# Estimation of model evidence and stacking with ensemble methods

A review with an overview and perhaps a survey

*Andreas S. Stordal*

# Bayesian Model Averaging and Stacking in Data Assimilation

- Bayesian framework for estimating an unknown quantity $X \sim p(x)$ given a measurement $Y \sim p(y|x)$ within the framework of several different models/scenarios/methods.

# Bayesian Model Averaging and Stacking in Data Assimilation

- Bayesian framework for estimating an unknown quantity $X \sim p(x)$ given a measurement $Y \sim p(y|x)$ within the framework of several different models/scenarios/methods.
- Different geological scenarios, different climate models etc.

# Bayesian Model Averaging and Stacking in Data Assimilation

- Bayesian framework for estimating an unknown quantity $X \sim p(x)$ given a measurement $Y \sim p(y|x)$ within the framework of several different models/scenarios/methods.
- Different geological scenarios, different climate models etc.
- For each model, $M_1, M_2, \ldots M_m$, we estimate

$$p(x|y, M_i) = p(x|M_i)p(y|x, M_i)\, C_i^{-1}, \; i = 1, \ldots, m$$

where $C_i = \int_X p(y|x, M_i)p(x|M_i)\, dx$ is the normalizing constant or model evidence.

# Bayesian Model Averaging

- The posterior distribution is

$$p(x|y) = \sum_{i=1}^{k} p(x|y, M_i)p(M_i|y),$$

$$p(M_i|y) = \frac{p(y|M_i)p(M_i)}{\sum_{j=1}^{k} p(y|M_j)p(M_j)}.$$

# Bayesian Model Averaging

- The posterior distribution is

$$p(x|y) = \sum_{i=1}^{k} p(x|y, M_i)p(M_i|y),$$

$$p(M_i|y) = \frac{p(y|M_i)p(M_i)}{\sum_{j=1}^{k} p(y|M_j)p(M_j)}.$$

- $p(M_i)$ is prior probability for model $i$ (known)

# Bayesian Model Averaging

- The posterior distribution is

$$p(x|y) = \sum_{i=1}^{k} p(x|y, M_i)p(M_i|y),$$
$$p(M_i|y) = \frac{p(y|M_i)p(M_i)}{\sum_{j=1}^{k} p(y|M_j)p(M_j)}.$$

- $p(M_i)$ is prior probability for model $i$ (known)
- $p(y|M_i) = C_i$ is the model evidence

# Sampling from the prior and posterior

$C = \int_X p(y|x)p(x)\,dx$ (one model for simplicity)

Prior sampling: $\{x^j\}_{j=1}^N \sim p(x)$

$$\widehat{C} = N^{-1}\sum_{j=1}^{N} p(y|x^j) \quad \text{(unbiased)}$$

Posterior sampling: $\{x^j\}_{j=1}^N \sim p(x|y)$

$$\widehat{C} = \frac{1}{N^{-1}\sum_{j=1}^{N} p(y|x^j)} \quad \text{(biased)}$$

# Importance sampling

Proposal sampling: $\{x^j\}_{j=1}^{N} \sim Q(x)$

$$\widehat{C} = N^{-1} \sum_{j=1}^{N} w(x^j) \quad \text{(unbiased)}$$

$w(x) = \frac{p(y|x)p(x)}{Q(x)}$

# Multi-fidelity approach

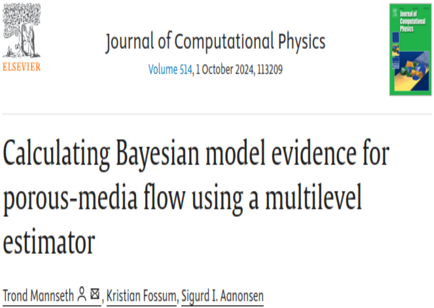
Journal of Computational Physics

Volume 514, 1 October 2024, 113209


## Calculating Bayesian model evidence for porous-media flow using a multilevel estimator

Trond Mannseth, Kristian Fossum, Sigurd I. Aanonsen

# Multi-fidelity approach


ELSEVIER

Journal of Computational Physics

Volume 514, 1 October 2024, 113209

Calculating Bayesian model evidence for porous-media flow using a multilevel estimator

Trond Mannseth, Kristian Fossum, Sigurd I. Aanonsen

- $\ell = 1, \ldots, L$ model levels ranging from fast(1) to slow($L$)

$$\widehat{C} = \sum_{\ell=1}^{L} w_\ell \widehat{C}_\ell,$$

$$w_\ell = N_\ell \Big(\sum_{j=1}^{L} N_j\Big)^{-1}$$

- Weights sum to one (in general)

# Gaussian approximation

RMetS
Royal Meteorological Society

**Estimating model evidence using data assimilation**

A. Carrassi,[a*] M. Bocquet,[b] A. Hannart[c] and M. Ghil[d,e]

[a]*Nansen Environmental and Remote Sensing Center, Bergen, Norway*
[b]*CEREA, Joint Laboratory École des Ponts ParisTech and EDF R&D, Université Paris-Est, Champs-sur-Marne, France*
[c]*IFAECI, CNRS-CONICET-UBA, Buenos Aires, Argentina*
[d]*École Normale Supérieure, Paris, France*
[e]*Department of Atmospheric and Oceanic Sciences, University of California, Los Angeles, USA*

*Correspondence to: A. Carrassi, Data Assimilation Group, Nansen Environmental and Remote Sensing Center, Thormøhlens gate 47, N-5006 Bergen, Norway. E-mail: alberto.carrassi@nersc.no

- Gaussian approximations for EnKF Ens4DVAR and IEnKS
- Explicit formulas involving mean and sample covariances (approximate Hessian)

# Ensemble on extended state space

Contents lists available at ScienceDirect

Advances in Water Resources

journal homepage: www.elsevier.com/locate/advwatres

ELSEVIER

Iterative ensemble smoothers in the annealed importance sampling framework

Andreas S. Stordal[a,*], Ahmed H. Elsheikh[b]

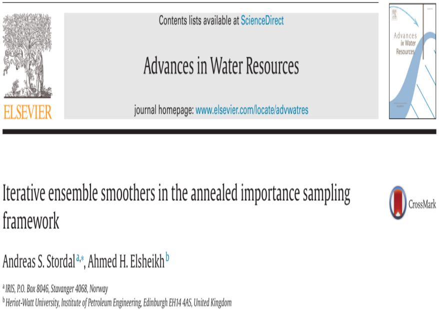
[a] IRIS, P.O. Box 8046, Stavanger 4068, Norway
[b] Heriot-Watt University, Institute of Petroleum Engineering, Edinburgh EH14 4AS, United Kingdom

- We define a new target or 'posterior' distribution in pseudo time

$$\gamma(x_{0:K}) = C^{-1} p(x_K) p(y|x_K) \prod_{j=0}^{K-1} B(x_k|x_{k+1}),$$

which leaves the posterior as marginal distribution for time $K$

# Ensemble on extended state space


Contents lists available at ScienceDirect

Advances in Water Resources

journal homepage: www.elsevier.com/locate/advwatres

ELSEVIER

Iterative ensemble smoothers in the annealed importance sampling framework

Andreas S. Stordal[a,*], Ahmed H. Elsheikh[b]

[a] IRIS, P.O. Box 8046, Stavanger 4068, Norway

[b] Heriot-Watt University, Institute of Petroleum Engineering, Edinburgh EH14 4AS, United Kingdom

- We define a new target or 'posterior' distribution in pseudo time

$$\gamma(x_{0:K}) = C^{-1} p(x_K) p(y|x_K) \prod_{j=0}^{K-1} B(x_k|x_{k+1}),$$

which leaves the posterior as marginal distribution for time $K$

- Our joint sampling distribution ,$Q$ (the proposal), is then given by

$$Q(x_{0:K}) = p(x_0) \prod_{j=1}^{K} F(x_k|x_{k-1})$$

# Cont'd

Target: $\gamma(x_0 : K) = C^{-1} p(x_K) p(y|x_K) \prod_{j=0}^{K-1} B(x_k|x_{k+1})$

Proposal: $Q(x_{0:K}) = p(x_0) \prod_{j=1}^{K} F(x_k|x_{k-1})$

# Cont'd

Target: $\gamma(x_0 : K) = C^{-1} p(x_K) p(y|x_K) \prod_{j=0}^{K-1} B(x_k|x_{k+1})$

Proposal: $Q(x_{0:K}) = p(x_0) \prod_{j=1}^{K} F(x_k|x_{k-1})$

- $F$ is a 'forward' kernel defined by the algorithm at hand (ESMDA, EnKF....)

# Cont'd

Target: $\gamma(x_0 : K) = C^{-1} p(x_K) p(y|x_K) \prod_{j=0}^{K-1} B(x_k|x_{k+1})$

Proposal: $Q(x_{0:K}) = p(x_0) \prod_{j=1}^{K} F(x_k|x_{k-1})$

- $F$ is a 'forward' kernel defined by the algorithm at hand (ESMDA, EnKF....)
- $B$ is a 'backward' kernel which has to be user defined (sadly)

# Cont'd

Target: $\gamma(x_0 : K) = C^{-1} p(x_K) p(y|x_K) \prod_{j=0}^{K-1} B(x_k|x_{k+1})$

Proposal: $Q(x_{0:K}) = p(x_0) \prod_{j=1}^{K} F(x_k|x_{k-1})$

- $F$ is a 'forward' kernel defined by the algorithm at hand (ESMDA, EnKF....)
- $B$ is a 'backward' kernel which has to be user defined (sadly)
- For sampling, an annealed version of the target is used at each iteration

# Cont'd

Target: $\gamma(x_0 : K) = C^{-1} p(x_K) p(y|x_K) \prod_{j=0}^{K-1} B(x_k|x_{k+1})$

Proposal: $Q(x_{0:K}) = p(x_0) \prod_{j=1}^{K} F(x_k|x_{k-1})$

- $F$ is a 'forward' kernel defined by the algorithm at hand (ESMDA, EnKF....)
- $B$ is a 'backward' kernel which has to be user defined (sadly)
- For sampling, an annealed version of the target is used at each iteration
- Not necessary for Evidence computation as we do not re-sample ensemble members

# Cont'd

Target: $\gamma(x_0:K) = C^{-1}p(x_K)p(y|x_K)\prod_{j=0}^{K-1} B(x_k|x_{k+1})$

Proposal: $Q(x_{0:K}) = p(x_0)\prod_{j=1}^{K} F(x_k|x_{k-1})$

- $F$ is a 'forward' kernel defined by the algorithm at hand (ESMDA, EnKF....)
- $B$ is a 'backward' kernel which has to be user defined (sadly)
- For sampling, an annealed version of the target is used at each iteration
- Not necessary for Evidence computation as we do not re-sample ensemble members
- $\widehat{C} = N^{-1}\sum_{j=1}^{N} w(x_{0:K}^{j}), \quad w = \gamma Q^{-1}$

# Optimality via discretized diffusion

## Score-Based Diffusion meets Annealed Importance Sampling

Arnaud Doucet, Will Grathwohl, Alexander G. D. G. Matthews & Heiko Strathmann *
DeepMind
{arnauddoucet,wgrathwohl,alexmatthews,strathmann}@google.com

# Optimality via discretized diffusion

- $F$ is selected as discretized diffusion

Score-Based Diffusion meets
Annealed Importance Sampling

Arnaud Doucet, Will Grathwohl, Alexander G. D. G. Matthews & Heiko Strathmann *
DeepMind
{arnauddoucet,wgrathwohl,alexmatthews,strathmann}@google.com

# Optimality via discretized diffusion

**Score-Based Diffusion meets Annealed Importance Sampling**

Arnaud Doucet, Will Grathwohl, Alexander G. D. G. Matthews & Heiko Strathmann *
DeepMind
{arnauddoucet,wgrathwohl,alexmatthews,strathmann}@google.com

- $F$ is selected as discretized diffusion
- Optimal $B$ is a 'backward' discretized diffusion involving $\nabla q_k(x)$, the 'log-score'

NORCE

# Optimality via discretized diffusion

**Score-Based Diffusion meets Annealed Importance Sampling**

Arnaud Doucet, Will Grathwohl, Alexander G. D. G. Matthews & Heiko Strathmann *
DeepMind
{arnauddoucet,wgrathwohl,alexmatthews,strathmann}@google.com

- $F$ is selected as discretized diffusion
- Optimal $B$ is a 'backward' discretized diffusion involving $\nabla q_k(x)$, the 'log-score'
- $q_k$ is the unknown marginal density of samples at iteration $k$

# Optimality via discretized diffusion

**Score-Based Diffusion meets Annealed Importance Sampling**

Arnaud Doucet, Will Grathwohl, Alexander G. D. G. Matthews & Heiko Strathmann *
DeepMind
{arnauddoucet,wgrathwohl,alexmatthews,strathmann}@google.com

- $F$ is selected as discretized diffusion
- Optimal $B$ is a 'backward' discretized diffusion involving $\nabla q_k(x)$, the 'log-score'
- $q_k$ is the unknown marginal density of samples at iteration $k$
- Neural Network, $S_\theta$ and score matching minimizing

$$\mathcal{L}(\theta) = \delta \sum_{k=1}^{K} \mathbf{E}_Q \left[ \| S_\theta(x_k) - \nabla F(x_k | x_{k-1}) \|^2 \right]$$

# Alternative to score matching

$$\begin{aligned}\nabla \log q(x_k) &= \int \nabla \log F(x_k|x_{k-1}) q(x_{k-1}|x_k) dx_{k-1}, \\ &= \int \nabla \log F(x_k|x_{k-1}) \frac{F(x_k|x_{k-1})}{q(x_k)} q(x_{k-1}) dx_{k-1}.\end{aligned}$$

Estimate $\nabla \log q(x_k)$ at iteration $k$ for particle $i$ using the set $\{x_{k-1}^j\}_{j=1}^N$

$$\nabla \log q(x_k^i) \approx \sum_{j=1}^{N} \nabla \log F(x_k^i|x_{k-1}^j)\omega(x_{k-1}^j),$$

$$\omega(x_{k-1}^j) = \frac{F(x_k^i|x_{k-1}^j)}{\sum_{\ell=1}^N F(x_k^i|x_{k-1}^\ell)}$$

# Algorithms

- For stochastic algorithms (e.g. ESMDA) we sequentially compute the log weights as

$$
\begin{aligned}
\log w_0 &= -\log p(x_0) \\
\log w_k &= \log w_{k-1} + \log B(x_{k-1}|x_k) - \log F(x_k|x_{k-1}), \quad k = 1, \ldots K-1 \\
\log w_K &= \log w_{K-1} + \log p(y|x_K) + \log p(x_K)
\end{aligned}
$$

# Algorithms

- For stochastic algorithms (e.g. ESMDA) we sequentially compute the log weights as

$$
\begin{aligned}
\log w_0 &= -\log p(x_0) \\
\log w_k &= \log w_{k-1} + \log B(x_{k-1}|x_k) - \log F(x_k|x_{k-1}), \quad k = 1, \dots K-1 \\
\log w_K &= \log w_{K-1} + \log p(y|x_K) + \log p(x_K)
\end{aligned}
$$

- For deterministic maps $F(x_k|x_{k-1}) = T(x_{k-1})$ (e.g. EnSRF) we require $\nabla T(x)$

# Some examples

- ESMDA updates the ensemble as

$$X_k = X_{k-1} + K(\alpha_k)(y - \mathcal{H}(X_{k-1}) + \alpha_k R^{-1/2} Z)$$

# Some examples

- ESMDA updates the ensemble as

$$X_k = X_{k-1} + K(\alpha_k)(y - \mathcal{H}(X_{k-1}) + \alpha_k R^{-1/2} Z)$$

- Forward kernel, $F(x_k | x_{k-1})$ is Gaussian with mean and covariance

$$\mu_k = x_{k-1} + K(\alpha_k)(y - \mathcal{H}(x_{k-1})$$
$$P_k = \alpha_k^2 K(\alpha_k) R_K^\top(\alpha_k)$$

## Some examples

- EnSF updates the ensemble as

$$\mu_k = \mu_{k-1} + \delta K_1 (y - \overline{\mathcal{H}(X_{k-1})})$$

# Some examples

- EnSF updates the ensemble as

$$\mu_k = \mu_{k-1} + \delta K_1 (y - \overline{\mathcal{H}(X_{k-1})})$$

- Forward kernel, $T(x_{k-1})$ is deterministic

$$T(x_{k-1}) = \mu_{k-1} + A x_{k-1}$$
$$\nabla T = N^{-1} I - N^{-1} \delta K_1 \nabla \mathcal{H} + (I - \delta K_2)^{1/2} (1 - 1/N) I$$

# Some examples

- EnSF updates the ensemble as

$$\mu_k = \mu_{k-1} + \delta K_1 (y - \overline{\mathcal{H}(X_{k-1})})$$

- Forward kernel, $T(x_{k-1})$ is deterministic

$$T(x_{k-1}) = \mu_{k-1} + A x_{k-1}$$
$$\nabla T = N^{-1} I - N^{-1} \delta K_1 \nabla \mathcal{H} + (I - \delta K_2)^{1/2} (1 - 1/N) I$$

- Gradient of measurement operator required (Same for RML, EnRML)

# Bayesian Stacking

Instead of using evidence in model weighting, find maximum over $\{w^i\}_{i=1}^m$

$$\sum_{i=1}^{d} \log \left( \sum_{i=1}^{m} w^i p(y_i | y_{(-i)}, M_i) \right)$$

LOO likelihood $p(y_i | y_{(-i)})$ requires rerunning $d$ experiments.

$$w_2(x_0, \dots, x_K) = \frac{p(y_i | x_J) p(x_K | y_{(-i)}) \prod_{j=0}^{K-1} B(x_k | x_{k+1})}{p(x_0) \prod_{k=1}^{J} F_k(x_k | x_{k-1})},$$

and rewrite

$$p(y_i | x_J) p(x_J | y_{(-i)}) = p(y_i | x_J) p(y_{(-i)} | x_J) p(x_J) C_{(-i)}^{-1}$$

# Stacking estimate

- The sum of the importance weights

$$w_2(x_{0:K}) = \frac{p(y|x_K)p(x_K)\prod_{k=0}^{K-1} B(x_k|x_{k+1})}{p(y_i|x_K)p(x_0)\prod_{k=1}^{K} F(x_k|x_{k-1})},$$

will converge to $C_{(-i)}$

# Stacking estimate

- The sum of the importance weights

$$w_2(x_{0:K}) = \frac{p(y|x_K)p(x_K)\prod_{k=0}^{K-1} B(x_k|x_{k+1})}{p(y_i|x_K)p(x_0)\prod_{k=1}^{K} F(x_k|x_{k-1})},$$

  will converge to $C_{(-i)}$

- Furthermore $\log w_2 = \log w - \log p(y_i|x_K)$ (from evidence computation)

# Stacking estimate

- The sum of the importance weights

$$w_2(x_{0:K}) = \frac{p(y|x_K)p(x_K)\prod_{k=0}^{K-1} B(x_k|x_{k+1})}{p(y_i|x_K)p(x_0)\prod_{k=1}^{K} F(x_k|x_{k-1})},$$

  will converge to $C_{(-i)}$
- Furthermore $\log w_2 = \log w - \log p(y_i|x_K)$ (from evidence computation)
- We can estimate $p(y_i|y_{(-i)})$ using

$$\hat{p}(y_i|y_{(-i)}) = \frac{\sum_{j=1}^{N} w^j}{\sum_{j=1}^{N} w_2^j}$$

  without doing any cross-validation

# Toy example, 1000 reps

- $X \sim N(\mu, \sigma^2)$, $Y = aX^2 + bX + \epsilon$
- Estimate model evidence with ESMDA(red) and unweighted posterior sampling(blue)

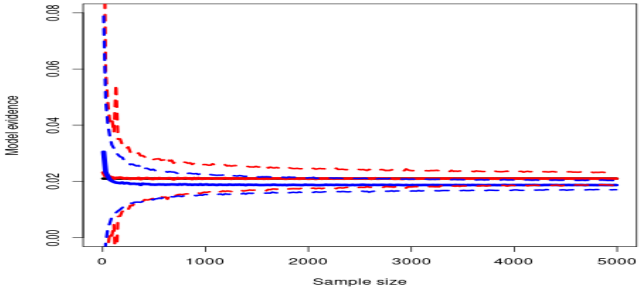
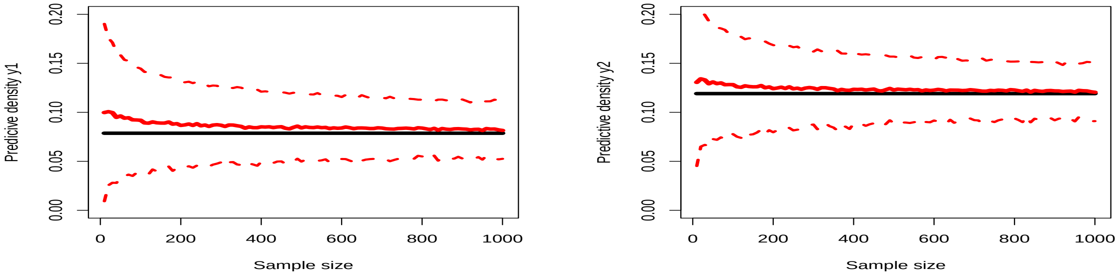



Figure: Evidence estimates as function sample size

# Two observations, stacking




Estimate of $p(y_1|y_2)$ and $p(y_2|y_1)$

# Summary

- Defined a new way to compute model evidence for iterative ensemble methods

# Summary

- Defined a new way to compute model evidence for iterative ensemble methods
- May be computed during iterations with almost no extra cost

# Summary

- Defined a new way to compute model evidence for iterative ensemble methods
- May be computed during iterations with almost no extra cost
- Extended to stacking with no additional cost

# Summary

- Defined a new way to compute model evidence for iterative ensemble methods
- May be computed during iterations with almost no extra cost
- Extended to stacking with no additional cost
- Large variance, reduce by combining with multi-fidelity methods

# Summary

- Defined a new way to compute model evidence for iterative ensemble methods
- May be computed during iterations with almost no extra cost
- Extended to stacking with no additional cost
- Large variance, reduce by combining with multi-fidelity methods
- Evaluate and compare with other methods proposed