1 Latent Gaussian Random Fields with ADMB

The paper [Skaug and Fournier, 2006] includes a simple example of a latent Poisson regression with spatially correlated counts. The computational time of this model grows fast when the number of random effects gets large. In this article we illustrate how to formulate the same model as a latent Gaussian Markov Random Field (GMRF) and thereby exploiting the internal sparse matrix computations of ADMB. Furthermore we generalize the model to a 2D lattice to illustrate how to deal with latent spatial processes of a very high dimension in ADMB. The model can easily account for missing data and as a by-product provide predictions of a latent spatial surface in grid points where no observations are available.

Model As starting point consider the latent Poisson regression model given by:

$$x_i|\lambda_i \sim \text{Poisson}(\lambda_i)$$

$$\boldsymbol{\lambda} \sim \text{MVLN}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}})$$

I.e. the observations are assumed independent Poisson conditional on an unobserved multivariate log-normal distribution with covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ containing unknown parameters of interest. For simplicity assume that the mean of the multivariate normal distribution is zero.

The full negative log-likelihood of this model is given by

$$l(\boldsymbol{\theta}, \boldsymbol{\eta}) = \sum_{i=1}^{n} \log \text{Pois}(x_i, e^{\eta_i}) + \frac{1}{2} \log |\boldsymbol{Q}_{\boldsymbol{\theta}}| + \frac{1}{2} \boldsymbol{\eta}' \boldsymbol{Q}_{\boldsymbol{\theta}} \boldsymbol{\eta}$$

If some of the Poisson counts are missing we simply need to remove the corresponding Poisson terms from the log-likelihood.

AR(1) sparseness The simple example of [Skaug and Fournier, 2006] considers the exponential correlation function

$$\rho(\Delta x) = e^{-\alpha \Delta x}$$

It is well known that this is the correlation function of a stationary 1st order Gaussian auto regressive process [Rue and Held, 2005]. Therefore the inverse covariance matrix (precision matrix) is a band-matrix with band width 1:

$$\boldsymbol{Q} = \begin{pmatrix} 1 & -\phi & & & & & \\ -\phi & 1+\phi^2 & -\phi & & & & \\ & & & & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & & & \\ & & & & -\phi & 1+\phi^2 & -\phi \\ & & & & & -\phi & 1 \end{pmatrix}$$

where $\alpha = -\log(\phi)$. The determinant is given by $|Q| = 1 - \phi^2$. When coding the model in ADMB-RE these analytical results can be exploited using the "SEPARABLE_FUNCTION" statement.

Lattice generalization One simple way to extend a 1D correlation to 2D is by assuming separability

$$\rho(\Delta x, \Delta y) = \rho_x(\Delta x)\rho_y(\Delta y)$$

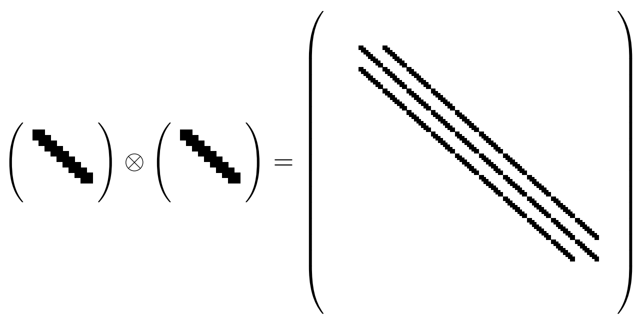
where the correlations ρ_x and ρ_y are 1D exponential correlations. The full covariance matrix takes the form of a Kronecker product

$$\boldsymbol{\Sigma} = \boldsymbol{\Sigma_x} \otimes \boldsymbol{\Sigma_y}$$

The inverse of a Kronecker product is found by inverting each of the factors

$$\boldsymbol{Q} = \boldsymbol{Q_x} \otimes \boldsymbol{Q_y}$$

so the resulting precision matrix $\boldsymbol{Q}$ will inherit the sparseness of $\boldsymbol{Q_x}$ and $\boldsymbol{Q_y}$:



Closed form expression of the determinant is available through $|\boldsymbol{Q}| = |\boldsymbol{Q_x}|^{n_y}|\boldsymbol{Q_y}|^{n_x}$

Results Here we present the results from running the model on simulated data. A square grid of size 50×50 is considered with "true" parameters $\phi_1 = 0.82$ and $\phi_2 = 0.90$. This parameter configuration implies a correlation range of the spatial field of $\approx 10\%$ of the grid size in the x-dimension and $\approx 20\%$ of the grid size in the y-dimension. A simulation from the GMRF is obtained along with Poisson observations attached randomly to 10% of the 2500 grid points (Fig. 1 top-left).
Confidence intervals of the estimated correlation functions are constructed on log-scale and then transformed to natural scale. The true correlation functions lies within the 95%-confidence bands (Fig. 1 top-right and bottom-left).
The re-constructed spatial surface given in terms of the estimated random effects (Fig. 1 bottom-right) agrees nicely with the true spatial surface.
The source code is given in the appendix and the ADMB-RE executable is run by

```
./gmrf -shess -ilmn 5 -ind gmrf50missing.dat -ndi 50000
```

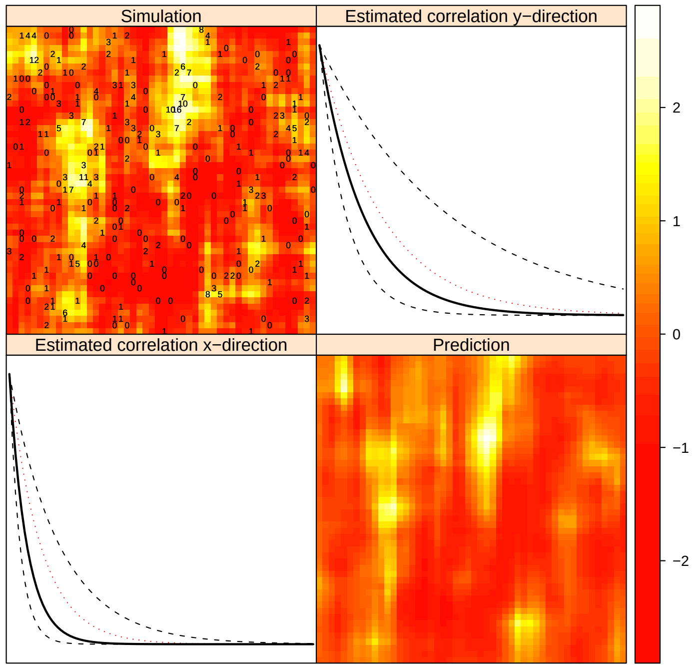


Figure 1: Results from running the model on 50×50 grid with simulated data. Image of simulated GMRF η in each lattice point with observed Poisson counts corresponding to $\approx$ 90% missing data (top-left). Estimated exponential correlation functions with 95%-confidence intervals (black lines) and true correlation functions (red dotted line) (top-right and bottom-left). Prediction of un-observed GMRF (bottom-right).

We used the command line flag "-shess" to activate the sparse matrix computations in the inner optimization problem.
The model can be run on grid sizes of 100×100 on a machine with 8Gb of memory.

A Source code

```
// ------------------------------------------------------------------
// Latent 2D Gaussian Markov Random Field with Poisson observations.
// Assume known mean=0 and variance=1.

// Q1: Precision matrix of stationary AR(1) process with parameter phi1.
// Q2: Precision matrix of stationary AR(1) process with parameter phi2.
```

```
// Q:  The kronecker product Q=Q1xQ2 (Full precision matrix of 2D field)
// n:  The dimension of Q1 and Q2
// N:  The observation vector of n*n counts

// Storage of Q1 and Q2 (triplet storage):
// Nonzero entries of e.g. Q1 are (Qi,Qj).
// Matrix only takes three different values: (-phi1, 1, 1+phi1^2)
// Which of the three values to insert in (Qi,Qj) is given by Qx
// ---------------------------------------------------------------


DATA_SECTION
  init_int n                        // Dimension of Q1 and Q2
  init_int nnz                      // Number of non-zeros of Q1 and Q2
  init_ivector Qi(0,nnz-1)          // Pattern of Q1 and Q2 in triplet storage
  init_ivector Qj(0,nnz-1)
  init_ivector Qx(0,nnz-1)
  init_ivector N(0,n*n-1)

PARAMETER_SECTION
  init_bounded_number phi1(0.001,0.999)
  init_bounded_number phi2(0.001,0.999)
  random_effects_vector eta(0,n*n-1)
  objective_function_value val
  sdreport_vector logcor1(0,100);
  sdreport_vector logcor2(0,100);

PROCEDURE_SECTION
  #define MYPRINT(x)
  //#define MYPRINT(x)std::cout << #x << "=" << x << "\n";

  val=0;
  for(int i=0;i<nnz;i++){    // Loop through non-zeros of Q1
    for(int j=0;j<nnz;j++){ // Loop through non-zeros of Q2
      quad_form(i,j,phi1,phi2,eta[Qrow(i,j)],eta[Qcol(i,j)]);
    }
  }

  // log-determinant of Kronecker product:
  // logdet(Q1xQ2)=n*logdet(Q1)+n*logdet(Q2)
  logdetQ_contrib(phi1);
  logdetQ_contrib(phi2);

  for(int i=0;i<n*n;i++){    // Loop through data-entries
    pois_loglik(i,eta[i]);
  }
```

```
  // sdreport
  if(sd_phase()){
    logcor1=phi2logcor(phi1);
    logcor2=phi2logcor(phi2);
  }



// Three different nonzero values in stationary AR(1) precision
// FUNCTION void setparms(dvar_vector& Qx, const dvariable &phi)
//    dvariable kappa=1/(1-phi*phi);
//    Qx[1]=-phi; Qx[2]=1; Qx[3]=1+phi*phi;
//    Qx=kappa*Qx; // Now the inverse of Q1 and Q2 is a correlation matrix

// Given i'th nonzero of Q1 and j'th nonzero of Q2
// Return the row and column index of Q and the corresponding value of Q.
FUNCTION int Qrow(int i, int j)
  return Qi[i]*n+Qi[j];
FUNCTION int Qcol(int i, int j)
  return Qj[i]*n+Qj[j];

// Add contribution of quadratic form corresponding to i'th nonzero
// of Q1 and j'th nonzero of Q2
SEPARABLE_FUNCTION void quad_form(int i,int j,const dvariable& phi1, const dvari
  dvar_vector Q1x(1,3);
  dvar_vector Q2x(1,3);

  //setparms(Q1x,phi1);  <--- Not work within SEPARABLE_FUNCTION
  dvariable kappa1=1/(1-phi1*phi1);
  Q1x[1]=-phi1; Q1x[2]=1; Q1x[3]=1+phi1*phi1;
  Q1x=kappa1*Q1x; // Now the inverse of Q1 and Q2 is a correlation matrix

  //setparms(Q2x,phi2);  <--- Not work within SEPARABLE_FUNCTION
  dvariable kappa2=1/(1-phi2*phi2);
  Q2x[1]=-phi2; Q2x[2]=1; Q2x[3]=1+phi2*phi2;
  Q2x=kappa2*Q2x; // Now the inverse of Q1 and Q2 is a correlation matrix

  dvariable Qij=Q1x[Qx[i]]*Q2x[Qx[j]];
  val += .5*etai*Qij*etaj;

// Add contribution from logdet(Q1) and logdet(Q2)
SEPARABLE_FUNCTION void logdetQ_contrib(const dvariable& phi)
  dvariable kappa=1/(1-phi*phi);
  dvariable logdet=log(1-phi*phi);
  logdet+=n*log(kappa);
```

```
  val -= .5*n*logdet;

// Add Poisson contribution
// Negative count is interpreted as missing value.
SEPARABLE_FUNCTION void pois_loglik(int i, const dvariable &etai)
  if(N(i)>=0) val += exp(etai)-N(i)*etai;

// Report estimated correlation function
FUNCTION dvar_vector phi2logcor(dvariable phi)
  dvar_vector x(0,100);
  for(int i=0;i<=100;i++)x(i)=(i*0.01)*n;
  x(0)=1e-12;
  dvariable alpha=-log(phi);
  return -alpha*x;


//REPORT_SECTION
//   logcor1=phi2logcor(phi1);
//   logcor2=phi2logcor(phi2);
//   report << "cor1" << endl;
//   report << cor1 << endl;
//   report << "cor2" << endl;
//   report << cor2 << endl;

TOP_OF_MAIN_SECTION
  arrmblsize=10000000;
  gradient_structure::set_GRADSTACK_BUFFER_SIZE(2000000);
  gradient_structure::set_CMPDIF_BUFFER_SIZE(100000000);
  gradient_structure::set_MAX_NVAR_OFFSET(100000);
  gradient_structure::set_NUM_DEPENDENT_VARIABLES(204);
```

References

[Rue and Held, 2005] Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields: Theory and Applications*, volume 104 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London.

[Skaug and Fournier, 2006] Skaug, H. and Fournier, D. (2006). Automatic approximation of the marginal likelihood in non-Gaussian hierarchical models. *Computational Statistics and Data Analysis*, 51(2):699–709.