Linear Triangular Transport at Scale

Berent Lunde

equinor

EnKF Workshop, Os Norway
June 17, 2024

Table of Contents

EnKF's and convergence

Method scalability

KLD, Structure, & EnIF

Innovations for scalability

Synthetic reservoir application: Sequential EnIF

Do EnKF's trivially converge at an infinite ensemble size?

?

Do EnKF's trivially converge at an infinite ensemble size?

No.

How come they do not converge?

- Consider the stochastic heat equation,

$$du_t(\boldsymbol{x}) = \alpha \operatorname{div} \nabla u_t(\boldsymbol{x})\, dt + \sigma\, dW_t. \tag{1}$$

  Let $\boldsymbol{u}$ be a p-vector of values indexed by time t and space $\boldsymbol{x}$.

How come they do not converge?

- Consider the stochastic heat equation,

$$du_t(\boldsymbol{x}) = \alpha \operatorname{div} \nabla u_t(\boldsymbol{x})\, dt + \sigma\, dW_t. \tag{1}$$

  Let $\boldsymbol{u}$ be a p-vector of values indexed by time t and space $\boldsymbol{x}$.

- Let $\boldsymbol{U}_n$ be an $n \times p$ matrix of n-samples, $\boldsymbol{u}^{(i)}$. The sample covariance

$$\hat{\boldsymbol{\Sigma}}^*_{\boldsymbol{u}} = \frac{1}{n-1} \sum_{i=1}^{n} (\boldsymbol{u}^{(i)} - \overline{\boldsymbol{u}})(\boldsymbol{u}^{(i)} - \overline{\boldsymbol{u}})^\top, \tag{2}$$

  does not *trivially* converge to the population covariance $\Sigma_{\boldsymbol{u}} = E[(\boldsymbol{u} - E[\boldsymbol{u}])(\boldsymbol{u} - E[\boldsymbol{u}])^\top]$ when both n *and* $p \to \infty$.

How come they do not converge?

- Consider the stochastic heat equation,

$$du_t(\boldsymbol{x}) = \alpha \operatorname{div} \nabla u_t(\boldsymbol{x})\, dt + \sigma\, dW_t. \tag{1}$$

  Let $\boldsymbol{u}$ be a p-vector of values indexed by time t and space $\boldsymbol{x}$.

- Let $\boldsymbol{U}_n$ be an $n \times p$ matrix of n-samples, $\boldsymbol{u}^{(i)}$. The sample covariance

$$\hat{\boldsymbol{\Sigma}}^*_{\boldsymbol{u}} = \frac{1}{n-1} \sum_{i=1}^{n} (\boldsymbol{u}^{(i)} - \overline{\boldsymbol{u}})(\boldsymbol{u}^{(i)} - \overline{\boldsymbol{u}})^\top, \tag{2}$$

  does not *trivially* converge to the population covariance $\Sigma_{\boldsymbol{u}} = E[(\boldsymbol{u} - E[\boldsymbol{u}])(\boldsymbol{u} - E[\boldsymbol{u}])^\top]$ when both n *and* $p \to \infty$.

- EnKF's (estimated) Kalman gains are a function of sample covariance++.

Why care about convergence also when $p \to \infty$?

We work with spatio-temporal models.

Why care about convergence also when $p \to \infty$?

We work with spatio-temporal models.

- Often, numerical integration promised to work as $\Delta \boldsymbol{x} \to 0$.

Why care about convergence also when $p \to \infty$?

We work with spatio-temporal models.

- Often, numerical integration promised to work as $\Delta \boldsymbol{x} \to 0$.
- But $\Delta \boldsymbol{x} \to 0$ implies $p \to \infty$ in a statistical setting.

Why care about convergence also when $p \to \infty$?

We work with spatio-temporal models.

- Often, numerical integration promised to work as $\Delta \boldsymbol{x} \to 0$.
- But $\Delta \boldsymbol{x} \to 0$ implies $p \to \infty$ in a statistical setting.
- For ensemble based methods, we need to guarantee convergence under simultaneous limits $p \to \infty$ and $n \to \infty$.

What would we expect if things go wrong?

Common statistical estimation theory on overfitting would suggest

1. Random updates in the mean or single realizations.
2. Overconfidence (loss of variability) due to belief in (random) connections and propagating a Bayesian update through them.

What would we expect if things go wrong?

Common statistical estimation theory on overfitting would suggest

1. Random updates in the mean or single realizations.
2. Overconfidence (loss of variability) due to belief in (random) connections and propagating a Bayesian update through them.

Uh, oh. This sounds familiar

What would we expect if things go wrong?

Common statistical estimation theory on overfitting would suggest

1. Random updates in the mean or single realizations.
2. Overconfidence (loss of variability) due to belief in (random) connections and propagating a Bayesian update through them.

Uh, oh. This sounds familiar

- Spurious correlations: Random updates in mean and single realizations.

What would we expect if things go wrong?

Common statistical estimation theory on overfitting would suggest

1. Random updates in the mean or single realizations.
2. Overconfidence (loss of variability) due to belief in (random) connections and propagating a Bayesian update through them.

Uh, oh. This sounds familiar

- Spurious correlations: Random updates in mean and single realizations.
- Ensemble collapse: Loss of variability.

What would we expect if things go wrong?

Common statistical estimation theory on overfitting would suggest

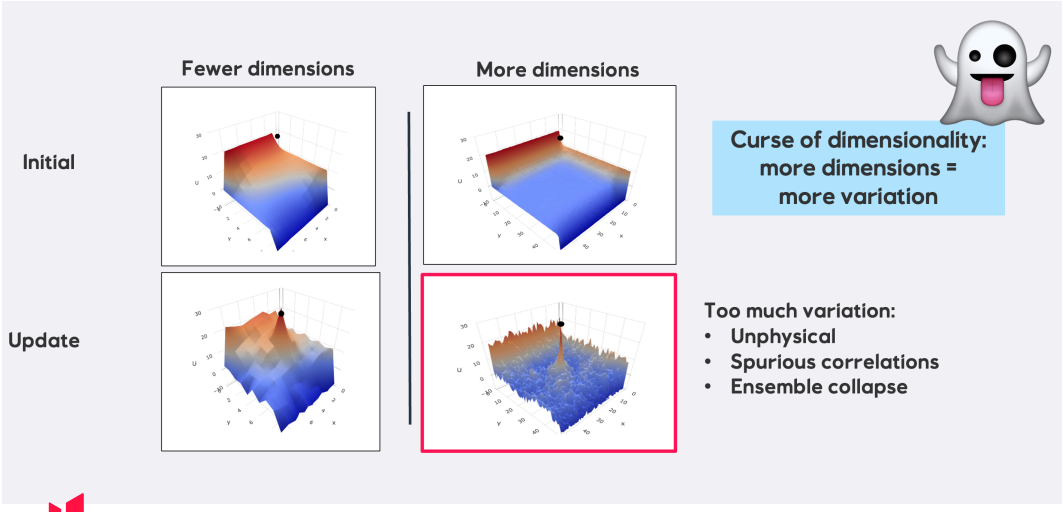
1. Random updates in the mean or single realizations.
2. Overconfidence (loss of variability) due to belief in (random) connections and propagating a Bayesian update through them.

Uh, oh. This sounds familiar

- Spurious correlations: Random updates in mean and single realizations.
- Ensemble collapse: Loss of variability.

From neglecting effects of $p \to \infty$ we have incurred a very real problem.

Spurious correlations and ensemble collapse



EnKF's and convergence

Method scalability

KLD, Structure, & EnIF

Innovations for scalability

Synthetic reservoir application: Sequential EnIF

Scalability also means good statistical properties!

A KLD / Likelihood perspective:

- Minimize

$$D_{KL}(P \parallel Q) = \int \int p(\boldsymbol{u}, \boldsymbol{y}) \log \left(\frac{p(\boldsymbol{u}, \boldsymbol{y})}{q(\boldsymbol{u}, \boldsymbol{y})} \right) d\boldsymbol{u} d\boldsymbol{y},$$

where P is the data-generating-process and $Q(\boldsymbol{\theta})$ is our model.

Scalability also means good statistical properties!

A KLD / Likelihood perspective:

- Minimize

$$D_{KL}(P \parallel Q) = \int \int p(\boldsymbol{u}, \boldsymbol{y}) \log \left(\frac{p(\boldsymbol{u}, \boldsymbol{y})}{q(\boldsymbol{u}, \boldsymbol{y})} \right) d\boldsymbol{u} d\boldsymbol{y},$$

  where P is the data-generating-process and $Q(\boldsymbol{\theta})$ is our model.

- When Q is Gaussian, we have a Kalman-type method.

Scalability also means good statistical properties!

A KLD / Likelihood perspective:

- Minimize

$$D_{KL}(P \parallel Q) = \int \int p(\boldsymbol{u}, \boldsymbol{y}) \log \left(\frac{p(\boldsymbol{u}, \boldsymbol{y})}{q(\boldsymbol{u}, \boldsymbol{y})} \right) d\boldsymbol{u} d\boldsymbol{y},$$

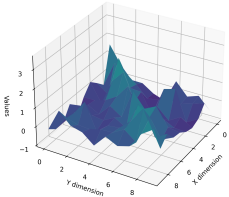
  where P is the data-generating-process and $Q(\boldsymbol{\theta})$ is our model.

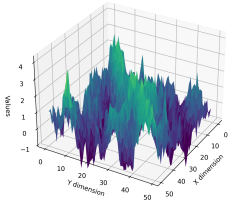
- When Q is Gaussian, we have a Kalman-type method.

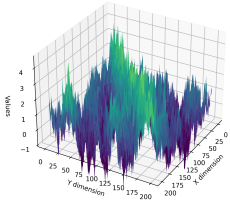
- $\boldsymbol{\theta}$ is typically estimated. **Statistical convergence** matters (not just asymptotic expectations!).

Ensemble Smoother (ES)

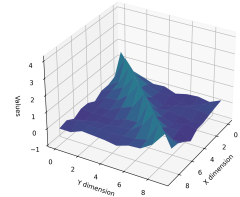
$n = 100$

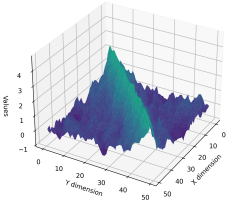


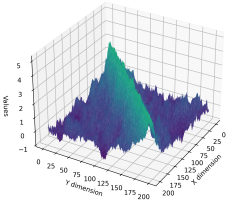




$n = 1000$



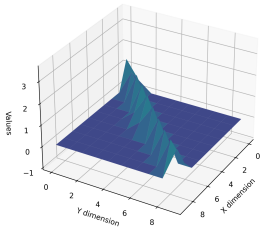


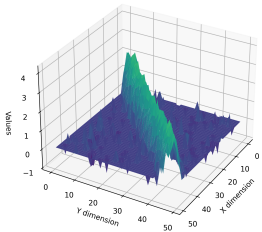


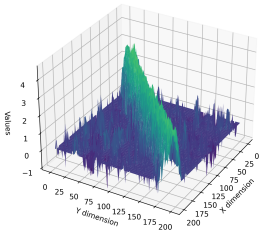
equinor

Berent's biased map of methods

Adaptive Localization $n = 100$

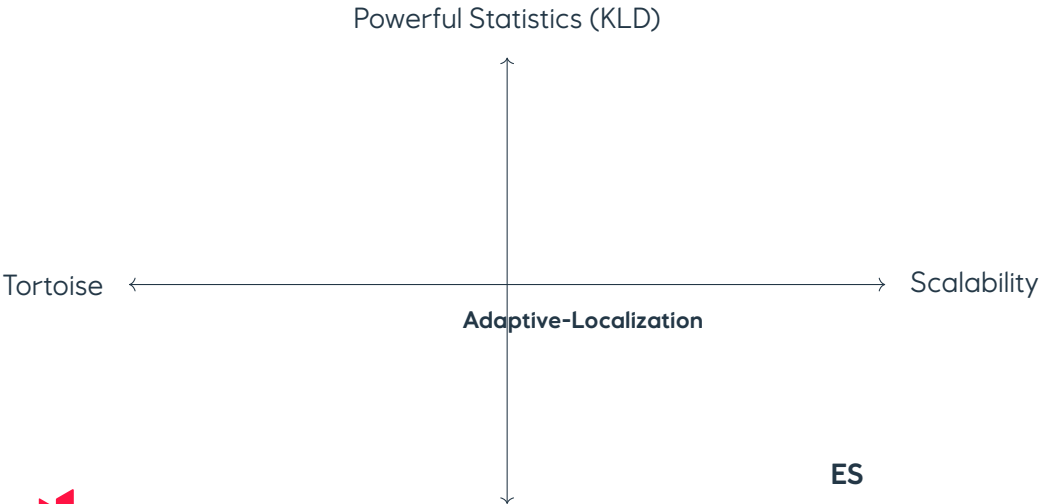






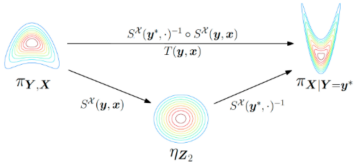
equinor

Berent's biased map of methods



Triangular measure transport: MIT at EnKF 2023

Presented a solution **that solves non-linearity and non-Gaussianity!**

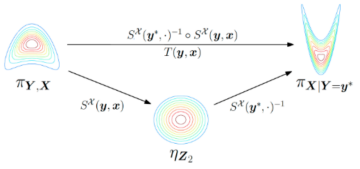


Very important: S is the **Knothe-Rosenblatt rearrangement**

$$S(\mathbf{x}) = \begin{bmatrix} S_1(x_1) \\ S_2(x_1, x_2) \\ \vdots \\ S_d(x_1, x_2, \ldots, x_d) \end{bmatrix}$$

Triangular measure transport: MIT at EnKF 2023

Presented a solution **that solves non-linearity and non-Gaussianity!**



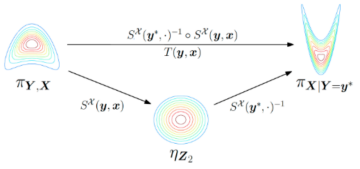
Very important: S is the **Knothe-Rosenblatt rearrangement**

$$S(\mathbf{x}) = \begin{bmatrix} S_1(x_1) \\ S_2(x_1, x_2) \\ \vdots \\ S_d(x_1, x_2, \ldots, x_d) \end{bmatrix}$$

How does it scale computationally?

Triangular measure transport: MIT at EnKF 2023

Presented a solution **that solves non-linearity and non-Gaussianity!**



Very important: S is the **Knothe-Rosenblatt rearrangement**

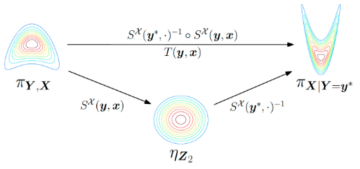
$$S(\mathbf{x}) = \begin{bmatrix} S_1(x_1) \\ S_2(x_1, x_2) \\ \vdots \\ S_d(x_1, x_2, \ldots, x_d) \end{bmatrix}$$

How does it scale computationally?

- Must search for the rearrangement.

Triangular measure transport: MIT at EnKF 2023

Presented a solution **that solves non-linearity and non-Gaussianity!**



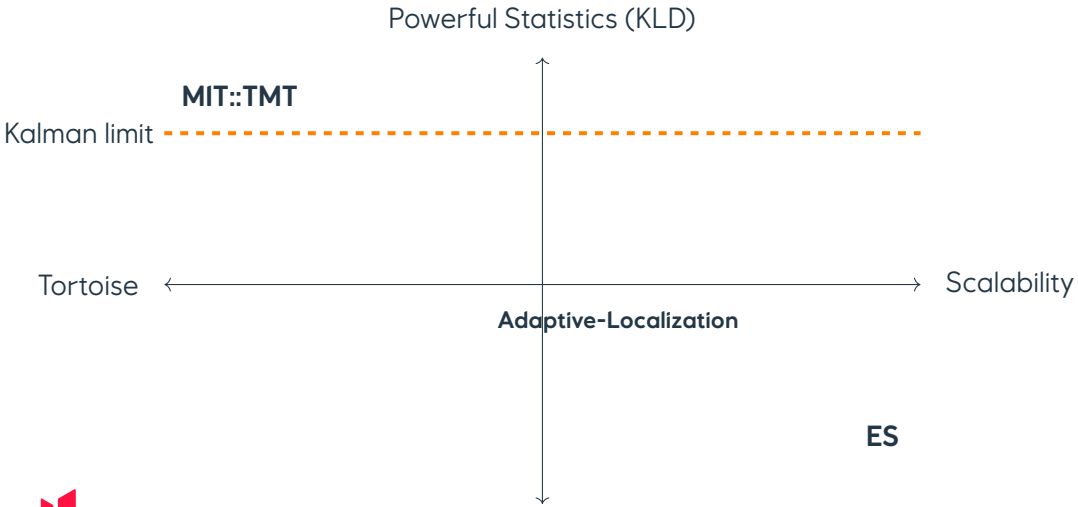
Very important: S is the **Knothe-Rosenblatt rearrangement**

$$S(\mathbf{x}) = \begin{bmatrix} S_1(x_1) \\ S_2(x_1, x_2) \\ \vdots \\ S_d(x_1, x_2, \ldots, x_d) \end{bmatrix}$$

How does it scale computationally?

- Must search for the rearrangement.
- Must learn the degree of non-linearity.

Berent's biased map of methods



The Kalman-type ensemble-based data assimilation

$$\boldsymbol{u}^j_{posterior} = \boldsymbol{u}^j_{prior} + \boldsymbol{K}_{EnKF}(\boldsymbol{d}^j - \boldsymbol{h}(\boldsymbol{u}^j_{prior}))$$

where

$$\boldsymbol{K}_{EnKF} = \boldsymbol{U}\boldsymbol{Y}^\top \left(\boldsymbol{Y}\boldsymbol{Y}^\top + \Sigma_\epsilon\right)^{-1}$$

The Kalman-type ensemble-based data assimilation

$$\boldsymbol{u}^j_{posterior} = \boldsymbol{u}^j_{prior} + \boldsymbol{K}_{EnKF}(\boldsymbol{d}^j - \boldsymbol{h}(\boldsymbol{u}^j_{prior}))$$

where

$$\begin{aligned}\boldsymbol{K}_{EnKF} &= \boldsymbol{U}\boldsymbol{Y}^\top \left(\boldsymbol{Y}\boldsymbol{Y}^\top + \Sigma_\epsilon\right)^{-1} \\ &= \boldsymbol{U}\boldsymbol{U}^\top(\boldsymbol{Y}\boldsymbol{U}^+)^\top \left(\boldsymbol{Y}\boldsymbol{Y}^\top + \Sigma_\epsilon\right)^{-1}\end{aligned}$$

LLS (**noisy**) on map $\boldsymbol{h} : \boldsymbol{u} \mapsto \boldsymbol{y}$

The Kalman-type ensemble-based data assimilation

$$\boldsymbol{u}^j_{posterior} = \boldsymbol{u}^j_{prior} + \boldsymbol{K}_{EnKF}(\boldsymbol{d}^j - \boldsymbol{h}(\boldsymbol{u}^j_{prior}))$$

where

$$\begin{aligned}
\boldsymbol{K}_{EnKF} &= \boldsymbol{U}\boldsymbol{Y}^\top \left(\boldsymbol{Y}\boldsymbol{Y}^\top + \Sigma_\epsilon\right)^{-1} \\
&= \boldsymbol{U}\boldsymbol{U}^\top(\boldsymbol{Y}\boldsymbol{U}^+)^\top \left(\boldsymbol{Y}\boldsymbol{Y}^\top + \Sigma_\epsilon\right)^{-1} \text{ LLS (**noisy**) on map } \boldsymbol{h} : \boldsymbol{u} \mapsto \boldsymbol{y} \\
&\approx \boldsymbol{U}\boldsymbol{D}^\top(\boldsymbol{D}\boldsymbol{D}^\top)^+ \\
&= \boldsymbol{U}\boldsymbol{D}^+ \text{ LLS on map } \boldsymbol{h}^{-1} : \boldsymbol{d} \mapsto \boldsymbol{u}
\end{aligned}$$

The Kalman-type ensemble-based data assimilation

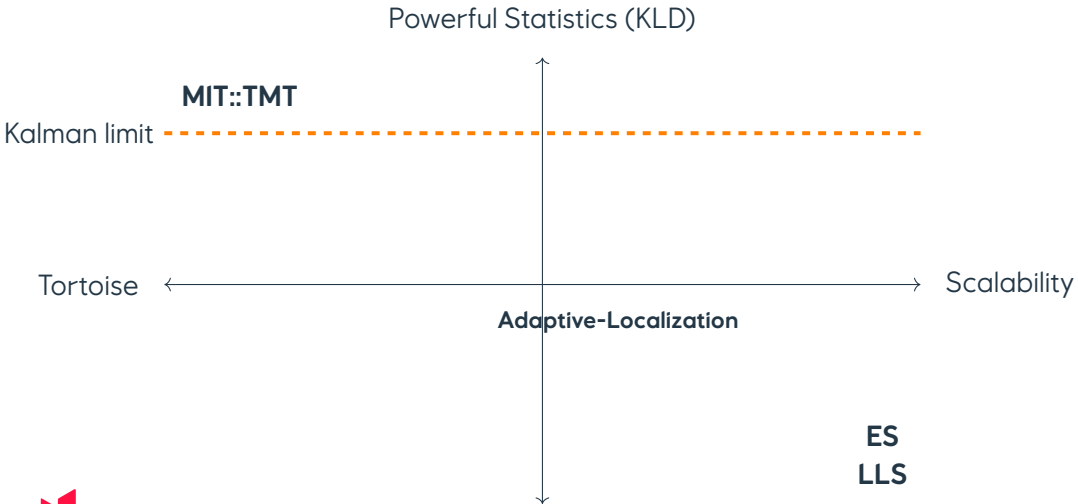
$$\boldsymbol{u}^j_{posterior} = \boldsymbol{u}^j_{prior} + \boldsymbol{K}_{EnKF}(\boldsymbol{d}^j - \boldsymbol{h}(\boldsymbol{u}^j_{prior}))$$

where

$$\begin{aligned}
\boldsymbol{K}_{EnKF} &= \boldsymbol{U}\boldsymbol{Y}^\top \left(\boldsymbol{Y}\boldsymbol{Y}^\top + \Sigma_\epsilon\right)^{-1} \\
&= \boldsymbol{U}\boldsymbol{U}^\top(\boldsymbol{Y}\boldsymbol{U}^+)^\top \left(\boldsymbol{Y}\boldsymbol{Y}^\top + \Sigma_\epsilon\right)^{-1} \text{ LLS (**noisy**) on map } \boldsymbol{h} : \boldsymbol{u} \mapsto \boldsymbol{y} \\
&\approx \boldsymbol{U}\boldsymbol{D}^\top(\boldsymbol{D}\boldsymbol{D}^\top)^+ \\
&= \boldsymbol{U}\boldsymbol{D}^+ \text{ LLS on map } \boldsymbol{h}^{-1} : \boldsymbol{d} \mapsto \boldsymbol{u}
\end{aligned}$$

But what is lost in the wave? Think Gauss-Markov and BLUE

Berent's biased map of methods



A 3-point plan after last years workshop

- ⋆ Regularised linear regression on map $\boldsymbol{h}^{-1} : \boldsymbol{d} \mapsto \boldsymbol{u}$ (Lasso).

A 3-point plan after last years workshop

- ⋆ Regularised linear regression on map $\boldsymbol{h}^{-1} : \boldsymbol{d} \mapsto \boldsymbol{u}$ (Lasso).
- ⋆⋆ The Ensemble Information Filter (EnIF).

A 3-point plan after last years workshop

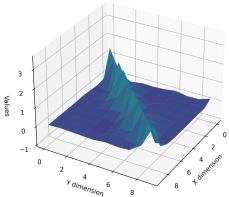
★ Regularised linear regression on map $\boldsymbol{h}^{-1} : \boldsymbol{d} \mapsto \boldsymbol{u}$ (Lasso).

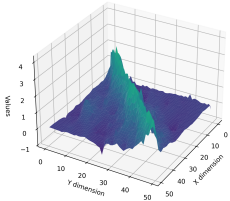
★★ The Ensemble Information Filter (EnIF).

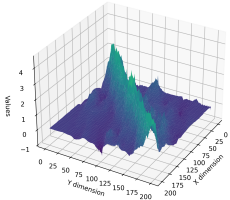
★★★ Information theoretic triangular measure transport (IT-TMT).

EnlF:Cholesky & EnlF::Direct $n = 100$

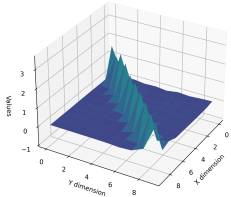
Cholesky

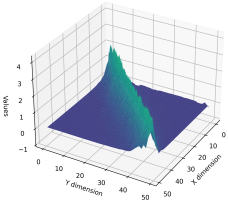


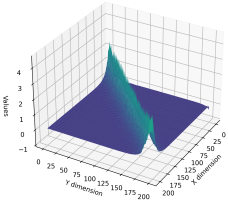




Direct

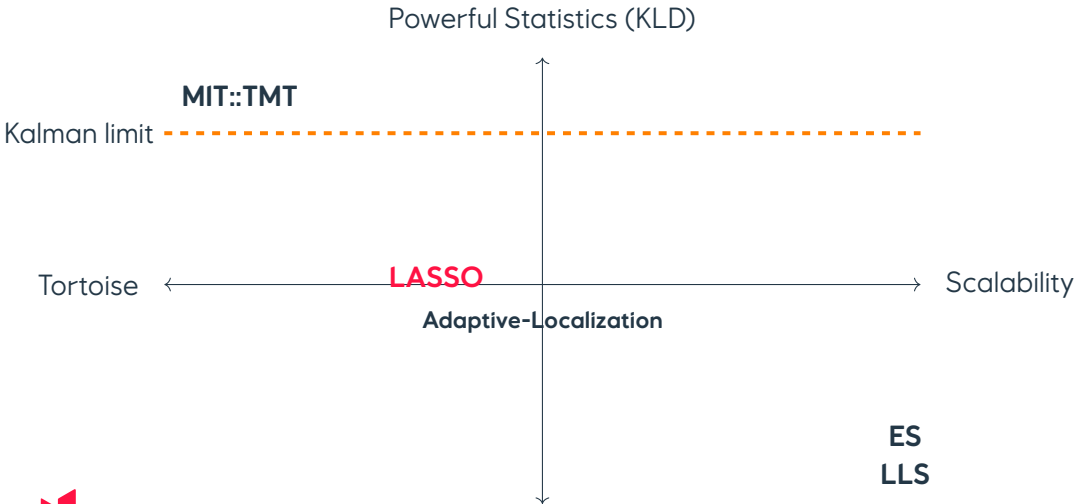




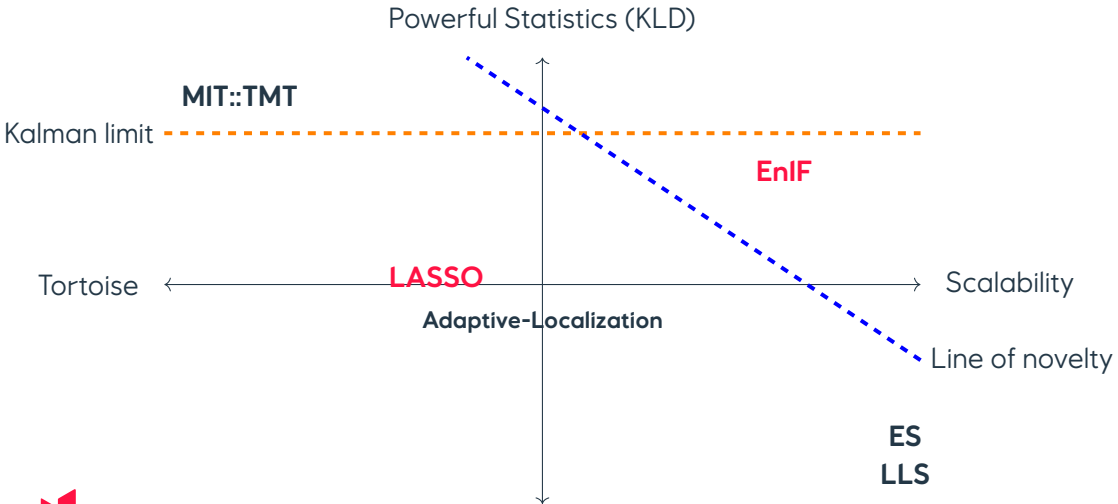


equinor

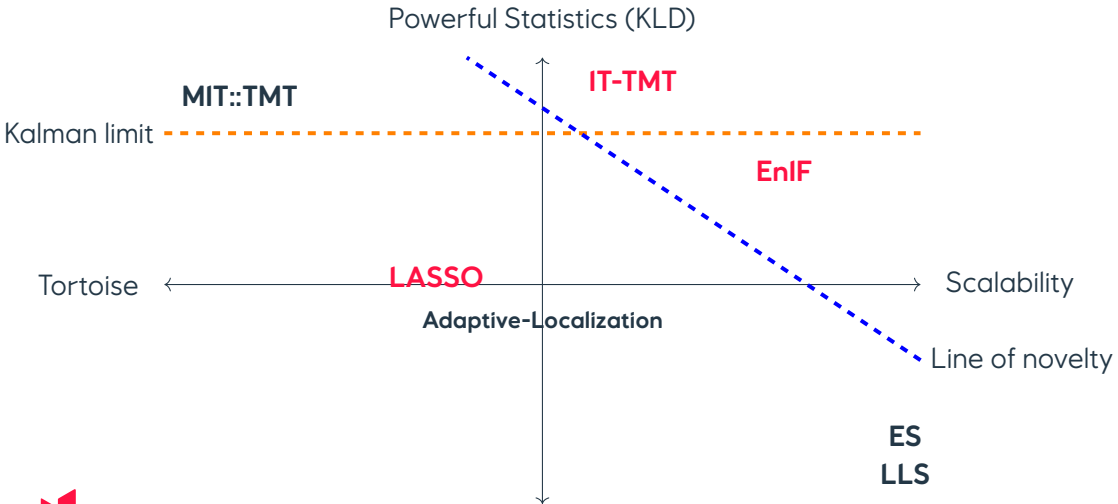
Berent's biased map of methods



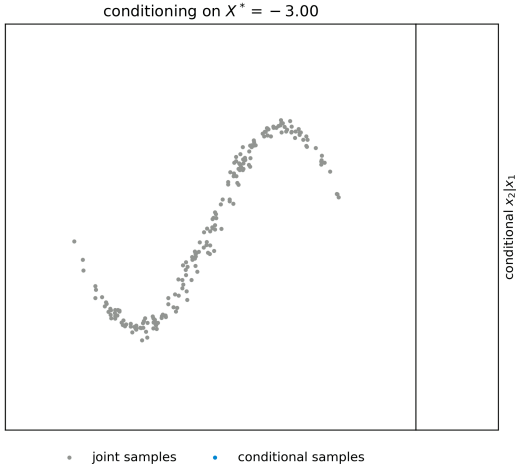
Berent's biased map of methods



Berent's biased map of methods



Information Theoretic (adaptive) Triangular Measure Transport



Recap so far

- The statistical convergence of methods cannot be neglected. Both p and n must be considered. Methods are spatio-temporal.
- When only considering asymptotic expectations, everything seems to be okay. Do not forget variance of statistics.
- The map of methods is my subjective and biased view of things.

EnKF's and convergence

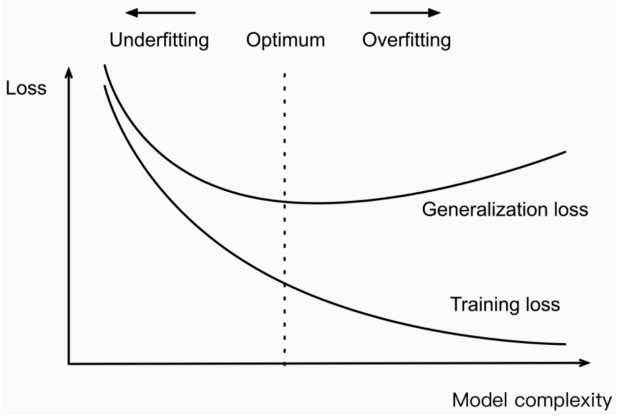
Method scalability

KLD, Structure, & EnIF

Innovations for scalability

Synthetic reservoir application: Sequential EnIF

Theoretical guidelines for KLD optimisation

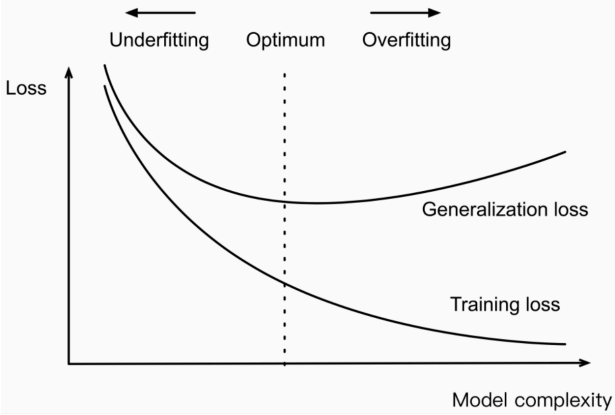


References: Akaike 1974; Takeuchi 1976; Claeskens and Hjort 2008; Hastie, Tibshirani, et al. 2009

Information criteria and tools

- Given model complexity
- Reason about test loss
  - TIC: $\mathrm{tr}(\nabla^2_{\boldsymbol{\theta}} l \, \mathrm{cov}(\hat{\boldsymbol{\theta}}))$
  - AIC: $p = \mathrm{len}(\boldsymbol{\theta})$

Theoretical guidelines for KLD optimisation



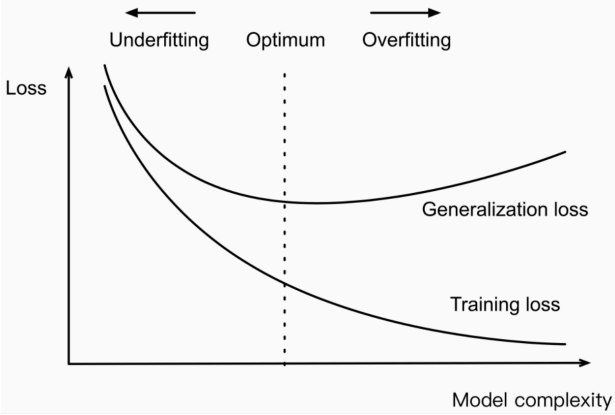
References: Akaike 1974; Takeuchi 1976; Claeskens and Hjort 2008; Hastie, Tibshirani, et al. 2009

Information criteria and tools

- Given model complexity
- Reason about test loss
  - TIC: $\mathrm{tr}(\nabla^2_{\boldsymbol{\theta}} I / \mathrm{cov}(\hat{\boldsymbol{\theta}}))$
  - AIC: $p = \mathrm{len}(\boldsymbol{\theta})$

$\rightarrow$ Reparametrisation, with smaller p.

Theoretical guidelines for KLD optimisation



References: Akaike 1974; Takeuchi 1976; Claeskens and Hjort 2008; Hastie, Tibshirani, et al. 2009

Information criteria and tools

- Given model complexity
- Reason about test loss
  - TIC: $\mathrm{tr}(\nabla^2_{\boldsymbol{\theta}} I / \mathrm{cov}(\hat{\boldsymbol{\theta}}))$
  - AIC: $p = \mathrm{len}(\boldsymbol{\theta})$

$\rightarrow$ Reparametrisation, with smaller p.

$\rightarrow$ Regularization: Trade bias for variance

Searching for a reparametrisation of θ

Do we have structure?

Let L be a differential operator, then the solution to

$$Lu(x) = W(\cdot)$$

is a Gaussian random field and it has the Markov property.

- Heuristically: derivatives (local) create the Markov properties (local).

Searching for a reparametrisation of θ

Do we have structure?

Let L be a differential operator, then the solution to

$$Lu(x) = W(\cdot)$$

is a Gaussian random field and it has the Markov property.

- Heuristically: derivatives (local) create the Markov properties (local).
- More robustly: through power spectrum, covariance operator and its inverse (precision) operator. Rozanov 1977 Lindgren, Håvard Rue, and Lindström 2011

Searching for a reparametrisation of θ

Do we have structure?

Let L be a differential operator, then the solution to

$$Lu(x) = W(\cdot)$$

is a Gaussian random field and it has the Markov property.

- Heuristically: derivatives (local) create the Markov properties (local).
- More robustly: through power spectrum, covariance operator and its inverse (precision) operator. Rozanov 1977 Lindgren, Håvard Rue, and Lindström 2011
- SPDE approach: approximate non-Markov field (solutions) by Markov fields.

Searching for a reparametrisation of θ

Do we have structure?

Let L be a differential operator, then the solution to

$$Lu(x) = W(\cdot)$$

is a Gaussian random field and it has the Markov property.

- Heuristically: derivatives (local) create the Markov properties (local).
- More robustly: through power spectrum, covariance operator and its inverse (precision) operator. Rozanov 1977 Lindgren, Håvard Rue, and Lindström 2011
- SPDE approach: approximate non-Markov field (solutions) by Markov fields.
- Computationally important: $\Lambda_u = \Sigma_u^{-1}$ is **sparse** for GMRF.

Searching for a reparametrisation of θ

Do we have structure?

Let L be a differential operator, then the solution to

$$Lu(x) = W(\cdot)$$

is a Gaussian random field and it has the Markov property.

- Heuristically: derivatives (local) create the Markov properties (local).
- More robustly: through power spectrum, covariance operator and its inverse (precision) operator. Rozanov 1977 Lindgren, Håvard Rue, and Lindström 2011
- SPDE approach: approximate non-Markov field (solutions) by Markov fields.
- Computationally important: $\Lambda_u = \Sigma_u^{-1}$ is **sparse** for GMRF.
- If Λ_u is sparse, then $p = len(\boldsymbol{\theta})$ is *much* smaller than in the covariance parametrisation. Training bias in KLD is positively monotone in p.

Easier with a discretised example

A stochastic wave equation

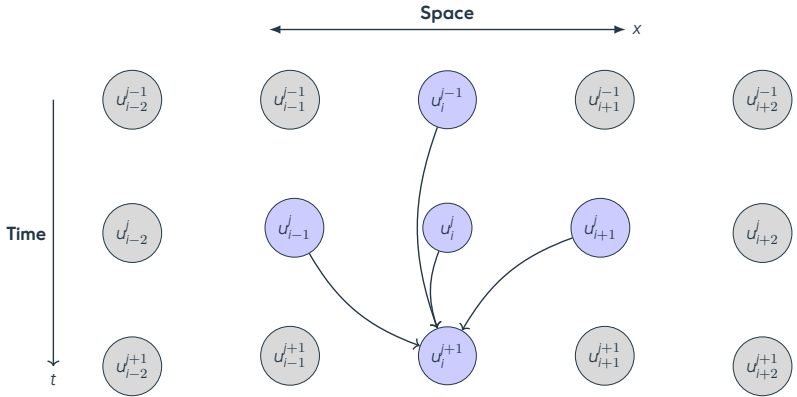
$$d^2u_t(\boldsymbol{x}) = c\,\mathrm{div}\,\nabla\, u_t(\boldsymbol{x})dt^2 + \sigma dW_t$$

suggests a simple finite difference discretization, in the 1-d case:

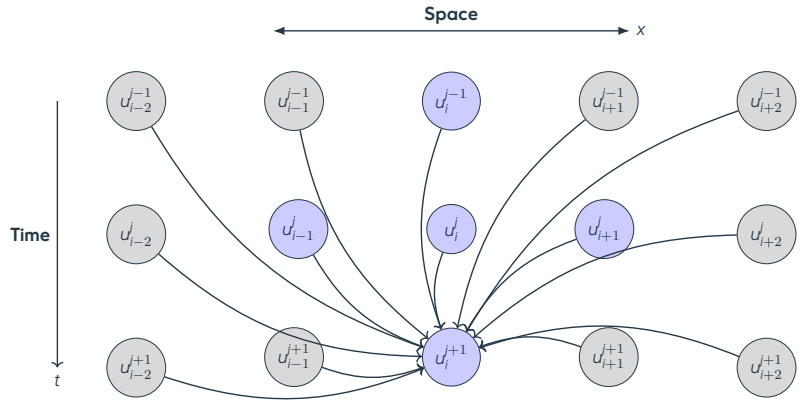
$$u_i^{j+1} = 2u_i^j - u_i^{j-1} + \frac{c^2\Delta t^2}{\Delta x^2}(u_{i+1}^j - 2u_i^j + u_{i-1}^j) + \sigma\sqrt{dt}Z,\ Z \sim \mathcal{N}(0,1)$$

So u_i^{j+1} is *only* a function of u_i^j, u_i^{j-1}, u_{i-1}^j, and u_{i+1}^j. **Not** all of $\boldsymbol{u}$.

Discretization incur spatio-temporal conditional independence



What are we doing with EnKFs?



What are we doing with EnKFs?

- The sample covariance correspond to estimation w.r.t. a complete graph.

What are we doing with EnKFs?

- The sample covariance correspond to estimation w.r.t. a complete graph.
- So there is no *local* solution, every displacement is a function of the global state.

What are we doing with EnKFs?

- The sample covariance correspond to estimation w.r.t. a complete graph.
- So there is no *local* solution, every displacement is a function of the global state.
- We also consider teleportation of information.

What are we doing with EnKFs?

- The sample covariance correspond to estimation w.r.t. a complete graph.
- So there is no *local* solution, every displacement is a function of the global state.
- We also consider teleportation of information.
- We try to learn the physics from scratch. We need a lot of data.

How to exploit this?

- Conditional (in)dependence depends on discretisation scheme.
  - Smoothing, filtering and parameter estimation are different.

How to exploit this?

- Conditional (in)dependence depends on discretisation scheme.
  - Smoothing, filtering and parameter estimation are different.
- **Smoothing**: *natural* graph from (s)pde.

How to exploit this?

- Conditional (in)dependence depends on discretisation scheme.
  - Smoothing, filtering and parameter estimation are different.
- **Smoothing**: *natural* graph from (s)pde.
- **Filtering**: A complete graph! No exact conditional independence.

How to exploit this?

- Conditional (in)dependence depends on discretisation scheme.
  - Smoothing, filtering and parameter estimation are different.
- **Smoothing**: *natural* graph from (s)pde.
- **Filtering**: A complete graph! No exact conditional independence.
- **Parameter estimation**: Sampling from independence or variograms.

How to exploit this?

- Conditional (in)dependence depends on discretisation scheme.
  - Smoothing, filtering and parameter estimation are different.
- **Smoothing**: *natural* graph from (s)pde.
- **Filtering**: A complete graph! No exact conditional independence.
- **Parameter estimation**: Sampling from independence or variograms.
- We only require a (parsimonious!) approximation.

Where to use conditional independence information?

How EnKF's work

Let $\boldsymbol{u}$ and $\boldsymbol{y}$ be jointly Gaussian. Then, a sample $(\boldsymbol{u}_i, \boldsymbol{y}_i)$ is mapped to a sample from the conditional $p(\boldsymbol{u}|\boldsymbol{y})$, having observed $\boldsymbol{y}^*$, via the formula

$$\boldsymbol{u}_i + \boldsymbol{K}(\boldsymbol{y}^* - \boldsymbol{y}_i) \sim p(\boldsymbol{u}|\boldsymbol{y}^*).$$

where the "Kalman gain" $\boldsymbol{K}$ is defined as $\boldsymbol{K} = \boldsymbol{\Sigma}_{\boldsymbol{uy}}\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}$, which is estimated.

Where to use conditional independence information?

How EnKF's work

Let $\boldsymbol{u}$ and $\boldsymbol{y}$ be jointly Gaussian. Then, a sample $(\boldsymbol{u}_i, \boldsymbol{y}_i)$ is mapped to a sample from the conditional $p(\boldsymbol{u}|\boldsymbol{y})$, having observed $\boldsymbol{y}^*$, via the formula

$$\boldsymbol{u}_i + \boldsymbol{K}(\boldsymbol{y}^* - \boldsymbol{y}_i) \sim p(\boldsymbol{u}|\boldsymbol{y}^*).$$

where the "Kalman gain" $\boldsymbol{K}$ is defined as $\boldsymbol{K} = \boldsymbol{\Sigma}_{\boldsymbol{uy}}\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}$, which is estimated.

- $\boldsymbol{K} = \boldsymbol{\Lambda}_{t|t}^{-1}\boldsymbol{H}^\top \boldsymbol{\Lambda}_{\boldsymbol{r}_t}$ using Woodbury (surprise) Moore and Anderson 1979.

Where to use conditional independence information?

How EnKF's work

Let $\boldsymbol{u}$ and $\boldsymbol{y}$ be jointly Gaussian. Then, a sample $(\boldsymbol{u}_i, \boldsymbol{y}_i)$ is mapped to a sample from the conditional $p(\boldsymbol{u}|\boldsymbol{y})$, having observed $\boldsymbol{y}^*$, via the formula

$$\boldsymbol{u}_i + \boldsymbol{K}(\boldsymbol{y}^* - \boldsymbol{y}_i) \sim p(\boldsymbol{u}|\boldsymbol{y}^*).$$

where the "Kalman gain" $\boldsymbol{K}$ is defined as $\boldsymbol{K} = \boldsymbol{\Sigma}_{\boldsymbol{uy}}\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}$, which is estimated.

- $\boldsymbol{K} = \boldsymbol{\Lambda}_{t|t}^{-1}\boldsymbol{H}^\top \boldsymbol{\Lambda}_{\boldsymbol{r}_t}$ using Woodbury (surprise) Moore and Anderson 1979.
- $\boldsymbol{\Lambda}_{t|t} = \boldsymbol{\Lambda}_{t|t-1} + \boldsymbol{H}^\top \boldsymbol{\Lambda}_{\boldsymbol{r}_t}\boldsymbol{H}$ is **dense** if $\boldsymbol{H}$ is dense.
  - Havard Rue and Held 2005 $\boldsymbol{H}$ is dense for geostatistcs. This won't work.
  - IF equations $\boldsymbol{u}_{t|t}^{(i)} = \Lambda_{t|t}^{-1}\boldsymbol{\eta}_{t|t}^{(i)}$ computationally infeasible.

Where to use conditional independence information?

How EnKF's work

Let $\boldsymbol{u}$ and $\boldsymbol{y}$ be jointly Gaussian. Then, a sample $(\boldsymbol{u}_i, \boldsymbol{y}_i)$ is mapped to a sample from the conditional $p(\boldsymbol{u}|\boldsymbol{y})$, having observed $\boldsymbol{y}^*$, via the formula

$$\boldsymbol{u}_i + \boldsymbol{K}(\boldsymbol{y}^* - \boldsymbol{y}_i) \sim p(\boldsymbol{u}|\boldsymbol{y}^*).$$

where the "Kalman gain" $\boldsymbol{K}$ is defined as $\boldsymbol{K} = \boldsymbol{\Sigma}_{\boldsymbol{uy}}\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}$, which is estimated.

- $\boldsymbol{K} = \boldsymbol{\Lambda}_{t|t}^{-1}\boldsymbol{H}^\top \boldsymbol{\Lambda}_{\boldsymbol{r}_t}$ using Woodbury (surprise) Moore and Anderson 1979.
- $\boldsymbol{\Lambda}_{t|t} = \boldsymbol{\Lambda}_{t|t-1} + \boldsymbol{H}^\top \boldsymbol{\Lambda}_{\boldsymbol{r}_t}\boldsymbol{H}$ is **dense** if $\boldsymbol{H}$ is dense.
  - Havard Rue and Held 2005 $\boldsymbol{H}$ is dense for geostatistcs. This won't work.
  - IF equations $\boldsymbol{u}_{t|t}^{(i)} = \Lambda_{t|t}^{-1}\boldsymbol{\eta}_{t|t}^{(i)}$ computationally infeasible.
- But $\boldsymbol{H}$ is estimated. Remember KLD. *Choose regularisation to obtain sparse* $\boldsymbol{H}$.

The Ensemble Information Filter

Sample from belief

$$\boldsymbol{u}_{t-1|t-1}^{(i)} \sim p(\boldsymbol{u}_{t-1|t-1})\; i = 1, \ldots, n$$

Predict

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

Estimate

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

And $\hat{\boldsymbol{H}}$ as a sparse linear map

Update realizations *and* precision

$$\boldsymbol{\eta}_{t|t-1}^{(i)} = \hat{\boldsymbol{\Lambda}}_{t|t-1}\boldsymbol{u}_{t|t-1}^{(i)}$$
$$\hat{\boldsymbol{\eta}}_{t|t} = \hat{\boldsymbol{\eta}}_{t|t-1} + \hat{\boldsymbol{H}}^{\top}\boldsymbol{\Lambda}_{\boldsymbol{r}}(\boldsymbol{y}_t - \boldsymbol{r}^{(i)})$$
$$\hat{\boldsymbol{\Lambda}}_{t|t} = \hat{\boldsymbol{\Lambda}}_{t|t-1} + \hat{\boldsymbol{H}}^{\top}\boldsymbol{\Lambda}_{\boldsymbol{r}}\hat{\boldsymbol{H}}$$

Bring realizations back to original space

$$\boldsymbol{u}_{t|t}^{(i)} = \hat{\boldsymbol{\Lambda}}_{t|t}^{-1}\boldsymbol{\eta}_{t|t}^{(i)}$$

Recap on EnIF

- KLD warrants the use of structure and regularisation.
- Structure can come from the model, e.g. (S)PDE.
- Derivatives (local) leads to Markov properties (local), perhaps approximately.
- EnIF is a reparametrisation of the Gaussian update in EnKF. Regularised and encoding (Markov) structure.
- Sparsity is a necessity for computation.

EnKF's and convergence

Method scalability

KLD, Structure, & EnIF

Innovations for scalability

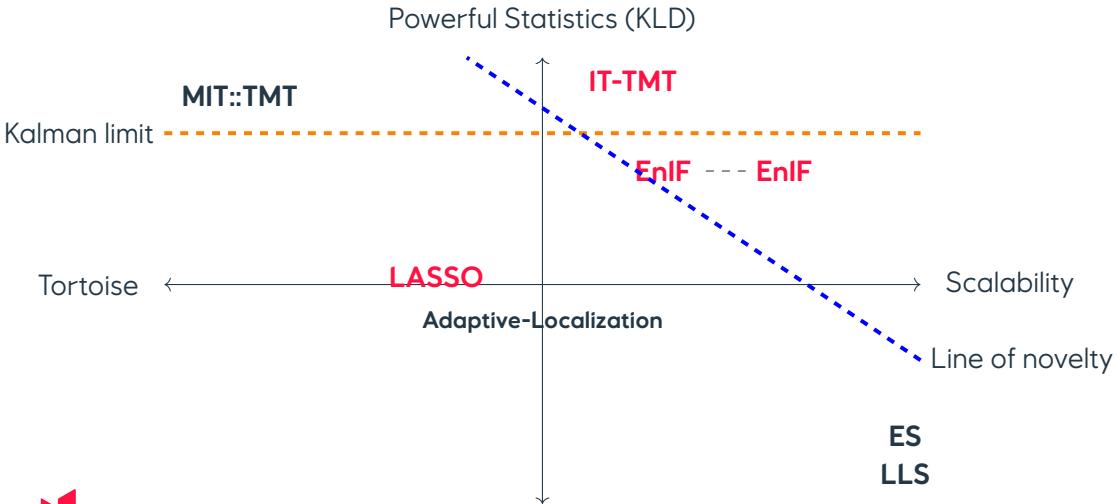
Synthetic reservoir application: Sequential EnIF

An Equinor history matching problem

- About 10 million parameters p.
  - About 100-1000 static parameters
  - Some surfaces of size about 300×300
  - Some 3D fields of size about $100 \times 100 \times 100$
- Ensemble size n about 100-200.
- Number of responses m about 100-1000, more if seismic is included.

And how to **understand the update** for a domain expert?

EnIF using off-the-shelf libraries vs. with innovations

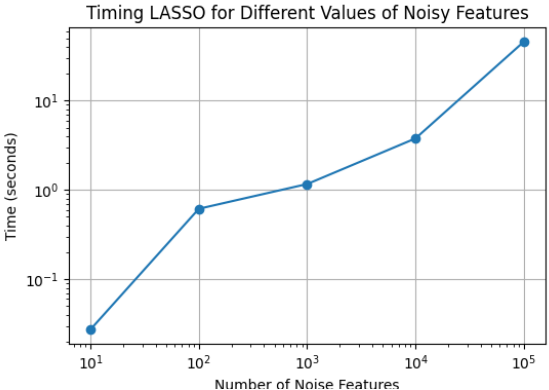


First hurdle: learning H

First thought: L1/LASSO regression Tibshirani 1996

- For sparsity the go-to solution.
- It is efficient, but...

First hurdle: learning H



First thought: L1/LASSO regression Tibshirani 1996

- For sparsity the go-to solution.
- It is efficient, but...
- Not *that* efficient.

Boosting out the solution path

Algorithm Boosting Monotone-LASSO

1: Initialize $\hat{\boldsymbol{\beta}}_0 = \mathbf{0}$
2: **while** $\mathrm{mse}_{cv-n}(\boldsymbol{X}, \boldsymbol{y}; \hat{\boldsymbol{\beta}}_k) > \mathrm{mse}_{cv-n}(\boldsymbol{X}, \boldsymbol{y}; \hat{\boldsymbol{\beta}}_{k+1})$ **do**
3: Calculate all 1d linear regressions
4: Select β_j as the one reducing training mse the most
5: $\hat{\boldsymbol{\beta}}_{k+1,j} += \epsilon\beta_j$
6: **end while**
7: **return** $\hat{\boldsymbol{\beta}}$

LASSO, LARS, FS-ϵ and Boosting relations.
Hastie, Taylor, et al. 2007

Boosting out the solution path

Algorithm Boosting Monotone-LASSO

1: Initialize $\hat{\boldsymbol{\beta}}_0 = \mathbf{0}$
2: **while** $\mathrm{mse}_{\mathrm{cv}-n}(\boldsymbol{X}, \boldsymbol{y}; \hat{\boldsymbol{\beta}}_k) > \mathrm{mse}_{\mathrm{cv}-n}(\boldsymbol{X}, \boldsymbol{y}; \hat{\boldsymbol{\beta}}_{k+1})$ **do**
3: Calculate all 1d linear regressions
4: Select β_j as the one reducing training mse the most
5: $\hat{\boldsymbol{\beta}}_{k+1,j} += \epsilon\beta_j$
6: **end while**
7: **return** $\hat{\boldsymbol{\beta}}$

- cv-n implies *n* times more computation. Unless...

LASSO, LARS, FS-ϵ and Boosting relations.
Hastie, Taylor, et al. 2007

Boosting out the solution path

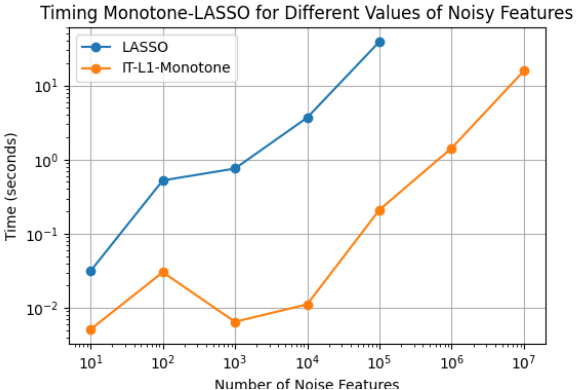
Algorithm Boosting Monotone-LASSO

1: Initialize $\hat{\boldsymbol{\beta}}_0 = \mathbf{0}$
2: **while** $\mathrm{mse}_{\mathrm{cv}-n}(\boldsymbol{X}, \boldsymbol{y}; \hat{\boldsymbol{\beta}}_k) > \mathrm{mse}_{\mathrm{cv}-n}(\boldsymbol{X}, \boldsymbol{y}; \hat{\boldsymbol{\beta}}_{k+1})$ **do**
3: Calculate all 1d linear regressions
4: Select β_j as the one reducing training mse the most
5: $\hat{\boldsymbol{\beta}}_{k+1,j} += \epsilon\beta_j$
6: **end while**
7: **return** $\hat{\boldsymbol{\beta}}$

LASSO, LARS, FS-ϵ and Boosting relations. Hastie, Taylor, et al. 2007

- cv-n implies *n* times more computation. Unless...
- $\hat{\theta}_{-i} \to_n \hat{\theta} - n^{-1} IF(y_i, \boldsymbol{x}_i)$, where the influence *IF* is found using the asymptotic properties of $\hat{\beta}_j$ as an M-estimators. cv-n and TIC relation through IF. Claeskens and Hjort 2008

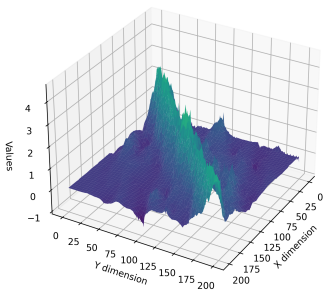
Information theoretic stopping criterion



Information criteria and tools

- Boost out monotone-LASSO solution paths.
- With information theoretic stopping criterion!

Second hurdle: Graph optimisation and fill-in



Given a permutation optimised for a sparse Cholesky factoer

$$\boldsymbol{L}_*\boldsymbol{L}_*^\top = \boldsymbol{P}_*^\top \boldsymbol{\Lambda}_{\boldsymbol{u}} \boldsymbol{P}_*$$

We can find a relation to the linear triangular transport map $\boldsymbol{C}(\pi^*)$

$$\boldsymbol{\Lambda}_{\boldsymbol{u}} = \boldsymbol{P}_r \boldsymbol{P}_* \boldsymbol{C}(\pi^*)^\top \boldsymbol{C}(\pi^*) \boldsymbol{P}_*^\top \boldsymbol{P}_r.$$

where $\boldsymbol{P}_r$ is the reverse permutation matrix.

Learn $\boldsymbol{C}(\pi^*)$ row-by-row like in TMT, with the same sparsity as $\boldsymbol{L}_*$ (but reversed and transposed).

Fill-in reducing algorithms

- Finding optimal permutation is NP-hard.

Fill-in reducing optimised permutation

Fill-in reducing algorithms

- Finding optimal permutation is NP-hard.
- AMD and METIS etc. good for 2D but struggle with 3D and 4D! Amestoy, Davis, and Duff 2004

Fill-in reducing optimised permutation

Fill-in reducing algorithms

- Finding optimal permutation is NP-hard.
- AMD and METIS etc. good for 2D but struggle with 3D and 4D! Amestoy, Davis, and Duff 2004
  - AMD on graph from 3D cube with 800000 elements. 20 minutes to optimise, with about 130 million elements in Cholesky factor. compared to about 2-3 million in precision matrix.

Fill-in reducing optimised permutation

Fill-in reducing algorithms

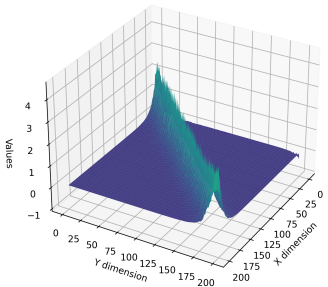
- Finding optimal permutation is NP-hard.
- AMD and METIS etc. good for 2D but struggle with 3D and 4D! Amestoy, Davis, and Duff 2004
  - AMD on graph from 3D cube with 800000 elements. 20 minutes to optimise, with about 130 million elements in Cholesky factor. compared to about 2-3 million in precision matrix.
  - The dense Cholesky would have 320×10^9.

Fill-in reducing optimised permutation

Fill-in reducing algorithms

- Finding optimal permutation is NP-hard.
- AMD and METIS etc. good for 2D but struggle with 3D and 4D! Amestoy, Davis, and Duff 2004
  - AMD on graph from 3D cube with 800000 elements. 20 minutes to optimise, with about 130 million elements in Cholesky factor. compared to about 2-3 million in precision matrix.
  - The dense Cholesky would have 320×10^9.
- This is a limitation also for triangular measure transport.

Optimise Λ directly column-by-column



Searched far and wide for an algorithm estimation Λ conditioned on $\mathcal{G}$.

Surprisingly little literature, and what exist often does not scale (e.g. ESL Alg. 17.1, the basis of much more well known Graphical-Lasso Alg. 17.2. Hastie, Tibshirani, et al. 2009

Reverted to `GraphSPME` library.
Benefits: very fast and no fill-in.
disadvantages: Does not optimise the likelihood directly, symmetry, and condition number.

Third hurdle: Map the realisations back

The final step of EnIF is to map from "canonical" realisations to physical ones.

$$\boldsymbol{u}_{t|t} = \boldsymbol{\Lambda}_{t|t}^{-1} \boldsymbol{\nu}_{t|t}$$

The natural solver is the (permutation optimised) sparse Cholesky solver.

Third hurdle: Map the realisations back

The final step of EnIF is to map from "canonical" realisations to physical ones.

$$\boldsymbol{u}_{t|t} = \boldsymbol{\Lambda}_{t|t}^{-1} \boldsymbol{\nu}_{t|t}$$

The natural solver is the (permutation optimised) sparse Cholesky solver.

From graph-estimation we know **this will fail** on large 3D problems.

Third hurdle: Map the realisations back

The final step of EnIF is to map from "canonical" realisations to physical ones.

$$\boldsymbol{u}_{t|t} = \boldsymbol{\Lambda}_{t|t}^{-1}\boldsymbol{\nu}_{t|t}$$

The natural solver is the (permutation optimised) sparse Cholesky solver.

From graph-estimation we know **this will fail** on large 3D problems.

An **iterative solver** is the solution: $\Lambda_{t|t}$ is SPD and sparse, thus **Conjugate gradient.**

Neighbourhood inversion

A "localization" effect from assuming Markov properties:

- Covariance effect through path (think AR-p) exhibits exponential decay in steps.

Neighbourhood inversion

A "localization" effect from assuming Markov properties:

- Covariance effect through path (think AR-p) exhibits exponential decay in steps.

We may pick out observations that are updated directly from the learnt $\hat{\boldsymbol{H}}$.

- Choose a neighbourhood-propagation, say k and update all observations within k neighbours "distance" from the direct updates.

Neighbourhood inversion

A "localization" effect from assuming Markov properties:

- Covariance effect through path (think AR-p) exhibits exponential decay in steps.

We may pick out observations that are updated directly from the learnt $\hat{\boldsymbol{H}}$.

- Choose a neighbourhood-propagation, say k and update all observations within k neighbours "distance" from the direct updates.

The system of equations $\boldsymbol{\Lambda u} = \boldsymbol{\eta}$ in block-form:

$$\begin{bmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \end{bmatrix}$$

Given $\boldsymbol{u}_1$ is known from a previous computation, we can update $\boldsymbol{u}_2$ as follows:

$$\boldsymbol{u}_2 = \boldsymbol{\Lambda}_{22}^{-1} \left(\boldsymbol{\eta}_2 - \boldsymbol{\Lambda}_{21} \boldsymbol{u}_1 \right)$$

Potentially a much smaller system.

EnKF's and convergence

Method scalability

KLD, Structure, & EnIF

Innovations for scalability

Synthetic reservoir application: Sequential EnIF

The Synthetic case

- About 8.5 million parameters p.
  - About 72 static parameters
  - $4 \times$ 2D surfaces of size 123921
  - $9 \times$ 3D fields of size 886512
- Ensemble size $n = 100$.
- Number of responses $m = 117$, more if seismic is included.

Full EnIF update in ERT takes about 20 minutes.

The Synthetic case

- About 8.5 million parameters p.
  - About 72 static parameters
  - $4 \times$ 2D surfaces of size 123921
  - $9 \times$ 3D fields of size 886512
- Ensemble size $n = 100$.
- Number of responses $m = 117$, more if seismic is included.

Full EnIF update in ERT takes about 20 minutes.

Question:

How to understand or inspect the update for a domain expert?

Problem: Understanding the update

Joint assimilation is fast! But yields little understanding

- The engineer can know more than learned from data.
- KLD is *not* the objective of the engineer. Understanding, tuning, and a story?

Algorithm Sequential EnIF

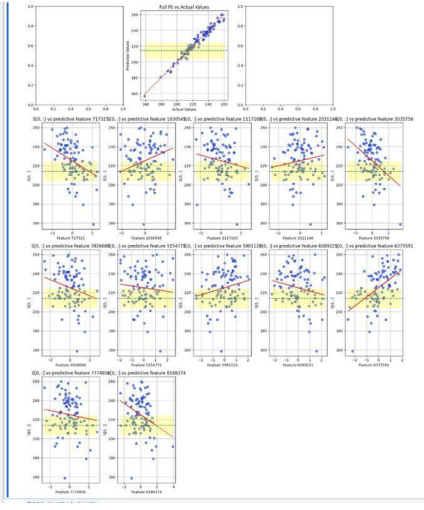
1: Sample ensemble, estimate prior precision
2: **for** each batch $\boldsymbol{d}_b$ of observations **do**
3: Fit the sparse linear sub-map $\hat{\boldsymbol{H}}_b$
4: **for** each observation k in batch b **do**
5: Inspect $\hat{\boldsymbol{H}}_{b(k)}$, tweak, understand effect, approve and a story
6: **end for**
7: Assimilate $\boldsymbol{d}_b$ using the (additive) EnIF update
8: **end for**

equinor

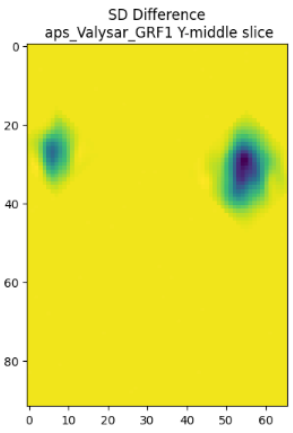
Demo in Notebook

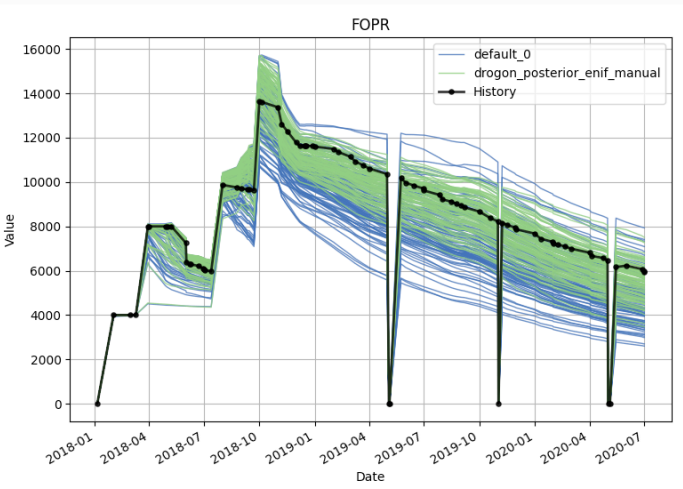
Go to jupyter notebook

If the demo did not work...



Update and Match





Summary of talk

- Statistial convergence!

Summary of talk

- Statistial convergence!
- Structure and regularisation!

Summary of talk

- Statistial convergence!
- Structure and regularisation!
- EnIF incorporates the above

Summary of talk

- Statistial convergence!
- Structure and regularisation!
- EnIF incorporates the above
- Computational innovations for extra scalability

Summary of talk

- Statistial convergence!
- Structure and regularisation!
- EnIF incorporates the above
- Computational innovations for extra scalability
- Care about users: Sequential assimilation for understanding

Thank you!

equinor

Bibliography I

- Akaike, Hirotugu (1974). "A new look at the statistical model identification". In: *IEEE transactions on automatic control* 19.6, pp. 716–723.
- Amestoy, Patrick R, Timothy A Davis, and Iain S Duff (2004). "Algorithm 837: AMD, an approximate minimum degree ordering algorithm". In: *ACM Transactions on Mathematical Software (TOMS)* 30.3, pp. 381–388.
- Claeskens, Gerda and Nils Lid Hjort (2008). "Model selection and model averaging". In: *Cambridge books*.
- Hastie, Trevor, Jonathan Taylor, et al. (2007). "Forward stagewise regression and the monotone lasso". In.
- Hastie, Trevor, Robert Tibshirani, et al. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer.

Bibliography II

- Lindgren, Finn, Håvard Rue, and Johan Lindström (2011). "An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73.4, pp. 423–498.
- Moore, John Barratt and B Anderson (1979). *Optimal filtering*. Prentice-Hall New York.
- Rozanov, Ju A (1977). "Markov random fields and stochastic partial differential equations". In: *Mathematics of the USSR-Sbornik* 32.4, p. 515.
- Rue, Havard and Leonhard Held (2005). *Gaussian Markov random fields: theory and applications*. Chapman and Hall/CRC.

Bibliography III

- Takeuchi, Kei (1976). "Distribution of Information Statistics and Validity Criteria of Models". In: *Mathematical Science* 153, pp. 12–18.
- Tibshirani, Robert (1996). "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 58.1, pp. 267–288.