INLA: an introduction

Håvard Rue[1]
Norwegian University of Science and Technology
Trondheim, Norway

May 2009

[1] Joint work with S.Martino (Trondheim) and N.Chopin (Paris)

Latent Gaussian models

Stage 1 Observed data $\mathbf{y} = (y_i)$,

$$y_i \mid \mathbf{x}, \boldsymbol{\theta} \sim \pi(y_i | x_i, \boldsymbol{\theta})$$

Stage 2 Latent Gaussian field

$$\mathbf{x} \mid \boldsymbol{\theta} \sim \mathcal{N}(\mu, \mathbf{Q}(\boldsymbol{\theta})^{-1}), \qquad \mathbf{A}\mathbf{x} = \mathbf{0}$$

Stage 3 Priors for the hyperparameters

$$\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$$

Unify many of the most used models in statistics

Latent Gaussian models

Stage 1 Observed data $\mathbf{y} = (y_i)$,

$$y_i \mid \mathbf{x}, \boldsymbol{\theta} \sim \pi(y_i | x_i, \boldsymbol{\theta})$$

Stage 2 Latent Gaussian field

$$\mathbf{x} \mid \boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{Q}(\boldsymbol{\theta})^{-1}), \qquad \mathbf{A}\mathbf{x} = \mathbf{0}$$

Stage 3 Priors for the hyperparameters

$$\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$$

Unify many of the most used models in statistics

Latent Gaussian models

Stage 1 Observed data $\mathbf{y} = (y_i)$,

$$y_i \mid \mathbf{x}, \boldsymbol{\theta} \sim \pi(y_i | x_i, \boldsymbol{\theta})$$

Stage 2 Latent Gaussian field

$$\mathbf{x} \mid \boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{Q}(\boldsymbol{\theta})^{-1}), \qquad \mathbf{Ax} = \mathbf{0}$$

Stage 3 Priors for the hyperparameters

$$\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$$

Unify many of the most used models in statistics

Latent Gaussian models

Stage 1 Observed data $\mathbf{y} = (y_i)$,

$$y_i \mid \mathbf{x}, \boldsymbol{\theta} \sim \pi(y_i | x_i, \boldsymbol{\theta})$$

Stage 2 Latent Gaussian field

$$\mathbf{x} \mid \boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{Q}(\boldsymbol{\theta})^{-1}), \qquad \mathbf{Ax} = \mathbf{0}$$

Stage 3 Priors for the hyperparameters

$$\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$$

Unify many of the most used models in statistics

Structured additive regression models

Linear predictor

$$\eta_i = \sum_k \beta_k z_{ki} + \sum_j w_{ji} f_j(z_{ji}) + \epsilon_i$$

- Linear effects of covariates $\{z_{ki}\}$
- Effects of $f_j(\cdot)$
  - Fixed weights $\{w_{ji}\}$
  - Commonly: $f_j(z_{ji}) = f_{j,z_{ji}}$
  - Account for smooth response
    - Temporal or spatially indexed covariates
    - Unstructured terms ("random effects")
  - Depend on some parameters θ

Structured additive regression models

Linear predictor

$$\eta_i = \sum_k \beta_k z_{ki} + \sum_j w_{ji} f_j(z_{ji}) + \epsilon_i$$

- Linear effects of covariates $\{z_{ki}\}$
- Effects of $f_j(\cdot)$
  - Fixed weights $\{w_{ji}\}$
  - Commonly: $f_j(z_{ji}) = f_{j,z_{ji}}$
  - Account for smooth response
    - Temporal or spatially indexed covariates
    - Unstructured terms ("random effects")
  - Depend on some parameters $\boldsymbol{\theta}$

Structured additive regression models

Linear predictor

$$\eta_i = \sum_k \beta_k z_{ki} + \sum_j w_{ji} f_j(z_{ji}) + \epsilon_i$$

- Linear effects of covariates $\{z_{ki}\}$
- Effects of $f_j(\cdot)$
  - Fixed weights $\{w_{ji}\}$
  - Commonly: $f_j(z_{ji}) = f_{j,z_{ji}}$
  - Account for smooth response
    - Temporal or spatially indexed covariates
    - Unstructured terms ("random effects")
  - Depend on some parameters $\boldsymbol{\theta}$

Structured additive regression models

Linear predictor

$$\eta_i = \sum_k \beta_k z_{ki} + \sum_j w_{ji} f_j(z_{ji}) + \epsilon_i$$

- Linear effects of covariates $\{z_{ki}\}$
- Effects of $f_j(\cdot)$
  - Fixed weights $\{w_{ji}\}$
  - Commonly: $f_j(z_{ji}) = f_{j,z_{ji}}$
  - Account for smooth response
    - Temporal or spatially indexed covariates
    - Unstructured terms ("random effects")
  - Depend on some parameters $\boldsymbol{\theta}$

Structured additive regression models

Linear predictor

$$\eta_i = \sum_k \beta_k z_{ki} + \sum_j w_{ji} f_j(z_{ji}) + \epsilon_i$$

- Linear effects of covariates $\{z_{ki}\}$
- Effects of $f_j(\cdot)$
  - Fixed weights $\{w_{ji}\}$
  - Commonly: $f_j(z_{ji}) = f_{j,z_{ji}}$
  - Account for smooth response
    - Temporal or spatially indexed covariates
    - Unstructured terms ("random effects")
  - Depend on some parameters $\boldsymbol{\theta}$

Structured additive regression models

Linear predictor

$$\eta_i = \sum_k \beta_k z_{ki} + \sum_j w_{ji} f_j(z_{ji}) + \epsilon_i$$

- Linear effects of covariates $\{z_{ki}\}$
- Effects of $f_j(\cdot)$
  - Fixed weights $\{w_{ji}\}$
  - Commonly: $f_j(z_{ji}) = f_{j,z_{ji}}$
  - Account for smooth response
    - Temporal or spatially indexed covariates
    - Unstructured terms ("random effects")
  - Depend on some parameters $\boldsymbol{\theta}$

Structured additive regression models

Linear predictor

$$\eta_i = \sum_k \beta_k z_{ki} + \sum_j w_{ji} f_j(z_{ji}) + \epsilon_i$$

- Linear effects of covariates $\{z_{ki}\}$
- Effects of $f_j(\cdot)$
  - Fixed weights $\{w_{ji}\}$
  - Commonly: $f_j(z_{ji}) = f_{j,z_{ji}}$
  - Account for smooth response
    - Temporal or spatially indexed covariates
    - Unstructured terms ("random effects")
  - Depend on some parameters $\boldsymbol{\theta}$

Structured additive regression models

Linear predictor

$$\eta_i = \sum_k \beta_k z_{ki} + \sum_j w_{ji} f_j(z_{ji}) + \epsilon_i$$

- Linear effects of covariates $\{z_{ki}\}$
- Effects of $f_j(\cdot)$
  - Fixed weights $\{w_{ji}\}$
  - Commonly: $f_j(z_{ji}) = f_{j,z_{ji}}$
  - Account for smooth response
    - Temporal or spatially indexed covariates
    - Unstructured terms ("random effects")
  - Depend on some parameters $\boldsymbol{\theta}$

Structured additive regression models

$$\eta_i = \sum_k \beta_k z_{ki} + \sum_j w_{ji} f_j(z_{ji}) + \epsilon_i$$

- Observations $\mathbf{y}$

$$\pi(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) = \prod_i \pi(y_i \mid \eta_i, \boldsymbol{\theta})$$

  from an (f.ex) exponential family with mean $\mu_i = g^{-1}(\eta_i)$.
- *Latent Gaussian model* if

$$\mathbf{x} = (\{\beta_k\}, \{f_{ji}\}, \{\eta_i\}) \mid \boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{Q}(\boldsymbol{\theta})^{-1})$$

Structured additive regression models

$$\eta_i = \sum_k \beta_k z_{ki} + \sum_j w_{ji} f_j(z_{ji}) + \epsilon_i$$

- Observations $\mathbf{y}$

$$\pi(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) = \prod_i \pi(y_i \mid \eta_i, \boldsymbol{\theta})$$

  from an (f.ex) exponential family with mean $\mu_i = g^{-1}(\eta_i)$.
- *Latent Gaussian model* if

$$\mathbf{x} = (\{\beta_k\}, \{f_{ji}\}, \{\eta_i\}) \mid \boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{Q}(\boldsymbol{\theta})^{-1})$$

Structured additive regression models

$$\eta_i = \sum_k \beta_k z_{ki} + \sum_j w_{ji} f_j(z_{ji}) + \epsilon_i$$

- Observations $\mathbf{y}$

$$\pi(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) = \prod_i \pi(y_i \mid \eta_i, \boldsymbol{\theta})$$

  from an (f.ex) exponential family with mean $\mu_i = g^{-1}(\eta_i)$.
- *Latent Gaussian model* if

$$\mathbf{x} = (\{\beta_k\}, \{f_{ji}\}, \{\eta_i\}) \mid \boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{Q}(\boldsymbol{\theta})^{-1})$$

Examples

- Dynamic linear models
- Stochastic volatility
- Generalised linear (mixed) models
- Generalised additive (mixed) models
- Spline smoothing
- Semiparametric regression
- Space-varying (semiparametric) regression models
- Disease mapping
- Log-Gaussian Cox-processes
- Model-based geostatistics (*)
- Spatio-temporal models

Examples

- Dynamic linear models
- Stochastic volatility
- Generalised linear (mixed) models
- Generalised additive (mixed) models
- Spline smoothing
- Semiparametric regression
- Space-varying (semiparametric) regression models
- Disease mapping
- Log-Gaussian Cox-processes
- Model-based geostatistics (*)
- Spatio-temporal models

Examples

- Dynamic linear models
- Stochastic volatility
- Generalised linear (mixed) models
- Generalised additive (mixed) models
- Spline smoothing
- Semiparametric regression
- Space-varying (semiparametric) regression models
- Disease mapping
- Log-Gaussian Cox-processes
- Model-based geostatistics (*)
- Spatio-temporal models

Examples

- Dynamic linear models
- Stochastic volatility
- Generalised linear (mixed) models
- Generalised additive (mixed) models
- Spline smoothing
- Semiparametric regression
- Space-varying (semiparametric) regression models
- Disease mapping
- Log-Gaussian Cox-processes
- Model-based geostatistics (*)
- Spatio-temporal models

Examples

- Dynamic linear models
- Stochastic volatility
- Generalised linear (mixed) models
- Generalised additive (mixed) models
- Spline smoothing
- Semiparametric regression
- Space-varying (semiparametric) regression models
- Disease mapping
- Log-Gaussian Cox-processes
- Model-based geostatistics (*)
- Spatio-temporal models

Examples

- Dynamic linear models
- Stochastic volatility
- Generalised linear (mixed) models
- Generalised additive (mixed) models
- Spline smoothing
- Semiparametric regression
- Space-varying (semiparametric) regression models
- Disease mapping
- Log-Gaussian Cox-processes
- Model-based geostatistics (*)
- Spatio-temporal models

Examples

- Dynamic linear models
- Stochastic volatility
- Generalised linear (mixed) models
- Generalised additive (mixed) models
- Spline smoothing
- Semiparametric regression
- Space-varying (semiparametric) regression models
- Disease mapping
- Log-Gaussian Cox-processes
- Model-based geostatistics (*)
- Spatio-temporal models

Examples

- Dynamic linear models
- Stochastic volatility
- Generalised linear (mixed) models
- Generalised additive (mixed) models
- Spline smoothing
- Semiparametric regression
- Space-varying (semiparametric) regression models
- Disease mapping
- Log-Gaussian Cox-processes
- Model-based geostatistics (*)
- Spatio-temporal models

Examples

- Dynamic linear models
- Stochastic volatility
- Generalised linear (mixed) models
- Generalised additive (mixed) models
- Spline smoothing
- Semiparametric regression
- Space-varying (semiparametric) regression models
- Disease mapping
- Log-Gaussian Cox-processes
- Model-based geostatistics (*)
- Spatio-temporal models

Examples

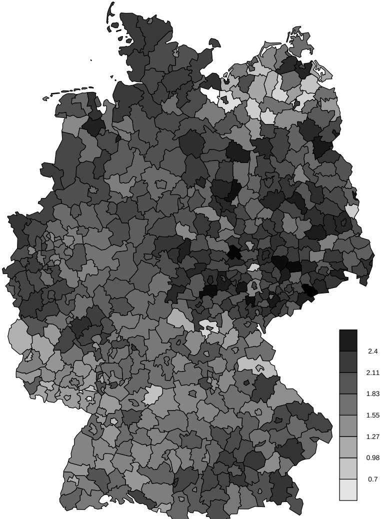
- Dynamic linear models
- Stochastic volatility
- Generalised linear (mixed) models
- Generalised additive (mixed) models
- Spline smoothing
- Semiparametric regression
- Space-varying (semiparametric) regression models
- Disease mapping
- Log-Gaussian Cox-processes
- Model-based geostatistics (*)
- Spatio-temporal models

Examples

- Dynamic linear models
- Stochastic volatility
- Generalised linear (mixed) models
- Generalised additive (mixed) models
- Spline smoothing
- Semiparametric regression
- Space-varying (semiparametric) regression models
- Disease mapping
- Log-Gaussian Cox-processes
- Model-based geostatistics (*)
- Spatio-temporal models

Example: Disease mapping (BYM-model)

- Data $y_i \sim \text{Poisson}(E_i exp(\eta_i))$
- Log-relative risk
  $\eta_i = u_i + v_i + \boldsymbol{\beta}^T \mathbf{z}_i$
- Structured component $\mathbf{u}$
- Unstructured component $\mathbf{v}$
- Covariates $\mathbf{z}_i$
- Log-precisions $\log \kappa_u$ and $\log \kappa_v$



Characteristic features

- Large dimension of the latent Gaussian field: $10^2 - 10^5$
- A lot of conditional independence in the latent Gaussian field
- Few hyperparameters θ: $\dim(\theta)$ between 1 and 5
- Non-Gaussian data

Characteristic features

- Large dimension of the latent Gaussian field: $10^2 - 10^5$
- A lot of conditional independence in the latent Gaussian field
- Few hyperparameters θ: $\dim(\theta)$ between 1 and 5
- Non-Gaussian data

Characteristic features

- Large dimension of the latent Gaussian field: $10^2 - 10^5$
- A lot of conditional independence in the latent Gaussian field
- Few hyperparameters $\boldsymbol{\theta}$: $\dim(\boldsymbol{\theta})$ between 1 and 5
- Non-Gaussian data

Characteristic features

- Large dimension of the latent Gaussian field: $10^2 - 10^5$
- A lot of conditional independence in the latent Gaussian field
- Few hyperparameters $\boldsymbol{\theta}$: $\dim(\boldsymbol{\theta})$ between 1 and 5
- Non-Gaussian data

Main task

- Compute the posterior marginals for the latent field

$$\pi(x_i \mid \mathbf{y}), \qquad i = 1, \ldots, n$$

- Compute the posterior marginals for the hyperparameters

$$\pi(\theta_j \mid \mathbf{y}), \qquad j = 1, \ldots, \dim(\boldsymbol{\theta})$$

- Today's "standard" approach, is to make use of MCMC
- Main difficulties
  - CPU-time
  - Additive MC-errors

Main task

- Compute the posterior marginals for the latent field

$$\pi(x_i \mid \mathbf{y}), \qquad i = 1, \ldots, n$$

- Compute the posterior marginals for the hyperparameters

$$\pi(\theta_j \mid \mathbf{y}), \qquad j = 1, \ldots, \dim(\boldsymbol{\theta})$$

- Today's "standard" approach, is to make use of MCMC
- Main difficulties
  - CPU-time
  - Additive MC-errors

Main task

- Compute the posterior marginals for the latent field

$$\pi(x_i \mid \mathbf{y}), \qquad i = 1, \ldots, n$$

- Compute the posterior marginals for the hyperparameters

$$\pi(\theta_j \mid \mathbf{y}), \qquad j = 1, \ldots, \dim(\boldsymbol{\theta})$$

- Today's "standard" approach, is to make use of MCMC
- Main difficulties
  - CPU-time
  - Additive MC-errors

Main task

- Compute the posterior marginals for the latent field

$$\pi(x_i \mid \mathbf{y}), \qquad i = 1, \ldots, n$$

- Compute the posterior marginals for the hyperparameters

$$\pi(\theta_j \mid \mathbf{y}), \qquad j = 1, \ldots, \dim(\boldsymbol{\theta})$$

- Today's "standard" approach, is to make use of MCMC
- Main difficulties
  - CPU-time
  - Additive MC-errors

Integrated nested Laplace approximations (INLA)

- Utilise of the latent Gaussian field
  - Laplace approximations
- Utilise the conditional independence properties of the latent Gaussian field
  - Numerical algorithms for sparse matrices
- Utilise small dim(θ)
  - *Integrated Nested* Laplace approximations

Integrated nested Laplace approximations (INLA)

- Utilise of the latent Gaussian field
  - Laplace approximations
- Utilise the conditional independence properties of the latent Gaussian field
  - Numerical algorithms for sparse matrices
- Utilise small $\dim(\theta)$
  - *Integrated Nested* Laplace approximations

Integrated nested Laplace approximations (INLA)

- Utilise of the latent Gaussian field
  - Laplace approximations
- Utilise the conditional independence properties of the latent Gaussian field
  - Numerical algorithms for sparse matrices
- Utilise small dim(θ)
  - *Integrated Nested* Laplace approximations

Integrated nested Laplace approximations (INLA)

- Utilise of the latent Gaussian field
  - Laplace approximations
- Utilise the conditional independence properties of the latent Gaussian field
  - Numerical algorithms for sparse matrices
- Utilise small dim(θ)
  - *Integrated Nested* Laplace approximations

Integrated nested Laplace approximations (INLA)

- Utilise of the latent Gaussian field
  - Laplace approximations
- Utilise the conditional independence properties of the latent Gaussian field
  - Numerical algorithms for sparse matrices
- Utilise small dim($\boldsymbol{\theta}$)
  - *Integrated Nested* Laplace approximations

Integrated nested Laplace approximations (INLA)

- Utilise of the latent Gaussian field
  - Laplace approximations
- Utilise the conditional independence properties of the latent Gaussian field
  - Numerical algorithms for sparse matrices
- Utilise small $\dim(\boldsymbol{\theta})$
  - *Integrated Nested* Laplace approximations

Main results

- HUGE improvement in both speed and accuracy compared to MCMC alternatives
- Relative error
- Practically "exact" results[2]
- Extensions: Marginal likelihood, DIC, Cross-validation, ...

INLA enable us to treat Bayesian latent Gaussian models properly and bring these models from the research communities to the end-users

[2] Can construct counter-examples

Main results

- HUGE improvement in both speed and accuracy compared to MCMC alternatives
- Relative error
- Practically "exact" results[2]
- Extensions: Marginal likelihood, DIC, Cross-validation, ...

INLA enable us to treat Bayesian latent Gaussian models properly and bring these models from the research communities to the end-users

[2] Can construct counter-examples

Main results

- HUGE improvement in both speed and accuracy compared to MCMC alternatives
- Relative error
- Practically "exact" results[2]
- Extensions: Marginal likelihood, DIC, Cross-validation, ...

INLA enable us to treat Bayesian latent Gaussian models properly and bring these models from the research communities to the end-users

[2] Can construct counter-examples

Main results

- HUGE improvement in both speed and accuracy compared to MCMC alternatives
- Relative error
- Practically "exact" results[2]
- Extensions: Marginal likelihood, DIC, Cross-validation, ...

INLA enable us to treat Bayesian latent Gaussian models properly and bring these models from the research communities to the end-users

[2] Can construct counter-examples

Main results

- HUGE improvement in both speed and accuracy compared to MCMC alternatives
- Relative error
- Practically "exact" results[2]
- Extensions: Marginal likelihood, DIC, Cross-validation, ...

INLA enable us to treat Bayesian latent Gaussian models properly and bring these models from the research communities to the end-users

[2] Can construct counter-examples

Main ideas (I)

$$\pi(z) = \frac{\pi(x, z)}{\pi(x|z)} \qquad \text{leading to} \qquad \widetilde{\pi}(z) = \frac{\pi(x, z)}{\widetilde{\pi}(x|z)}\Big|_{\text{mode}(z)}$$

- When $\widetilde{\pi}(x|z)$ is the Gaussian-approximation, this is the Laplace-approximation
- Want $\pi(x|z)$ to be "almost Gaussian".

Main ideas (I)

$$\pi(z) = \frac{\pi(x, z)}{\pi(x|z)} \qquad \text{leading to} \qquad \widetilde{\pi}(z) = \frac{\pi(x, z)}{\widetilde{\pi}(x|z)}\Big|_{\text{mode}(z)}$$

- When $\widetilde{\pi}(x|z)$ is the Gaussian-approximation, this is the Laplace-approximation
- Want $\pi(x|z)$ to be "almost Gaussian".

Main ideas (I)

$$\pi(z) = \frac{\pi(x,z)}{\pi(x|z)} \qquad \text{leading to} \qquad \widetilde{\pi}(z) = \frac{\pi(x,z)}{\widetilde{\pi}(x|z)}\Big|_{\text{mode}(z)}$$

- When $\widetilde{\pi}(x|z)$ is the Gaussian-approximation, this is the Laplace-approximation
- Want $\pi(x|z)$ to be "almost Gaussian".

Main ideas (II)

Posterior

$$\pi(\mathbf{x}, \boldsymbol{\theta} \mid \mathbf{y}) \propto \pi(\boldsymbol{\theta})\, \pi(\mathbf{x} \mid \boldsymbol{\theta}) \prod_{i \in \mathcal{I}} \pi(y_i \mid x_i, \boldsymbol{\theta})$$

Do the integration wrt $\boldsymbol{\theta}$ numerically

Main ideas (II)

Posterior

$$\pi(\mathbf{x}, \boldsymbol{\theta} \mid \mathbf{y}) \propto \pi(\boldsymbol{\theta})\, \pi(\mathbf{x} \mid \boldsymbol{\theta}) \prod_{i \in \mathcal{I}} \pi(y_i \mid x_i, \boldsymbol{\theta})$$

Do the integration wrt $\boldsymbol{\theta}$ numerically

$$\pi(x_i \mid \mathbf{y}) = \int \pi(\boldsymbol{\theta} \mid \mathbf{y})\, \pi(x_i \mid \boldsymbol{\theta}, \mathbf{y})\, d\boldsymbol{\theta}$$

$$\pi(\theta_j \mid \mathbf{y}) = \int \pi(\boldsymbol{\theta} \mid \mathbf{y})\, d\boldsymbol{\theta}_{-j}$$

Main ideas (II)

Posterior

$$\pi(\mathbf{x}, \boldsymbol{\theta} \mid \mathbf{y}) \propto \pi(\boldsymbol{\theta})\, \pi(\mathbf{x} \mid \boldsymbol{\theta}) \prod_{i \in \mathcal{I}} \pi(y_i \mid x_i, \boldsymbol{\theta})$$

Do the integration wrt $\boldsymbol{\theta}$ numerically

$$\widetilde{\pi}(x_i \mid \mathbf{y}) = \int \widetilde{\pi}(\boldsymbol{\theta} \mid \mathbf{y})\, \widetilde{\pi}(x_i \mid \boldsymbol{\theta}, \mathbf{y})\, d\boldsymbol{\theta}$$

$$\widetilde{\pi}(\theta_j \mid \mathbf{y}) = \int \widetilde{\pi}(\boldsymbol{\theta} \mid \mathbf{y})\, d\boldsymbol{\theta}_{-j}$$

Remarks

1. Expect $\widetilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$ to be accurate, since
   - $\mathbf{x}|\boldsymbol{\theta}$ is *a priori* Gaussian
   - Likelihood models are 'well-behaved' so

$$\pi(\mathbf{x} \mid \boldsymbol{\theta}, \mathbf{y})$$

   is *almost* Gaussian.

2. There are no distributional assumptions on $\boldsymbol{\theta}|\mathbf{y}$
3. Similar remarks are valid to

$$\widetilde{\pi}(x_i \mid \boldsymbol{\theta}, \mathbf{y})$$

Remarks

1. Expect $\widetilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$ to be accurate, since
   - $\mathbf{x}|\boldsymbol{\theta}$ is *a priori* Gaussian
   - Likelihood models are 'well-behaved' so

$$\pi(\mathbf{x} \mid \boldsymbol{\theta}, \mathbf{y})$$

   is *almost* Gaussian.

2. There are no distributional assumptions on $\boldsymbol{\theta}|\mathbf{y}$
3. Similar remarks are valid to

$$\widetilde{\pi}(x_i \mid \boldsymbol{\theta}, \mathbf{y})$$

Remarks

1. Expect $\widetilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$ to be accurate, since
   - $\mathbf{x}|\boldsymbol{\theta}$ is *a priori* Gaussian
   - Likelihood models are 'well-behaved' so

     $$\pi(\mathbf{x} \mid \boldsymbol{\theta}, \mathbf{y})$$

     is *almost* Gaussian.
2. There are no distributional assumptions on $\boldsymbol{\theta}|\mathbf{y}$
3. Similar remarks are valid to

$$\widetilde{\pi}(x_i \mid \boldsymbol{\theta}, \mathbf{y})$$

Remarks

1. Expect $\widetilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$ to be accurate, since
   - $\mathbf{x}|\boldsymbol{\theta}$ is *a priori* Gaussian
   - Likelihood models are 'well-behaved' so

     $$\pi(\mathbf{x} \mid \boldsymbol{\theta}, \mathbf{y})$$

     is *almost* Gaussian.
2. There are no distributional assumptions on $\boldsymbol{\theta}|\mathbf{y}$
3. Similar remarks are valid to

$$\widetilde{\pi}(x_i \mid \boldsymbol{\theta}, \mathbf{y})$$

Remarks

1. Expect $\widetilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$ to be accurate, since
   - $\mathbf{x}|\boldsymbol{\theta}$ is *a priori* Gaussian
   - Likelihood models are 'well-behaved' so

     $$\pi(\mathbf{x} \mid \boldsymbol{\theta}, \mathbf{y})$$

     is *almost* Gaussian.
2. There are no distributional assumptions on $\boldsymbol{\theta}|\mathbf{y}$
3. Similar remarks are valid to

$$\widetilde{\pi}(x_i \mid \boldsymbol{\theta}, \mathbf{y})$$

Computational issues

- Our approach in its “raw” form is not computational feasible
- Main issue is the large dimension, n, of the latent field
- Various strategies/tricks are required for obtaining a practical good solution

Computational issues

- Our approach in its "raw" form is not computational feasible
- Main issue is the large dimension, n, of the latent field
- Various strategies/tricks are required for obtaining a practical good solution

Computational issues

- Our approach in its “raw” form is not computational feasible
- Main issue is the large dimension, n, of the latent field
- Various strategies/tricks are required for obtaining a practical good solution

Gaussian Markov random fields

Make use of the conditional independence properties in the latent field

$$x_i \perp x_j \mid \mathbf{x}_{-ij} \quad \Longleftrightarrow \quad Q_{ij} = 0$$

where $\mathbf{Q}$ is the precision matrix (inverse covariance)

Use numerical methods for sparse matrices!

Gaussian Markov random fields

Make use of the conditional independence properties in the latent field

$$x_i \perp x_j \mid \mathbf{x}_{-ij} \quad \Longleftrightarrow \quad Q_{ij} = 0$$

where $\mathbf{Q}$ is the precision matrix (inverse covariance)

Use numerical methods for sparse matrices!

Rank one updates

- Have computed the Cholesky factorisation

$$\text{Prec}(\mathbf{x}) = \mathbf{Q} = \mathbf{L}\mathbf{L}^T$$

- Want to compute the conditional mean and variances for

$$\mathbf{x}_{-i} \mid x_i$$

- No need to factorise $\text{Prec}(\mathbf{x}_{-i}|x_i)$; can compute the correction for conditioning on x_i.

Rank one updates

- Have computed the Cholesky factorisation

$$\text{Prec}(\mathbf{x}) = \mathbf{Q} = \mathbf{L}\mathbf{L}^T$$

- Want to compute the conditional mean and variances for

$$\mathbf{x}_{-i} \mid x_i$$

- No need to factorise $\text{Prec}(\mathbf{x}_{-i}|x_i)$; can compute the correction for conditioning on x_i.

Rank one updates

- Have computed the Cholesky factorisation

$$\text{Prec}(\mathbf{x}) = \mathbf{Q} = \mathbf{L}\mathbf{L}^T$$

- Want to compute the conditional mean and variances for

$$\mathbf{x}_{-i} \mid x_i$$

- No need to factorise $\text{Prec}(\mathbf{x}_{-i}|x_i)$; can compute the correction for conditioning on x_i.

Simplified Laplace Approximation

Expand the Laplace approximation of $\pi(x_i|\boldsymbol{\theta}, \mathbf{y})$:

$$\log \widetilde{\pi}(x_i|\boldsymbol{\theta}, \mathbf{y}) = -\frac{1}{2}x_i^2 + b_i x_i + \frac{1}{6}d_i \, x_i^3 + \cdots$$

Remarks

- Correct the Gaussian approximation for error in shift and skewness through b_i and d_i
- Fit a skew-Normal density

$$2\phi(x)\Phi(ax)$$

- Computational fast
- Sufficient accurate for most applications

Simplified Laplace Approximation

Expand the Laplace approximation of $\pi(x_i|\boldsymbol{\theta}, \mathbf{y})$:

$$\log \widetilde{\pi}(x_i|\boldsymbol{\theta}, \mathbf{y}) = -\frac{1}{2}x_i^2 + b_i x_i + \frac{1}{6}d_i\, x_i^3 + \cdots$$

Remarks

- Correct the Gaussian approximation for error in shift and skewness through b_i and d_i
- Fit a skew-Normal density

$$2\phi(x)\Phi(ax)$$

- Computational fast
- Sufficient accurate for most applications

Simplified Laplace Approximation

Expand the Laplace approximation of $\pi(x_i|\boldsymbol{\theta}, \mathbf{y})$:

$$\log \widetilde{\pi}(x_i|\boldsymbol{\theta}, \mathbf{y}) = -\frac{1}{2}x_i^2 + b_i x_i + \frac{1}{6} d_i\ x_i^3 + \cdots$$

Remarks

- Correct the Gaussian approximation for error in shift and skewness through b_i and d_i
- Fit a skew-Normal density

$$2\phi(x)\Phi(ax)$$

- Computational fast
- Sufficient accurate for most applications

Simplified Laplace Approximation

Expand the Laplace approximation of $\pi(x_i|\boldsymbol{\theta}, \mathbf{y})$:

$$\log \widetilde{\pi}(x_i|\boldsymbol{\theta}, \mathbf{y}) = -\frac{1}{2}x_i^2 + b_i x_i + \frac{1}{6}d_i\ x_i^3 + \cdots$$

Remarks

- Correct the Gaussian approximation for error in shift and skewness through b_i and d_i
- Fit a skew-Normal density

$$2\phi(x)\Phi(ax)$$

- Computational fast
- Sufficient accurate for most applications

Simplified Laplace Approximation

Expand the Laplace approximation of $\pi(x_i|\boldsymbol{\theta}, \mathbf{y})$:

$$\log \widetilde{\pi}(x_i|\boldsymbol{\theta}, \mathbf{y}) = -\frac{1}{2}x_i^2 + b_i x_i + \frac{1}{6}d_i\ x_i^3 + \cdots$$

Remarks

- Correct the Gaussian approximation for error in shift and skewness through b_i and d_i
- Fit a skew-Normal density

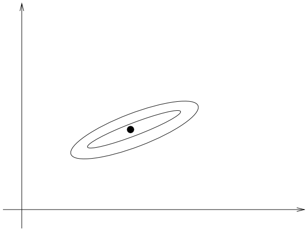
$$2\phi(x)\Phi(ax)$$

- Computational fast
- Sufficient accurate for most applications

The integrated nested Laplace approximation (INLA) I

Step I Explore $\widetilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$

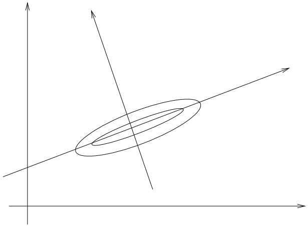
- Locate the mode
- Use the Hessian to construct new variables
- Grid-search



The integrated nested Laplace approximation (INLA) I

Step I Explore $\widetilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$

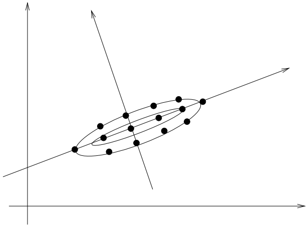
- Locate the mode
- Use the Hessian to construct new variables
- Grid-search



The integrated nested Laplace approximation (INLA) I

Step I Explore $\widetilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$

- Locate the mode
- Use the Hessian to construct new variables
- Grid-search



The integrated nested Laplace approximation (INLA) II

Step II For each $\boldsymbol{\theta}_j$

- For each i, compute the (simplified) Laplace approximation for x_i

The integrated nested Laplace approximation (INLA) III

Step III Sum out $\boldsymbol{\theta}_j$

- For each i, sum out $\boldsymbol{\theta}$

$$\widetilde{\pi}(x_i \mid \mathbf{y}) \propto \sum_j \widetilde{\pi}(x_i \mid \mathbf{y}, \boldsymbol{\theta}_j) \times \widetilde{\pi}(\boldsymbol{\theta}_j \mid \mathbf{y})$$

- Build a log-spline corrected Gaussian

$$\mathcal{N}(x_i;\ \mu_i, \sigma_i^2) \times \exp(\text{spline})$$

to represent $\widetilde{\pi}(x_i \mid \mathbf{y})$.

The integrated nested Laplace approximation (INLA) III

Step III Sum out $\boldsymbol{\theta}_j$

- For each i, sum out $\boldsymbol{\theta}$

$$\widetilde{\pi}(x_i \mid \mathbf{y}) \propto \sum_j \widetilde{\pi}(x_i \mid \mathbf{y}, \boldsymbol{\theta}_j) \times \widetilde{\pi}(\boldsymbol{\theta}_j \mid \mathbf{y})$$

- Build a log-spline corrected Gaussian

$$\mathcal{N}(x_i;\ \mu_i, \sigma_i^2) \times \exp(\text{spline})$$

to represent $\widetilde{\pi}(x_i \mid \mathbf{y})$.

How can we assess the errors in the approximations?

Important, but asymptotic arguments are difficult:

$$\dim(\mathbf{y}) = \mathcal{O}(n) \quad \text{and} \quad \dim(\mathbf{x}) = \mathcal{O}(n)$$

Errors in the approximations of $\pi(x_i|\mathbf{y})$

Compare a sequence of improved approximations

1. Gaussian approximation
2. Simplified Laplace
3. Laplace

Compute the full Laplace-approximation for $\pi(x_i|\mathbf{y}, \theta_j)$ only if the Gaussian and the Simplified Laplace approximation disagree.

Errors in the approximations of $\pi(x_i|\mathbf{y})$

Compare a sequence of improved approximations

1. Gaussian approximation
2. Simplified Laplace
3. Laplace

Compute the full Laplace-approximation for $\pi(x_i|\mathbf{y}, \boldsymbol{\theta}_j)$ only if the Gaussian and the Simplified Laplace approximation disagree.

Overall check: Equivalent number of replicates

Tool 3: Estimate the "effective" number of parameters

- From the *Deviance Information Criteria*:

$$p_D(\boldsymbol{\theta}) \approx n - \text{trace}\left(\mathbf{Q}_{\text{prior}}(\boldsymbol{\theta})\, \mathbf{Q}_{\text{post.}}(\boldsymbol{\theta})^{-1}\right)$$

- Compare with the number of observations:

$$\#\text{observations}/p_D(\boldsymbol{\theta})$$

  high ratio is good
- Theoretical justification

Overall check: Equivalent number of replicates

Tool 3: Estimate the "effective" number of parameters

- From the *Deviance Information Criteria*:

$$p_{\mathrm{D}}(\boldsymbol{\theta}) \approx n - \operatorname{trace}\left(\mathbf{Q}_{\text{prior}}(\boldsymbol{\theta})\ \mathbf{Q}_{\text{post.}}(\boldsymbol{\theta})^{-1}\right)$$

- Compare with the number of observations:

$$\#\text{observations}/p_{\mathrm{D}}(\boldsymbol{\theta})$$

  high ratio is good

- Theoretical justification

Overall check: Equivalent number of replicates

Tool 3: Estimate the "effective" number of parameters

- From the *Deviance Information Criteria*:

$$p_{\mathrm{D}}(\boldsymbol{\theta}) \approx n - \mathrm{trace}\left(\mathbf{Q}_{\text{prior}}(\boldsymbol{\theta})\ \mathbf{Q}_{\text{post.}}(\boldsymbol{\theta})^{-1}\right)$$

- Compare with the number of observations:

$$\#\text{observations}/p_{\mathrm{D}}(\boldsymbol{\theta})$$

  high ratio is good

- Theoretical justification

Examples

- 10:30-11:30: Andrea Riebler: Performance of INLA analysing bivariate meta-regression and age-period-cohort models.
- 12:30-13:30: Birgit Schrödle: Spatio-temporal disease mapping using INLA.
- 13:45-14:45: Virgilio Gómez-Rubio: Approximate Bayesian Inference for Small Area Estimation
- 15:00-16:00: Lea Fortunato: About the *Rapid Inquiry Facility*, and spatial analyses with WinBUGS and INLA.
- 09:00-10:00: Rupali Akerkar: Approximate Bayesian Inference for Survival models.
- 10:15-11:30: Ingelin Steinsland & Anna Marie Holand: Animal Model and INLA.

Marginal likelihood

Marginal likelihood is the normalising constant for $\pi(\boldsymbol{\theta}|\mathbf{y})$

Deviance Information Criteria

$$D(\mathbf{x};\boldsymbol{\theta}) = -2\sum_i \log(y_i \mid x_i, \boldsymbol{\theta})$$

$$DIC = 2 \times \text{Mean}\left(D(\mathbf{x};\boldsymbol{\theta})\right) - D(\text{Mean}(\mathbf{x});\boldsymbol{\theta}^*)$$

Cross-validation

- Based on

$$\pi(x_i|\mathbf{y}_{-i}, \boldsymbol{\theta}) \propto \frac{\pi(x_i|\mathbf{y}, \boldsymbol{\theta})}{\pi(y_i|x_i, \boldsymbol{\theta})}$$

  we can compute

$$\pi(y_i \mid \mathbf{y}_{-i})$$

- Similar with $\pi(\boldsymbol{\theta}|\mathbf{y}_{-i})$
- Keep the integration points $\{\boldsymbol{\theta}_j\}$ fixed.
- Detect "surprising" observations:

$$\text{Prob}(y_i^{\text{new}} \leq y_i \mid \mathbf{y}_{-i})$$

Cross-validation

- Based on

$$\pi(x_i|\mathbf{y}_{-i}, \boldsymbol{\theta}) \propto \frac{\pi(x_i|\mathbf{y}, \boldsymbol{\theta})}{\pi(y_i|x_i, \boldsymbol{\theta})}$$

  we can compute

$$\pi(y_i \mid \mathbf{y}_{-i})$$

- Similar with $\pi(\boldsymbol{\theta}|\mathbf{y}_{-i})$
- Keep the integration points $\{\boldsymbol{\theta}_j\}$ fixed.
- Detect "surprising" observations:

$$\text{Prob}(y_i^{\text{new}} \leq y_i \mid \mathbf{y}_{-i})$$

Cross-validation

- Based on

$$\pi(x_i|\mathbf{y}_{-i}, \boldsymbol{\theta}) \propto \frac{\pi(x_i|\mathbf{y}, \boldsymbol{\theta})}{\pi(y_i|x_i, \boldsymbol{\theta})}$$

  we can compute

$$\pi(y_i \mid \mathbf{y}_{-i})$$

- Similar with $\pi(\boldsymbol{\theta}|\mathbf{y}_{-i})$
- Keep the integration points $\{\boldsymbol{\theta}_j\}$ fixed.
- Detect "surprising" observations:

$$\text{Prob}(y_i^{\text{new}} \leq y_i \mid \mathbf{y}_{-i})$$

Cross-validation

- Based on
$$\pi(x_i|\mathbf{y}_{-i},\boldsymbol{\theta}) \propto \frac{\pi(x_i|\mathbf{y},\boldsymbol{\theta})}{\pi(y_i|x_i,\boldsymbol{\theta})}$$
we can compute
$$\pi(y_i \mid \mathbf{y}_{-i})$$
- Similar with $\pi(\boldsymbol{\theta}|\mathbf{y}_{-i})$
- Keep the integration points $\{\boldsymbol{\theta}_j\}$ fixed.
- Detect "surprising" observations:
$$\text{Prob}(y_i^{\text{new}} \leq y_i \mid \mathbf{y}_{-i})$$

"Recent" developments in (R-)INLA

- Improving the code: speedups and improved parallel performance
- Improving the R-interface
- Extending the building-blocks: prior-models and likelihood-models

Lot of things still todo...

- Documentation
- Worked out examples
- Webpage
- +++

Improving the code

The two largest changes the last year

- (summer 2008) Change the way `inla` works in a multi-core environment.
  Especially important for more than "dual-core"
- (xmas 2008) Implement a more efficient rank-one update formula (Thanks Birgit!).
  Especially important for models with many constraints, but gave a huge speedup also for models with a single constraint.

Improving the code

The two largest changes the last year

- (summer 2008) Change the way `inla` works in a multi-core environment.
  Especially important for more than "dual-core"
- (xmas 2008) Implement a more efficient rank-one update formula (Thanks Birgit!).
  Especially important for models with many constraints, but gave a huge speedup also for models with a single constraint.

inla & *multi-core*

- Gradient and Hessian computations are done in parallel

$$\frac{\partial}{\partial\theta_i}\widetilde{\pi}(\boldsymbol{\theta} \mid \mathbf{y}) \quad \text{and} \quad \frac{\partial^2}{\partial\theta_i\partial\theta_j}\widetilde{\pi}(\boldsymbol{\theta} \mid \mathbf{y})$$

- All computations of

$$\widetilde{\pi}(x_i \mid \boldsymbol{\theta}_j, \mathbf{y})$$

is now done in parallel wrt j, and not i as before.

Multi-core example

inla-example 7: CANCER-INCIDENCE, $n = 7138$, $\dim(\boldsymbol{\theta}) = 3$

Number of cores	Time used
1	20.0s
2	13.3s
4	9.1s
6	8.6s
8	8.3s

- Linear algebra (Cholesky/Solve/Inverse): $\approx 50\%$
- Administration: $\approx 50\%$

Recent developments in (R-)INLA

Improved R-interface

- `inla`-binary is now bundled with the `R`-package.
- Make use of `model.matrix()` in `R`:

  *formula ~ ... + x*z*

  will expand as

  formula ~ ... + x + z + x:z

- Allow for

  formula ~ .... + offset(a) + offset(b)

  defining a+b to be fixed offset in `formula`

Larger change for survival-models (more on this tomorrow...)

Recent developments in (R-)INLA

Improved R-interface

- `inla`-binary is now bundled with the `R`-package.
- Make use of `model.matrix()` in `R`:

  ```
  formula ~ ...  + x*z
  ```

  will expand as

  ```
  formula ~ ...  + x + z + x:z
  ```

- Allow for

  ```
  formula ~ ....  + offset(a) + offset(b)
  ```

  defining a+b to be fixed offset in `formula`

Larger change for survival-models (more on this tomorrow...)

Recent developments in (R-)INLA

Improved R-interface

- inla-binary is now bundled with the R-package.
- Make use of model.matrix() in R:

```
formula ~ ...  + x*z
```

  will expand as

```
formula ~ ...  + x + z + x:z
```

- Allow for

```
formula ~ ....  + offset(a) + offset(b)
```

  defining a+b to be fixed offset in formula

Larger change for survival-models (more on this tomorrow...)

Recent developments in (R-)INLA

Improved R-interface

- inla-binary is now bundled with the R-package.
- Make use of model.matrix() in R:

  *formula ~ ... + x*z*

  will expand as

  formula ~ ... + x + z + x:z

- Allow for

  formula ~ .... + offset(a) + offset(b)

  defining a+b to be fixed offset in formula

Larger change for survival-models (more on this tomorrow...)

Building blocks

```
> names(inla.models()$models)
 [1] "iid"                 "rw1"                 "rw2"
 [4] "crw2"                "seasonal"            "besag"
 [7] "ar1"                 "generic"             "2diid"
[10] "2diidwishart"        "2diidwishartpart0"   "2diidwishartpart1"
[13] "3diidwishartpart0"   "3diidwishartpart1"   "3diidwishartpart2"
[16] "z"                   "rw2d"                "matern2d"
```

The "z"-model

The z-models is

$$\boldsymbol{\eta} = \ldots + \mathbf{Z}z + \ldots.$$

where $\mathbf{Z}$ is a $n \times k$ matrix and $\mathbf{z} \sim \mathcal{N}_k(0, \tau \mathbf{I})$.

Likelihood models

```
> names(inla.models()$lmodels)
 [1] "poisson"                  "binomial"
 [3] "exponential"              "piecewise.constant"
 [5] "piecewise.linear"         "gaussian"
 [7] "laplace"                  "weibull"
 [9] "weibullcure"              "stochvol_t"
[11] "zeroinflated_poisson_0"   "zeroinflated_poisson_1"
[13] "zeroinflated_binomial_0" "zeroinflated_binomial_1"
[15] "T"                        "stochvol.nig"
```

Zero-inflated models

The (type 0) likelihood is defined as

$$\text{Prob}(y \mid \ldots) = p \times 1_{[y=0]} + (1-p) \times \text{Poission}(y \mid y > 0)$$

The (type 1) likelihood is defined as

$$\text{Prob}(y \mid \ldots) = p \times 1_{[y=0]} + (1-p) \times \text{Poission}(y)$$

[Birgit: Mixture over p is still on the list... sorry.]

The TODO-list

- Better support for spatial (GMRF) models. More on this tomorrow (Finn.L).
- Alternative spline-models, B-splines? (Thomas.K ?)
- Simultaneous credibility intervals
- Documentation
- Worked out examples
- Webpage
- +++

Summary (I)

- Latent Gaussian models unifies many models into the same framework: unified approach towards Bayesian inference
  - Use the same computer code
  - Near optimal numerical algorithms for the sparse matrices
  - Achieve nice speed-ups in a multi-core environment
  - Practically "exact" results
- Prototype implementation
  - `inla-program`: Inference for structured additive regression models
  - GAM-like `R`-interface to the `inla-program`
  - Available for Linux/Mac/Windows
  - Open source

Summary (I)

- Latent Gaussian models unifies many models into the same framework: unified approach towards Bayesian inference
  - Use the same computer code
  - Near optimal numerical algorithms for the sparse matrices
  - Achieve nice speed-ups in a multi-core environment
  - Practically “exact” results
- Prototype implementation
  - `inla-program`: Inference for structured additive regression models
  - `GAM`-like `R`-interface to the `inla`-program
  - Available for Linux/Mac/Windows
  - Open source

Summary (I)

- Latent Gaussian models unifies many models into the same framework: unified approach towards Bayesian inference
  - Use the same computer code
  - Near optimal numerical algorithms for the sparse matrices
  - Achieve nice speed-ups in a multi-core environment
  - Practically "exact" results
- Prototype implementation
  - `inla-program`: Inference for structured additive regression models
  - GAM-like `R`-interface to the `inla-program`
  - Available for Linux/Mac/Windows
  - Open source

Summary (I)

- Latent Gaussian models unifies many models into the same framework: unified approach towards Bayesian inference
  - Use the same computer code
  - Near optimal numerical algorithms for the sparse matrices
  - Achieve nice speed-ups in a multi-core environment
  - Practically "exact" results
- Prototype implementation
  - `inla-program`: Inference for structured additive regression models
  - GAM-like R-interface to the `inla-program`
  - Available for Linux/Mac/Windows
  - Open source

Summary (I)

- Latent Gaussian models unifies many models into the same framework: unified approach towards Bayesian inference
  - Use the same computer code
  - Near optimal numerical algorithms for the sparse matrices
  - Achieve nice speed-ups in a multi-core environment
  - Practically "exact" results
- Prototype implementation
  - `inla-program`: Inference for structured additive regression models
  - GAM-like `R`-interface to the `inla-program`
  - Available for Linux/Mac/Windows
  - Open source

Summary (I)

- Latent Gaussian models unifies many models into the same framework: unified approach towards Bayesian inference
  - Use the same computer code
  - Near optimal numerical algorithms for the sparse matrices
  - Achieve nice speed-ups in a multi-core environment
  - Practically "exact" results
- Prototype implementation
  - `inla`-program: Inference for structured additive regression models
  - `GAM`-like `R`-interface to the `inla`-program
  - Available for Linux/Mac/Windows
  - Open source

Summary (I)

- Latent Gaussian models unifies many models into the same framework: unified approach towards Bayesian inference
  - Use the same computer code
  - Near optimal numerical algorithms for the sparse matrices
  - Achieve nice speed-ups in a multi-core environment
  - Practically "exact" results
- Prototype implementation
  - `inla`-program: Inference for structured additive regression models
  - GAM-like `R`-interface to the `inla`-program
  - Available for Linux/Mac/Windows
  - Open source

Summary (I)

- Latent Gaussian models unifies many models into the same framework: unified approach towards Bayesian inference
  - Use the same computer code
  - Near optimal numerical algorithms for the sparse matrices
  - Achieve nice speed-ups in a multi-core environment
  - Practically "exact" results
- Prototype implementation
  - `inla`-program: Inference for structured additive regression models
  - `GAM`-like `R`-interface to the `inla`-program
  - Available for Linux/Mac/Windows
  - Open source

Summary (I)

- Latent Gaussian models unifies many models into the same framework: unified approach towards Bayesian inference
  - Use the same computer code
  - Near optimal numerical algorithms for the sparse matrices
  - Achieve nice speed-ups in a multi-core environment
  - Practically "exact" results
- Prototype implementation
  - `inla`-program: Inference for structured additive regression models
  - `GAM`-like `R`-interface to the `inla`-program
  - Available for Linux/Mac/Windows
  - Open source

Summary (I)

- Latent Gaussian models unifies many models into the same framework: unified approach towards Bayesian inference
  - Use the same computer code
  - Near optimal numerical algorithms for the sparse matrices
  - Achieve nice speed-ups in a multi-core environment
  - Practically "exact" results
- Prototype implementation
  - `inla`-program: Inference for structured additive regression models
  - `GAM`-like `R`-interface to the `inla`-program
  - Available for Linux/Mac/Windows
  - Open source

Summary (II)

It is our view that the prospects of this work are more important than this work itself:

- Make latent Gaussian models, more applicable, useful and appealing for the end user
- Allow us to use latent Gaussian models as baseline models, even in cases where more complex models are more appropriate

Summary (II)

It is our view that the prospects of this work are more important than this work itself:

- Make latent Gaussian models, more applicable, useful and appealing for the end user
- Allow us to use latent Gaussian models as baseline models, even in cases where more complex models are more appropriate

Summary (II)

It is our view that the prospects of this work are more important than this work itself:

- Make latent Gaussian models, more applicable, useful and appealing for the end user
- Allow us to use latent Gaussian models as baseline models, even in cases where more complex models are more appropriate