Course 02443
Project in stochastic simulations

Contents

Instructions

- Below is a list of subjects that can be chosen for your project. You can freely modify and extend a subject according to your imagination.
- It is also possible to propose your own subject. This second option is actually warmly recommended! In this case, please write down a small description and submit it to gigu@dtu.dk for approval.
- Please work in group as much as possible (no more than four persons in a group though).
- Your work will be evaluated on the basis of a report and optionnaly (in case of notably weak reports) oral questions.
- Reports must contain:
  - a description of the problem in your own words,
  - a list of questions in terms of the real world application,
  - a list of computational tasks that must be performed to address these real world questions,
  - description of algorithms in pseudo-code
  - the actual code written to implement these algorithms (must be commented and placed in appendix)
  - illustration of how your program(s) work (figures with detailed captions are best)
  - a final discussion with comments on the method and results, ideas for extensions, improvements etc...
- You are free to use whatever interpreter you like.

Please make sure the names of all the persons who took part to the work appear clearly on the report cover. In case of unequal contributions, outline briefly the contribution of each group member on the cover. Lastly, name your pdf file Name1_Name2_..._Namexx.pdf and upload it on Campusnet.

1 Why cars in the next lane seem to go faster?

According to an article published in the scientific journal *Nature*, experiments show that the majority of drivers think that they have been driving in the slower of two lanes even though both lanes have the same average speed.

The goal of this project is to implement a traffic model similar to the one described in the article, simulate data under this model and perform statistical analysis to discuss the paradox.

Suggested work steps:

- Imagine a simple model for one car on one lane accelarating and decelarating randomly. Make some assumptions about how changes in speed occur. At a later stage you may want to study how a change in these assumptions affect the system. It can be helpful to discretize time at a fixed time-step (say 1 sec.) and simulate the various variables involved at each time step. Implement this model and make some graphical representation of the system (accelaration, speed or position of the system)
- Implement a model with two cars on one lane.
- Implement a model with many cars on one lane.
- Implement a model with many cars on two lanes without lane switching
- Analyse the distribution of times "being overtaken" and of time "being overtaking"
- If time permits, try to modify your model so as to make it more realistic

Reference: Redelmeier and Tibshirani, Nature 401 (35), 1999
www2.imm.dtu.dk/courses/02443/projects/Redelmier.pdf

2 Detecting clusters of galaxies from velocity data

Astronomers predict that gravitational pull should lead to some clustering of galaxies and that velocities measured from our galaxy should be multimodal. The goal of this project is to analyse galaxy velocity data published by Roeder et al. in J. Am. Stat. Soc 85(411) 617-624, 1990.

1. A first simple model consists in assuming that if a galaxy belongs to a certain cluster, its velocity is a normal random variable with cluster-specific mean and variance. Denoting the number of clusters by K and assuming that the different clusters are observed with equal probabilities, write the general form of the density of a velocity when the cluster membership is known then when the cluster membership in unknown.

2. Simulate a set of $n = 200$ velocities with $K = 2$ for some arbitrary combination of mean and variances. Plot the histogram of the simulated velocities. Store this set of values for later use.

3. Let us denote by μ_k and σ_k the mean and variance of velocities in cluster k. If the μ_k's and σ_k's are known. Estimate the cluster membership c_i of observation i with velocity x_i using the Bayes theorem by maximising the probability $p(c_i = k|x_i, \mu_k, \sigma_k)$

4. Implement an algorithm to solve the previous question and test it on the dataset simulated above.

5. As a preliminary step to the next questions (and although it is not necessary), it is suggested to write an MCMC algorithm that simulates $(c_1, ..., c_n)$ from the distribution $p(c_1, ..., c_n|x, \mu, \sigma)$ and check that you get the same results as above.

6. If the μ_k's are unknown (σ_k's still known), the vector of unknown quantities consists of the vector of cluster memberships c, and of the vector of cluster means μ. We denote $z = (x, \mu)$. From a Bayesian perspective, z can be estimated by simulating a sample from $p(z|x)$ and taking the mode of simulated values. Implement an algorithm doing the above and check its accuracy on your simulated data.

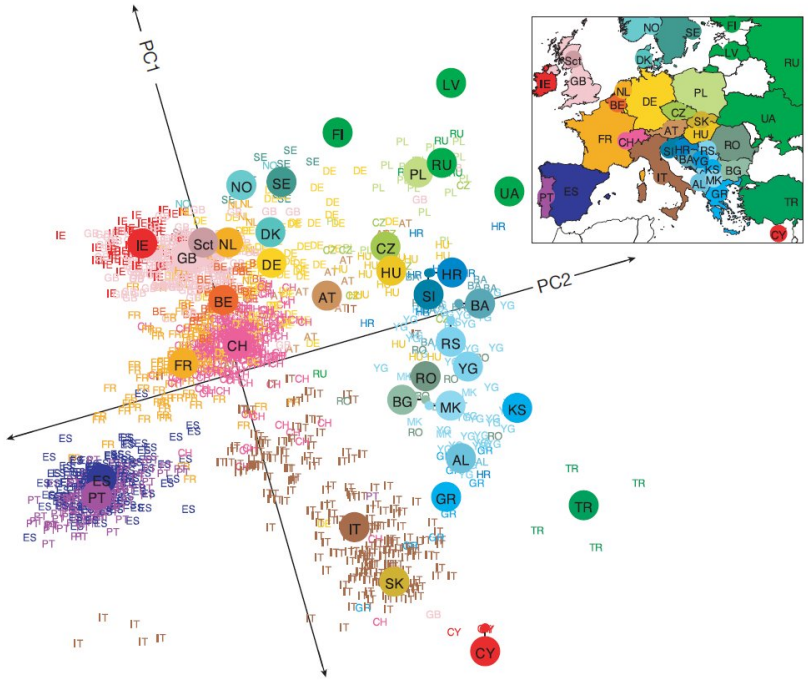
7. Note that the likelihod $p(x|c, \mu)$ is invariant by relabelling the clusters. Can you forsee any annoying consequence of this property on the MCMC simulation? Do you observe any related problem in your MCMC simulation? If so, how can you handle it?

8. Same as above for the case where both μ_k's and σ_k's are unknown.

Paper: `http://www2.imm.dtu.dk/courses/02443/projects/Roeder_JASA_1990.pdf`

Data: `http://www2.imm.dtu.dk/courses/02443/projects/data/galaxy.txt`

3 Detecting genetically homogeneous groups



If two populations evolve independently (i.e. without exchange of migrants), it is expected that the frequency of an allele in each population will differ because of the random nature of the process of allele transmission from generation to generation (a.k.a genetic drift). This observation is the basis of a widely used method to detect isolated populations and hence understand factor impeding migration (e.g. because of habitat fragmentation). It is also widely used in biomedical genetics to ensure that association analyses are not affected by confounding factors. The present subject consists in implementing a model-based clustering algorithm to detect population sub-structure.

Suggested steps:

1. We consider a single chromosome location (locus) at which two alleles only are observed in the population and denote by f_k the frequency in cluster k of an arbitrary allele chosen as reference allele (the other allele is present at a frequency $1 - f_k$). We assume that the various allele frequencies are independent with a flat Beta distribution. We denote by z_i the number of reference alleles carried by individual i. Simulate some values (f, z) for 100 diploid individuals assumed to be sampled among $K = 2$ clusters with equal probability. Store this dataset for use in the next steps.

2. We denote by $c = c_1, .., c_n$ the vector of unknown cluster memberships and we assume that allele frequencies in the various clusters are known. Give the expression of $p(c|z)$ and implement a Metropolis-Hasting within Gibbs algorithm to sample from this distribution.

3. Test the algorithm above on the 'fake' data you simulated previously.

4. Assuming now that cluster memberships are known and allele frequencies are unknwon, write the expression of $(f|z)$ and implement a Gibbs algorithm to sample from this distribution.

5. Assuming that f and c are unknow, implement an algorithm that alternates updates of f and c as above to sample from $p(c, f|z)$

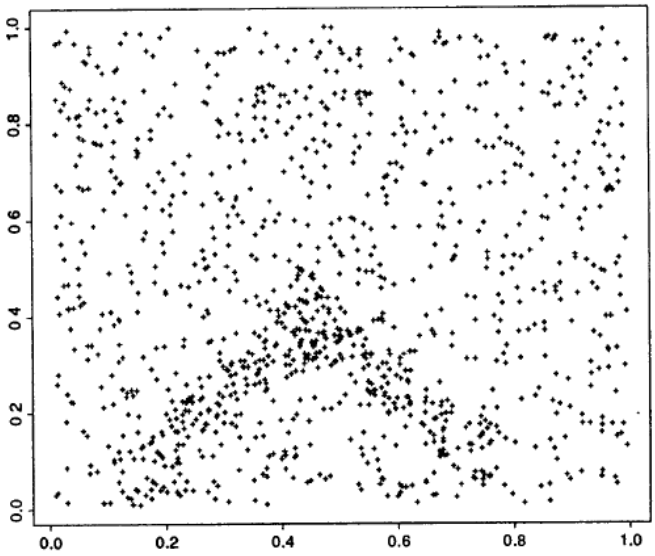
6. Implement the global algorithm above on your 'fake' data.

7. Implement the global algorithm above on the human data from Finland and Sweden `http://www2.imm.dtu.dk/courses/02443/projects/data/Finland_Sweden.csv`

Reference: the algorithm above is inspired by Pritchard et al, Genetics, 2000 p.947. Article available from the Genetics web site: `http://www.genetics.org/content/155/2/945.full`

4 Detecting a mine field from aerial photos

The picture below (simulated) gives an idea of how an area encompassing a mine field looks like after processing an aerial photo: a set of points in the plane consisting of actual mines and other objects that can be stones, small holes in the ground, splinters from previously exploded mines etc... The area with a higher density of points is presumably the mine field. The goal of this project is to implement a method to delineate the area of a mine field in an automated and objective way.



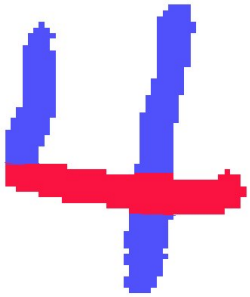
Let $X = (x_1, .., x_n)$ be a set of points in the plane. The Voronoi cell C_i induced by $x_i \in X$ is defined as the set of points in the plane closer to x_i then to any other points in X. More formally $C_i = \{s \in \mathbb{R}^2, d(s, x_i) < d(s, x_j), j \neq i\}$.

For simplicity, it can be assumed that the study area can be divided into an area free of mines A and an area containing mines B that fits with the Voronoi tessellation induced by X. In A and B, the number of points can be assumed to be Poisson processes with rates λ_A and λ_B respectively. Under this assumption, detecting the mine field consist in estimating the *land type* of any Voronoi cell.

1. Have a look at `http://en.wikipedia.org/wiki/Poisson_process/Spatial`
2. Simulate some data according to the model outlined above
3. Save one simulated dataset that will be used in the sequel as "fake" data to check your algorithm.
4. Assume that λ_A and λ_B are known, propose and implement an algorithm to estimate the membership of each Voronoi cell (mine/mine-free).
5. Same as above when λ_A and λ_B are unknown

This subject is inspired from Byers, S.D. and Raftery, A.E. (2002). Bayesian Estimation and Segmentation of Spatial Point Processes using Voronoi Tilings. In Spatial Cluster Modelling (A.G. Lawson and D. G.T. Denison, eds.), London: Chapman and Hall/CRC Press. `http://www2.imm.dtu.dk/courses/02443/projects/Byers-Raftery-Voronoi.pdf`

5 Automatic recognition of hand-written characters



A hand-written character can be viewed as a deformation of a standard template by some additive noise. The goal of this project is to develop an algorithm to classify some characters (whose shape is observed as coordinates of small number of landmarks) into a small alphabet.

To simplify the problem and focus on the statistical aspects, we will consider that the letters belong to a small simplified "alphabet" consisting of a circle, a square and an equilateral triangle. We consider that a character is observed at a set of landmarks and that a hand-written character is obtained by deforming continuously one of the characters of the alphabet.

1. Make some assumptions about the distribution of the deformation
2. Simulate some data according to the model outlined above and represent them graphically
3. What are the unknown quantities in this problem?
4. Assume that the parameters of the distribution of the noise are known. Propose and implement an algorithm to classify a hand-written character.
5. Same as above when the parameters of the noise are unknown.

6 Evaluating the probability of extreme rainfall

Floods occur when strong rainfall events affect simultaneously many areas in a hydrological catchment. To evaluate the probability of flood, it is not not enough to be able to evaluate the probability of strong rainfall at a single location, the probability of simultaneous strong rainfall at several location is required. This is not tractable analytically in most models. The goal of this project is to implement a method to evaluate the probability of extreme rainfall on the basis of a spatial statistical model.

A simple model could be to assume that a vector of observed rainfall at n sites $Y_1, ..., Y_n$ is of the form $\phi(X_1), ..., \phi(X_n)$ where ϕ is a function known up to a few parameters and $X_1, ..., X_n$ is a centred random vector whose covariance matrix depends on the relative locations of the n sites. The function ϕ must be chosen so as to produce Y values that have the same distribution as commonly observed rainfall (an aspect that depends on location and seasons). A decent choice about ϕ consists in assuming that the Y values are exponentially distributed.

A common assumption about this covariance matrix is that

$$Cov(X_i, X_j) = \exp(|s_i - s_j|/ - \alpha) \tag{1}$$

where s_i is the geographical location at which X_i is observed.

The various steps of the project could be as follows:

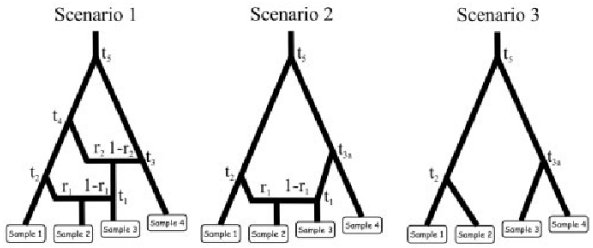
1. Implement a program to simulate some data Y_is from the above model.
   Hint:
   - Generate some points $(s_i)_{i=1,...,n}$ (say $n = 100$) in the plane either regularly (nodes of a grid) or irregularly, that will be used as geographical sites of observation of rainfall.
   - Assume that $X = (X_1, ..., X_n)$ is a normal random vector $N(0, \Sigma)$ where Σ is a spatial covariance matrix as in equation (1) above. Build the covariance matrix Σ corresponding to the n sites generated above.
   - Simulate $X = (X_1, ..., X_n)$ using the Choleski decomposition of Σ.
   - Represent graphically the value simulated in space (hint in R: use function `as.image` of R package `fields`)
   - Simulate $Y = (Y_1, ..., Y_n)$
   - Try to explain intuitively and graphically by repeating the simulation process how the spatial pattern is affected by α
2. We consider now the situation where we have a dataset consisting of observed rainfall at say $n = 100$ geographical locations, and we want to estimate a single unknown parameter α we assume that the exponential distribution as a known parameter λ. We take a Bayesian perspective and want to estimate α as the average of the posterior distribution $p(\alpha|y_1, ..., y_n)$. Implement an MH algorithm to simulate from this distribution and to estimate the posterior mode of α. Test the accuracy of this algorithm on various simulated

y values. You can use e.g. the (fake) rainfall data available from `http://www2.imm.dtu.dk/courses/02443/projects/data/rain.txt`

3. Assuming now that α is known, propose a strategy to based on simulations to evaluate the probability that the mean rainfal over an area is over a certain threshold.

4. Implement your strategy from the previous question and compare the result to that obtain in a model assuming no spatial auto-correlation.

7 Inference on mixture models with approximate Bayesian Computations (ABC)



The simplest algorithm to simulate approximately from the posterior distribution $\pi(\theta|x_0) \propto \pi(\theta)\pi(x_0|\theta)$ is as follows:

1. `simulate` θ_0 `from` $\pi(\theta)$
2. `simulate` x `from` $\pi(x|\theta_0)$
   (a) `if` $d(x, x_0) < \varepsilon$ `deliver` θ_0
   (b) `else goto 1`

Although crude, this algorithm is very useful in areas where a model is available to simulate from the prior/likelihood but the prior and/or the likelihood is not available in closed form. This situation is common in ecology and molecular biology. The goal of this project is to see how this algorithm can be applied and modified for some common models such as mixture of distributions.

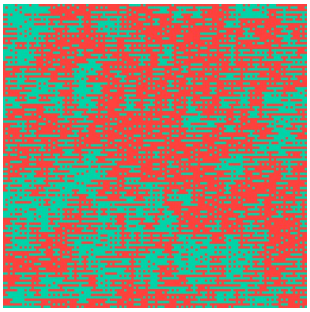
Reference: Marjoram, P., Molitor, J., Plagnol, V. and Tavaré, S. (2003). Markov chain Monte Carlo without likelihoods. P. Natl. Acad. Sci. USA 100 (26):15324-15328.

1. A toy model: a person tosses a coins (p unknown) ten times and report the fraction f of heads among those ten outcomes. It is natural to assume that p is a random variable and consider that prior to the first 10 tossings, p has a uniform distribution on $[0, 1]$. What is the density of probability of p given f? The person tosses again the coin once. What is the probability to get a head given f? Implement an ABC solution to this problem.
2. We consider now a mixture of two univariate normal distributions. Implement an ABC solution for the case where the mixing weights and the variances are known.
3. Same as above with only known weights
4. Same as above with means, variances and weights unknown.
5. Try to implement the same algorithm to cluster observations.

The alogrithm developped could be implemented on the acidity[1] or the enzyme[2] data. Some background about the data can be found in the JRSS B article of Richardson and Green [3].

[1] `http://www2.imm.dtu.dk/courses/02443/projects/data/acidity.txt`
[2] `http://www2.imm.dtu.dk/courses/02443/projects/data/enzyme.txt`
[3] `http://www2.imm.dtu.dk/courses/02443/projects/RichardsonGreen_JRSSB_1997.pdf`

8 Image analysis with the Potts model



The Potts model is a model for random vectors representing discrete values at the pixels of a grid. This model enjoys one key feature: values at pairs of neighbouring pixels tend to be more dependent than values at pairs of remote pixels. The strength of the dependence is controlled by the so-called interaction parameter. This model is used for example for image denoising. The goal of the project is to implement an algorithm allowing to simulate from the prior but also to denoise an image blurred by a noise.

The Potts model is defined by the following equation:

$$p(x_1, ..., x_n) \propto \exp\left[\psi \sum_{i \sim j} I_{x_i = x_j}\right]$$

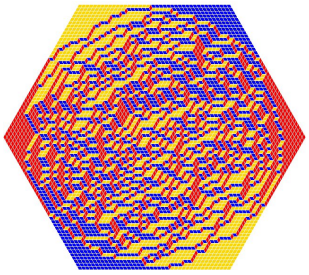
In the above equation, $i \sim j$ reads i in relation with j and means that the sum is computed over pairs of pixels being neighbors (with a four or eight nearest neighbor scheme).

1. Simulate the above model with a Metroplois-Hastings sampler for $K = 2$ (Ising model)
2. The model used for image denoising is defined as follows: $y_i|x_i \sim N(x_i, \sigma^2)$. Use this model to sample from $p(x|y)$
3. Implement your algorithm on the (fake) data available from `http://www2.imm.dtu.dk/courses/02443/projects/data/potts.txt`.

   To be read with `as.matrix(read.table())` and visalized e.g. with

```
nx <- 500
ny <- 500
xmax <- 1
ymax <- 1
n <- nx*ny
image(seq(0,xmax,l=nx),seq(0,ymax,l=ny),x,
      main="Variable x",
      xlab="x",ylab="y",
      col=terrain.colors(8))
```

9 Perfect simulation with Markov chains: coupling from the past



The MCMC methods described during the lecture suffer all from the same flaw: theory predicts what the asymptotic distribution of the chain is, but it does not say (except in very simple cases) how close to this limit distribution the chain is after a finite number of iterations. Propp and Wilson proposed a method known as *coupling from the past* to bypass this issue. The goal of this project is to implement this algorithm first on a toy model for a Markov chain with values in $\{0, 1\}$ or $\{1, 2, 3\}$, then in the Ising model on a $2^n \times 2^n$ grid.. This model originates from statistical physics and is now used in areas as diverse as image analysis, material science, geology, genetics.

References:

- An introduction to some key ideas in the the book "Simulation", S. Ross, Academic Press. Copies of relevant pages to be found
`www2.imm.dtu.dk/courses/02443/projects/Ross289.JPG`
`www2.imm.dtu.dk/courses/02443/projects/Ross290.JPG`
`www2.imm.dtu.dk/courses/02443/projects/Ross291.JPG`

- A review by Xeni Dimakos, Int. Stat. Review 2001
`www2.imm.dtu.dk/courses/02443/projects/Dimakos_IntStatRev_2001.pdf`

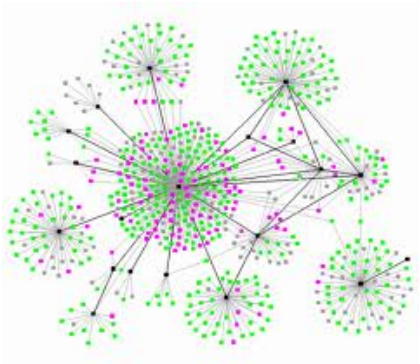
- The seminal article by James G. Propp and David B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. Random Structures and Algorithms, 9:223–252, 1996.

- A reference for the dominating argument is to be found in the article by George Casella, Michael Lavine, and Christian P. Robert: Explaining the perfect sampler. The American Statistician 55(4):299-305, 2001. available here `www2.imm.dtu.dk/courses/02443/projects/Casella_Lavine_Robert_AmStat_2001.pdf`

- A recent synthesis can be found in the book by Robert and Casella, Monte Carlo Statistical methods, 2nd edition Springer 2004, chapter 13.

- An interesting presentation with connections in genetics, unfortunately only available in French, mentionned here mostly for the record `www2.imm.dtu.dk/courses/02443/projects/AgregMath_CFTP_Mutation.pdf`.

10 Simulation and analysis of random networks



For a finite set of n vertices, a network can be defined as an $n \times n$ matrix X with entries in $\{0, 1\}$. In words: this matrix says whether two vertices are connected by an edge or not.

In the sequel, we consider that no vertex is connected to itself $X_{ii} = 0$ and the graph is not directed $X_{ij} = X_{ji}$. A model of random network sometimes referred to as as the Erdös-Rényi model is defined by the following property: all edges are equally likely with probability p. This model is simple and therefore not appealing for many applications.

We consider here the Erdös-Rényi mixture defined as follows. Vertices are spread into Q classes with prior probabilities $\alpha_1, ..., \alpha_Q$.

In the following, we use the indicator variables Z_{iq} (with $\sum_q Ziq = 1$). $\alpha_q = Pr(Z_{iq} = 1) = Pr(i \in q)$, with $\sum_q \alpha_q = 1$.

Then we denote π_q the probability for a vertex from class q to be connected with a vertex from class l. Because the graph is undirected, these probabilities must be symmetric such that $\pi_{ql} = \pi_{lq}$ We finally suppose that edges are conditionally independent given the classes and that $X_{ij}|(i \in q, j \in l) \sim b(\pi_{ql})$ for all i, j.

The idea of this project is to implement an algorithm to estimate parameters.

Suggested steps:

1. Simulate random networks from the model above with various sets of parameters and represent them graphically

2. If network data are known to arise from the model above, how can we estimate its parameters in a Bayesian setting? One may consider e.g. MCMC inference or Approximate Bayesian Computations (ABC, cf. related project above).