Plan for today

- Recall Latent Gaussian models and INLA
- Template Model Builder.

Latent Gaussian model

1. Observed data $\boldsymbol{y} = (y_1, \ldots, y_n)$ where

$$\pi(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\eta}) = \pi(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{j=1}^{n} \pi(y_j \mid x_j)$$

   Often exponential family: Normal, Poisson, binomial, etc.
   $\log \pi(y_j|x_j) = \frac{y_j x_j - b(x_j)}{a(\phi)} + c(\phi, y_j)$. $b(x)$ canonical link.

2. Latent Gaussian process $\boldsymbol{x} = (x_1, \ldots, x_n)$

$$\pi(\boldsymbol{x} \mid \boldsymbol{\eta}) = N[\boldsymbol{\mu},\ \boldsymbol{\Sigma}(\boldsymbol{\eta})]$$

3. Prior for hyperparameters $\pi(\boldsymbol{\eta})$ (Bayesian).
4. Hyperparameters considered fixed, but unknown (Frequentist).

Inference and prediction in Latent Gaussian models

Around 1990-2000s, **Markov chain Monte Carlo** was very popular for Bayesian inference and prediction in latent Gaussian models.
Today MCMC is still very popular, but not so much for latent Gaussian models. Alternatives are **Laplace approximations** or INLA.

Idea behind INLA Inference

Split the joint density

$$\pi(\boldsymbol{x}, \boldsymbol{\eta}, \boldsymbol{y}) = \pi(\boldsymbol{\eta})\pi(\boldsymbol{x}|\boldsymbol{\eta})\pi(\boldsymbol{y} \mid \boldsymbol{x}) = \pi(\boldsymbol{y})\pi(\boldsymbol{\eta} \mid \boldsymbol{y})\pi(\boldsymbol{x} \mid \boldsymbol{\eta}, \boldsymbol{y})$$

Clearly:

$$\pi(\boldsymbol{\eta} \mid \boldsymbol{y}) = \frac{\pi(\boldsymbol{\eta})\pi(\boldsymbol{x}|\boldsymbol{\eta})\pi(\boldsymbol{y} \mid \boldsymbol{x})}{\pi(\boldsymbol{y})\pi(\boldsymbol{x} \mid \boldsymbol{\eta}, \boldsymbol{y})} \propto \frac{\pi(\boldsymbol{\eta})\pi(\boldsymbol{x}|\boldsymbol{\eta})\pi(\boldsymbol{y} \mid \boldsymbol{x})}{\pi(\boldsymbol{x} \mid \boldsymbol{\eta}, \boldsymbol{y})}$$

Marginalization:

$$\pi(\boldsymbol{x}_j \mid \boldsymbol{y}) = \int_{\boldsymbol{\eta}} \pi(\boldsymbol{\eta} \mid \boldsymbol{y})\pi(\boldsymbol{x}_j \mid \boldsymbol{\eta}, \boldsymbol{y}) d\boldsymbol{\eta}$$

Inference

Laplace approximation

$$\hat{\pi}(\boldsymbol{\eta} \mid \boldsymbol{y}) \propto \left. \frac{\pi(\boldsymbol{\eta})\pi(\boldsymbol{x}|\boldsymbol{\eta})\pi(\boldsymbol{y} \mid \boldsymbol{x})}{\hat{\pi}(\boldsymbol{x} \mid \boldsymbol{\eta}, \boldsymbol{y})} \right|_{\boldsymbol{x}=\hat{\boldsymbol{m}}(\boldsymbol{\eta},\boldsymbol{y})}$$

Use a *Gaussian* approximation to full conditional $\hat{\pi}(\boldsymbol{x} \mid \boldsymbol{\eta}, \boldsymbol{y})$.
$\hat{\boldsymbol{m}} = \hat{\boldsymbol{m}}(\eta, \boldsymbol{y}) = argmax_{\boldsymbol{x}}[\pi(\boldsymbol{x}|\boldsymbol{\eta})\pi(\boldsymbol{y} \mid \boldsymbol{x})]$.

Gaussian approximation of full posterior

$$\pi(\boldsymbol{x} \mid \boldsymbol{\eta}, \boldsymbol{y}) \propto \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) + \sum_{j=1}^{n} \log \pi(y_j|x_j)\right)$$

- $\log \pi(y_j|x_j) = \frac{y_j x_j - b(x_j)}{a(\phi)} + c(\phi, y_j)$. $b(x)$ is canonical link. (Poisson likelihood : $b(x) = m\exp(x)$. Binomial : $b(x) = m\log[1+\exp(x)]$. Fixed m.)
- Expand GLM part $\log \pi(y_j|x_j)$ to second order.
- Iterative solution to posterior mode $\hat{\boldsymbol{m}} = \hat{\boldsymbol{m}}(\eta, \boldsymbol{y})$. ('Scoring').
- $\hat{\boldsymbol{m}} = \boldsymbol{\mu} - \boldsymbol{\Sigma}\boldsymbol{A}'[\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}' + \boldsymbol{P}]^{-1}(\boldsymbol{z}(\boldsymbol{y}, \hat{\boldsymbol{m}}) - \boldsymbol{A}\boldsymbol{\mu})$.
- Fit Gaussian approximation from Hessian at posterior mode: $\hat{\pi}(\boldsymbol{x} \mid \boldsymbol{\eta}, \boldsymbol{y}) = N(\hat{\boldsymbol{m}}, \hat{\boldsymbol{V}})$.
- $\boldsymbol{P} = \boldsymbol{P}(\hat{\boldsymbol{m}})$. Size $n \times n$ matrix factorization required.

Expansion - at each iteration to reach conditional mode

$$\pi(\boldsymbol{x} \mid \boldsymbol{\eta}, \boldsymbol{y}) \propto \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) + \sum_{j=1}^{n} y_j x_j - b(x_j^0) - b'(x_j)(x_j - x_j^0)\right.$$

The Gaussian approximation to full conditional has

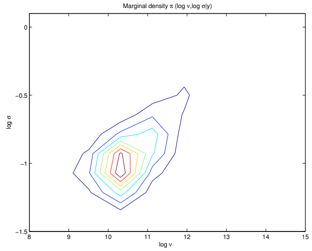
$$\hat{\boldsymbol{V}}_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\eta}}(\boldsymbol{x}^0) = \boldsymbol{\Sigma} - \boldsymbol{\Sigma}\boldsymbol{R}^{-1}\boldsymbol{\Sigma}, \qquad \boldsymbol{R} = \boldsymbol{\Sigma} + \text{diag}(1/b''(x_j^0))$$

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\eta}}(\boldsymbol{y},\boldsymbol{x}^0) = \boldsymbol{\mu} + \boldsymbol{\Sigma}\boldsymbol{R}^{-1}[\boldsymbol{z}(\boldsymbol{y},\boldsymbol{x}^0) - \boldsymbol{\mu}]$$

$$z_j(y_j, x_j^0) = [y_j - b'(x_j^0) + x_j^0 b''(x_j^0)]/b''(x_j^0), \qquad j = 1, \ldots, n$$

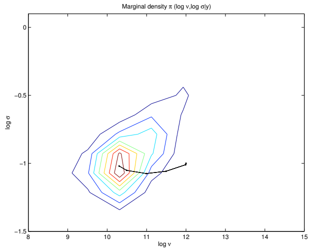
Practical implementation

Numerical approximation of $\hat{\pi}(\boldsymbol{\eta}|\boldsymbol{y})$



Practical implementation

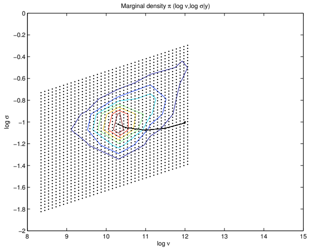
Numerical approximation of $\hat{\pi}(\boldsymbol{\eta}|\boldsymbol{y})$, Step 1: Find mode



Each step requires $\boldsymbol{m}(\boldsymbol{\eta}, \boldsymbol{y})$, $\hat{\pi}(\boldsymbol{x} \mid \boldsymbol{\eta}, \boldsymbol{y})$ and Laplace.

Practical implementation

Numerical approximation of $\hat{\pi}(\boldsymbol{\eta}|\boldsymbol{y})$, Step 2: Use Hessian at mode to set grid



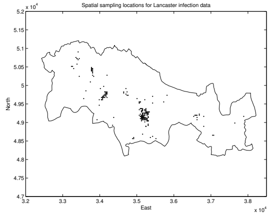
Nested approximation of $\pi(x_j|\boldsymbol{y})$

$$\pi(x_j|\boldsymbol{y},\boldsymbol{\theta}) \propto \frac{\pi(\boldsymbol{y}|\boldsymbol{x})\pi(\boldsymbol{x}|\boldsymbol{\theta})}{\pi(\boldsymbol{x}_{-j}|x_j,\boldsymbol{y},\boldsymbol{\theta})},$$

Using the Laplace approximation again, for fixed x_j.
$\hat{\pi}(\boldsymbol{x}_{-j}|x_j,\boldsymbol{y},\boldsymbol{\theta})$ approximated by a Gaussian (for each x_j on a grid or design points).

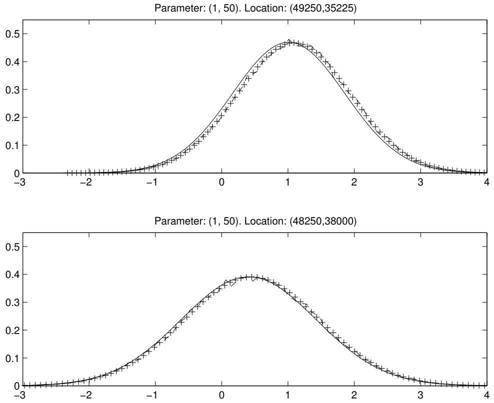
Example: Lancaster disease map

- Number of infections in different regions.
- Binomial data (with small counts).



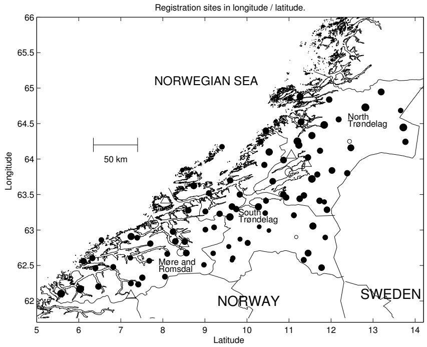
Example: Lancaster disease map

LA, INLA and MCMC prediction at one site, for two parameter sets.



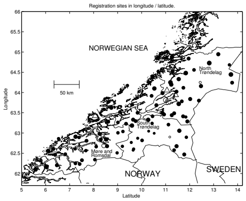
Example: Precipitation in Middle Norway

Number of days with rain for $k = 92$ sites in September-October 2006.



Example: Precipitation in Middle Norway

Binomial data $y_j = \mathrm{Binomial}[\frac{e^{x_j}}{1+e^{x_j}}, 61]$.
Standard GLM gives no significance to East, North, Altitude.
Include only spatial trend.



- Outlier detection
- Spatial design

Outlier detection

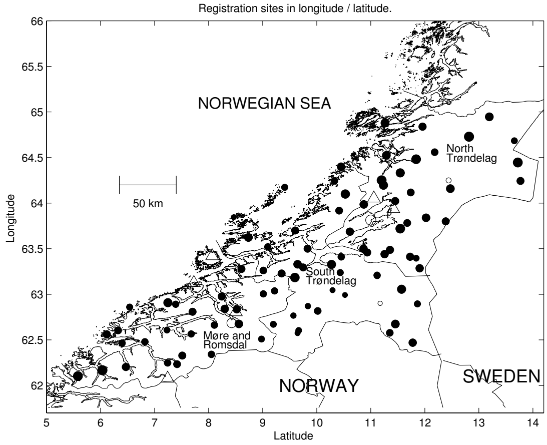
Use crossvalidation $\pi(y_j \mid \boldsymbol{y}_{-j})$.

$$\hat{\pi}(y_j \mid \boldsymbol{y}_{-j}) = \int_{x_j} \sum_{l} \hat{\pi}(\boldsymbol{\eta}_l \mid \boldsymbol{y}_{-j}) \hat{\pi}(x_j \mid \boldsymbol{\eta}_l, \boldsymbol{y}_{-j}) \pi(y_j \mid x_j) dx_j$$

Inference separately for each y_j. I.e. n times. Approximate predictive percentiles
$\sum_{y_j=0}^{y_{lower}} \hat{\pi}(y_j \mid \boldsymbol{y}_{-j}) = \alpha/2$, $\sum_{y_j=0}^{y_{upper}} \hat{\pi}(y_j \mid \boldsymbol{y}_{-j}) = 1 - \alpha/2$.
Compare (y_{lower}, y_{upper}) with observed y_j.

Results : Outlier detection

Results $\alpha/2 = 0.01$: detect 4 outliers (open circles).



Spatial design

Prospective view: $\boldsymbol{y} \rightarrow (\boldsymbol{y}, \boldsymbol{y}_a)$.
$\boldsymbol{y}_a$ extra data at 'new' spatial registration sites.
'Imagine' these observations - do not acquire them.

Design criterion is: Integrated prediction variance.

$$\hat{I} = \sum_{\boldsymbol{y}_a} \sum_{j} \hat{V}(x_j \mid \boldsymbol{y}, \boldsymbol{y}_a)\hat{\pi}(\boldsymbol{y}_a|\boldsymbol{y})$$

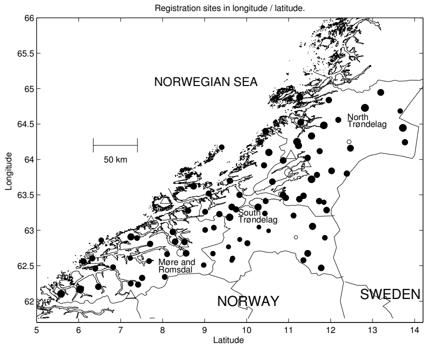
Results: Spatial design

Results of *three* design.

0: Existing design with 88 points (outliers excluded).

A: Currently installed stations, 88 plus 10 known sites (4 outlier sites and 6 sites out of service).

B: 88 plus 10 $= 2 \cdot 5$ new random sites around two existing sites (50km radius).



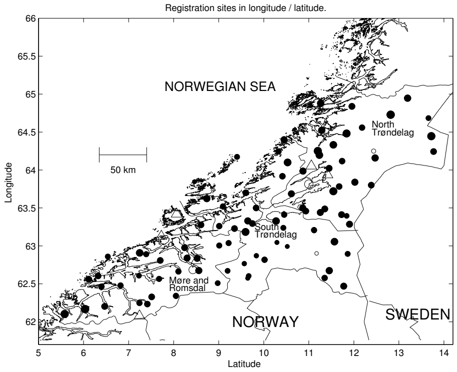
Results: Spatial design

Results of *three* designs.

0: Existing design: $\hat{I}_0 = 18.68$.

A: Currently installed stations: $\hat{I}_A = 17.94$.

B: Random around two existing sites: $\hat{I}_B = 17.85$



INLA software

INLA software: http://www.r-inla.org
Easy to call:
>inla(y x + f(nu,model=”iid”), family = c(”poisson”), data = data, control.predictor=list(link=1))
Rue et al. (2009)
Routine runs on Gaussian Markov random fields.

(Project Feb 20)

Template model builder (TMB)

Frequentist analysis (to latent Gaussian models), which fits the maximum marginal likelihood estimates. Then plug this in for prediction, or approximate the Fisher information of the estimate.
The marginalization over latent variables is approximated by the **Laplace approximation**.
The optimization requires *derivatives*.

Laplace approximation (again)

Version 1:

$$l(\boldsymbol{\eta}; \boldsymbol{y}) \propto \left. \frac{\pi_{\boldsymbol{\eta}}(\boldsymbol{x})\pi_{\boldsymbol{\eta}}(\boldsymbol{y} \mid \boldsymbol{x})}{\pi_{\boldsymbol{\eta}}(\boldsymbol{x} \mid \boldsymbol{y})} \right|_{\boldsymbol{x}=\hat{\boldsymbol{m}}(\boldsymbol{\eta},\boldsymbol{y})}$$

(Approximate denominator by Gaussian.)

Version 2:

$$l(\boldsymbol{\eta}; \boldsymbol{y}) = \int_{\boldsymbol{x}} \pi_{\boldsymbol{\eta}}(\boldsymbol{x})\pi_{\boldsymbol{\eta}}(\boldsymbol{y} \mid \boldsymbol{x}) d\boldsymbol{x}$$

(Approximate integral by quadratic form.)

Laplace approximation (integral solution)

Version 2:

$$L^*(\boldsymbol{\eta}) = L^*(\boldsymbol{\eta}; \boldsymbol{y}) = (2\pi)^{n/2} |H(\boldsymbol{\eta})|^{1/2} \exp[f(\hat{\boldsymbol{x}}, \boldsymbol{\eta})]$$

(Expand exponent to a quadratic form.)

$$\hat{\boldsymbol{x}} = argmax_{\boldsymbol{x}}[\pi_{\boldsymbol{\eta}}(\boldsymbol{x})\pi_{\boldsymbol{\eta}}(\boldsymbol{y} \mid \boldsymbol{x})]$$
$$f(\boldsymbol{x}, \boldsymbol{\eta}) = \log[\pi_{\boldsymbol{\eta}}(\boldsymbol{x})\pi_{\boldsymbol{\eta}}(\boldsymbol{y} \mid \boldsymbol{x})]$$

$$H(\boldsymbol{\eta}) = -\frac{d^2 f(\hat{\boldsymbol{x}}, \boldsymbol{\eta})}{d\boldsymbol{x}^2}$$

Maximum likelihood estimator

$$\hat{\boldsymbol{\eta}} = argmax_{\boldsymbol{\eta}} L^*(\boldsymbol{\eta})$$

This has nice asymptotic properties (when data size n goes to infinity), under some regularity conditions.
For instance, the MLE is asymptotically normal, and has a variance defined by the second derivative of $L^*(\boldsymbol{\eta})$ at the MLE $\hat{\boldsymbol{\eta}}$.
MLE and properties of MLE requires derivatives!

Generality of Laplace approximation

- Fast and reliable for Latent Gaussian models (Gaussian approximation to full conditional).
- More generally applicable if stable derivatives.

Automatic differentiation (AD)

The essence of TMB is stable differentiation using software for calculating derivatives of all elementary operations. And then combining this for reliable optimization.
The code is usually compiler-based, so first and second derivatives are compiled together with the original program. (TMB calls C++ routines, but you don't see them.)

AD and elementary operations

- partial derivatives are computed for elementary operation; binary (plus, minus, multiplication, division) and unary (log, exp)
- savings by organizing computations in a graph (forward-backward extraction of derivatives).

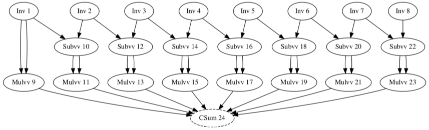


Figure 1: **CppAD** tape T1 for $f(\xi_1,\ldots,\xi_8) = \xi_1^2 + \sum_{i=2}^{8}(\xi_i - \xi_{i-1})^2$. Nodes "Inv 1"–"Inv 8" correspond to $\xi_1,\ldots,\xi_8$ and node "CSum 24" corresponds to $f(\xi_1,\ldots,\xi_8)$. Node labels indicate the elementary operations, numbering indicates the order in these operations are evaluated, arrows point from operation arguments to results, double arrows correspond to the square operator `x^2` which is implemented as `x*x`.

General automatic differentiation (AD)

General form

$$t = \phi(r, s)$$

$$t' = \frac{d\phi}{dr} r' + \frac{d\phi}{ds} s'$$

Example:

$$t = \phi(r, s) = rs$$

$$t' = sr' + rs'$$

Each time $t = rs$ is evaluated, $t' = sr' + rs'$ is also computed automatically.

AD for MLE

$$\hat{\boldsymbol{\eta}} = argmax_{\boldsymbol{\eta}} L^*(\boldsymbol{\eta})$$

Nested optimizations:

- Inner optimization : $\hat{\boldsymbol{x}}(\boldsymbol{\eta}, \boldsymbol{y})$, usually some kind of Newton method.
- Outer optimization for $\hat{\boldsymbol{\eta}}$, usually some kind of quasi Newton.

Both these benefit from exact derivatives from AD. Each time a function is called, its derivatives are automatically provided.
This is usually applicable for a wide range of non-linear regression problems with random effects.

TMB software

TMB software: $model <- makeADfun()$
Kristensen et al. (2015)

Watch TMB tutorial video:
https://www.youtube.com/watch?v=A5CLrhzNzVU&t=

(Project Feb 20)