# Graphical Sparse Precision Matrix Estimation and the Ensemble Information Filter

Berent, Feda and Sondre

equinor

The International EnKF Workshop 2022
Balestrand, Norway
June 2, 2022

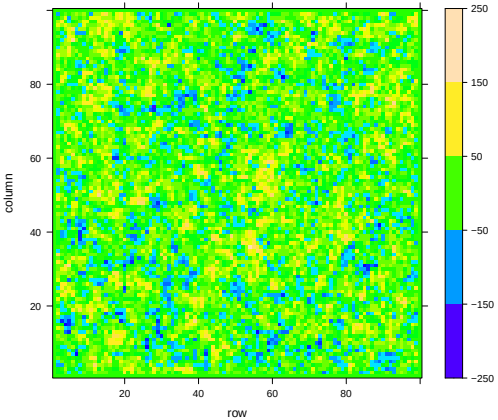
# Who we are: A subset of Scientific COmpUting Team (SCOUT)

# A stochastic spatio-temporal dynamical model

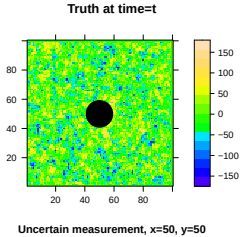
**The stochastic heat equation**

with SPDE

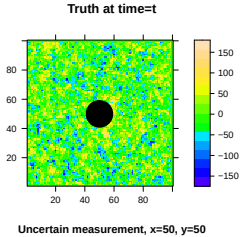
$$\partial_t \boldsymbol{u} = \alpha \nabla^2 \boldsymbol{u} + \sigma d\boldsymbol{W}_t$$



# The EnKF solution (Evensen, 1994; Burgers, Van Leeuwen, and Evensen, 1998)



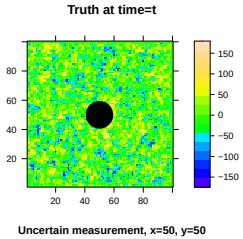
# The EnKF solution (Evensen, 1994; Burgers, Van Leeuwen, and Evensen, 1998)




Gaussian prior at time $t-1$

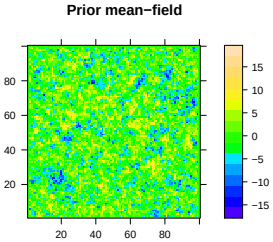
$$\mathcal{N}\left(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}\right) \underset{100 \text{ realizations}}{\longrightarrow}$$

# The EnKF solution (Evensen, 1994; Burgers, Van Leeuwen, and Evensen, 1998)

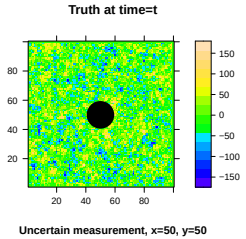



Gaussian prior at time $t-1$

$$\mathcal{N}\left(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}\right) \xrightarrow[\text{100 realizations}]{}$$

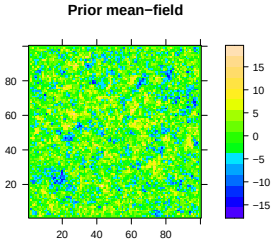


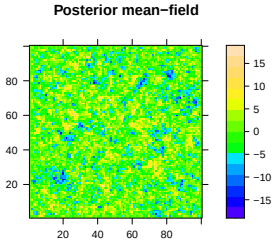
# The EnKF solution (Evensen, 1994; Burgers, Van Leeuwen, and Evensen, 1998)




Gaussian prior at time $t-1$

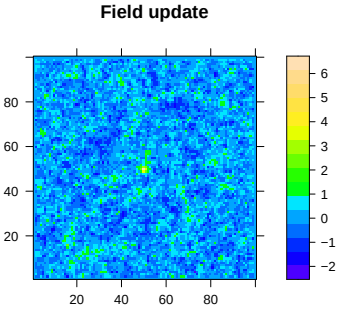
$$\mathcal{N}\left(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}\right) \xrightarrow[\text{100 realizations}]{}$$



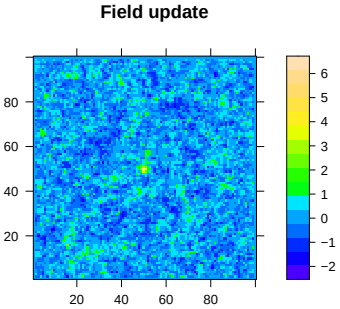



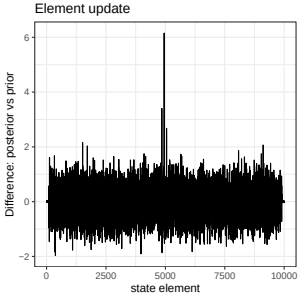

equinor

# EnKF, all is well?

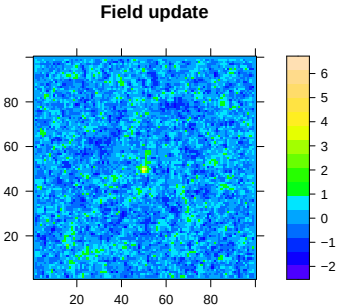


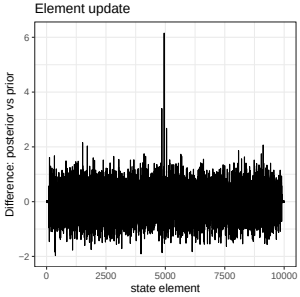
# EnKF, all is well?

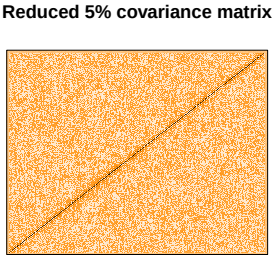


# EnKF, all is well?

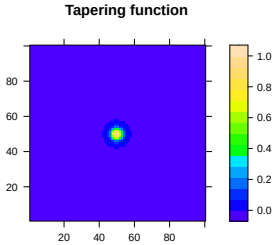


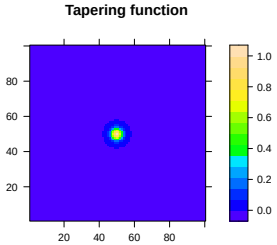
# The EnKF localisation solution

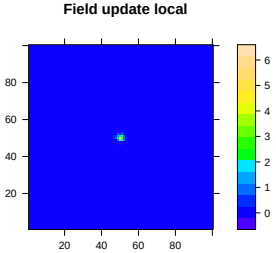
Illustrated with local analysis type localisation.
(Anderson, 2003; Evensen, 2003; Ott et al., 2004; Hunt, Kostelich, and Szunyogh, 2007)

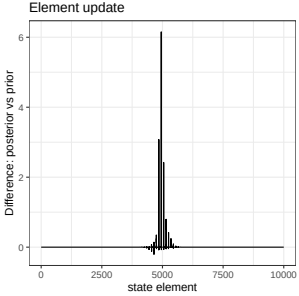


# The EnKF localisation solution

Illustrated with local analysis type localisation.
(Anderson, 2003; Evensen, 2003; Ott et al., 2004; Hunt, Kostelich, and Szunyogh, 2007)



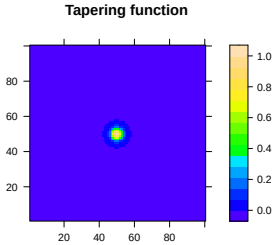



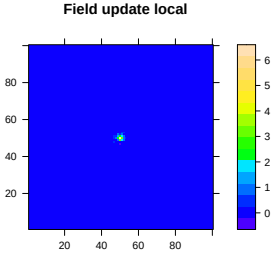



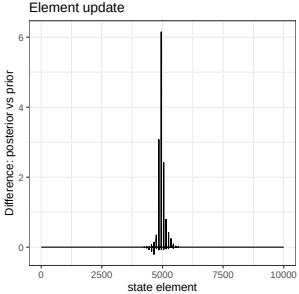
# The EnKF localisation solution

Illustrated with local analysis type localisation.
(Anderson, 2003; Evensen, 2003; Ott et al., 2004; Hunt, Kostelich, and Szunyogh, 2007)

- Involves tuning of hyperparameters (functional form and parameters of tapering function).



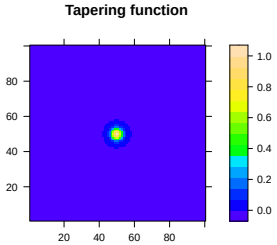



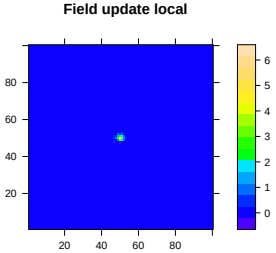



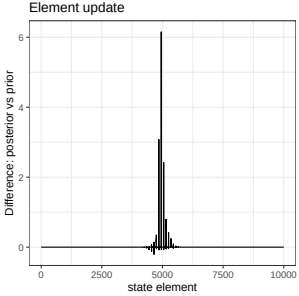
# The EnKF localisation solution

Illustrated with local analysis type localisation.
(Anderson, 2003; Evensen, 2003; Ott et al., 2004; Hunt, Kostelich, and Szunyogh, 2007)

- Involves tuning of hyperparameters (functional form and parameters of tapering function).
- Works on existing (implicit) covariance or residuals - only weakens the *direct* connection between $(i, j)$ and $(k, l)$, but still allow direct connections.



# Reparametrisation: The computational-statistics mindset

**Question**

Is there a way to reparametrise the problem to avoid constrained estimation?

# Reparametrisation: The computational-statistics mindset

## Question

Is there a way to reparametrise the problem to avoid constrained estimation?

## Examples:

- Estimate logit-transformed probability instead of constrained probability.

Constrained VS unconstrained

for $0 < p < 1$ do

$$\arg\min_p -\sum_i y_i \log(p) + (1 - y_i)\log(1 - p)$$

VS

given transform $h(\theta) = \dfrac{1}{1 + e^{-\theta}}$ do

$$\arg\min_\theta -\sum_i y_i \log\left(h(\theta)\right) + (1 - y_i)\log\left(1 - h(\theta)\right)$$

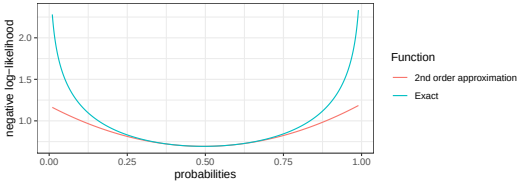
# Reparametrisation: The computational-statistics mindset

## Question

Is there a way to reparametrise the problem to avoid constrained estimation?

## Examples:

- Estimate logit-transformed probability instead of constrained probability.
- The link-functions in Generalized Linear Models



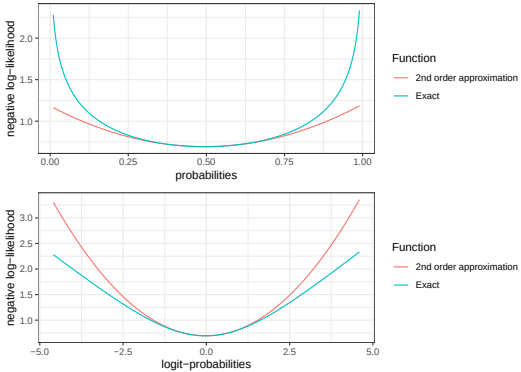
# Reparametrisation: The computational-statistics mindset

## Question

Is there a way to reparametrise the problem to avoid constrained estimation?

## Examples:

- Estimate logit-transformed probability instead of constrained probability.
- The link-functions in Generalized Linear Models



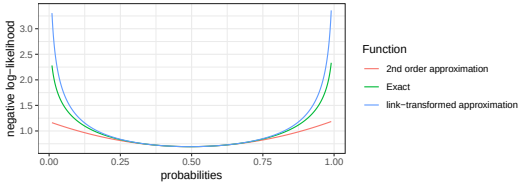
# Reparametrisation: The computational-statistics mindset

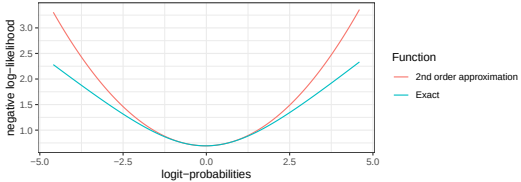
## Question

Is there a way to reparametrise the problem to avoid constrained estimation?

## Examples:

- Estimate logit-transformed probability instead of constrained probability.
- The link-functions in Generalized Linear Models



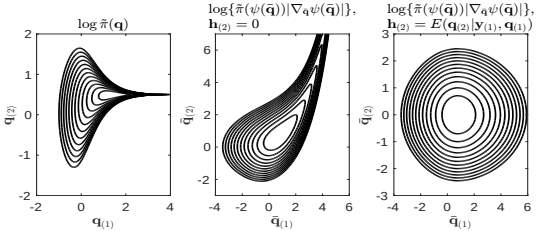
# Reparametrisation: The computational-statistics mindset

## Question

Is there a way to reparametrise the problem to avoid constrained estimation?

## Examples:

- Estimate logit-transformed probability instead of constrained probability.
- The link-functions in Generalized Linear Models
- Constant information parametrisation instead of RMHMC



# Reparametrisation for spatio-temporal models

## Question

- Is there such a reparametrisation? What are we actually looking for in a reparametrisation?

# Reparametrisation for spatio-temporal models

## Question

- Is there such a reparametrisation? What are we actually looking for in a reparametrisation?

## Temporary answer with some guidance

- Dependence is local! If this could be baked into the parametrisation pre estimation then this would be (highly) beneficial.

Currently, with the ordinary covariance parametrisation of the Gaussian and corresponding likelihood-estimate, every parameter is potentially connected with all other parameters. There is initially no preference on "local connections".

# Reparametrisation for spatio-temporal models

## Question

- Is there such a reparametrisation? What are we actually looking for in a reparametrisation?

## Temporary answer with some guidance

- Dependence is local! If this could be baked into the parametrisation pre estimation then this would be (highly) beneficial.

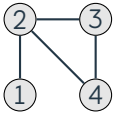
Currently, with the ordinary covariance parametrisation of the Gaussian and corresponding likelihood-estimate, every parameter is potentially connected with all other parameters. There is initially no preference on "local connections".

Seek parametrisation with "preference" for local connections/dependence

# Formalising connectivity and locality

**A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$**

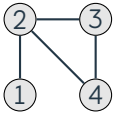
- **Vertices** $\mathcal{V} = \{1, \ldots, d\}$
- **Edges** $\mathcal{E} = \{(i,j)\}$ so that $(i,j) \in \mathcal{E}$ if $i$ and $j$ are directly connected
- **Neighbours:** $ne(i) = \{j; (i,j) \in \mathcal{E}\}$



# Formalising connectivity and locality

**A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$**

- **Vertices** $\mathcal{V} = \{1, \ldots, d\}$
- **Edges** $\mathcal{E} = \{(i,j)\}$ so that $(i,j) \in \mathcal{E}$ if $i$ and $j$ are directly connected
- **Neighbours:** $ne(i) = \{j; (i,j) \in \mathcal{E}\}$




In the example above

- $\mathcal{V} = \{1, 2, 3, 4\}$
- $\mathcal{E} = \{(1, 2), (2, 3), (2, 4), (3, 4)\}$
- $ne(1) = \{2\}$, $ne(2) = \{1, 3, 4\}$

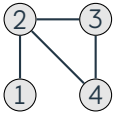
# Formalising connectivity and locality

**A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$**

- **Vertices** $\mathcal{V} = \{1, \ldots, d\}$
- **Edges** $\mathcal{E} = \{(i,j)\}$ so that $(i,j) \in \mathcal{E}$ if $i$ and $j$ are directly connected
- **Neighbours:** $ne(i) = \{j; (i,j) \in \mathcal{E}\}$

In the example above

- $\mathcal{V} = \{1, 2, 3, 4\}$
- $\mathcal{E} = \{(1,2), (2,3), (2,4), (3,4)\}$
- $ne(1) = \{2\}$, $ne(2) = \{1, 3, 4\}$




**Markov Random Field (MRF)**

$\boldsymbol{x} \in R^d$ is MRF w.r.t. a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V} = 1, \ldots, d$ if Markov property

$$x_i \perp x_{-(ne(i),i)} | x_{ne(i)}, \text{ holds } \forall i \in \mathcal{V}$$

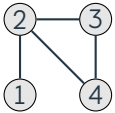
# Formalising connectivity and locality

**A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$**

- **Vertices** $\mathcal{V} = \{1, \ldots, d\}$
- **Edges** $\mathcal{E} = \{(i,j)\}$ so that $(i,j) \in \mathcal{E}$ if $i$ and $j$ are directly connected
- **Neighbours:** $ne(i) = \{j; (i,j) \in \mathcal{E}\}$

In the example above

- $\mathcal{V} = \{1, 2, 3, 4\}$
- $\mathcal{E} = \{(1,2), (2,3), (2,4), (3,4)\}$
- $ne(1) = \{2\}$, $ne(2) = \{1, 3, 4\}$




**Markov Random Field (MRF)**

$\boldsymbol{x} \in R^d$ is MRF w.r.t. a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V} = 1, \ldots, d$ if Markov property

$$x_i \perp x_{-(ne(i),i)} | x_{ne(i)}, \text{ holds } \forall i \in \mathcal{V}$$

In the example we would require that

$$x_1 \perp (x_3, x_4) | x_2, \; x_3 \perp x_1 | (x_2, x_4), \; x_4 \perp x_1 | (x_2, x_3)$$

# Markov properties

Keep modelling at a local level...

# Markov properties

## Keep modelling at a local level...

... by using Markov properties

$$E[x_i|x_{-i}] = E[x_i|x_{ne(i)}]$$

and even stronger that

$$x_i|x_{-i} = x_i|x_{ne(i)}$$
$$\Updownarrow$$
$$x_i \perp x_{-(i,ne(i))} = x_i|x_{ne(i)}$$

# Gaussien Markov Random Fields (GMRF)

## Gaussian Markov Random Field (Rue and Held, 2005)

A random vector $\boldsymbol{x} \in R^d$ is a Gaussian Markov Random Field with respect to the graph $\mathcal{G} = (\{1, \ldots, d\}, \mathcal{E})$, with mean $\mu$ and SPD precision matrix $\Lambda$ if

$$p(x) = (2\pi)^{-\frac{d}{2}} \sqrt{|\Lambda|} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top \Lambda (\boldsymbol{x} - \boldsymbol{\mu})\right)$$

and

$$\Lambda_{i,j} \neq 0 \Leftrightarrow (i,j) \in \mathcal{E} \forall i \neq j.$$

# Gaussien Markov Random Fields (GMRF)

## Gaussian Markov Random Field (Rue and Held, 2005)

A random vector $\boldsymbol{x} \in R^d$ is a Gaussian Markov Random Field with respect to the graph $\mathcal{G} = (\{1, \ldots, d\}, \mathcal{E})$, with mean $\mu$ and SPD precision matrix $\Lambda$ if

$$p(x) = (2\pi)^{-\frac{d}{2}} \sqrt{|\Lambda|} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top \Lambda (\boldsymbol{x} - \boldsymbol{\mu})\right)$$

and

$$\Lambda_{i,j} \neq 0 \Leftrightarrow (i,j) \in \mathcal{E} \forall i \neq j.$$

Notice that connectivity and local dependence is *directly* specified through the non-zero elements of the precision matrix. No other constraints needed.

# GMRF parametrisation matters: Estimation

Connectivity and Markov properties implied by the precision matrix.

- Unconstrained optimisation (ML-estimation)
- Massively important for estimation: Spurious correlations disappear.

$$\Lambda = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} & 0 & \cdots & & & & & \cdots & 0 \\ \Lambda_{21} & \Lambda_{22} & \Lambda_{23} & 0 & \cdots & & & & \cdots & 0 \\ 0 & \Lambda_{32} & \Lambda_{33} & \Lambda_{34} & 0 & \cdots & & & \cdots & 0 \\ \vdots & & & & & \ddots & & & & \vdots \\ 0 & \cdots & & & \cdots & 0 & \Lambda_{d-2,d-3} & \Lambda_{d-2,d-2} & \Lambda_{d-2,d-18} & 0 \\ 0 & \cdots & & & & \cdots & 0 & \Lambda_{d-1,d-2} & \Lambda_{d-1,d-1} & \Lambda_{d-1,d} \\ 0 & \cdots & & & & & \cdots & 0 & \Lambda_{d,d-1} & \Lambda_{d,d} \end{bmatrix}$$

# GMRF parametrisation matters: Inversion

$$\Sigma = \begin{bmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{bmatrix}, \quad \Lambda = \Sigma^{-1}$$

## Covariance parametrisation

## Precision parametrisation

# GMRF parametrisation matters: Inversion

$$\Sigma = \begin{bmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{bmatrix}, \quad \Lambda = \Sigma^{-1}$$

## Covariance parametrisation

Convenient for working with marginal distributions of $\boldsymbol{x}$

- $E\left[\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}\right] = \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}$
- $Var\left[\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}\right] = \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\top}$

Not for conditional distributions (requires matrix inversion, e.g. Kalman filter).

## Precision parametrisation

# GMRF parametrisation matters: Inversion

$$\Sigma = \begin{bmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{bmatrix}, \quad \Lambda = \Sigma^{-1}$$

## Covariance parametrisation

Convenient for working with marginal distributions of $\boldsymbol{x}$

- $E\left[\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}\right] = \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}$
- $Var\left[\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}\right] = \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^\top$

Not for conditional distributions (requires matrix inversion, e.g. Kalman filter).

## Precision parametrisation

Convenient for working with conditional distributions of $\boldsymbol{x}$

- $E[x_i|x_{-i}] = \mu_i - \frac{1}{\Lambda_{i,i}} \sum_{j \neq i} \Lambda_{i,j}(x_j - \mu_j)$
- $Prec(\boldsymbol{x}_A|\boldsymbol{x}_{-A}) = \Lambda_{A,A}$
- $Corr(x_i, x_j|x_{-ij}) = -\frac{\Lambda_{i,j}}{\sqrt{\Lambda_{i,i}\Lambda j,j}} \; i \neq j$

# Autoregressive (1) example


$$x_t = \phi x_{t-1} + \epsilon_t,\ x_1 \sim \mathcal{N}\left(0, \frac{1}{1-\phi^2}\right),\ \epsilon_t \sim \mathcal{N}(0,1)$$

Covariance parametrisation...

Precision parametrisation...

# Autoregressive (1) example


$$x_t = \phi x_{t-1} + \epsilon_t,\ x_1 \sim \mathcal{N}\left(0, \frac{1}{1-\phi^2}\right),\ \epsilon_t \sim \mathcal{N}(0,1)$$

## Covariance parametrisation...

... is dense!

$$\mathbf{\Sigma} = \begin{bmatrix} B(1,1) & \cdots & B(1,T) \\ \vdots & \ddots & \vdots \\ B(T,1) & \cdots & B(T,T) \end{bmatrix},\ B(i,j) = \frac{\phi^{|i-j|}}{1-\phi^2}$$

## Precision parametrisation...

# Autoregressive (1) example


$$x_t = \phi x_{t-1} + \epsilon_t,\ x_1 \sim \mathcal{N}\left(0, \frac{1}{1-\phi^2}\right),\ \epsilon_t \sim \mathcal{N}(0,1)$$

## Covariance parametrisation...

... is dense!

$$\boldsymbol{\Sigma} = \begin{bmatrix} B(1,1) & \cdots & B(1,T) \\ \vdots & \ddots & \vdots \\ B(T,1) & \cdots & B(T,T) \end{bmatrix},\ B(i,j) = \frac{\phi^{|i-j|}}{1-\phi^2}$$

## Precision parametrisation...

... is sparse!

$$\Lambda = \begin{bmatrix} 1 & -\phi & & & \\ -\phi & 1+\phi^2 & -\phi & & \\ & \ddots & \ddots & \ddots & \\ & & -\phi & 1+\phi^2 & -\phi \\ & & & -\phi & 1 \end{bmatrix}$$

# The Information Filter (Moore and Anderson, 1979)

Employs the canonical parametrization of the multivatiate Gaussian:

$$\boldsymbol{\nu} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$$

**Predict step**

**Update step**

# The Information Filter (Moore and Anderson, 1979)

Employs the canonical parametrization of the multivatiate Gaussian:

$$\boldsymbol{\nu} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$$

**Predict step**

`<Non-Beautiful-Equations>`

**Update step**

# The Information Filter (Moore and Anderson, 1979)

Employs the canonical parametrization of the multivatiate Gaussian:

$$\boldsymbol{\nu} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$$

**Predict step**

`<Non-Beautiful-Equations>`

**Update step**

$$\boldsymbol{\nu}_{t|t} = \boldsymbol{\nu}_{t|t-1} + \boldsymbol{H}_t^\top \boldsymbol{\Lambda}_{\boldsymbol{y}_t} \boldsymbol{y}_t$$

$$\boldsymbol{\Lambda}_{t|t} = \boldsymbol{\Lambda}_{t|t-1} + \boldsymbol{H}_t^\top \boldsymbol{\Lambda}_{\boldsymbol{y}_t} \boldsymbol{H}_t$$

# Extension to the ensemble variant?

**Sample from belief**

$$\boldsymbol{x}_{t-1|t-1}^{(i)} \sim \mathcal{N}\left(\boldsymbol{\mu}_{t-1|t-1}, \boldsymbol{\Lambda}_{t-1|t-1}\right) \; i = 1, \ldots, n$$

**Predict**

$$\boldsymbol{x}_{t|t-1}^{(i)} = g(\boldsymbol{x}_{t-1|t-1}^{(i)})$$

**Estimate**

Using sample $\{\boldsymbol{x}_{t|t-1}^{(i)}\}_{i=1}^{n}$ estimate $\hat{\boldsymbol{\mu}}_{t|t-1}$ and $\hat{\boldsymbol{\Lambda}}_{t|t-1}$ w.r.t. graph $\mathcal{G}$

**Update**

$$\hat{\boldsymbol{\nu}}_{t|t-1} = \hat{\boldsymbol{\Lambda}}_{t|t-1}\hat{\boldsymbol{\mu}}_{t|t-1}$$
$$\hat{\boldsymbol{\nu}}_{t|t} = \hat{\boldsymbol{\nu}}_{t|t-1} + \boldsymbol{H}_t^\top \boldsymbol{\Lambda}_{\boldsymbol{y}_t}\boldsymbol{y}_t$$
$$\hat{\boldsymbol{\Lambda}}_{t|t} = \hat{\boldsymbol{\Lambda}}_{t|t-1} + \boldsymbol{H}_t^\top \boldsymbol{\Lambda}_{\boldsymbol{y}_t}\boldsymbol{H}_t$$

# Recapture

- Spatio-temporal models leads to difficulties for ensemble algorithms due to spurious correlations

# Recapture

- Spatio-temporal models leads to difficulties for ensemble algorithms due to spurious correlations
- We want to model at a local level, leads naturally to Markov properties. Can we do a reparametrisation?

# Recapture

- Spatio-temporal models leads to difficulties for ensemble algorithms due to spurious correlations
- We want to model at a local level, leads naturally to Markov properties. Can we do a reparametrisation?
- Magically, for GMRF, the precision matrix is sparse under Markov assumptions

# Recapture

- Spatio-temporal models leads to difficulties for ensemble algorithms due to spurious correlations
- We want to model at a local level, leads naturally to Markov properties. Can we do a reparametrisation?
- Magically, for GMRF, the precision matrix is sparse under Markov assumptions
- Reparametrisation in the Kalman filter leads to the Information filter, having extremely easy additive updating

# Recapture

- Spatio-temporal models leads to difficulties for ensemble algorithms due to spurious correlations
- We want to model at a local level, leads naturally to Markov properties. Can we do a reparametrisation?
- Magically, for GMRF, the precision matrix is sparse under Markov assumptions
- Reparametrisation in the Kalman filter leads to the Information filter, having extremely easy additive updating
- Bonus: Newton optimisation in the canonical parametrisation leads to natural gradients (ml) or Fisher scoring (stats). Gradients with optimality properties.

# Recapture

- Spatio-temporal models leads to difficulties for ensemble algorithms due to spurious correlations
- We want to model at a local level, leads naturally to Markov properties. Can we do a reparametrisation?
- Magically, for GMRF, the precision matrix is sparse under Markov assumptions
- Reparametrisation in the Kalman filter leads to the Information filter, having extremely easy additive updating
- Bonus: Newton optimisation in the canonical parametrisation leads to natural gradients (ml) or Fisher scoring (stats). Gradients with optimality properties.
- Extension of the information filter to an ensemble variant is not straight forward.

Background

Graphical Sparse Precision Matrix Estimation

The Ensemble Information Filter

Examples

# The sample covariance estimate...

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n-1} \sum_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top$$

| Positive properties | Negative properties |
| --- | --- |
| | |

# The sample covariance estimate...

$$\hat{\Sigma} = \frac{1}{n-1} \sum_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top$$

## Positive properties

- A small scaling away from the Gaussian maximum-likelihood estimate (AUMVE)

## Negative properties

# The sample covariance estimate...

$$\hat{\Sigma} = \frac{1}{n-1} \sum_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top$$

## Positive properties

- A small scaling away from the Gaussian maximum-likelihood estimate (AUMVE)

## Negative properties

# The sample covariance estimate...

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n-1} \sum_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top$$

## Positive properties

- A small scaling away from the Gaussian maximum-likelihood estimate (AUMVE)
- Distribution independent and unbiased

## Negative properties

# The sample covariance estimate...

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n-1} \sum_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top$$

## Positive properties

- A small scaling away from the Gaussian maximum-likelihood estimate (AUMVE)
- Distribution independent and unbiased
- For Kalman filter, there is no need for explicit calculation ($R^{p \times p}$). Use a centred ensemble ($R^{n \times p}$) for updates (in ensemble space)

## Negative properties

# The sample covariance estimate...

$$\hat{\Sigma} = \frac{1}{n-1} \sum_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top$$

## Positive properties

- A small scaling away from the Gaussian maximum-likelihood estimate (AUMVE)
- Distribution independent and unbiased
- For Kalman filter, there is no need for explicit calculation ($R^{p \times p}$). Use a centred ensemble ($R^{n \times p}$) for updates (in ensemble space)

## Negative properties

- Unstable, and even singular for $n < p$.

# The sample covariance estimate...

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n-1} \sum_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top$$

## Positive properties

- A small scaling away from the Gaussian maximum-likelihood estimate (AUMVE)
- Distribution independent and unbiased
- For Kalman filter, there is no need for explicit calculation ($R^{p \times p}$). Use a centred ensemble ($R^{n \times p}$) for updates (in ensemble space)

## Negative properties

- Unstable, and even singular for $n < p$.
- Inefficient in the non-asymptotic case

# The sample covariance estimate...

$$\hat{\Sigma} = \frac{1}{n-1} \sum_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top$$

## Positive properties

- A small scaling away from the Gaussian maximum-likelihood estimate (AUMVE)
- Distribution independent and unbiased
- For Kalman filter, there is no need for explicit calculation ($R^{p \times p}$). Use a centred ensemble ($R^{n \times p}$) for updates (in ensemble space)

## Negative properties

- Unstable, and even singular for $n < p$.
- Inefficient in the non-asymptotic case
- Does not employ information on locality known pre-estimation

# (Stein-type) shrinkage of the covariance estimator

**We employ the Ledoit-Wolf shrinkage estimator** (Ledoit and Wolf, 2004)

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{T}} = (1-\lambda)\hat{\boldsymbol{\Sigma}} + \lambda \boldsymbol{T}$$

Targets to fix stability, singularity and efficiency.

## In general...

- Choice of objective
- Choice of target matrix, or shrinkage in general
- Computationally intensive with cross validation

## Specifics of what we employ

# (Stein-type) shrinkage of the covariance estimator

**We employ the Ledoit-Wolf shrinkage estimator** (Ledoit and Wolf, 2004)

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{T}} = (1-\lambda)\hat{\boldsymbol{\Sigma}} + \lambda \boldsymbol{T}$$

Targets to fix stability, singularity and efficiency.

## In general...

- Choice of objective
- Choice of target matrix, or shrinkage in general
- Computationally intensive with cross validation

## Specifics of what we employ

- The Frobenius norm $||\boldsymbol{\Sigma} - \hat{\boldsymbol{\Sigma}}_{\boldsymbol{T}}||_{fb}$

# (Stein-type) shrinkage of the covariance estimator

**We employ the Ledoit-Wolf shrinkage estimator** (Ledoit and Wolf, 2004)

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{T}} = (1-\lambda)\hat{\boldsymbol{\Sigma}} + \lambda \boldsymbol{T}$$

Targets to fix stability, singularity and efficiency.

## In general...

- Choice of objective
- Choice of target matrix, or shrinkage in general
- Computationally intensive with cross validation

## Specifics of what we employ

- The Frobenius norm $||\boldsymbol{\Sigma} - \hat{\boldsymbol{\Sigma}}_{\boldsymbol{T}}||_{fb}$
- The SCV diagonal $\boldsymbol{T} = \texttt{diag}\left(\hat{\boldsymbol{\Sigma}}\right)$

# (Stein-type) shrinkage of the covariance estimator

**We employ the Ledoit-Wolf shrinkage estimator** (Ledoit and Wolf, 2004)

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{T}} = (1-\lambda)\hat{\boldsymbol{\Sigma}} + \lambda \boldsymbol{T}$$

Targets to fix stability, singularity and efficiency.

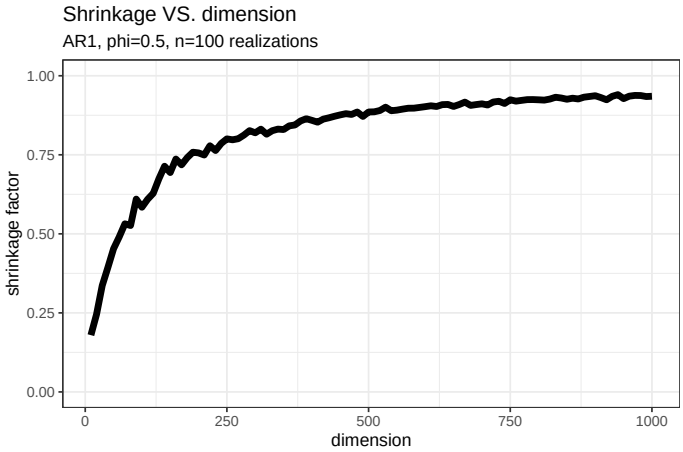
## In general...

- Choice of objective
- Choice of target matrix, or shrinkage in general
- Computationally intensive with cross validation

## Specifics of what we employ

- The Frobenius norm $||\boldsymbol{\Sigma} - \hat{\boldsymbol{\Sigma}}_{\boldsymbol{T}}||_{fb}$
- The SCV diagonal $\boldsymbol{T} = \mathtt{diag}\left(\hat{\boldsymbol{\Sigma}}\right)$
- Asymptotic optimality results for shrinkage factor $\hat{\lambda}$ (Touloumis, 2015)

# The Stein-type covariance shrinkage estimate: In high dimensions



# Asymptotic Stein-type shrinkage

**Very smart!**

1. Non-parametric model with loose moment conditions
2. Find exact solution, $\hat{\lambda}$, in terms of (unknown) trace-expectations
3. Omit (asymptotically) negligible terms and find consistent estimators for others
4. Find efficient ways to calculate consistent estimators for big $p$ small $n$

# Asymptotic Stein-type shrinkage

**Very smart!**

1. Non-parametric model with loose moment conditions
2. Find exact solution, $\hat{\lambda}$, in terms of (unknown) trace-expectations
3. Omit (asymptotically) negligible terms and find consistent estimators for others
4. Find efficient ways to calculate consistent estimators for big $p$ small $n$

- So smart that it has recently been proposed applied with the EnKF: Nino-Ruiz, Guzman, and Jabba (2021)!

# Asymptotic Stein-type shrinkage

**Very smart!**

1. Non-parametric model with loose moment conditions
2. Find exact solution, $\hat{\lambda}$, in terms of (unknown) trace-expectations
3. Omit (asymptotically) negligible terms and find consistent estimators for others
4. Find efficient ways to calculate consistent estimators for big $p$ small $n$

- So smart that it has recently been proposed applied with the EnKF: Nino-Ruiz, Guzman, and Jabba (2021)!

**Lacks the informed structure of locality $\rightarrow$ shrinks off-diagonal elements to zero**

# The sparsity of the precision. Built-in graph and Markov order

**The precision matrix may intrinsically hold such knowledge!**

# The sparsity of the precision. Built-in graph and Markov order

**The precision matrix may intrinsically hold such knowledge!**

But can we estimate the precision from data?

# The sparsity of the precision. Built-in graph and Markov order

**The precision matrix may intrinsically hold such knowledge!**

But can we estimate the precision from data?

- Must match the ease of estimation as the sample covariance matrix...

# The sparsity of the precision. Built-in graph and Markov order

**The precision matrix may intrinsically hold such knowledge!**

## But can we estimate the precision from data?

- Must match the ease of estimation as the sample covariance matrix...
- ... and possible to use with numerical linear algebra. Not enough memory to hold a dense $p \times p$ matrix

# The sparsity of the precision. Built-in graph and Markov order

**The precision matrix may intrinsically hold such knowledge!**

But can we estimate the precision from data?

- Must match the ease of estimation as the sample covariance matrix...
- ... and possible to use with numerical linear algebra. Not enough memory to hold a dense $p \times p$ matrix

2008 Graphical lasso: L1 penalized maximum likelihood estimate over SPD matrices (Friedman, Hastie, and Tibshirani, 2008)

# The sparsity of the precision. Built-in graph and Markov order

**The precision matrix may intrinsically hold such knowledge!**

But can we estimate the precision from data?

- Must match the ease of estimation as the sample covariance matrix...
- ... and possible to use with numerical linear algebra. Not enough memory to hold a dense $p \times p$ matrix

2008 Graphical lasso: L1 penalized maximum likelihood estimate over SPD matrices (Friedman, Hastie, and Tibshirani, 2008)

2010 -> Column-by-column methods (Yuan, 2010; Cai, Liu, and Luo, 2011)

# The sparsity of the precision. Built-in graph and Markov order

**The precision matrix may intrinsically hold such knowledge!**

But can we estimate the precision from data?

- Must match the ease of estimation as the sample covariance matrix...
- ... and possible to use with numerical linear algebra. Not enough memory to hold a dense $p \times p$ matrix

2008 Graphical lasso: L1 penalized maximum likelihood estimate over SPD matrices (Friedman, Hastie, and Tibshirani, 2008)

2010 -> Column-by-column methods (Yuan, 2010; Cai, Liu, and Luo, 2011)

2017 Tuning Intensive Graph Estimation and Regression (TIGER and EPIC) (Zhao and Liu, 2014; Liu and Wang, 2017)

# Precision matrix estimation with respect to a graph

The algorithms estimate the graph through Lasso-type algorithms (L1-penalization)

... but we already know $\mathcal{G}$!

# Precision matrix estimation with respect to a graph

The algorithms estimate the graph through Lasso-type algorithms (L1-penalization)

... but we already know $\mathcal{G}$!

This problem has received little attention, but solutions can be found in e.g. (Hastie et al., 2009; Zhou et al., 2011)

- Constrained Gaussian maximum likelihood estimation. Iterative algorithms, requiring the Gaussian likelihood.

# Precision matrix estimation with respect to a graph

The algorithms estimate the graph through Lasso-type algorithms (L1-penalization)

... but we already know $\mathcal{G}$!

This problem has received little attention, but solutions can be found in e.g. (Hastie et al., 2009; Zhou et al., 2011)

- Constrained Gaussian maximum likelihood estimation. Iterative algorithms, requiring the Gaussian likelihood.

Thankfully, Le and Zhong (2022) just came up with exactly what we need

# Precision matrix estimation with respect to a graph

The algorithms estimate the graph through Lasso-type algorithms (L1-penalization)

... but we already know $\mathcal{G}$!

This problem has received little attention, but solutions can be found in e.g. (Hastie et al., 2009; Zhou et al., 2011)

- Constrained Gaussian maximum likelihood estimation. Iterative algorithms, requiring the Gaussian likelihood.

Thankfully, Le and Zhong (2022) just came up with exactly what we need

- **Non-parametric precision matrix estimation with respect to a known graphical structure**

# SPD precision matrix estimation with respect to a graph

## The method of Le and Zhong (2022)

- Column-by-column sub-sample covariance estimate inversion
- Sub-sample covariance estimate blocks identified due to the knowledge of the graph $\mathcal{G}$

## Pitfalls

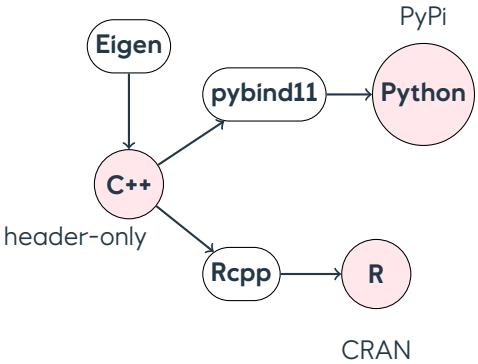
- Not necessarily positive definite
- Not necessarily symmetric

## Add some ingredients

- The efficient asymptotic shrinkage of Touloumis (2015)
- Symmetry conversion: $\hat{\mathbf{\Lambda}} = \frac{1}{2}\left(\tilde{\mathbf{\Lambda}} + \tilde{\mathbf{\Lambda}}^{\top}\right)$

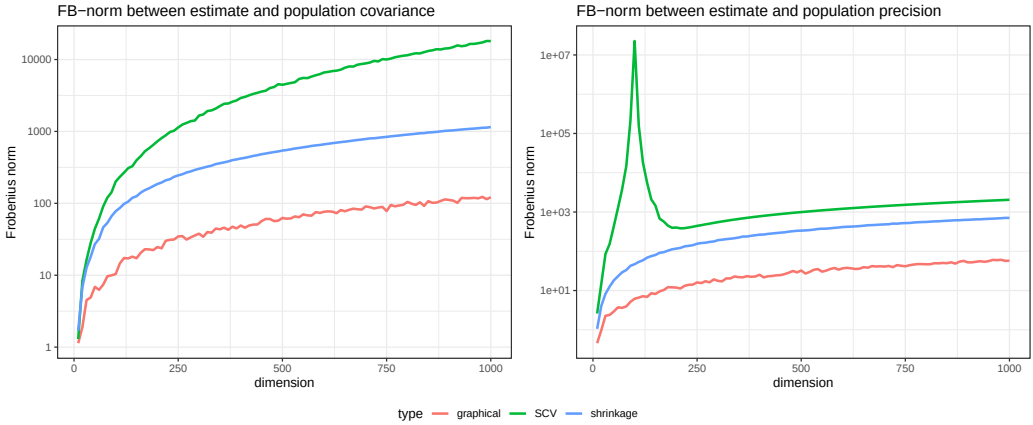
# GraphSPME: Graphical Sparse Precision Matrix Estimation

**Open sourced at** `https://github.com/equinor/GraphSPME`
**Paper at** `https://arxiv.org/abs/2205.07584`




```python
from graphspme import prec_sparse, cov_shrink_spd
import numpy as np
from scipy import sparse
def rar1(T, phi):
    ...
# Sample data
n, p, psi = 100, 100, 0.6
x = np.tile(rar1(p,psi), (n,1))
# Estimate covariance
Sigma = cov_shrink_spd(x)
# Estimate precision
diagonals = [[1]*p,[1]*(p-1),[1]*(p-1)]
G = sparse.diags(diagonals, [0, -1, 1], format="csr")
Prec = prec_sparse(x, G)
```

# AR1 estimation results



# Recapture

- The sample covariance estimate is easy to compute and numerically tractable for the EnKF, but is typically not the best possible estimator (singular and has high variance)

# Recapture

- The sample covariance estimate is easy to compute and numerically tractable for the EnKF, but is typically not the best possible estimator (singular and has high variance)
- A typical remedy is to use shrinkage, giving SPD estimates. But does not utilize information of sparsity

# Recapture

- The sample covariance estimate is easy to compute and numerically tractable for the EnKF, but is typically not the best possible estimator (singular and has high variance)
- A typical remedy is to use shrinkage, giving SPD estimates. But does not utilize information of sparsity
- GraphSPME combines the method of Le and Zhong (2022) for precision estimation w.r.t. a graph, with asymptotic shrinkage methods Touloumis (2015) to ensure SPD estimates.

Background

Graphical Sparse Precision Matrix Estimation

The Ensemble Information Filter

Examples

# The Ensemble Information Filter

**Sample from belief**

$$\boldsymbol{x}_{t-1|t-1}^{(i)} \sim \mathcal{N}\left(\boldsymbol{\mu}_{t-1|t-1}, \boldsymbol{\Lambda}_{t-1|t-1}\right) \; i = 1, \ldots, n$$

**Predict**

$$\boldsymbol{x}_{t|t-1}^{(i)} = g(\boldsymbol{x}_{t-1|t-1}^{(i)})$$

**Estimate**

Using sample $\{\boldsymbol{x}_{t|t-1}^{(i)}\}_{i=1}^{n}$ estimate $\hat{\boldsymbol{\mu}}_{t|t-1}$ and $\hat{\boldsymbol{\Lambda}}_{t|t-1}$ w.r.t. graph $\mathcal{G}$ (using GraphSPME)

**Update**

$$\hat{\boldsymbol{\nu}}_{t|t-1} = \hat{\boldsymbol{\Lambda}}_{t|t-1}\hat{\boldsymbol{\mu}}_{t|t-1}$$
$$\hat{\boldsymbol{\nu}}_{t|t} = \hat{\boldsymbol{\nu}}_{t|t-1} + \boldsymbol{H}_t^{\mathsf{T}}\boldsymbol{\Lambda}_{\boldsymbol{y}_t}\boldsymbol{y}_t$$
$$\hat{\boldsymbol{\Lambda}}_{t|t} = \hat{\boldsymbol{\Lambda}}_{t|t-1} + \boldsymbol{H}_t^{\mathsf{T}}\boldsymbol{\Lambda}_{\boldsymbol{y}_t}\boldsymbol{H}_t$$

# Numerical black magic for scalability

- The Cholesky decomposition $\boldsymbol{L}\boldsymbol{L}^\top = \boldsymbol{\Lambda}$ is computationally efficiently retrieved due to sparsity.

# Numerical black magic for scalability

- The Cholesky decomposition $\boldsymbol{L}\boldsymbol{L}^\top = \boldsymbol{\Lambda}$ is computationally efficiently retrieved due to sparsity.
- Computing $\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{L}^{-\top}\boldsymbol{z}$ by back-substitution is not more costly than $\boldsymbol{L}\boldsymbol{z}$ as $\boldsymbol{L}$ is triangular.

# Numerical black magic for scalability

- The Cholesky decomposition $\boldsymbol{L}\boldsymbol{L}^\top = \boldsymbol{\Lambda}$ is computationally efficiently retrieved due to sparsity.
- Computing $\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{L}^{-\top}\boldsymbol{z}$ by back-substitution is not more costly than $\boldsymbol{L}\boldsymbol{z}$ as $\boldsymbol{L}$ is triangular.
- Solving a sparse linear system like $\boldsymbol{\Lambda}\boldsymbol{\mu} = \boldsymbol{\eta}$ is efficiently done using either e.g. Cholesky decomposition or LU factorization when $\boldsymbol{\Lambda}$ can be held in memory, or by sparsity aware conjugate gradient when $\boldsymbol{\Lambda}$ is too large to be held in memory.

# Numerical black magic for scalability

- The Cholesky decomposition $\boldsymbol{L}\boldsymbol{L}^\top = \boldsymbol{\Lambda}$ is computationally efficiently retrieved due to sparsity.
- Computing $\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{L}^{-\top}\boldsymbol{z}$ by back-substitution is not more costly than $\boldsymbol{L}\boldsymbol{z}$ as $\boldsymbol{L}$ is triangular.
- Solving a sparse linear system like $\boldsymbol{\Lambda}\boldsymbol{\mu} = \boldsymbol{\eta}$ is efficiently done using either e.g. Cholesky decomposition or LU factorization when $\boldsymbol{\Lambda}$ can be held in memory, or by sparsity aware conjugate gradient when $\boldsymbol{\Lambda}$ is too large to be held in memory.
- `sparse_prec` efficiently estimates a sparse precision matrix according to the GraphSPME algorithm.

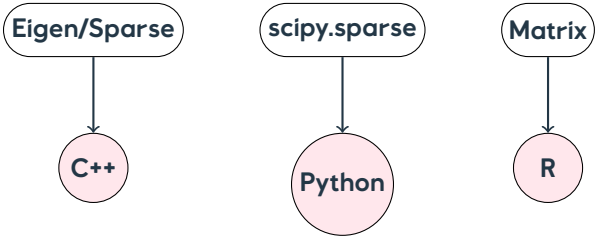
# Numerical black magic for scalability

- The Cholesky decomposition $\boldsymbol{L}\boldsymbol{L}^\top = \boldsymbol{\Lambda}$ is computationally efficiently retrieved due to sparsity.
- Computing $\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{L}^{-\top}\boldsymbol{z}$ by back-substitution is not more costly than $\boldsymbol{L}\boldsymbol{z}$ as $\boldsymbol{L}$ is triangular.
- Solving a sparse linear system like $\boldsymbol{\Lambda}\boldsymbol{\mu} = \boldsymbol{\eta}$ is efficiently done using either e.g. Cholesky decomposition or LU factorization when $\boldsymbol{\Lambda}$ can be held in memory, or by sparsity aware conjugate gradient when $\boldsymbol{\Lambda}$ is too large to be held in memory.
- `sparse_prec` efficiently estimates a sparse precision matrix according to the GraphSPME algorithm.
- $\hat{\boldsymbol{\Lambda}}$ is guaranteed to be symmetric positive definite by the GraphSPME algorithm, and a solution to the mean-precision parametrisation thus always exists, furthermore the covariance matrix may in principle be retrieved.

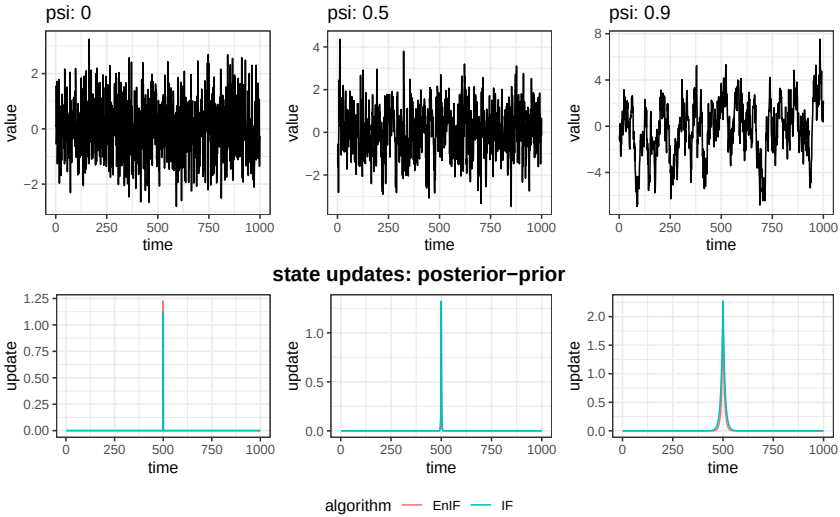
# Implementation

**Examples at** `https://github.com/equinor/Enkf-Workshop-2022`

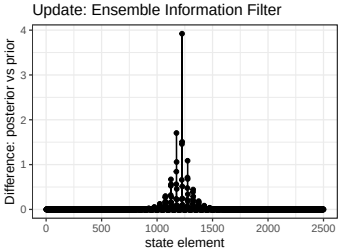
Generally easy to implement, but requires a sparse matrix library with corresponding sparse linear solvers.

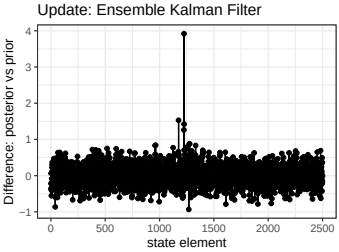


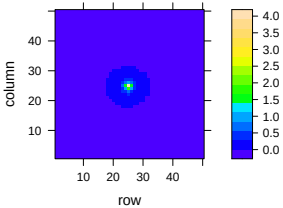
# AR1 updates

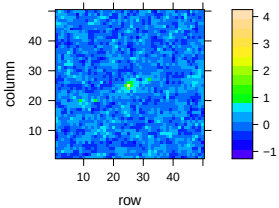

psi: 0
psi: 0.5
psi: 0.9
value
time
state updates: posterior−prior
update
algorithm
EnIF
IF

# Random field updates, $n = 200$ vs $n = 10000$

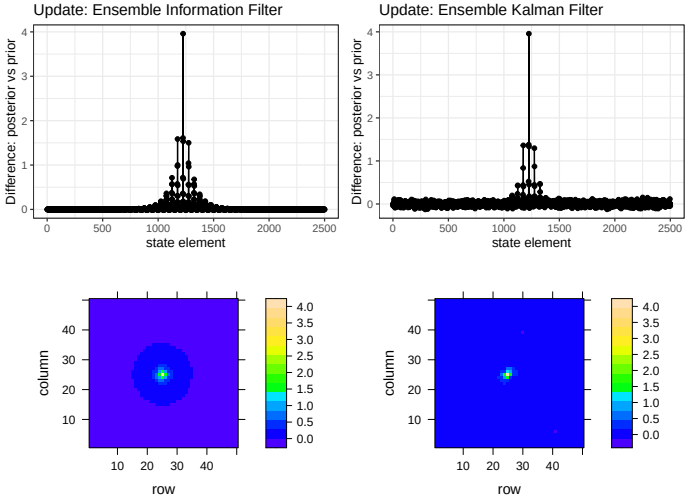


# Random field updates, $n = 200$ vs $n = 10000$



# Recapture

- GraphSPME allows the extension to the Ensemble Information Filter.

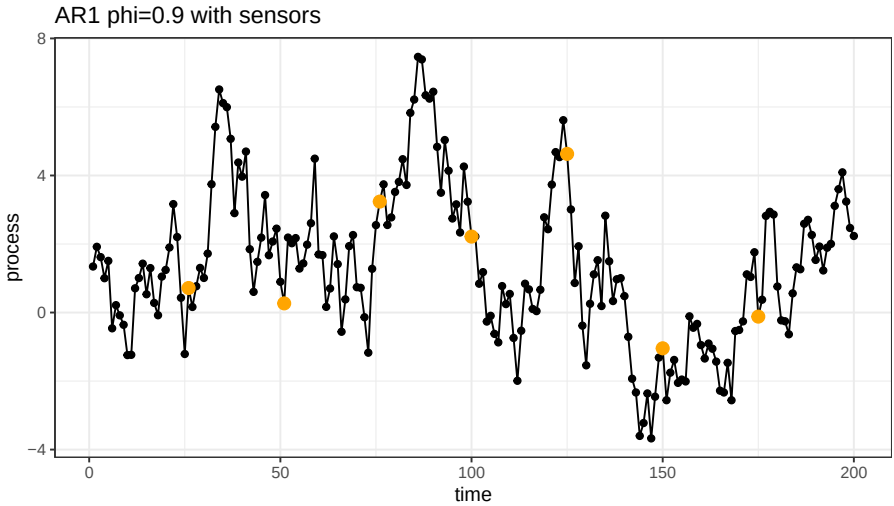
# Recapture

- GraphSPME allows the extension to the Ensemble Information Filter.
- The numerical linear algebra relies heavily on smart algorithms for sparse matrices and linear solvers.

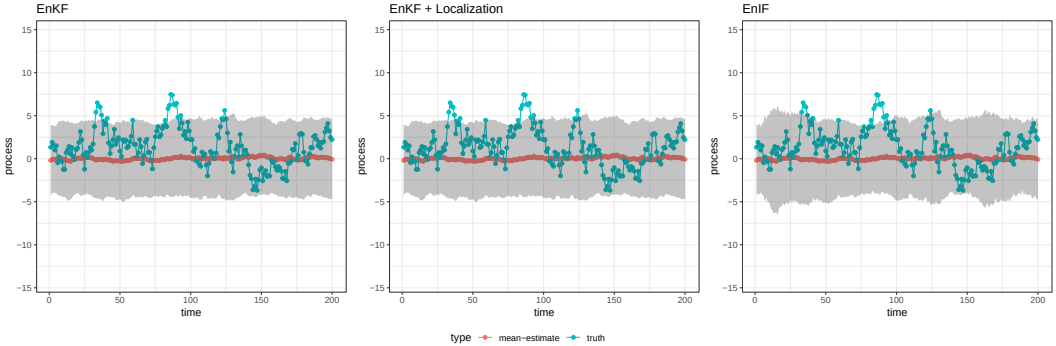
# Recapture

- GraphSPME allows the extension to the Ensemble Information Filter.
- The numerical linear algebra relies heavily on smart algorithms for sparse matrices and linear solvers.
- Filtering updates are much less noisy than for that of the EnKF, and seems to solve the problem of localisation without requiring tuning of arbitrary kernels.

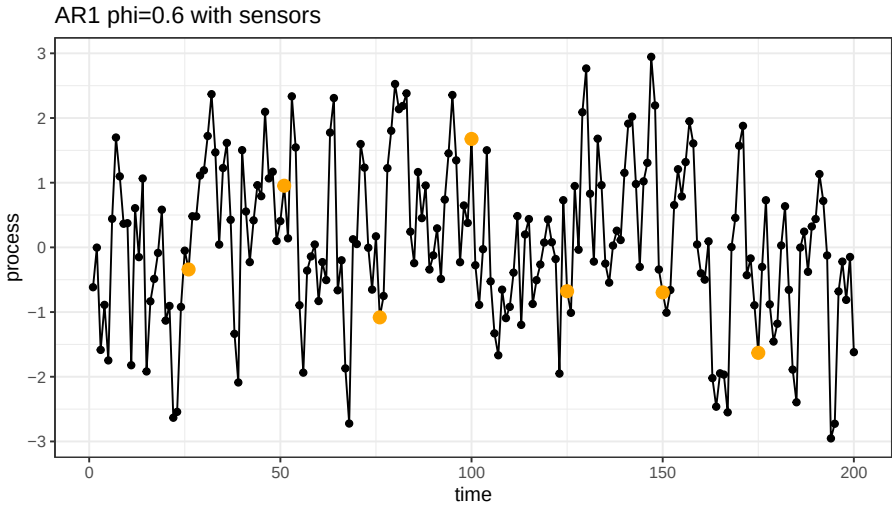
# AR1, strong dependence



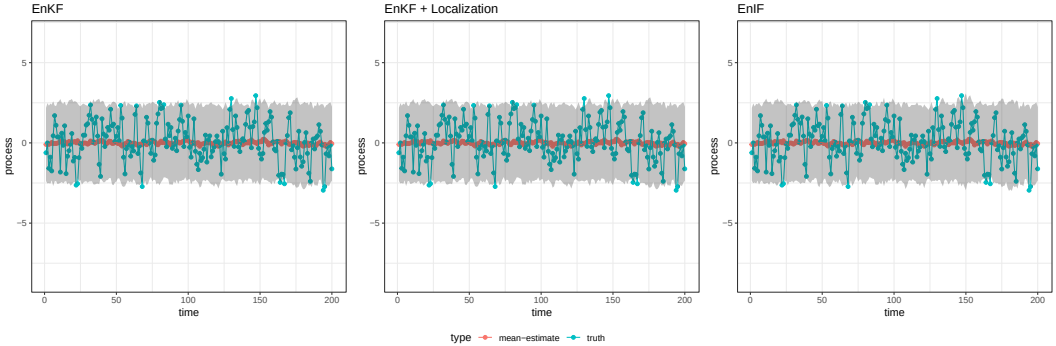
# AR1: $\phi = 0.9$


EnKF
EnKF + Localization
EnIF
process
time
15
10
5
0
−5
−10
−15
0
50
100
150
200
type
mean−estimate
truth

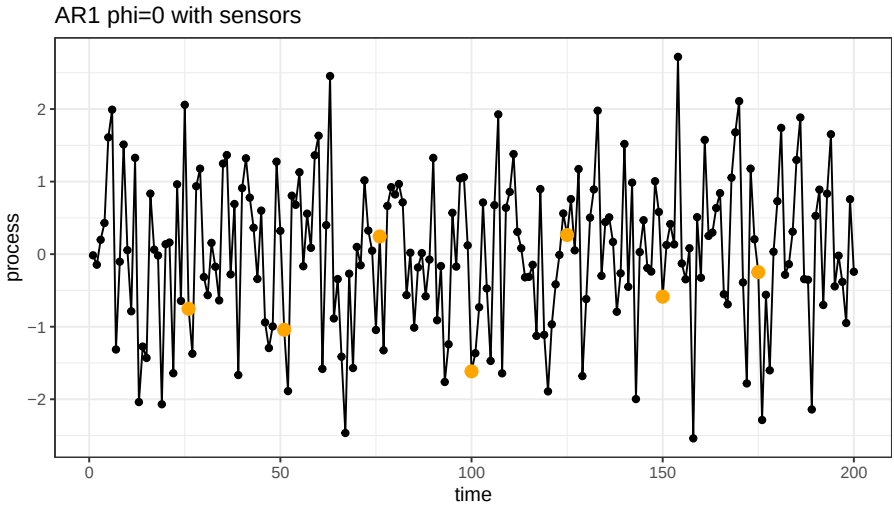
# AR1, middle dependence



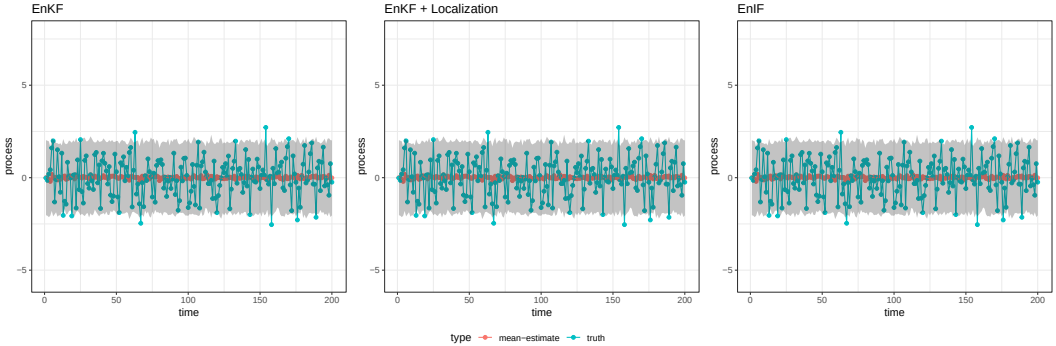
# AR1: $\phi = 0.6$


EnKF
EnKF + Localization
EnIF
process
time
5
0
−5
0
50
100
150
200
type
mean−estimate
truth

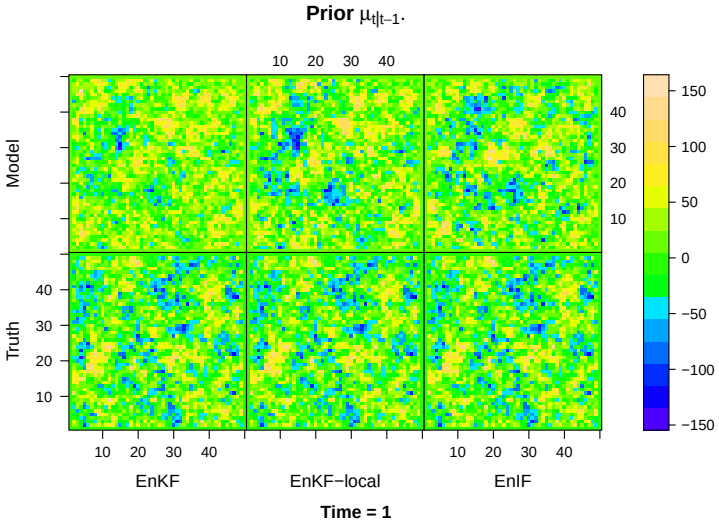
# AR1, no dependence, noise



# AR1: $\phi = 0.0$

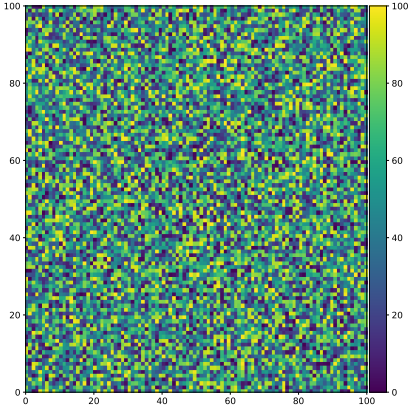

EnKF
EnKF + Localization
EnIF
process
time
5
0
−5
0
50
100
150
200
type
mean−estimate
truth

# Stochastic heat equation: Pure state estimation



# Stochastic heat equation: Combined parameter-state estimation

$\partial_t \boldsymbol{u} = \alpha \nabla^2 \boldsymbol{u} + \sigma d\boldsymbol{W}_t,\ (\alpha, \sigma)$ are unknown and must be estimated.



# Stochastic heat equation: Combined parameter-state estimation

$$\partial_t \boldsymbol{u} = \alpha \nabla^2 \boldsymbol{u} + \sigma d\boldsymbol{W}_t, \ (\alpha, \sigma) \text{ are unknown and must be estimated.}$$

- Have not yet started to think about the joint estimation problem.

# Stochastic heat equation: Combined parameter-state estimation

$$\partial_t \boldsymbol{u} = \alpha \nabla^2 \boldsymbol{u} + \sigma d\boldsymbol{W}_t, \ (\alpha, \sigma) \text{ are unknown and must be estimated.}$$

- Have not yet started to think about the joint estimation problem.
- In principle, possible to do in the same way as for EnKF (assume additive and specify a fully connected graph).

# Stochastic heat equation: Combined parameter-state estimation

$$\partial_t \boldsymbol{u} = \alpha \nabla^2 \boldsymbol{u} + \sigma d\boldsymbol{W}_t, \ (\alpha, \sigma) \text{ are unknown and must be estimated.}$$

- Have not yet started to think about the joint estimation problem.
- In principle, possible to do in the same way as for EnKF (assume additive and specify a fully connected graph).
- Perhaps more elegant with the two-step estimation procedure e.g. when using the Laplace approximatin to integrate out a latent state.

# Full recap

- Filtering algorithms needs to be informed on locality to avoid noisy updates due to $n << p$.

# Full recap

- Filtering algorithms needs to be informed on locality to avoid noisy updates due to $n << p$.
- A natural assumption for spatio-temporal models is assumptions of Markov properties.

# Full recap

- Filtering algorithms needs to be informed on locality to avoid noisy updates due to $n << p$.
- A natural assumption for spatio-temporal models is assumptions of Markov properties.
- For Gaussian Markov Random Fields, the precision matrix is sparse. The corresponding information filter utilizes this.

# Full recap

- Filtering algorithms needs to be informed on locality to avoid noisy updates due to $n << p$.
- A natural assumption for spatio-temporal models is assumptions of Markov properties.
- For Gaussian Markov Random Fields, the precision matrix is sparse. The corresponding information filter utilizes this.
- We allow the extension to the Ensemble Information Filter by creating GraphSPME: graphical sparse precision matrix estimation

# Full recap

- Filtering algorithms needs to be informed on locality to avoid noisy updates due to $n << p$.
- A natural assumption for spatio-temporal models is assumptions of Markov properties.
- For Gaussian Markov Random Fields, the precision matrix is sparse. The corresponding information filter utilizes this.
- We allow the extension to the Ensemble Information Filter by creating GraphSPME: graphical sparse precision matrix estimation
- The filtering updates from the ensemble information filter seem to be smooth and comparatively without noise (information efficient). The method is free of tuning and arbitrary measures of distance.

# Questions?

# Bibliography I


- Anderson, Jeffrey L (2003). "A local least squares framework for ensemble filtering". In: *Monthly Weather Review* 131.4, pp. 634–642.
- Burgers, Gerrit, Peter Jan Van Leeuwen, and Geir Evensen (1998). "Analysis scheme in the ensemble Kalman filter". In: *Monthly weather review* 126.6, pp. 1719–1724.
- Cai, Tony, Weidong Liu, and Xi Luo (2011). "A constrained â□□ 1 minimization approach to sparse precision matrix estimation". In: *Journal of the American Statistical Association* 106.494, pp. 594–607.
- Evensen, Geir (1994). "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics". In: *Journal of Geophysical Research: Oceans* 99.C5, pp. 10143–10162.
- – (2003). "The ensemble Kalman filter: Theoretical formulation and practical implementation". In: *Ocean dynamics* 53.4, pp. 343–367.

# Bibliography II


- Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2008). "Sparse inverse covariance estimation with the graphical lasso". In: *Biostatistics* 9.3, pp. 432–441.
- Hastie, Trevor et al. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer.
- Hunt, Brian R, Eric J Kostelich, and Istvan Szunyogh (2007). "Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter". In: *Physica D: Nonlinear Phenomena* 230.1-2, pp. 112–126.
- Le, Thien-Minh and Ping-Shou Zhong (2022). "High-dimensional precision matrix estimation with a known graphical structure". In: *Stat* 11.1, e424.
- Ledoit, Olivier and Michael Wolf (2004). "A well-conditioned estimator for large-dimensional covariance matrices". In: *Journal of multivariate analysis* 88.2, pp. 365–411.

# Bibliography III


- Liu, Han and Lie Wang (2017). "Tiger: A tuning-insensitive approach for optimally estimating gaussian graphical models". In: *Electronic Journal of Statistics* 11.1, pp. 241–294.
- Moore, John Barratt and B Anderson (1979). *Optimal filtering*. Prentice-Hall New York.
- Nino-Ruiz, Elias D, Luis Guzman, and Daladier Jabba (2021). "An ensemble kalman filter implementation based on the ledoit and wolf covariance matrix estimator". In: *Journal of Computational and Applied Mathematics* 384, p. 113163.
- Ott, Edward et al. (2004). "A local ensemble Kalman filter for atmospheric data assimilation". In: *Tellus A: Dynamic Meteorology and Oceanography* 56.5, pp. 415–428.
- Rue, Havard and Leonhard Held (2005). *Gaussian Markov random fields: theory and applications*. Chapman and Hall/CRC.

# Bibliography IV


- Touloumis, Anestis (2015). "Nonparametric Stein-type shrinkage covariance matrix estimators in high-dimensional settings". In: *Computational Statistics & Data Analysis* 83, pp. 251–261.
- Yuan, Ming (2010). "High dimensional inverse covariance matrix estimation via linear programming". In: *The Journal of Machine Learning Research* 11, pp. 2261–2286.
- Zhao, Tuo and Han Liu (2014). "Calibrated precision matrix estimation for high-dimensional elliptical distributions". In: *IEEE transactions on Information Theory* 60.12, pp. 7874–7887.
- Zhou, Shuheng et al. (2011). "High-dimensional covariance estimation based on Gaussian graphical models". In: *The Journal of Machine Learning Research* 12, pp. 2975–3026.