Machine learning assisted Bayesian evidence computation

The *learnt* harmonic mean estimator

Jason McEwen
www.jasonmcewen.org
@jasonmcewen

Mullard Space Science Laboratory (MSSL)
University College London (UCL)

Inverse Problems from Theory to Application, UCL, September 2019

Outline

Outline

Bayesian inference
Parameter estimation

Bayes' theorem

$$\underbrace{\mathsf{P}(\theta \,|\, \boldsymbol{y}, M)}_{\text{posterior}} = \frac{\overbrace{\mathsf{P}(\boldsymbol{y} \,|\, \theta, M)}^{\text{likelihood}} \; \overbrace{\mathsf{P}(\theta \,|\, M)}^{\text{prior}}}{\underbrace{\mathsf{P}(\boldsymbol{y} \,|\, M)}_{\text{constant}}},$$

for parameters θ, model M and observed data $\boldsymbol{y}$.

Bayesian inference

Parameter estimation

Bayes' theorem

$$\underbrace{\mathrm{P}(\theta \mid \boldsymbol{y}, M)}_{\text{posterior}} = \frac{\overbrace{\mathrm{P}(\boldsymbol{y} \mid \theta, M)}^{\text{likelihood}} \; \overbrace{\mathrm{P}(\theta \mid M)}^{\text{prior}}}{\underbrace{\mathrm{P}(\boldsymbol{y} \mid M)}_{\text{constant}}},$$

for parameters θ, model M and observed data $\boldsymbol{y}$.

Shorthand notation:

$$\underbrace{\mathrm{P}(\theta \mid \boldsymbol{y})}_{\text{posterior}} = \frac{\overbrace{\mathcal{L}(\theta)}^{\text{likelihood}} \; \overbrace{\pi(\theta)}^{\text{prior}}}{\underbrace{z}_{\text{constant}}},$$

Bayesian inference

Parameter estimation

Bayes' theorem

$$\underbrace{\mathrm{P}(\theta \mid \boldsymbol{y}, M)}_{\text{posterior}} = \frac{\overbrace{\mathrm{P}(\boldsymbol{y} \mid \theta, M)}^{\text{likelihood}} \; \overbrace{\mathrm{P}(\theta \mid M)}^{\text{prior}}}{\underbrace{\mathrm{P}(\boldsymbol{y} \mid M)}_{\text{constant}}},$$

for parameters θ, model M and observed data $\boldsymbol{y}$.

Shorthand notation:

$$\underbrace{\mathrm{P}(\theta \mid \boldsymbol{y})}_{\text{posterior}} = \frac{\overbrace{\mathcal{L}(\theta)}^{\text{likelihood}} \; \overbrace{\pi(\theta)}^{\text{prior}}}{\underbrace{z}_{\text{constant}}},$$

For **parameter estimation**, typically draw samples from the posterior by *Markov chain Monte Carlo (MCMC)* sampling.

Bayesian inference

Model selection

For **model selection**, consider the posterior model probabilities:

$$\underbrace{\frac{\mathrm{P}(M_1 \mid \boldsymbol{y})}{\mathrm{P}(M_2 \mid \boldsymbol{y})}}_{\text{posterior odds}} = \underbrace{\frac{\mathrm{P}(M_1)}{\mathrm{P}(M_2)}}_{\text{prior odds}} \times \underbrace{\frac{\mathrm{P}(\boldsymbol{y} \mid M_1)}{\mathrm{P}(\boldsymbol{y} \mid M_2)}}_{\text{Bayes factor}} .$$

Bayesian inference

Model selection

For **model selection**, consider the posterior model probabilities:

$$\underbrace{\frac{\mathrm{P}(M_1 \mid \boldsymbol{y})}{\mathrm{P}(M_2 \mid \boldsymbol{y})}}_{\text{posterior odds}} = \underbrace{\frac{\mathrm{P}(M_1)}{\mathrm{P}(M_2)}}_{\text{prior odds}} \times \underbrace{\frac{\mathrm{P}(\boldsymbol{y} \mid M_1)}{\mathrm{P}(\boldsymbol{y} \mid M_2)}}_{\text{Bayes factor}} .$$

Must compute the **Bayesian evidence** or **marginal likelihood** given by the normalising constant

$$z = \mathrm{P}(\boldsymbol{y} \mid M) = \int \mathrm{d}\theta \, \mathcal{L}(\theta)\pi(\theta) .$$

Bayesian inference
Model selection

For **model selection**, consider the posterior model probabilities:

$$\underbrace{\frac{\mathrm{P}(M_1 \mid \boldsymbol{y})}{\mathrm{P}(M_2 \mid \boldsymbol{y})}}_{\text{posterior odds}} = \underbrace{\frac{\mathrm{P}(M_1)}{\mathrm{P}(M_2)}}_{\text{prior odds}} \times \underbrace{\frac{\mathrm{P}(\boldsymbol{y} \mid M_1)}{\mathrm{P}(\boldsymbol{y} \mid M_2)}}_{\text{Bayes factor}} .$$

Must compute the **Bayesian evidence** or **marginal likelihood** given by the normalising constant

$$z = \mathrm{P}(\boldsymbol{y} \mid M) = \int \mathrm{d}\theta \, \mathcal{L}(\theta)\pi(\theta) .$$

$\rightarrow$ **Challenging computational problem in high-dimensions.**

Bayesian inference

Model selection

For **model selection**, consider the posterior model probabilities:

$$\underbrace{\frac{\mathrm{P}(M_1 \mid \boldsymbol{y})}{\mathrm{P}(M_2 \mid \boldsymbol{y})}}_{\text{posterior odds}} = \underbrace{\frac{\mathrm{P}(M_1)}{\mathrm{P}(M_2)}}_{\text{prior odds}} \times \underbrace{\frac{\mathrm{P}(\boldsymbol{y} \mid M_1)}{\mathrm{P}(\boldsymbol{y} \mid M_2)}}_{\text{Bayes factor}} .$$

Must compute the **Bayesian evidence** or **marginal likelihood** given by the normalising constant

$$z = \mathrm{P}(\boldsymbol{y} \mid M) = \int \mathrm{d}\theta \, \mathcal{L}(\theta) \pi(\theta) .$$

→ **Challenging computational problem in high-dimensions.**

Variety of powerful methods exist but often place restrictions on sampling method and struggle to push to high dimensional settings.

Desirable properties for Bayesian evidence estimators

Seek estimator that is:

- Agnostic to sampling method and **uses posterior samples**.
- Scales to **high-dimensions**.

Desirable properties for Bayesian evidence estimators

Seek estimator that is:

- Agnostic to sampling method and **uses posterior samples**.
- Scales to **high-dimensions**.

Harmonic mean estimator has potential to meet these criteria but has serious shortcomings as originally posed.

Original harmonic mean estimator

Harmonic mean relationship (Newton & Raftery 1994)

$$\rho = \mathbb{E}_{\mathsf{P}(\theta \mid \boldsymbol{y})} \left[\frac{1}{\mathcal{L}(\theta)} \right]$$

Original harmonic mean estimator

Harmonic mean relationship (Newton & Raftery 1994)

$$\rho = \mathbb{E}_{\mathsf{P}(\theta \,|\, \boldsymbol{y})}\left[\frac{1}{\mathcal{L}(\theta)}\right] = \int \mathrm{d}\theta \frac{1}{\mathcal{L}(\theta)} \mathsf{P}(\theta \,|\, \boldsymbol{y})$$

Original harmonic mean estimator

Harmonic mean relationship (Newton & Raftery 1994)

$$\rho = \mathbb{E}_{\mathsf{P}(\theta \,|\, \boldsymbol{y})} \left[\frac{1}{\mathcal{L}(\theta)} \right] = \int \mathrm{d}\theta \frac{1}{\mathcal{L}(\theta)} \mathsf{P}(\theta \,|\, \boldsymbol{y})$$

$$= \int \mathrm{d}\theta \frac{1}{\mathcal{L}(\theta)} \frac{\mathcal{L}(\theta)\pi(\theta)}{z}$$

Original harmonic mean estimator

Harmonic mean relationship (Newton & Raftery 1994)

$$\rho = \mathbb{E}_{\mathrm{P}(\theta \,|\, \boldsymbol{y})}\left[\frac{1}{\mathcal{L}(\theta)}\right] = \int \mathrm{d}\theta \frac{1}{\mathcal{L}(\theta)} \mathrm{P}(\theta \,|\, \boldsymbol{y})$$
$$= \int \mathrm{d}\theta \frac{1}{\mathcal{L}(\theta)} \frac{\mathcal{L}(\theta)\pi(\theta)}{z}$$
$$= \frac{1}{z}$$

Original harmonic mean estimator

Harmonic mean relationship (Newton & Raftery 1994)

$$\rho = \mathbb{E}_{\mathrm{P}(\theta \,|\, \boldsymbol{y})}\left[\frac{1}{\mathcal{L}(\theta)}\right] = \int \mathrm{d}\theta \, \frac{1}{\mathcal{L}(\theta)} \mathrm{P}(\theta \,|\, \boldsymbol{y}) = \int \mathrm{d}\theta \, \frac{1}{\mathcal{L}(\theta)} \frac{\mathcal{L}(\theta)\pi(\theta)}{z} = \frac{1}{z}$$

Original harmonic mean estimator (Newton & Raftery 1994)

$$\hat{\rho} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\mathcal{L}(\theta_i)} , \quad \theta_i \sim \mathrm{P}(\theta \,|\, \boldsymbol{y})$$

Original harmonic mean estimator

Harmonic mean relationship (Newton & Raftery 1994)

$$\begin{aligned}\rho = \mathbb{E}_{\mathrm{P}(\theta \,|\, \boldsymbol{y})}\left[\frac{1}{\mathcal{L}(\theta)}\right] &= \int \mathrm{d}\theta \, \frac{1}{\mathcal{L}(\theta)} \mathrm{P}(\theta \,|\, \boldsymbol{y}) \\ &= \int \mathrm{d}\theta \, \frac{1}{\mathcal{L}(\theta)} \frac{\mathcal{L}(\theta)\pi(\theta)}{z} \\ &= \frac{1}{z}\end{aligned}$$

Original harmonic mean estimator (Newton & Raftery 1994)

$$\hat{\rho} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\mathcal{L}(\theta_i)} , \quad \theta_i \sim \mathrm{P}(\theta \,|\, \boldsymbol{y})$$

Very simple approach but **can fail catastrophically** (Neal 1994).

Original harmonic mean estimator

Importance sampling interpretation

Alternative derivation of harmonic mean relationship:

$$\rho = \frac{1}{z} = \frac{\displaystyle\int \mathrm{d}\theta \frac{\pi(\theta)}{\mathsf{P}(\theta\,|\,\boldsymbol{y})} \mathsf{P}(\theta\,|\,\boldsymbol{y})}{z} = \int \mathrm{d}\theta \frac{1}{\mathcal{L}(\theta)} \mathsf{P}(\theta\,|\,\boldsymbol{y}) \,.$$

Original harmonic mean estimator

Importance sampling interpretation

Alternative derivation of harmonic mean relationship:

$$\rho = \frac{1}{z} = \frac{\overbrace{\int \mathrm{d}\theta \, \frac{\pi(\theta)}{\mathsf{P}(\theta \,|\, \boldsymbol{y})} \mathsf{P}(\theta \,|\, \boldsymbol{y})}^{\text{importance sampling}}}{z} = \int \mathrm{d}\theta \, \frac{1}{\mathcal{L}(\theta)} \mathsf{P}(\theta \,|\, \boldsymbol{y}) \, .$$

Importance sampling interpretation:

- Importance **sampling target distribution is prior** $\pi(\theta)$.
- Importance **sampling density is posterior** $\mathsf{P}(\theta \,|\, \boldsymbol{y})$.

Original harmonic mean estimator

Importance sampling interpretation

Alternative derivation of harmonic mean relationship:

$$\rho = \frac{1}{z} = \frac{\overbrace{\int \mathrm{d}\theta \frac{\pi(\theta)}{\mathrm{P}(\theta \,|\, \boldsymbol{y})} \mathrm{P}(\theta \,|\, \boldsymbol{y})}^{\text{importance sampling}}}{z} = \int \mathrm{d}\theta \frac{1}{\mathcal{L}(\theta)} \mathrm{P}(\theta \,|\, \boldsymbol{y}) \, .$$

Importance sampling interpretation:

- Importance **sampling target distribution is prior** $\pi(\theta)$.
- Importance **sampling density is posterior** $\mathrm{P}(\theta \,|\, \boldsymbol{y})$.

For importance sampling, typically want sampling density to have fatter tails than target.

Original harmonic mean estimator

Importance sampling interpretation

Alternative derivation of harmonic mean relationship:

$$\rho = \frac{1}{z} = \frac{\overbrace{\int \mathrm{d}\theta \frac{\pi(\theta)}{\mathrm{P}(\theta \,|\, \boldsymbol{y})} \mathrm{P}(\theta \,|\, \boldsymbol{y})}^{\text{importance sampling}}}{z} = \int \mathrm{d}\theta \frac{1}{\mathcal{L}(\theta)} \mathrm{P}(\theta \,|\, \boldsymbol{y}) \, .$$

Importance sampling interpretation:

- Importance **sampling target distribution is prior** $\pi(\theta)$.
- Importance **sampling density is posterior** $\mathrm{P}(\theta \,|\, \boldsymbol{y})$.

For importance sampling, typically want sampling density to have fatter tails than target.

Not the case when importance sampling density is the posterior and the target is the prior.

Original harmonic mean estimator

Simulation pseudo bias

Simulation pseudo bias (Lenk 2009)

In practice posterior simulation support Ω is a subset of the prior support Θ, hence do not fully capture prior (target distribution).

Original harmonic mean estimator

Simulation pseudo bias

Simulation pseudo bias (Lenk 2009)

In practice posterior simulation support Ω is a subset of the prior support Θ, hence do not fully capture prior (target distribution).

Corrected harmonic mean estimator (Lenk 2009)

$$\hat{\rho} = \mathrm{P}(\Omega) \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\mathcal{L}(\theta_i)} , \quad \theta_i \sim \mathrm{P}(\theta \,|\, \boldsymbol{y}) ,$$

where $\mathrm{P}(\Omega)$ is the prior probability of the posterior simulation support $\Omega \subset \Theta$.

Original harmonic mean estimator
Simulation pseudo bias

Simulation pseudo bias (Lenk 2009)

In practice posterior simulation support Ω is a subset of the prior support Θ, hence do not fully capture prior (target distribution).

Corrected harmonic mean estimator (Lenk 2009)

$$\hat{\rho} = \mathrm{P}(\Omega) \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\mathcal{L}(\theta_i)} , \quad \theta_i \sim \mathrm{P}(\theta \,|\, \boldsymbol{y}) ,$$

where $\mathrm{P}(\Omega)$ is the prior probability of the posterior simulation support $\Omega \subset \Theta$.

Mitigates simulation pseudo bias but does not eliminate.

Re-targeted harmonic mean estimator

Introduce an arbitrary importance sampling target $\varphi(\theta)$ (which must be normalised).

Re-targeted harmonic mean estimator

Introduce an arbitrary importance sampling target $\varphi(\theta)$ (which must be normalised).

***Re-targeted* harmonic mean relationship** (Gelfand & Dey 1994)

$$\rho = \mathbb{E}_{\mathrm{P}(\theta \,|\, \boldsymbol{y})}\left[\frac{\varphi(\theta)}{\mathcal{L}(\theta)\pi(\theta)}\right] = \int \mathrm{d}\theta \, \frac{\varphi(\theta)}{\mathcal{L}(\theta)\pi(\theta)} \mathrm{P}(\theta \,|\, \boldsymbol{y})$$

Re-targeted harmonic mean estimator

Introduce an arbitrary importance sampling target $\varphi(\theta)$ (which must be normalised).

***Re-targeted* harmonic mean relationship** (Gelfand & Dey 1994)

$$\rho = \mathbb{E}_{\mathsf{P}(\theta \,|\, \boldsymbol{y})}\left[\frac{\varphi(\theta)}{\mathcal{L}(\theta)\pi(\theta)}\right] = \int \mathrm{d}\theta \, \frac{\varphi(\theta)}{\mathcal{L}(\theta)\pi(\theta)} \mathsf{P}(\theta \,|\, \boldsymbol{y})$$

$$= \int \mathrm{d}\theta \, \frac{\varphi(\theta)}{\mathcal{L}(\theta)\pi(\theta)} \frac{\mathcal{L}(\theta)\pi(\theta)}{z}$$

Re-targeted harmonic mean estimator

Introduce an arbitrary importance sampling target $\varphi(\theta)$ (which must be normalised).

***Re-targeted* harmonic mean relationship** (Gelfand & Dey 1994)

$$\begin{aligned}\rho = \mathbb{E}_{\mathsf{P}(\theta\,|\,\boldsymbol{y})}\left[\frac{\varphi(\theta)}{\mathcal{L}(\theta)\pi(\theta)}\right] &= \int \mathrm{d}\theta\,\frac{\varphi(\theta)}{\mathcal{L}(\theta)\pi(\theta)}\mathsf{P}(\theta\,|\,\boldsymbol{y}) \\ &= \int \mathrm{d}\theta\,\frac{\varphi(\theta)}{\mathcal{L}(\theta)\pi(\theta)}\,\frac{\mathcal{L}(\theta)\pi(\theta)}{z} \\ &= \frac{1}{z}\end{aligned}$$

Re-targeted harmonic mean estimator

Introduce an arbitrary importance sampling target $\varphi(\theta)$ (which must be normalised).

***Re-targeted* harmonic mean relationship** (Gelfand & Dey 1994)

$$\rho = \mathbb{E}_{\mathrm{P}(\theta\,|\,\boldsymbol{y})}\left[\frac{\varphi(\theta)}{\mathcal{L}(\theta)\pi(\theta)}\right] = \int \mathrm{d}\theta\, \frac{\varphi(\theta)}{\mathcal{L}(\theta)\pi(\theta)} \mathrm{P}(\theta\,|\,\boldsymbol{y}) = \int \mathrm{d}\theta\, \frac{\varphi(\theta)}{\mathcal{L}(\theta)\pi(\theta)} \frac{\mathcal{L}(\theta)\pi(\theta)}{z} = \frac{1}{z}$$

***Re-targeted* harmonic mean estimator** (Gelfand & Dey 1994)

$$\hat{\rho} = \frac{1}{N}\sum_{i=1}^{N} \frac{\varphi(\theta_i)}{\mathcal{L}(\theta_i)\pi(\theta_i)}\,, \quad \theta_i \sim \mathrm{P}(\theta\,|\,\boldsymbol{y})$$

Re-targeted harmonic mean estimator

Importance sampling interpretation

Importance sampling derivation:

$$\rho = \frac{1}{z} = \frac{\int \mathrm{d}\theta \frac{\varphi(\theta)}{\mathsf{P}(\theta \,|\, \boldsymbol{y})} \mathsf{P}(\theta \,|\, \boldsymbol{y})}{z} = \int \mathrm{d}\theta \frac{\varphi(\theta)}{\mathcal{L}(\theta)\pi(\theta)} \mathsf{P}(\theta \,|\, \boldsymbol{y}) \,.$$

Re-targeted harmonic mean estimator

Importance sampling interpretation

Importance sampling derivation:

$$\rho = \frac{1}{z} = \frac{\int \mathrm{d}\theta \frac{\varphi(\theta)}{\mathrm{P}(\theta \,|\, \boldsymbol{y})} \mathrm{P}(\theta \,|\, \boldsymbol{y})}{z} = \int \mathrm{d}\theta \frac{\varphi(\theta)}{\mathcal{L}(\theta)\pi(\theta)} \mathrm{P}(\theta \,|\, \boldsymbol{y}) \,.$$

- Ensure importance sampling target $\varphi(\theta)$ does not have fatter tails than posterior $\mathrm{P}(\theta \,|\, \boldsymbol{y})$ (importance sampling density).

Re-targeted harmonic mean estimator

Importance sampling interpretation

Importance sampling derivation:

$$\rho = \frac{1}{z} = \frac{\int \mathrm{d}\theta \frac{\varphi(\theta)}{\mathsf{P}(\theta \,|\, \boldsymbol{y})} \mathsf{P}(\theta \,|\, \boldsymbol{y})}{z} = \int \mathrm{d}\theta \frac{\varphi(\theta)}{\mathcal{L}(\theta)\pi(\theta)} \mathsf{P}(\theta \,|\, \boldsymbol{y}) \,.$$

- Ensure importance sampling target $\varphi(\theta)$ does not have fatter tails than posterior $\mathsf{P}(\theta \,|\, \boldsymbol{y})$ (importance sampling density).

$\rightarrow$ **How set importance sampling target distribution $\varphi(\theta)$?**

Re-targeted harmonic mean estimator

How set importance sampling target distribution $\varphi(\theta)$?

Re-targeted harmonic mean estimator

How set importance sampling target distribution $\varphi(\theta)$?

Optimal target:

$$\varphi^{\text{optimal}}(\theta) = \frac{\mathcal{L}(\theta)\pi(\theta)}{z}$$

(resulting estimator has zero variance).

Re-targeted harmonic mean estimator

How set importance sampling target distribution $\varphi(\theta)$?

Optimal target:

$$\varphi^{\text{optimal}}(\theta) = \frac{\mathcal{L}(\theta)\pi(\theta)}{z}$$

(resulting estimator has zero variance).

Recall:

$$\hat{\rho} = \frac{1}{N}\sum_{i=1}^{N} \frac{\varphi(\theta_i)}{\mathcal{L}(\theta_i)\pi(\theta_i)}\ , \quad \theta_i \sim \mathrm{P}(\theta\,|\,\boldsymbol{y})$$

Re-targeted harmonic mean estimator

How set importance sampling target distribution $\varphi(\theta)$?

Optimal target:

$$\varphi^{\text{optimal}}(\theta) = \frac{\mathcal{L}(\theta)\pi(\theta)}{z}$$

(resulting estimator has zero variance).

Recall:

$$\hat{\rho} = \frac{1}{N}\sum_{i=1}^{N} \frac{\varphi(\theta_i)}{\mathcal{L}(\theta_i)\pi(\theta_i)}\ , \quad \theta_i \sim \mathrm{P}(\theta \,|\, \boldsymbol{y})$$

But clearly **not feasible** since requires knowledge of the evidence z (recall the target must be normalised) $\rightarrow$ **requires problem to have been solved already!**

Learnt harmonic mean estimator

Learn an approximation of the optimal target distribution:

$$\boxed{\varphi(\theta) \overset{\text{ML}}{\simeq} \varphi^{\text{optimal}}(\theta) = \frac{\mathcal{L}(\theta)\pi(\theta)}{z}} \; .$$

Learnt harmonic mean estimator

Learn an approximation of the optimal target distribution:

$$\varphi(\theta) \overset{\text{ML}}{\simeq} \varphi^{\text{optimal}}(\theta) = \frac{\mathcal{L}(\theta)\pi(\theta)}{z} \, .$$

- Approximation not required to be highly accurate.
- Must not have fatter tails than posterior.

Learnt harmonic mean estimator

Learn an approximation of the optimal target distribution:

$$\varphi(\theta) \overset{\text{ML}}{\simeq} \varphi^{\text{optimal}}(\theta) = \frac{\mathcal{L}(\theta)\pi(\theta)}{z} .$$

- Approximation not required to be highly accurate.
- Must not have fatter tails than posterior.

Also develop strategy to estimate the variance of the estimator, its variance, and other sanity checks.

Learnt harmonic mean estimator

Learning the target distribution

Consider a **variety of machine learning approaches**:

- Uniform hyper-ellipsoid
- Kernel Density Estimation (KDE)
- Modified Gaussian mixture model (MGMM)

Modify learning objective function to include **variance penalty and regularisation**.

Solve by bespoke **mini-batch stochastic gradient descent**.

Cross-validation to select machine learning approach and hyperparameters.

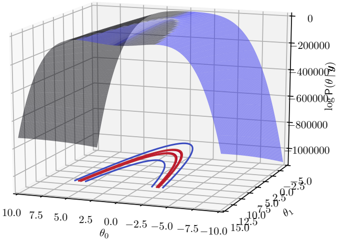
Outline

Rosenbrock example

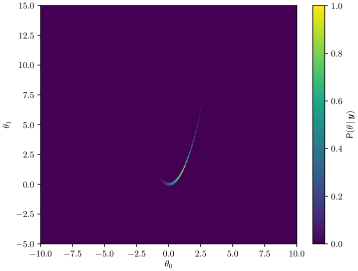
Posterior

Rosenbrock function is the classical example of a **pronounced thin curving degeneracy**, with likelihood defined by

$$f(\theta) = \sum_{i=1}^{n-1} \left[(a - \theta_i)^2 + b(\theta_{i+1} - \theta_i^2)^2 \right] , \qquad \log(\mathcal{L}(\theta)) = -f(\theta) \, .$$



(a) Log-Posterior



(b) Posterior

Figure: Rosenbrock posterior evaluated on grid.

Rosenbrock example

MCMC sampling and learning the target distribution φ

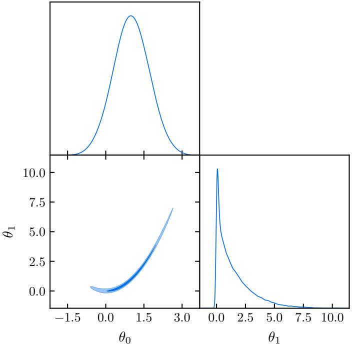


Figure: Posterior recovered by MCMC sampling.

Rosenbrock example

MCMC sampling and learning the target distribution φ

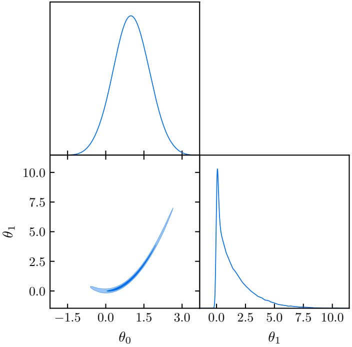


Figure: Posterior recovered by MCMC sampling.

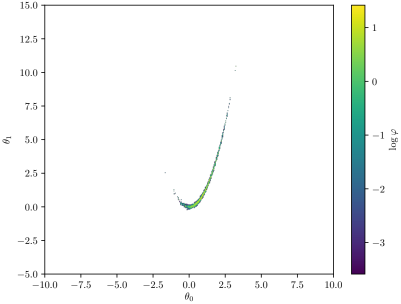
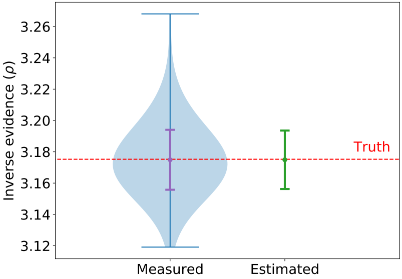


Figure: Learnt target distribution φ (by KDE).

Rosenbrock example

Accuracy of learnt harmonic mean estimator

- Compare to Monte Carlo simulations, repeating entire analysis.
- Also estimate the variance of the estimator and its variance.



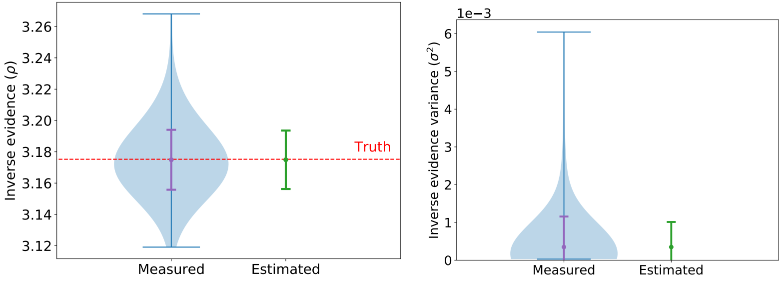
(a) Inverse evidence

Figure: Accuracy of learnt harmonic mean estimator for Rosenbrock example.

Rosenbrock example

Accuracy of learnt harmonic mean estimator

- Compare to Monte Carlo simulations, repeating entire analysis.
- Also estimate the variance of the estimator and its variance.



(a) Inverse evidence **(b) Variance of inverse evidence**

Figure: Accuracy of learnt harmonic mean estimator for Rosenbrock example.

Normal-Gamma example

Model

Pathological example (Friel & Wyse 2012) where original harmonic mean estimator fails.

Normal-Gamma example
Model

Pathological example (Friel & Wyse 2012) where original harmonic mean estimator fails.

Data model:

$$y_i \sim \mathrm{N}(\mu, \tau^{-1})$$

Prior model:

$$\text{Mean: } \mu \sim \mathrm{N}\big(\mu_0, (\tau_0 \tau)^{-1}\big)$$
$$\text{Precision: } \tau \sim \mathrm{Ga}(a_0, b_0)$$

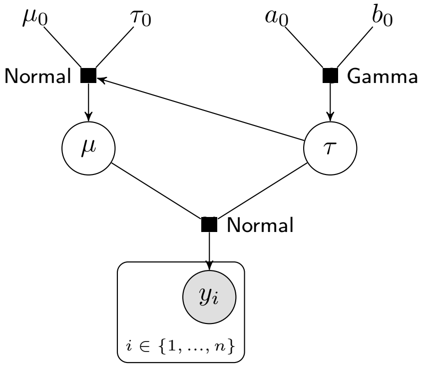


Figure: Graph of hierarchical Bayesian model of Normal-Gamma example.

Normal-Gamma example

Analytic evidence

Analytic evidence:

$$z = (2\pi)^{-n/2} \frac{\Gamma(a_n)}{\Gamma(a_0)} \frac{b_0^{a_0}}{b_n^{a_n}} \left(\frac{\tau_0}{\tau_n} \right)^{1/2}$$

where

$$\tau_n = \tau_0 + n \, , \quad a_n = a_0 + n/2 \, , \quad b_n = b_0 + \frac{1}{2} \sum_{i=1}^{n} (y_i - \bar{y})^2 + \frac{\tau_0 n (\bar{y} - \mu_0)^2}{2(\tau_0 + n)} \, .$$

Normal-Gamma example

Accuracy of learnt harmonic mean estimator and sensitivity to prior

Table: Analytic and estimated evidence for various prior sizes τ_0.

Prior size τ_0	10^{-4}	10^{-3}	10^{-2}	10^{-1}	10^{0}
Analytic $\log(z)$	-160.3888	-159.2375	-158.0863	-156.9359	-155.7935
Estimated $\log(\hat{z})$	-160.3883	-159.2370	-158.0851	-156.9359	-155.7921
Error (learnt harmonic mean)	-0.0005	-0.0005	-0.0012	0.0000	-0.0014
Error (original harmonic mean)*	-12.2100	–	-9.7900	-8.5000	-7.1000

*Friel & Wyse (2012)

Normal-Gamma example

Accuracy of learnt harmonic mean estimator and sensitivity to prior

Table: Analytic and estimated evidence for various prior sizes τ_0.

Prior size τ_0	10^{-4}	10^{-3}	10^{-2}	10^{-1}	10^{0}
Analytic $\log(z)$	-160.3888	-159.2375	-158.0863	-156.9359	-155.7935
Estimated $\log(\hat{z})$	-160.3883	-159.2370	-158.0851	-156.9359	-155.7921
Error (learnt harmonic mean)	-0.0005	-0.0005	-0.0012	0.0000	-0.0014
Error (original harmonic mean)*	-12.2100	–	-9.7900	-8.5000	-7.1000

*Friel & Wyse (2012)

Normal-Gamma example

Accuracy of learnt harmonic mean estimator and sensitivity to prior

Table: Analytic and estimated evidence for various prior sizes τ_0.

Prior size τ_0	10^{-4}	10^{-3}	10^{-2}	10^{-1}	10^{0}
Analytic $\log(z)$	-160.3888	-159.2375	-158.0863	-156.9359	-155.7935
Estimated $\log(\hat{z})$	-160.3883	-159.2370	-158.0851	-156.9359	-155.7921
Error (learnt harmonic mean)	**-0.0005**	**-0.0005**	**-0.0012**	**0.0000**	**-0.0014**
Error (original harmonic mean)*	-12.2100	–	-9.7900	-8.5000	-7.1000

*Friel & Wyse (2012)

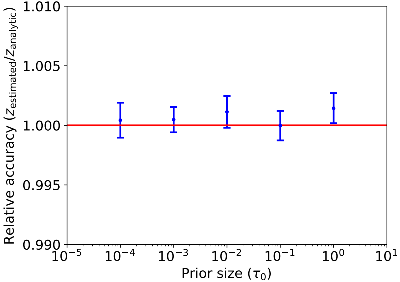


Figure: Accuracy for various prior sizes τ_0.

Normal-Gamma example

Accuracy of learnt harmonic mean estimator and sensitivity to prior

Table: Analytic and estimated evidence for various prior sizes τ_0.

Prior size τ_0	10^{-4}	10^{-3}	10^{-2}	10^{-1}	10^{0}
Analytic $\log(z)$	-160.3888	-159.2375	-158.0863	-156.9359	-155.7935
Estimated $\log(\hat{z})$	-160.3883	-159.2370	-158.0851	-156.9359	-155.7921
Error (learnt harmonic mean)	**-0.0005**	**-0.0005**	**-0.0012**	**0.0000**	**-0.0014**
Error (original harmonic mean)*	**-12.2100**	–	**-9.7900**	**-8.5000**	**-7.1000**

*Friel & Wyse (2012)

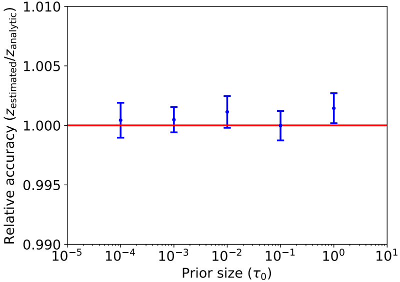


Figure: Accuracy for various prior sizes τ_0.

Non-nested linear regression: Radiata pine example
Data

Radiata pine data-set has become **classical benchmark** for evaluating evidence estimators:

- maximum compression strength parallel to grain y_i,
- density x_i,
- density adjust for resin content z_i,

for $i \in \{1, \ldots, n\}$ where $n = 42$ specimens.

Is density or resin-adjusted density a better predictor of compression strength?

Non-nested linear regression: Radiata pine example
Data

Radiata pine data-set has become **classical benchmark** for evaluating evidence estimators:

- maximum compression strength parallel to grain y_i,
- density x_i,
- density adjust for resin content z_i,

for $i \in \{1, \ldots, n\}$ where $n = 42$ specimens.

Is density or resin-adjusted density a better predictor of compression strength?

Non-nested linear regression: Radiata pine example

Models

Gaussian linear models:

$$M_1: \quad y_i = \alpha + \boxed{\beta(x_i - \bar{x})} + \epsilon_i \,, \quad \epsilon_i \sim \mathsf{N}(0, \tau^{-1}) \,.$$

Density

$$M_2: \quad y_i = \gamma + \boxed{\delta(z_i - \bar{z})} + \eta_i \,, \quad \eta_i \sim \mathsf{N}(0, \lambda^{-1}) \,.$$

Resin-adjusted density

Priors for model 1 (similar for model 2):

$$\alpha \sim \mathsf{N}\big(\mu_\alpha, (r_0\tau)^{-1}\big) \,,$$
$$\beta \sim \mathsf{N}\big(\mu_\beta, (s_0\tau)^{-1}\big) \,,$$
$$\tau \sim \mathsf{Ga}(a_0, b_0) \,,$$

($\mu_\alpha = 3000$, $\mu_\beta = 185$, $r_0 = 0.06$, $s_0 = 6$, $a_0 = 3$, $b_0 = 2 \times 300^2$).

Non-nested linear regression: Radiata pine example

Models

Gaussian linear models:

$$M_1: \quad y_i = \alpha + \underbrace{\boxed{\beta(x_i - \bar{x})}}_{\text{Density}} + \epsilon_i \,, \qquad \epsilon_i \sim \mathrm{N}(0, \tau^{-1}) \,.$$

$$M_2: \quad y_i = \gamma + \underbrace{\boxed{\delta(z_i - \bar{z})}}_{\text{Resin-adjusted density}} + \eta_i \,, \qquad \eta_i \sim \mathrm{N}(0, \lambda^{-1}) \,.$$

Priors for model 1 (similar for model 2):

$$\alpha \sim \mathrm{N}\big(\mu_\alpha, (r_0\tau)^{-1}\big) \,,$$
$$\beta \sim \mathrm{N}\big(\mu_\beta, (s_0\tau)^{-1}\big) \,,$$
$$\tau \sim \mathrm{Ga}(a_0, b_0) \,,$$

($\mu_\alpha = 3000$, $\mu_\beta = 185$, $r_0 = 0.06$, $s_0 = 6$, $a_0 = 3$, $b_0 = 2 \times 300^2$).

Non-nested linear regression: Radiata pine example
Models

Gaussian linear models:

$$M_1 : \quad y_i = \alpha + \boxed{\beta(x_i - \bar{x})} + \epsilon_i \,, \quad \epsilon_i \sim \mathsf{N}(0, \tau^{-1}) \,.$$

Density

$$M_2 : \quad y_i = \gamma + \boxed{\delta(z_i - \bar{z})} + \eta_i \,, \quad \eta_i \sim \mathsf{N}(0, \lambda^{-1}) \,.$$

Resin-adjusted density

Priors for model 1 (similar for model 2):

$$\begin{aligned} \alpha &\sim \mathsf{N}\big(\mu_\alpha, (r_0\tau)^{-1}\big) \,, \\ \beta &\sim \mathsf{N}\big(\mu_\beta, (s_0\tau)^{-1}\big) \,, \\ \tau &\sim \mathsf{Ga}(a_0, b_0) \,, \end{aligned}$$

($\mu_\alpha = 3000$, $\mu_\beta = 185$, $r_0 = 0.06$, $s_0 = 6$, $a_0 = 3$, $b_0 = 2 \times 300^2$).

Non-nested linear regression: Radiata pine example

Models

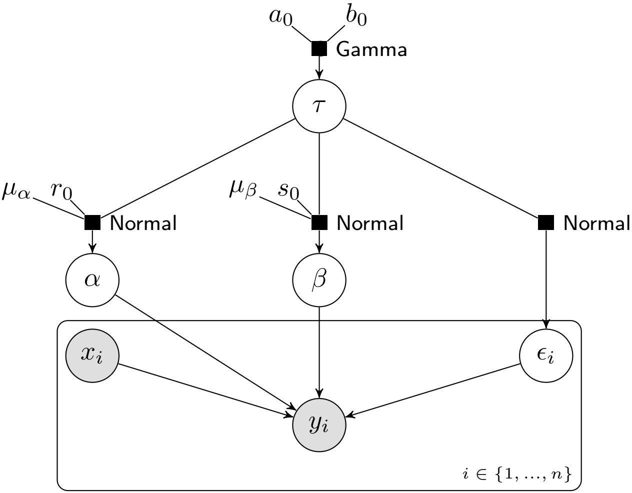


Figure: Graph of hierarchical Bayesian model for Radiata pine example (for model 1; model 2 is similar).

Non-nested linear regression: Radiata pine example

Analytic evidence

Analytic evidence:

$$z = \pi^{-n/2} b_0^{a_0} \frac{\Gamma(a_0 + n/2)}{\Gamma(a_0)} \frac{|Q_0|^{1/2}}{|M|^{1/2}} \left(\boldsymbol{y}^\mathsf{T}\boldsymbol{y} + \boldsymbol{\mu_0}^\mathsf{T} Q_0 \boldsymbol{\mu_0} - \boldsymbol{\nu_0}^\mathsf{T} M \boldsymbol{\nu_0} + 2b_0\right)^{-a_0-n/2}$$

where $\boldsymbol{\mu_0} = (\mu_\alpha, \mu_\beta)^\mathsf{T}$, $Q_0 = \text{diag}(r_0, s_0)$, and $M = X^\mathsf{T} X + Q_0$.

Non-nested linear regression: Radiata pine example

Accuracy of learnt harmonic mean estimator

Table: Analytic and estimated evidence.

	Model M_1 $\log(z_1)$	Model M_2 $\log(z_2)$	$\log \textsf{BF}_{21}$ $= \log(z_2) - \log(z_1)$
Analytic	-310.12833	-301.70460	8.42368
Estimated	-310.12839	-301.70489	8.42350
Error (learnt harmonic mean)	0.00006	0.00029	0.00018
Error (original harmonic mean)*	–	–	0.17372

*Friel & Wyse (2012)

Non-nested linear regression: Radiata pine example

Accuracy of learnt harmonic mean estimator

Table: Analytic and estimated evidence.

	Model M_1 $\log(z_1)$	Model M_2 $\log(z_2)$	$\log \mathsf{BF}_{21}$ $= \log(z_2) - \log(z_1)$
Analytic	-310.12833	-301.70460	8.42368
Estimated	-310.12839	-301.70489	8.42350
Error (learnt harmonic mean)	**0.00006**	**0.00029**	**0.00018**
Error (original harmonic mean)*	–	–	0.17372

*Friel & Wyse (2012)

Non-nested linear regression: Radiata pine example

Accuracy of learnt harmonic mean estimator

Table: Analytic and estimated evidence.

	Model M_1 $\log(z_1)$	Model M_2 $\log(z_2)$	$\log \text{BF}_{21}$ $= \log(z_2) - \log(z_1)$
Analytic	-310.12833	-301.70460	8.42368
Estimated	-310.12839	-301.70489	8.42350
Error (learnt harmonic mean)	**0.00006**	**0.00029**	**0.00018**
Error (original harmonic mean)*	–	–	**0.17372**

*Friel & Wyse (2012)

Outline

Code
Python package: harmonic

Harmonic python package implementing *learnt* harmonic mean estimator.

User-facing features:

- **Ease of use** (modular python package).
- Follow **software engineering best-practice** (*e.g.* well documented, extensive test suite, CI).
- Cython for **speed**.
- **Flexible** choice of sampler (we use **emcee**).
- Bespoke integrated **cross-validation** to select machine learning algorithm and hyperparameters.

Under the hood:

- Bespoke objective functions with **variance penalty** and **regularisation**.
- Solve by bespoke **mini-batch stochastic gradient descent**.

Code

Python package: harmonic

Harmonic python package implementing *learnt* harmonic mean estimator.

User-facing features:

- **Ease of use** (modular python package).
- Follow **software engineering best-practice** (*e.g.* well documented, extensive test suite, CI).
- Cython for **speed**.
- **Flexible** choice of sampler (we use **emcee**).
- Bespoke integrated **cross-validation** to select machine learning algorithm and hyperparameters.

Under the hood:

- Bespoke objective functions with **variance penalty** and **regularisation**.
- Solve by bespoke **mini-batch stochastic gradient descent**.

Code

Python package: harmonic

Harmonic python package implementing *learnt* harmonic mean estimator.

User-facing features:

- **Ease of use** (modular python package).
- Follow **software engineering best-practice** (*e.g.* well documented, extensive test suite, CI).
- Cython for **speed**.
- **Flexible** choice of sampler (we use **emcee**).
- Bespoke integrated **cross-validation** to select machine learning algorithm and hyperparameters.

Under the hood:

- Bespoke objective functions with **variance penalty** and **regularisation**.
- Solve by bespoke **mini-batch stochastic gradient descent**.

Code

Pseudo code example

```
# Import packages
import numpy as np
import emcee
import harmonic
```

Code

Pseudo code example

```
# Import packages
import numpy as np
import emcee
import harmonic
```

```
# Run MCMC sampler
sampler = emcee.EnsembleSampler(nchains, ndim, ln_posterior, args=[args])
sampler.run_mcmc(pos, samples_per_chain)
samples = np.ascontiguousarray(sampler.chain[:,nburn:,:])
lnprob = np.ascontiguousarray(sampler.lnprobability[:,nburn:])
```

Code

Pseudo code example

```
# Import packages
import numpy as np
import emcee
import harmonic
```

```
# Run MCMC sampler
sampler = emcee.EnsembleSampler(nchains, ndim, ln_posterior, args=[args])
sampler.run_mcmc(pos, samples_per_chain)
samples = np.ascontiguousarray(sampler.chain[:,nburn:,:])
lnprob = np.ascontiguousarray(sampler.lnprobability[:,nburn:])
```

```
# Set up chains
chains = harmonic.Chains(ndim)
chains.add_chains_3d(samples, lnprob)
```

Code

Pseudo code example

```
# Import packages
import numpy as np
import emcee
import harmonic
```

```
# Run MCMC sampler
sampler = emcee.EnsembleSampler(nchains, ndim, ln_posterior, args=[args])
sampler.run_mcmc(pos, samples_per_chain)
samples = np.ascontiguousarray(sampler.chain[:,nburn:,:])
lnprob = np.ascontiguousarray(sampler.lnprobability[:,nburn:])
```

```
# Set up chains
chains = harmonic.Chains(ndim)
chains.add_chains_3d(samples, lnprob)
```

```
# Fit model
chains_train, chains_test = harmonic.utils.split_data(chains, train_prop=0.05)
model = harmonic.model.KernelDensityEstimate(ndim, domain, hyper_parameters)
model.fit(chains_train.samples, chains_train.ln_posterior)
```

Code

Pseudo code example

```
# Import packages
import numpy as np
import emcee
import harmonic
```

```
# Run MCMC sampler
sampler = emcee.EnsembleSampler(nchains, ndim, ln_posterior, args=[args])
sampler.run_mcmc(pos, samples_per_chain)
samples = np.ascontiguousarray(sampler.chain[:,nburn:,:])
lnprob = np.ascontiguousarray(sampler.lnprobability[:,nburn:])
```

```
# Set up chains
chains = harmonic.Chains(ndim)
chains.add_chains_3d(samples, lnprob)
```

```
# Fit model
chains_train, chains_test = harmonic.utils.split_data(chains, train_prop=0.05)
model = harmonic.model.KernelDensityEstimate(ndim, domain, hyper_parameters)
model.fit(chains_train.samples, chains_train.ln_posterior)
```

```
# Compute evidence
evidence = harmonic.Evidence(chains_test.nchains, model)
evidence.add_chains(chains_test)
ln_evidence, ln_evidence_std = evidence.compute_ln_evidence()
```

Summary and future work

Problems of harmonic mean estimator can be fixed by re-targeting.

Apply machine learning to approximate optimal importance sampling target.

$\Rightarrow$ ***Learnt* harmonic mean estimator**

Ongoing and future work:

- Numerical optimisations.
- Apply to more examples and push to higher dimensions.
- Make code public.
- Extend general approach to other statistical problems.

Summary and future work

Problems of harmonic mean estimator can be fixed by re-targeting.

Apply machine learning to approximate optimal importance sampling target.

$\Rightarrow$ ***Learnt* harmonic mean estimator**

Ongoing and future work:

- Numerical optimisations.
- Apply to more examples and push to higher dimensions.
- Make code public.
- Extend general approach to other statistical problems.