Least squares variational inference

NeurIPS 2025

Yvann Le Fay

CREST-ENSAE, Institut Polytechnique de Paris

Joint work with


Nicolas Chopin (CREST-ENSAE, IP Paris), Simon Barthelmé (CNRS)

Table of Contents

Given an un-normalised target π:

1. Sample approximately from π

Given an un-normalised target π:

1. Sample approximately from π
2. Estimate expectation under $\bar{\pi}$, with $\bar{\pi} = \pi / \int \pi$ (the normalised version of π)

Given an un-normalised target π:

1. Sample approximately from π
2. Estimate expectation under $\bar{\pi}$, with $\bar{\pi} = \pi / \int \pi$ (the normalised version of π)

Two possibilities:

1. Asymptotically exact methods.
2. Approximating methods (Variational inference).

Parametric Variational Inference

Given an un-normalised target π, compute:

$$\arg \min_{q \in \mathcal{Q}} \mathrm{KL}\left(q | \bar{\pi}\right)$$

Parametric Variational Inference

Given an un-normalised target π, compute:

$$\arg\min_{q \in \mathcal{Q}} \mathrm{KL}\left(q|\bar{\pi}\right)$$

where

- $\mathcal{Q}$: parametric family ($\mathcal{Q} = \{q_\eta : \eta \in \mathcal{V}\}$)
- $\mathrm{KL}\left(q|\bar{\pi}\right) := \int q \log(q/\bar{\pi})$

Existing methods and their limitations

The minimisation is carried out through gradient-based procedures:

Existing methods and their limitations

The minimisation is carried out through gradient-based procedures:

- it requires estimating gradients of $\eta \mapsto \mathbb{E}_{q_\eta}[h]$ (for some h)

Existing methods and their limitations

The minimisation is carried out through gradient-based procedures:

- it requires estimating gradients of $\eta \mapsto \mathbb{E}_{q_\eta}[h]$ (for some h)
  - via the log-derivative trick $\nabla_\eta \mathbb{E}_{q_\eta}[\cdot] = \mathbb{E}_\eta[(\nabla_\eta \log q_\eta) h]$ (BBVI, Giordano et al. (2024))

Existing methods and their limitations

The minimisation is carried out through gradient-based procedures:

- it requires estimating gradients of $\eta \mapsto \mathbb{E}_{q_\eta}[h]$ (for some h)
  - via the log-derivative trick $\nabla_\eta \mathbb{E}_{q_\eta}[\cdot] = \mathbb{E}_\eta[(\nabla_\eta \log q_\eta) h]$ (BBVI, Giordano et al. (2024))
    - known to be noisy

Existing methods and their limitations

The minimisation is carried out through gradient-based procedures:

- it requires estimating gradients of $\eta \mapsto \mathbb{E}_{q_\eta}[h]$ (for some h)
  - via the log-derivative trick $\nabla_\eta \mathbb{E}_{q_\eta}[\cdot] = \mathbb{E}_\eta[(\nabla_\eta \log q_\eta)h]$ (BBVI, Giordano et al. (2024))
    - known to be noisy
  - after performing a reparameterisation (ADVI, Kucukelbir et al. (2017))

Existing methods and their limitations

The minimisation is carried out through gradient-based procedures:

- it requires estimating gradients of $\eta \mapsto \mathbb{E}_{q_\eta}[h]$ (for some h)
  - via the log-derivative trick $\nabla_\eta \mathbb{E}_{q_\eta}[\cdot] = \mathbb{E}_\eta[(\nabla_\eta \log q_\eta)h]$ (BBVI, Giordano et al. (2024))
    - known to be noisy
  - after performing a reparameterisation (ADVI, Kucukelbir et al. (2017))
    - requires $\log \pi$ to be differentiable,
    - $x \sim \pi$ is not continuous (e.g., Bernoulli)

Existing methods and their limitations

The minimisation is carried out through gradient-based procedures:

- it requires estimating gradients of $\eta \mapsto \mathbb{E}_{q_\eta}[h]$ (for some h)
  - via the log-derivative trick $\nabla_\eta \mathbb{E}_{q_\eta}[\cdot] = \mathbb{E}_\eta[(\nabla_\eta \log q_\eta)h]$ (BBVI, Giordano et al. (2024))
    - known to be noisy
  - after performing a reparameterisation (ADVI, Kucukelbir et al. (2017))
    - requires $\log \pi$ to be differentiable,
    - $x \sim \pi$ is not continuous (e.g., Bernoulli)
- Diagnosing convergence may be tricky (Welandawe et al., 2024).

Our approach

- Assume $\mathcal{Q}$ is exponential (including Gaussians, Bernoulli, Beta, etc.):

$$q_\eta(x) \propto \exp\left\{\eta^T s(x)\right\}.$$

Our approach

- Assume $\mathcal{Q}$ is exponential (including Gaussians, Bernoulli, Beta, etc.):

$$q_\eta(x) \propto \exp\left\{\eta^T s(x)\right\}.$$

- **Core idea:** The log density of the variational family is linear in $s \Rightarrow$ Amounts to linear regression $\log \pi(X) \approx \eta^\top s(X)$.
- gradient-free, fast convergence, supporting theory.

Our approach

- Assume $\mathcal{Q}$ is exponential (including Gaussians, Bernoulli, Beta, etc.):

$$q_\eta(x) \propto \exp\left\{\eta^T s(x)\right\}.$$

- **Core idea:** The log density of the variational family is linear in $s \Rightarrow$ Amounts to linear regression $\log \pi(X) \approx \eta^\top s(X)$.
- gradient-free, fast convergence, supporting theory.
- Connection to linear regression is not new: see Salimans and Knowles (2013)

Table of Contents

Minimising over a set of un-normalised densities

Consider an exponential family of *un-normalised* densities:

$$q_\eta(x) := \exp\{\eta^\top s(x)\}, \qquad \eta \in \mathcal{V} := \{\eta : Z(q_\eta) < \infty\}$$

with $Z(q) := \int_{\mathcal{X}} q$, and $s : \mathcal{X} \to \mathbb{R}^m$ such that

Minimising over a set of un-normalised densities

Consider an exponential family of *un-normalised* densities:

$$q_\eta(x) := \exp\{\eta^\top s(x)\}, \qquad \eta \in \mathcal{V} := \{\eta : Z(q_\eta) < \infty\}$$

with $Z(q) := \int_{\mathcal{X}} q$, and $s : \mathcal{X} \to \mathbb{R}^m$ such that

$$s(x) = \begin{pmatrix} 1 \\ \bar{s}(x) \end{pmatrix},$$

Minimising over a set of un-normalised densities

Consider an exponential family of *un-normalised* densities:

$$q_\eta(x) := \exp\{\eta^\top s(x)\}, \qquad \eta \in \mathcal{V} := \{\eta : Z(q_\eta) < \infty\}$$

with $Z(q) := \int_{\mathcal{X}} q$, and $s : \mathcal{X} \to \mathbb{R}^m$ such that

$$s(x) = \begin{pmatrix} 1 \\ \bar{s}(x) \end{pmatrix},$$

Replace KL by un-normalised KL (Minka, 2005):

$$\mathrm{uKL}(q \mid \pi) := \int q \log\left(\frac{q}{\pi}\right) + Z(\pi) - Z(q)$$

LS mapping

Given $f = \log \pi$, define $\phi : \mathcal{V} \to \mathbb{R}^m$ as:

$$\phi(\eta) := \text{argmin}_{\beta \in \mathbb{R}^m} \mathbb{E}_\eta \left[\left\{ f(x) - \beta^T s(x) \right\}^2 \right]$$

LS mapping

Given $f = \log \pi$, define $\phi : \mathcal{V} \to \mathbb{R}^m$ as:

$$\begin{aligned}\phi(\eta) &:= \operatorname{argmin}_{\beta \in \mathbb{R}^m} \mathbb{E}_\eta \left[\left\{ f(x) - \beta^T s(x) \right\}^2 \right] \\ &= \left(\mathbb{E}_\eta \left[s s^\top \right] \right)^{-1} \mathbb{E}_\eta \left[f s \right].\end{aligned}$$

LS mapping

Given $f = \log \pi$, define $\phi : \mathcal{V} \to \mathbb{R}^m$ as:

$$\begin{aligned}\phi(\eta) &:= \operatorname{argmin}_{\beta \in \mathbb{R}^m} \mathbb{E}_\eta \left[\left\{ f(x) - \beta^T s(x) \right\}^2 \right] \\ &= \left(\mathbb{E}_\eta \left[s s^\top \right] \right)^{-1} \mathbb{E}_\eta \left[f s \right].\end{aligned}$$

Proposition

If $\nabla_\eta \, \mathrm{uKL}(q_\eta \mid \pi) = 0$, then η is a fixed-point of ϕ: $\phi(\eta) = \eta$.

Exact LSVI: fixed-point iteration

Iterate:

$$\eta_{t+1} = \phi(\eta_t)$$

Exact LSVI: fixed-point iteration

Iterate:

$$\eta_{t+1} = \phi(\eta_t)$$

or more generally, to make sure that $\eta_{t+1} \in \mathcal{V}$: find ε_t s.t

$$\eta_{t+1} = \varepsilon_t \phi(\eta_t) + (1 - \varepsilon_t)\eta_t \in \mathcal{V},$$

Exact LSVI: fixed-point iteration

Iterate:

$$\eta_{t+1} = \phi(\eta_t)$$

or more generally, to make sure that $\eta_{t+1} \in \mathcal{V}$: find ε_t s.t

$$\eta_{t+1} = \varepsilon_t \phi(\eta_t) + (1 - \varepsilon_t)\eta_t \in \mathcal{V},$$

Connexion with tempering: update is equivalent to LSVI towards $q_{\eta_t}^{1-\varepsilon_t} \pi^{\varepsilon_t}$.

exact LSVI = natural gradient descent

LSVI iteration is derived from the first-order condition on $\eta \mapsto \mathsf{uKL}(q_\eta, \pi)$

exact LSVI = natural gradient descent

LSVI iteration is derived from the first-order condition on $\eta \mapsto \mathrm{uKL}(q_\eta, \pi)$

Proposition

The LSVI iteration is equivalent to NGD:

$$\eta_{t+1} = \eta_t - \frac{\varepsilon_t}{Z_{\eta_t}} F_{\eta_t}^{-1} \nabla_\eta l(\eta_t),$$

where $l(\eta) := \mathrm{uKL}(q_\eta, \pi)$, $F_\eta := \mathbb{E}_\eta[ss^\top]$ is the Fisher information matrix.

exact LSVI = natural gradient descent

LSVI iteration is derived from the first-order condition on $\eta \mapsto \mathrm{uKL}(q_\eta, \pi)$

Proposition

The LSVI iteration is equivalent to NGD:

$$\eta_{t+1} = \eta_t - \frac{\varepsilon_t}{Z_{\eta_t}} F_{\eta_t}^{-1} \nabla_\eta l(\eta_t),$$

where $l(\eta) := \mathrm{uKL}(q_\eta, \pi)$, $F_\eta := \mathbb{E}_\eta[ss^\top]$ is the Fisher information matrix.

Actual algorithm

Require: $\eta_0 \in \mathcal{V}$, $N \geq 1$, $f := \log \pi$

1: $\hat{\eta}_0 \leftarrow \eta_0$

Actual algorithm

Require: $\eta_0 \in \mathcal{V}$, $N \geq 1$, $f := \log \pi$

1: $\hat{\eta}_0 \leftarrow \eta_0$
2: **while** not converged **do**
3: $\quad X_1, \ldots, X_N \overset{\text{i.i.d.}}{\sim} q_{\hat{\eta}_t}$

Actual algorithm

Require: $\eta_0 \in \mathcal{V}$, $N \geq 1$, $f := \log \pi$

1: $\hat{\eta}_0 \leftarrow \eta_0$
2: **while** not converged **do**
3: $\quad X_1, \ldots, X_N \overset{\text{i.i.d.}}{\sim} q_{\hat{\eta}_t}$
4: $\quad \hat{\eta}'_{t+1} \leftarrow \texttt{OLS}\left(\{s(X_n), f(X_n)\}_{n=1,\ldots,N}\right)$ $\quad$ {Linear regression}
5: $\quad \varepsilon_t \leftarrow \texttt{stepsize}(\hat{F}, \hat{z}, \hat{\eta}'_{t+1}, \hat{\eta}_t, X, \ldots)$

Actual algorithm

Require: $\eta_0 \in \mathcal{V}$, $N \geq 1$, $f := \log \pi$

$\hat{\eta}_0 \leftarrow \eta_0$

while not converged **do**

  $X_1, \ldots, X_N \overset{\text{i.i.d.}}{\sim} q_{\hat{\eta}_t}$

  $\hat{\eta}'_{t+1} \leftarrow \texttt{OLS}\left(\{s(X_n), f(X_n)\}_{n=1,\ldots,N}\right)$ {Linear regression}

  $\varepsilon_t \leftarrow \texttt{stepsize}(\hat{F}, \hat{z}, \hat{\eta}'_{t+1}, \hat{\eta}_t, X, \ldots)$

  $\hat{\eta}_{t+1} \leftarrow \varepsilon_t \hat{\eta}'_{t+1} + (1 - \varepsilon_t)\hat{\eta}_t$

end while

Table of Contents

An example: Gaussian

Let $\mathcal{Q}$ be the family of (un-normalised) Gaussian densities over $\mathbb{R}^d$.

An example: Gaussian

Let $\mathcal{Q}$ be the family of (un-normalised) Gaussian densities over $\mathbb{R}^d$.

$$s(x) = (1, \underbrace{x_1, \ldots, x_d}_{\text{1st order terms}}, \underbrace{x_1^2, x_1 x_2, \ldots, x_d^2}_{\text{2nd order}})^\top, \tag{1}$$

of size $m = 1 + d + d^2$.

An example: Gaussian

Let $\mathcal{Q}$ be the family of (un-normalised) Gaussian densities over $\mathbb{R}^d$.

$$s(x) = (1, \underbrace{x_1, \ldots, x_d}_{\text{1st order terms}}, \underbrace{x_1^2, x_1 x_2, \ldots, x_d^2}_{\text{2nd order}})^\top, \tag{1}$$

of size $m = 1 + d + d^2$. A natural parameter $\eta \in \mathcal{V}$

$$\eta = \begin{pmatrix} \eta^{(0)} \\ \eta^{(1)} \\ \eta^{(2)} \end{pmatrix} \begin{matrix} \} \, 1 \\ \} \, d \\ \} \, d^2 \end{matrix} \tag{2}$$

An example: Gaussian

Let $\mathcal{Q}$ be the family of (un-normalised) Gaussian densities over $\mathbb{R}^d$.

$$s(x) = (1, \underbrace{x_1, \ldots, x_d}_{\text{1st order terms}}, \underbrace{x_1^2, x_1 x_2, \ldots, x_d^2}_{\text{2nd order}})^\top, \tag{1}$$

of size $m = 1 + d + d^2$. A natural parameter $\eta \in \mathcal{V}$

$$\eta = \begin{pmatrix} \eta^{(0)} \\ \eta^{(1)} \\ \eta^{(2)} \end{pmatrix} \begin{matrix} \} \, 1 \\ \} \, d \\ \} \, d^2 \end{matrix} \tag{2}$$

defines a unique Gaussian

$$(\mu, \Sigma) = \left(-\frac{1}{2} \eta^{(2),-1} \eta^{(1)}, -\frac{1}{2} \operatorname{unvec}(\eta^{(2)})^{-1} \right). \tag{3}$$

(Costly) algorithm for Gaussian

Require: $\mu_0, \Sigma_0 \succ 0,\ N \geq 1$

1: $(\hat{\mu}_0, \hat{\Sigma}_0) \leftarrow (\mu_0, \Sigma_0)$

2: $\hat{\eta}_0 \leftarrow (-\infty, -\Sigma^{-1}\mu, -\frac{1}{2}\operatorname{vec}\hat{\Sigma}^{-1})$

(Costly) algorithm for Gaussian

Require: $\mu_0, \Sigma_0 \succ 0,\ N \geq 1$

1: $(\hat{\mu}_0, \hat{\Sigma}_0) \leftarrow (\mu_0, \Sigma_0)$
2: $\hat{\eta}_0 \leftarrow (-\infty, -\Sigma^{-1}\mu, -\frac{1}{2}\operatorname{vec}\hat{\Sigma}^{-1})$
3: **while** not converged **do**
4: $\quad X_1, \ldots, X_N \sim \mathcal{N}(\hat{\mu}_t, \hat{\Sigma}_t)$
5: $\quad \hat{\eta}'_{t+1} \leftarrow (N^{-1}\sum_{i=1}^{N} ss^\top(X_i))^{-1}(N^{-1}\sum_{i=1}^{N} s(X_i)f(X_i))$
$\quad \{\text{OLS}\}$

(Costly) algorithm for Gaussian

Require: $\mu_0, \Sigma_0 \succ 0,\ N \geq 1$

1: $(\hat{\mu}_0, \hat{\Sigma}_0) \leftarrow (\mu_0, \Sigma_0)$
2: $\hat{\eta}_0 \leftarrow (-\infty, -\Sigma^{-1}\mu, -\frac{1}{2}\operatorname{vec}\hat{\Sigma}^{-1})$
3: **while** not converged **do**
4: $\quad X_1, \ldots, X_N \sim \mathcal{N}(\hat{\mu}_t, \hat{\Sigma}_t)$
5: $\quad \hat{\eta}'_{t+1} \leftarrow (N^{-1}\sum_{i=1}^{N} ss^\top(X_i))^{-1}(N^{-1}\sum_{i=1}^{N} s(X_i)f(X_i))$ $\{\text{OLS}\}$
6: $\quad \varepsilon_t \leftarrow$ `proper stepsize`
7: $\quad \hat{\eta}_{t+1} \leftarrow \varepsilon_t \hat{\eta}'_{t+1} + (1 - \varepsilon_t)\hat{\eta}_t$

(Costly) algorithm for Gaussian

Require: $\mu_0, \Sigma_0 \succ 0,\ N \geq 1$

$(\hat{\mu}_0, \hat{\Sigma}_0) \leftarrow (\mu_0, \Sigma_0)$

$\hat{\eta}_0 \leftarrow (-\infty, -\Sigma^{-1}\mu, -\frac{1}{2}\operatorname{vec}\hat{\Sigma}^{-1})$

while not converged **do**

  $X_1, \ldots, X_N \sim \mathcal{N}(\hat{\mu}_t, \hat{\Sigma}_t)$

  $\hat{\eta}'_{t+1} \leftarrow (N^{-1}\sum_{i=1}^{N} ss^\top(X_i))^{-1}(N^{-1}\sum_{i=1}^{N} s(X_i)f(X_i))$ $\{\text{OLS}\}$

  $\varepsilon_t \leftarrow$ `proper stepsize`

  $\hat{\eta}_{t+1} \leftarrow \varepsilon_t \hat{\eta}'_{t+1} + (1-\varepsilon_t)\hat{\eta}_t$

  $\hat{\mu}_{t+1} \leftarrow -\frac{1}{2}\hat{\eta}^{(2),-1}_{t+1}\hat{\eta}^{(1)}_{t+1}$

  $\hat{\Sigma}_{t+1} \leftarrow -\frac{1}{2}\operatorname{unvec}(\hat{\eta}^{(2)}_{t+1})^{-1}$

end while

(Costly) algorithm for Gaussian

Require: $\mu_0, \Sigma_0 \succ 0,\ N \geq 1$

1: $(\hat{\mu}_0, \hat{\Sigma}_0) \leftarrow (\mu_0, \Sigma_0)$
2: $\hat{\eta}_0 \leftarrow (-\infty, -\Sigma^{-1}\mu, -\frac{1}{2}\operatorname{vec}\hat{\Sigma}^{-1})$
3: **while** not converged **do**
4: $\quad X_1, \ldots, X_N \sim \mathcal{N}(\hat{\mu}_t, \hat{\Sigma}_t)$
5: $\quad \hat{\eta}'_{t+1} \leftarrow (N^{-1}\sum_{i=1}^{N} ss^\top(X_i))^{-1}(N^{-1}\sum_{i=1}^{N} s(X_i)f(X_i))$ $\{\text{OLS}\}$
6: $\quad \varepsilon_t \leftarrow$ `proper stepsize`
7: $\quad \hat{\eta}_{t+1} \leftarrow \varepsilon_t \hat{\eta}'_{t+1} + (1-\varepsilon_t)\hat{\eta}_t$
8: $\quad \hat{\mu}_{t+1} \leftarrow -\frac{1}{2}\hat{\eta}^{(2),-1}_{t+1}\hat{\eta}^{(1)}_{t+1}$
9: $\quad \hat{\Sigma}_{t+1} \leftarrow -\frac{1}{2}\operatorname{unvec}(\hat{\eta}^{(2)}_{t+1})^{-1}$
10: **end while**

Bottleneck!

(Costly) algorithm for Gaussian

Require: $\mu_0, \Sigma_0 \succ 0$, $N \geq 1$

1: $(\hat{\mu}_0, \hat{\Sigma}_0) \leftarrow (\mu_0, \Sigma_0)$
2: $\hat{\eta}_0 \leftarrow (-\infty, -\Sigma^{-1}\mu, -\frac{1}{2}\operatorname{vec}\hat{\Sigma}^{-1})$
3: **while** not converged **do**
4: $\quad X_1, \ldots, X_N \sim \mathcal{N}(\hat{\mu}_t, \hat{\Sigma}_t)$
5: $\quad \hat{\eta}'_{t+1} \leftarrow (N^{-1}\sum_{i=1}^{N} ss^\top(X_i))^{-1}(N^{-1}\sum_{i=1}^{N} s(X_i)f(X_i))$ $\{\text{OLS}\}$
6: $\quad \varepsilon_t \leftarrow$ `proper stepsize`
7: $\quad \hat{\eta}_{t+1} \leftarrow \varepsilon_t \hat{\eta}'_{t+1} + (1-\varepsilon_t)\hat{\eta}_t$
8: $\quad \hat{\mu}_{t+1} \leftarrow -\frac{1}{2}\hat{\eta}^{(2),-1}_{t+1}\hat{\eta}^{(1)}_{t+1}$
9: $\quad \hat{\Sigma}_{t+1} \leftarrow -\frac{1}{2}\operatorname{unvec}(\hat{\eta}^{(2)}_{t+1})^{-1}$
10: **end while**

Bottleneck! Requires inverting a $m \times m$ matrix, $m = \mathcal{O}(d^2)$, $\mathcal{O}(d^6)$ complexity.

Complexity

Since LSVI relies on a linear regression with $m = \dim(s)$ regressors, it has complexity $\mathcal{O}(m^3)$.

Complexity

Since LSVI relies on a linear regression with $m = \dim(s)$ regressors, it has complexity $\mathcal{O}(m^3)$.

- full-rank Gaussian family: $m = \mathcal{O}(d^2)$, so $\mathcal{O}(d^6)$ complexity.

Complexity

Since LSVI relies on a linear regression with $m = \dim(s)$ regressors, it has complexity $\mathcal{O}(m^3)$.

- full-rank Gaussian family: $m = \mathcal{O}(d^2)$, so $\mathcal{O}(d^6)$ complexity.
- mean field Gaussian family: $m = \mathcal{O}(d)$, so $\mathcal{O}(d^3)$ complexity.

Complexity

Since LSVI relies on a linear regression with $m = \dim(s)$ regressors, it has complexity $\mathcal{O}(m^3)$.

- full-rank Gaussian family: $m = \mathcal{O}(d^2)$, so $\mathcal{O}(d^6)$ complexity.
- mean field Gaussian family: $m = \mathcal{O}(d)$, so $\mathcal{O}(d^3)$ complexity.

However, we can use reparametrisation tricks to get lower complexities.

Reparametrisation trick for Gaussians

If $X \sim \mathcal{N}(\mu, \Sigma)$, with $\Sigma = CC^\top$, then $X = \mu + CZ$, with $Z \sim \mathcal{N}(0, I)$: use as covariates

$$\tilde{s}(z) := \left(1, z^\top, \frac{z_1^2 - 1}{\sqrt{2}}, z_1 z_2, \ldots, z_1 z_d, \frac{z_2^2 - 1}{\sqrt{2}}, z_2 z_3, \ldots, \frac{z_d^2 - 1}{\sqrt{2}}\right)^\top, \tag{4}$$

and

$$\gamma := \mathrm{argmin}_{\gamma \in \mathbb{R}^m} \mathbb{E}_Z \left[\{\gamma^\top \tilde{s}(Z) - f(\mu + CZ)\}^2\right]. \tag{5}$$

Reparametrisation trick for Gaussians

If $X \sim \mathcal{N}(\mu, \Sigma)$, with $\Sigma = CC^\top$, then $X = \mu + CZ$, with $Z \sim \mathcal{N}(0, I)$: use as covariates

$$\tilde{s}(z) := \left(1, z^\top, \frac{z_1^2 - 1}{\sqrt{2}}, z_1 z_2, \ldots, z_1 z_d, \frac{z_2^2 - 1}{\sqrt{2}}, z_2 z_3, \ldots, \frac{z_d^2 - 1}{\sqrt{2}}\right)^\top, \tag{4}$$

and

$$\gamma := \mathrm{argmin}_{\gamma \in \mathbb{R}^m} \mathbb{E}_Z \left[\{\gamma^\top \tilde{s}(Z) - f(\mu + CZ)\}^2\right]. \tag{5}$$

Then $\mathbb{E}[\tilde{s}(Z)\tilde{s}(Z)^\top] = I_m$.
No inversion of the FIM
Going from γ to $\phi(\eta)$ is doable in $\mathcal{O}(d^3)$.

Sketch of the proof

For any $z \in \mathbb{R}^d$, let $x(z) = \mu + Cz$.

Sketch of the proof

For any $z \in \mathbb{R}^d$, let $x(z) = \mu + Cz$. Solve for γ:

$$\gamma^\top \tilde{s}(z) = \beta^\top s(x(z)). \tag{6}$$

Sketch of the proof

For any $z \in \mathbb{R}^d$, let $x(z) = \mu + Cz$. Solve for γ:

$$\gamma^\top \tilde{s}(z) = \beta^\top s(x(z)). \tag{6}$$

Then

$$\min_{\beta \in \mathbb{R}^m} \mathbb{E}_{X \sim \mathcal{N}(\mu, CC^\top)}[\{\beta^\top s(x) - f(x)\}^2] \tag{7}$$

is equivalent to

$$\min_{\gamma \in \mathbb{R}^m} \mathbb{E}_{Z \sim \mathcal{N}(0, I)}[\{\gamma^\top \tilde{s}(Z) - f(x(Z))\}^2]. \tag{8}$$

Cheaper algorithm for full-rank Gaussians

Require: $\mu_0, \Sigma_0 \succ 0$, $N \geq 1$
  $(\hat{\mu}_0, \hat{\Sigma}_0) \leftarrow (\mu_0, \Sigma_0)$
  $\hat{\eta}_0 \leftarrow (-\infty, -\Sigma^{-1}\mu, -\frac{1}{2}\operatorname{vec}\hat{\Sigma}^{-1})$
  while not converged **do**
    $\hat{C}_t \leftarrow \text{Cholesky}(\hat{\Sigma}_t)$
    $Z_1, \ldots, Z_N \sim \mathcal{N}(0, I)$

Cheaper algorithm for full-rank Gaussians

Require: $\mu_0, \Sigma_0 \succ 0$, $N \geq 1$

$(\hat{\mu}_0, \hat{\Sigma}_0) \leftarrow (\mu_0, \Sigma_0)$

$\hat{\eta}_0 \leftarrow (-\infty, -\Sigma^{-1}\mu, -\frac{1}{2}\operatorname{vec}\hat{\Sigma}^{-1})$

while not converged **do**

$\hat{C}_t \leftarrow \text{Cholesky}(\hat{\Sigma}_t)$

$Z_1, \ldots, Z_N \sim \mathcal{N}(0, I)$

$\hat{\gamma}_{t+1} \leftarrow \frac{1}{N}\sum_{i=1}^{N} t(Z_i) f(\hat{\mu}_t + \hat{C}_t Z_i)$

Invert $\hat{\gamma}_{t+1}$ to get $\hat{\eta}'_{t+1}$ {No matrix to invert}

$\varepsilon_t \leftarrow \texttt{stepsize}(\hat{\gamma}_{t+1}, \hat{\eta}'_{t+1}, \hat{\eta}_t, Z_{1:N})$

$\hat{\eta}_{t+1} \leftarrow \varepsilon_t \hat{\eta}'_{t+1} + (1 - \varepsilon_t)\hat{\eta}_t$

$\hat{\mu}_{t+1} \leftarrow -\frac{1}{2}\hat{\eta}_{2,t+1}^{-1}\hat{\eta}_{1,t+1}$

$\hat{\Sigma}_{t+1} \leftarrow -\frac{1}{2}\operatorname{unvec}(\hat{\eta}_{2,t+1})^{-1}$

end while

Mean-field case

Similar reparametrisation trick, gives $\mathcal{O}(d)$ complexity, see paper.

List of algorithms

1. Generic LSVI (any family $\mathcal{Q}$, $\mathcal{O}(m^3)$ complexity)
2. full-rank LSVI (full-rank Gaussian family, $\mathcal{O}(d^3)$ complexity)
3. mean-field LSVI (mean-field Gaussian family, $\mathcal{O}(d)$ complexity)

List of algorithms

1. Generic LSVI (any family $\mathcal{Q}$, $\mathcal{O}(m^3)$ complexity)
2. full-rank LSVI (full-rank Gaussian family, $\mathcal{O}(d^3)$ complexity)
3. mean-field LSVI (mean-field Gaussian family, $\mathcal{O}(d)$ complexity)

Note that these Algorithms tend to converge at a slower rate (but their iterations are cheaper than for Algorithm 1).

Logistic regression

- Typical benchmark in Bayesian computation.

Logistic regression

- Typical benchmark in Bayesian computation.
- Features $x_i \in \mathbb{R}^d$, labels $y_i \in \{-1, 1\}$, $i = 1, \ldots, n$, the posterior distribution of a logistic regression model is:

$$\pi(\beta) \propto p(\beta) \prod_{i=1}^{n} F(y_i x_i^\top \beta),$$

  p Gaussian, F logistic function.
- gradient of $f = \log \pi$ is easy to compute, so SGD may be implemented.

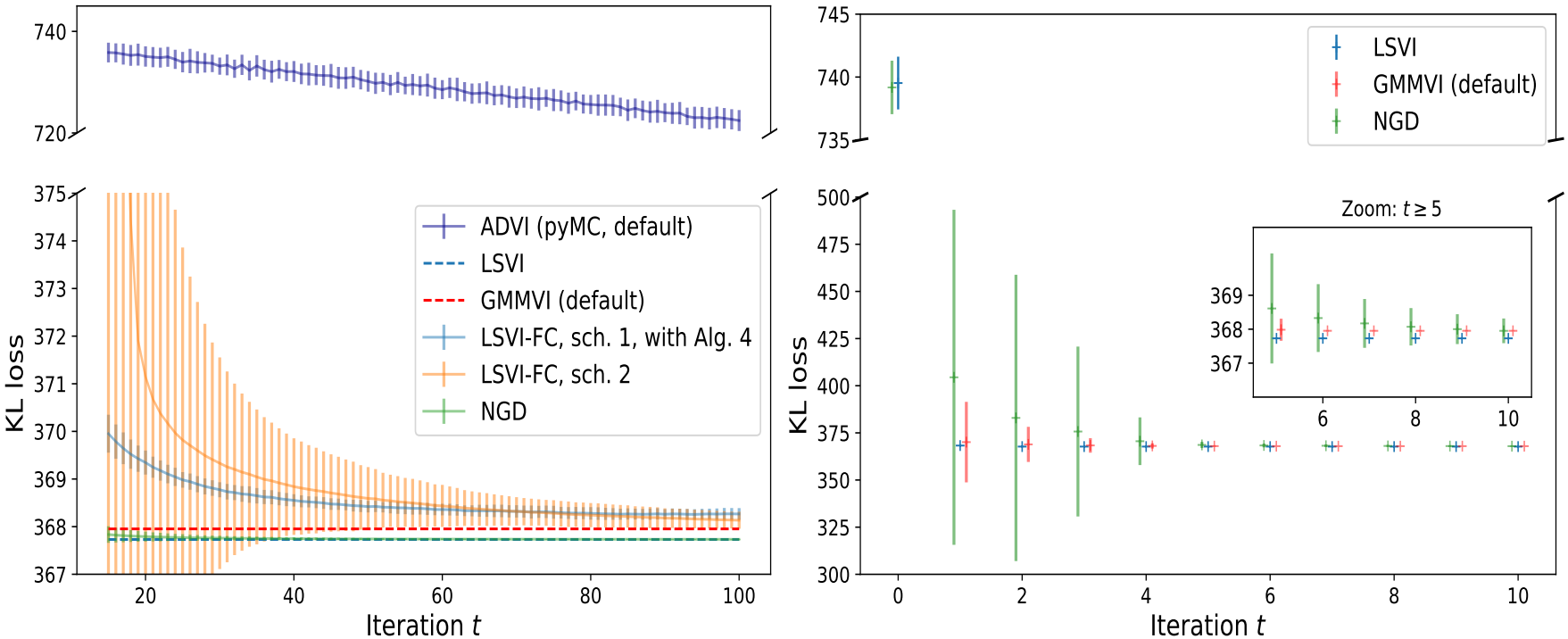
Logistic regression

- Typical benchmark in Bayesian computation.
- Features $x_i \in \mathbb{R}^d$, labels $y_i \in \{-1, 1\}$, $i = 1, \ldots, n$, the posterior distribution of a logistic regression model is:

$$\pi(\beta) \propto p(\beta) \prod_{i=1}^{n} F(y_i x_i^\top \beta),$$

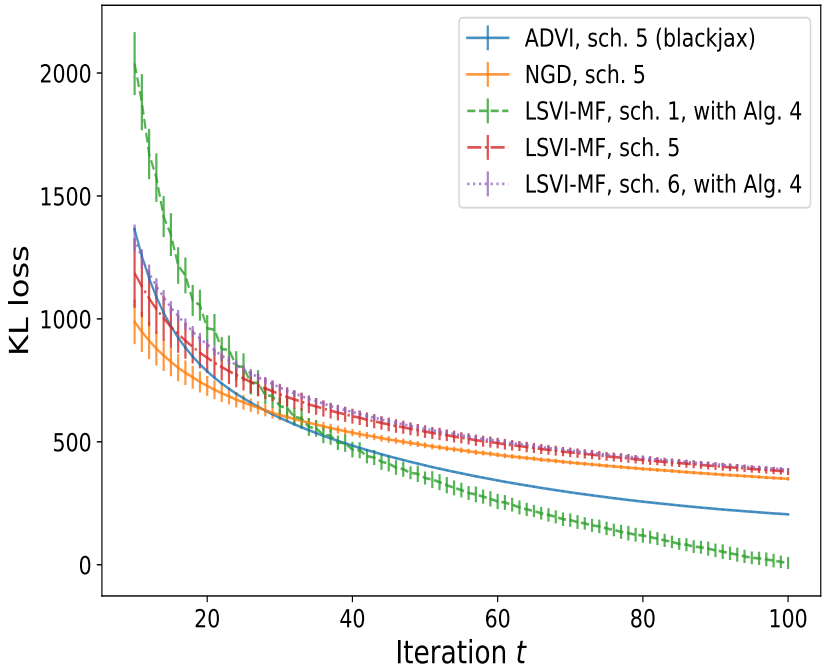
  p Gaussian, F logistic function.
- gradient of $f = \log \pi$ is easy to compute, so SGD may be implemented.
- Is LSVI still competitive in this case?
- Compare the three variants of LSVI with ADVI, NGD and other alternatives (GMMVI).

Convergence: Pima



Full-covariance approximation, KL divergence. Left: truncated from iteration $t \geq 20$ for better readability. Right: focus on GMMVI, LSVI and NGD. Mean over 100 repetitions and one standard deviation interval.

Convergence: MNIST



Diagonal covariance approximation, LSVI-MF, NGD and Blackjax (meanfield_vi) implementations. KL divergence.

Table of Contents

Convergence guarantees

Convergence analysis requires assumptions on $I : \eta \mapsto \mathrm{uKL}(q_\eta \mid \pi)$.

Convergence guarantees

Convergence analysis requires assumptions on $l : \eta \mapsto \mathrm{uKL}(q_\eta \mid \pi)$.

- l is smooth,

Convergence guarantees

Convergence analysis requires assumptions on $l : \eta \mapsto \mathrm{uKL}(q_\eta \mid \pi)$.

- l is smooth, and (strongly)-convex

Convergence guarantees

Convergence analysis requires assumptions on $l : \eta \mapsto \mathrm{uKL}(q_\eta \mid \pi)$.

- l is smooth, and (strongly)-convex

Theorem (Informal)

Let $k \geq 0$, $\delta \in (0, 1)$, Provided $N \geq N(\delta, k)$,

1. *there exists an event $\mathcal{A}_k$ that occurs with prob. at least $1 - \delta$,*
2. *conditioned on this event, the weighted average of the iterates $\bar{\eta}_{0:k}$ satisfies*

$$\mathbb{E}[l(\bar{\eta}_{0:k})] - l^\star \mid \mathcal{A}_k] = \underbrace{\mathcal{O}(N^{-1})}_{\text{MC error}} + \underbrace{\mathcal{O}(k^{-1})}_{\text{descent error}} + \underbrace{\mathcal{O}(N^{-1}k^{-1}\log(k))}_{\text{cross term}} \tag{9}$$

Convergence guarantees

Convergence analysis requires assumptions on $l : \eta \mapsto \mathrm{uKL}(q_\eta \mid \pi)$.

- l is smooth, and (strongly)-convex

Theorem (Informal)

Let $k \geq 0$, $\delta \in (0, 1)$, Provided $N \geq N(\delta, k)$,

1. *there exists an event $\mathcal{A}_k$ that occurs with prob. at least $1 - \delta$,*
2. *conditioned on this event, the weighted average of the iterates $\bar{\eta}_{0:k}$ satisfies*

$$\mathbb{E}[l(\bar{\eta}_{0:k})] - l^\star \mid \mathcal{A}_k] = \underbrace{\mathcal{O}(N^{-1})}_{\text{MC error}} + \underbrace{\mathcal{O}(k^{-1})}_{\text{descent error}} + \underbrace{\mathcal{O}(N^{-1}k^{-1}\log(k))}_{\text{cross term}} \tag{9}$$

Local convexity is sufficient in practice.

Table of Contents

Variable selection

Take π to be the *marginal* posterior of $\gamma \in \{0,1\}^d$, the vector of inclusion variables in a linear regression model:

$$y_i = x_i^\top \operatorname{diag}(\gamma)\beta + \sigma\varepsilon_i, \quad \varepsilon_i \sim N(0,1).$$

Variable selection

Take π to be the *marginal* posterior of $\gamma \in \{0,1\}^d$, the vector of inclusion variables in a linear regression model:

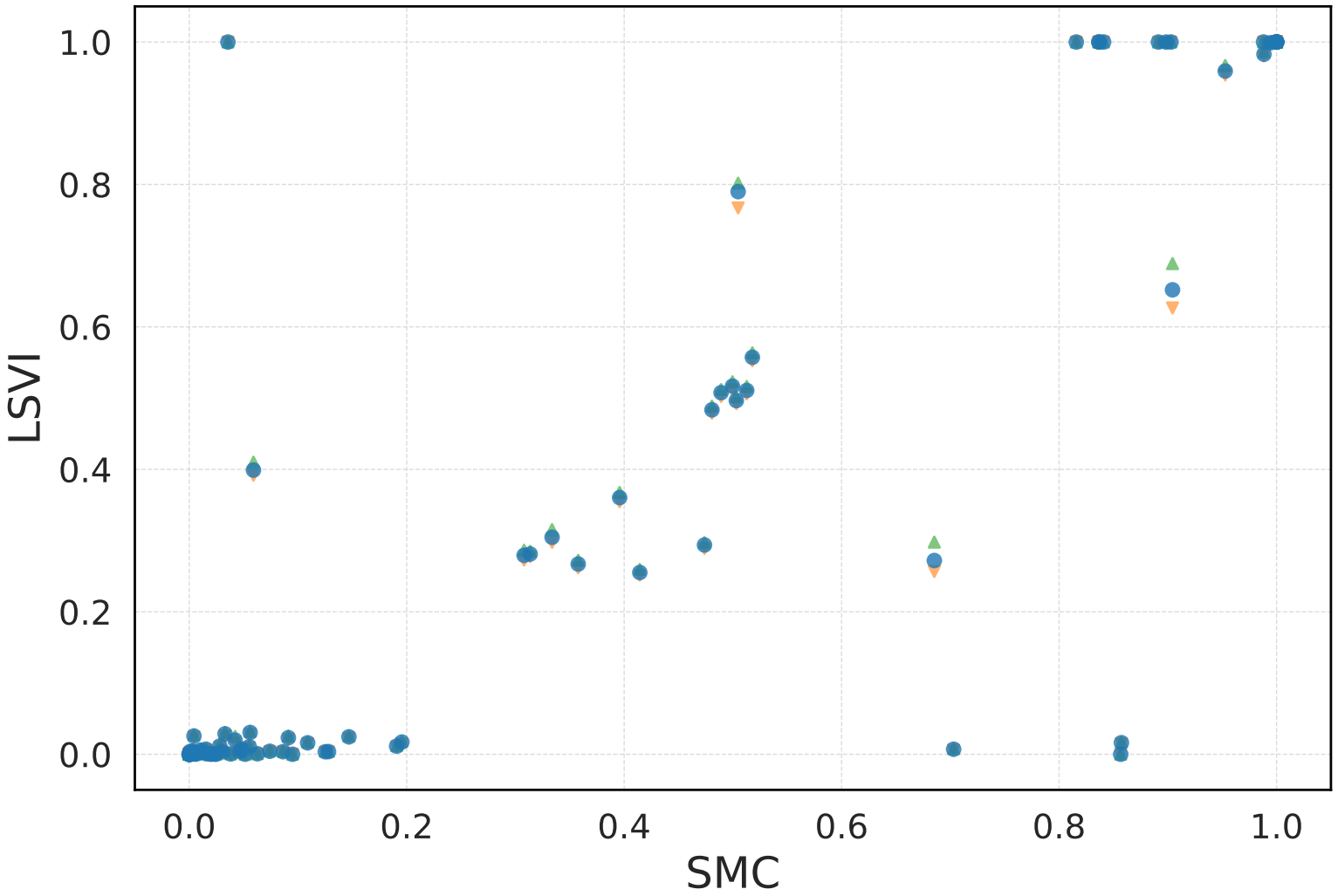
$$y_i = x_i^\top \operatorname{diag}(\gamma)\beta + \sigma\varepsilon_i, \quad \varepsilon_i \sim N(0,1).$$

Parametric family: Bernoulli product,

$$q(\gamma) = \prod_{i=1}^{d} q_i^{\gamma_i}(1-q_i)^{1-\gamma_i}.$$

No parametrisation trick.

Results



Variable selection, concrete dataset ($d = 92$), posterior marginal probabilities $\pi(\gamma_i = 1|\mathcal{D})$: LSVI approximation vs SMC.

BSL (Bayesian synthetic likelihood)

Likelihood-free inference: model is defined through a simulator, likelihood is intractable.

BSL (Bayesian synthetic likelihood)

Likelihood-free inference: model is defined through a simulator, likelihood is intractable.

BSL: assume $g(y) \sim N(\mu(\theta), \Sigma(\theta))$, where g is a chosen summary of the data. Replace likelihood by pseudo-likelihood:

$$\frac{(2\pi)^{-d_g/2}}{|\hat{\Sigma}(\theta)|^{1/2}} \exp\left\{-\frac{1}{2}(x-\hat{\mu}(\theta))^\top \hat{\Sigma}(\theta)^{-1}(x-\hat{\mu}(\theta))\right\}$$

where $\hat{\mu}(\theta)$, $\hat{\Sigma}(\theta)$ are computed from data simulated from the model (given θ).

BSL (Bayesian synthetic likelihood)

Likelihood-free inference: model is defined through a simulator, likelihood is intractable.

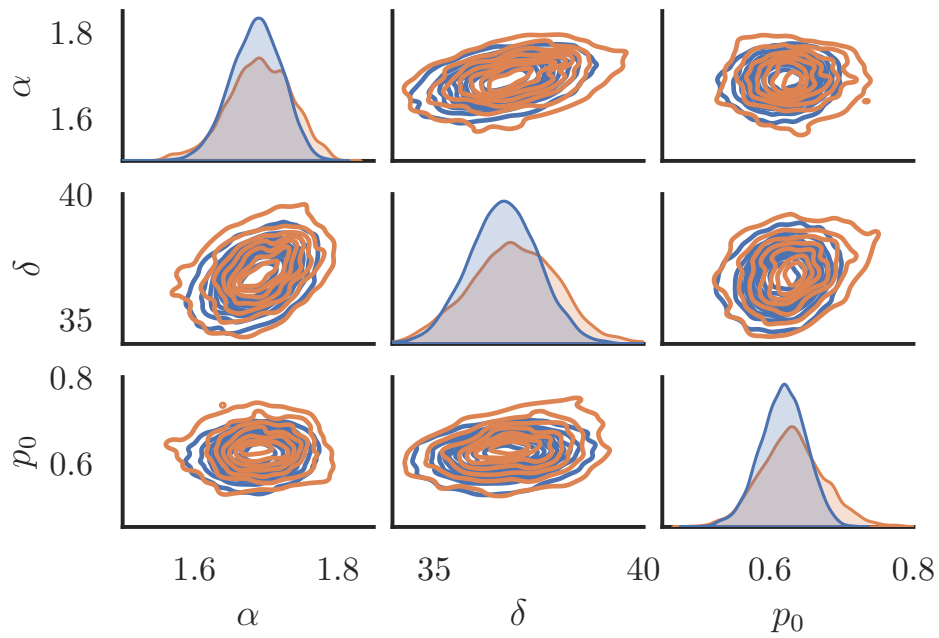
BSL: assume $g(y) \sim N(\mu(\theta), \Sigma(\theta))$, where g is a chosen summary of the data. Replace likelihood by pseudo-likelihood:

$$\frac{(2\pi)^{-d_g/2}}{|\hat{\Sigma}(\theta)|^{1/2}} \exp\left\{-\frac{1}{2}(x-\hat{\mu}(\theta))^\top \hat{\Sigma}(\theta)^{-1}(x-\hat{\mu}(\theta))\right\}$$

where $\hat{\mu}(\theta)$, $\hat{\Sigma}(\theta)$ are computed from data simulated from the model (given θ).

Pseudo-likelihood is noisy, non-differentiable.

Toad's stochastic displacement model (Marchand et al., 2017), $d = 3$, full-rank Gaussian approximations.



Variational approximation (LSVI, blue), MCMC approximation (orange).

Runtimes and max memory usage.

Experiment	Runtime (seconds)			max resident set size (memory usage)
	mean (std)	min	max	(gigabytes)
BSL Gaussian, LSVI, $(N, T) = (100, 50)$ (JAX)	72.9 ($\pm$2.8)	71.5	77.8	1.07
BSL Truncated MF Gaussian, LSVI, $(100, 50)$ (JAX)	137.5 ($\pm$0.6)	137.3	138.7	1.05
BSL MCMC, Blackjax (JAX)	268.1 ($\pm$3.4)	266.5	274.3	1.16

Table of Contents

Benefits of LSVI

- No gradient needed. Makes it possible to consider:
  - Discrete families
  - noisy and/or non-differentiable $\log \pi$.

Benefits of LSVI

- No gradient needed. Makes it possible to consider:
  - Discrete families
  - noisy and/or non-differentiable $\log \pi$.

Benefits of LSVI

- No gradient needed. Makes it possible to consider:
  - Discrete families
  - noisy and/or non-differentiable $\log \pi$.
- Converges quickly. Leverage highly optimised routines of linear algebra.

Benefits of LSVI

- No gradient needed. Makes it possible to consider:
  - Discrete families
  - noisy and/or non-differentiable $\log \pi$.
- Converges quickly. Leverage highly optimised routines of linear algebra.
- Supporting theory.

Benefits of LSVI

- No gradient needed. Makes it possible to consider:
  - Discrete families
  - noisy and/or non-differentiable $\log \pi$.
- Converges quickly. Leverage highly optimised routines of linear algebra.
- Supporting theory.
- Tailored schemes for Gaussians. Optimal one-iteration costs.

Benefits of LSVI

- No gradient needed. Makes it possible to consider:
  - Discrete families
  - noisy and/or non-differentiable $\log \pi$.
- Converges quickly. Leverage highly optimised routines of linear algebra.
- Supporting theory.
- Tailored schemes for Gaussians. Optimal one-iteration costs.
- Several interesting extensions to consider, including mixture of exponential families (Arenz et al., 2018)

Paper and package

Le Fay Y., Chopin N. and Barthelmé S. (2025). Least squares variational inference, arXiv:2502.18475 (NeurIPS 2025).
Python (JAX) package: `https://github.com/ylefay/LSVI`

References I

Arenz, O., Neumann, G., and Zhong, M. (2018). Efficient gradient-free variational inference using policy search. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 234–243. PMLR.

Giordano, R., Ingram, M., and Broderick, T. (2024). Black box variational inference with a deterministic objective: Faster, more accurate, and even more black box. *Journal of Machine Learning Research*, 25(18):1–39.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. (2017). Automatic differentiation variational inference. *J. Mach. Learn. Res.*, 18(1):430–474.

Marchand, P., Boenke, M., and Green, D. M. (2017). A stochastic movement model reproduces patterns of site fidelity and long-distance dispersal in a population of fowler's toads (anaxyrus fowleri). *Ecological Modelling*, 360:63–69.

References II

Minka, T. P. (2005). Divergence measures and message passing. In *Divergence measures and message passing*.

Salimans, T. and Knowles, D. A. (2013). Fixed-Form Variational Posterior Approximation through Stochastic Linear Regression. *Bayesian Analysis*, 8(4):837 – 882.

Welandawe, M., Andersen, M. R., Vehtari, A., and Huggins, J. H. (2024). A framework for improving the reliability of black-box variational inference. *Journal of Machine Learning Research*, 25(219):1–71.