A model for spatial data

A Gaussian model for spatial data is:

\[ Y \sim N(X\beta, \Sigma(\theta, \sigma^2_e)) \quad \Sigma(\theta, \sigma^2_e) = \Sigma_m(\theta) + I\sigma^2_e. \]

Collect the covariance parameters in \( \Psi = \{\theta, \sigma^2_e\} \).

The density of the observations is

\[
p(Y) = \frac{1}{(2\pi)^n/2|\Sigma(\Psi)|^{1/2}} \exp \left( -\frac{1}{2}(Y - X\beta)^\top \Sigma^{-1}(\Psi)(Y - X\beta) \right)
\]
The “Big N” problem

The log-likelihood becomes
\[ l(\Psi, \beta | Y) = -\frac{1}{2} \log |\Sigma(\Psi)| - \frac{1}{2} \left( Y - X\beta \right)^\top \Sigma^{-1}(\Psi) \left( Y - X\beta \right). \]

Given (estimated) parameters, predictions at the unobserved locations are given by
\[ E \left( Y_u | Y_k, \hat{\Psi} \right) = \mu_u + \Sigma_{uk} \Sigma_{kk}^{-1} (Y_k - \mu_k). \]

The “Big N” problem

Given \( N \) observations:
- The covariance matrix has \( O(N^2) \) unique elements.
- Computations scale as \( O(N^3) \) (due to \( |\Sigma| \) and \( \Sigma^{-1} \)).

The Markov property

The Markov property for a time discrete stochastic process:
\[ P(x_t | x_{t-1}, \ldots, x_0) = P(x_t | x_{t-1}). \]

Example: Consider an AR(1)-process
\[ x_t = ax_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma^2) \text{ and independent.} \]

The neighbours \( N_t \) of \( x_t \) are points “close” to \( x_t \).

The Markov property – An example

The Markov property does not imply that events far apart are independent. Only that if we know what happens close by we can “ignore” things further away.

Example

We want to predict the temperature in Lund
1. If we know the temperature in Malmö, the temperature in Amsterdam contributes very little information.
2. However, if we don’t know the temperature in Malmö, the temperature in Amsterdam would help.
Gaussian Markov random fields (GMRF)

Let the **neighbours** \( N_i \) to a point \( s_i \) be the points \( \{ s_j, j \in N_i \} \) that are “close” to \( s_i \).

**Gaussian Markov random field (GMRF)**

A Gaussian random field \( x \sim N(\mu, \Sigma) \) that satisfies

\[
p(x_i | \{ x_j : j \neq i \}) = p(x_i | \{ x_j : j \in N_i \})
\]

is a Gaussian Markov random field.

**Good neighbours**

Let me introduce: The **precision matrix** \( Q = \Sigma^{-1} \)

\( Q = \Sigma^{-1} \) is called the **precision matrix**.

Using the **precision matrix** the model becomes

\[
X \sim N(\mu, Q^{-1}) \quad \text{cf.} \quad X \sim N(\mu, \Sigma)
\]

A Gaussian distribution has a **Cannonical representation**

\[
p(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp \left( -\frac{1}{2}(x - \mu)^\top \Sigma^{-1} (x - \mu) \right)
\]

\[
\log p(x) = \text{const.} - \frac{1}{2} x^\top \Sigma^{-1} x + x^\top \Sigma^{-1} \mu
\]

Notation and parameter-links are given by

\[
X \in N(\mu, \Sigma), \quad X \in N_C(b, Q), \quad \begin{cases}
\mu = Q^{-1}b, \\
\Sigma = Q^{-1}
\end{cases}
\]
Let me introduce: The precision matrix $Q = \Sigma^{-1}$

$Q = \Sigma^{-1}$ is called the precision matrix.

Using the precision matrix the model becomes

$$X \sim \mathcal{N}(\mu, Q^{-1}) \quad \text{cf.} \quad X \sim \mathcal{N}(\mu, \Sigma)$$

With conditional expectation

$$E(X_u | X_k) = \mu_u - Q_{uu}^{-1} Q_{uk} (X_k - \mu_k),$$

instead of

$$E(X_u | X_k) = \mu_u + \Sigma_{uk} \Sigma_{kk}^{-1} (X_k - \mu_k).$$

and conditional variance

$$V(X_u | X_k) = Q_{uu}^{-1},$$

instead of

$$V(X_u | X_k) = \Sigma_{uu} - \Sigma_{uk} \Sigma_{kk}^{-1} \Sigma_{ku}.$$ 

The conditional expectation for a single location is

$$E(x_i | x_j, j \neq i) = \mu_i - \frac{\sum_{j \neq i} Q_{ij} (x_j - \mu_j)}{Q_{ii}}$$

$$= \mu_i - \frac{1}{Q_{ii}} \left( \sum_{j \in N_i} Q_{ij} (x_j - \mu_j) + \sum_{j \notin \{N_i, i\}} Q_{ij} (x_j - \mu_j) \right)$$

If $Q_{ij} = 0$ for all $j \notin N_i$ then

$$E(x_i | x_j, j \neq i) = E(x_i | x_j, j \in N_i).$$

The precision matrix is sparse

Elements in the precision matrix of a Gaussian Markov random field are non-zero only for neighbours and diagonal elements.

$$j \notin \{i, N_i\} \iff Q_{ij} = 0.$$

Computational costs

For an AR(1) like-process the number of neighbours will be $\approx 2d$ for a grid in $\mathbb{Z}^d$.

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Elements</th>
<th>Cholesky</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Sigma$</td>
<td>$\mathcal{O}(N^2)$</td>
<td>$\mathcal{O}(N^3)$</td>
</tr>
<tr>
<td>$Q$</td>
<td>$\mathcal{O}(N)$</td>
<td>$\mathcal{O}(N^{1.5})$</td>
</tr>
</tbody>
</table>

Reordering the precision matrix gives a sparser Cholesky factorisation.
### Computational details

If $Q$ is a sparse matrix then (under mild conditions) the Cholesky factorisation $Q = R^T R$ will also be sparse.

- Simulation of $X \in \mathcal{N}(\mu, Q^{-1})$.
  \[ X = \mu + R^{-1} E, \quad E \in \mathcal{N}(0, I) \]

- Conditional expectations
  \[ E(X_u | X_k) = \mu_u - R_{uu}^{-1} R_{uk} (Q_{uk} (X_k - \mu_k)) \]

- Computing the determinant
  \[ \frac{1}{2} \log |Q| = \frac{1}{2} \log |R^T R| = \log |R| = \sum \log R_{ii} \]

- Never compute $R^{-1}$, use back substitution for triangular systems instead.

---

### Sparse matrices and GMRF

**Q**

- \( nz = 69202 \)

**cho(Q)**

- \( nz = 1000099 \)

**Q reordered**

- \( nz = 298184 \)

**cho(Q) reordered**

- \( nz = 259784 \)

---

### Implementation Details

The conditional expectation

\[ E(X_u | X_k) = \mu_u - R_{uu}^{-1} Q_{uk} (X_k - \mu_k) \]

should be computed without computing $Q^{-1}$.

\[ E = \mu_u - Q_{uu} \backslash (Q_{uk} * (X_k - \mu_k)) ; \]

Given the Cholesky factor

\[ E = \mu_u - R_{uu} \backslash (R_{uu}^T \backslash (Q_{uk} * (X_k - \mu_k)) ) ; \]

It is even more efficient to avoid the matrix transpose

\[ E = \mu_u - R_{uu} \backslash ((Q_{uk} * (X_k - \mu_k))^T / R_{uu} ) ; \]
Implementation Details

Reordering of $Q$ and sparse Cholesky factor

```matlab
p = amd(Q);
Q = Q(p,p);
R = chol(Q);
spy(Q);
spy(R);
```

When reordering the precision matrix it is important to also reorder the observations, mean value, covariates, etc.
To recover the original reordering

```matlab
y_reorder = y(p);
y_org = zeros(size(y));
y_org(p) = y_reorder;
```

Constructing GMRF models

- In order to use GMRFs instead of full covariance models, we need to construct useful, sparse, $Q$-matrices.
- A simple, traditional choice, is the conditional autoregressive, CAR(1) model, with local $q$-pattern

$$
q = \begin{bmatrix}
  0 & -1 & 0 \\
  -1 & 4 + \lambda^2 & -1 \\
  0 & -1 & 0 
\end{bmatrix}
$$

Here the conditional expectations is the weighted sum of the four nearest neighbours.

Interpreting $q$

For stationary fields the local $q$-pattern defines each column in $Q$:

$$
q = \begin{bmatrix}
  0 & -1 & 0 \\
  -1 & 4 + \lambda^2 & -1 \\
  0 & -1 & 0 
\end{bmatrix}, \quad Q_{i,j} = 4 + \lambda^2
$$
Interpreting \( q \)

Given the conditional expectation for a GMRF

\[
E(X_u|X_k) = \mu_u - Q_u^{-1}Q_{uk}(X_k - \mu_k).
\]

\( q \) can also be interpreted as the conditional weighting of nearby pixels

\[
q = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 + K^2 & -1 \\ 0 & -1 & 0 \end{bmatrix}
\]

\[
X_{\text{image}} = \begin{bmatrix} \cdots & x_{i-1,j-1} & x_{i,j-1} & x_{i-1,j+1} & \cdots \\ \cdots & x_{i,j-1} & x_{i,j} & x_{i,j+1} & \cdots \\ \cdots & x_{i+1,j-1} & x_{i+1,j} & x_{i+1,j+1} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}
\]

Assuming \( \mu = 0 \):

\[
E(x_{ij}|\text{all other pixels}) = \frac{x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1}}{4 + K^2}
\]