Spatial modelling in ADMB - A review

Honolulu, March, 2012

Hans J. Skaug
University of Bergen

Type of models

- GLM type of models (latent Gaussian random field)
  - Random effects functionality of ADMB
- Possible to do classical geostistics with the non-RE stuff

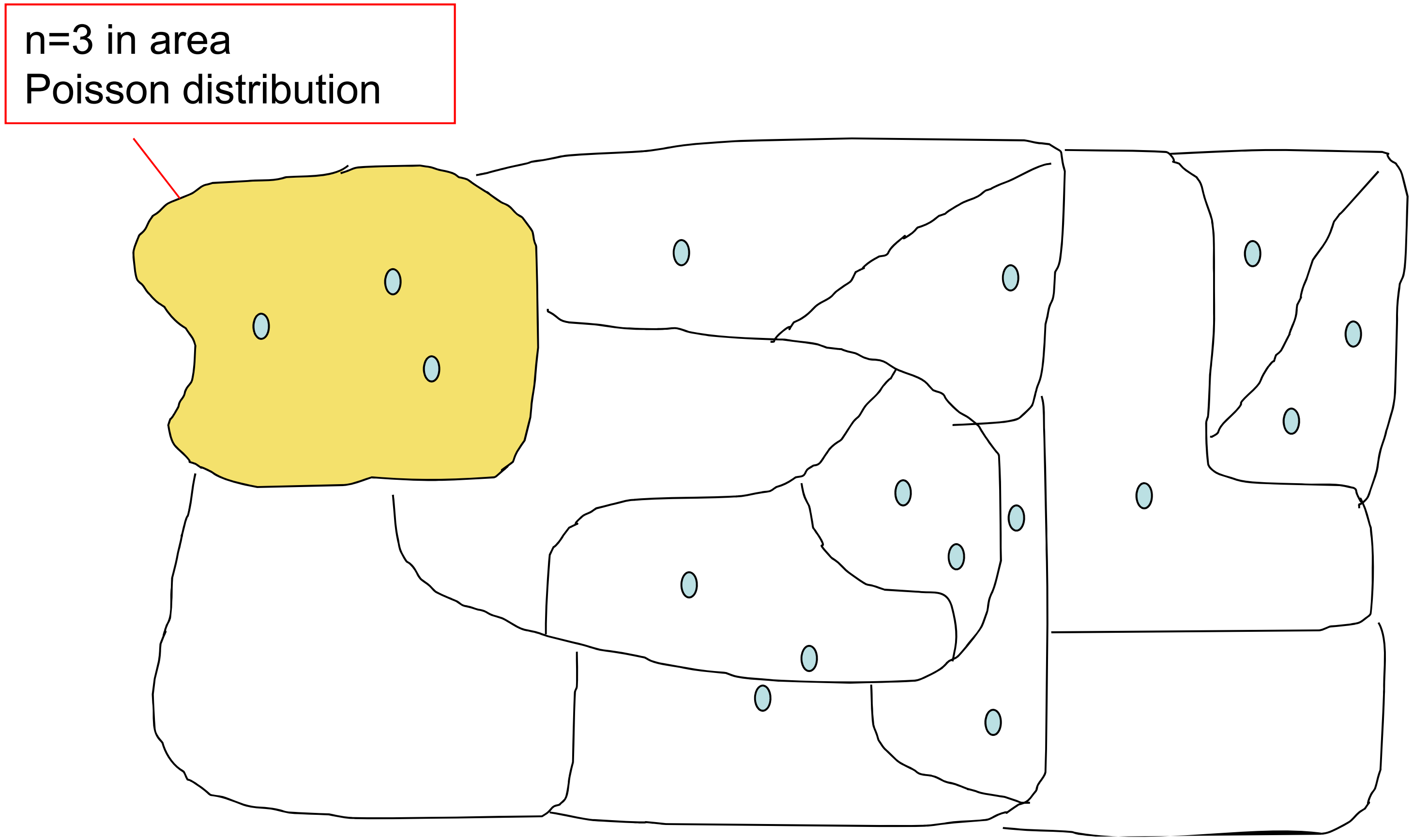
- Example collection:

http://admb-project.org/examples/spatial-models

Approaches in ADMB-RE

1. The geostatistical approach
   – Specify the spatial covariance matrix
2. GMRF (Gaussian Markov random field)
   – Where "neighbors" can be defined
3. Separable covariance function
   – Hybrid between 1) and 2)
4. Thin plate splines (2-dim splines)
   – Unexplored

Example: Counting animals by area



Model

- Number of animals in each area Poisson distributed
- Intensity of animals varies spatially

$$\log[\text{intensity}(x,y)] = u(x,y) \sim \text{Gaussian}$$

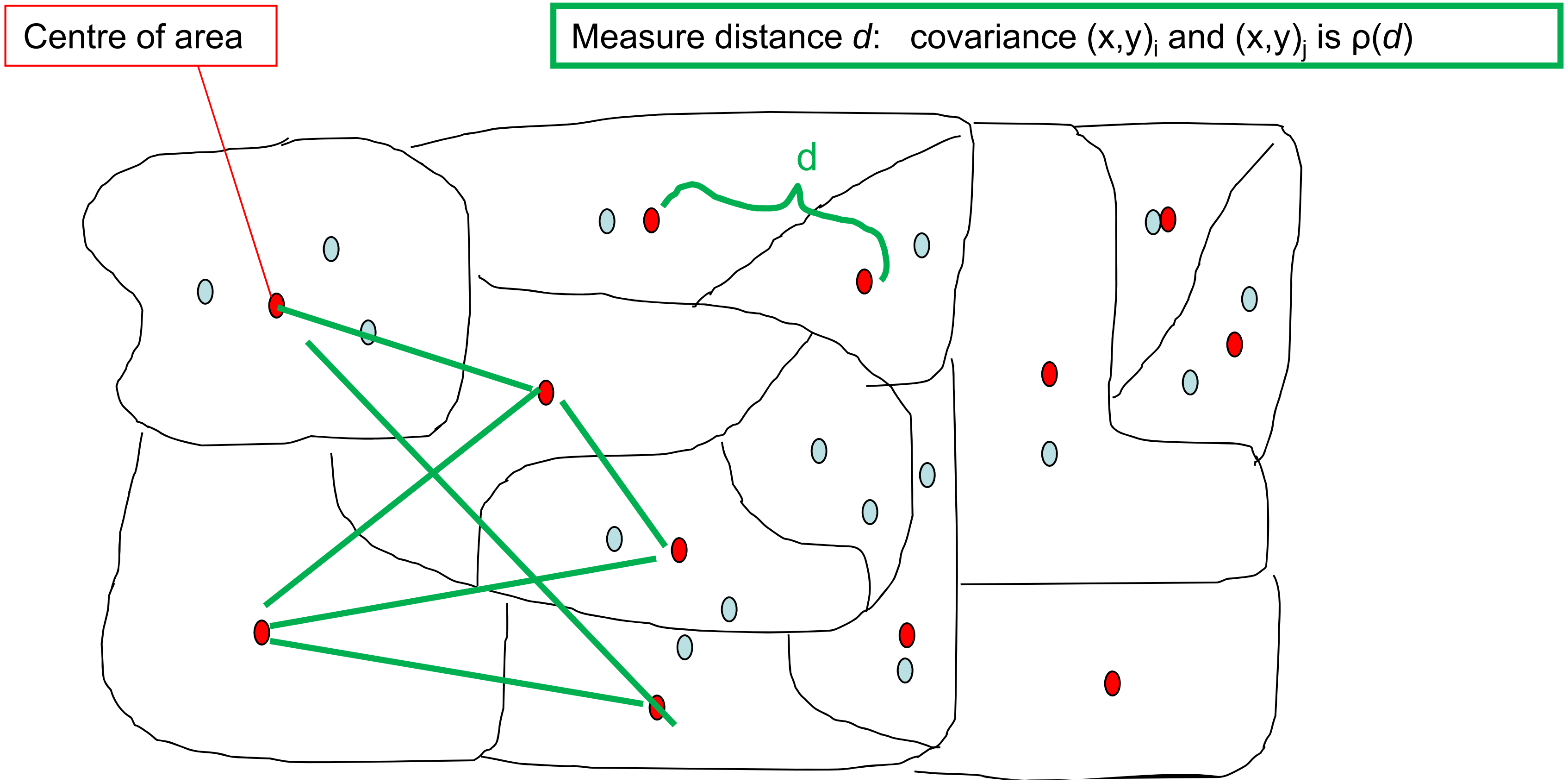
- Approximation

Centre of area

$$\log[\text{expected \# in area } i] = A_i * u(x_i, y_i)$$

Size of area

Geostatistical approach



Calculate pairwise distances

- Construct covariance matrix: M

```
for (i=1;i<=n;i++)
  pois_loglik(i,u(i),b,log_sigma);                    // Likelihood contribution from i'th

SEPARABLE_FUNCTION void pois_loglik(int i,const dvariable& ui,const dvar_vector& _b,const dva
    dvariable eta = X(i)*_b + exp(_log_sigma)*ui;     // Linear predictor
    dvariable lambda = mfexp(eta);                    // Mean in Poisson distribution
    l += lambda-y(i)*eta;

NORMAL_PRIOR_FUNCTION void get_M(const dvariable& _a)
  int i,j;
  dvar_matrix tmpM(1,n,1,n);

  for (i=1;i<=n;i++)
  {
    tmpM(i,i)=1.0;
    for ( j=1;j<i;j++)
    {
      tmpM(i,j)=exp(-_a*dd(i,j));                    // Exponentially decaying correlation
      tmpM(j,i)=tmpM(i,j);
    }
  }
  M=tmpM;
```

In this case points are on a 2-dim grid

Parameter vector

Should not apply df1b2 approach to prior

Gaussian prior

$$g(u,\theta) = l_{Pois}(u,\theta) - \boxed{\overbrace{\tfrac{1}{2}u^T M u}^{\text{Gaussian prior}}}$$

Spatial random field

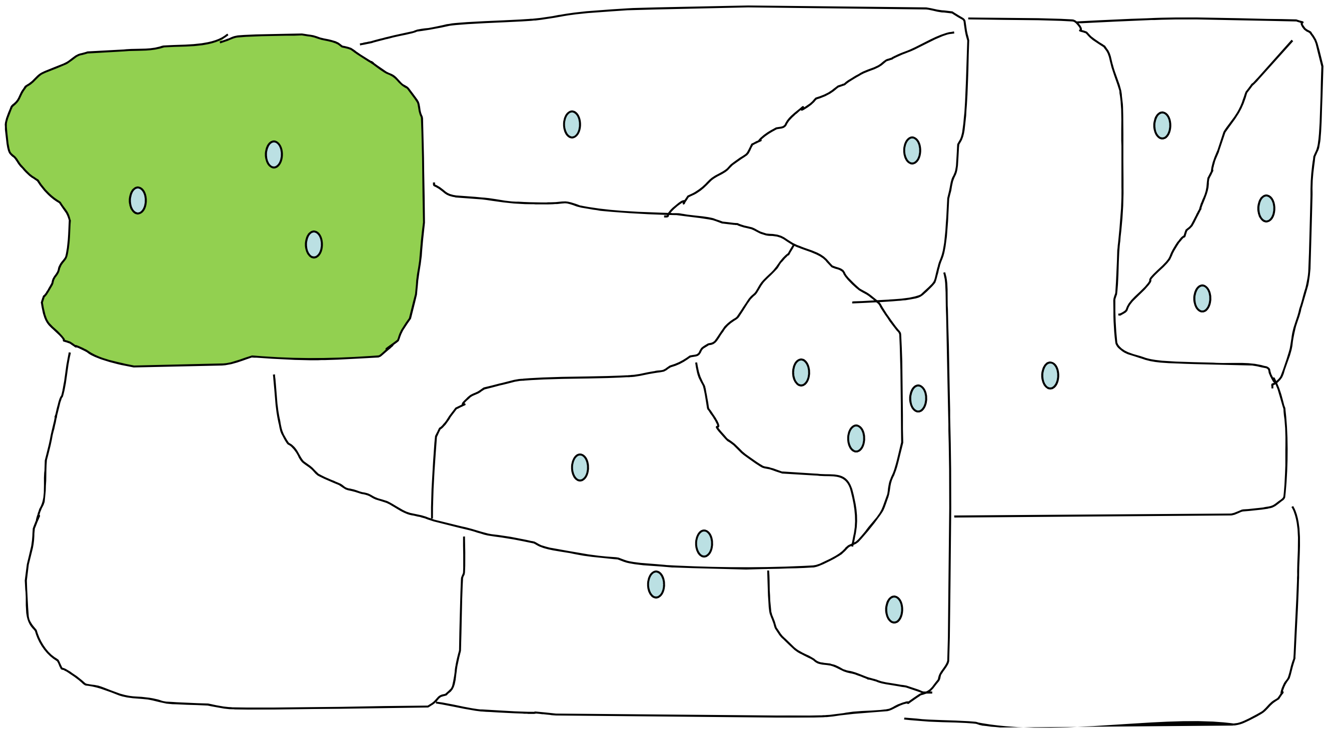
On list-to-do

- The covariance matrix M should only be evaluated once during inner optimization
  - 10 times speed up
  - Dave has provided «hack»
    - Best solution
  - Dave’s hack needs implementation in flex
    - Would be nice to do during this workshop

Neighbour approach: GMRF (CAR model)

Model the distribution of
Green area conditionally on neighbours

- Do this for all areas
- Ideal for Gibbs sampling
- Gives sparse Hessian



```
for (i=1;i<=n;i++)
{
  int is_island = (m(i) == 0);
```

Has no neighboors

Number of neighbors

```
  ll_poisson(phi(i),theta(i),b,sigma,tau,i,is_island);

  n01_prior(theta(i));          // Independent effects

  if(is_island)
    n01_prior(phi(i));          // In this case phi(i) should not be used,
                                // according to the winbugs example. In ADMB
                                // we nevertheless must assign a prior to it.
                                // The alternative would be to omit it from
                                // the phi-vector, but that is notationally clumsy.
```

One prior per area

```
  else
    car_prior(phi(W(i)),i);     // Here: phi(W(i) = area-i and its neighboors.
}
```

index of neighbors of area "i"

spatial field

```
SEPARABLE_FUNCTION void n01_prior(const dvariable& phi)
  g -= -0.5*square(phi);

SEPARABLE_FUNCTION void car_prior(const dvar_vector& phi,const int i)

  dvariable mean = sum(phi(1,m(i)))/m(i);
  g -= -0.5*square(phi(0)-mean)*m(i);
```

$\phi_i \sim N\left(mean, \frac{1}{m_i}\right)$

Hybrid approach: separable covariance function Data on grid

$$\rho(\Delta x, \Delta y) = \rho_1(\Delta x) \cdot \rho_2(\Delta x)$$

$$\rho(z) = \exp\left(-\alpha \left|\Delta z\right|\right)$$

$$\rho(\Delta x, \Delta y, \Delta t) = \rho_1(\Delta x) \cdot \rho_2(\Delta x) \cdot \rho_3(\Delta t)$$

Yields sparse Hessian

Not ideal for spatial correlation

- Non-isometry unless ρ Gaussian

```
  for (i=1;i<=n;i++)
    pois_loglik(i,u(i),b,log_sigma);                    // Likelihood contribution from i'th

SEPARABLE_FUNCTION void pois_loglik(int i,const dvariable& ui,const dvar_vector& _b,const dva
    dvariable eta = X(i)*_b + exp(_log_sigma)*ui;        // Linear predictor
    dvariable lambda = mfexp(eta);                       // Mean in Poisson distribution
    l += lambda-y(i)*eta;

NORMAL_PRIOR_FUNCTION void get_M(const dvariable& _a)
  int i,j;
  dvar_matrix tmpM(1,n,1,n);

  for (i=1;i<=n;i++)
  {
    tmpM(i,i)=1.0;
    for ( j=1;j<i;j++)
    {
      tmpM(i,j)=exp(-_a*dd(i,j));                        // Exponentially decaying correlation
      tmpM(j,i)=tmpM(i,j);
    }
  }
  M=tmpM;
```

In this case points are on a 2-dim grid

Parameter vector

Should not apply

df1b2 approach to prior

$$g(u,\theta) = l_{Pois}(u,\theta) - \boxed{\overbrace{\frac{1}{2} u^T M u}^{\text{Gaussian prior}}}$$

Spatial random field

Need Gaussian prior with sparse M