STK 4150 – Environmental and Spatial Statistics
Spring 2008

Course Notes and Exercises
by Nils Lid Hjort

– This version: as of 4 April 2008 –

1. Time series with autoregression

Suppose a time series $y_1, \ldots, y_T$ is observed, along with suitable covariate information $x_1, \ldots, x_T$, where (x_t, y_t) is associated with time points t, for $t = 1, \ldots, T$. The standard linear regression model uses

$$y_t = \beta_0 + \beta_1 x_t + \varepsilon_t \quad \text{for } t = 1, \ldots, T,$$

where $\varepsilon_1, \ldots, \varepsilon_T$ are assumed i.i.d. with mean zero and standard deviation σ, say.

(a) Standard textbook methods give parameter estimates and their standard errors, etc. Make in particular sure you know how to find a 95% confidence interval for the β_1 parameter (the influence of x_t on y_t), and discuss briefly the traditional assumptions that support these methods. In **R**, using `lm(y~x)` will provide the basics needed for evaluating parameter estimates, confidence intervals, etc., with real data.

– This exercise is however concerned with models that include *statistical dependence* in the data. We shall study the class of models given in Smith's Section 1.2, where

$$y_t = \beta_0 + \beta_1 x_t + \eta_t \quad \text{for } t = 1, \ldots, T,$$

but where the η_t now follow a so-called AR(m) (autoregressive) model, for some m. Specifically,

$$\eta_t = \phi_1 \eta_{t-1} + \cdots + \phi_m \eta_{t-m} + \varepsilon_t, \qquad (*)$$

with $\varepsilon_1, \ldots, \varepsilon_T$ being i.i.d. N$(0, \sigma^2)$ and $\phi = (\phi_1, \ldots, \phi_m)$ the vector of autoregression parameters. In all this model has $m + 3$ parameters, and the standard regression model is included as the special case $m = 0$.

(b) Simulate and display (x_t, y_t) data from the AR(2) model, where you use $(\beta_0, \beta_1) = (10.00, 0.02)$ and for example $\phi = (0.66, 0.22)$, along with a couple of choices of noise level σ. (If we think in terms of temperatures, and time is measured in years, $\beta_1 = 0.02$ corresponds to a temperature increase of two degrees over hundred years.) Try out different values of the AR parameters and the AR model order m, and for different lengths T of the time series. – One needs to make eq. $(*)$ operationally precise for the first time values, as e.g. η_1 there is ideally defined in terms of $\eta_0, \eta_{-1}, \ldots, \eta_{-(m-1)}$.

There are different ways of coping with this, the simplest of which is to set these equal to zero. We thus take $(*)$ to be valid for all of $t = 1, \ldots, T$, with this understanding of η_t values before observations started.

(c) Now extend the model by allowing also the noise level to depend on time, using

$$\sigma_t = \sigma \exp\{\gamma(x_t - \bar{x})\} \quad \text{for the standard deviation of } \varepsilon_t$$

for $t = 1, \ldots, T$, with $\bar{x} = (1/T)\sum_{t=1}^{T} x_t$. Simulate data from this extended model, for a couple of suitable values of γ. What are the effects of a perhaps small but positive γ value on the data?

2. Estimating parameters in the autoregression model

Here we give details pertaining to the maximum likelihood (ML) estimation inference method for the AR models of Exercise 1.

(a) For the AR(m) model, show that the log-likelihood function may be expressed as

$$\ell_T(\theta) = -T \log \sigma - \tfrac{1}{2} \frac{1}{\sigma^2} \sum_{t=1}^{T} \varepsilon_t^2 - \tfrac{1}{2} T \log(2\pi),$$

where $\theta = (\beta_0, \beta_1, \sigma, \phi_1, \ldots, \phi_m)$ is the parameter vector, and the ε_t values are computed via $\eta_t = y_t - \beta_0 - \beta_1 x_t$ and

$$\varepsilon_t = \eta_t - (\phi_1 \eta_{t-1} + \cdots + \phi_m \eta_{t-m}).$$

As commented upon in the previous exercise, we make this equation valid also for the first values of t by setting $\eta_t = 0$ for $t = 0, -1, \ldots, -(m-1)$.

(b) For a concrete example of generated data, with $T = 100$, $\beta = (10.00, 0.02)$, $\phi = (0.66, 0.22)$, $\sigma = 1.3579$, make an **R** programme that evaluates the log-likelihood function above, say `logL2` corresponding to the AR(2) model. Then set

$$\texttt{minuslogL2} = \texttt{function(para)}\,\{-\texttt{logL2(para)}\}$$

for the minus-logL2 function, and apply

$$\texttt{look2} = \texttt{nlm(minuslogL2, starthere, hessian = T)}$$

for a suitably chosen start position `starthere` in the parameter space. The non-linear minimisation algorithm `nlm` will then have succeeded in finding

$$\texttt{ML2} = \texttt{look2\$estimate} \quad \text{and} \quad \texttt{J2} = \texttt{look2\$hessian}\,,$$

the maximum likelihood estimates $\widehat{\theta}$ and the Hessian matrix $\widehat{J}$ for the model under study. Do this, for the data you have generated.

(c) The matrix

$$\widehat{J} = -\frac{\partial^2 \ell_T(\widehat{\theta})}{\partial\theta\partial\theta^{\mathrm{t}}},$$

which is found numerically as a by-product of the `nlm` algorithm, as indicated above, is called the *observed information matrix*. It is linked to the precision of the maximum likelihood estimates, as follows:

$$\widehat{\theta} \approx_d \mathrm{N}_{m+3}(\theta, \widehat{J}^{-1})$$

(i.e. the distribution of $\widehat{\theta}$ is approximately a multi-normal, with the right means and the indicated variance matrix). This may be made precise and proved in various ways, using large-sample (asymptotics) methodology. In particular, the diagonal elements of $\widehat{J}^{-1}$ are approximations to the variances of the parameter estimates, so

$$\texttt{se2} = \texttt{sqrt}(\texttt{diag}(\texttt{solve}(\texttt{J2})))$$

provides *standard errors* (estimated standard deviations) of each parameter estimate, in a computationally simple fashion (as long as `nlm` works). – Execute these computations for some of your generated data sets, and provide in each instance 95% confidence intervals for the influence parameter β_1 as well as for the primary autoregression parameter ϕ_1. Check the influence of e.g. time series length and noise level on the precision of your estimates.

– I note that there are pre-programmed algorithms that may be used for handling some of the AR models [to be discussed and used later], but the general strategy outlined here is powerful and more flexible – in the practical sense that variations and extensions falling outside the list of standard models might be handled as above (via programming the logL function and using `nlm`). This is in particular valid for models that use non-constant noise level, as in Exercise 1(c).

3. Competing candidate models and the AIC

Quite often different parametric models might come into consideration for a given phenomenon (and a given data set). Then there is a need to select the (in some sense) 'best model', and, even more ambitiously, to rank them, from the 'best' to the 'worst'. – Among the more popular model selectors is that of the AIC, the Akaike's Information Criterion. It consists in computing the *AIC score* for each candidate model, and then selecting the model with lowest such score. It is defined as

$$\mathrm{AIC} = -2\,\ell_{\max} + 2\,\dim,$$

featuring the maximised log-likelihood and the dimension (the number of estimated parameters) of the model. One often writes

$$\ell_n(\theta) \quad \text{and} \quad \ell_{n,\max} = \ell_n(\widehat{\theta}),$$

instead of merely ℓ and $\ell_{\max}$, to indicate that the likelihood function stems from n data points. For time series models worked with in Exercises 1 and 2, one often uses T instead of n.

For the following, assume data pairs (x_t, y_t) are observed for $t = 1, \ldots, T$. We shall try four different models, where the point is to be able to compute

(i) the parameter estimates $\widehat{\theta}$;
(ii) their standard errors $\mathrm{se}(\widehat{\theta})$;
(iii) the associated maximal log-likelihood value $\ell_{n,\max} = \ell_n(\widehat{\theta})$;
(iv) the AIC score,

for each of these. You are asked to construct computer code that manages to deal with each of the four models.

(a) Model 0 is the simplest one, the familiar linear regression model that uses

$$y_t = \beta_0 + \beta_1 x_t + \varepsilon_t,$$

with ε_t being i.i.d. and $\mathrm{N}(0, \sigma^2)$. Find for this model explicit formulae for the maximum likelihood estimators, for $\ell_{\max}$, and hence for AIC.

(b) Model 1 is the AR(1) one for the error terms, as in Exercise 1. The model dimension is $2 + 1 + 1 = 4$. Give an explicit recipe for computing the log-likelihood function and for finding its maximiser, via `nlm` in **R**, as in Exercise 2.

(b) Model 1 is the AR(1) one for the error terms, as in Exercise 1. The model dimension is $2 + 1 + 1 = 4$.

(c) Model 2 is the AR(2) one for the error terms, as in Exercise 2, with parameter $\beta_0, \beta_1, \sigma, \phi_1, \phi_2$; its dimension is 5.

(d) Finally Model 3 is the extension of the AR(2) one that uses non-homogeneous standard deviations, via

$$\sigma_t = \exp\{\gamma(x_t - \bar{x})\} \quad \text{for } t = 1, \ldots, T,$$

as per Exercise 1(c).

4. The Batmobile data: fitting and comparing models

We shall indeed encounter genuine environmetric data later on, but now we focus on a simple time series data set that is also readily available. Go to the DASL website ('The Data and Story Library') `lib.stat.cmu.edu/DASL/Stories/bat.html`, and find the 'datafile' in question. The data are `acc` [injuries and fatalities from Wednesday to Saturday nighttime accidents] and `fuel` [fuel consumption (million gallons)] in Albuquerque. The data were collected via 'Batmobiles' (Breath Alcohol Testing devices), for quarters $1, 2, 3, \ldots, 52$, ranging from 1979 to 1992.

(a) Enter these data suitably into your computer – I do this by first making a data file `batmobile-data`, using paste and copy, and then using

```
data = matrix(scan("batmobile-data", skip=7), byrow=T, ncol=3)
qtr = data[ ,1]
acc = data[ ,2]
fuel = data[ ,3]
```

Plot the variable y_t, defined as `acc/fuel`, against time x_t (quarters).

(b) Go (patiently and persistently) through Models 0, 1, 2, 3, 4 of the previous exercise, for each finding parameter estimates, their standard errors, the maximised log-likelihoods, and the AIC scores. Rank the four models.

(c) Attempt to find a model that is better than these four.

5. The multinormal distribution

'Multivariate statistics' is broadly speaking the area of statistical modelling and analysis where data exhibit dependencies. The most important multivariate distribution is the multinormal one. We say that $X = (X_1, \ldots, X_k)^{\mathrm{t}}$ is multinormal with mean vector ξ (a k-vector) and variance matrix Σ (a positive definite $k \times k$ matrix) if its density has the form

$$f(x) = (2\pi)^{-k/2}|\Sigma|^{-1/2} \exp\{-\tfrac{1}{2}(x-\xi)^{\mathrm{t}}\Sigma^{-1}(x-\xi)\} \quad \text{for } x \in I\!R^k.$$

We write $X \sim \mathrm{N}_k(\xi, \Sigma)$ to indicate this. For dimension $k = 1$ this corresponds to the traditional Gaußian $\mathrm{N}(\xi, \sigma^2)$.

(a) Show that if $X \sim \mathrm{N}_k(\xi, \Sigma)$ and A is $k \times k$ of full rank, and b a k-vector, then

$$Y = AX + b \sim \mathrm{N}_k(A\xi + b, A\Sigma A^{\mathrm{t}}).$$

Generalise to the situation where A is of dimension $m \times k$ (rather than merely $k \times k$).

(b) Show that if $X \sim \mathrm{N}_k(\xi, \Sigma)$, then indeed

$$\mathrm{E}\, X = \xi \quad \text{and} \quad \mathrm{Var}\, X = \Sigma,$$

justifying the semantic terms used above.

(c) Show that X is multinormal if and only if all linear combinations are normal. In particular, if $X \sim \mathrm{N}_k(\xi, \Sigma)$, then $a^{\mathrm{t}}X = a_1X_1 + \cdots + a_kX_k$ is $\mathrm{N}(a^{\mathrm{t}}\xi, a^{\mathrm{t}}\Sigma a)$. – We will also allow saying '$X \sim \mathrm{N}_k(\xi, \Sigma)$' in cases where Σ has less than full rank. in particular, a constant may be seen as a normal distribution with zero variance.

(d) An important property of the multinormal is that a subset of components, conditional on another subset of components, remains multinormal. Show in fact that if

$$X = \begin{pmatrix} X^{(1)} \\ X^{(2)} \end{pmatrix} \sim \mathrm{N}_{k_1+k_2}\Big(\begin{pmatrix} \xi^{(1)} \\ \xi^{(2)} \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}\Big),$$

then

$$X^{(1)} \,|\, \{X^{(2)} = x^{(2)}\} \sim \mathrm{N}_{k_1}(\xi^{(1)} + \Sigma_{12}\Sigma_{22}^{-1}(x^{(2)} - \xi^{(2)}), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}).$$

(e) How tall is Professor Hjort? Assume that the heights of Norwegian men above the age of twenty follows the normal distribution $\mathrm{N}(\xi, \sigma^2)$, with $\xi = 180$ cm and $\sigma = 9$ cm. Thus, if you have *not yet seen* or bothered to notice this particular aspect of Professor Hjort and his lectures, your point estimate of his height ought to be $\xi = 180$ and a 95% prediction interval for his height would be $\xi \pm 1.96\,\sigma$, or $[162.4, 197.6]$. – Assume now that you learn that his four brothers are actually 195 cm, 207 cm, 196 cm, 200 cm tall, and furthermore that correlations between brothers' heights in the population of Norwegian men is equal to $\rho = 0.80$. Use this information about his four brothers (still assuming that you have not noticed Professor Hjort's height) to revise your initial point estimate of Professor Hjort's height. Is he a five-percent statistical outlier in his family (i.e. outside the 95% prediction interval)?

6. Simulating from the multinormal distribution

There are special routines that manage to simulate directly from the multinormal distribution, as `mvrnorm` in **R** (preceded by `library(MASS)`, if necessary). These sometimes do not work well for high dimensions. At any rate it is useful to work out different simulation strategies for the multinormal, also for use in Gaußian processes and Gaußian random fields.

(a) Let Σ be a $k \times k$ positive definite symmetric matrix (which is equivalent to saying that it is a covariance matrix, for a suitable k-dimensional probability distribution). Let $\Sigma^{1/2}$ be any matrix square root of Σ, i.e. a symmetric matrix with the property that $\Sigma^{1/2}\Sigma^{1/2} = \Sigma$ (there may in general be several matrices with this property, see the following point). Show that when $U = (U_1, \ldots, U_k)^{\mathrm{t}}$ is a vector of independent standard normals, then

$$X = \Sigma^{1/2} U \sim \mathrm{N}_k(0, \Sigma).$$

This is accordingly a general recipe for simulating from a multinormal vector, via independent standard normals, provided one manages to compute the square root matrix numerically.

(b) By a famous linear algebra theorem, there exist a unitary (or orthonormal) matrix P (with the property that $PP^{\mathrm{t}} = I_k = P^{\mathrm{t}}P$, i.e. its transpose is its inverse) such that

$$P\Sigma P^{\mathrm{t}} = \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_k),$$

where the diagonal Λ matrix has the eigenvalues of Σ along its diagonal (in decreasing order). The P matrix and the $\lambda_1, \ldots, \lambda_k$ values are found numerically in **R** using the `eigen` operation: use

$$\texttt{lambda} = \texttt{eigen}(\texttt{Sigma}, \texttt{symmetric} = \texttt{T})\$\texttt{values},$$

$$\texttt{P} = \texttt{t}(\texttt{eigen}(\texttt{Sigma}, \texttt{symmetric} = \texttt{T})\$\texttt{vectors}),$$

and use these to define Λ. (The `symmetric=T` part is not really required, but helps numerical stability for big matrices.) Then indeed the relations above hold, and these imply $\Sigma = P^{\mathrm{t}}\Lambda P$. Show that $\Sigma^{1/2} = P^{\mathrm{t}}\Lambda^{1/2}P$ is symmetric and does the job.

7. Gaußian processes

Let $Y = \{Y(t): t \geq 0\}$ be a stochastic (or random) process. One may in general terms prove that the full probability distribution of such a process is completely specified via all its finite-dimensional distributions. In other words, if Y and Z are two random processes such that the distributions of $(Y(t_1), \ldots, Y(t_k))$ and $(Z(t_1), \ldots, Z(t_k))$ are identical, for all finite subsets $\{t_1, \ldots, t_k\}$, then Y and Z are probabilistically equivalent, with $\Pr\{Y \in A\} = \Pr\{Z \in A\}$ for all measurable sets A. When defining a stochastic process in such a way, via its finite-dimensional distributions, one must also check certain coherence criteria ('Kolmogorov's consistency conditions'), but we will not go into this here.

(a) We say that a stochastic process is *normal*, or *Gaußian*, if all its finite-dimensional distributions are multinormal. Show that it then suffices to define its *mean function* $m(t) = \mathrm{E}\, Y(t)$ and *covariance function* $k(s,t) = \mathrm{cov}\{Y(s), Y(t)\}$. Show that $k(s,t)$ must satisfy the *nonnegative definite condition*, which is that

$$a^{\mathrm{t}} \Sigma a = \sum_{i=1}^{m} \sum_{j=1}^{m} a_i a_j k(t_i, t_j) \geq 0$$

for all $t_1, \ldots, t_m$ and $a_1, \ldots, a_m$, and all finite m. (Here Σ denotes the $m \times m$ matrix of all $k(t_i, t_j)$.) Conversely one may show that a given function $k(s,t)$ satisfying this condition is really a covariance function for a Gaußian process.

(b) Let Y be a Gaußian process over say $[0, 10]$ with mean function $m(t) = 0$ and some given covariance function $k(s,t)$ – in particular, the standard deviation of $Y(t)$ is $k(t,t)^{1/2}$. Give a general recipe for simulating paths from Y, via values across a grid.

(c) Consider the particular Gaußian process Y defined over the unit interval $[0,1]$ that has mean zero and covariance function

$$k(s,t) = \min(s,t) \quad \text{for } s, t \in [0,1].$$

This is the *Wiener process* or *Brownian motion* (and it may be defined over bigger time windows than merely $[0,1]$). Simulate say ten realisations of Y, and display them in the same diagram. Use a grid of type $0, 1/m, 2/m, \ldots, m/m$, perhaps with $m = 100$ or more, and explore where the matrix square root operation (described in Exercise 6) might have its current pain limit. (In the **R** version included with my 2007 laptop, I manage to use this recipe with grid size up to say $m = 1000$ without real problems, but the computations slow down with increasing grid size m, as these involve eigenvalues and squarerooting of general symmetric $m \times m$ matrices. For $m = 100$ computations they take 1 second; for $m = 500$ they need about 10 seconds; for $m = 1000$ they need about 40 seconds; for $m = 2000$ they take about three minutes. These computations are what is needed to compute the square root matrix via eigenvalues decomposition. When this matrix is stored, the following computations required for simulating a large number of process paths take very little extra time.)

8. The Ornstein–Uhlenbeck process

The *Ornstein–Uhlenbeck process* may be defined in various ways, but its main purpose is to portray a normal process with constant variability level (whereas e.g. the Wiener process has standard deviation growing as the square root of elapsed time). For our purposes, say that $Y = \{Y(t): t \geq 0\}$ is an Ornstein–Uhlenbeck process if is Gaußian, with mean zero, and covariance function

$$k(s,t) = \mathrm{cov}\{Y(s), Y(t)\} = \exp\{-a|s-t|\} = \rho^{|s-t|}.$$

Here $\rho = \exp(-a)$ is the correlation between random process points positioned 1 time unit away from each other.

(a) Since Y is Gaußian with a given covariance function, the 'brute force' method of Exercise 20(c) may be used to simulate paths across a given grid, say along points with inter-distance $1/m$ for $m = 100$ or $m = 1000$. Use this to simulate paths of Y over the time interval $[1, 4]$.

(b) There is however an alternative definition or representation of Y that makes simulation rather easier (and more mathematically transparent), as a normalised Wiener process. Define

$$Z(t) = \frac{W(\exp(2at))}{\exp(at)} \quad \text{for } t \in I\!R.$$

Show that this is actually an Ornstein–Uhlenbeck process. Use this to simulate say 10,000 paths over the time interval $[1, 4]$, and compute mean and standard deviations for $U = \min_t Z(t)$, $V = \max_t Z(t)$, and their inter-correlation.

(c) Consider the process

$$Z(t) = (1-c)\, Y(t) + c\, W(t) \quad \text{for } 0 \leq t \leq 4,$$

where Y is an Ornstein–Uhlenbeck process independent of the Brownian motion process W, and where c is a given constant, perhaps small or zero. Simulate paths of this process. Determine the distribution of $\max_{t\in[0,4]} |Z(t)|$, for $c = 0$ and for other positive values of c. Suggest ways in which to test the statistical null hypothesis that 'the world is stable' (i.e. $c = 0$) versus the alternative hypothesis that 'the world is changing' (i.e. $c > 0$).

9. Spatial interpolation for Gaußian processes

Suppose $Z = \{Z(x): x \in D\}$ is a Gaußian process, defined over some domain D. The dimension in which Z lives matters for computations and display, but not for the main aspects of the mathematics, in what follows. We assume here, for simplicity of this initial presentation, that there is a constant mean, a constant level of variability, and a stationary and isotropic correlation function:

$$\mathrm{E}\, Z(x) = m, \quad \mathrm{sd}\{Z(x)\} = \sigma, \quad \mathrm{corr}\{Z(x), Z(x')\} = K(\|x - x'\|).$$

The $K(h)$ function satisfies the compulsory requirement of nonnegative definite-ness, as per Exercise 7.

(a) Assume that the random field Z is observed in locations $x_1, \ldots, x_n$, giving a data vector Z_d equal to $(Z(x_1), \ldots, Z(x_n))^{\mathrm{t}}$. With any new point $x \in D$, show that

$$\begin{pmatrix} Z(x) \\ Z_d \end{pmatrix} \sim \mathrm{N}_{n+1}(\begin{pmatrix} m \\ m\mathbf{1} \end{pmatrix}, \sigma^2 \begin{pmatrix} 1 & c^{\mathrm{t}} \\ c & \Omega \end{pmatrix}),$$

in which $c = c(x)$ is the vector with components $K(\|x - x_i\|)$ and Ω the $n \times n$-matrix with components $K(\|x_i - x_j\|)$, and where finally $\mathbf{1} = (1, \ldots, 1)^{\mathrm{t}}$.

(b) Use results of Exercise 5 to show that

$$\widehat{Z}(x) = \mathrm{E}\{Z(x) \,|\, \mathrm{data}\} = m + c^{\mathrm{t}}\Omega^{-1}(Z_d - m\mathbf{1}) = (1 - c^{\mathrm{t}}\Omega^{-1}\mathbf{1})m + c^{\mathrm{t}}\Omega^{-1}Z_d.$$

This 'conditional mean given everything we know' operation may be seen as the *canonical spatial interpolator*. Note that $c = c(x)$ depends on position x, of course.

(c) Show that the unbiasedness property

$$\mathrm{E}\{\widehat{Z}(x) - Z(x)\} = 0$$

holds, and also that the *prediction error*, defined as the standard deviation of the interpolator, becomes

$$\mathrm{pe}(x) = \sigma\{1 - c(x)^{\mathrm{t}}\Omega^{-1}c(x)\}^{1/2}.$$

(d) Show that the interval

$$\mathrm{CI}(x) = \widehat{Z}(x) \pm 1.96\,\mathrm{pe}(x)$$

contains the unknown and random $Z(x)$ with probability equal to 95%.

10. Spatial interpolation practical for dim = 1

A continuous random surface $Z = \{Z(x) : 0 \le x \le 5\}$ has been sampled in $n = 10$ locations x, leading to the following table of $(x, Z(x))$ values:

```
 0.27  1.11   1.20  1.70   2.16   3.23   4.13   4.25  4.52   4.92
10.39  9.76  10.01  9.82  10.00  10.57  10.31  10.08  9.62   9.55
```

(a) Plot these points in a diagram.

(b) To perform spatial interpolation, one uses a Gaußian process model as in Exercise 9, with mean $m = 10.00$, standard deviation $\sigma = 1.00$, and correlation function

$$\mathrm{corr}\{Z(x), Z(x')\} = \exp\{-\lambda|x' - x|\},$$

with $\lambda = 0.333$. Give a useful interpretation of the λ parameter. Compute and display the spatial interpolator $\widehat{Z}(x)$, in the same diagram as the ten observed points.

(c) Compute also the prediction error function $\mathrm{pe}(x)$, and display it in a diagram. Show that $\mathrm{pe}(x) = 0$ when x is equal to any of the n data locations.

(d) Using the same data, experiment a bit with different values of mean m, noise level σ, and intercorrelation parameter λ. Attempt to summarise some of your findings.

(e) The machinery above assumed that the Gaußian process used known values for m, σ, and λ. Keeping λ fixed at its 0.333 value, estimate m and σ from the data, and use these to compute and display the estimated spatial interpolator

$$Z^*(x) = \widehat{m} + c^{\mathrm{t}}\Omega^{-1}(Z_d - \widehat{m}\mathbf{1}) = (1 - c^{\mathrm{t}}\Omega^{-1}\mathbf{1})\widehat{m} + c^{\mathrm{t}}\Omega^{-1}Z_d.$$

How would you adjust the prediction error function?

(f) Finally (for now), estimate all model parameters (m, σ, λ) from the $n = 10$ data points, and use these values to compute and display the estimated interpolator

$$Z^*(x) = \widehat{m} + \widehat{c}(x)^{\mathrm{t}}\widehat{\Omega}^{-1}(Z_d - \widehat{m}\mathbf{1}) = \{1 - \widehat{c}(x)^{\mathrm{t}}\widehat{\Omega}^{-1}\mathbf{1}\}\widehat{m} + \widehat{c}(x)^{\mathrm{t}}\widehat{\Omega}^{-1}Z_d.$$

– Note that the mathematical complexity of carrying out spatial interpolation does not change much when taking the problem from dimension 1 to dimension 2 (or higher), even though burdens associated with computation, organisation and display increase in weight and pain.

11. Estimating parameters in spatial models (I)

We consider a random field $Z = \{Z(x) : x \in D\}$, where observations $Z(x_i)$ are recorded at locations x_i for $i = 1, \ldots, n$. We shall investigate estimation strategies for some of the more popular types of models for Z. It is assumed in each case below that Z is a Gaußian random field (i.e. all finite-dimensional distributions are multinormal).

(a) Assume Z has constant mean m, constant standard deviation σ, and a known correlation function $K(\|x - x'\|)$. Then

$$Z_d \sim \mathrm{N}_n(m\mathbf{1}, \sigma^2\Omega),$$

where Ω is the $n \times n$ matrix of $K(\|x_i - x_j\|)$, cf. Exercise 9. Also, Z_d is the vector of data. Show that the log-likelihood function may be expressed as

$$\ell_n(m, \sigma) = -n \log \sigma - \tfrac{1}{2} \log |\Omega| - \tfrac{1}{2}(1/\sigma^2)(Z_d - m\mathbf{1})^{\mathrm{t}}\Omega^{-1}(Z_d - m\mathbf{1}) - \tfrac{1}{2}n \log(2\pi).$$

(b) Show that the ML estimator $\widehat{m}$ of m is the same as

$$\text{minimiser of } (Z_d - m\mathbf{1})^{\mathrm{t}}\Omega^{-1}(Z_d - m\mathbf{1}),$$

which we also naturally term the weighted least squares estimator. Show also that

$$\widehat{m} = \frac{\mathbf{1}^{\mathrm{t}}\Omega^{-1}Z_d}{\mathbf{1}^{\mathrm{t}}\Omega^{-1}\mathbf{1}}.$$

(c) Show that $\widehat{m}$ is unbiased with variance $\sigma^2/\mathbf{1}^{\mathrm{t}}\Omega^{-1}\mathbf{1}$. Specialise these results to the case where data are independent.

(d) Go on to show that the ML estimator of σ is $\widehat{\sigma} = \sqrt{Q_0/n}$, where

$$Q_0 = \min_m (Z_d - m\mathbf{1})^{\mathrm{t}}\Omega^{-1}(Z_d - m\mathbf{1}) = (Z_d - \widehat{m}\mathbf{1})^{\mathrm{t}}\Omega^{-1}(Z_d - \widehat{m}\mathbf{1}).$$

Try also to show that $Q_0 \sim \sigma^2\chi^2_{n-1}$.

(e) For the little data set of Exercise 10, with known $\lambda = 0.333$, compute ML estimates, both using these exact formulae and a numerical optimiser (I find 10.0283 and 0.6810 for $\widehat{m}$ and $\widehat{\sigma}$). Explore also the extent to which different values of the correlation function parameter λ influences the ML estimates for m and σ.

(f) So far we have assumed a constant mean function for the random field. Suppose now, more generally, that there is a trend function of the type

$$\mathrm{E}\, Z(x) = \sum_{j=1}^{p} \beta_j g_j(x) = g(x)^{\mathrm{t}}\beta,$$

where the functions $g_1(x), \ldots, g_p(x)$ are known. These may be seen as carrying relevant and available covariance information, e.g. topography, etc. A simple example is the trend surface $a + b_1 x_1 + b_2 x_2$. – The consequent model for the observed data now takes then form

$$Z_d \sim \mathrm{N}_n(G\beta, \sigma^2\Omega),$$

where G is the $n \times p$ matrix with $g_1(x_i), \ldots, g_p(x_i)$ as ith row. Show that the log-likelihood function may be written

$$\ell_n(\beta, \sigma) = -n\log\sigma - \tfrac{1}{2}\log|\Omega| - \tfrac{1}{2}(1/\sigma^2)(Z_d - G\beta)^{\mathrm{t}}\Omega^{-1}(Z_d - G\beta) - \tfrac{1}{2}n\log(2\pi).$$

(g) Generalise earlier methods and results, leading to ML estimators

$$\widehat{\beta} = (G^{\mathrm{t}}\Omega^{-1}G)^{-1}G^{\mathrm{t}}\Omega^{-1}Z_d$$

and $\widehat{\sigma} = (Q_0/n)^{1/2}$, with

$$Q_0 = \min_\beta (Z_d - G\beta)^{\mathrm{t}}\Omega^{-1}(Z_d - G\beta) = (Z_d - G\widehat{\beta})^{\mathrm{t}}\Omega^{-1}(Z_d - G\widehat{\beta}).$$

Find the explicit distributions for $\widehat{\beta}$ and $\widehat{\sigma}$ (under the normal model). It is assumed that $n \geq p$ and that the $n \times p$ matrix G has full rank p.

12. Estimating parameters in spatial models (II)

The results reached in the previous exercise may be seen and interpreted as saying that as long as the correlation function is known, theory proceeds more or less 'as normal': one may give exact inference recipes for quantities related to m and σ, including exact confidence intervals, etc. The situation is more complicated when there are unknown parameters in the correlation function, however.

(a) Assume that the correlation function is $K_\lambda(\|x - x'\|)$, for some λ, e.g. as in the favourite case $\exp(-\lambda\|x - x'\|)$. Let otherwise the model have trend function $g(x)^{\rm t}\beta$ and constant standard deviation σ, as in Exercise 11. Show that the log-likelihood function becomes

$$\ell_n(\beta, \sigma, \lambda) = -n \log \sigma - \tfrac{1}{2} \log |\Omega(\lambda)| - \tfrac{1}{2}(1/\sigma^2)(Z_d - G\beta)^{\rm t}\Omega(\lambda)^{-1}(Z_d - G\beta) - \tfrac{1}{2} n \log(2\pi),$$

where the λ is present in the correlation matrix $\Omega(\lambda)$.

(b) For computing the ML estimates $(\widehat{\beta}, \widehat{\sigma}, \widehat{\lambda})$ one may throw the log-likelihood function to a brute-force optimiser (like `nlm` in **R**); this will typically work fine, with a reasonable start position for the optimiser in question. One may also utilise the structure found above, for the form of the ML estimators for β and σ, given the value of λ. This simplifies the numerical task, and stabilises precision. For each candidate λ, define

$$\widehat{\beta}(\lambda) = \{G^{\rm t}\Omega(\lambda)^{-1}G\}^{-1}G^{\rm t}\Omega(\lambda)^{-1}Z_d$$

and $\widehat{\sigma}(\lambda) = \{Q_0(\lambda)/n\}^{1/2}$, where

$$Q_0(\lambda) = \min_\beta (Z_d - G\beta)^{\rm t}\Omega(\lambda)^{-1}(Z_d - G\beta) = \{Z_d - G\widehat{\beta}(\lambda)\}^{\rm t}\Omega(\lambda)^{-1}\{Z_d - G\widehat{\beta}(\lambda)\}.$$

The ML estimates are then

$$\widehat{\beta} = \widehat{\beta}(\widehat{\lambda}) \quad \text{and} \quad \widehat{\sigma} = \widehat{\sigma}(\widehat{\lambda}),$$

where $\widehat{\lambda}$ is the maximiser of the *profile log-likelihood* function

$$\ell_{n,\text{prof}}(\lambda) = \ell_n(\widehat{\beta}(\lambda), \widehat{\sigma}(\lambda), \lambda).$$

Show that finding the ML for λ is equivalent to minimising

$$n \log \widehat{\sigma}(\lambda) + \tfrac{1}{2} \log |\Omega(\lambda)|.$$

(c) For the toy data set of Exercise 9, estimate the three parameters in the model (I find 10.0584, 0.3451, 2.2854 for $\widehat{m}$, $\widehat{\sigma}$, $\widehat{\lambda}$). Use both numerical methods.

(d) The distribution of $\widehat{\beta}(\lambda)$ is exactly a normal, for each given λ value. Explain why the distribution of the ML estimator $\widehat{\beta}$ is nevertheless not normal. You may set up a simple simulation experiment to demonstrate this, using the $x_1, \ldots, x_{10}$ values of Exercise 9, where you use as true model some given values of m, σ, λ. The point is to simulate perhaps 10,000 values of $\widehat{m}$, and to show in which ways their distribution is not quite normal.

13. Kriging: constant mean

We go back to the situation of Exercise 9, involving a stationary random field with mean m, standard deviation σ, and a given correlation function $K(\|x - x'\|)$. Using a Gaußian assumption, we derived there the formula

$$\widehat{Z}(x) = c(x)^{\mathrm{t}}\Omega^{-1}Z_d + \{1 - c(x)^{\mathrm{t}}\Omega^{-1}\mathbf{1}\}m \tag{1}$$

for the natural spatial interpolator, assuming m, σ, K known. – There is another route to this and similar formulae, known is *Kriging*, which we survey now.

(a) In the *first formulation*, we ask which direct linear combination of data

$$Z^*(x) = \sum_{i=1}^{n} b_i Z(x_i) = b^{\mathrm{t}}Z_d$$

manages to minimise the squared prediction error, under the constraint of unbiasedness. Show that these wishes are equivalent to the minimisation of

$$\mathrm{E}\,\{Z^*(x) - Z(x)\}^2 = \sigma^2(b^{\mathrm{t}}\Omega b - 2b^{\mathrm{t}}c + 1)$$

under constraint $b^{\mathrm{t}}\mathbf{1} = \sum_{i=1}^{n} b_i = 1$. Here c and indeed b depend on x, which at the moment is fixed.

(b) To solve the minimisation problem we use the *Lagrange multiplier method*: minimising $H(b) = b^{\mathrm{t}}\Omega b - 2b^{\mathrm{t}}c + 1$ (of n variables) under constraint $b^{\mathrm{t}}\mathbf{1} - 1 = 0$ is equivalent to minimising the bigger function

$$\widetilde{H}(b) = H(b) + 2\nu(b^{\mathrm{t}}\mathbf{1} - 1)$$

(of $n+1$ variables) without constraints. The 2ν factor is called a Langrange multiplier, and we may use $\nu' = 2\nu$ if we wish (but the 2ν formulation leads to slightly easier calculations and expressions). Show that

$$\partial\widetilde{H}(b)/\partial b = 2\Omega b - 2c + 2\nu\mathbf{1},$$

and that this leads to

$$b = \Omega^{-1}(c - \nu\mathbf{1}) \quad \text{with} \quad \nu = \frac{c^{\mathrm{t}}\Omega^{-1}\mathbf{1} - 1}{\mathbf{1}^{\mathrm{t}}\Omega^{-1}\mathbf{1}}.$$

(c) Then show that this implies

$$Z^*(x) = c(x)^{\mathrm{t}}\Omega^{-1}Z_d + \{1 - c(x)^{\mathrm{t}}\Omega^{-1}\mathbf{1}\}\frac{\mathbf{1}^{\mathrm{t}}\Omega^{-1}Z_d}{\mathbf{1}^{\mathrm{t}}\Omega^{-1}\mathbf{1}}, \tag{2}$$

which is not fully equivalent to (1), but rather of the same form as (1), but with $\widehat{m}$ inserted for m, where $\widehat{m} = \mathbf{1}^{\mathrm{t}}\Omega^{-1}Z_d/\mathbf{1}^{\mathrm{t}}\Omega^{-1}\mathbf{1}$ is the weighted least squares estimator found in Exercise 11. So, arguably, this problem formulation has the advantage that it does not need m, or σ, as known input values; the m parameter is implicitly estimated 'on the go'.

(d) There is also a *second formulation* of the Kriging problem, where the linear combination in question is allowed to include an intercept parameter. Thus one now searches over $n+1$ coefficients a and b such that $Z^*(x) = a + b^{\mathrm{t}} Z_d$ achieves minimal squared prediction error, which again is $\sigma^2(b^{\mathrm{t}}\Omega b - 2b^{\mathrm{t}}c + 1)$, under the constraint that $Z^*(x) - Z(x)$ has mean zero, which means $a + b^{\mathrm{t}}\mathbf{1}m - m = 0$. Find this second version formula of the Kriging strategy.

14. Kriging: linear trend surface

The previous setup shall now be generalised to the case where there is a trend function surface of the type $\mathrm{E}\, Z(x) = g(x)^{\mathrm{t}}\beta$, involving covariate information functions $g_1(x), \ldots, g_p(x)$, cf. Exercises 11–12. The first formulation of the Kriging problem then becomes: which direct linear combination $Z^*(x) = b^{\mathrm{t}} Z_d$ manages to minimise the squared prediction error, under the constraint of unbiasedness?

(a) Show that this again corresponds to minimising $H(b) = b^{\mathrm{t}}\Omega b - 2b^{\mathrm{t}}c + 1$, but now under more extensive side conditions, namely

$$\mathrm{E}\,\{Z^*(x) - Z(x)\} = b^{\mathrm{t}}G\beta - g(x)^{\mathrm{t}}\beta = 0,$$

which when required to hold for all β is equivalent to the p equations

$$b^{\mathrm{t}}G = g(x)^{\mathrm{t}}, \quad \text{or } G^{\mathrm{t}}b = g(x).$$

The Lagrange problem is to minimise the bigger function

$$\widetilde{H}(b) = H(b) + 2\nu^{\mathrm{t}}(G^{\mathrm{t}}b - g)$$

(writing g for $g(x)$), as a function of $n+p$ parameters, with $\nu = (\nu_1, \ldots, \nu_p)^{\mathrm{t}}$ a vector of Lagrange multipliers. Show that

$$b = \Omega^{-1}(c + G\nu),$$

and that the side condition fulfilled demands

$$\nu = (G^{\mathrm{t}}\Omega^{-1}G)^{-1}(g - G^{\mathrm{t}}\Omega^{-1}g).$$

(b) Show that these ingredients finally lead to the following spatial interpolator, at position x:

$$Z^*(x) = c(x)^{\mathrm{t}}\Omega^{-1}Z_d + \{g(x) - G^{\mathrm{t}}\Omega^{-1}c(x)\}^{\mathrm{t}}(G^{\mathrm{t}}\Omega^{-1}G)^{-1}G^{\mathrm{t}}\Omega^{-1}Z_d.$$

(c) Use results of Exercise 11 to show that this result is almost the same as the natural model-based interpolator

$$\begin{aligned}\widehat{Z}(x) &= \mathrm{E}\{Z(x) \,|\, \text{data}\} \\ &= g(x)^{\mathrm{t}}\beta + c(x)^{\mathrm{t}}\Omega^{-1}(Z_d - G\beta) = c(x)^{\mathrm{t}}\Omega^{-1}Z_d + \{g(x) - G^{\mathrm{t}}\Omega^{-1}c(x)\}^{\mathrm{t}}\beta.\end{aligned}$$

More precisely, the Kriging interpolator $Z^*(x)$ is equal to the estimated version of the model-based interpolator $\widehat{Z}(x)$, with the weighted least squares estimator $\widehat{\beta}$ replacing β. Note that this properly generalises results of Exercise 13.

15. Correlation and covariance functions

Let $C(h) = \mathrm{cov}\{Z(x), Z(x+h)\}$ be the covariance function of some stationary random function $Z(\cdot)$.

(a) When $a = (a_1, \ldots, a_m)^{\mathrm{t}}$ is any vector, and $x_1, \ldots, x_m$ any locations, show that

$$\mathrm{Var}(a^{\mathrm{t}} Z_d) = \mathrm{Var}\Big\{\sum_{j=1}^{m} a_j Z(x_j)\} = a^{\mathrm{t}}\Omega a = \sum_{j,k} a_j a_k K(x_j - x_k).$$

Here Ω is the associated $m \times m$ matrix of covariances. Hence a basic requirement is that this quadratic form needs to be nonnegative, for all m, all locations, and all coefficients. This property is called *nonnegative definiteness*.

(b) Suppose that the function $C(\cdot)$ may be represented in the form

$$C(h) = \int \cos(h^{\mathrm{t}}\omega)\,\mathrm{d}G(\omega),$$

where $h^{\mathrm{t}}\omega = h_1\omega_1 + \cdots + h_p\omega_p$ in the appropriate dimension, and where G is some finite measure. Then show that $C(\dot{)}$ has the nonnegative definite property. – Hint: First argue that G may be symmetrised, which implies

$$C(h) = \int \exp(ih^{\mathrm{t}}\omega)\,\mathrm{d}G(\omega),$$

where $\exp(iu) = \cos u + i \sin u$, in the complex plane. Use this to show that the quadratic form is nonnegative.

– *Bochner's theorem* says that the converse is also true: $C(\cdot)$ is nonnegative if and only if it may be represented as above, for a suitable finite measure G. If G admits a density, i.e. $\mathrm{d}G(\omega) = g(\omega)\,\mathrm{d}\omega$ for some g with finite integral, then the representation is

$$C(h) = \int \cos(h^{\mathrm{t}}\omega) g(\omega)\,\mathrm{d}\omega.$$

Hence each integrable g we have a valid covariance function.

(c) For the one-dimensional case, show that in fact

$$\int \cos(h\omega) g_0(\omega)\,\mathrm{d}\omega = \exp(-|h|), \quad \text{for} \quad g_0(\omega) = \frac{1}{\pi}\frac{1}{1+\omega^2}.$$

The g_0 is the Cauchy density. Show also that

$$\exp(-\lambda|h|) = \int \cos(hv) g_0(v/\lambda)\,\mathrm{d}v/\lambda,$$

for each positive λ, so the geostatistician's favourite choice $\exp(-\lambda|h|)$ is indeed a valid covariance function.

16. Kriging in dimension two

We consider a random field

$$Z = \{Z(x_1, x_2) : (x_1, x_2) \in D\}, \quad \text{where } D = [0, 10] \times [0, 10],$$

assumed to follow a stationary distribution with unknown mean m, unknown standard deviation σ, but known correlation function

$$K(x, x') = \exp(-\lambda \|x - x'\|) \quad \text{for } x, x' \in [0, 10]^2.$$

For the initial computations below, take $\lambda = 0.888$ to be known. This random surface is observed in $n = 10$ locations, yielding the table below:

x1	x2	z
0.847	8.706	3.430
5.647	8.298	3.588
1.615	2.172	2.257
4.377	0.296	2.129
7.040	2.657	3.942
4.591	2.220	2.327
8.194	4.679	3.892
7.527	6.262	4.176
1.901	6.668	2.315
6.432	3.947	4.802

(a) Compute the BLUE (best linear unbiased estimate) for m,

$$\widehat{m} = \frac{\mathbf{1}^{\mathrm{t}} \Omega^{-1} Z_d}{\mathbf{1}^{\mathrm{t}} \Omega^{-1} \mathbf{1}},$$

where as earlier $\mathbf{1}$ is the vector of 1s and Ω the correlation of matrix $K(x_i, x_j)$ (where $x_i = (x_{i,1}, x_{i,2})$ etc.). Note that $\widehat{m}$ may be computed without knowledge of σ. I find $\widehat{m} = 3.2341$.

(b) Verify that the Kriging interpolator takes the form

$$Z^*(x) = c(x)^{\mathrm{t}} \Omega^{-1} Z_d + \{1 - c(x)^{\mathrm{t}} \Omega^{-1} \mathbf{1}\} \widehat{m},$$

for $x = (x_1, x_2) \in D$. Compute this interpolator across a grid of positions in the D domain; specifically, for

$$x_1 \in \{0.0, 0.1, 0.2, \ldots, 9.9, 10.0\} \quad \text{and} \quad x_2 \in \{0.0, 0.1, 0.2, \ldots, 9.9, 10.0\}.$$

Organise the result in a 101×101 matrix of $Z^*(x, x')$ values.

(c) Then display the Kriging interpolator in the form of a contour map:

```
contour(x1val, x2val, Zhatval,
    xlab="x1", ylab="x2", xlim=c(0,10), ylim=c(0,10))
```

(d) In a different contour map, display the (estimated) prediction error surface

$$\mathrm{pe}(x_1, x_2) = \left[\mathrm{E}\{Z^*(x_1, x_2) - Z(x_1, x_2)\}^2\right]^{1/2}.$$

For this you also need an estimate of σ.

(e) Estimate also λ from this dataset, and re-do the spatial interpolation with this value of correlation function parameter.

(f) You should programme your computations in a manner that makes it easy to switch from one correlation function to another. Experiment a bit with how the Kriging surface is influenced by a couple of different correlation functions.

17. Simulating from a random field, conditional on data

Suppose as on some earlier occasions that $Z = \{Z(x): x \in D\}$ is a random field, with constant mean m, constant standard deviation σ, and stationary correlation function $K(x-x')$. In this exercise we take m, σ, K to be fully known. Also, the random surface is assumed to be Gaußian, and is observed in locations $x_1, \ldots, x_n$.

(a) Give an explicit formula for

$$\mu_n(x) = \mathrm{E}\{Z(x) \,|\, \text{data}\}.$$

One version of this formula is

$$\mu_n(x) = c(x)^{\mathrm{t}}\Omega^{-1}Z_d + \{1 - c(x)^{\mathrm{t}}\Omega^{-1}\mathbf{1}\}m,$$

with $c(x)$ the vector of $K(x-x_i)$ and Ω the $n \times n$ matrix of $K(x_i-x_j)$; see for example Exercise 9.

(b) We shall now also need a formula for

$$C_n(x,x') = \mathrm{cov}\{Z(x), Z(x') \,|\, \text{data}\}.$$

Use results from Exercise 5 to prove that

$$C_n(x,x') = \sigma^2\{K(x-x') - c(x)^{\mathrm{t}}\Omega^{-1}c(x')\}.$$

Note that this properly generalises a formula from Exercise 9(c). Show in particular that the variance of $Z(x)$, given data, is equal to zero whenever x is one of the data locations x_i.

(c) For the baby dataset of Exercise 10, but this time using $\lambda = 3.333$ in the correlation function, simulate say 100 realisations of $Z(x)$, across the $[0,5]$ range, from the appropriate distribution of $Z(\cdot)$ given the observed data. Display these in a diagram, along with the $\mu_n(x)$ curve.

(d) Compute the probability

$$p = \Pr\{\max_{0\le x\le 5} Z(x) \ge 12.50 \,|\, \text{data}\}.$$

Also display a good histogram of the full distribution of $\max_{0\le x\le 5} Z(x)$ given the data.

(e) Consider the average value of $Z(\cdot)$ over a sub-interval, say

$$\theta = \frac{1}{b-a}\int_a^b Z(x)\,\mathrm{d}x.$$

Find and display the distribution of θ, conditional on the data values, using simulations, for the case of $[a,b] = [2,4]$. Attempt also to find the exact distribution of θ (again, conditional on the data).

18. Markov chains

Let $X_0, X_1, \ldots$ be a sequence of random variables. We say that these form a *Markov process* if 'today, given the past, does only depend on yesterday':

$$p(x_i \,|\, x_0, \ldots, x_{i-1}) = p(x_i \,|\, x_{i-1}) \quad \text{for all } i. \tag{A}$$

They were introduced about a century ago, in the 1906 article Распространение закона больших чисел на величины, зависящие друг от друга ['Extending the law of large numbers for variables that are dependent of each other'] by A.A. Markov, in Известия Физико-математического общества при Казанском университете **15** (2-я серия), 124–156. In most applications the chain is taken *stationary*, so that the transition distribution of $X_i \,|\, x_{i-1}$ is the same, across time points i. Also, in nearly all applications, the sample space for the X_is remains the same throughout. In situations where this sample space is finite (say $\{1, \ldots, k\}$) or countably infinite (say $\{0, 1, 2, \ldots\}$), the process is called a *Markov chain*.

(a) Show that the Markov assumption (A) implies two other useful consequences: first, that 'past and future, given today, are independent',

$$p(x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n \,|\, x_i) = p(x_0, \ldots, x_{i-1} \,|\, x_i)p(x_{i+1}, \ldots, x_n \,|\, x_i) \text{ for all } i; \tag{B}$$

and, secondly, that 'today given everything is else is the same as today given only yesterday and tomorrow',

$$p(x_i \,|\, \text{rest}) = p(x_i \,|\, x_{i-1}, x_{i+1}) \quad \text{for all } i. \tag{C}$$

Assume for your proofs that the sample space is finite, i.e. working with Markov chains rather than with more general Markov processes.

(b) Attempt also to show that (B) and (C) are fully equivalent re-characterisations of the Markov condition (A).

(c) Consider a Markov chain on the state space $\{1, 2, 3\}$, with transition probability matrix

$$\mathbf{P} = \begin{pmatrix} 0.7 & 0.2 & 0.1 \\ 0.1 & 0.6 & 0.3 \\ 0.1 & 0.1 & 0.8 \end{pmatrix},$$

where the rows are made up of

$$P_{i,j} = \Pr\{X_{n+1} = j \,|\, X_n = i\} \quad \text{for } i, j = 1, 2, 3.$$

Simulate one long chain from this distribution. Count the number of times the chain visits the sites 1, 2, 3, and compare with the exact equilibrium distribution of the chain. [It may be useful to use `sample` in **R**:

$$\texttt{sample}(\texttt{c}(1,2,3), \texttt{k}, \texttt{replace} = \texttt{T}, \texttt{prob} = \texttt{c}(.1,.8,.1))$$

generates k samples from $\{1, 2, 3\}$, with replacement, with the indicated probabilities $(0.1, 0.8, 0.1)$.]

(d) Such random sequences may also have memory length greater than one step. If

$$p(x_i \,|\, x_0, \ldots, x_{i-1}) = p(x_i \,|\, x_{i-2}, x_{i-1}) \quad \text{for all } i \geq 2,$$

then the sequence is said to be a Markov chain of order two, etc. Suppose such a chain, with states $\{1, 2, 3\}$ as above, has two-step transition probabilities as given in this table:

from/to	11	12	13	21	22	23	31	32	33
11	0.7	0.2	0.1						
12				0.5	0.3	0.2			
13							0.1	0.3	0.6
21	0.5	0.3	0.2						
22				0.8	0.1	0.1			
23							0.2	0.4	0.4
31	0.6	0.2	0.2						
32				0.3	0.4	0.3			
33							0.2	0.3	0.5

Simulate a long chain from this second-order Markov chain distribution. Compute the relative frequencies. Could these numbers have been anticipated (and calculated) in advance?

19. Markov chains, local characteristics, and the Gibbs Sampler

When attempting to generalise the Markovian dependence concept from dimension one to dimension two (and higher), characterisation (C) above is more fruitful than (A) and (B). The view is then that $X_0, X_1, \ldots$ forms a Markov chain of order one provided (C) holds; that it forms a Markov chain of order two provided

$$p(x_i \,|\, \text{rest}) = p(x_i \,|\, x_{i-2}, x_{i-1}, x_{i+1}, x_{i+2}) \quad \text{for all } i,$$

etc. More generally, the operating model assumption is that

$$p(x_i \,|\, \text{rest}) = p(x_i \,|\, x_{\partial i}) \quad \text{for all } i, \qquad (*)$$

where ∂i is the collection of neighbours of site i, and $x_{\partial i} = \{x_j : j \sim i\}$. Here '$j \sim i$' denotes '$j$ is a neighbour of i', and we demand merely that this is a reflexive relation, with $j \sim i$ if and only if $i \sim j$. The definition $(*)$ then makes sense both in dimension one and in higher dimensions (as long as there is a pre-defined notion of what 'j is a neighbour of i' is). The conditional probabilities $(*)$ are called *the local characteristics* of the process.

(a) For the Markov chain on $\{1, 2, 3\}$ given in Exercise 18(c), compute the list of all local characteristics.

(b) Use these to simulate a long Markov chain in the *Gibbs Sampler* way, as follows. Start out with any given chain, say

$$x^1 = (1, \ldots, 1).$$

Then scan through all sites (in the order $1, \ldots, n$), leading to the 2nd chain

$$x^2 = (x_0^2, \ldots, x_n^2),$$

where x_i^2 at site i is drawn randomly from the distribution $p(x_i \,|\, x_{i-1}^2, x_{i+1}^1)$. Then iterate further, producing in effect a *Markov chain of Markov chains*, with

$$x^t = (x_0^t, \ldots, x_n^t) \quad \text{for iterations } t = 3, 4, 5, \ldots,$$

where

$$x_i^t \sim p(x_i \,|\, x_{i-1}^t, x_{i+1}^{t-1}).$$

Results from MCMC theory (Markov Chain Monte Carlo) imply that indeed x^t has a well-defined equilibrium distribution, as t increases, equal to the distribution of the original Markov chain $(X_0, \ldots, X_n)$. Execute this scheme, and compare with what you found for Exercise 18(c).

(c) The above point describes the 'systematic scan version' of the Gibbs Sampler. Another version is the 'random site version', which works as follows. From iteration $t-1$, consisting in a full chain $(x_0^{t-1}, \ldots, x_n^{t-1})$, choose only one site randomly, say i, and update only this one, from the local characteristics $p(x_i \,|\, x_{i-1}^{t-1}, x_{i+1}^{t-1})$. This defines the tth generation x^t. Execute this regime too, and compare with what you found above.

(d) For one-dimensional Markov chains, the direct forward method of simulations is typically an easier task than via the Gibbs Sampler and local characteristics. The point is that the Gibbs Sampler also works in much more complicated situations, and, specifically, for two- and higher-dimensional Markov random fields. As another illustration, still in dimension one, generate a long chain, perhaps of length 500, from a 2nd order Markov chain distribution with local characteristics of the form

$$p(x_i \,|\, \text{rest}) = \frac{\exp\{\beta H(x_i, x_{\partial i})\}}{\sum_{a=1}^3 \exp\{\beta H(a, x_{\partial i})\}},$$

where $x_{\partial i}$ comprises neighbours of order two (i.e. $x_{i-2}, x_{i-1}, x_{i+1}, x_{i+2}$), and

$$H(a, x_{\partial i}) = \sum_{j \sim i} I\{x_j = a\}$$

counts the number of neighbours equal to the guy in the middle. Play a bit with different values of β.

20. Markov random fields and the Gibbs Sampler

Consider a two-dimensional Markov random field, defined on the $m \times m$ grid

$$\mathcal{G}_m = \{1, \ldots, m\} \times \{1, \ldots, m\},$$

with possible values $x_i \in \{0, 1\}$ only. The neighbourhood system defined via the nearest neighbours only; i.e. a position inside the grid has four neighbours, a position along one of the four borders has three, and the four extreme corners have two. We assume that the local characteristics have the form

$$p(x_i \,|\, \text{rest}) = \frac{\exp\{\beta H_i(x_i, x_{\partial i})\}}{\exp\{\beta H_i(0, x_{\partial i})\} + \exp\{\beta H_i(1, x_{\partial i})\}},$$

where $H_i(a, x_{\partial i})$ is the number of neighbours j (of the site i) with $x_j = a$, for $a = 0, 1$. A bigger value of β encourages outcomes with greater spatial continuity than for smaller β values (and $\beta = 0$ corresponds to full independence across all sites).

(a) Use the Gibbs Sampler to simulate a Markov chain of random fields on $\mathcal{G}_m$. Use systematic scans over the image, in the order of say $i = 1, \ldots, m$ and $j = 1, \ldots, m$, where the new label put in position (i, j) in generation t is drawn via

$$x^t_{i,j} \sim p(x_{i,j} \,|\, x^t_{i-1,j}, x^t_{i,j-1}, x^{t-1}_{i+1,j}, x^{t-1}_{i,j+1})$$

(i.e. using two 'very recent' and two 'old' neighbours). When iterated in this way, the process has the Markov random field as its equilibrium distribution. Again, play a bit with different values of β, to monitor the effect on the spatial continuity of the random outcomes.

When programming this, leading to say

$$\texttt{gibbs} = \texttt{array}(\texttt{data} = \texttt{NA}, \texttt{dim} = \texttt{c}(\texttt{sim}, \texttt{mm}, \texttt{mm})),$$

the idea being that `gibbs[ss, , ]` is to be the `ss`th image over the $m \times m$ grid, include something like

$$\texttt{contour}(1 : \texttt{mm}, 1 : \texttt{mm}, \texttt{gibbs}[\texttt{ss}, \,, \,], \texttt{levels} = \texttt{c}(0, 1))$$

at the end of each loop, so that you can see the effect of each full Gibbs scan of your image. [In my own and current implementation, I am being lazy, and condition on whatever might be on the four border sides of the grid, i.e. do my Gibbsian sampling only inside the $\{2,\ldots,m-1\}^2$ grid. The point is that this is easier and cleaner, since here all sites have four neighbours; a bit of extra programming is required to deal specially with sites along the borders.]

(b) Generalise the model, and your programming efforts, to the case where each site has eight (and not only four) neighbours, and where the local characteristics are

$$p(x_i \,|\, \text{rest}) = \frac{\exp\{\beta H_i(x_i, x_{\text{near}}) + \gamma J_i(x_i, x_{\text{diag}})\}}{\exp\{\beta H_i(0, x_{\text{near}}) + \gamma J_i(0, x_{\text{diag}})\} + \exp\{\beta H_i(1, x_{\text{near}}) + \gamma J_i(1, x_{\text{diag}})\}},$$

where H_i as above counts the number of the four immediate neighbours equal to the guy in the middle, while J_i counts the number of the four diagonal neighbours that are equal to the guy in the middle. Try a positive β and a negative γ, to encourage 'crosses', and report about what happens.

(c) Take any image, say of size 100×100, and perhaps generated by yourself according to the model above, and attempt to estimate the parameters of the model – i.e. β alone, in the first model, and (β, γ) in the second model.

21. Markov chains via the Gibbs Sampler

Markov chains are typically thought of and worked with in terms of processes 'moving forward in time', but other interpretations and characterisations are more fruitful for two- and higher-dimensional generalisations. The point of the present exercise is to learn how to (i) simulate Markov chains and (ii) estimate parameters in such chains, not from the usual transition probabilities

$$p(x_{i+1} \,|\, x_i) = \Pr\{X_{i+1} = x_{i+1} \,|\, X_i = x_i\}$$

but rather from the local characteristics

$$p(x_i \,|\, x_{i-1}, x_{i+1}) = \Pr\{X_i = x_i \,|\, X_{i-1} = x_{i-1}, X_{i+1} = x_{i+1}\}.$$

This is useful when it comes to two- and higher-dimensional situations; cf. comments already made in Exercise 19.

(a) For a Markov chain with transition probabilities $p_{i,j}$, show that

$$p(b \,|\, a, c) = \Pr\{X_i = b \,|\, X_{i-1} = a, X_{i+1} = c\} = \frac{p_{a,b}p_{b,c}}{\sum_d p_{a,d}p_{d,c}} = \frac{p_{a,b}p_{b,c}}{\sum_d p^{(2)}_{a,c}}.$$

For illustration, consider a Markov chain on states $\{1,2,3\}$ with transition probability matrix

$$\mathbf{P} = \begin{pmatrix} 1-2\theta & \theta & \theta \\ \theta & 1-2\theta & \theta \\ \theta & \theta & 1-2\theta \end{pmatrix},$$

where θ is in $(0, \frac{1}{2})$. Find all the local characteristics, i.e. the $3 \times 3 = 9$ probability distributions $p(\cdot \,|\, a, b)$.

(b) Your task is now to simulate a long chain $X_0, X_1, \ldots, X_n$ from the Markov chain mechanism used in (a), perhaps with $n = 1000$. This is most simply done in a 'direct, forward' fashion, simulating X_{k+1} given its predecessor X_k. Do this, and compute the empirical transition counts

$$N = \begin{pmatrix} N_{1,1} & N_{1,2} & N_{1,3} \\ N_{2,1} & N_{2,2} & N_{2,3} \\ N_{3,1} & N_{3,2} & N_{3,3} \end{pmatrix},$$

where

$$N_{a,b} = \sum_{i=1}^{n-1} I\{X_i = a, X_{i+1} = b\} \quad \text{for } a, b = 1, 2, 3$$

counts the number of (a, b) transitions. Check that

$$\widehat{p}_{a,b} = \frac{N_{a,b}}{N_{a,\cdot}} = \frac{N_{a,b}}{\sum_c N_{a,c}} \quad \text{for } a, b = 1, 2, 3$$

are appropriately close to the real $p_{a,b}$ (if the chain is long enough). Use e.g. $\theta = 0.05$ for the spatial context parameter.

(c) Then do the job once more, but this time using the Gibbs Sampler, as explained in Exercise 19. Simulate `sim` chain iterations, e.g. `sim` $= 1000$, and treat the final chain (say `x[sim, ]`) as a genuine realisation. Compute again the N transition count matrix, and check that the consequent $\widehat{p}_{a,b}$ estimates match the real $p_{a,b}$ – this is of course a check on your methods, implementation, and on whether your chain of chains 'has converged' at its appropriate equilibrium distribution. In your simulation programme, inside the necessary `for` loop, include a line of the type

```
matplot(1:nn, xx[ss, ], type="l", ylim=c(1,3))
```

that makes it possible to view the result of each iteration in your chain of chains.

(d) Let us turn to the estimation task: take the chain $x_0, \ldots, x_n$ you constructed (via the Gibbs Sampler) in (c), and consider the *pseudo-likelihood function*

$$\mathrm{pl}(\theta) = \prod_{i=1}^{n-1} p_\theta(x_i \,|\, x_{i-1}, x_{i+1}).$$

Compute and display the log-pl function

$$\log \mathrm{pl}(\theta) = \sum_{i=1}^{n-1} \log \mathrm{pl}_\theta(x_i \,|\, x_{i-1}, x_{i+1}),$$

and find, in particular, the pseudo-likelihood estimate $\widehat{\theta}$ that maximises this function.

– In general, this estimation method is somewhat less efficient than full maximum likelihood, for Markov chain models (see N.L. Hjort and C. Varin, 'ML, PL, QL for

Markov chain models', *Scandinavian Journal of Statistics*, 2008), but the present point is to gain experience and understanding for the one-dimensional cases. The two- and higher-dimensional worlds are rather more complicated to live in ('it's a two-dimensional jungle out there'), and there pseudo-likelihood methods are much more prevalent and almost necessary. It is also important to note that methods of Gibbs Sampler and pseudo-likelihood give us the opportunity to *model and analyse contextual dependence* in terms of local characteristics, instead of via transition probabilities. We may e.g. use constructions of the type

$$p(x_i \,|\, x_{i-3}, x_{i-2}, x_{i-1}, x_{i+1}, x_{i+2}, x_{i+3}) \propto \exp\bigl\{\beta_1 H_1(x_i, x_{i\pm1}) + \beta_2 H_2(x_i, x_{i\pm2}) + \beta_3 H_3(x_i, x_{i\pm3})\bigr\},$$

for suitable 'context functions' H_1, H_2, H_3.

22. Chain restoration via Markov chains

This exercise is concerned with *image reconstruction* techniques, which are easier to write out and work with in dimension one, so we start there. A typical task takes the following form: suppose there is a true image $x = (x_1, \ldots, x_n)$ that is observed with noise. How can we 'reconstruct' x, i.e. arrive at a good statistical estimate $\widehat{x}$ of the original image?

(a) Download the file `markov-with-noise` from the course website, which actually contains both the true image $x_1, \ldots, x_n$ (of length $n = 200$) and the data $y_1, \ldots, y_n$; our aim is to 'reconstruct' the true $x_1, \ldots, x_n$. To visualise the task, do

```
matplot(1:nn, cbind(xtrue, yy), type="l",
        xlab="time", ylab="truth and data")
```

(b) Suppose $x_1, \ldots, x_n$ have arisen following a Markov chain over states $1, \ldots, k$, with transition probabilities $p_{a,b} = p(b \,|\, a)$, and assume that we only can observe $y = x + \varepsilon$, or

$$y_i = x_i + \varepsilon_i \quad \text{for } i = 1, \ldots, n,$$

where the ε_i are independent and $\mathrm{N}(0, \sigma^2)$. Thus

$$y_i \,|\, x_i \sim f(y_i \,|\, x_i) = \mathrm{N}(x_i, \sigma^2)(y_i)$$

(and generalisations to other error distributions will not be difficult). Show that x given y also follows a Markov chain. Use in essence that

$$p(x \,|\, y) = \frac{p(x)p(y \,|\, x)}{p(y)} \propto p(x) \prod_{i=1}^{n} f(y_i \,|\, x_i).$$

(c) Show also that the local characteristics, of the unknown truth given what we have observed, take the form

$$p(x_i \,|\, x_{\mathrm{rest}}, y) = \frac{p_{x_{i-1},x_i} p_{x_i,x_{i+1}} f(y_i \,|\, x_i)}{\sum_a p_{x_{i-1},a} p_{a,x_{i+1}} f(y_i \,|\, a)}. \tag{1}$$

Show that this may also be expressed as

$$p(x_i \,|\, x_{\text{rest}}, y) = \frac{\widetilde{p}_i(x_i \,|\, x_{i-1})\widetilde{p}_{i+1}(x_{i+1} \,|\, x_i)}{\sum_a \widetilde{p}_i(a \,|\, x_{i-1})\widetilde{p}_{i+1}(x_{i+1} \,|\, a)},$$

in terms of the 'updated' or 'revised' transition probabilities

$$\widetilde{p}_i(a \,|\, x_{i-1}) = c_1 p(a \,|\, x_i) f(y_i \,|\, a)^{1/2} = \frac{p(a \,|\, x_i) f(y_i \,|\, a)^{1/2}}{\sum_b p(b \,|\, x_i) f(y_i \,|\, b)^{1/2}},$$
$$\widetilde{p}_{i+1}(x_{i+1} \,|\, a) = c_2 p(x_{i+1} \,|\, a) f(y_i \,|\, a)^{1/2} = \frac{p(x_{i+1} \,|\, a) f(y_i \,|\, a)^{1/2}}{\sum_b p(x_{i+1} \,|\, b) f(y_i \,|\, b)^{1/2}}.$$

Note that stationarity is lost: $x \,|\, y$ is still following a Markov chain, but its transition probabilities are influenced differently at each time point i, depending upon the particular y_i observed there.

– In the case symmetric Markov chains, where $p_{a,b} = p_{b,a}$ for all pairs a, b, there are simplifications of the updated probabilities above, also pertaining to interpretation. The main point, regarding both structure, Gibbs Sampling, and estimation, is that $x \,|\, y$ is a (non-stationary) Markov chain with well-defined local characteristics given by (1).

(d) Attempting to reconstruct the original image, let us first employ a simple and direct classifier: we allocate y_i to the integer $1, 2, 3$ that is closest to y_i:

$$\widehat{x}_i = \begin{cases} 1 & \text{if } y_i \text{ is closest to } 1, \\ 2 & \text{if } y_i \text{ is closest to } 2, \\ 3 & \text{if } y_i \text{ is closest to } 3. \end{cases}$$

Compute $\widehat{x}_i$, and show that it makes *71 mistakes out of 200*, thus bothered by an error rate of $71/200 = 35.5\%$. Note that this method is 'non-contextual', and makes no use of any underlying spatial continuity of the x chain – neither does it use any particular properties of the $f(y_i \,|\, x_i)$ distributions, apart from sensibly allocating y_i to the closest of its three class centres $1, 2, 3$.

(e) To perform a contextual classification (more laborious, but also more interesting, promising and fruitful), let us go back to the three-state Markov chain of Exercise 21, with transition probability matrix

$$\mathbf{P} = \begin{pmatrix} 1-2\theta & \theta & \theta \\ \theta & 1-2\theta & \theta \\ \theta & \theta & 1-2\theta \end{pmatrix}.$$

Compute the necessary characteristics of the the chain $x \,|\, y$, using the `markov-with-noise` data set. Simulate say `sim` $= 500$ chains from $p(x \,|\, y)$ using the Gibbs sampler.

Use $\sigma = 0.75$ for the noise level, $\theta = 0.05$ for the spatial context. Then allocate y_i to the state (among $1, 2, 3$) with highest probability. In other words,

$$\text{estimate } p_i(k \,|\, \text{data}) = \Pr\{x_i = k \,|\, \text{data}\} \quad \text{with } \widehat{p}_i(k \,|\, \text{data}) = \frac{1}{A} \sum_{t=1}^{A} I\{x_i^t = k\}$$

for $k = 1, 2, 3$, across $A =$ `sim` Gibbs scans, and classify y_i to the state with the highest of these three estimated probabilities. Compute the error rate for this contextual restoration. [I found *8 mistakes out of 200*, i.e. error rate reduced to $8/200 = 4.0\%$.]

(f) Above we used known values of both σ and θ. In the applied image analysis literature it is often claimed that σ is 'not difficult' to estimate separately. Attempt to estimate θ using the fixed value 0.75 for σ.

(g) **Master project:** Attempt to estimate both θ and σ from only the y data. Investigate other Markov chain models. Invent model selection procedures that might distinguish between such models. Explore properties of the resulting classification methods. Speculate also about how these methods and algorithms may be lifted to the two-dimensional case. Find an real-world application. Report to the authorities and cash in a good Master's Degree.