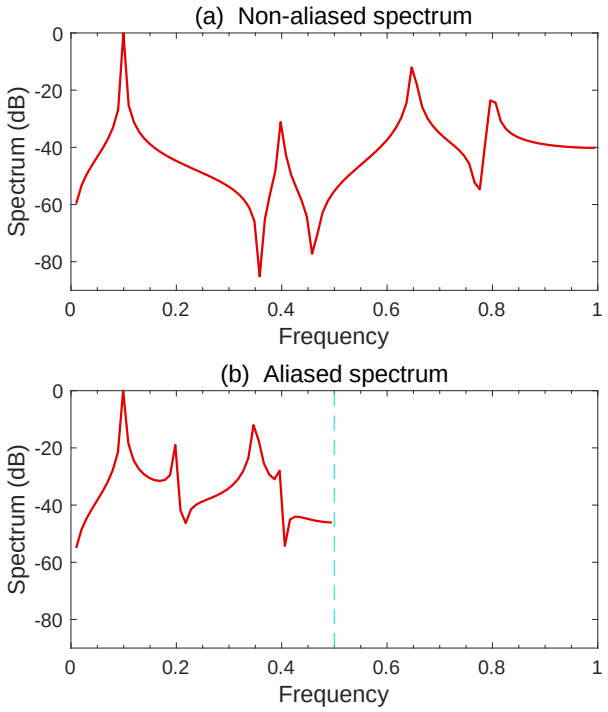


Figure 11.5 The (a) non-aliased spectrum where all four signals have frequencies below the Nyquist frequency $\nu_N = 1$, and (b) the aliased spectrum where by sampling at half the rate, $\nu_N = 0.5$ and two of the higher frequency signals in (a) are reflected or folded back across the vertical dashed line at $\nu_N = 0.5$, creating spurious peaks at the frequencies of 0.2 and 0.35.

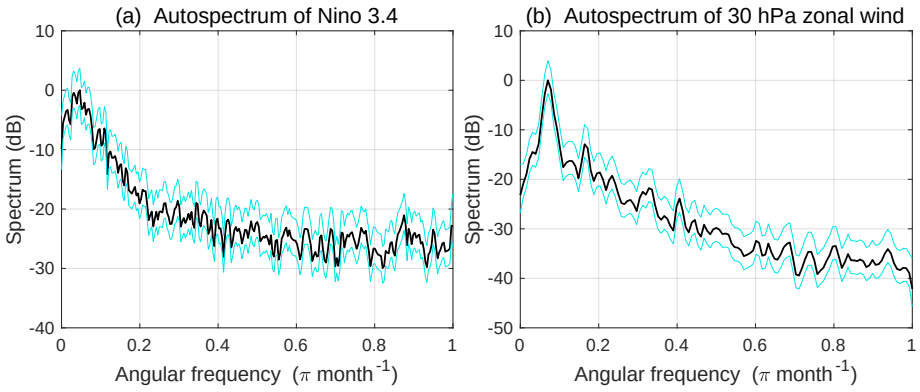


Figure 11.6 Autospectra of (a) the Niño 3.4 index and (b) the equatorial zonal wind at the 30 hPa level, with the 95% confidence interval given by the thin lines. The Welch method (with 8 blocks, 50% overlap and the Hamming window) was used to compute the autospectra after detrending.

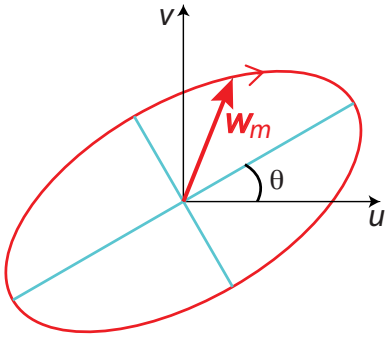


Figure 11.7 In general, in the (u, v) plane, as the complex velocity w_m rotates with angular frequency ω_m, the tip of the velocity vector traces out an ellipse. Here it is shown rotating clockwise as the clockwise component A_m^- happens to be larger than the anti-clockwise component A_m^+. If $A_m^+ = A_m^-$, the ellipse narrows to a straight line.

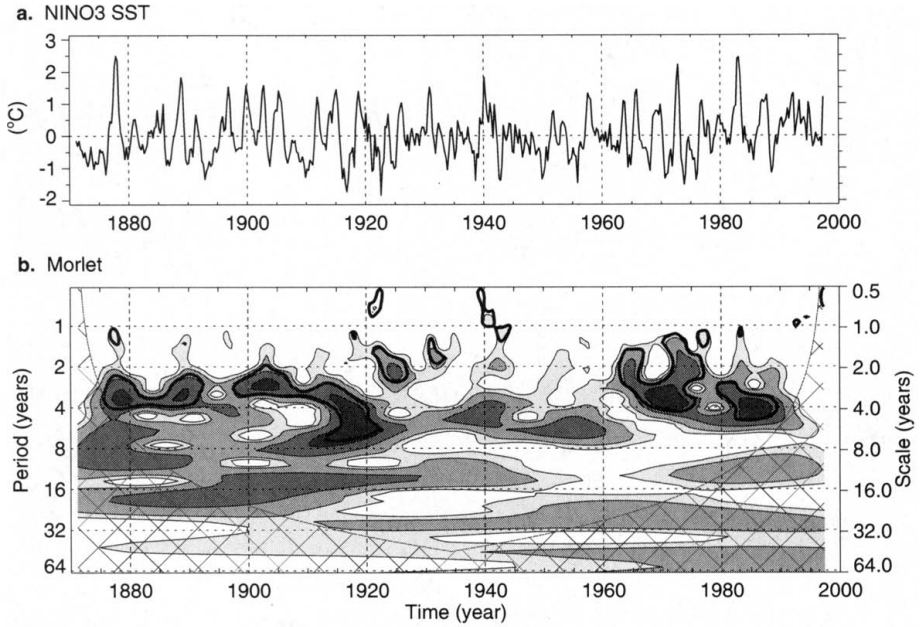


Figure 11.8 (a) The time series of the SST anomalies in the Niño 3 region. (b) The local wavelet power spectrum of the Niño 3 time series using the Morlet wavelet. The left axis is the period in years, corresponding to the wavelet scale on the right axis. The shaded contours are at normalized variances of 1, 2, 5 and 10, with thick contours enclosing regions above 95% confidence for a lag-1 red noise process (see Section 11.8.2). Cross-hatched regions on either end indicate the 'cone of influence', where edge effects become important. [Reproduced from Torrence and Compo (1998, figure 1), ©American Meteorological Society. Used with permission.]

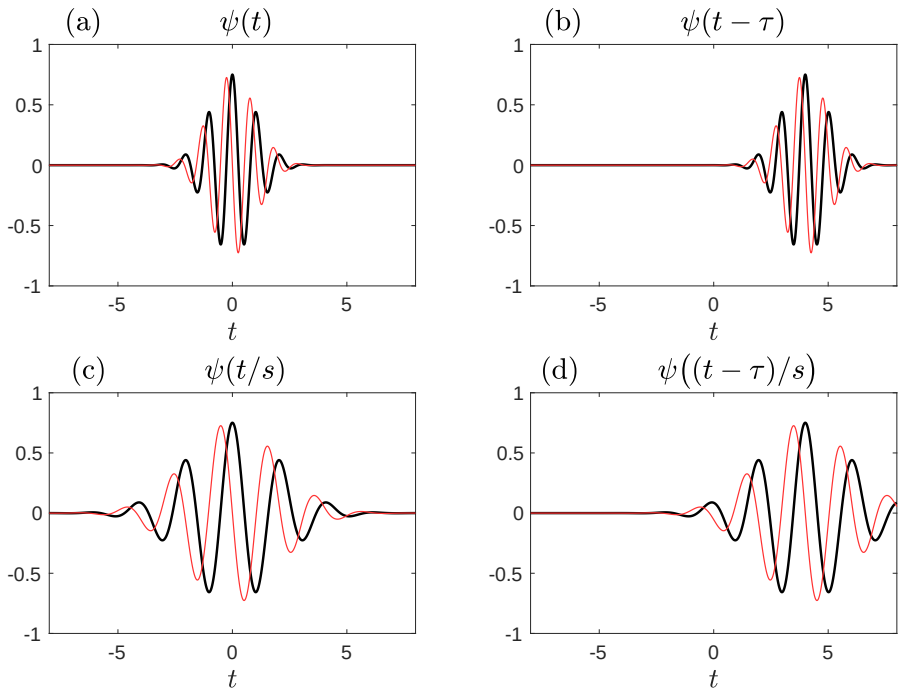


Figure 11.9 (a) The Morlet wavelet with real part (thick line) and imaginary part (thin line), (b) wavelet shifted to the right by τ, (c) wavelet scaled by the factor $s = 2$ and (d) wavelet shifted and scaled.

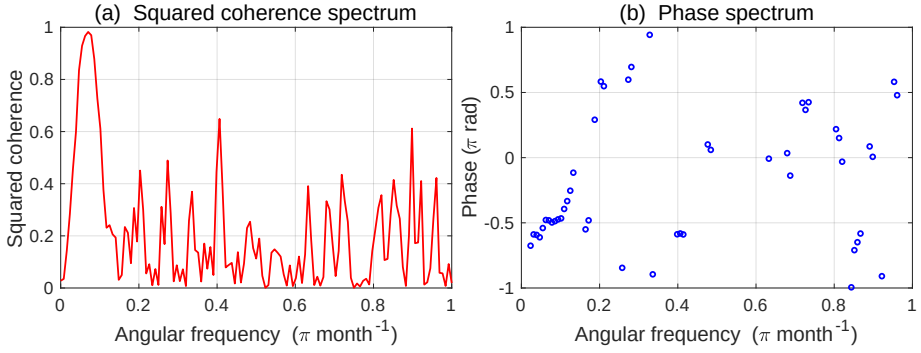


Figure 11.10 (a) The squared coherence and (b) phase from the cross-spectrum between the equatorial zonal wind at the 30 hPa and at the 10 hPa levels. The phase is only plotted when the squared coherence value is ≥ 0.2. The phase is positive if the wind at the 30 hPa level leads that at the 10 hPa level and negative if vice versa. After detrending, the Welch method (with 8 blocks, 50% overlap and the Hamming window) was used to compute the cross-spectrum.

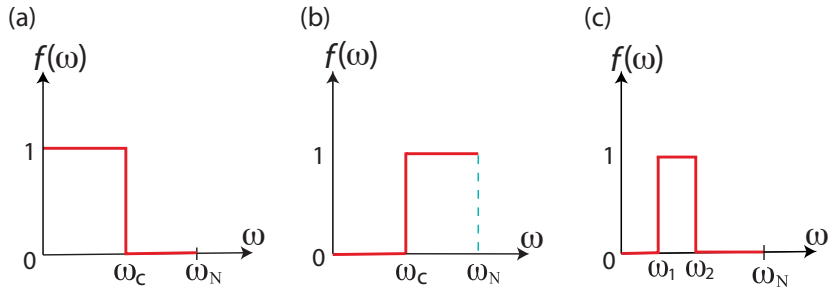


Figure 11.11 Ideal filters: (a) Low-pass, (b) high-pass and (c) band-pass, where $f(\omega)$ is the filter response function and $\omega_{\rm N}$ is the Nyquist frequency. In (a), frequencies below the cutoff frequency $\omega_{\rm c}$ are allowed to pass through the filter, while frequencies above $\omega_{\rm c}$ are eliminated. In (b), the situation is reversed, while in (c), only frequencies within a selected band ($\omega_1 \leq \omega \leq \omega_2$) are allowed to pass through the filter. In these ideal filters, jump discontinuities in $f(\omega)$ give infinitely sharp transitions, which are not attainable in practice.

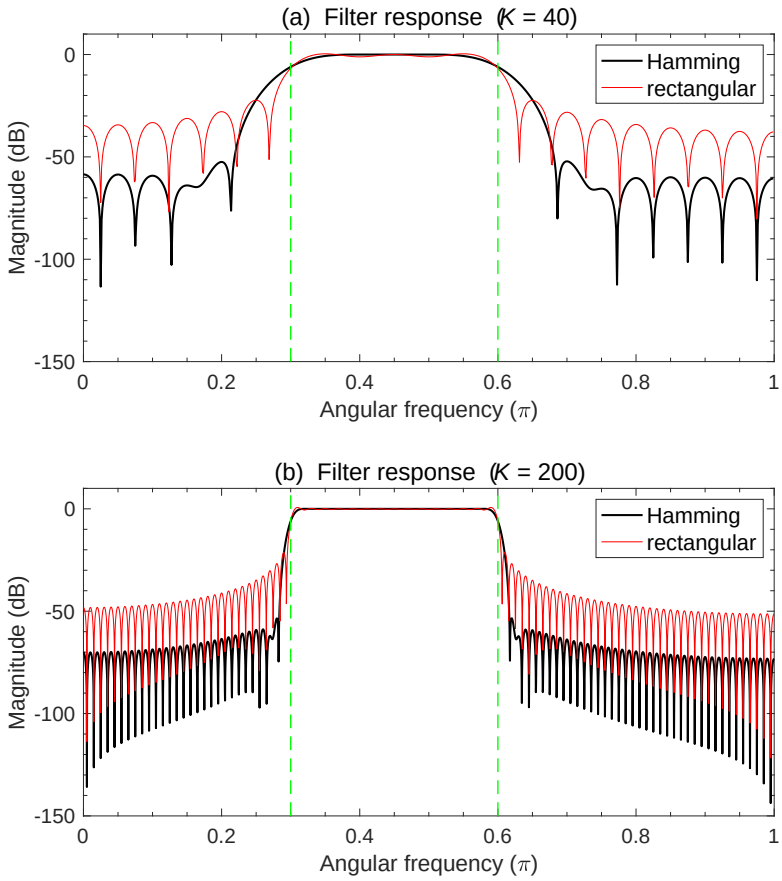


Figure 11.12 Magnitude of the filter response function $f(\omega)$ plotted as a function of the angular frequency ω (in units of π), using the Hamming window (thick line) and the rectangular window (thin line), with the filter order (a) $K = 40$ and (b) $K = 200$. The ideal filter has infinitely sharp transitions at $\omega_1 = 0.3\pi$ and $\omega_2 = 0.6\pi$, as marked by the vertical dashed lines.

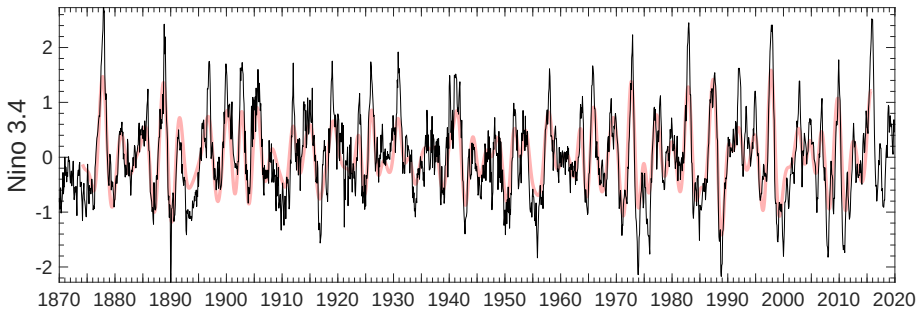


Figure 11.13 The monthly Niño 3.4 time series, unfiltered (thin line) and band-pass filtered (thick line). The grid mark for a year marks the January of that year.

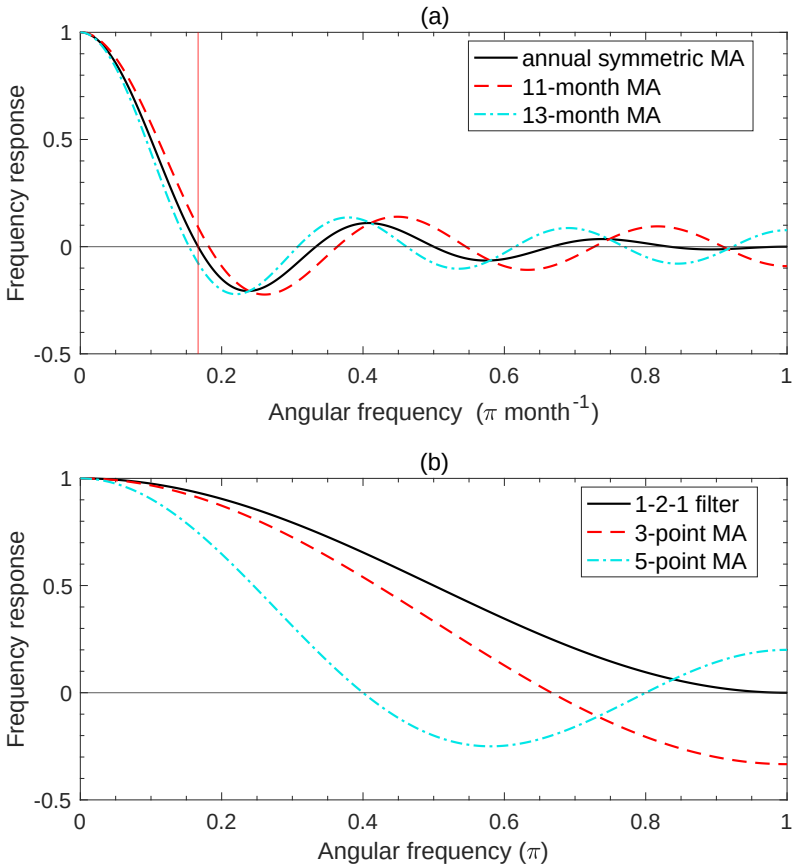


Figure 11.14 Comparison of the filter response $f(\omega)$ for various filters: (a) The annual symmetric moving average (MA) filter and the 11-month and 13-month MA filters and (b) the 1-2-1 filter and the three-point and five-point MA filters. The angular frequency ω is in units of π and the vertical line in (a) marks the frequency ω_{annual} for the annual cycle.

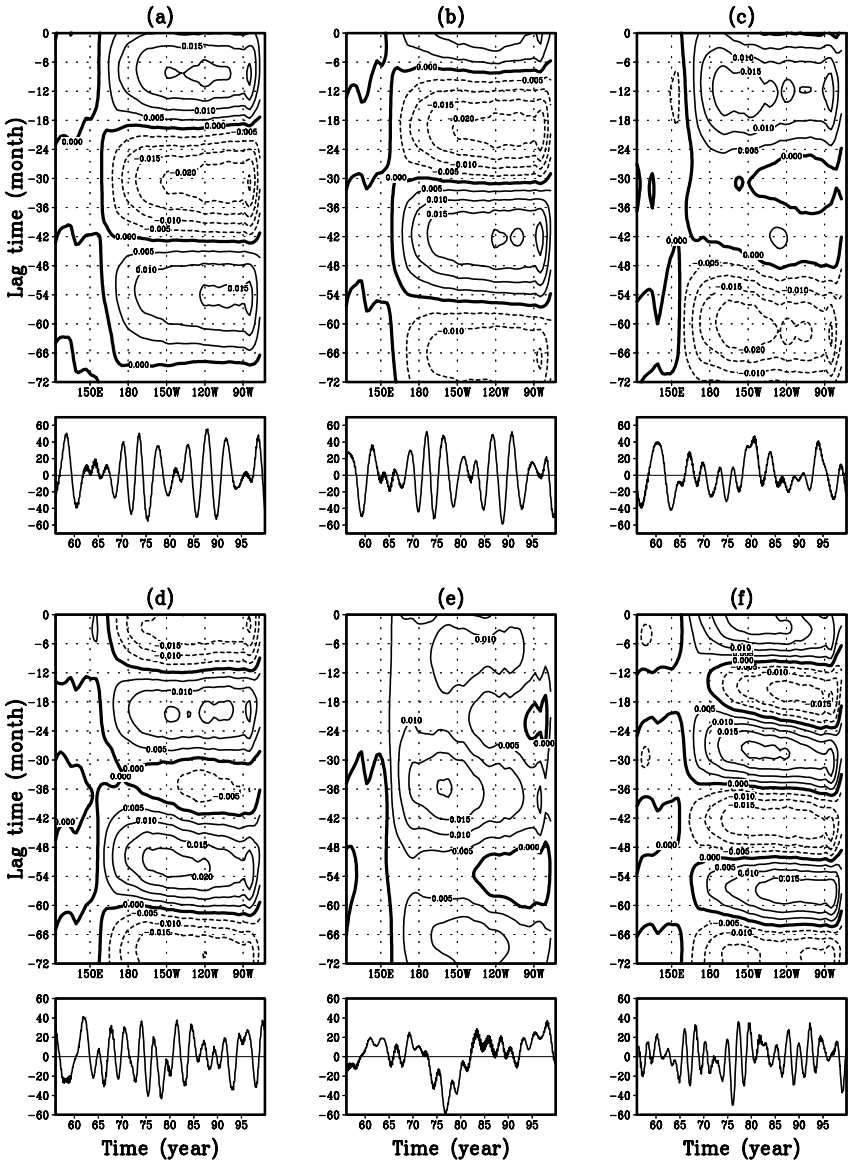


Figure 11.15 The SSA modes 1–6 for the tropical Pacific SSTA shown in (a)–(f), respectively. The contour plots show the SSTA along the equator as a function of the lag. The zero contour is marked by the thick curve, and positive and negative anomalies by solid and dashed curves, respectively. The PC time series is shown beneath each contour plot. [Reproduced from Hsieh and Wu (2002).]

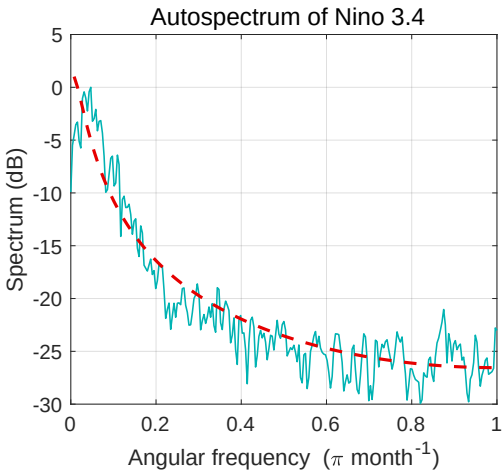


Figure 11.16 Autospectrum of the Niño 3.4 index (solid line) and the AR(1) model (dashed line). The AR(1) model parameter ϕ is 0.92.

Chapter 12: Classification

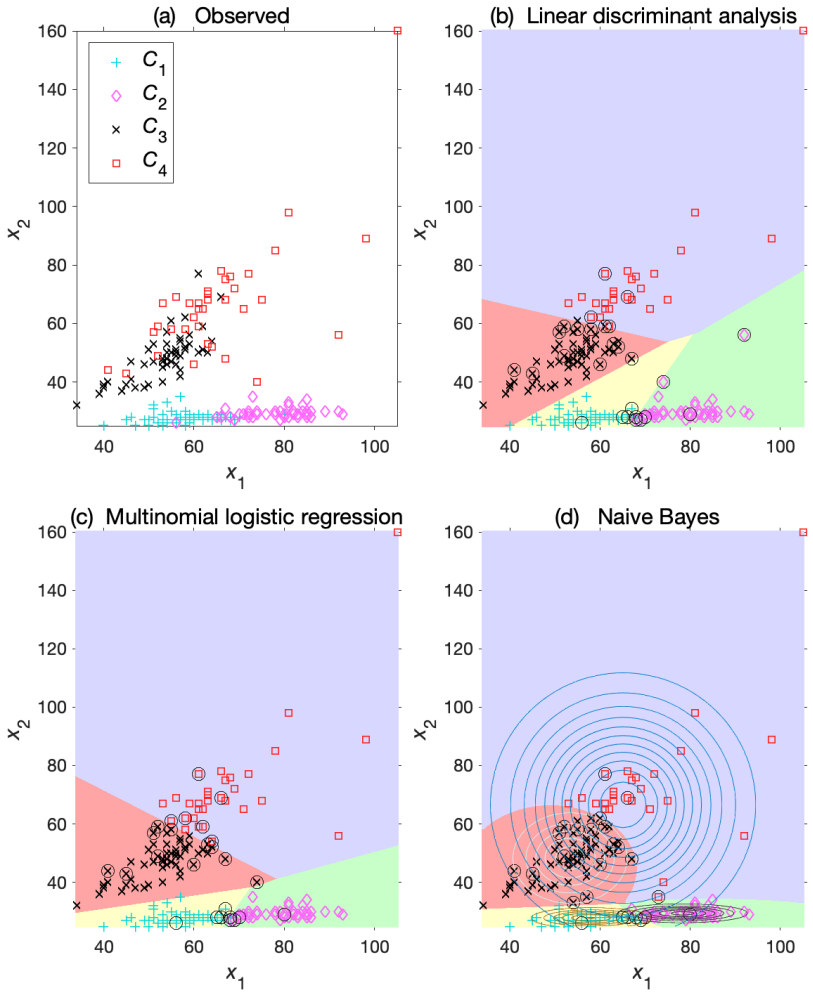


Figure 12.1 (a) Training dataset with two predictor variables x_1 and x_2 from two spectral bands and four classes $(C_1, \ldots, C_4)$ of land surface and (b) the same data classified by LDA. Decision regions for the four classes are shown by different background shading and misclassified data points are circled. (c) Data classified by multinomial logistic regression and (d) data classified by naive Bayes, where the Gaussian distributions for the four classes are shown by contours extending out to 4 σ_{ij}. [Data source: B. Johnson et al. (2012), UCI Machine Learning Repository http://archive.ics.uci.edu/ml, Irvine, CA: University of California, School of Information and Computer Science.]

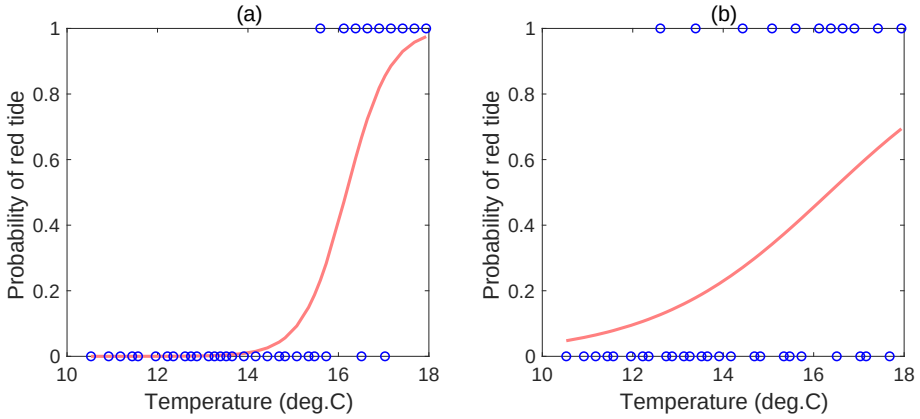


Figure 12.2 Logistic regression fit to synthetic datasets representing the occurrence of red tide for various water temperatures, with data points marked by circles. The transition between classes is relatively sharp in (a) and gradual in (b).

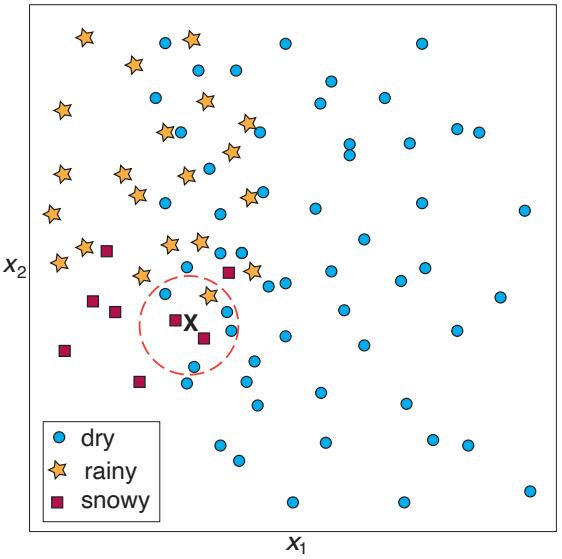


Figure 12.3 Illustrating seven nearest neighbours (within the dashed circle) around a particular feature vector $\mathbf{x}$ in a 2-D feature space for the KNN classifier with $K = 7$. The three classes represent 'dry', 'rainy' and 'snowy' conditions. Since 'dry' has the most votes in this neighbourhood, the model outputs 'dry'.

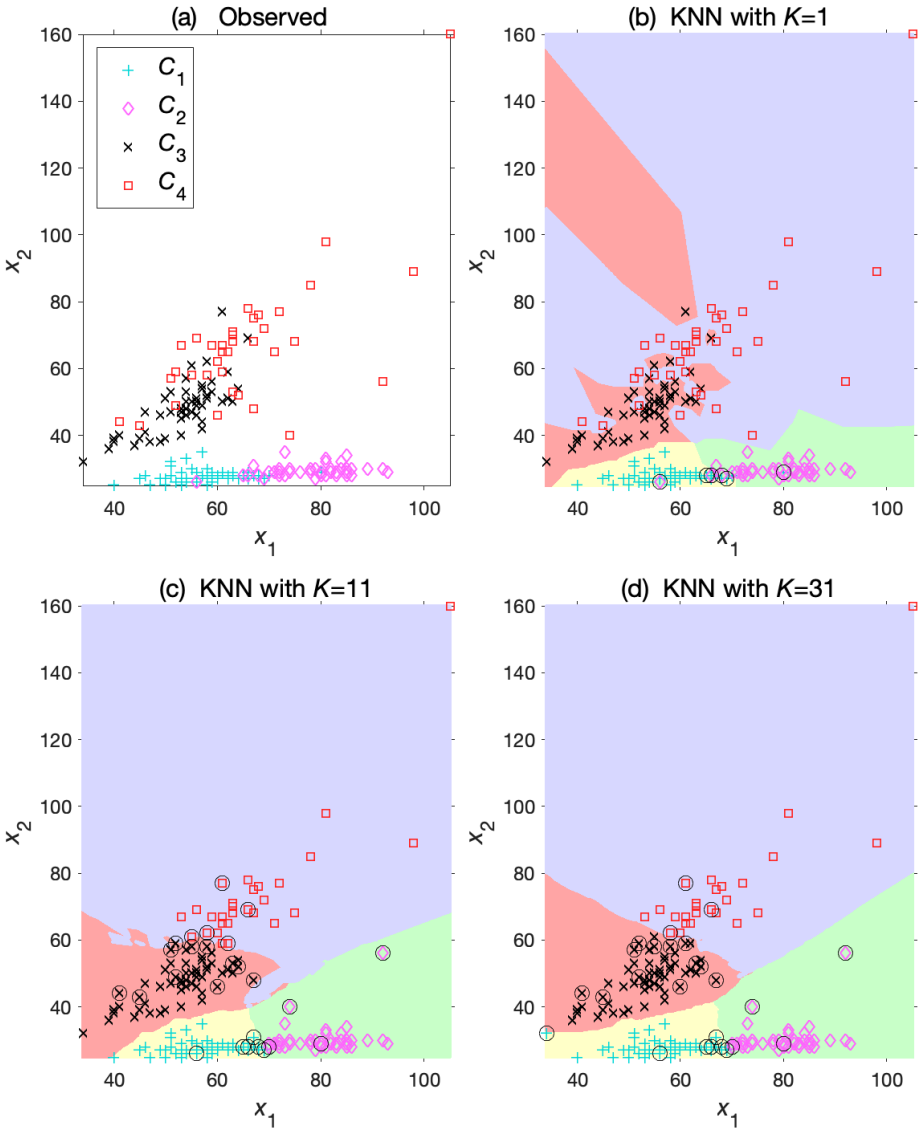


Figure 12.4 (a) Given the same training dataset as in Fig. 12.1(a) for forest classification, the data are classified by the method of K-nearest neighbours, with (b) $K = 1$, (c) $K = 11$ and (d) $K = 31$. Decision regions for the four classes are shown by different background shading, and misclassified data points are circled.

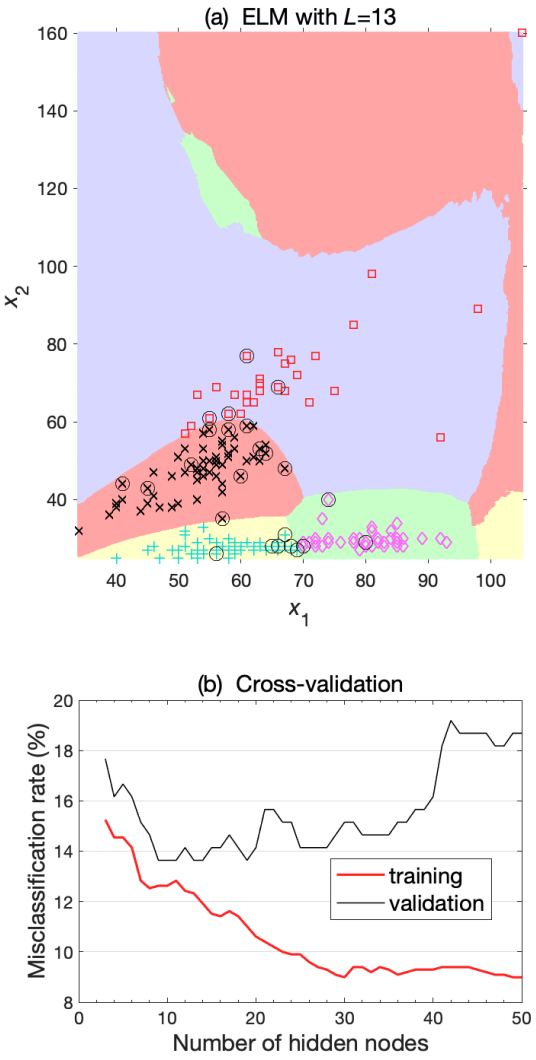


Figure 12.5 Given the training dataset in Fig. 12.1(a) for forest classification, the data are classified by an ensemble of 99 ELM models with (a) $L = 13$ hidden nodes. Class C_1 is indicated by $+$, C_2 by $\Diamond$, C_3 by $\times$ and C_4 by $\square$. Decision regions for the four classes are shown by different background shading, and misclassified data points are circled. (b) Misclassification rate for the training and validation data under a 6-fold cross-validation scheme, with the number of hidden nodes ranging from 3 to 50. The minimum validation error occurred when $L = 13$. This estimated value for L can vary considerably if different random weights are used, since the validation error is quite noisy due to the small sample size of 198 data points in the training dataset.

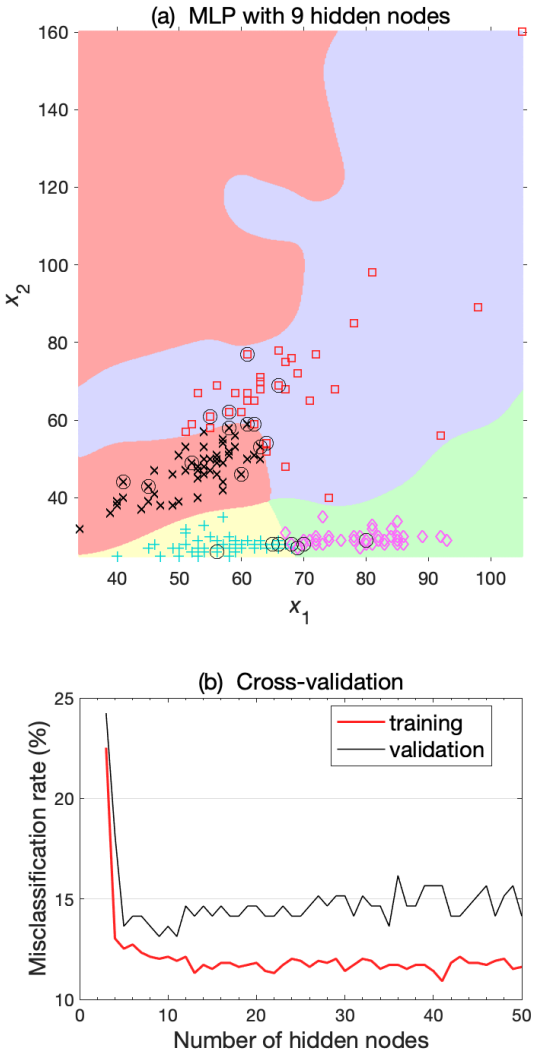


Figure 12.6 Given the training dataset in Fig. 12.1(a) for forest classification, the data are classified by an ensemble of 25 MLP models with (a) $L = 9$ hidden nodes. Class C_1 is indicated by $+$, C_2 by $\Diamond$, C_3 by $\times$ and C_4 by $\square$. Decision regions for the four classes are shown by different background shading, and misclassified data points are circled. (b) Misclassification rate for the training and validation data under a 6-fold cross-validation scheme, with the number of hidden nodes ranging from 3 to 50. The minimum validation error occurred when $L = 9$. This estimated value for L can vary considerably if different random weights are used, since the validation error is quite noisy due to the small sample size of 198 data points in the training dataset.

Chapter 13: Kernel methods

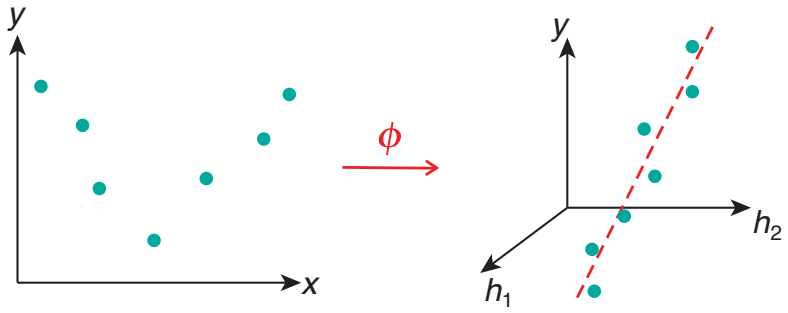


Figure 13.1 Illustrating the effect of the non-linear mapping ϕ from the input space to the hidden space, where a non-linear relation between the input x and the response y becomes a linear relation (dashed line) between the hidden variables $\mathbf{h}$ and y.

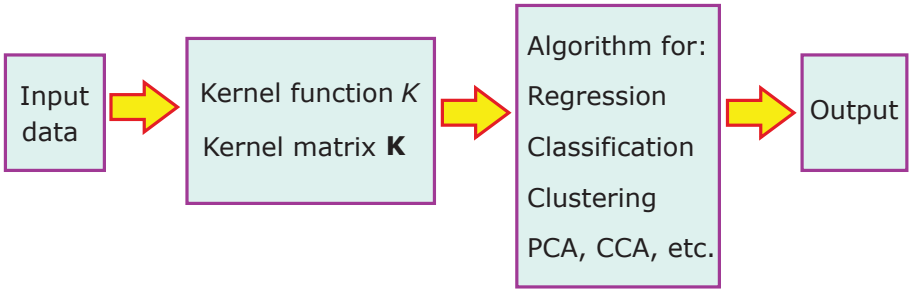


Figure 13.2 The modular architecture of the kernel method. [Follows Shawe-Taylor and Cristianini (2004, figure 2.4).]

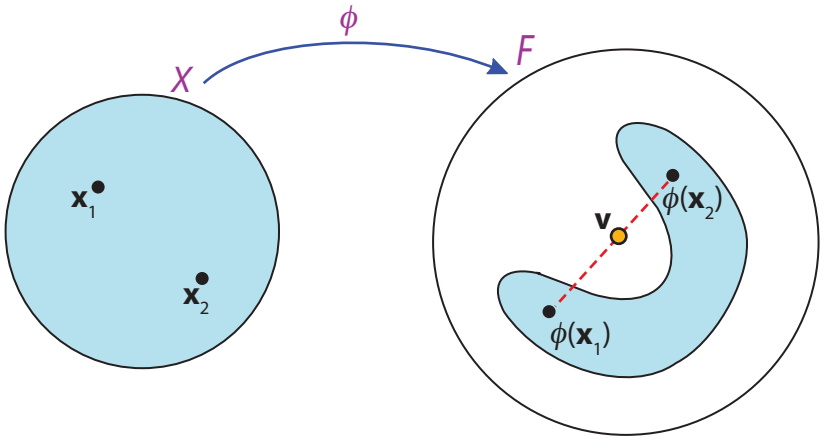


Figure 13.3 Illustrating the pre-image problem in kernel methods. The input space X is mapped by $\boldsymbol{\phi}$ to the shaded area in the much larger feature space F. Two data points $\mathbf{x}_1$ and $\mathbf{x}_2$ are mapped to $\boldsymbol{\phi}(\mathbf{x}_1)$ and $\boldsymbol{\phi}(\mathbf{x}_2)$, respectively, in F. Although $\mathbf{v}$ is a linear combination of $\boldsymbol{\phi}(\mathbf{x}_1)$ and $\boldsymbol{\phi}(\mathbf{x}_2)$, it lies outside the shaded area in F, hence there is no 'pre-image' $\mathbf{x}$ in X, such that $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{v}$. [Source: Hsieh (2009)]

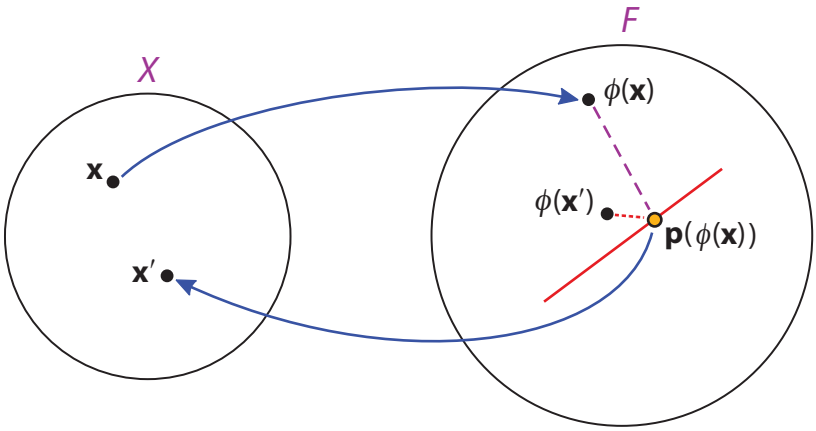


Figure 13.4 Illustrating the approach used by Mika et al. (1999) to extract an approximate pre-image in the input space X for a point $\mathbf{p}(\boldsymbol{\phi}(\mathbf{x}))$ in the feature space F. Here, for example, $\mathbf{p}(\boldsymbol{\phi}(\mathbf{x}))$ is shown as the projection of $\boldsymbol{\phi}(\mathbf{x})$ onto the direction of the first PCA eigenvector (solid line). The optimization algorithm looks for $\mathbf{x}'$ in X that minimizes the squared distance between $\boldsymbol{\phi}(\mathbf{x}')$ and $\mathbf{p}(\boldsymbol{\phi}(\mathbf{x}))$. [Source: Hsieh (2009)]

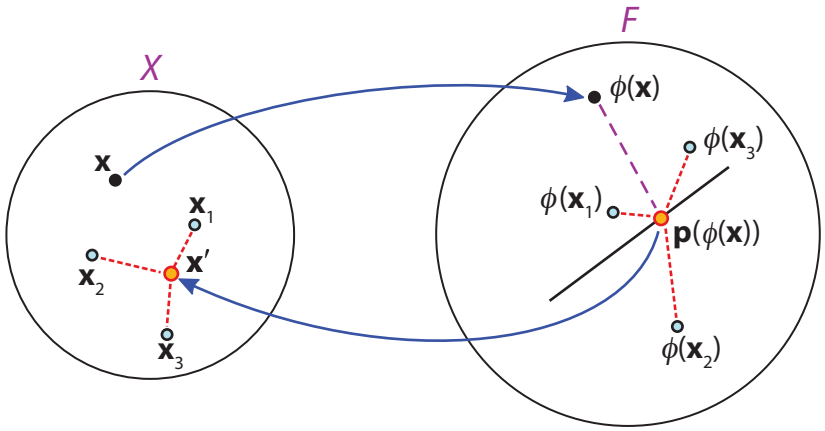


Figure 13.5 Illustrating the approach used by Kwok and I. W.-H. Tsang (2004) to extract an approximate pre-image in the input space X for a point $\mathbf{p}(\phi(\mathbf{x}))$ in the feature space F. The distance information in F between $\mathbf{p}(\phi(\mathbf{x}))$ and its several nearest neighbours (e.g. $\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \ldots$), and the relationship between distance in X and distance in F are exploited to allow $\mathbf{x}_1, \mathbf{x}_2, \ldots$ to pinpoint the desired approximate pre-image $\mathbf{x}'$ in X. [Source: Hsieh (2009)]

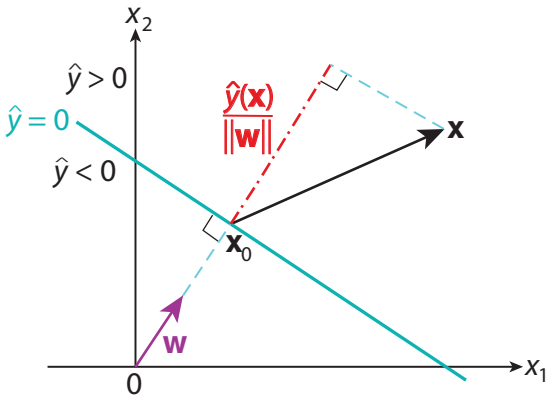


Figure 13.6 The hyperplane $\hat{y} = 0$, with the vector $\mathbf{w}$ perpendicular to this hyperplane. In this example with $\mathbf{x}$ being two-dimensional, the hyperplane $\hat{y} = 0$ reduces to a straight line in the x_1–x_2 plane. The component of the vector $\mathbf{x} - \mathbf{x}_0$ projected onto the $\mathbf{w}$ direction is shown by the dot-dashed line.

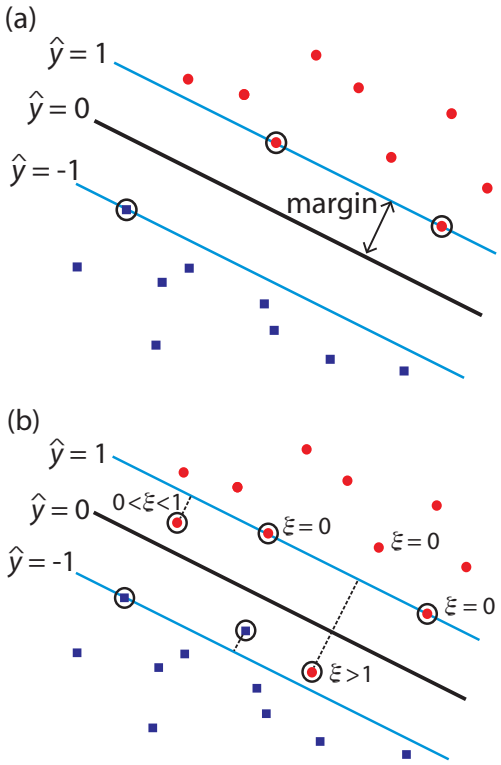


Figure 13.7 (a) A dataset containing two classes (shown by solid circles and squares) separable by a hyperplane decision boundary $\hat{y} = 0$. The margin is maximized. Support vectors, that is, data points used in determining the margins $\hat{y} = \pm 1$, are circled. (b) A dataset not separable by a hyperplane boundary. Slack variables $\xi_n \geq 0$ are introduced, with $\xi_n = 0$ for data points lying on or within the correct margin, $\xi_n > 1$ for points lying to the wrong side of the decision boundary. Support vectors are circled.

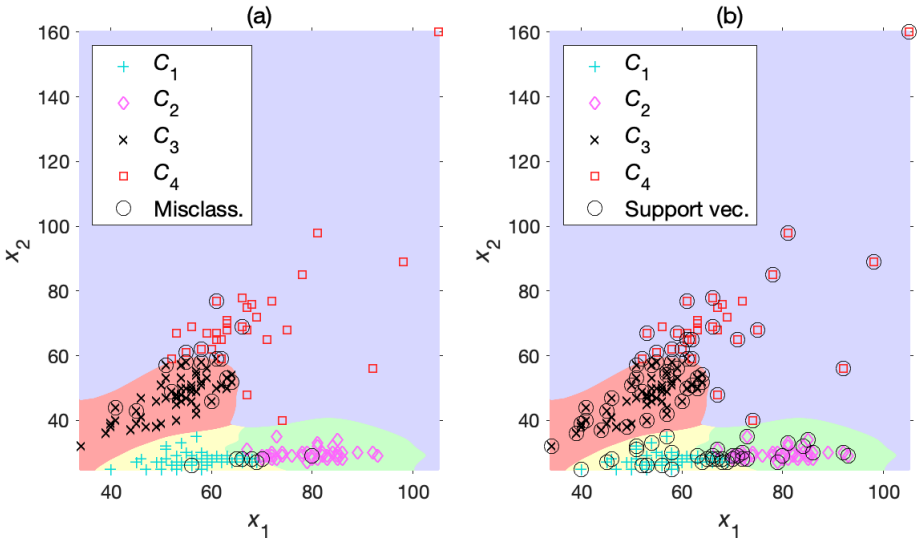


Figure 13.8 One-versus-one classification by SVM on the training dataset in Fig. 12.1(a) for types of forest cover. Class C_1 is indicated by $+$, C_2 by $\Diamond$, C_3 by $\times$ and C_4 by $\square$. Decision regions for the four classes are shown by different background shading. Circled data points in (a) are the misclassified data and in (b) the support vectors.

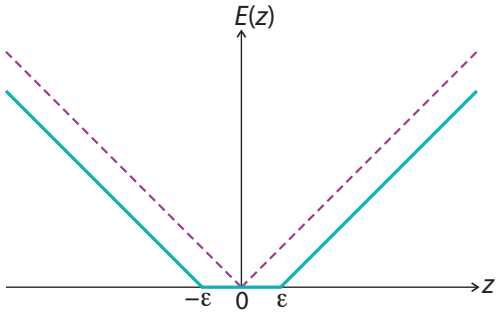


Figure 13.9 The ϵ-insensitive error function $E_\epsilon(z)$ shown by solid line and the mean absolute error (MAE) function by dashed line.

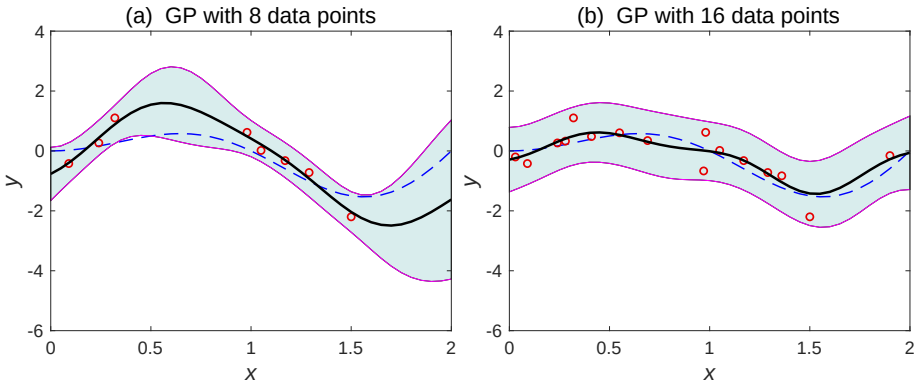


Figure 13.10 GP regression using the isotropic Gaussian kernel, with the number of data points (small circles) being (a) 8 and (b) 16. The thick curve shows the predicted mean, with the two thin curves showing the boundaries of the 95% prediction interval (i.e. ± 2 standard deviations). The true underlying signal is indicated by the dashed curve.

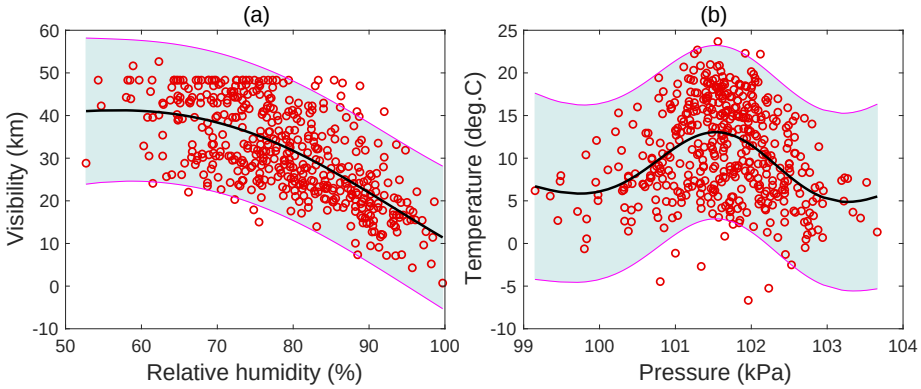


Figure 13.11 GP regression applied to daily weather variables from Vancouver, BC, Canada, with 1/20 the data from 1993–2017 used: (a) visibility as a function of relative humidity and (b) temperature as a function of pressure. The mean and variance from (13.114) and (13.115) are used to draw the thick curve and to shade the 95% prediction interval. [Data source: weatherstats.ca based on Environment and Climate Change Canada data.]

Chapter 14: Decision trees, random forests and boosting

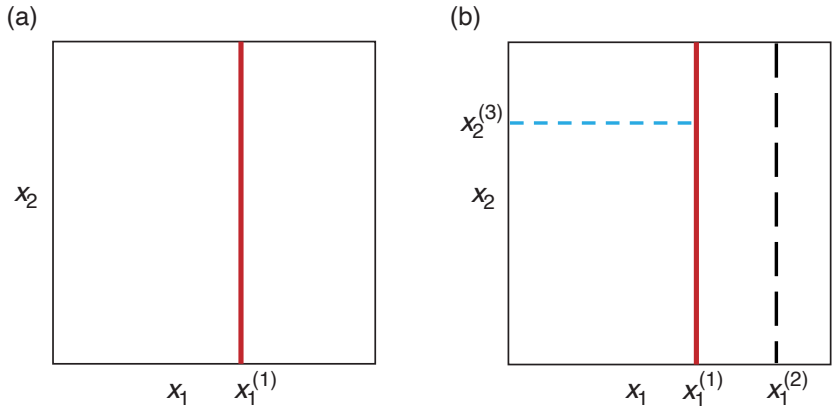


Figure 14.1 Illustrating the partitioning or splitting of the predictor $\mathbf{x}$-space by CART. (a) First split at $x_1 = x_1^{(1)}$ yields two regions, each with a constant value for the output $\hat{y}$. (b) Second split at $x_1 = x_1^{(2)}$ (long dash line) is followed by a third split at $x_2 = x_2^{(3)}$ (short dash), yielding four regions of constant $\hat{y}$ values.

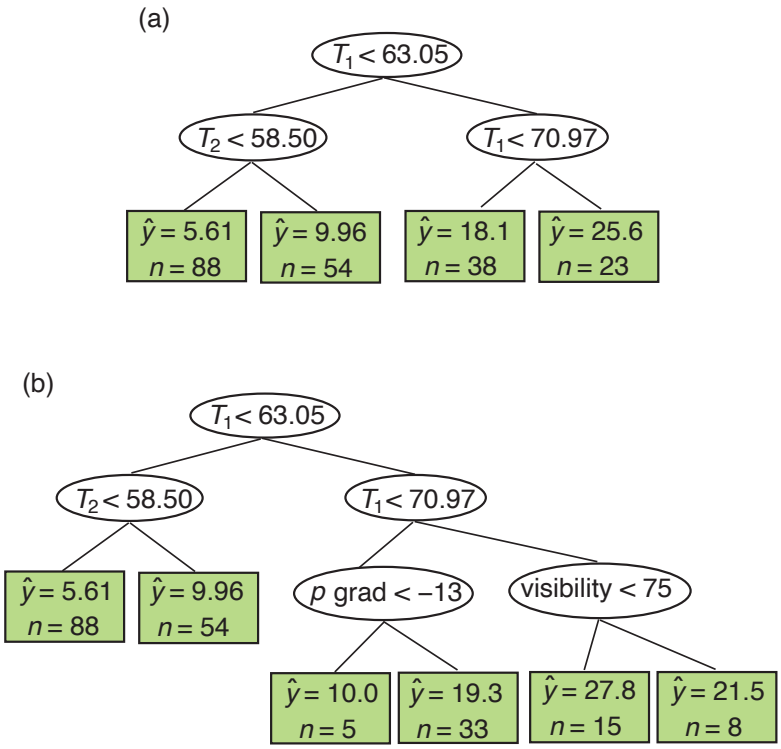


Figure 14.2 Regression tree from CART where the output $\hat{y}$ is the Los Angeles ozone level (in ppm), and there are nine predictor variables. The 'tree' is plotted upside down, with the 'leaves' (i.e. terminal nodes) drawn as rectangular boxes at the bottom and the non-terminal nodes (i.e. internal nodes) as ellipses. (a) The tree after three splits has four leaf nodes. (b) The tree after five splits has six leaf nodes. In each ellipse, a condition is given. Starting from the top ellipse, if the condition is satisfied, proceed along the left branch down to the next node; if not, proceed along the right branch. Continue until a leaf node is reached. In each rectangular box, the constant value of model output $\hat{y}$ (computed from the mean of the target data y) in the partitioned region associated with the particular leaf node is given, as well as n, the number of data points in that region. Among the nine predictor variables, the most relevant are the temperatures T_1 and T_2 (in °F) at two stations, p grad (pressure gradient in mm Hg) and visibility (in miles).

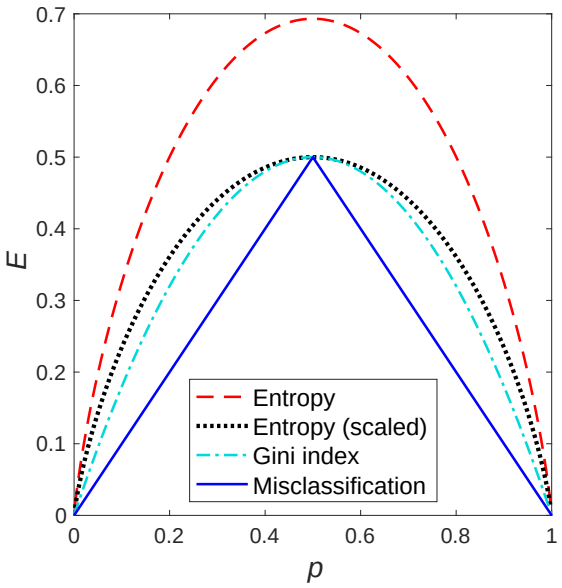


Figure 14.3 The error E for a leaf node in binary classification, where p is the fraction of data belonging to class 1. E is taken to be the entropy impurity (dashed), the Gini impurity (dot-dashed) and the misclassification rate (solid). The dotted line shows the entropy scaled to have the same maximum value as the Gini index to facilitate comparison.

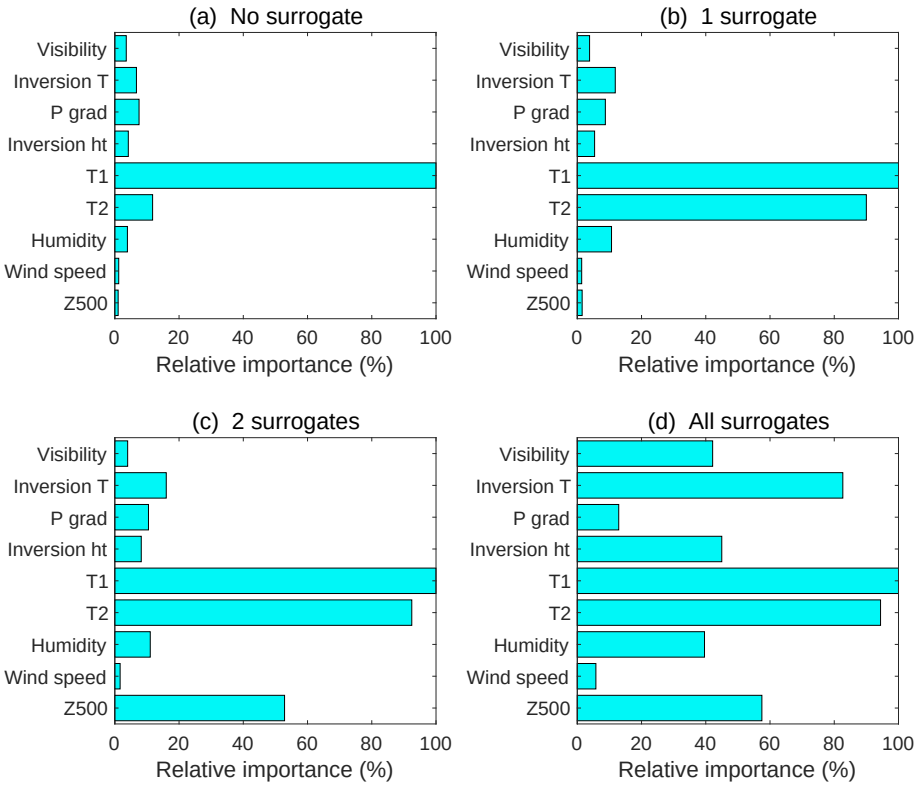


Figure 14.4 Relative importance of the predictors in the CART model for the Los Angeles ozone level, when using (a) no surrogate, (b) one surrogate, (c) two surrogates and (d) all surrogate splits.

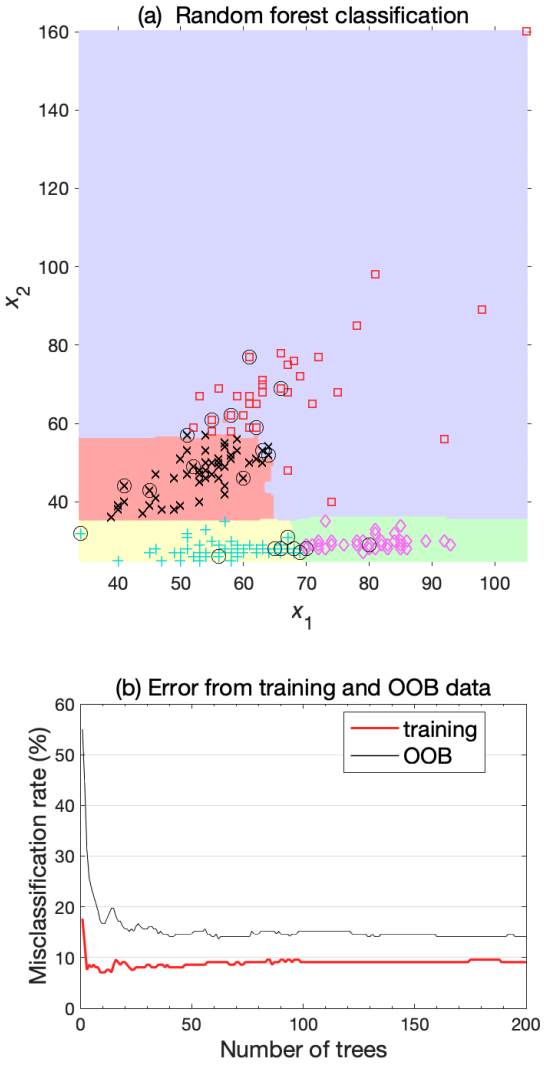


Figure 14.5 Given the training dataset in Fig. 12.1(a) for forest classification, the data are classified by a random forest model with 200 trees in (a), where class C_1 is indicated by $+$, C_2 by $\Diamond$, C_3 by $\times$ and C_4 by $\square$. Decision regions for the four classes are shown by different background shading, and misclassified data points are circled. (b) Misclassification rate for the training data and the OOB data as the number of trees increases from 1 to 200.

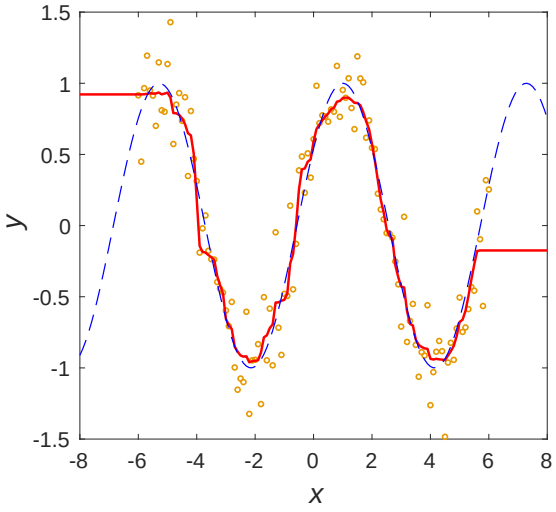


Figure 14.6 The ensemble average (solid curve) of an RF model containing 200 regression trees, the true signal (dashed) and the training data (circles).

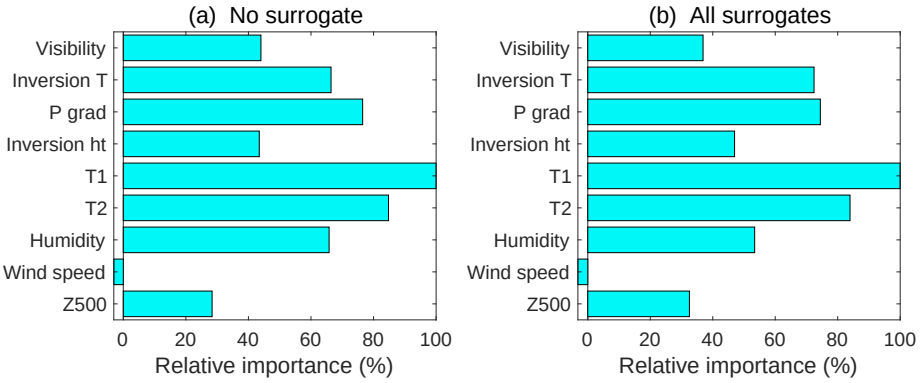


Figure 14.7 Relative importance of the predictors in the random forest regression model for the Los Angeles ozone level, when using (a) no surrogate and (b) all surrogate splits. The most important predictors are the temperatures T_1 and T_2, the pressure gradient and the inversion temperature.

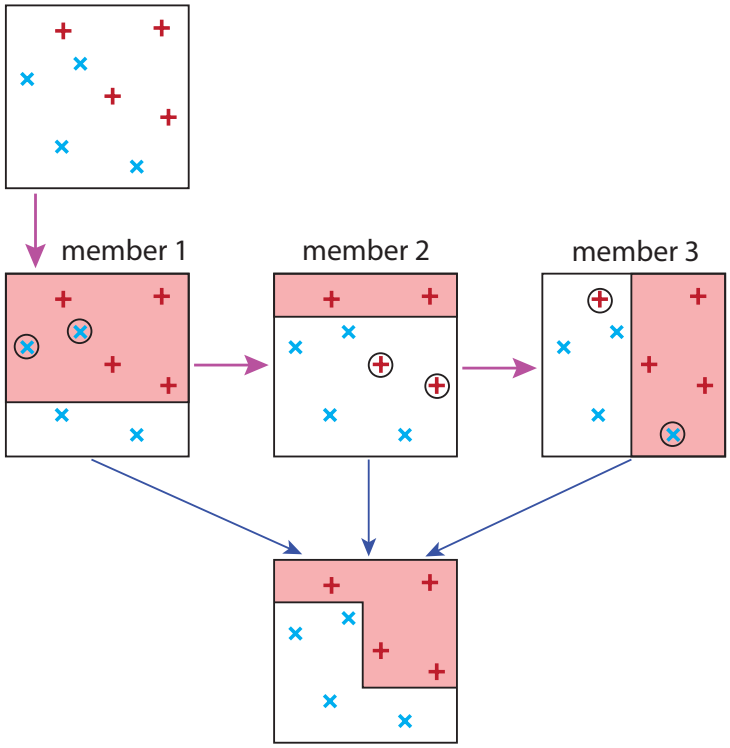


Figure 14.8 How boosting works in classification. Top box contains eight input data points belonging to two classes, '+' and '×'. Next, ensemble member 1 is built using a weak learner, a simple decision tree which splits the domain into two regions, with the shaded region predicting class '+' and the white region predicting class '×'. The shaded region contains two '×' data points, which are circled to indicate their being misclassified. More effort is devoted to improving the two misclassified points when building member 2, so they are classified correctly here, but now two '+' points are misclassified in member 2. With more effort, these two points are correctly classified in member 3, but there are another two misclassified points. Finally, majority voting by the three members gives the more complicated decision regions in the bottom box, where all eight data points are correctly classified.

Chapter 15: Deep learning

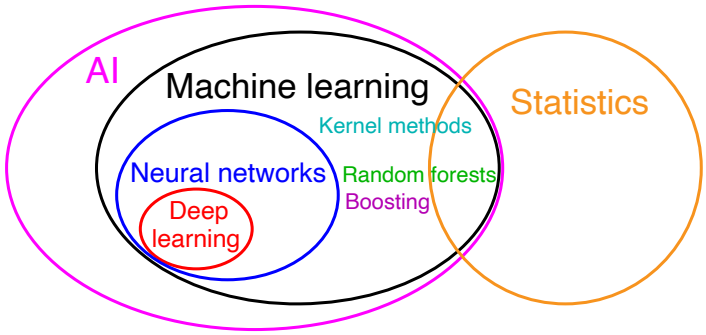


Figure 15.1 A schematic Venn diagram illustrating the relation between AI, statistics, machine learning, neural networks and deep learning, as well as kernel methods (Chapter 13), random forests and boosting (Chapter 14). [Reproduced from Hsieh (2022).]

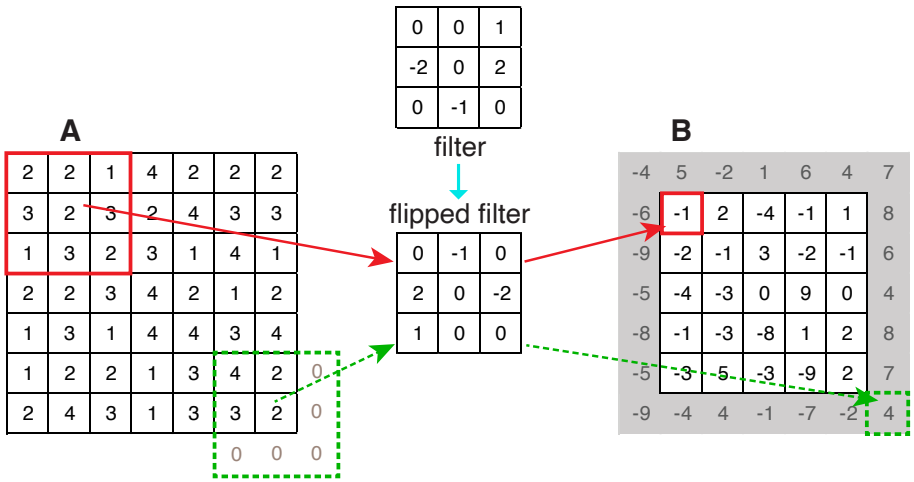


Figure 15.2 Convolution operation as illustrated by applying a 3×3 filter to a 7×7 matrix. The filter $\mathbf{F}$ is first flipped (both rows and columns) to give the flipped filter $\mathbf{F}'$. Nine elements of the input matrix $\mathbf{A}$ on the left are multiplied by the nine elements of the flipped filter, then summed and placed in the output matrix $\mathbf{B}$ on the right. If $\mathbf{B}$ is to remain the same size as $\mathbf{A}$, zeros must be padded outside the boundary of $\mathbf{A}$ to produce the extra elements shaded in grey (e.g. the dashed pixel).

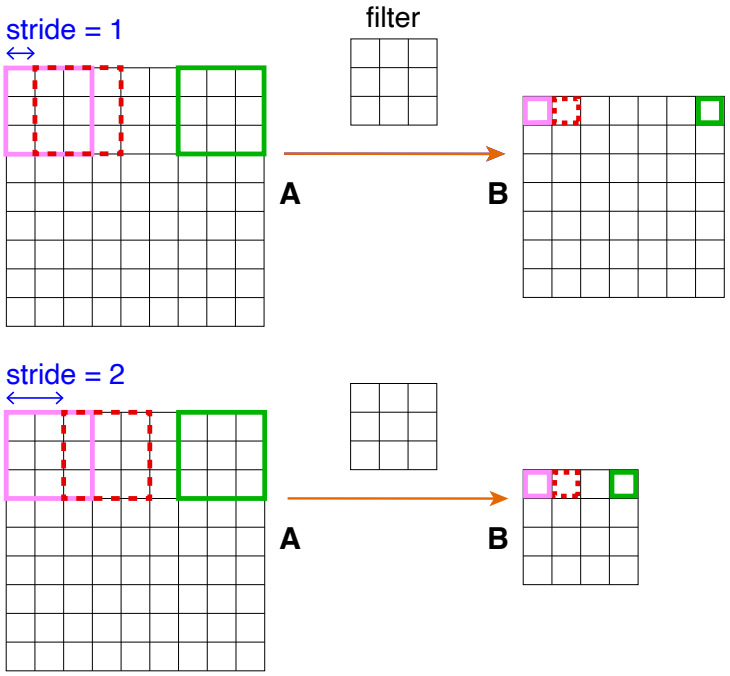
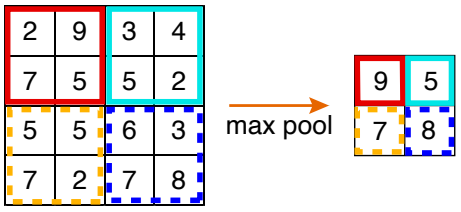


Figure 15.3 Effect of moving the (flipped) filter over the input matrix **A** at different stride s. **A** is of size 9×9 and the filter 3×3, while the output **B** is of size 7×7 (for $s = 1$) and 4×4 ($s = 2$).

Figure 15.4 Example of a 4×4 input array undergoing a max pooling operation, where the output is the maximum value from each 2×2 patch. Here, the filter width $f = 2$ and the stride $s = 2$.



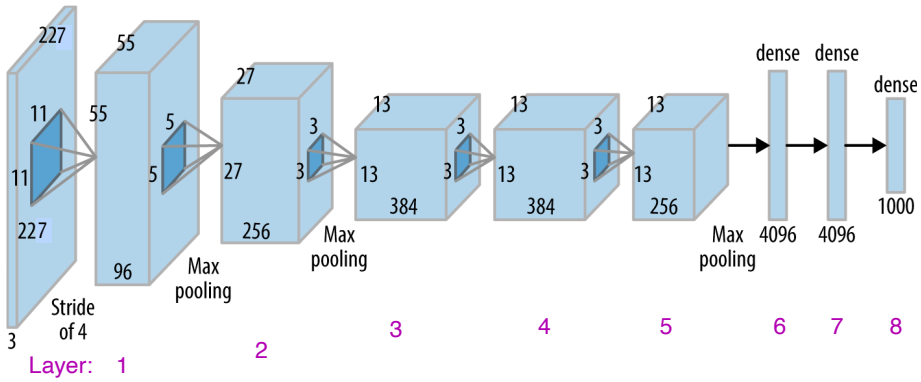


Figure 15.5 The eight-layer AlexNet CNN model has input images of 227×227 pixels and three colour channels (RGB), with the number of channels indicated by the depth (i.e. thickness) of the input block. Convoluting the input by a 11×11 filter with a stride of $s = 4$ led to an array of 55×55, and the use of 96 such filters led to 96 channels in layer 1. After max pooling, convoluting by a 5×5 filter (with $z = 2$ for zero padding) led to the layer 2 array of size 27×27, with 256 channels. All max pooling operations are done using a 3×3 filter with $s = 2$. Again max pooling, then convoluting by a 3×3 filter (with $z = 1$) led to the layer 3 array of 13×13, with 384 channels. Convoluting with a 3×3 filter (with $z = 1$) led to the layer 4 array of 13×13, with 384 channels, and convoluting again led to layer 5 of 13×13, with 256 channels. After max pooling, the resulting 6×6 array with 256 channels is reshaped into a 1-D array of 4,096 nodes and is fully or densely connected to layer 6 with 4,096 nodes, which is fully connected to layer 7 with 4,096 nodes. Layer 7 is fully connected to layer 8 with 1,000 nodes, with the softmax activation function indicating which one of the 1,000 classes (e.g. cats, dogs, cars, etc.) the output belongs to. [Adapted from *TensorFlow for Deep Learning*, by Bharath Ramsundar and Reza Bosagh Zadeh. Copyright © 2018 Reza Zadeh, Bharath Ramsundar. Published by O'Reilly Media, Inc. Used with permission.]

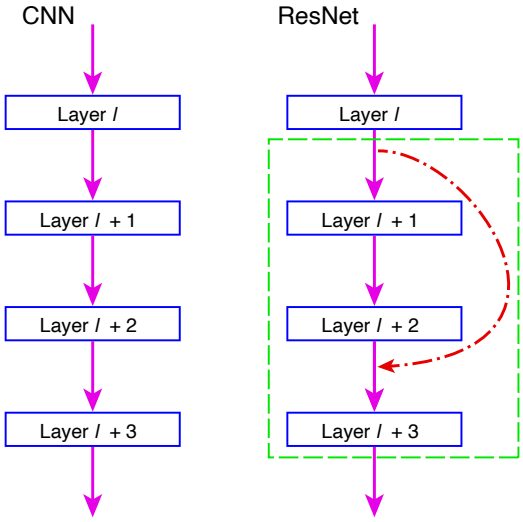


Figure 15.6 Unlike the standard CNN architecture on the left, the ResNet architecture allows skip connections (dot-dashed line) to connect the output from layer l directly to layer $l + 3$. The basic building block for residual learning (dashed) is repeated to give a deep network structure.

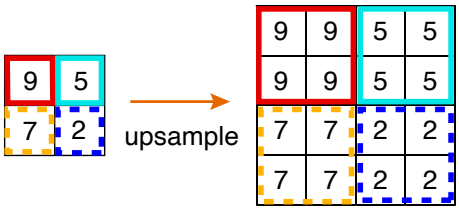


Figure 15.7 Example of a 2×2 input array using nearest neighbour upsampling to generate values for a 4×4 grid.

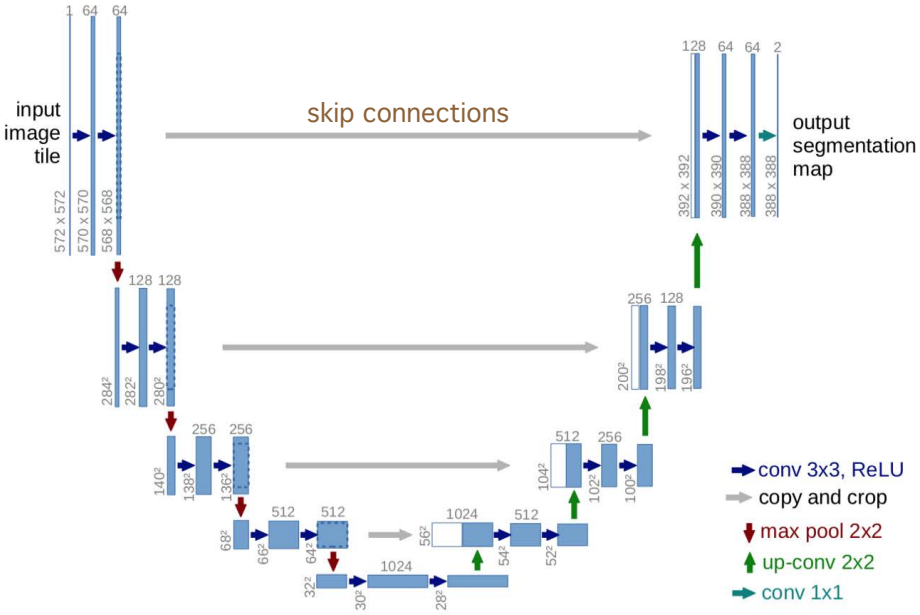


Figure 15.8 In this U-net model, the input image of 572×572 pixels passes through two convolutional layers (each with 64 channels) using 3×3 filters and the ReLU activation function, then undergoes 2×2 max pooling to a 284×284 layer with 64 channels. Descending the left arm of the 'U' structure is the encoding part, where the spatial resolution decreases but the number of channel increases, reaching a 30×30 layer with 1,024 channels at the bottom of the 'U'. From this bottleneck, ascending the right arm of the 'U' structure is the decoding part, where up-convolution (i.e. upsampling followed by 2×2 convolution) increases the spatial resolution but decreases the number of channels. Skip connections linking layers in the encoder to the corresponding layers in the decoder are used to avoid the loss of details in the output. At the final layer, a 1×1 convolution is used to map from the 64 channels to the desired number of classes (two classes in this example). In total, the network has 23 convolutional layers. [Adapted from Ronneberger et al. (2015, figure 1).]

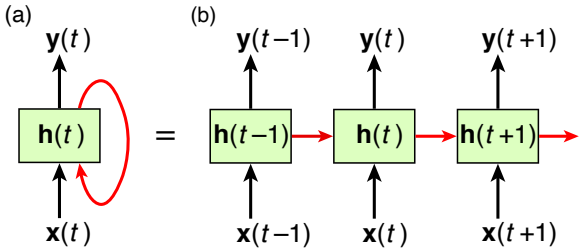


Figure 15.9 (a) Recurrent neural network (RNN) with one hidden layer $\mathbf{h}(t)$. The network can be unfolded to give the equivalent structure in (b), where $\mathbf{h}(t+1)$ receives $\mathbf{x}(t+1)$ and $\mathbf{h}(t)$ as input and the output is $\mathbf{y}(t+1)$.

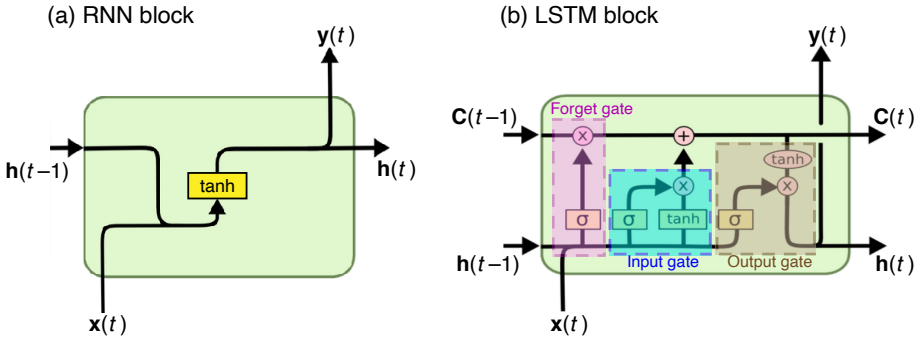


Figure 15.10 (a) A hidden layer block from the the RNN in Fig. 15.9(b) has the combined input from $\mathbf{x}(t)$ and $\mathbf{h}(t-1)$ passing through an activation function (e.g. tanh) to give $\mathbf{h}(t)$, which is further passed on to the output node $\mathbf{y}(t)$ and as input to the next block at time $t+1$. (b) A corresponding block from LSTM where the main difference is the addition of the memory cell vector $\mathbf{C}$, which stores the long-term memory to supplement the short-term memory stored in $\mathbf{h}$. There are three components inside the block: the forget gate, which decides whether to clear the long-term memory from $\mathbf{C}$; the input gate, which updates $\mathbf{C}$; and the output gate, which outputs $\mathbf{h}(t)$. Logistic sigmoidal functions σ provide smooth switching between on and off, while $\otimes$ and $\oplus$ denote element-wise multiplication and addition. The equations for the LSTM block are given in Ki et al. (2020) and Kratzert, Klotz, Brenner, et al. (2018).

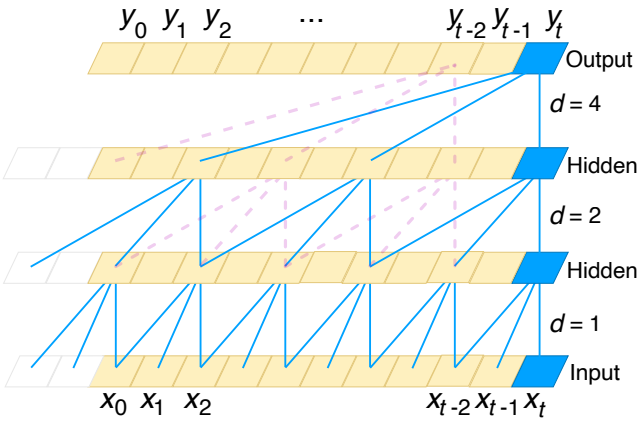


Figure 15.11 Dilated causal convolution layers with dilation factors $d = 1, 2, 4$ and filter size $f = 3$. [Adapted from Bai et al. (2018, figure 1)].

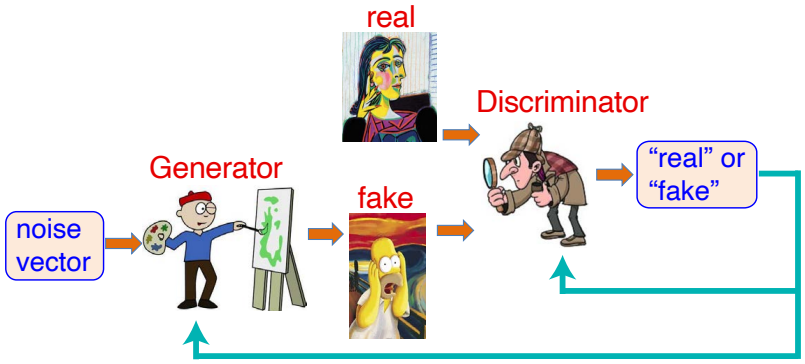


Figure 15.12 Generative adversarial network (GAN) with the generator creating a fake image (e.g. a fake Picasso painting) from random noise input, and the discriminator classifying images as either real or fake. Whether the discriminator classifies a fake image rightly or wrongly leads, respectively, to further training for the generator or for the discriminator. [Reproduced from Hsieh (2022).]

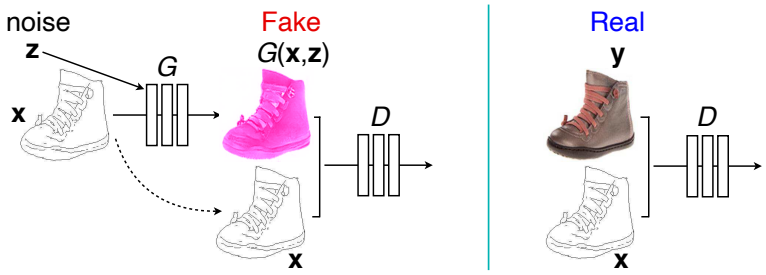


Figure 15.13 Conditional generative adversarial network (CGAN) where the generator G receives an image $\mathbf{x}$ and a random noise vector $\mathbf{z}$ as input. The discriminator D receives $\mathbf{x}$ plus either a fake image from G or a real image $\mathbf{y}$ as input. Here, a line drawing is converted to a photo image; similarly, a photo image can be converted to a line drawing. [Adapted from Isola et al. (2017, figure 2).]

Chapter 16: Forecast verification and post-processing

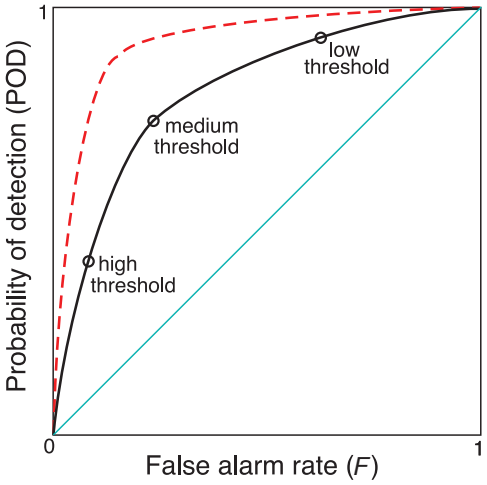


Figure 16.1 A schematic relative operating characteristic (ROC) diagram illustrating the trade-off between the false alarm rate (F) and the probability of detection (POD) as the classification decision threshold is varied for a given model (solid curve). The dashed curve shows the ROC of a better model while the diagonal line (POD = F) indicates a model with zero skill. ROC can be characterized by a single number, the *area under the curve* (AUC), where AUC is larger for the better model.

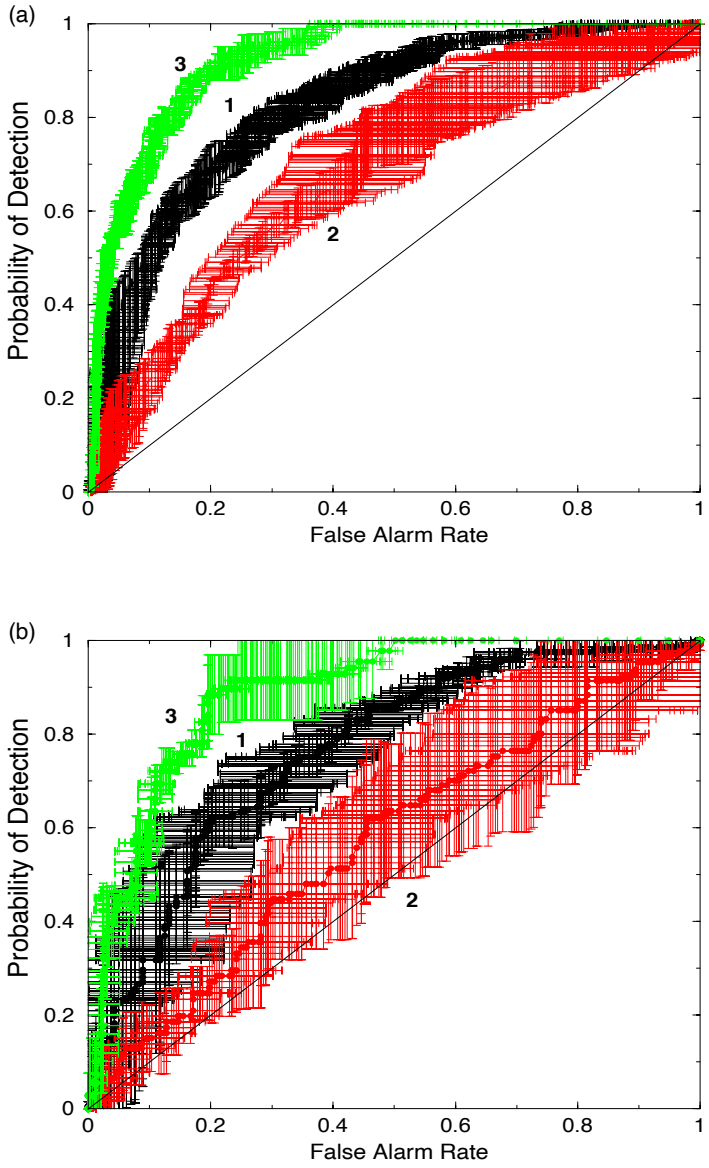


Figure 16.2 ROC diagrams for hailstone classes 1, 2 and 3, using (a) training and (b) validation data. The error bars in the horizontal and vertical directions are the one standard deviation intervals based on bootstrapping. The diagonal line indicates a model with zero skill. [Reproduced from Marzban and Witt (2001, figure 7), ©American Meteorological Society. Used with permission.]

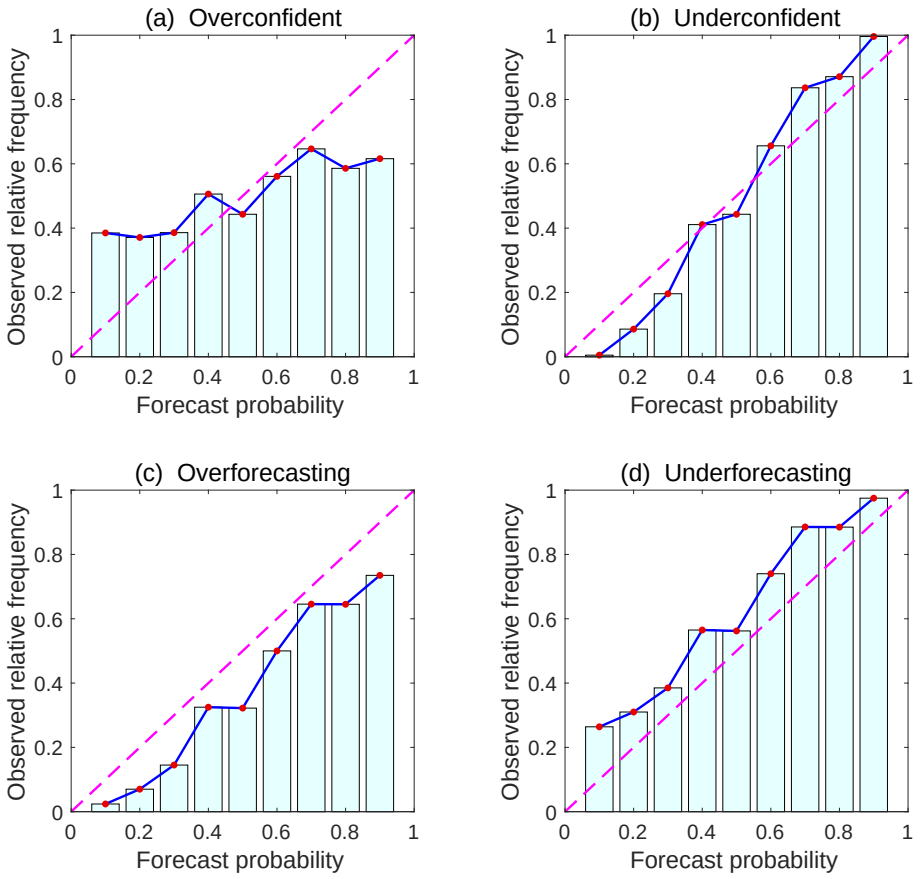


Figure 16.3 Four types of behaviour seen in *reliability diagrams*, where the observed relative frequency is plotted as a function of the forecast probability: (a) overconfident forecasts, (b) underconfident forecasts, (c) overforecasting and (d) underforecasting, with the dashed diagonal line indicating a perfect model.

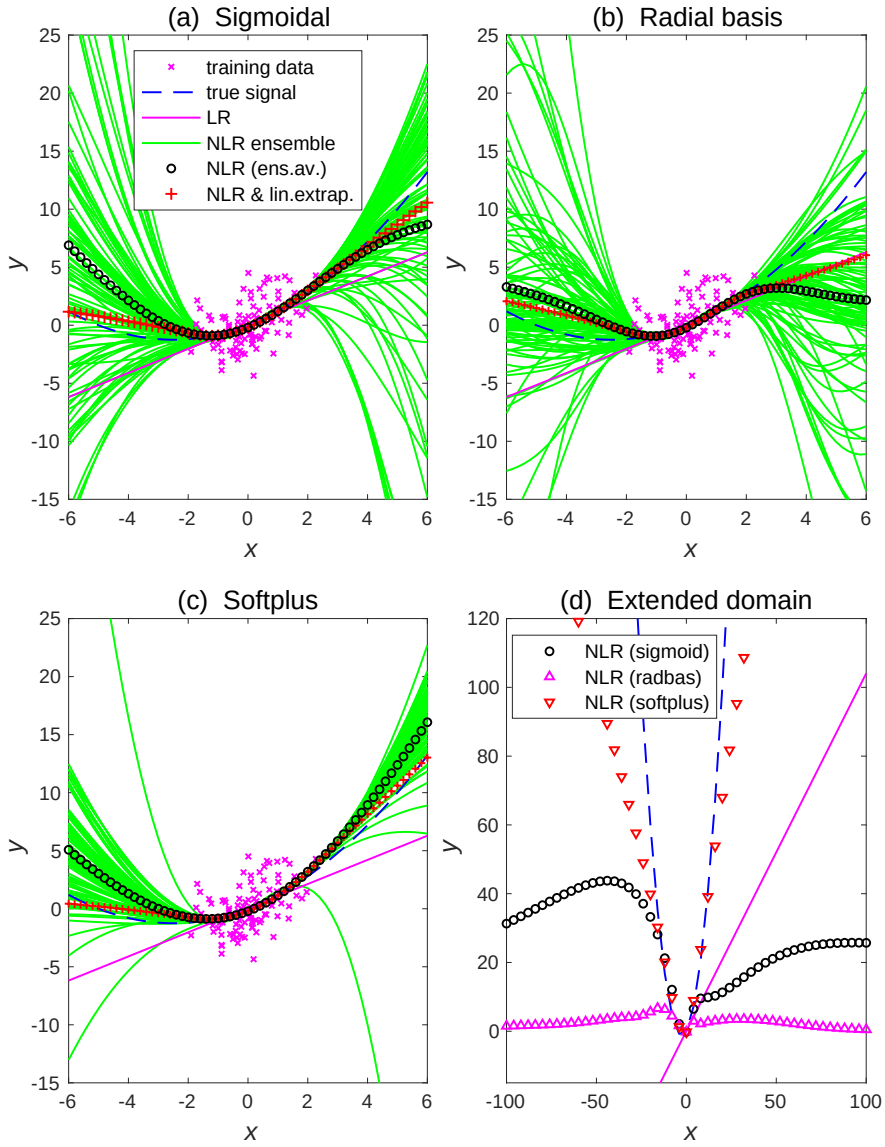


Figure 16.4 Effects of extrapolation with the (a) sigmoidal, (b) radial basis and (c) softplus activation function used in the ELM NLR model. *Linear* extrapolation of the NLR model beyond the training domain is marked by '+'. (d) shows extrapolation of the NLR model over an extended domain for the three activation functions, as well as the true signal (dashed) and LR (solid line). [Adapted from Hsieh (2020, figure 1)].

Chapter 17: Merging of machine learning and physics

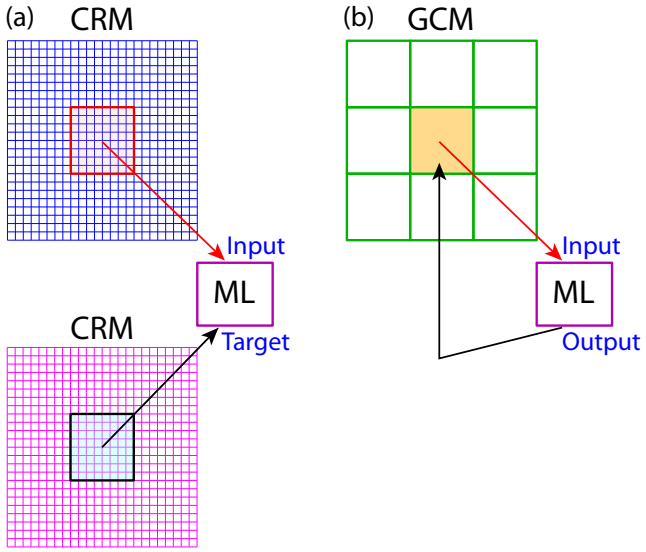


Figure 17.1 Using ML to learn parameterization from a high-resolution numerical model such as a cloud resolving model (CRM). In stage (a), high-resolution data from the CRM are coarse-grained (i.e. averaged over a number of grids to match the coarser GCM grid size), then supplied as input data and target data for training the ML model. In stage (b), the trained ML model is coupled to the GCM, with the ML output supplying moist convection and/or other parameterization to the GCM.

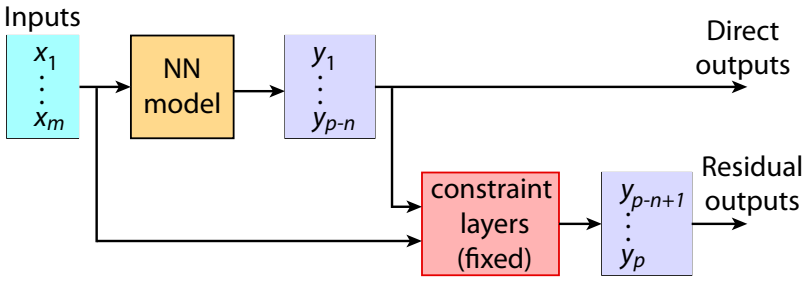


Figure 17.2 The ACnet: With predictors $x_1, \ldots, x_m$, the neural network model generates the direct outputs $y_1, \ldots, y_{p-n}$. The constraint layers take in $x_1, \ldots, x_m$ and $y_1, \ldots, y_{p-n}$, then use the physics constraints to give the residual outputs $y_{p-n+1}, \ldots, y_p$. The NN model weights are optimized by minimizing the MSE between the outputs $y_1, \ldots, y_p$ and corresponding target data. [Follows Beucler, M. Pritchard, Rasp, et al. (2021, figure 2).]

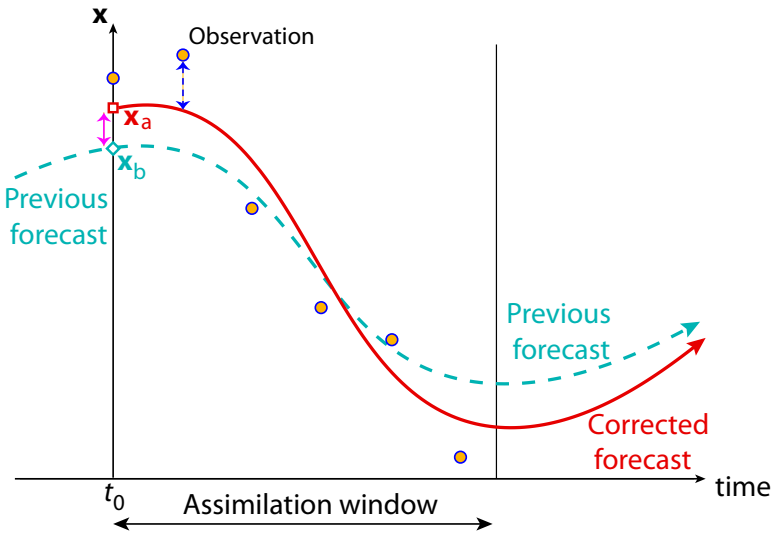


Figure 17.3 In 4D-Var, observations are assimilated over a time window starting at t_0. The solid curve, generated by integrating the dynamical model, is fitted to the observations, as well as to the background forecast $\mathbf{x}_b$ at t_0, by minimizing the objective or cost function J. The optimally estimated $\mathbf{x}_a$ at t_0 serves as the initial condition for integrating the dynamical model forward in time to generate future forecasts.