The title

The subtitle

your name

17 de setembro de 2019

Outline

The title

The subtitle

your name

17 de setembro de 2019

Introduction

Some introduction.

Introduction

Bayesian theory

Building blocks

- Prior probability $p(\theta)$
- Likelihood $p(\mathcal{D}|\theta)$

Posterior probability

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta')p(\theta')d\theta'}$$

Introduction

Bayesian theory

Building blocks

- Prior probability $p(\theta)$
- Likelihood $p(\mathcal{D}|\theta)$

Posterior probability

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta')p(\theta')d\theta'}$$

Using posterior probability:

$$\int_{\Theta} f(\theta)p(\theta|\mathcal{D}, M)d\theta$$

Introduction

Bayesian theory

Building blocks

- Prior probability $p(\theta)$
- Likelihood $p(\mathcal{D}|\theta)$

Posterior probability

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta')p(\theta')d\theta'}$$

Using posterior probability:

$$\int_{\Theta} f(\theta)p(\theta|\mathcal{D}, M)d\theta$$

Bayesian theory requires integration!

Introduction

Approximate inference

Ways to integrate:

- Monte Carlo

$$\int_{\Theta} f(\theta)p(\theta|\mathcal{D})d\theta \approx \frac{1}{N}\sum_{i=1}^{N} f(\theta_i),\ \theta_i \sim p(\theta|\mathcal{D})$$

- Approximate distribution

$$p(\theta|\mathcal{D}) \approx q(\theta), \quad \int_{\Theta} f(\theta)q(\theta)d\theta$$

Introduction

Approximate inference

Ways to integrate:

- Monte Carlo

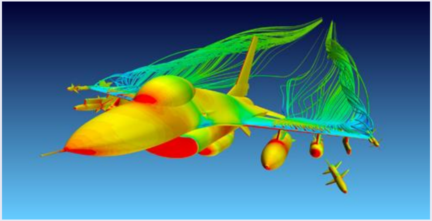
$$\int_{\Theta} f(\theta)p(\theta|\mathcal{D})d\theta \approx \frac{1}{N}\sum_{i=1}^{N} f(\theta_i),\ \theta_i \sim p(\theta|\mathcal{D})$$

- Approximate distribution

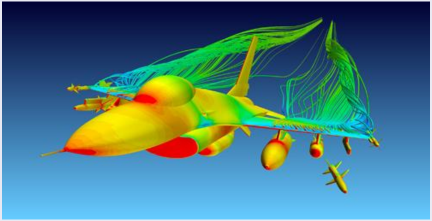
$$p(\theta|\mathcal{D}) \approx q(\theta), \quad \int_{\Theta} f(\theta)q(\theta)d\theta$$

Usual methods demands many evaluations of $p(\mathcal{D}|\theta)p(\theta)$. However, *this is not always feasible.*

In science, there are many cases that $p(\mathcal{D}|\theta)$ demands the computation of a forward model $g(\theta)$, which comes from an expensive simulation.

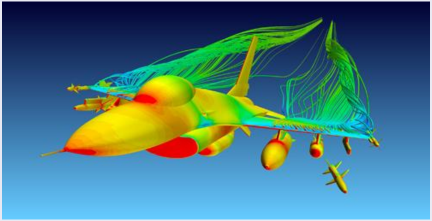


In science, there are many cases that $p(\mathcal{D}|\theta)$ demands the computation of a forward model $g(\theta)$, which comes from an expensive simulation.



This requires approximate inference methods "on a budget". In this work, one such method is developed, based on preexisting work. We name it

In science, there are many cases that $p(\mathcal{D}|\theta)$ demands the computation of a forward model $g(\theta)$, which comes from an expensive simulation.

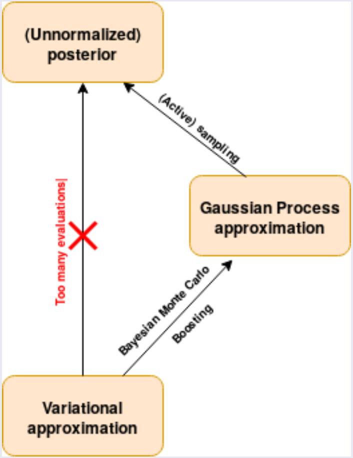


This requires approximate inference methods "on a budget". In this work, one such method is developed, based on preexisting work. We name it

Boosted Variational Bayesian Monte Carlo (BVBMC).

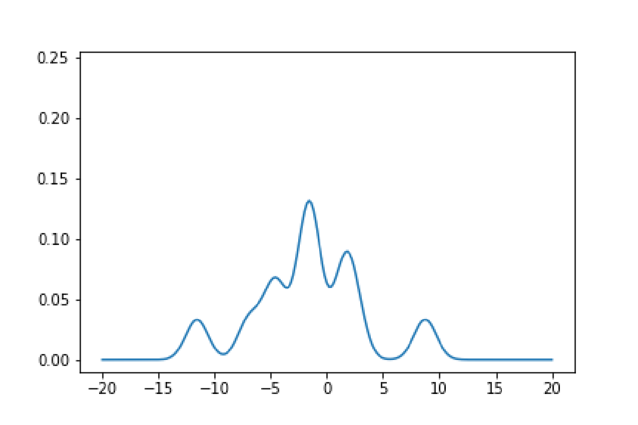
Introduction

BVBMC schema



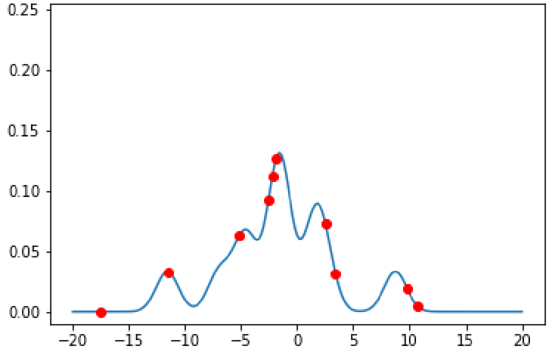
Introduction

An illustration of BVBMC



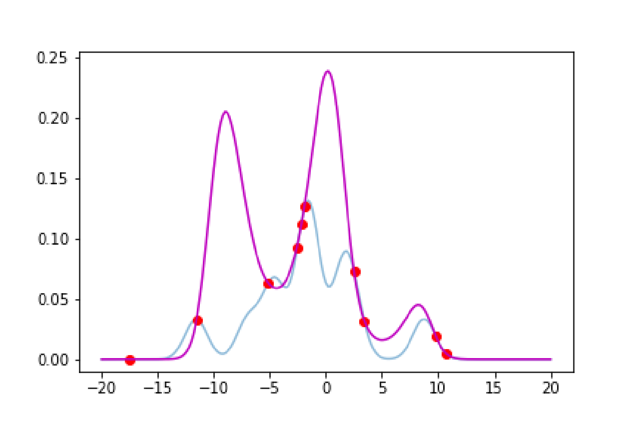
Introduction

An illustration of BVBMC



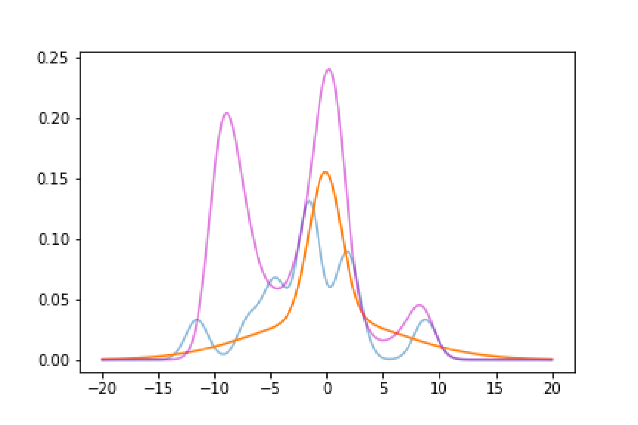
Introduction

An illustration of BVBMC



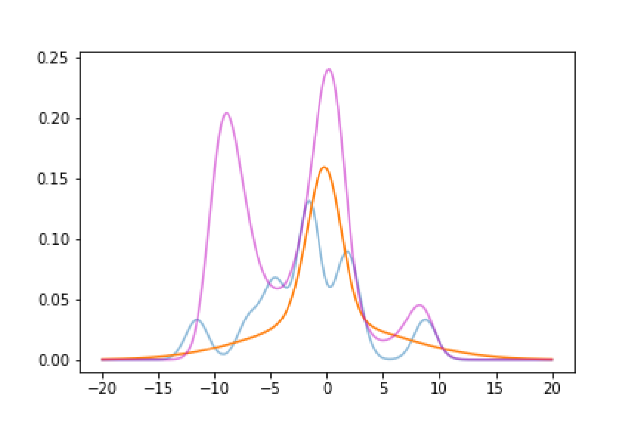
Introduction

An illustration of BVBMC



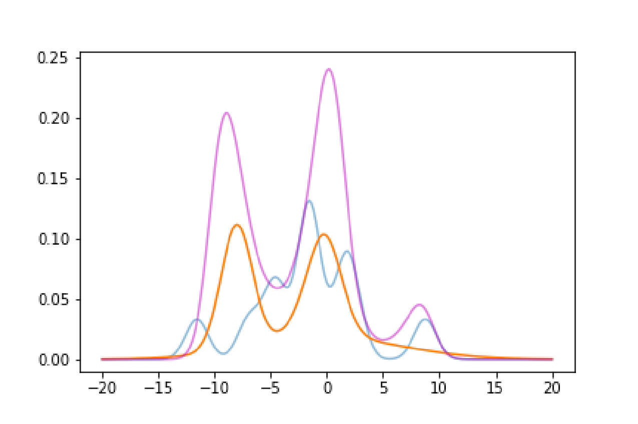
Introduction

An illustration of BVBMC



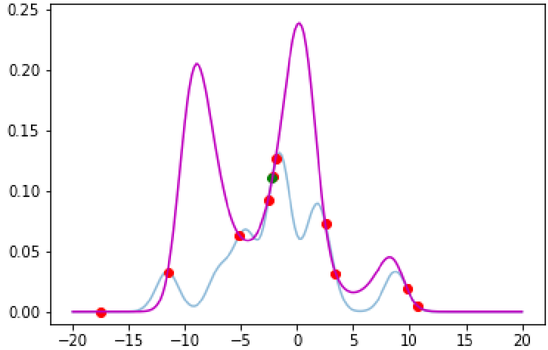
Introduction

An illustration of BVBMC



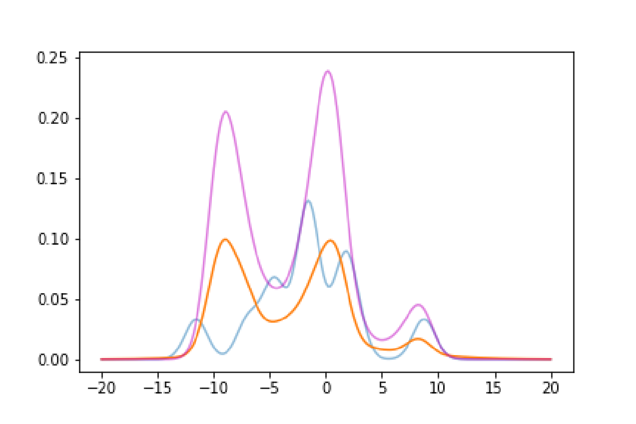
Introduction

An illustration of BVBMC



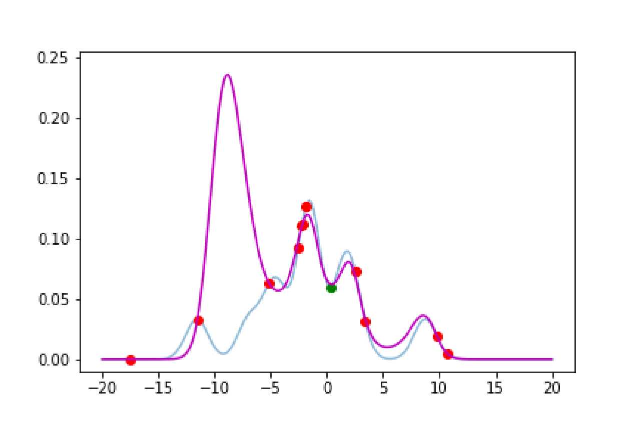
Introduction

An illustration of BVBMC



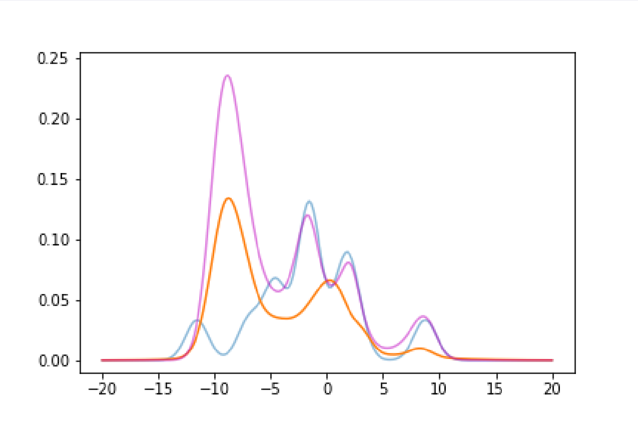
Introduction

An illustration of BVBMC



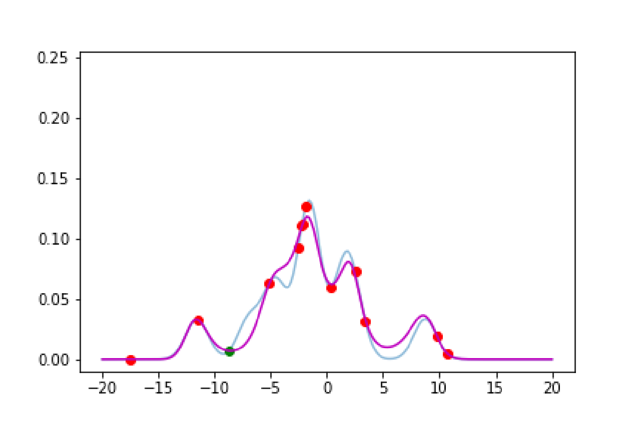
Introduction

An illustration of BVBMC



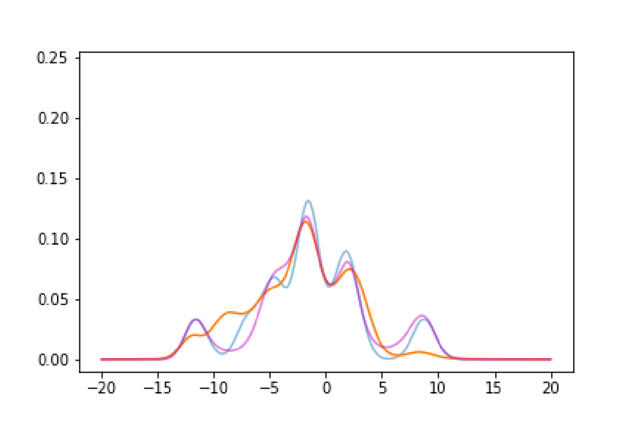
Introduction

An illustration of BVBMC



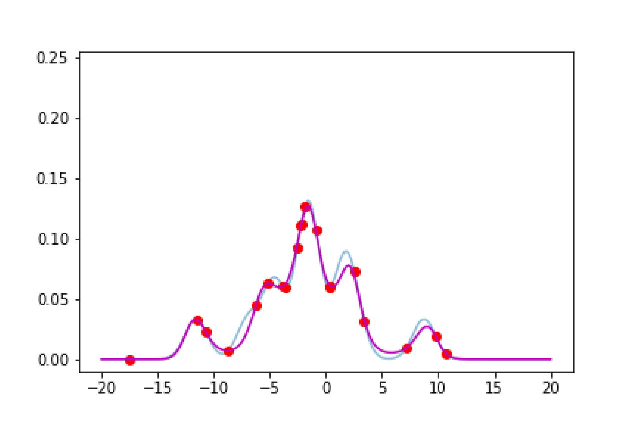
Introduction

An illustration of BVBMC



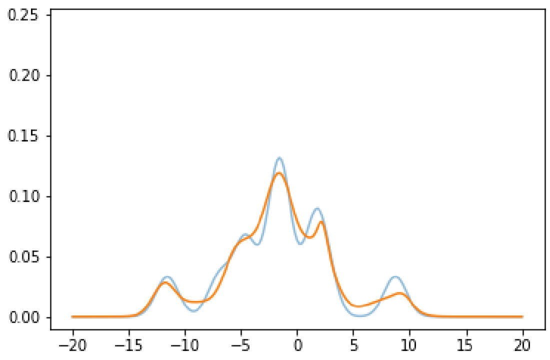
Introduction

An illustration of BVBMC



Introduction

An illustration of BVBMC



Gaussian Processes

Definition

Gaussian processes (GP): distribution over functions $f : \mathcal{X} \to \mathbb{R}$ such that $f(\mathbf{x}) = (f(x_1), \ldots, f(x_n))$ follows a multivariate normal distribution. A GP is completely defined by:

- $m(x) := \mathbb{E}[f(x)]$, mean function.
- $k(x, x') := \mathbb{E}[f(x), f(x')]$, covariance function or kernel.

such that $f(\mathbf{x}) \sim \mathcal{N}(m(\mathbf{x}), K(\mathbf{x}, \mathbf{x}))$.

Gaussian Processes

Definition

Gaussian processes (GP): distribution over functions $f : \mathcal{X} \to \mathbb{R}$ such that $f(\mathbf{x}) = (f(x_1), \ldots, f(x_n))$ follows a multivariate normal distribution. A GP is completely defined by:

- $m(x) := \mathbb{E}[f(x)]$, mean function.
- $k(x, x') := \mathbb{E}[f(x), f(x')]$, covariance function or kernel.

such that $f(\mathbf{x}) \sim \mathcal{N}(m(\mathbf{x}), K(\mathbf{x}, \mathbf{x}))$.

Gaussian process regression

Given $\mathcal{D} = (x, y)_{i=1}^{N}$, a Gaussian process regression is made by assuming $p(y|x) = p(y|f(x))$, with f following a prior $GP(m, k)$.

Gaussian Processes

Posterior GP

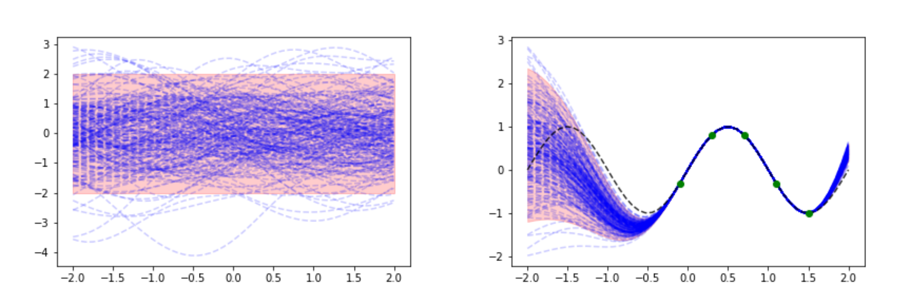
If $p(y|f(x)) = \mathcal{N}(f(x), \sigma_n^2)$, $f|\mathcal{D} \sim GP(m_{\mathcal{D}}, k_{\mathcal{D}})$, where

$$m_{\mathcal{D}}(x) := m(x) + K(x^\star, \mathbf{x})(K(\mathbf{x}, \mathbf{x}) + \sigma_n^2)^{-1}(\mathbf{y} - m(\mathbf{x}))$$
$$k_{\mathcal{D}}(x, x') := k(x, x') - K(x, \mathbf{x})(K(\mathbf{x}, \mathbf{x}) + \sigma_n^2)^{-1} K(\mathbf{x}, x)$$

Reduces to deterministic measurement when $\sigma_n^2 = 0$. More general $p(y|f(x))$ must resort to explicit marginalization.

Gaussian Processes

Example case



(a) GP prior

(b) GP posterior

Gaussian Processes

Kernels

The exigence that $K(\mathbf{x},\mathbf{x})$ limits which functions can be kernels. Some examples of kernels in $\mathbb{R}$ are:

- $k_{SQE}(x,x') = \theta_0 \exp\left(-\frac{1}{2}\frac{(x-x')^2}{l^2}\right)$
- $k_{\text{Matern},3/2}(x,x') = \theta_0 \left(\sqrt{3}\frac{(x-x')}{l}\right) \exp\left(-\sqrt{3}\frac{(x-x')}{l}\right)$

Kernels in $\mathbb{R}^D$ can be constructed by changing $\frac{(x-x')}{l}$ for $\sqrt{\sum_{i=1}^{D} \frac{(x_i-x_i')}{l_i}}$.
If k_1,k_2 are kernels, the following, among others are kernels:
$k_1(x,x') + k_2(x,x')$,$k_1(x,x')k_2(x,x')$,$k_1(x,x')k_2(y,y')$,$k_1(f(y),f(y'))$.

Gaussian Processes

Kernels

The exigence that $K(\mathbf{x},\mathbf{x})$ limits which functions can be kernels. Some examples of kernels in $\mathbb{R}$ are:

- $k_{SQE}(x,x') = \theta_0 \exp\left(-\frac{1}{2}\frac{(x-x')^2}{l^2}\right)$
- $k_{\text{Matern},3/2}(x,x') = \theta_0 \left(\sqrt{3}\frac{(x-x')}{l}\right) \exp\left(-\sqrt{3}\frac{(x-x')}{l}\right)$

Kernels in $\mathbb{R}^D$ can be constructed by changing $\frac{(x-x')}{l}$ for $\sqrt{\sum_{i=1}^{D} \frac{(x_i-x_i')}{l_i}}$.
If k_1,k_2 are kernels, the following, among others are kernels:
$k_1(x,x') + k_2(x,x')$,$k_1(x,x')k_2(x,x')$,$k_1(x,x')k_2(y,y')$,$k_1(f(y),f(y'))$.

Mean functions

In general,they are less important than kernels, since the latter determines the structure of the posterior GP. However, *outside the sampling area the GP prediction defaults to the mean*, which may be of importance.

Gaussian processes

Handling hyperparameters

$$\log p(\mathcal{D}|M,\sigma_n) = -\frac{1}{2}(\mathbf{y}-m(\mathbf{x}))^T(K(\mathbf{x},\mathbf{x})+\sigma_n\mathbf{I})^{-1}(\mathbf{y}-m(\mathbf{x}))+ \\ -\frac{1}{2}\log\det(K(\mathbf{x},\mathbf{x})+\sigma_n\mathbf{I})-\frac{1}{2}N\log(2\pi).$$

Inference can be done either by MLE, MAP, or integration techniques.

Scaling

The bottleneck of GP regression: $(K(\mathbf{x},\mathbf{x})+\sigma_n\mathbf{I})^{-1}$. Cost is $\mathcal{O}(N^3)$.
In online learning, each new sample is incorporated in $\mathcal{O}(N^2)$.

Bayesian Monte Carlo

Integrating a GP

$$Z = \int f(x)p(x)dx$$

If $f \sim GP(m, k)$, given $\mathcal{D} = \{(x_i, f(x_i))\}_{i=1}^N$, $Z_{\mathcal{D}} = \int f_{\mathcal{D}}(x)p(x)dx$ is Gaussian:

$$\mathbb{E}[Z_{\mathcal{D}}] = \int m(x)p(x)dx - \mathbf{z}^T K^{-1}(\mathbf{f} - m(\mathbf{x})), \quad \mathrm{Var}[Z_{\mathcal{D}}] = \Gamma - \mathbf{z}^T K^{-1}\mathbf{z},$$

$$z_i = \int k(x, x_i)p(x)dx, \quad \Gamma = \int \int k(x, x')p(x)p(x')dxdx'.$$

Bayesian Monte Carlo

Integrating a GP

$$Z = \int f(x)p(x)dx$$

If $f \sim GP(m, k)$, given $\mathcal{D} = \{(x_i, f(x_i))\}_{i=1}^N$, $Z_\mathcal{D} = \int f_\mathcal{D}(x)p(x)dx$ is Gaussian:

$$\mathbb{E}[Z_\mathcal{D}] = \int m(x)p(x)dx - \mathbf{z}^T K^{-1}(\mathbf{f} - m(\mathbf{x})), \quad \mathrm{Var}[Z_\mathcal{D}] = \Gamma - \mathbf{z}^T K^{-1}\mathbf{z},$$

$$z_i = \int k(x, x_i)p(x)dx, \quad \Gamma = \int\int k(x, x')p(x)p(x')dxdx'.$$

Name Bayesian *Monte Carlo* is misleading.

Bayesian Monte Carlo

Integrating a GP

$$Z = \int f(x)p(x)dx$$

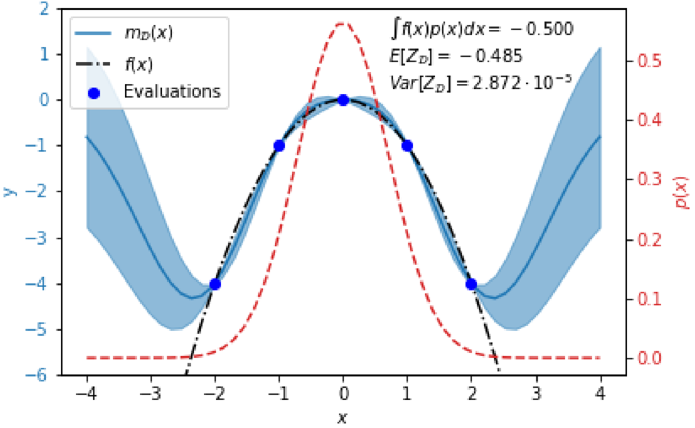
If $f \sim GP(m, k)$, given $\mathcal{D} = \{(x_i, f(x_i))\}_{i=1}^N$, $Z_\mathcal{D} = \int f_\mathcal{D}(x)p(x)dx$ is Gaussian:

$$\mathbb{E}[Z_\mathcal{D}] = \int m(x)p(x)dx - \mathbf{z}^T K^{-1}(\mathbf{f} - m(\mathbf{x})), \quad \mathrm{Var}[Z_\mathcal{D}] = \Gamma - \mathbf{z}^T K^{-1}\mathbf{z},$$

$$z_i = \int k(x, x_i)p(x)dx, \quad \Gamma = \int\int k(x, x')p(x)p(x')dxdx'.$$

Name Bayesian *Monte Carlo* is misleading.
Treating f as a random variable may be philosophically odd.

Bayesian Monte Carlo



Bayesian Monte Carlo

Kernel integral terms

In the general case, they can be estimated by Monte Carlo. When $p(x)$ is Gaussian or a mixture of Gaussians:

- Analytically tractable when $k(x, x')$ is the SQE kernel.
- Efficiently tractable when $k(x, x') = k(x_1, y_1) \ldots k(x_D, y_D)$.

Active sampling

Given $\{(x_1, f(x_1), \ldots, x_N, f(x_N)\}$, x_{N+1} may be chosen by optimizing acquisition functions.

$$\alpha^N_{\mathrm{MMLT}}(x) = e^{2m_{\mathcal{D}}(x)+k_{\mathcal{D}}(x,x)} \left(e^{k_{\mathcal{D}}(x,x)} - 1\right).$$

Variational Inference

Back to approximating posteriors $p(\theta|\mathcal{D}) \approx q(\theta;\lambda)$

Variational Inference: given $g(\theta)$, seeks minimization of $D_{KL}(q(\cdot;\lambda)||g)$. Given unnormalized $\bar{g}$, this is equivalent to maximizing the evidence lower bound (ELBO)

$$\mathcal{L}(\lambda) = \int \log \bar{g}(\theta) q(\theta) d\theta - \int \log q(\theta) q(\theta) d\theta$$

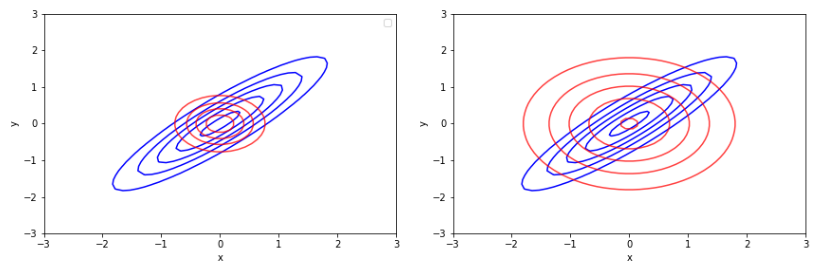
The family of variational posteriors $q(\theta;\lambda)$ must be easy to treat, in order for the approximation to be useful.

$D_{KL}(q(\cdot;\lambda)||g)$ vs $D_{KL}(g||q(\cdot;\lambda))$

$D_{KL}(q(\cdot;\lambda)||g) \neq D_{KL}(g||q(\cdot;\lambda))$: two minimization objectives. Gives two different algorithms (the second one, *expectation propagation*, is not treated here).

Variational Inference

Illustration



(c) $D_{KL}(q||g)$ (d) $D_{KL}(g||q)$

Variational Inference

Mean field variational inference

Consider factorized proposals $q(\theta) = q(\theta_1) \dots q(\theta_D)$.
Training by coordinate descent

$$q_j^*(\theta_j; q_{-j}) \propto \exp \mathbb{E}_{\theta_{-j} \sim q_{-j}}[\log \bar{g}(\theta)].$$

Generic variational inference

Uses stochastic gradient descent to find $q(\theta; \lambda)$.
REINFORCE: $\nabla \mathcal{L}(\lambda) = \mathbb{E}_{q(\theta;\lambda)} \left[\left(\log \left(\frac{\bar{g}(\theta)}{q(\theta;\lambda)} \right) + C \right) \nabla_\lambda \log q(\theta; \lambda) \right]$
Reparametrization:
$\nabla \mathcal{L}(\lambda) = \nabla \left(\mathbb{E}_{r(\epsilon)} \left[\log \frac{\bar{g}(s(\epsilon;\lambda))}{q(s(\epsilon;\lambda);\lambda)} \right] \right) \approx \frac{1}{K} \sum_{i \in [K], \epsilon_i \sim r(\epsilon)} \nabla \left(\log \frac{\bar{g}(s(\epsilon;\lambda))}{q(s(\epsilon;\lambda);\lambda)} \right).$

Variational Inference

Mixture of Gaussians

$q_k(\theta; \lambda) = \sum_{i=1}^k w_i f_i(\theta) = \sum_{i=1}^k w_i \mathcal{N}(\theta; \mu, \Sigma_i)$. Analytical mean and covariance. Samples can be easily generated.

Covariance parameterizations:

- $\Sigma_i = \text{diag}(\sigma_{i,1}^2, \ldots, \sigma_{i,D}^2)$
- $\Sigma_i = \mathbf{u}_i \mathbf{u}_i^T + \text{diag}(\sigma_{i,1}^2, \ldots, \sigma_{i,D}^2)$

Weights parameterizations $w_i(\nu_i) = \frac{\phi(\nu_i)}{\sum_{i=1}^k \phi(\nu_k)}$. ϕ can be:

- $\phi(\nu) = \exp(\nu)$
- $\phi(\nu) = \text{softplus}(\nu) = \log(1 + \exp(\nu))$

$$\mathcal{L}(\lambda) = \sum_{i=1}^k w_i(\nu_i) \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)} \left[\log \frac{\bar{g}(s(\epsilon; \mu_i, \sigma_i))}{q_k(s(\epsilon; \mu_i, \sigma_i); \lambda)} \right]$$

Variational Inference

Boosting mixtures

Problem: no way to know how many mixtures is needed. Adding mixtures sequentially can become costly. One solution: boosting.

$q_{i-1}(\theta) = \sum_{j=1}^{i-1} w_j f_j(\theta)$

$q_i(\theta) = \sum_{j=1}^{i-1} (1 - w_i) w_j f_j(\theta) + w_i f_i(\theta)$

How to find w_i and $f_i(\theta) = \mathcal{N}(\theta; \mu_i, \Sigma_i)$?

- Optimize jointly $\mathcal{L}_i(w_i, \mu_i, \Sigma_i)$
- Seek good proposal $f_i(\theta)$ and optimize $\mathcal{L}_i(w_i)$ via it's derivative

$$\mathcal{L}_i'(w_i) = \int \log(\bar{g}(\theta))(f_i(\theta) - q_{i-1}(\theta))d\theta - \int \log((1 - w_i)q_{i-1}(\theta) + w_i f_i(\theta))(f_i(\theta) - q_{i-1}(\theta))d\theta.$$

Variational Inference

Gradient boosting of mixtures

$$f_i = \arg\min_f \nabla D_{KL}(q_{i-1}||g) \cdot f = \arg\min_f \int \log \frac{q_{i-1}(\theta)}{g(\theta)} f(\theta) d\theta.$$

Problem: degenerate solution. Needs regularization.
Maximization objective for mixture of Gaussians:

$$\begin{aligned} \text{RELBO}(\mu_i, \Sigma_i) = & \int \log(\bar{g}(\theta))\mathcal{N}(\theta|\mu_i, \Sigma_i)d\theta - \\ & \int \log(q_{i-1}(\theta))\mathcal{N}(\theta|\mu_i, \Sigma_i)d\theta + \frac{\lambda}{4}\log|\Sigma|, \end{aligned}$$

Estimated by the reparameterization trick.

Variational Inference

```
procedure VariationalBoosting(log ḡ, μ_0, Σ_0)
    ▷ μ_0, Σ_0 the are initial boosting values
    w_0 := 1.0
    for t = 1, ..., T do
        μ_t, Σ_t := arg max RELBO(μ_t, Σ_t)      ▷ Using reparameterization
        w_t := arg max L_i(w_i)        ▷ Using L'_t(w_t) for gradient descent
        for j = 0, ..., t − 1 do
            w_j ← (1 − w_t) w_j
        end for
    end for
    return {(μ_t, Σ_t, w_t)}_{t=1}^T
end procedure
```

Variational Inference

Variational Bayesian Monte Carlo (VBMC)

$\mathcal{L}(\lambda) = \int \log \bar{g}(\theta) q(\theta;\lambda) d\theta - \int \log q(\theta;\lambda) q(\theta;\lambda) d\theta$

Use Bayesian Monte Carlo:

$\mathcal{L}_{\mathcal{D}}(\lambda) = \int \log \bar{g}_{\mathcal{D}}(\theta) q(\theta;\lambda) d\theta - \int \log q(\theta;\lambda) q(\theta;\lambda) d\theta$

$$\text{Maximize } \mathbb{E}[\mathcal{L}_{\mathcal{D}}(\lambda)] = M(\lambda) + \mathbf{z}^T \mathbf{w} - \int \log q(\theta;\lambda) q(\theta;\lambda) d\theta$$

$$\mathbf{w} = K^{-1} \mathbf{y}$$

$$M(\lambda) = \int m(\theta) q(\theta;\lambda) d\theta$$

$$\mathbf{z}_i = \int k(x, x_i) q(\theta;\lambda) dx.$$

Variational Inference

Mean function

$m(\theta) = 0$: $\log \bar{g}_{\mathcal{D}}(\theta)$ is not a log probability
Principled solution: $m(\theta) = -\frac{1}{2}\sum_{i=1}^{D} \frac{(\theta_i - c_i)^2}{l_i^2}$. Lends analytical $M(\lambda)$.
Ad-hoc solution: $m(\theta) = C$, with C being a large negative constant.

Active evaluation

Just as in BMC, it is possible to do active evaluation. Some options:

- $\alpha_{\text{US}}^{\mathcal{D}}(\theta_{N+1}) = k_{\mathcal{D}}(\theta_{N+1}, \theta_{N+1}) q_k(\theta_{N+1}; \lambda)^2$.
- $\alpha_{\text{PROP}}^{\mathcal{D}}(\theta_{N+1}) = k_{\mathcal{D}}(\theta_{N+1}, \theta_{N+1}) \exp(m_{\mathcal{D}}(\theta_{N+1})) q_k(\theta_{N+1}; \lambda)^2$

Boosted Variational Bayesian Monte Carlo

BVBMC

BVBMC = VBMC + boosting + small changes

BMC in boosted variational inference

$$\text{RELBO}_{\mathcal{D}}(\mu_i, \Sigma_i) = \int \mathbb{E}[\log \bar{g}_{\mathcal{D}}(\theta)] \mathcal{N}(\theta|\mu_i, \Sigma_i) d\theta - \int \log(q_{i-1}(\theta)) \mathcal{N}(\theta|\mu_i, \Sigma_i) d\theta + \frac{\lambda}{4} \log |\Sigma_i|$$

$$\mathcal{L}_{i,\mathcal{D}}(w) = \int \log \bar{g}_{\mathcal{D}}(\theta)((1 - w_i) q_{i-1}(\theta) + w_i f_i(\theta)) d\theta - \int \log((1 - w_i) q_{i-1}(\theta) + w_i f_i(\theta))((1 - w_i) q_{i-1}(\theta) + w_i f_i(\theta)) d\theta$$

Boosted Variational Bayesian Monte Carlo

Practical considerations

- RELBO stabilization

$$\mathrm{RELBO}_{\mathcal{D}}^{\delta_D}(\mu_i, \Sigma_i) = \int \log\left(\frac{r_{\mathcal{D}}(\theta)}{q_{i-1}(\theta) + \delta_D}\right) \mathcal{N}(\theta; \mu_i, \Sigma_i) d\theta + \log|\Sigma_i|.$$

- Output scaling

$$\tilde{y}_i = (y_i - m_y)/\sigma_y,\ \tilde{\mathcal{D}} = \{x_i, \tilde{y}_i\},\ \sigma_y \log g_{\tilde{\mathcal{D}}}(x) + \mu_y$$

- Component pruning: discard negligible components
- Initialization: either large covariance or maximize ELBO for first Gaussian component.
- Mean function: $m(\theta) = C$ found to be more stable.

Practical considerations

- Periodic joint parameter updating: sometimes maximize $\mathbb{E}[\mathcal{L}_{\mathcal{D}}(\lambda)]$ for all parameters in $\sum_{i=1}^{k} w_k \mathcal{N}(\theta; \mu_k, \Sigma_k)$.
- Product of Matern kernels:

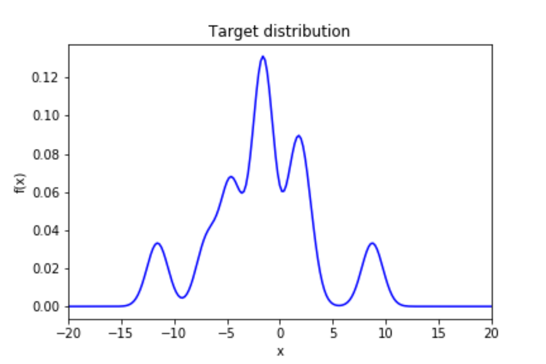
$$k_{\text{PMat},\nu}(x, x'; \theta, l) = \theta \prod_{d=1}^{D} k_{\text{Matern},\nu}(|x_i - x_i'|; l_d).$$

  Is integrated in BVBMC by Gauss-Hermite quadrature. Found to be more stable than the SQE kernel.
- More acquisition functions:

$$\alpha^{\mathcal{D}}_{MMLT}(x_{m+1}) = e^{2m_{\mathcal{D}}(x) + k_{\mathcal{D}}(x,x)} \left(e^{k_{\mathcal{D}}(x,x')} - 1 \right).$$

$$\alpha^{\mathcal{D}}_{MMLT_P}(x_{m+1}) = e^{2m_{\mathcal{D}}(x) + k_{\mathcal{D}}(x,x)} \left(e^{k_{\mathcal{D}}(x,x')} - 1 \right) q_k(\theta_{N+1}; \lambda)^2.$$

1-d mixture of Gaussians



$$f(x) = \sum_{i=1}^{12} w_i \mathcal{N}(x; \mu_i, \sigma_i^2),$$

$$w_i = \tfrac{1}{12},\ \mu_i \sim \mathcal{N}(0, \sqrt{5}),\ \sigma_i^2 = 1.$$