Fast statistical computing based on sparse symmetric matrices

Thomas Kneib

14.11.2003

1. Structured additive regression

- Consider a nonparametric regression model

$$y = f_1(x_1) + \ldots + f_r(x_r) + \varepsilon$$

with $\varepsilon \sim N(0, \sigma^2)$.

- x_j may be a continuous covariate, a time scale, a location index or a cluster index.
- f_j is a (not necessarily smooth) function of x_j.
- Possible model terms:
  - P-Splines or random walks for nonlinear functions of continuous covariates and time scales.
  - Markov Random Fields or two dimensional P-Splines for spatial effects.

- I.i.d. Gaussian effects for cluster specific heterogeneity.
- ...

- Each vector of function evaluations $f_j = (f_j(x_{1j}), \ldots, f_j(x_{nj}))^T$ may be reexpressed as
$$f_j = X_j \beta_j.$$
X_j is a $n \times p_j$ design matrix, β_j is a $p_j \times 1$ vector of regression coefficients.

- General prior for β_j:
$$p(\beta_j) \propto \exp\left(-\frac{1}{2\tau_j^2}\beta_j^T K_j \beta_j\right).$$
K_j is a $p_j \times p_j$ penalty matrix, τ_j^2 controls the smoothness of f_j.

- In a frequentist approach $p(\beta_j)$ defines a penalty for β_j. The smoothing parameter is given by $\lambda_j = \sigma^2/\tau_j^2$.

- Bayesian analysis based on MCMC simulation techniques: The full conditional for β_j is given by

$$\beta_j|\cdot \sim N\left((X_j^T X_j + \lambda_j K_j)^{-1} X_j^T (y - \eta_{-j}), (X_j^T X_j + \lambda_j K_j)^{-1}\right)$$

  with

$$\eta_{-j} = \eta - f_j.$$

  How can we draw samples from this distribution efficiently?

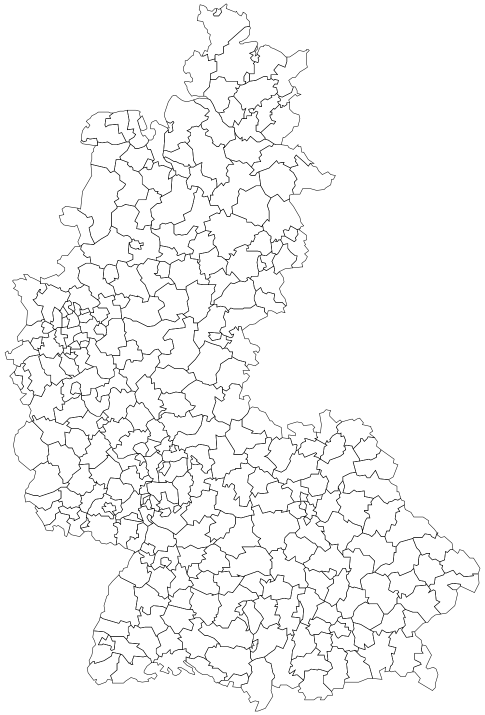
- Model selection based on information criteria or generalized cross validation: How can we compute the degrees of freedom associated with f_j efficiently?

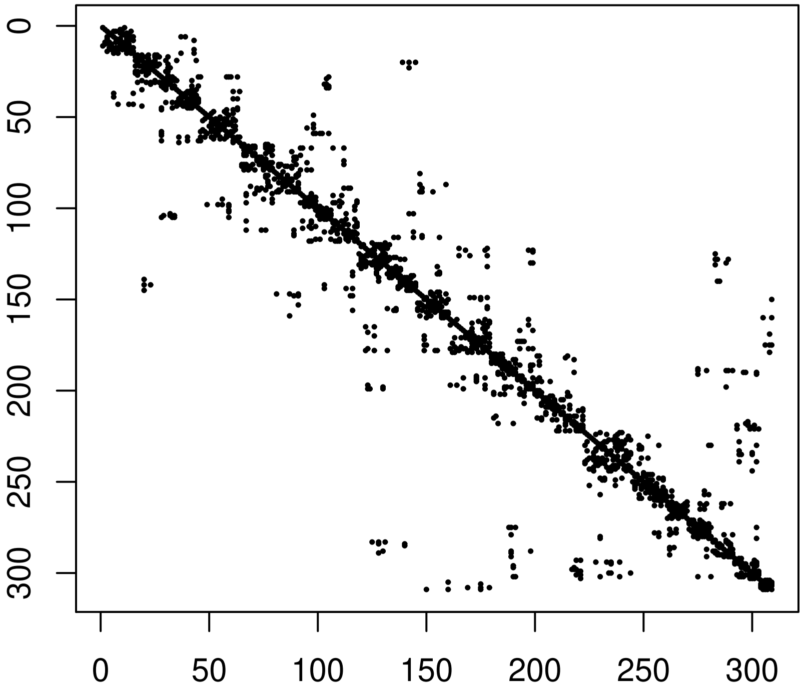
$$df(f_j) = tr\left(X_j (X_j^T X_j + \lambda_j K_j)^{-1} X_j^T\right)$$

- Key: The matrices $X_j^T X_j$, K_j and $X_j^T X_j + \lambda_j K_j$ are sparse matrices. The special structure of these matrices can be exploited in numerical routines.

- Example: Markov random fields
  - K_j has the form of an adjacency matrix.
  - β_j contains the different values of the spatial function.
  - X_j is a 0/1 incidence matrix.

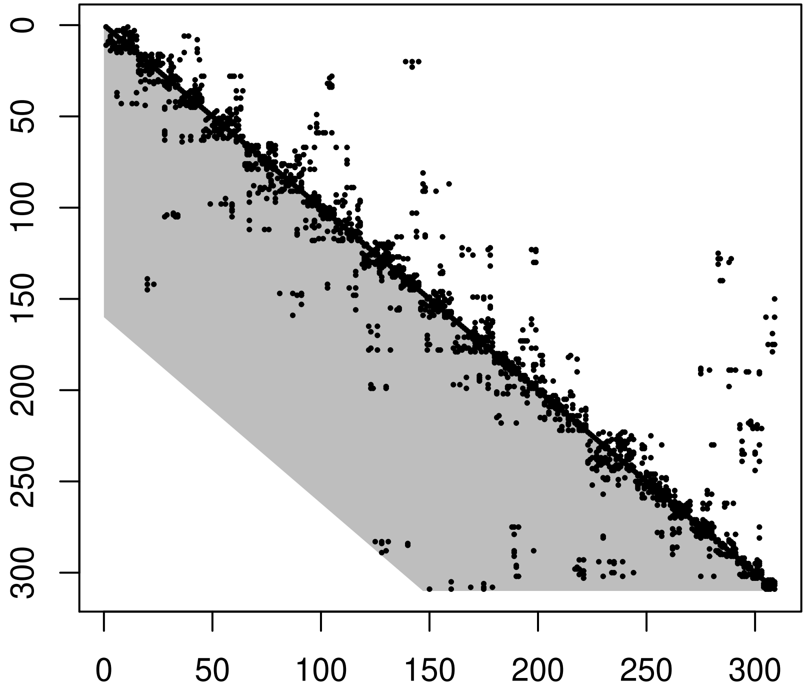
$\Rightarrow$ $X_j^T X_j$ is diagonal and $X_j^T X_j + \lambda_j K_j$ has the same (sparse) structure as K_j.





2. Envelope methods

- Sparse matrices have only a small number of nonzero entries.
- Well known example: Band matrices.



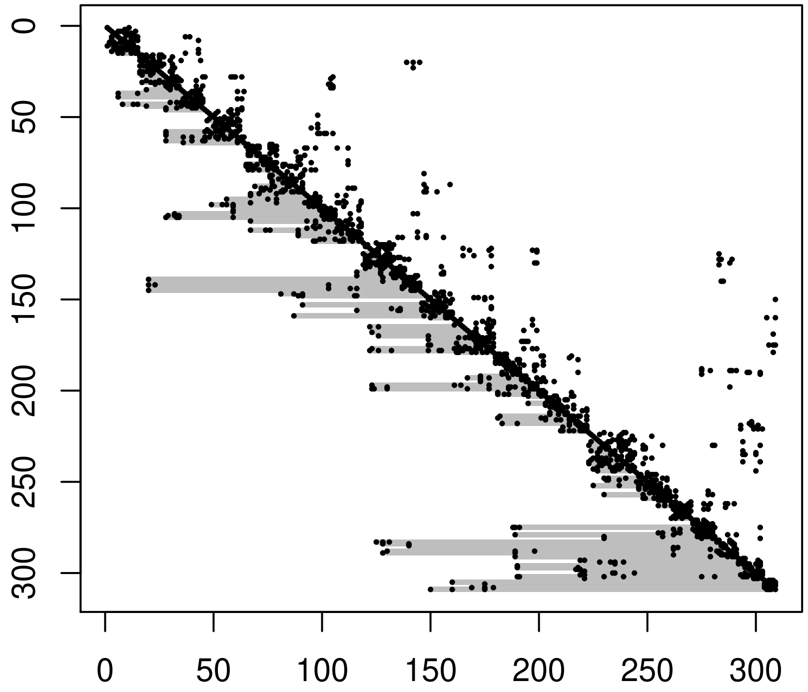
- More general: matrices with *local bandwidth*.

- Methods for matrices with local bandwidth are called *envelope methods* (George & Liu, 1981).

- Define for a sparse matrix $A = (a_{ij})$:

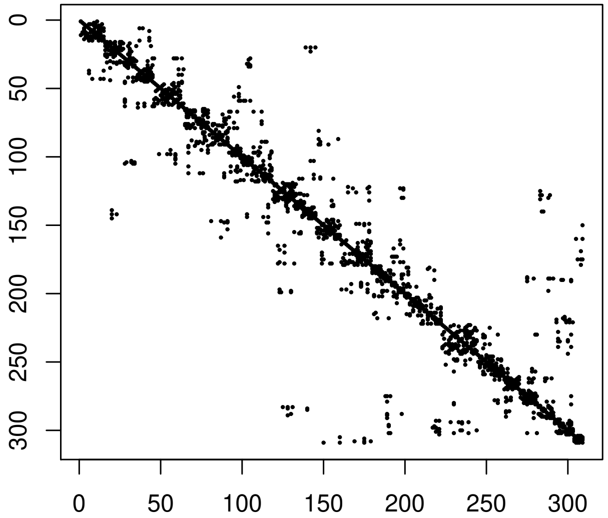
$$b_i = i - \min\{j : a_{ij} \neq 0\} \qquad \text{local bandwidth in row } i$$

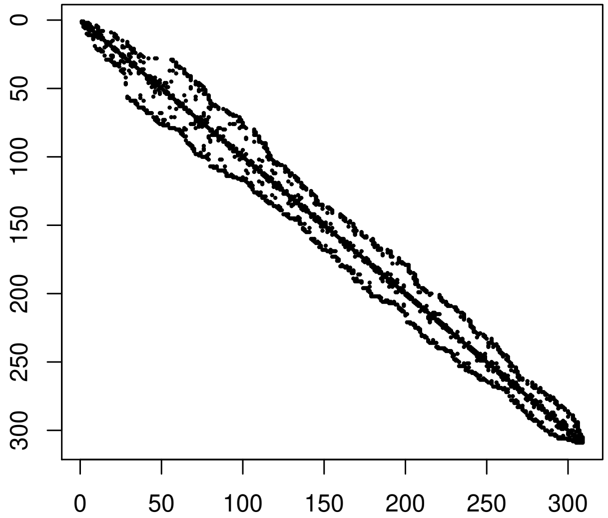
$$Env(A) = \{a_{i,i-j}, j = 1, \dots, b_i, i = 1, \dots, n\} \qquad \text{envelope of } A$$



3. Reordering to minimize the envelope

- Reordering a matrix may lead to a considerable reduction of the envelope.
- A smaller envelope implies faster computations.
- Reordering also reduces the maximal bandwidth, but envelope methods are still superior to band methods.

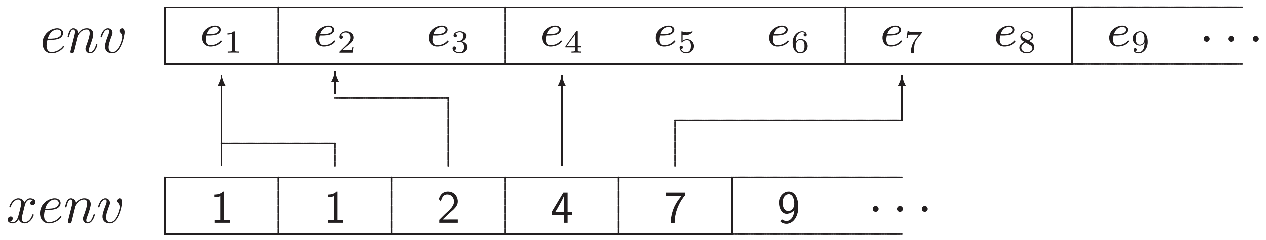




Maximal bandwidth:	161	28
Size of the Envelope:	6456	3857

4. Storing envelope matrices

- Storing a complete $p \times p$ matrix requires p^2 storage locations.
- Reduces to $p + p(p-1)/2$ for a symmetric matrix.
- Sparse matrices: Store only the diagonal elements and the elements of the envelope.
  - Vector $diag$ containing the diagonal elements.
  - Vector env containing the envelope elements row by row.
  - Vector $xenv$ of indices pointing to the start of each row portion in env.
- Results in $2p + |Env(A)|$ storage locations.



- The local bandwidths can be computed as

$$b_i = xenv(i+1) - xenv(i).$$

5. Solving systems of linear equations

- Solve $Ax = c$ with A sparse (and positive definite).

- Usual approach based on Cholesky decomposition of A:
  - Decompose $A = LL^T$ with L lower triangular ($O(p^3)$ computation steps).
  - Solve $L\nu = c$ by forward substitution ($O(p^2)$ computation steps).
  - Solve $L^T x = \nu$ by backward substitution ($O(p^2)$ computation steps).

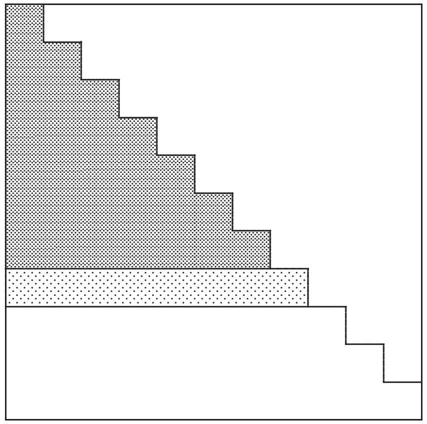
- Simplified Cholesky decomposition for sparse matrices:
  - L has the same local bandwidths as the original matrix A.
  - Take advantage of the local bandwidth.

$\Rightarrow$ $O(pb^2)$ computation steps for a band matrix with bandwidth b.

- Algorithm: Cholesky decomposition $A = LL^T$

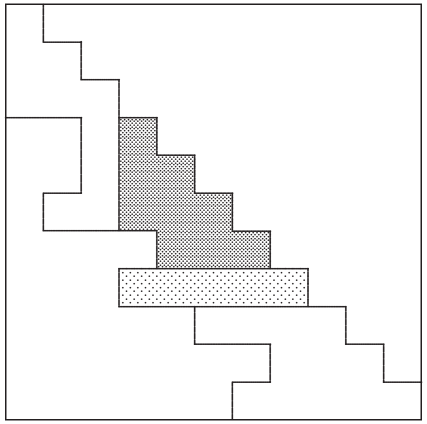
Full matrices:

$$\begin{aligned}
&i = 1, \dots, p \\
&\quad k = 1, \dots, i-1 \\
&\qquad l_{ik} = \left(a_{ik} - \sum_{j=1}^{k-1} l_{kj} l_{ij} \right) / l_{kk} \\
&\quad l_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2 \right)^{\frac{1}{2}}
\end{aligned}$$

i


Sparse matrices:

$$\begin{aligned}
&i = 1, \dots, p \\
&\quad k = i - b_i, \dots, i-1 \\
&\qquad l_{ik} = \left(a_{ik} - \sum_{j=i-\min\{b_i, b_k\}}^{k-1} l_{kj} l_{ij} \right) / l_{kk} \\
&\quad l_{ii} = \left(a_{ii} - \sum_{j=i-b_i}^{i-1} l_{ij}^2 \right)^{\frac{1}{2}}
\end{aligned}$$

i


- Forward and backward substitution can be simplified in a similar way.

$\Rightarrow$ $O(pb)$ for a bandmatrix with bandwidth b.

6. The inverse of a matrix

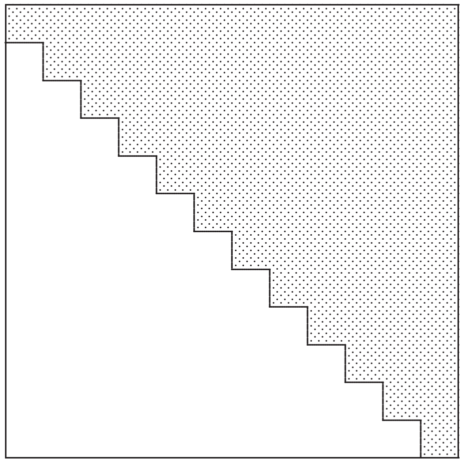
- In general the inverse $B = A^{-1}$ of a sparse matrix A is not sparse.
- In some applications we do not need the whole inverse but only parts of it.
- Based on the rational cholesky decomposition $A = UD^{-1}U^T$ we have

$$A^{-1} = U^{-T}D + A^{-1}(I - U).$$

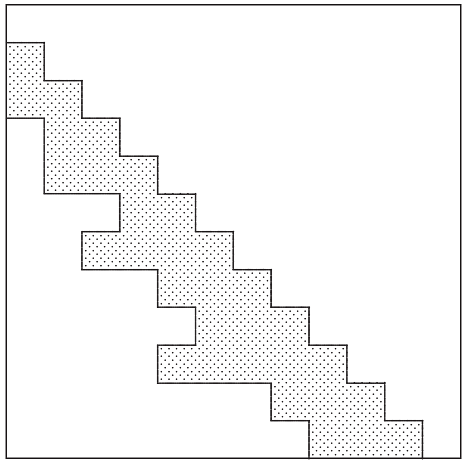
- Matrices involved have special structure: U is unit root lower triangular with the same local bandwidths as A, $U^{-T}D$ is upper triangular with $diag(U^{-T}D) = D$.
- This representation allows the computation of the elements of A^{-1} corresponding to the envelope of A based on a recurrence formula.
- Restriction: We can not compute less than the 'envelope' of the inverse!

- The 'envelope' elements of the inverse are computed row by row starting with the last element in the last row.

$B =$



$+\ B$



DU^{-T} $\qquad$ $(I-U)$

$$
\begin{aligned}
b_{pp} &= d_{pp}\\
b_{p,p-1} &= -b_{pp}u_{p,p-1}\\
b_{p,p-2} &= -b_{p,p-1}u_{p-1,p-2} - b_{pp}u_{p,p-1}\\
\dots\ & \quad \dots\\
b_{p-1,p-1} &= d_{p-1,p-1} - b_{p-1,p}u_{p,p-1}\\
\dots\ & \quad \dots
\end{aligned}
$$

7. Structured additive regression (revisited)

- Sampling from the full conditional

$$N(\underbrace{(X_j^T X_j + \lambda_j K_j)^{-1} X_j^T (y - \eta_{-j})}_{=m_j}, \underbrace{(X_j^T X_j + \lambda_j K_j)^{-1}}_{=P_j^{-1}})$$

  can be done as follows:

  - Simulate $z \sim N(0, I)$.
  - Decompose $P_j = LL^T$.
  - Solve $L^T x = z$. Now x is a random number from $N(0, P_j^{-1})$.
  - Solve $P_j m_j = X_j^T (y - \eta_{-j})$.
  - Adding m_j to x yields a random number from the desired distribution.

- Similar techniques may be used in frequentist approaches based on backfitting.

- Model Selection: Note that
  - an elementary property of the trace gives

$$df(f_j) = tr(\underbrace{X_j^T X_j}_{=A} \underbrace{(X_j^T X_j + \lambda_j K_j)^{-1}}_{=B}),$$

  - due to the symmetry of A and B we have

$$tr(A \cdot B) = \sum_{i,j} a_{ij} b_{ij}$$

  - and the envelope of $(X_j^T X_j + \lambda_j K_j)$ always exceeds the envelope of $X_j^T X_j$.

- Based on these findings we may compute the degrees of freedom associated with f_j in the following way (Hutchinson & de Hoog, 1985):
  - Compute the 'envelope' of B.
  - Compute

$$df(f_j) = \sum_{i=1}^{p} a_{ii}b_{ii} + 2 \sum_{i,j: a_{ij} \in Env(A)} a_{ij}b_{ij}.$$

- Order of computation: Reduces from $O(p_j^3)$ to $O(p_j b^2)$ for a band matrix with bandwidth b.

- The presented methods are used in the public domain software *BayesX* for inference in generalized structured additive regression models.

- *BayesX* is developed jointly by Andreas Brezger, Thomas Kneib and Stefan Lang and is available from

`http://www.stat.uni-muenchen.de/~lang/bayesx/bayesx.html`

References

George, A. & Liu, J. (1981). *Computer solution of large sparse positive definite systems*, Prentice Hall, London.

Hutchinson, M. F. & de Hoog, F. R. (1985). Smoothing noisy data with spline functions, *Numerische Mathematik* **47**: 99–106.

Rue, H. (2001). Fast sampling of Gaussian Markov random fields, *Journal of the Royal Statistical Society B* **63**: 325–338.