Generalised Additive (Mixed) Models
I refuse to use a soft ‘G’

Daniel Simpson

Department of Mathematical Sciences
University of Bath
Outline

Background

Latent Gaussian model

Example: Continuous vs Discrete

Priors on functions
Two main paradigms for statistical analysis

- Let \( y \) denote a set of observations, distributed according to a probability model \( \pi(y; \theta) \).
- Based on the observations, we want to estimate \( \theta \).

The classical approach:
\( \theta \) denotes **parameters** (unknown fixed numbers), estimated for example by maximum likelihood.

The Bayesian approach:
\( \theta \) denotes **random variables**, assigned a prior \( \pi(\theta) \). Estimate \( \theta \) based on the posterior:

\[
\pi(\theta | y) = \frac{\pi(y | \theta) \pi(\theta)}{\pi(y)} \propto \pi(y | \theta) \pi(\theta).
\]
Two main paradigms for statistical analysis

- Let \( y \) denote a set of observations, distributed according to a probability model \( \pi(y; \theta) \).
- Based on the observations, we want to estimate \( \theta \).

**The classical approach:**
\( \theta \) denotes parameters (unknown fixed numbers), estimated for example by maximum likelihood.

**The Bayesian approach:**
\( \theta \) denotes random variables, assigned a prior \( \pi(\theta) \). Estimate \( \theta \) based on the posterior:

\[
\pi(\theta | y) = \frac{\pi(y | \theta)\pi(\theta)}{\pi(y)} \propto \pi(y | \theta)\pi(\theta).
\]
Example (Ski flying records)

Assume a simple linear regression model with Gaussian observations \( y = (y_1, \ldots, y_n) \), where

\[
\mathbb{E}(y_i) = \alpha + \beta x_i, \quad \text{Var}(y_i) = \tau^{-1}, \quad i = 1, \ldots, n
\]

World records in ski jumping, 1961 - 2011
The Bayesian approach

Assign priors to the parameters $\alpha$, $\beta$ and $\tau$ and calculate posteriors:

- **PostDens [(Intercept)]**
  - Mean = 137.354
  - SD = 1.508

- **PostDens [x]**
  - Mean = 2.14
  - SD = 0.054

- **PostDens [Precision for the Gaussian observations]**
  - Mean = 137.354
  - SD = 1.508

- **PostDens [x]**
  - Mean = 2.14
  - SD = 0.054
Real-world datasets are usually much more complicated!

Using a Bayesian framework:

- Build (hierarchical) models to account for potentially complicated dependency structures in the data.
- Attribute uncertainty to model parameters and latent variables using priors.

Two main challenges:

1. Need computationally efficient methods to calculate posteriors.
2. Select priors in a sensible way.
Real-world datasets are usually much more complicated!

Using a Bayesian framework:

▶ Build (hierarchical) models to account for potentially complicated dependency structures in the data.

▶ Attribute uncertainty to model parameters and latent variables using priors.

Two main challenges:

1. Need computationally efficient methods to calculate posteriors.

2. Select priors in a sensible way.
Background

Latent Gaussian model
  Computational framework and approximations
  Semiparametric regression

Example: Continuous vs Discrete

Priors on functions
What is a latent Gaussian model?

Classical multiple linear regression model

The mean $\mu$ of an $n$-dimensional observational vector $y$ is given by

$$\mu_i = E(Y_i) = \alpha + \sum_{j=1}^{n_\beta} \beta_j z_{ji}, \quad i = 1, \ldots, n$$

where

$\alpha$ : Intercept

$\beta$ : Linear effects of covariates $z$
Generalized linear model (GLM)

The mean $\mu$ is linked to the linear predictor $\eta_i$:

$$
\eta_i = g(\mu_i) = \alpha + \sum_{j=1}^{n_\beta} \beta_j z_{ji}, \quad i = 1, \ldots, n
$$

where $g(.)$ is a link function and

$\alpha$ : Intercept

$\beta$ : Linear effects of covariates $z$
Account for non-linear effects of covariates

Generalized additive model (GAM)

The mean $\mu$ is linked to the linear predictor $\eta_i$:

$$\eta_i = g(\mu_i) = \alpha + \sum_{k=1}^{n_f} f_k(c_{ki}), \quad i = 1, \ldots, n$$

where $g(.)$ is a link function and

- $\alpha$ : Intercept
- $\{f_k(\cdot)\}$ : Non-linear smooth effects of covariates $c_k$
Structured additive regression models

GLM/GAM/GLMM/GAMM+++  
The mean $\mu$ is linked to the linear predictor $\eta_i$:

$$
\eta_i = g(\mu_i) = \alpha + \sum_{j=1}^{n_\beta} \beta_j z_{ji} + \sum_{k=1}^{n_f} f_k(c_{ki}) + \epsilon_i, \quad i = 1, \ldots, n
$$

where $g(.)$ is a link function and

- $\alpha$: Intercept
- $\beta$: Linear effects of covariates $z$
- $\{f_k(\cdot)\}$: Non-linear smooth effects of covariates $c_k$
- $\epsilon$: iid random effects
Latent Gaussian models

- Collect all parameters (random variables) in the linear predictor in a latent field

\[ \mathbf{x} = \{ \alpha, \beta, \{ f_k(\cdot) \}, \eta \}. \]

- A latent Gaussian model is obtained by assigning Gaussian priors to all elements of \( \mathbf{x} \).

- Very flexible due to many different forms of the unknown functions \( \{ f_k(\cdot) \} \):
  - Include temporally and/or spatially indexed covariates.

- Hyperparameters account for variability and length/strength of dependence.
Latent Gaussian models

- Collect all parameters (random variables) in the linear predictor in a latent field

\[ x = \{ \alpha, \beta, \{ f_k(\cdot) \}, \eta \}. \]

- A latent Gaussian model is obtained by assigning Gaussian priors to all elements of \( x \).

- Very flexible due to many different forms of the unknown functions \( \{ f_k(\cdot) \} \):
  - Include temporally and/or spatially indexed covariates.

- Hyperparameters account for variability and length/strength of dependence.
Latent Gaussian models

- Collect all parameters (random variables) in the linear predictor in a latent field

\[
x = \{\alpha, \beta, \{f_k(\cdot)\}, \eta\}.
\]

- A latent Gaussian model is obtained by assigning Gaussian priors to all elements of \(x\).

- Very flexible due to many different forms of the unknown functions \(\{f_k(\cdot)\}:

  - Include temporally and/or spatially indexed covariates.

- Hyperparameters account for variability and length/strength of dependence.
Some examples of latent Gaussian models

- Generalized linear and additive (mixed) models
- Semiparametric regression
- Disease mapping
- Survival analysis
- Log-Gaussian Cox-processes
- Geostatistical models
- Spatial and spatio-temporal models
- Stochastic volatility
- Dynamic linear models
- State-space models

+++
Unified framework: A three-stage hierarchical model

1. Observations: $y$

2. Latent field: $x$

3. Hyperparameters: $\theta$
Unified framework: A three-stage hierarchical model

1. Observations: $y$
   Assumed conditionally independent given $x$ and $\theta_1$:

2. Latent field: $x$
   Assumed to be a GMRF with a sparse precision matrix $Q(\theta_2)$:

3. Hyperparameters: $\theta = (\theta_1, \theta_2)$
   Precision parameters of the Gaussian priors:
Unified framework: A three-stage hierarchical model

1. Observations: $y$
   Assumed conditionally independent given $x$ and $\theta_1$:
   \[
   y \mid x, \theta_1 \sim \prod_{i=1}^{n} \pi(y_i \mid x_i, \theta_1).
   \]

2. Latent field: $x$
   Assumed to be a GMRF with a sparse precision matrix $Q(\theta_2)$:
   \[
   x \mid \theta_2 \sim \mathcal{N}\left(\mu(\theta_2), Q^{-1}(\theta_2)\right).
   \]

3. Hyperparameters: $\theta = (\theta_1, \theta_2)$
   Precision parameters of the Gaussian priors:
   \[
   \theta \sim \pi(\theta).
   \]
Model summary

The joint posterior for the latent field and hyperparameters:

\[ \pi(x, \theta \mid y) \propto \pi(y \mid x, \theta) \pi(x, \theta) \]
\[ \propto n \prod_{i=1}^{n} \pi(y_i \mid x_i, \theta) \pi(x \mid \theta) \pi(\theta) \]

Remarks:

- \( m = \text{dim}(\theta) \) is often quite small, like \( m \leq 6 \).
- \( n = \text{dim}(x) \) is often large, typically \( n = 10^2 - 10^6 \).
Target densities are given as high-dimensional integrals

We want to estimate:

1. The marginals of all components of the latent field:

\[
\pi(x_i \mid y) = \int \int \pi(x, \theta \mid y) \, dx_{-i} \, d\theta = \int \pi(x_i \mid \theta, y) \pi(\theta \mid y) \, d\theta, \quad i = 1, \ldots, n.
\]

2. The marginals of all the hyperparameters:

\[
\pi(\theta_j \mid y) = \int \int \pi(x, \theta \mid y) \, dx \, d\theta_{-j} = \int \pi(\theta \mid y) \, d\theta_{-j}, \quad j = 1, \ldots m.
\]
Target densities are given as high-dimensional integrals

We want to estimate:

- The marginals of all components of the latent field:

\[
\pi(x_i \mid y) = \int \int \pi(x, \theta \mid y) dx_{-i} d\theta
= \int \pi(x_i \mid \theta, y) \pi(\theta \mid y) d\theta, \quad i = 1, \ldots, n.
\]

- The marginals of all the hyperparameters:

\[
\pi(\theta_j \mid y) = \int \int \pi(x, \theta \mid y) dx d\theta_{-j}
= \int \pi(\theta \mid y) d\theta_{-j}, \quad j = 1, \ldots m.
\]
Example: Logistic regression, $2 \times 2$ factorial design

Consider the proportion of seeds that germinates on each of 21 plates. We have two seed types ($x_1$) and two root extracts ($x_2$).

```r
> data(Seeds)
> head(Seeds)
    r  n x1 x2 plate
1 10 39 0 0   1
2 23 62 0 0   2
3 23 81 0 0   3
4 26 51 0 0   4
5 17 39 0 0   5
6  5  6 0 1   6
```
Summary data set

Number of seeds that germinated in each group:

<table>
<thead>
<tr>
<th>Seed types</th>
<th>$x_1 = 0$</th>
<th>$x_1 = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_2 = 0$</td>
<td>99/272</td>
<td>49/123</td>
</tr>
<tr>
<td>Root extract</td>
<td>201/295</td>
<td>75/141</td>
</tr>
</tbody>
</table>
Statistical model

- Assume that the number of seeds that germinate on plate $i$ is binomial

$$r_i \sim \text{Binomial}(n_i, p_i), \quad i = 1, \ldots, 21,$$

- Logistic regression model:

$$\text{logit}(p_i) = \log \left( \frac{p_i}{1 - p_i} \right) = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{1i} x_{2i} + \epsilon_i$$

where $\epsilon_i \sim N(0, \sigma^2_\epsilon)$ are iid.

**Aim:**

Estimate the main effects, $\beta_1$ and $\beta_2$ and a possible interaction effect $\beta_3$. 
Statistical model

- Assume that the number of seeds that germinate on plate $i$ is binomial

\[ r_i \sim \text{Binomial}(n_i, p_i), \quad i = 1, \ldots, 21, \]

- Logistic regression model:

\[
\text{logit}(p_i) = \log \left( \frac{p_i}{1 - p_i} \right) = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{1i} x_{2i} + \epsilon_i
\]

where $\epsilon_i \sim N(0, \sigma^2_\epsilon)$ are iid.

Aim:
Estimate the main effects, $\beta_1$ and $\beta_2$ and a possible interaction effect $\beta_3$. 
> formula = r ~ x1 + x2 + x1*x2 + f(plate, model="iid")
> result = inla(formula, data = Seeds,
>               family = "binomial",
>               Ntrials = n,
>               control.predictor =
>               list(compute = T, link=1),
>               control.compute = list(dic = T))

Default priors

Default prior for fixed effects is

\[ \beta \sim N(0, 1000). \]

Change using the `control.fixed` argument in the `inla`-call.
> summary(result)

Call:
"inla(formula = formula, family = "binomial", data = Seeds, Ntrials = n)"

Time used:

<table>
<thead>
<tr>
<th></th>
<th>Pre-processing</th>
<th>Running inla</th>
<th>Post-processing</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.1354</td>
<td>0.0911</td>
<td>0.0347</td>
<td>0.2613</td>
</tr>
</tbody>
</table>

Fixed effects:

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>sd</th>
<th>0.025quant</th>
<th>0.5quant</th>
<th>0.975quant</th>
<th>kld</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>-0.5581</td>
<td>0.1261</td>
<td>-0.8076</td>
<td>-0.5573</td>
<td>-0.3130</td>
<td>0e+00</td>
</tr>
<tr>
<td>x1</td>
<td>0.1461</td>
<td>0.2233</td>
<td>-0.2933</td>
<td>0.1467</td>
<td>0.5823</td>
<td>0e+00</td>
</tr>
<tr>
<td>x2</td>
<td>1.3206</td>
<td>0.1776</td>
<td>0.9748</td>
<td>1.3197</td>
<td>1.6716</td>
<td>1e-04</td>
</tr>
<tr>
<td>x1:x2</td>
<td>-0.7793</td>
<td>0.3066</td>
<td>-1.3799</td>
<td>-0.7796</td>
<td>-0.1774</td>
<td>0e+00</td>
</tr>
</tbody>
</table>

Random effects:

Name   Model
plate  IID model

Model hyperparameters:

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>sd</th>
<th>0.025quant</th>
<th>0.5quant</th>
<th>0.975quant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision for plate</td>
<td>18413.03</td>
<td>18280.63</td>
<td>1217.90</td>
<td>13003.76</td>
<td>66486.29</td>
</tr>
</tbody>
</table>

Expected number of effective parameters (std dev): 4.014(0.0114)
Number of equivalent replicates: 5.231
Estimated germination probabilities

> result$summary.fitted.values$mean
More in the practicals ...

```r
> plot(result)
> result$summary.fixed
> result$summary.random
> result$summay.linear.predictor
> result$summay.fitted.values
> result$marginals.fixed
> result$marginals.hyperpar
> result$marginals.linear.predictor
> result$marginals.fitted.values
```
Example: Semiparametric regression

Example (Annual global temperature anomalies)
Estimating a smooth non-linear trend

- Assume the model

\[ y_i = \alpha + f(x_i) + \epsilon_i, \quad i = 1, \ldots, n, \]

where the errors are iid, \( \epsilon_i \sim N(0, \sigma^2_\epsilon) \).

- Want to estimate the true underlying curve \( f(\cdot) \).
Define formula and run model

```r
> formula = y ~ f(x, model = "rw2", hyper = ...)  
> result = inla(formula, data = data.frame(y, x))
```

The default prior for the hyperparameter of `rw2`:

```r
hyper = list(prec =  
  list(prior = "loggamma",  
    param = c(1, 0.00005)))
```
> summary(result)
> plot(result)

The mean effect of \( x \):

> result$summary.random$x$mean

Note that this effect is constrained to sum to 0.

Resulting fitted curve

> result$summary.fitted.values$mean
Estimated fit using the default prior

Example (Annual global temperature anomalies)
Estimated fit using **R-INLA** compared with **smooth.spline**

**Example (Annual global temperature anomalies)**
Using different priors for the precision

Example (Annual global temperature anomalies)
“Hepatitis Z” in Ontario

Somebody gave us all of the cases of a disease throughout Ontario.

- Analyse risk factors
- We have counts in each postcode
- Try something simple first...

(data: diseasesmapping toolbox in R)
Oops - doesn’t work!

Fit a model with age (in groups) and sex
- It doesn’t fit!
- Lots of under-prediction
- Not enough zeros (56 vs 143)
We need to do something else? Zero-inflation?? Over-dispersion????
Not just yet...

Let’s actually look at the data!

- Nearby counts are similar
- Big counts are in the cities
- Maybe each observation is *not* independent?
- Maybe the *risk* is not independent?
Outline

Background

Latent Gaussian model

Example: Continuous vs Discrete

Priors on functions
Leukaemia survival data (Henderson et al, 2002, JASA), 1043 cases.

*Fig. 1.* Leukaemia survival data: districts of Northwest England and locations of the observations.
Survival models are different for many models in statistics. There are two types of observations: an event (death) or we stop measuring (censoring).

Rather than directly modelling the hazard (instantaneous risk)

\[
\begin{align*}
  h(y) \, dy &= \text{Prob}(y \leq Y < y + \, dy \mid Y > y) \\
  h(y) &= \frac{f(t)}{S(t)}
\end{align*}
\]
Cox proportional hazards model

Write the hazard function for each patient as:

\[ h(y_i|w_i, x_i) = h_0(y_i) \ w_i \ \exp(c_i^T \beta) \ \exp(x(s_i)); \ i = 1, \ldots, 1043 \]

where

- \( h_0(\cdot) \) is the baseline hazard function
- \( w_i \) is the log-Normal frailty effect associated with patient \( i \)
- \( c_i \) is the vector of observed covariates for patient \( i \)
- \( \beta \) is a vector of unknown parameters
- \( x(s_i) \) is the value of the spatial effect \( x(s) \) for patient \( i \).
Spatial survival: example

\[
\log(\text{hazard}) = \log(\text{baseline}) + f(\text{age}) + f(\text{white blood cell count}) + f(\text{deprivation index}) + f(\text{spatial}) + \text{sex}
\]
data(Leuk)
g = system.file("demodata/Leuk.graph", package="INLA")

formula = inla.surv(Leuk$time, Leuk$cens) ~ sex + age +
  f(inla.group(wbc), model="rw1") +
  f(inla.group(tpi), model="rw2") +
  f(district, model="besag", graph = g)

result = inla(formula, family="coxph", data=Leuk)

source(system.file("demodata/Leuk-map.R", package="INLA"))
Leuk.map(result$summary.random$district$mean)
plot(result)
baseline.hazard

PostMean  0.025%  0.5%  0.975%
PostDens [(Intercept)]
Mean = −9.401 SD = 0.261

PostDens [sex]
Mean = 0.077 SD = 0.069

PostDens [age]
Mean = 0.036 SD = 0.002
Outline

Background

Latent Gaussian model

Example: Continuous vs Discrete

Priors on functions
Let’s talk about sets, baby

We know that there is no “uniform” measure over all random functions, so we need to think about useful subsets.

- These are choices about smoothness.
- How differentiable do we want these things to be?
- Can there be sharp changes in the functions or their derivatives?
- Should these functions “look the same” everywhere?

And a much better question: How do we put a probability over a subset of functions?
Let’s talk about sets, baby

We know that there is no “uniform” measure over all random functions, so we need to think about useful subsets.

▶ These are choices about smoothness.
▶ How differentiable do we want these things to be?
▶ Can there be sharp changes in the functions or their derivatives?
▶ Should these functions “look the same” everywhere?

And a much better question: How do we put a probability over a subset of functions?
Let’s talk about sets, baby

We know that there is no “uniform” measure over all random functions, so we need to think about useful subsets.

- These are choices about smoothness.
- How differentiable do we want these things to be?
- Can there be sharp changes in the functions or their derivatives?
- Should these functions “look the same” everywhere?

And a much better question: How do we put a probability over a subset of functions?
Let’s talk about sets, baby

We know that there is no “uniform” measure over all random functions, so we need to think about useful subsets.

- These are choices about smoothness.
- How differentiable do we want these things to be?
- Can there be sharp changes in the functions or their derivatives?
- Should these functions “look the same” everywhere?

And a much better question: How do we put a probability over a subset of functions?
Let’s talk about sets, baby

We know that there is no “uniform” measure over all random functions, so we need to think about useful subsets.

▷ These are choices about smoothness.
▷ How differentiable do we want these things to be?
▷ Can there be sharp changes in the functions or their derivatives?
▷ Should these functions “look the same” everywhere?

And a much better question: How do we put a probability over a subset of functions?
Let’s talk about sets, baby

We know that there is no “uniform” measure over all random functions, so we need to think about useful subsets.

▶ These are choices about smoothness.
▶ How differentiable do we want these things to be?
▶ Can there be sharp changes in the functions or their derivatives?
▶ Should these functions “look the same” everywhere?

And a much better question: How do we put a probability over a subset of functions?
Aside: The Gaussian distribution

The most important distribution in probability and statistics is the Gaussian distribution, which has density

\[ p(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x - \mu)^2}{2\sigma^2}}. \]

- \( \mu \) is the mean (the centre of the distribution)
- \( \sigma^2 \) is the variance. Approximately 99.7% of the probability mass is within 3\( \sigma \) of the mean.
- A *standard* normal RV has mean 0 and variance 1.
- The *central limit theorem* says that empirical averages of independent RVs are approximately normal.

\[
\frac{n^{-1} \sum_{i=1}^{n} X_i - \mu}{\sqrt{n\sigma}} \xrightarrow{w} N(0, 1).
\]
Aside: The Gaussian distribution

The most important distribution in probability and statistics is the Gaussian distribution, which has density

\[ p(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \]

- \( \mu \) is the mean (the centre of the distribution)
- \( \sigma^2 \) is the variance. Approximately 99.7\% of the probability mass is within 3\( \sigma \) of the mean.
- A standard normal RV has mean 0 and variance 1.
- The central limit theorem says that empirical averages of independent RVs are approximately normal.

\[ \frac{n^{-1} \sum_{i=1}^{n} X_i - \mu}{\sqrt{n} \sigma} \xrightarrow{w} N(0,1). \]
Aside: The Gaussian distribution

The most important distribution in probability and statistics is the Gaussian distribution, which has density

\[ p(x) = \frac{1}{\sqrt{2\pi \sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \]

- \( \mu \) is the mean (the centre of the distribution)
- \( \sigma^2 \) is the variance. Approximately 99.7% of the probability mass is within 3\( \sigma \) of the mean.
- A *standard* normal RV has mean 0 and variance 1.
- The *central limit theorem* says that empirical averages of independent RVs are approximately normal.

\[ \frac{n^{-1} \sum_{i=1}^{n} X_i - \mu}{\sqrt{n\sigma}} \stackrel{w}{\to} N(0, 1). \]
Aside: The Gaussian distribution

The most important distribution in probability and statistics is the Gaussian distribution, which has density

\[ p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \]

- \( \mu \) is the mean (the centre of the distribution)
- \( \sigma^2 \) is the variance. Approximately 99.7% of the probability mass is within 3\( \sigma \) of the mean.
- A standard normal RV has mean 0 and variance 1.
- The central limit theorem says that empirical averages of independent RVs are approximately normal.

\[
\frac{n^{-1} \sum_{i=1}^{n} X_i - \mu}{\sqrt{n\sigma}} \xrightarrow{w} N(0, 1).
\]
Multivariate Gaussian distributions

Now, in the cases we care about, things are not univariate.

The multivariate Gaussian distribution has density

\[ p(x) = \frac{1}{\sqrt{2\pi|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}. \]

- \( \mu \) is still the average (now a d-dimensional vector)
- \( \Sigma \) is a \( d \times d \) covariance matrix. \( \Sigma_{ij} = \text{Cov}(x_i, x_j) \)
- For any constant vector \( a \), \( a^T \Sigma \) is normally distributed.
Multivariate Gaussian distributions

Now, in the cases we care about, things are not univariate.

The multivariate Gaussian distribution has density

\[ p(x) = \frac{1}{\sqrt{2\pi|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}. \]

- \( \mu \) is still the average (now a d-dimensional vector)
- \( \Sigma \) is a \( d \times d \) covariance matrix. \( \Sigma_{ij} = \text{Cov}(x_i, x_j) \)
- For any constant vector \( a \), \( a^T \Sigma \) is normally distributed.
Multivariate Gaussian distributions

Now, in the cases we care about, things are not unvarivariate.

The *multivariate Gaussian distribution* has density

\[
p(x) = \frac{1}{\sqrt{2\pi|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}.
\]

- \(\mu\) is still the average (now a d-dimensional vector)
- \(\Sigma\) is a \(d \times d\) covariance matrix. \(\Sigma_{ij} = \text{Cov}(x_i, x_j)\)
- For any constant vector \(a\), \(a^T \Sigma\) is normally distributed.
What does a multivariate Gaussian look like?

(from wikipedia!)
Aside: Sampling from multivariate Gaussian distributions

An important and useful skill is sampling from multivariate Gaussians.

- Compute the Cholesky factorisation of the covariance matrix $\Sigma = RR^T$. Then $x = Rz \sim N(0, \Sigma)$ if $z \sim N(0, I)$.

- Compute the Cholesky factorisation of the precision matrix $Q = \Sigma^{-1} = LL^T$. Then $x = L^{-T}z \sim N(0, \Sigma)$.

- General calculations with a Gaussian cost $O(n^3)$ flops, where $n$ is the dimension of the problem.

- For large $n$, we obviously don’t compute with “general” Gaussians!
Aside: Sampling from multivariate Gaussian distributions

An important and useful skill is sampling from multivariate Gaussians.

- Compute the Cholesky factorisation of the covariance matrix $\Sigma = RR^T$. Then $x = Rz \sim N(0, \Sigma)$ if $z \sim N(0, I)$.

- Compute the Cholesky factorisation of the precision matrix $Q = \Sigma^{-1} = LL^T$. Then $x = L^{-T}z \sim N(0, \Sigma)$.

- General calculations with a Gaussian cost $O(n^3)$ flops, where $n$ is the dimension of the problem.

- For large $n$, we obviously don’t compute with “general” Gaussians!
Aside: Sampling from multivariate Gaussian distributions

An important and useful skill is sampling from multivariate Gaussians.

- Compute the Cholesky factorisation of the covariance matrix \( \Sigma = RR^T \). Then \( x = Rz \sim N(0, \Sigma) \) if \( z \sim N(0, I) \).

- Compute the Cholesky factorisation of the precision matrix \( Q = \Sigma^{-1} = LL^T \). Then \( x = L^{-T}z \sim N(0, \Sigma) \).

- General calculations with a Gaussian cost \( \mathcal{O}(n^3) \) flops, where \( n \) is the dimension of the problem.

- For large \( n \), we obviously don’t compute with “general” Gaussians!
Aside: Sampling from multivariate Gaussian distributions

An important and useful skill is sampling from multivariate Gaussians.

- Compute the Cholesky factorisation of the covariance matrix \( \Sigma = RR^T \). Then \( x = Rz \sim N(0, \Sigma) \) if \( z \sim N(0, I) \).

- Compute the Cholesky factorisation of the precision matrix \( Q = \Sigma^{-1} = LL^T \). Then \( x = L^{-T}z \sim N(0, \Sigma) \).

- General calculations with a Gaussian cost \( O(n^3) \) flops, where \( n \) is the dimension of the problem.

- For large \( n \), we obviously don’t compute with “general” Gaussians!
If we have a process that is occurring everywhere in space, it is natural to try to model it using some sort of function.

- This is hard (for good measure theory-ish reasons)!
- We typically make our lives easier by making everything Gaussian.
- What makes a function Gaussian?
Gaussian random fields

If we have a process that is occurring everywhere in space, it is natural to try to model it using some sort of function.

- This is hard (for good measure theory-ish reasons)!

- We typically make our lives easier by making everything Gaussian.

- What makes a function Gaussian?
If we have a process that is occurring everywhere in space, it is natural to try to model it using some sort of function.

- This is hard (for good measure theory-ish reasons)!
- We typically make our lives easier by making everything Gaussian.
- What makes a function Gaussian?
Gaussian random fields

If we are trying to model $u(s)$ what sort of things do we need?

- We don’t ever observe a function everywhere.

- If $u$ is a vector of observations of $u(s)$ at different locations, we want this to be normally distributed:

  $$ u = (u(s_1), \ldots, u(s_p))^T \sim N(0, \Sigma) $$

- This is actually quite tricky: the covariance matrix $\Sigma$ will need to depend on the set of observation sites and always has to be positive definite.

- It turns out you can actually do this by setting $\Sigma_{ij} = c(s_i, s_j)$ for some covariance function $c(\cdot, \cdot)$.

- Not every function will ensure that $\Sigma$ is positive definite!
Gaussian random fields

If we are trying to model $u(s)$ what sort of things do we need?

- We don’t ever observe a function everywhere.

- If $u$ is a vector of observations of $u(s)$ at different locations, we want this to be normally distributed:

  $$u = (u(s_1), \ldots, u(s_p))^T \sim \mathcal{N}(0, \Sigma)$$

- This is actually quite tricky: the covariance matrix $\Sigma$ will need to depend on the set of observation sites and always has to be positive definite.

- It turns out you can actually do this by setting $\Sigma_{ij} = c(s_i, s_j)$ for some covariance function $c(\cdot, \cdot)$.

- Not every function will ensure that $\Sigma$ is positive definite!
Gaussian random fields

If we are trying to model $u(s)$ what sort of things do we need?

- We don’t ever observe a function everywhere.

- If $u$ is a vector of observations of $u(s)$ at different locations, we want this to be normally distributed:

$$
\mathbf{u} = (u(s_1), \ldots, u(s_p))^T \sim \mathcal{N}(\mathbf{0}, \Sigma)
$$

- This is actually quite tricky: the covariance matrix $\Sigma$ will need to depend on the set of observation sites and always has to be positive definite.

- It turns out you can actually do this by setting $\Sigma_{ij} = c(s_i, s_j)$ for some covariance function $c(\cdot, \cdot)$.

- Not every function will ensure that $\Sigma$ is positive definite!
Gaussian random fields

If we are trying to model \( u(s) \) what sort of things do we need?

- We don’t ever observe a function everywhere.

- If \( u \) is a vector of observations of \( u(s) \) at different locations, we want this to be normally distributed:

\[
\mathbf{u} = (u(s_1), \ldots, u(s_p))^T \sim \mathcal{N}(\mathbf{0}, \Sigma)
\]

- This is actually quite tricky: the covariance matrix \( \Sigma \) will need to depend on the set of observation sites and always has to be positive definite.

- It turns out you can actually do this by setting \( \Sigma_{ij} = c(s_i, s_j) \) for some covariance function \( c(\cdot, \cdot) \).

- Not every function will ensure that \( \Sigma \) is positive definite!
Gaussian random fields

If we are trying to model $u(s)$ what sort of things do we need?

- We don’t ever observe a function everywhere.

- If $u$ is a vector of observations of $u(s)$ at different locations, we want this to be normally distributed:

  $$u = (u(s_1), \ldots, u(s_p))^T \sim N(0, \Sigma)$$

- This is actually quite tricky: the covariance matrix $\Sigma$ will need to depend on the set of observation sites and always has to be positive definite.

- It turns out you can actually do this by setting $\Sigma_{ij} = c(s_i, s_j)$ for some covariance function $c(\cdot, \cdot)$.

- Not every function will ensure that $\Sigma$ is positive definite!
Gaussian random fields

**Defn: Gaussian random fields**

A Gaussian random field $u(s)$ is defined by a mean function $\mu(s)$ and a covariance function $c(s_1, s_2)$. It has the property that, for every finite collection of points $\{s_1, \ldots, s_p\}$,

$$ u \equiv (u(s_1), \ldots, u(s_p))^T \sim N(0, \Sigma), $$

where $\Sigma_{ij} = c(s_i, s_j)$.

- $\Sigma$ will almost never be sparse.
- It is typically very hard to find families of parameterised covariance functions.
- It isn’t straightforward to make this work for multivariate, spatiotemporal, or processes on non-flat spaces.