Computer exercise 4

Gaussian Markov random fields

Spatial statistics and image analysis, TMS016

1 Introduction

The purpose of this computer exercise is to give an introduction to simulation and kriging using Gaussian Markov random field (GMRF) models for spatial data. When in doubt about how to use a specific function in Matlab, use `help` and `doc` to get more information.

2 Simulation of GMRFs

As discussed in the lectures, we typically specify the precision matrix (inverse covariance matrix) of GMRFs using stencils. The function `stencil2prec` can be used to compute the precision matrix for a given stencil. For example

```
>> kappa = 1;
>> q = kappa^2*[0 0 0;0 1 0;0 0 0] + [0 -1 0; -1 4 -1;0 -1 0];
>> Q = stencil2prec([100,100],q);
```

computes the precision matrix for GMRF on a 100×100 lattice, with stencil

$$\kappa^2 \begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} + \begin{pmatrix} \cdot & -1 & \cdot \\ -1 & 4 & -1 \\ \cdot & -1 & \cdot \end{pmatrix} = \begin{pmatrix} \cdot & -1 & \cdot \\ -1 & 4+\kappa^2 & -1 \\ \cdot & -1 & \cdot \end{pmatrix}$$

In the following tasks, try some different stencils, such as the one above,

$$\begin{pmatrix} -10 & -0.1 & \cdot \\ -0.1 & 20.4+\kappa^2 & -0.1 \\ \cdot & -0.1 & -10 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} \cdot & -1 & -10 \\ -1 & 24+\kappa^2 & -1 \\ -10 & -1 & \cdot \end{pmatrix}.$$

- To get a feeling for the covariance structure that the stencils imply, compute and plot the covariance between the middle pixel of the image, and all other pixels. Writing

  ```
  >> v = zeros(m^2,1);
  >> v(ind) = 1;
  >> c = Q\v;
  ```

  gives a column-stacked image (of size $m \times m$) of the covariance between the pixel with index `ind` and all other pixels in the image. Try this for some different stencils and different values of $\kappa > 0$.

- Simulate mean-zero Gaussian fields with the precision matrices above, and see how the different stencils affect the realizations. To simulate the field, use sparse Cholesky factorization:

  ```
  >> R = chol(Q);
  >> x = R\randn(m^2,1);
  ```

- Finally, test how the covariances and simulations are affected by using stencils with more neighbors. You can either construct a larger stencil manually, or use that $\mathbf{Q}_2 = \mathbf{Q}\mathbf{Q}$ gives a GMRF with a higher order neighborhood structure. If you want to view the sparsity structure of $\mathbf{Q}$, you can use the `spy` command in matlab.

3 Image reconstruction using GMRFs

Choose you favourite GMRF model from above and simulate and image $\mathbf{x}$. We will now do kriging for missing pixels in an image: Randomly extract N pixels which are observed

```
>> ind = randperm(m^2);
>> ind_obs = ind(1:N);
>> ind_mis = ind(N+1:end);
>> x_obs = x(ind_obs);
>> x_mis = x(ind_mis);
```

- Extract the precision matrices for the observed and missing pixels, $\mathbf{Q}_{op}, \mathbf{Q}_o, \mathbf{Q}_p$. For example, `Qop = Q(ind_obs,ind_mis)`.
- Compute the posterior mean $\mathsf{E}(\mathbf{x}_{mis}|\mathbf{x}_{obs})$ and reconstruct the complete image. Compare with the true image for different values of N. How high can the percentage of missing pixels be so that you still get a good reconstruction?

4 Noise reduction using GMRFs

Choose you favourite GMRF model from above and simulate and image $\mathbf{x}$. We will now do kriging reconstruction of this image when it is corrupted by noise: Simulate an observed noisy image

```
>> y = x + sigma_e*randn(m^2,1)}
```

where `sigma_e` is the standard deviation of the noise.

- Compute the posterior precision matrix $\mathbf{Q}_{x|y}$ for the noisy data. Based on this, compute the posterior mean $\mathsf{E}(\mathbf{x}|\mathbf{y})$ and compare with the true image $\mathbf{x}$. Test how the size of σ_e affect the reconstruction.