Sequentially guided MCMC proposals for synthetic likelihoods and correlated synthetic likelihoods.

Umberto Picchini

Dept. Mathematical Sciences, Chalmers and Gothenburg University

@uPicchini

World ISBA meeting, Montreal, 26 June – 1 July 2022

Published online in *Bayesian Analysis* with

- Umberto Simola (Uni. Helsinki);
- Jukka Corander (Uni Oslo; Uni. Helsinki).

Bayesian Analysis (2022) **TBA**, Number TBA, pp. 1–31

Sequentially Guided MCMC Proposals for Synthetic Likelihoods and Correlated Synthetic Likelihoods*

Umberto Picchini[†], Umberto Simola[‡], and Jukka Corander[§]

Also in the Wednesday poster session.

Published online in *Bayesian Analysis* with

- Umberto Simola (Uni. Helsinki);
- Jukka Corander (Uni Oslo; Uni. Helsinki).

Bayesian Analysis (2022) **TBA**, Number TBA, pp. 1–31

Sequentially Guided MCMC Proposals for Synthetic Likelihoods and Correlated Synthetic Likelihoods*

Umberto Picchini[†], Umberto Simola[‡], and Jukka Corander[§]

Also in the Wednesday poster session.

What is the **synthetic likelihood methodology**?

It is a simulation-based inference approach for models with intractable likelihoods.

Just like Approximate Bayesian Comutation (ABC), synthetic likelihood (SL) uses the output of a computer simulator to substitute the unavailable data likelihood $p(y|\theta)$ with an approximation based on informative summary statistics of the data:

$$p(y|\theta) \approx p(s_y|\theta).$$

Therefore with SL the inference about θ will make use of reduced information, as implied by using data-summaries rather than the actual full dataset y.

What is the **synthetic likelihood methodology**?

It is a simulation-based inference approach for models with intractable likelihoods.

Just like Approximate Bayesian Comutation (ABC), synthetic likelihood (SL) uses the output of a computer simulator to substitute the unavailable data likelihood $p(y|\theta)$ with an approximation based on informative summary statistics of the data:

$$p(y|\theta) \approx p(s_y|\theta).$$

Therefore with SL the inference about θ will make use of reduced information, as implied by using data-summaries rather than the actual full dataset y.

What is the **synthetic likelihood methodology**?

It is a simulation-based inference approach for models with intractable likelihoods.

Just like Approximate Bayesian Comutation (ABC), synthetic likelihood (SL) uses the output of a computer simulator to substitute the unavailable data likelihood $p(y|\theta)$ with an approximation based on informative summary statistics of the data:

$$p(y|\theta) \approx p(s_y|\theta).$$

Therefore with SL the inference about θ will make use of reduced information, as implied by using data-summaries rather than the actual full dataset y.

For observed data y, parameters θ and an intractable likelihood $p(y|\theta)$, simulate many (M) times from the stochastic model simulator $\mathcal{S}(\theta)$:

- given θ^*, independently simulate M datasets $y_1^*, ..., y_M^*$, using $\mathcal{S}(\theta^*) \to y_m^*$, $m = 1, ..., M$;
- compute finite-dimensional summary statistics $s_m^* = T(y_m^*)$, $m = 1, ..., M$;
- compute sample mean and covariance matrix

$$\hat{\mu}_{M,\theta^*} = \frac{\sum_{m=1}^{M} s_m^*}{M}, \qquad \hat{\Sigma}_{M,\theta^*} = \frac{\sum_{m=1}^{M}(s_m^* - \hat{\mu}_{\theta^*})(s_m^* - \hat{\mu}_{\theta^*})'}{M-1},$$

- for data y get $s_y = T(y)$ and approximate $p(y|\theta) \approx p(s_y|\theta) \approx \mathcal{N}(s_y; \hat{\mu}_{M,\theta}, \hat{\Sigma}_{M,\theta}) \equiv p_M(s_y|\theta)$.

For observed data y, parameters θ and an intractable likelihood $p(y|\theta)$, simulate many (M) times from the stochastic model simulator $\mathcal{S}(\theta)$:

- given θ^*, independently simulate M datasets $y_1^*, ..., y_M^*$, using $\mathcal{S}(\theta^*) \to y_m^*$, $m = 1, ..., M$;
- compute finite-dimensional summary statistics $s_m^* = T(y_m^*)$, $m = 1, ..., M$;
- compute sample mean and covariance matrix

$$\hat{\mu}_{M,\theta^*} = \frac{\sum_{m=1}^{M} s_m^*}{M}, \qquad \hat{\Sigma}_{M,\theta^*} = \frac{\sum_{m=1}^{M}(s_m^* - \hat{\mu}_{\theta^*})(s_m^* - \hat{\mu}_{\theta^*})'}{M-1},$$

- for data y get $s_y = T(y)$ and approximate $p(y|\theta) \approx p(s_y|\theta) \approx \mathcal{N}(s_y; \hat{\mu}_{M,\theta}, \hat{\Sigma}_{M,\theta}) \equiv p_M(s_y|\theta)$.

For observed data y, parameters θ and an intractable likelihood $p(y|\theta)$, simulate many (M) times from the stochastic model simulator $\mathcal{S}(\theta)$:

- given θ^*, independently simulate M datasets $y_1^*, ..., y_M^*$, using $\mathcal{S}(\theta^*) \to y_m^*$, $m = 1, ..., M$;
- compute finite-dimensional summary statistics $s_m^* = T(y_m^*)$, $m = 1, ..., M$;
- compute sample mean and covariance matrix

$$\hat{\mu}_{M,\theta^*} = \frac{\sum_{m=1}^M s_m^*}{M}, \qquad \hat{\Sigma}_{M,\theta^*} = \frac{\sum_{m=1}^M (s_m^* - \hat{\mu}_{\theta^*})(s_m^* - \hat{\mu}_{\theta^*})'}{M-1},$$

- for data y get $s_y = T(y)$ and approximate $p(y|\theta) \approx p(s_y|\theta) \approx \mathcal{N}(s_y; \hat{\mu}_{M,\theta}, \hat{\Sigma}_{M,\theta}) \equiv p_M(s_y|\theta)$.

For observed data y, parameters θ and an intractable likelihood $p(y|\theta)$, simulate many (M) times from the stochastic model simulator $\mathcal{S}(\theta)$:

- given θ^*, independently simulate M datasets $y_1^*, ..., y_M^*$, using $\mathcal{S}(\theta^*) \to y_m^*$, $m = 1, ..., M$;
- compute finite-dimensional summary statistics $s_m^* = T(y_m^*)$, $m = 1, ..., M$;
- compute sample mean and covariance matrix

$$\hat{\mu}_{M,\theta^*} = \frac{\sum_{m=1}^M s_m^*}{M}, \qquad \hat{\Sigma}_{M,\theta^*} = \frac{\sum_{m=1}^M (s_m^* - \hat{\mu}_{\theta^*})(s_m^* - \hat{\mu}_{\theta^*})'}{M-1},$$

- for data y get $s_y = T(y)$ and approximate $p(y|\theta) \approx p(s_y|\theta) \approx \mathcal{N}(s_y; \hat{\mu}_{M,\theta}, \hat{\Sigma}_{M,\theta}) \equiv p_M(s_y|\theta)$.

For observed data y, parameters θ and an intractable likelihood $p(y|\theta)$, simulate many (M) times from the stochastic model simulator $\mathcal{S}(\theta)$:

- given θ^*, independently simulate M datasets $y_1^*, ..., y_M^*$, using $\mathcal{S}(\theta^*) \rightarrow y_m^*$, $m = 1, ..., M$;
- compute finite-dimensional summary statistics $s_m^* = T(y_m^*)$, $m = 1, ..., M$;
- compute sample mean and covariance matrix

$$\hat{\mu}_{M,\theta^*} = \frac{\sum_{m=1}^{M} s_m^*}{M}, \qquad \hat{\Sigma}_{M,\theta^*} = \frac{\sum_{m=1}^{M}(s_m^* - \hat{\mu}_{\theta^*})(s_m^* - \hat{\mu}_{\theta^*})'}{M-1},$$

- for data y get $s_y = T(y)$ and approximate $p(y|\theta) \approx p(s_y|\theta) \approx \mathcal{N}(s_y; \hat{\mu}_{M,\theta}, \hat{\Sigma}_{M,\theta}) \equiv p_M(s_y|\theta)$.

Synthetic likelihood (SL) is a methodology initially proposed by Wood (2010) and later studied by Price et al. (2018) in a purely Bayesian context.

- The SL $p_M(s_y|\theta)$ is typically plugged into a Metropolis-Hastings sampler in place of $p(y|\theta)$;
- so we obtain a Markov chain with stationary distr.

$$\pi_M(\theta|s_y) \propto p_M(s_y|\theta)\pi(\theta).$$

MCMC via SL often expensive: for each proposed θ it may need $M = 10^3$ or more.

It is especially difficult to initialize SL at some arbitrary initial θ_0 (very large variability in the SL and non-positive definite covariances).

Synthetic likelihood (SL) is a methodology initially proposed by Wood (2010) and later studied by Price et al. (2018) in a purely Bayesian context.

- The SL $p_M(s_y|\theta)$ is typically plugged into a Metropolis-Hastings sampler in place of $p(y|\theta)$;
- so we obtain a Markov chain with stationary distr.

$$\pi_M(\theta|s_y) \propto p_M(s_y|\theta)\pi(\theta).$$

MCMC via SL often expensive: for each proposed θ it may need $M = 10^3$ or more.

It is especially difficult to initialize SL at some arbitrary initial θ_0 (very large variability in the SL and non-positive definite covariances).

Synthetic likelihood (SL) is a methodology initially proposed by Wood (2010) and later studied by Price et al. (2018) in a purely Bayesian context.

- The SL $p_M(s_y|\theta)$ is typically plugged into a Metropolis-Hastings sampler in place of $p(y|\theta)$;
- so we obtain a Markov chain with stationary distr.

$$\pi_M(\theta|s_y) \propto p_M(s_y|\theta)\pi(\theta).$$

MCMC via SL often expensive: for each proposed θ it may need $M = 10^3$ or more.

It is especially difficult to initialize SL at some arbitrary initial θ_0 (very large variability in the SL and non-positive definite covariances).

When θ_0 is far from the bulk of the posterior the chain may fail to mix.

This is due to high-variance in the estimated $\hat{\mu}_{M,\theta_0}$ and $\hat{\Sigma}_{M,\theta_0}$ causing unwanted overestimations of the $p(s_y|\theta_0)$ resulting in many rejections.

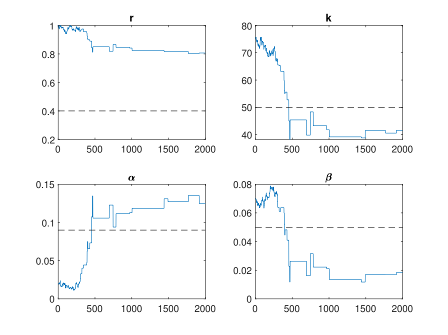


Figure 1: Boom and bust model: traces for robust semiBSL. Dashed lines are true parameter values.

In the figure we even used the robustified semiBSL of An et al (2020)

When θ_0 is far from the bulk of the posterior the chain may fail to mix.

This is due to high-variance in the estimated $\hat{\mu}_{M,\theta_0}$ and $\hat{\Sigma}_{M,\theta_0}$ causing unwanted overestimations of the $p(s_y|\theta_0)$ resulting in many rejections.

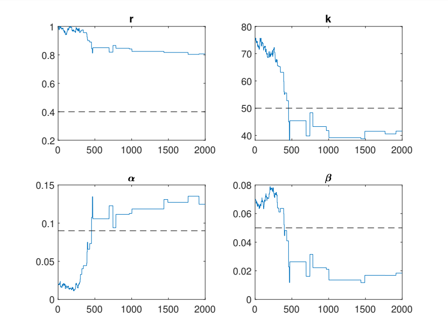


Figure 1: Boom and bust model: traces for robust semiBSL. Dashed lines are true parameter values.

In the figure we even used the robustified semiBSL of An et al (2020)

Most of literature for synthetic likelihood focusses on how to robustify the inference (eg relax the Gaussianity assumption or deal with model misspecification) or reduce the computational cost by regularizing the covariance matrix.

However not much is available in terms of how to help the sampler to more rapidly reach the bulk of the posterior.

An exception is the use of a GP-based surrogate of the logdistances $\log ||s^* - s_y||$, as implemented in the `ELFI` Python package (Lintusaari et al 2018).[1] Then they rapidly find a minimizer of $\log ||s^* - s_y||$ via Bayesian optimization.

This minimizer could then be used to initialize SL.

[1] Lintusaari, et al (2018). ELFI: Engine for likelihood-free inference. Journal of Machine Learning Research, 19(16), 1-7.

Most of literature for synthetic likelihood focusses on how to robustify the inference (eg relax the Gaussianity assumption or deal with model misspecification) or reduce the computational cost by regularizing the covariance matrix.

However not much is available in terms of how to help the sampler to more rapidly reach the bulk of the posterior.

An exception is the use of a GP-based surrogate of the logdistances $\log ||s^* - s_y||$, as implemented in the `ELFI` Python package (Lintusaari et al 2018).[1] Then they rapidly find a minimizer of $\log ||s^* - s_y||$ via Bayesian optimization.

This minimizer could then be used to initialize SL.

[1] Lintusaari, et al (2018). ELFI: Engine for likelihood-free inference. Journal of Machine Learning Research, 19(16), 1-7.

Most of literature for synthetic likelihood focusses on how to robustify the inference (eg relax the Gaussianity assumption or deal with model misspecification) or reduce the computational cost by regularizing the covariance matrix.

However not much is available in terms of how to help the sampler to more rapidly reach the bulk of the posterior.

An exception is the use of a GP-based surrogate of the logdistances $\log ||s^* - s_y||$, as implemented in the `ELFI` Python package (Lintusaari et al 2018).[1] Then they rapidly find a minimizer of $\log ||s^* - s_y||$ via Bayesian optimization.

This minimizer could then be used to initialize SL.

[1] Lintusaari, et al (2018). ELFI: Engine for likelihood-free inference. Journal of Machine Learning Research, 19(16), 1-7.

Initial idea to guide θ

In-nuce, here is the basic idea to construct a proposal sampler $g(\theta|s_y)$ which conditions on data summaries.

- At iteration k of Metropolis-Hastings say we collected M summaries $\{s_k^{*1}, ..., s_k^{*M}\}$ simulated using θ_k^*.
- Compute

$$\bar{s}_k^* = \frac{\sum_{m=1}^{M} s_k^{*m}}{M}.$$

- By CLT, for M large $\bar{s}_k$ is approximately Gaussian.
- After K iterations we have pairs $\{\theta_k^*, \bar{s}_k^*\}_{k=1}^{K}$.

Consider a proposal sampler for the pair $(\theta_k^*, \bar{s}_k^*)$ to be a d-dimensional Gaussian, with

$$(\theta_k^*, \bar{s}_k^*) \sim \mathcal{N}_d(m, S).$$

Initial idea to guide θ

In-nuce, here is the basic idea to construct a proposal sampler $g(\theta|s_y)$ which conditions on data summaries.

- At iteration k of Metropolis-Hastings say we collected M summaries $\{s_k^{*1}, ..., s_k^{*M}\}$ simulated using θ_k^*.
- Compute

$$\bar{s}_k^* = \frac{\sum_{m=1}^{M} s_k^{*m}}{M}.$$

- By CLT, for M large $\bar{s}_k$ is approximately Gaussian.
- After K iterations we have pairs $\{\theta_k^*, \bar{s}_k^*\}_{k=1}^K$.

  Consider a proposal sampler for the pair $(\theta_k^*, \bar{s}_k^*)$ to be a d-dimensional Gaussian, with

$$(\theta_k^*, \bar{s}_k^*) \sim \mathcal{N}_d(m, S).$$

Initial idea to guide θ

In-nuce, here is the basic idea to construct a proposal sampler $g(\theta|s_y)$ which conditions on data summaries.

- At iteration k of Metropolis-Hastings say we collected M summaries $\{s_k^{*1}, ..., s_k^{*M}\}$ simulated using θ_k^*.
- Compute

$$\bar{s}_k^* = \frac{\sum_{m=1}^{M} s_k^{*m}}{M}.$$

- By CLT, for M large $\bar{s}_k$ is approximately Gaussian.
- After K iterations we have pairs $\{\theta_k^*, \bar{s}_k^*\}_{k=1}^K$.

Consider a proposal sampler for the pair $(\theta_k^*, \bar{s}_k^*)$ to be a d-dimensional Gaussian, with

$$(\theta_k^*, \bar{s}_k^*) \sim \mathcal{N}_d(m, S).$$

Initial idea to guide θ

In-nuce, here is the basic idea to construct a proposal sampler $g(\theta|s_y)$ which conditions on data summaries.

- At iteration k of Metropolis-Hastings say we collected M summaries $\{s_k^{*1}, ..., s_k^{*M}\}$ simulated using θ_k^*.
- Compute
$$\bar{s}_k^* = \frac{\sum_{m=1}^{M} s_k^{*m}}{M}.$$
- By CLT, for M large $\bar{s}_k$ is approximately Gaussian.
- After K iterations we have pairs $\{\theta_k^*, \bar{s}_k^*\}_{k=1}^{K}$.

  Consider a proposal sampler for the pair $(\theta_k^*, \bar{s}_k^*)$ to be a d-dimensional Gaussian, with
$$(\theta_k^*, \bar{s}_k^*) \sim \mathcal{N}_d(m, S).$$

Set a d-dimensional mean vector $m \equiv (m_\theta, m_s)$ and the $d \times d$ covariance matrix

$$S \equiv \begin{bmatrix} S_\theta & S_{\theta s} \\ S_{s\theta} & S_s \end{bmatrix}.$$

We estimate m and S using $\{\theta_k^*, \bar{s}_k^*\}_{k=1}^K$ as follows.

Define $x_k := (\theta_k^*, \bar{s}_k^*)$ then

$$\hat{m} = \frac{\sum_{k=1}^K x_k}{K}, \quad \hat{S} = \frac{\sum_{k=1}^K (x_k - \hat{m})(x_k - \hat{m})'}{K-1}.$$

Once $\hat{m}$ and $\hat{S}$ are obtained, we extract the corresponding entries $(\hat{m}_\theta, \hat{m}_s)$ and $\hat{S}_\theta$, $\hat{S}_s$, $\hat{S}_{s\theta}$, $\hat{S}_{\theta s}$. Use well known formulae for conditionals of a multivariate Gaussians, to obtain a

$$\textbf{guided proposal } g(\theta|s_y) \equiv \mathcal{N}(\hat{m}_{\theta|s}, \hat{S}_{\theta|s})$$

$$\hat{m}_{\theta|s} = \hat{m}_\theta + \hat{S}_{\theta s}(\hat{S}_s)^{-1}(s_y - \hat{m}_s)$$
$$\hat{S}_{\theta|s} = \hat{S}_\theta - \hat{S}_{\theta s}(\hat{S}_s)^{-1}\hat{S}_{s\theta}.$$

Set a d-dimensional mean vector $m \equiv (m_\theta, m_s)$ and the $d \times d$ covariance matrix

$$S \equiv \begin{bmatrix} S_\theta & S_{\theta s} \\ S_{s\theta} & S_s \end{bmatrix}.$$

We estimate m and S using $\{\theta_k^*, \bar{s}_k^*\}_{k=1}^K$ as follows.

Define $x_k := (\theta_k^*, \bar{s}_k^*)$ then

$$\hat{m} = \frac{\sum_{k=1}^K x_k}{K}, \quad \hat{S} = \frac{\sum_{k=1}^K (x_k - \hat{m})(x_k - \hat{m})'}{K-1}.$$

Once $\hat{m}$ and $\hat{S}$ are obtained, we extract the corresponding entries $(\hat{m}_\theta, \hat{m}_s)$ and $\hat{S}_\theta$, $\hat{S}_s$, $\hat{S}_{s\theta}$, $\hat{S}_{\theta s}$. Use well known formulae for conditionals of a multivariate Gaussians, to obtain a

guided proposal $g(\theta|s_y) \equiv \mathcal{N}(\hat{m}_{\theta|s}, \hat{S}_{\theta|s})$

$$\hat{m}_{\theta|s} = \hat{m}_\theta + \hat{S}_{\theta s}(\hat{S}_s)^{-1}(s_y - \hat{m}_s)$$
$$\hat{S}_{\theta|s} = \hat{S}_\theta - \hat{S}_{\theta s}(\hat{S}_s)^{-1}\hat{S}_{s\theta}.$$

Set a d-dimensional mean vector $m \equiv (m_\theta, m_s)$ and the $d \times d$ covariance matrix

$$S \equiv \begin{bmatrix} S_\theta & S_{\theta s} \\ S_{s\theta} & S_s \end{bmatrix}.$$

We estimate m and S using $\{\theta_k^*, \bar{s}_k^*\}_{k=1}^K$ as follows.

Define $x_k := (\theta_k^*, \bar{s}_k^*)$ then

$$\hat{m} = \frac{\sum_{k=1}^K x_k}{K}, \quad \hat{S} = \frac{\sum_{k=1}^K (x_k - \hat{m})(x_k - \hat{m})'}{K-1}.$$

Once $\hat{m}$ and $\hat{S}$ are obtained, we extract the corresponding entries $(\hat{m}_\theta, \hat{m}_s)$ and $\hat{S}_\theta$, $\hat{S}_s$, $\hat{S}_{s\theta}$, $\hat{S}_{\theta s}$. Use well known formulae for conditionals of a multivariate Gaussians, to obtain a

$$\textbf{guided proposal } g(\theta|s_y) \equiv \mathcal{N}(\hat{m}_{\theta|s}, \hat{S}_{\theta|s})$$

$$\hat{m}_{\theta|s} = \hat{m}_\theta + \hat{S}_{\theta s}(\hat{S}_s)^{-1}(s_y - \hat{m}_s)$$
$$\hat{S}_{\theta|s} = \hat{S}_\theta - \hat{S}_{\theta s}(\hat{S}_s)^{-1}\hat{S}_{s\theta}.$$

So what we described essentially is a way to **initialize a guided method**.

- We run a non-guided burnin using K iterations of our favourite MCMC sampler;
- we collect the K pairs $\{\theta_k^*, \bar{s}_k^*\}_{k=1}^{K}$;
- at iteration K+1 we build a first guided sampler $g(\theta|s_y) \equiv \mathcal{N}(\hat{m}_{\theta|s}, \hat{S}_{\theta|s})$ ad described.

And then?

How do we keep guiding our sampler for next iterations?

We got inspired by the *Sequential Neural Likelihood* method of Papamakarios-Sterratt-Murray (AISTATS 2019).

So what we described essentially is a way to **initialize a guided method**.

- We run a non-guided burnin using K iterations of our favourite MCMC sampler;
- we collect the K pairs $\{\theta_k^*, \bar{s}_k^*\}_{k=1}^K$;
- at iteration K+1 we build a first guided sampler $g(\theta|s_y) \equiv \mathcal{N}(\hat{m}_{\theta|s}, \hat{S}_{\theta|s})$ ad described.

And then?

How do we keep guiding our sampler for next iterations?

We got inspired by the *Sequential Neural Likelihood* method of Papamakarios-Sterratt-Murray (AISTATS 2019).

So what we described essentially is a way to **initialize a guided method**.

- We run a non-guided burnin using K iterations of our favourite MCMC sampler;
- we collect the K pairs $\{\theta_k^*, \bar{s}_k^*\}_{k=1}^K$;
- at iteration K+1 we build a first guided sampler $g(\theta|s_y) \equiv \mathcal{N}(\hat{m}_{\theta|s}, \hat{S}_{\theta|s})$ ad described.

And then?

How do we keep guiding our sampler for next iterations?

We got inspired by the *Sequential Neural Likelihood* method of Papamakarios-Sterratt-Murray (AISTATS 2019).

So what we described essentially is a way to **initialize a guided method**.

- We run a non-guided burnin using K iterations of our favourite MCMC sampler;
- we collect the K pairs $\{\theta_k^*, \bar{s}_k^*\}_{k=1}^K$;
- at iteration K+1 we build a first guided sampler $g(\theta|s_y) \equiv \mathcal{N}(\hat{m}_{\theta|s}, \hat{S}_{\theta|s})$ ad described.

And then?

How do we keep guiding our sampler for next iterations?

We got inspired by the *Sequential Neural Likelihood* method of Papamakarios-Sterratt-Murray (AISTATS 2019).

Here is the SNL method of Papamakarios-Sterratt-Murray (AISTATS '19).

Input: observed data y, estimator $q_\phi(y^*|\theta)$, number of rounds R, simulations per round N
Set $\hat{\pi}_0(\theta|y) = \pi(\theta)$ and $\mathcal{D} = \{\emptyset\}$

for $r = 1 : R$ **do**
    for $n = 1 : N$ **do**
        sample $\theta_n^* \sim \hat{\pi}_{r-1}(\theta|y)$ with MCMC
        simulate $\mathcal{S}(\theta_n^*) \to y_n^*$ add (θ_n^*, y_n^*) into $\mathcal{D}$
    end for
    (re-)train $q_\phi(y^*\theta)$ on $\mathcal{D}$ and set $\hat{\pi}_r(\theta|y) \propto q_\phi(y|\theta)\pi(\theta)$
end for
Return $\hat{\pi}_R(\theta|y)$

Notice SNL has nothing to do with synthetic likelihoods. It's a generic method.

Here is the SNL method of Papamakarios-Sterratt-Murray (AISTATS ’19).

Input: observed data y, estimator $q_\phi(y^*|\theta)$, number of rounds R, simulations per round N
Set $\hat{\pi}_0(\theta|y) = \pi(\theta)$ and $\mathcal{D} = \{\emptyset\}$
for $r = 1 : R$ **do**
    for $n = 1 : N$ **do**
        sample $\theta_n^* \sim \hat{\pi}_{r-1}(\theta|y)$ with MCMC
        simulate $\mathcal{S}(\theta_n^*) \to y_n^*$ add (θ_n^*, y_n^*) into $\mathcal{D}$
    end for
    (re-)train $q_\phi(y^*\theta)$ on $\mathcal{D}$ and set $\hat{\pi}_r(\theta|y) \propto q_\phi(y|\theta)\pi(\theta)$
end for
Return $\hat{\pi}_R(\theta|y)$

Notice SNL has nothing to do with synthetic likelihoods. It’s a generic method.

Input: K pairs $\{\theta_k^*, \bar{s}_k^*\}_{k=1}^K$ from burnin. Positive integers N and T. Initialize $\mathcal{D} := \{\theta_k^*, \bar{s}_k^*\}_{k=1}^K$.

Construct starting conditional Gaussian proposal g_0 using $\{\theta_k^*, \bar{s}_k^*\}_{k=1}^K$ as already shown. Set $\theta_0 := \theta_K^*$.

for $t = 1 : T$ **do**

  Starting at θ_{t-1} run N MCMC iterations (SL) using g_{t-1}, producing $\{\theta_n^*, \bar{s}_n^*\}_{n=1}^N$.

  Form $\mathcal{D} := \mathcal{D} \cup \{\theta_n^*, \bar{s}_n^*\}_{n=1}^N$, compute $(\hat{m}^{0:t}, \hat{S}^{0:t})$ on $\mathcal{D}$, update $(\hat{m}_{\theta|s}^{0:t}, \hat{S}_{\theta|s}^{0:t})$ to construct $g_t(\theta) = \mathcal{N}(\hat{m}_{\theta|s}^{0:t}, \hat{S}_{\theta|s}^{0:t})$ where

$$
\begin{aligned}
\hat{m}_{\theta|s}^{0:t} &= \hat{m}_{\theta}^{0:t} + \hat{S}_{\theta s}^{0:t}(\hat{S}_{s}^{0:t})^{-1}(s - \hat{m}_{s}^{0:t}) \\
\hat{S}_{\theta|s}^{0:t} &= \hat{S}_{\theta}^{0:t} - \hat{S}_{\theta s}^{0:t}(\hat{S}_{s}^{0:t})^{-1}\hat{S}_{s\theta}^{0:t}.
\end{aligned}
$$

  Set $\theta_t := \theta_N^*$.

end for

Return $\theta_1, ..., \theta_T$ to be provided as input to another adaptive MCMC algorithm for SL.

Best results with $N = 1$, i.e. we **immediately make use of each accepted draw towards guiding the algorithm to modal posterior regions.**

Input: K pairs $\{\theta_k^*, \bar{s}_k^*\}_{k=1}^K$ from burnin. Positive integers N and T. Initialize $\mathcal{D} := \{\theta_k^*, \bar{s}_k^*\}_{k=1}^K$.

Construct starting conditional Gaussian proposal g_0 using $\{\theta_k^*, \bar{s}_k^*\}_{k=1}^K$ as already shown. Set $\theta_0 := \theta_K^*$.

for $t = 1 : T$ **do**

    Starting at θ_{t-1} run N MCMC iterations (SL) using g_{t-1}, producing $\{\theta_n^*, \bar{s}_n^*\}_{n=1}^N$.

    Form $\mathcal{D} := \mathcal{D} \cup \{\theta_n^*, \bar{s}_n^*\}_{n=1}^N$, compute $(\hat{m}^{0:t}, \hat{S}^{0:t})$ on $\mathcal{D}$, update $(\hat{m}_{\theta|s}^{0:t}, \hat{S}_{\theta|s}^{0:t})$ to construct $g_t(\theta) = \mathcal{N}(\hat{m}_{\theta|s}^{0:t}, \hat{S}_{\theta|s}^{0:t})$ where

$$\hat{m}_{\theta|s}^{0:t} = \hat{m}_{\theta}^{0:t} + \hat{S}_{\theta s}^{0:t}(\hat{S}_{s}^{0:t})^{-1}(s - \hat{m}_{s}^{0:t})$$
$$\hat{S}_{\theta|s}^{0:t} = \hat{S}_{\theta}^{0:t} - \hat{S}_{\theta s}^{0:t}(\hat{S}_{s}^{0:t})^{-1}\hat{S}_{s\theta}^{0:t}.$$

    Set $\theta_t := \theta_N^*$.

end for

Return $\theta_1, ..., \theta_T$ to be provided as input to another adaptive MCMC algorithm for SL.

Best results with $N = 1$, i.e. we **immediately make use of each accepted draw towards guiding the algorithm to modal posterior regions.**

I said *towards modal regions*, because this algorithm has no known stationary distribution.

So we will not use it to provide posterior sampling.

It will very rapidly guide the chain towards the mode of $\pi(\theta|s_y)$ with $T \approx 50$ iterations.

After its last iterations we can use the accepted draws to initialize another MCMC sampling with known ergodic properties.

In our example, we use our guided procedure to initialize the classic adaptive MCMC of Haario et al (Bernoulli 2001).

SL MCMC will run much faster after our procedure since we can reduce the number of model simulations M.

I said *towards modal regions*, because this algorithm has no known stationary distribution.

So we will not use it to provide posterior sampling.

It will very rapidly guide the chain towards the mode of $\pi(\theta|s_y)$ with $T \approx 50$ iterations.

After its last iterations we can use the accepted draws to initialize another MCMC sampling with known ergodic properties.

In our example, we use our guided procedure to initialize the classic adaptive MCMC of Haario et al (Bernoulli 2001).

SL MCMC will run much faster after our procedure since we can reduce the number of model simulations M.

I said *towards modal regions*, because this algorithm has no known stationary distribution.

So we will not use it to provide posterior sampling.

It will very rapidly guide the chain towards the mode of $\pi(\theta|s_y)$ with $T \approx 50$ iterations.

After its last iterations we can use the accepted draws to initialize another MCMC sampling with known ergodic properties.

In our example, we use our guided procedure to initialize the classic adaptive MCMC of Haario et al (Bernoulli 2001).

SL MCMC will run much faster after our procedure since we can reduce the number of model simulations M.

Example: g-and-k model

Super-standard toy model in likelihood-free inference.

Its density is unavailable in closed form, but easy to sample from its quantiles function.

Four parameters to infer (A, B, g, k).

Typical challenge with SL: to make the chain mix when the starting parameter θ_0 is far from the data generating value of θ.

Usually chains patterns are very sticky or its impossible to get any acceptance in reasonable time when θ_0 is highly unlikely.

Example: g-and-k model

Super-standard toy model in likelihood-free inference.

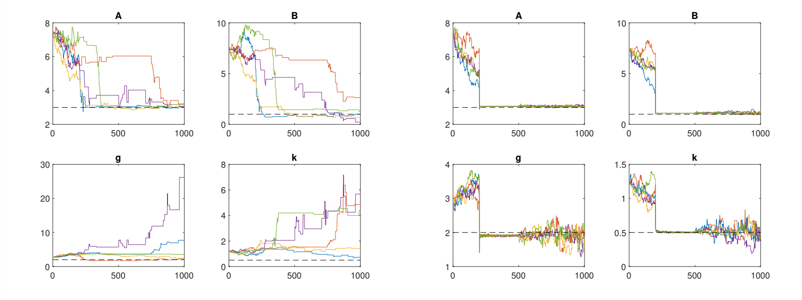
Its density is unavailable in closed form, but easy to sample from its quantiles function.

Four parameters to infer (A, B, g, k).

Typical challenge with SL: to make the chain mix when the starting parameter θ_0 is far from the data generating value of θ.

Usually chains patterns are very sticky or its impossible to get any acceptance in reasonable time when θ_0 is highly unlikely.

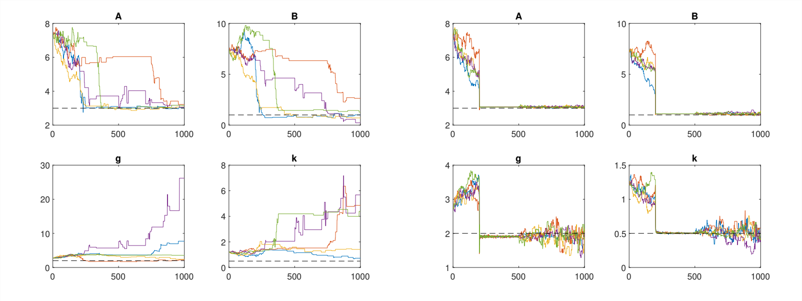
We consider five independent runs starting at a relatively remote θ_0.



(a) non-guided approach **(b)** guided approach

Figure 2: (a) non-guided approach using the Haario et al 2001 sampler. (b) uses the guided sampler from iteration 200 to 500. We display the first 1,000 iterations to emphasize the effect of the guided sampler. The black dashed lines mark ground-truth parameters.

We consider five independent runs starting at a relatively remote θ_0.



(a) non-guided approach

(b) guided approach

Figure 2: (a) non-guided approach using the Haario et al 2001 sampler. (b) uses the guided sampler from iteration 200 to 500. We display the first 1,000 iterations to emphasize the effect of the guided sampler. The black dashed lines mark ground-truth parameters.

Example: bimodal target

Here our likelihood is a 2 components bivariate Gaussian mixture *(way more complex case-studies are in the paper)*.

Of course we absolutely do not need synthetic likelihoods here, but it's a useful case study.

$$y \sim 0.5\mathcal{N}(\mu_1, \Sigma_1) + 0.5\mathcal{N}(\mu_2, \Sigma_2).$$

We wish to fit its mean values $\mu_1 = (\mu_{11}, \mu_{12})^{\mathrm{T}}$, $\mu_2 = (\mu_{21}, \mu_{22})^{\mathrm{T}}$.

The only unknowns are the two vectors μ_1 and μ_2.

Example: bimodal target

Here our likelihood is a 2 components bivariate Gaussian mixture *(way more complex case-studies are in the paper)*.

Of course we absolutely do not need synthetic likelihoods here, but it's a useful case study.

$$y \sim 0.5\mathcal{N}(\mu_1, \Sigma_1) + 0.5\mathcal{N}(\mu_2, \Sigma_2).$$

We wish to fit its mean values $\mu_1 = (\mu_{11}, \mu_{12})^{\mathrm{T}}$, $\mu_2 = (\mu_{21}, \mu_{22})^{\mathrm{T}}$.

The only unknowns are the two vectors μ_1 and μ_2.

At each proposed $\theta = (\mu_1, \mu_2)$:

- we simulate 5,000 data points from the mixture;
- we fit the 5,000 points with a bivariate 2-components Gaussian mixture: the fitted means are used as summary statistics.

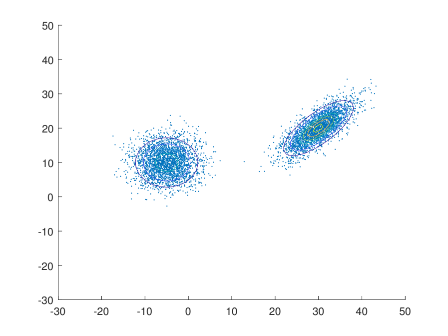
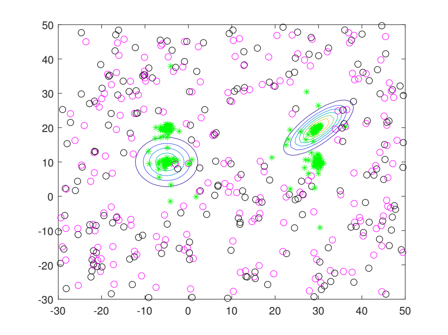


Figure 3: Data

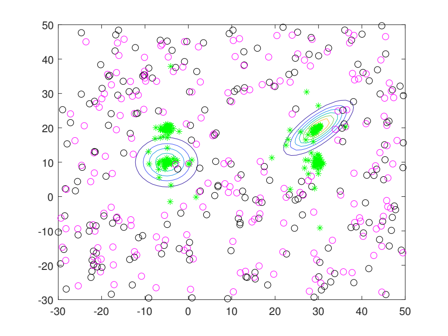
We pick 100 starting values for $\mu_1 = (\mu_{11}, \mu_{12})^{\mathrm{T}}$, $\mu_2 = (\mu_{21}, \mu_{22})^{\mathrm{T}}$, randomly and uniformly in the square below and run 100 MCMC chains.



Starting values are the black circles. After 49 iterations of random walk proposals we get the circles in magenta. These are followed by a single guided iteration in green.

Evidently **a single guided iteration is way more effective than the previous 49 standard random walks.**

We pick 100 starting values for $\mu_1 = (\mu_{11}, \mu_{12})^{\mathrm{T}}$, $\mu_2 = (\mu_{21}, \mu_{22})^{\mathrm{T}}$, randomly and uniformly in the square below and run 100 MCMC chains.



Starting values are the black circles. After 49 iterations of random walk proposals we get the circles in magenta. These are followed by a single guided iteration in green.

Evidently **a single guided iteration is way more effective than the previous 49 standard random walks.**

[In our paper we also have another idea that considerably helped chain mixing: this is about correlating numerator and denominator in the MH ratio. No time to discuss this, but feel free to ask also at the Wednesday poster session.]

Our guided method appears to work also with summaries that are highly non-Gaussian, which is reassuring.

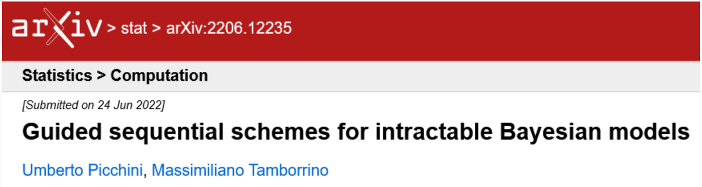
[In our paper we also have another idea that considerably helped chain mixing: this is about correlating numerator and denominator in the MH ratio. No time to discuss this, but feel free to ask also at the Wednesday poster session.]

Our guided method appears to work also with summaries that are highly non-Gaussian, which is reassuring.

This work is published on *Bayesian Analysis*.

I am ready to chat about this also at the poster session.

This work has inspired a paper on **guided sequential ABC methods** with Massimiliano Tamborrino (Warwick), that Massimiliano presented on Monday (can be found on arXiv).


arXiv > stat > arXiv:2206.12235

Statistics > Computation

[Submitted on 24 Jun 2022]

Guided sequential schemes for intractable Bayesian models

Umberto Picchini, Massimiliano Tamborrino

Thank you

@uPicchini

This work is published on *Bayesian Analysis*.

I am ready to chat about this also at the poster session.

This work has inspired a paper on **guided sequential ABC methods** with Massimiliano Tamborrino (Warwick), that Massimiliano presented on Monday (can be found on arXiv).

arXiv > stat > arXiv:2206.12235

Statistics > Computation

[Submitted on 24 Jun 2022]

Guided sequential schemes for intractable Bayesian models

Umberto Picchini, Massimiliano Tamborrino

Thank you

@uPicchini

This work is published on *Bayesian Analysis*.

I am ready to chat about this also at the poster session.

This work has inspired a paper on **guided sequential ABC methods** with Massimiliano Tamborrino (Warwick), that Massimiliano presented on Monday (can be found on arXiv).

arXiv > stat > arXiv:2206.12235

Statistics > Computation

[Submitted on 24 Jun 2022]

Guided sequential schemes for intractable Bayesian models

Umberto Picchini, Massimiliano Tamborrino

Thank you

@uPicchini

Appendix

Correlated synthetic likelihoods

Denote with U the vector of all "auxiliary variables", i.e. pseudorandom numbers (typically standard Gaussian or uniform) that are necessary to produce a non-negative likelihood approximation $\hat{p}(s_y|\theta, \mathrm{U})$ at a given parameter θ.

In Tran et al. (2016) the set U is divided into G blocks $U = (U_{(1)}, ..., U_{(G)})$, and one of these blocks is updated jointly with θ in each MCMC iteration.

We can write the acceptance probability as

$$\alpha = \min\left\{1, \frac{p_M\left(s|\theta^p, \mathrm{U}^c_{(1)}, ..., \mathrm{U}^c_{(k-1)}, \mathrm{U}^p_{(k)}, \mathrm{U}^c_{(k+1)}, ..., \mathrm{U}^c_{(G)}\right)\pi\left(\theta^p\right)}{p_M\left(s|\theta^c, \mathrm{U}^c_{(1)}, ..., \mathrm{U}^c_{(k-1)}, \mathrm{U}^c_{(k)}, \mathrm{U}^c_{(k+1)}, ..., \mathrm{U}^c_{(G)}\right)\pi\left(\theta^c\right)} \frac{g\left(\theta^c|\theta^p\right)}{g\left(\theta^p|\theta^c\right)}\right\} \tag{1}$$

which we therefore call "correlated synthetic likelihood" (CSL) approach.

If we take p_M to be the unbiased SL of Price et al (2018), then we are still targeting the exact $\pi(\theta|s_y)$ (Tran et al. 2016).

Left: standard SL.

Right: correlated SL.