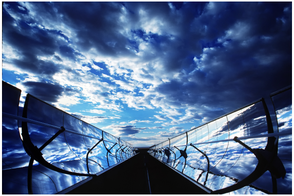
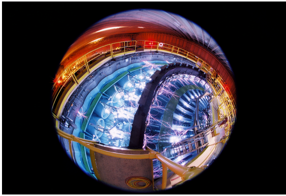
Exceptional service in the national interest

Sandia National Laboratories





Scalable^3-BayesOpt: Big Data meets HPC

A scalable parallel high-dimensional BO framework on supercomputers

ASME IDETC-CIE 2021. August 17–19, 2021. Virtual.

Anh Tran (SNL)

U.S. DEPARTMENT OF ENERGY — NNSA National Nuclear Security Administration

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.
SAND NO. SAND2021-???? C

boratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LL
Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE

Acknowledgment

Joint work with Bart G van Bloemen Waanders (SNL).
Funded by DOE/Office of Science/ASCR.

Advantages/Disadvantages

Bayesian optimization in a nutshell

Bayesian optimization = Gaussian process + sampling strategy

Advantages:

- optimize with uncertainty consideration
- active machine learning (balance exploration-exploitation)
- derivative free (avoid computing Jacobian)
- global optimization (convergence in probability to global optimum)
- good convergence rate (provably asymptotic regret, $\mathcal{O}\left(n^{-\frac{1}{d}}\right)$)

Disadvantages:

- high-dimensionality
- scalability: computational bottleneck $\mathcal{O}(n^3)$ when $n \geq \mathcal{O}(10^3)$

Bayesian optimization features

very **versatile** (open for methodological extensions)

- acquisition functions: PI, EI, UCB, Thompson sampling, entropy-based, KG, or combination among these
- constrained on objectives (known + unknown constraints) ✓
- multi-objective(Pareto frontier/optimal, domination) ✓
- multi-output ✗
- multi-fidelity (couple multiple low-, high-fidelity models) ✓
- batch parallelization ✓ → **asynchronous parallel** ✓
- stochastic ✗
- time-series (forecasting, e.g. causal kernel) ✗
- mixed-integer (discrete/categorical + continuous) ✓
- **scalable** ✓
- latent variable model ✗
- gradient-enhanced ✓
- **high-dimensional** (with low effective dimensionality) ✓
- physics-constrained: monotonic, discontinuous, symmetry, bound ✗
- outlier: student-t distribution ✗
- non-stationary ✗

Outline

An outline for this talk

- warning: will be dense in mathematics
- delibrate use of Sherman–Morrison–Woodbury formula for matrix inversion
- formulations
  1. (scalable + low-rank) sparse GP – variational inference for ELBO
  2. high-dimensional problem with low effective dimensionality: Johnson-Lindenstrauss lemma for Gaussian random projection[1]
  3. asynchronous (nonmyopic/lookahead) parallel on HPC
- our contribution: a unifying framework to tackle scalability with respect to different fronts: (1) data size, (2) dimensionality, (3) computational resource
- demonstration with distinct numerical examples
  - 1M data point
  - 10,000D with low-d
  - 20 concurrent workers for parallelization

[1] https://en.wikipedia.org/wiki/Random_projection

Notation

- $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^D$: inputs,
- $\mathbf{z} \in \mathcal{Z} \subset \mathbb{R}^d$: random embedded inputs,
- $\mathbf{X_u} \in \mathcal{X} \subset \mathbb{R}^D$: inducing inputs,
- $\mathbf{Z_u} \in \mathcal{Z} \subset \mathbb{R}^d$: random embedded inducing inputs,
- $\mathbf{u} \in \mathbb{R}$: inducing random embedded outputs,
- $\mathbf{y} \in \mathbb{R}$: outputs,
- D: dimensionality of x (before embedding),
- $d \ll D$: dimensionality of z (after embedding),
- $\mathbf{A} \in \mathbb{R}^{D \times d}$: normal random matrix, $\mathbf{a}_{ij} \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, 1)$.

Classical GP

Let $\mathcal{D}_n = \{\mathbf{x}_i, y_i\}_{i=1}^n$ denote the dataset. Assume observations jointly Gaussian

$$\mathbf{f}|\mathbf{X} \sim \mathcal{N}(\mathbf{m}, \mathbf{K}_{\mathbf{f},\mathbf{f}}), \tag{1}$$

and

$$\mathbf{y}|\mathbf{f}, \sigma^2 \sim \prod_{i=1}^{n} p(y_i|f_i) = \mathcal{N}(\mathbf{f}, \sigma^2\mathbf{I}), \tag{2}$$

then the Gaussian process posterior mean and variance is given by

$$\mu(\mathbf{x}^*) = \mathbf{m}(\mathbf{x}) + \mathbf{k}(\mathbf{x}^*)^\top(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I})^{-1}(\mathbf{y} - \mathbf{m}), \tag{3}$$

$$\sigma^2(\mathbf{x}^*) = \mathbf{K}(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*)^\top(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I})^{-1}\mathbf{k}(\mathbf{x}^*). \tag{4}$$

Classical GP

Formulation:

- assume stationary: only depends on distance $r = ||\mathbf{x} - \mathbf{x}^*||$
- then the covariance matrix is symmetric positive-semidefinite matrix made up of pairwise inner products

$$\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_j, \mathbf{x}_i) = \mathbf{K}_{ji} \tag{5}$$

- unknown function is *presumably* smooth, i.e. squared exponential is infinitely differentiable $\mathcal{C}^\infty$
- $\mathbf{x}$ are continuous, i.e. $f : \mathcal{X} \subset \mathbb{R}^D \to \mathbb{R}$.

Implementation:

- MLE to estimate the hyper-parameter $\theta \in \mathbb{R}^D$
- MLE is equivalent to MAP if prior is uniform
- cost complexity $\mathbf{K}^{-1}$ at the cost of $\mathcal{O}(n^3)$, $\mathbf{K} \in \mathbb{R}^{n \times n}$

Classical GP: A Bayesian perspective

Mostly follow Quiñonero-Candela and Hansen 2004; Quiñonero-Candela and Rasmussen 2005.
Denote training $\mathbf{f}$, testing $\mathbf{f}_*$, the joint GP prior is

$$p(\mathbf{f}, \mathbf{f}_*) = \mathcal{N}\left(\begin{bmatrix} \mathbf{m} \\ \mathbf{m} \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}} & \mathbf{K}_{*,\mathbf{f}} \\ \mathbf{K}_{\mathbf{f},*} & \mathbf{K}_{*,*} \end{bmatrix} \right). \tag{6}$$

By Bayes' rule

$$\begin{aligned} p(\mathbf{f}_*|\mathbf{y}) &= \int p(\mathbf{f}, \mathbf{f}_*|\mathbf{y}) d\mathbf{f} \\ &= \frac{1}{p(\mathbf{y})} \int p(\mathbf{y}|\mathbf{f})\ p(\mathbf{f}, \mathbf{f}_*) d\mathbf{f} \\ &= \mathcal{N}(\mathbf{m} + \mathbf{K}_{*,\mathbf{f}}[\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}]^{-1}(\mathbf{y} - \mathbf{m}), \mathbf{K}_{*,*} - \mathbf{K}_{*,\mathbf{f}}[\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}]^{-1}\mathbf{K}_{\mathbf{f},*}), \end{aligned} \tag{7}$$

Log of marginal likelihood function:

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{X}) &= \log \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\mathbf{X}) d\mathbf{f} \\ &= -\frac{n}{2} \log (2\pi) - \frac{1}{2} \log |\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}| - \frac{1}{2}(\mathbf{y} - \mathbf{m})^\top (\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I})^{-1}(\mathbf{y} - \mathbf{m}). \end{aligned} \tag{8}$$

Classical GP: A Bayesian perspective

A conditional of a Gaussian is also Gaussian.

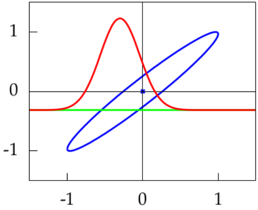


Figure: Photo courtesy of from Lawrence 2016.

If

$$P(\mathbf{x}, \mathbf{y}) = \mathcal{N}\left(\begin{bmatrix} \mu_{\mathbf{x}} \\ \mu_{\mathbf{y}} \end{bmatrix}, \begin{bmatrix} A & C \\ C^\top & B \end{bmatrix}\right) \tag{9}$$

then

$$P(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mu_{\mathbf{x}} + CB^{-1}(y - \mu_{\mathbf{y}}), A - CB^{-1}C^\top) \tag{10}$$

(cf. App. A, Quiñonero-Candela and Rasmussen 2005).

Sparse GP

Low-rank approximation[2] for $\mathbf{K}_{\mathbf{f},\mathbf{f}}$

Low-rank approximation $\mathbf{K} \approx \widetilde{\mathbf{K}} = \mathbf{K}_{n\times m}\mathbf{K}_{m\times m}^{-1}\mathbf{K}_{m\times n}$ (cf. Section 8.1 Rasmussen 2006) and scales as $\mathcal{O}(nm^2 + m^3)$ instead of $\mathcal{O}(n^3)$.
For $n \gg m$, this method scales as $\mathcal{O}(nm^2)$.

Following Quiñonero-Candela and Rasmussen 2005; Quiñonero-Candela, Rasmussen, and Williams 2007, Chalupka, Williams, and Murray 2013, Vanhatalo et al. 2012; Vanhatalo et al. 2013.
Cost complexity:

- local GP: $\mathcal{O}(m^3)$
- sparse GP: $\mathcal{O}(nm^2)$
- classical GP (Cholesky decomposition): $\mathcal{O}\left(\frac{1}{3}n^3\right)$
- classical GP (LU decomposition): $\mathcal{O}\left(\frac{2}{3}n^3\right)$
- classical GP (QR decomposition): $\mathcal{O}\left(\frac{4}{3}n^3\right)$

[2] https://en.wikipedia.org/wiki/Low-rank_matrix_approximations

Sparse GP

- $p(\cdot)$: true pdf
- $q(\cdot)$: approximate pdf

Assume the fully independent training conditional (FITC) Quiñonero-Candela and Rasmussen 2005; Quiñonero-Candela, Rasmussen, and Williams 2007, augment the joint model $p(\mathbf{f}_*, \mathbf{f})$ as

$$p(\mathbf{f}_*, \mathbf{f}) = \int p(\mathbf{f}_*, \mathbf{f}, \mathbf{u}) d\mathbf{u} = \int p(\mathbf{f}_*, \mathbf{f}|\mathbf{u}) p(\mathbf{u}) d\mathbf{u}, \tag{11}$$

$\mathbf{u}$: inducing variables at m locations $\mathbf{X}_\mathbf{u}$. The training and testing conditionals are

$$p(\mathbf{f}|\mathbf{u}) = \mathcal{N}(\mathbf{m} + \mathbf{K}_{\mathbf{f},\mathbf{u}} \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} (\mathbf{u} - \mathbf{m}),\ \mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{Q}_{\mathbf{f},\mathbf{f}}), \tag{12}$$

and

$$p(\mathbf{f}_*|\mathbf{u}) = \mathcal{N}(\mathbf{m} + \mathbf{K}_{*,\mathbf{u}} \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} (\mathbf{u} - \mathbf{m}),\ \mathbf{K}_{*,*} - \mathbf{Q}_{*,*}), \tag{13}$$

where

$$\mathbf{Q}_{\mathbf{a},\mathbf{b}} := \mathbf{K}_{\mathbf{a},\mathbf{u}} \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{K}_{\mathbf{u},\mathbf{b}}. \tag{14}$$

The likelihood and inducing priors remain the same, i.e. $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{f}, \sigma^2 \mathbf{I})$, and $p(\mathbf{u}) = \mathcal{N}(\mathbf{m}, \mathbf{K}_{\mathbf{u},\mathbf{u}})$.

Sparse GP

Sandia National Laboratories

FITC training prior based on the inducing priors is modified as

$$q(\mathbf{f}|\mathbf{u}) = \prod_{i=1}^{n} p(\mathbf{f}_i|\mathbf{u}) = \mathcal{N}(\mathbf{m} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}(\mathbf{u} - \mathbf{m}), \text{Diag}[\mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{Q}_{\mathbf{f},\mathbf{f}}]) \tag{15}$$

and keeping the testing prior the same

$$q(\mathbf{f}_*|\mathbf{u}) = p(\mathbf{f}_*|\mathbf{u}) = \mathcal{N}(\mathbf{m} + \mathbf{K}_{*,\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}(\mathbf{u} - \mathbf{m}),\ \mathbf{K}_{*,*} - \mathbf{Q}_{*,*}), \tag{16}$$

the effective prior under the FITC assumption is

$$q(\mathbf{f}, \mathbf{f}_*) = \mathcal{N}\left(\begin{bmatrix}\mathbf{m}\\ \mathbf{m}\end{bmatrix}, \begin{bmatrix}\mathbf{Q}_{\mathbf{f},\mathbf{f}} - \text{Diag}[\mathbf{Q}_{\mathbf{f},\mathbf{f}} - \mathbf{K}_{\mathbf{f},\mathbf{f}}] & \mathbf{Q}_{\mathbf{f},*}\\ \mathbf{Q}_{*,\mathbf{f}} & \mathbf{K}_{*,*}\end{bmatrix}\right), \tag{17}$$

which implies the testing distribution as

$$\begin{aligned} q(\mathbf{f}_*|\mathbf{y}) &= \mathcal{N}(\mathbf{m} + \mathbf{Q}_{*,\mathbf{f}}(\mathbf{Q}_{\mathbf{f},\mathbf{f}} + \Lambda)^{-1}(\mathbf{y} - \mathbf{m}), \mathbf{K}_{*,*} - \mathbf{Q}_{*,\mathbf{f}}(\mathbf{Q}_{\mathbf{f},\mathbf{f}} + \Lambda)^{-1}\mathbf{Q}_{\mathbf{f},*}) \\ &= \mathcal{N}(\mathbf{m} + \mathbf{K}_{*,\mathbf{u}}\Sigma\mathbf{K}_{\mathbf{u},\mathbf{f}}\Lambda^{-1}(\mathbf{y} - \mathbf{m}), \mathbf{K}_{*,*} - \mathbf{Q}_{*,*} + \mathbf{K}_{*,\mathbf{u}}\Sigma\mathbf{K}_{\mathbf{u},*}) \end{aligned}, \tag{18}$$

where $\Sigma = [\mathbf{K}_{\mathbf{u},\mathbf{u}} + \mathbf{K}_{\mathbf{u},\mathbf{f}}\Lambda^{-1}\mathbf{K}_{\mathbf{f},\mathbf{u}}]^{-1}$ and $\Lambda = \text{Diag}[\mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{Q}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}]$.

Sparse GP

The marginal likelihood conditioned on the inducing inputs is therefore

$$q(\mathbf{y}|\mathbf{X_u}) = \int\int p(\mathbf{y}|\mathbf{f})q(\mathbf{f}|\mathbf{u})p(\mathbf{u}|\mathbf{X_u})d\mathbf{u}d\mathbf{f} = \int p(\mathbf{y}|\mathbf{f})q(\mathbf{f}|\mathbf{X_u})d\mathbf{f}, \tag{19}$$

which implies the log marginal likelihood as

$$\log q(\mathbf{y}|\mathbf{X_u}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{Q_{f,f}} + \Lambda| - \frac{1}{2}(\mathbf{y}-\mathbf{m})^\top[\mathbf{Q_{f,f}} + \Lambda]^{-1}(\mathbf{y}-\mathbf{m}), \tag{20}$$

where $\Lambda = \text{Diag}[\mathbf{K_{f,f}} - \mathbf{Q_{f,f}}] + \sigma^2\mathbf{I}$.
Note that by Sherman–Morrison–Woodbury formula,

$$\begin{aligned}[\mathbf{Q_{f,f}} + \Lambda]^{-1} &= [\Lambda + \mathbf{K_{f,u}}\mathbf{K_{u,u}^{-1}}\mathbf{K_{u,f}}]^{-1} \\ &= \Lambda^{-1} - \Lambda^{-1}\mathbf{K_{f,u}}[\mathbf{K_{u,u}} + \mathbf{K_{u,f}}\Lambda^{-1}\mathbf{K_{f,u}}]^{-1}\mathbf{K_{u,f}}\Lambda^{-1}.\end{aligned} \tag{21}$$

Cost complexity: $\mathcal{O}(nm^2)$ Williams and Seeger 2001; Li, Kwok, and Lü 2010, if $\mathbf{X_u}$ is chosen in advanced.

Variational inference: a hand-waving argument

Follows Frigola, Chen, and Rasmussen 2014 and Rasmussen's corresponding slides. By Bayes' rule,

$$p(\mathbf{f}|\mathbf{y},\theta) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\theta)}{p(\mathbf{y}|\theta)} \Leftrightarrow p(\mathbf{y}|\theta) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\theta)}{p(\mathbf{f}|\mathbf{y},\theta)}. \tag{22}$$

The idea: approximate the (computationally intractable) $p(\mathbf{f}|\mathbf{y},\theta)$ by a (computationally tractable) parameterized variational $q(\mathbf{f})$. For any $q(\mathbf{f})$,

$$p(\mathbf{y}|\theta) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\theta)}{p(\mathbf{f}|\mathbf{y},\theta)}\frac{q(\mathbf{f})}{q(\mathbf{f})} \Leftrightarrow \log p(\mathbf{y}|\theta) = \log\frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\theta)}{q(\mathbf{f})} + \log\frac{q(\mathbf{f})}{p(\mathbf{f}|\mathbf{y},\theta)}. \tag{23}$$

Apply $\int q(\mathbf{f})d\mathbf{f}$ to both sides

$$\underbrace{\log p(\mathbf{y}|\theta)}_{\text{marginal likelihood}} = \underbrace{\int q(\mathbf{f})\log\frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\theta)}{q(\mathbf{f})}d\mathbf{f}}_{\text{Evidence Lower BOund}} + \underbrace{\int q(\mathbf{f})\log\frac{q(\mathbf{f})}{p(\mathbf{f}|\mathbf{y},\theta)}d\mathbf{f}}_{KL(q(\mathbf{f})||p(\mathbf{f}|\mathbf{y},\theta))} \tag{24}$$

Somewhat related to the reparameterization trick in VAE Diederik and Welling 2014.

Turn our attention to maximizing the variational ELBO (or equivalently minimizing the KL divergence) instead of maximizing the log marginal likelihood.

Variational inference: a rigorous approach

Mostly follow Titsias 2009a; Titsias 2009b and Bonilla, Krauth, and Dezfouli 2019.
Definition of conditionally independent condition:

$$p(\mathbf{f}|\mathbf{u},\mathbf{y}) = p(\mathbf{f}|\mathbf{u}), \tag{25}$$

which implies $p(\mathbf{f},\mathbf{u}|\mathbf{y}) = p(\mathbf{f}|\mathbf{u},\mathbf{y})p(\mathbf{u}|\mathbf{y}) \approx q(\mathbf{f},\mathbf{u}) = p(\mathbf{f}|\mathbf{u})q(\mathbf{u})$, where $q(\mathbf{u})$ is the approximate variational posterior. Main tool: Jensen's inequality.

$$\begin{aligned}
\log q(\mathbf{y}|\mathbf{X_u}) &= \log \int\int p(\mathbf{y}|\mathbf{f})q(\mathbf{f}|\mathbf{u})p(\mathbf{u}|\mathbf{X_u}) \times \frac{q(\mathbf{u},\mathbf{f})}{q(\mathbf{u},\mathbf{f})} d\mathbf{u}d\mathbf{f} \\
&\geq \int\int q(\mathbf{u},\mathbf{f}) \log \frac{p(\mathbf{y}|\mathbf{f})q(\mathbf{f}|\mathbf{u})p(\mathbf{u}|\mathbf{X_u})}{q(\mathbf{u},\mathbf{f})} d\mathbf{u}d\mathbf{f} \\
&= \int\int p(\mathbf{f}|\mathbf{u})q(\mathbf{u}) \log \frac{p(\mathbf{y}|\mathbf{f})\cancel{q(\mathbf{f}|\mathbf{u})}p(\mathbf{u}|\mathbf{X_u})}{\cancel{p(\mathbf{f}|\mathbf{u})}q(\mathbf{u})} d\mathbf{u}d\mathbf{f} \\
&= \int q(\mathbf{u}) \left\{ \int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} + \log \frac{p(\mathbf{u}|\mathbf{X_u})}{q(\mathbf{u})} \right\} d\mathbf{u} \\
&= \int q(\mathbf{u}) \left\{ \log G(\mathbf{u},\mathbf{y}) + \log \frac{p(\mathbf{u}|\mathbf{X_u})}{q(\mathbf{u})} \right\} d\mathbf{u} \\
&= \int q(\mathbf{u}) \left\{ \log \frac{G(\mathbf{u},\mathbf{y})p(\mathbf{u}|\mathbf{X_u})}{q(\mathbf{u})} \right\} d\mathbf{u} := \mathcal{F}_V(\mathbf{X_u},\mathbf{u}),
\end{aligned} \tag{26}$$

$$\begin{aligned}
\log G(\mathbf{u},\mathbf{y}) &= \int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} \\
&= \int p(\mathbf{f}|\mathbf{u}) \left\{ -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \mathrm{Tr}\left[\mathbf{y}\mathbf{y}^\top - 2\mathbf{y}\mathbf{f}^\top + \mathbf{f}\mathbf{f}^\top\right] \right\} d\mathbf{f} \\
&= -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \mathrm{Tr}\left[\mathbf{y}\mathbf{y}^\top - 2\mathbf{y}\alpha^\top + \alpha\alpha^\top + \mathbf{Q_{f,f}} - \mathbf{K_{f,f}}\right] \\
&= \mathcal{N}(\mathbf{y}|\alpha,\sigma^2\mathbf{I}) - \frac{1}{2\sigma^2}\mathrm{Tr}[\mathrm{Cov}(\alpha)],
\end{aligned} \tag{27}$$

Variational inference: a rigorous approach

where $\alpha = \mathbf{f}|\mathbf{u}$, with

$$\mathbb{E}[\alpha] = \mathbb{E}[\mathbf{f}|\mathbf{u}] = \mathbf{m} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}(\mathbf{u} - \mathbf{m}) \tag{28}$$

and

$$\mathsf{Cov}[\alpha] = \mathsf{Cov}[\mathbf{f}|\mathbf{u}] = \mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{Q}_{\mathbf{f},\mathbf{f}} = \mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}. \tag{29}$$

Reverse Jensen's inequality to maximize the variational evidence lower bound $\mathcal{F}_V(\mathbf{X}_\mathbf{u}, \mathbf{u})$ w.r.t. $q(\mathbf{u})$

$$\begin{aligned}
\mathcal{F}_V(\mathbf{X}_\mathbf{u}, \mathbf{u}) &= \int q(\mathbf{u}) \left\{ \log \frac{G(\mathbf{u},\mathbf{y}) p(\mathbf{u}|\mathbf{X}_\mathbf{u})}{q(\mathbf{u})} \right\} d\mathbf{u} \\
&\leq \int \log G(\mathbf{u},\mathbf{y}) p(\mathbf{u}|\mathbf{X}_\mathbf{u}) d\mathbf{u} \\
&= \log[\mathcal{N}(\mathbf{y}|\mathbf{m}, \sigma^2\mathbf{I} + \mathbf{Q}_{\mathbf{f},\mathbf{f}})] - \frac{1}{2\sigma^2} \mathsf{Tr}\left[\mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}\right] =: \mathcal{F}_V(\mathbf{X}_\mathbf{u})
\end{aligned} \tag{30}$$

Train sparse GP by maximizing $\mathcal{F}_V(\mathbf{X}_\mathbf{u})$. See also Vanhatalo et al. 2012; Vanhatalo et al. 2013, Bauer, Wilk, and Rasmussen 2016; Burt, Rasmussen, and Wilk 2020, Matthews et al. 2016.

Numerical benchmark: Big Data

- Intel Xeon Platinum 8160 CPU @ 2.10GHz
- 24 cores, 48 threads
- RHEL 7.1 (Maipo)
- 180 GB of memory
- sphere function $y = \sum_{i=1}^{3} (x_i)^2$, $\mathcal{X} = [-1, 1]^3$
- training data points: $n \in \{10^1, 10^2, \dots, 10^6\}$
- number of inducing points: $m \in \{10, 50, 100, \dots, 300\}$
- GPstuff with SuitSparse toolbox on MATLAB
- $m = 300$, $n = 10^6$ takes $\sim$48 minutes

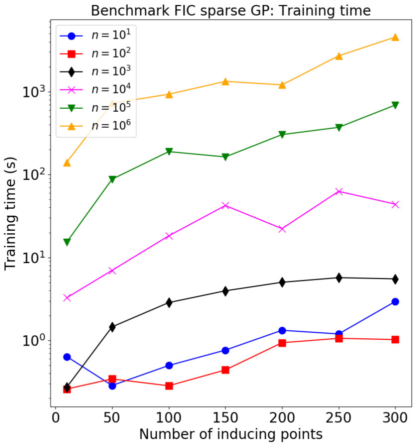


Figure: Benchmark of training time.

Numerical benchmark: Big Data

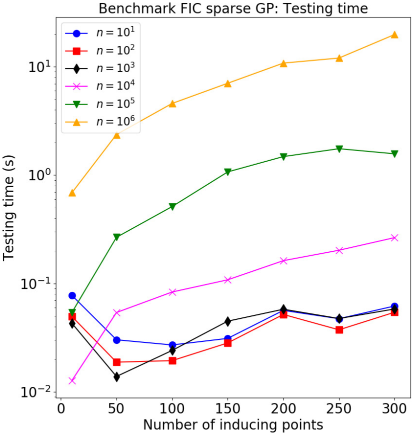


Figure: Benchmark of testing time.

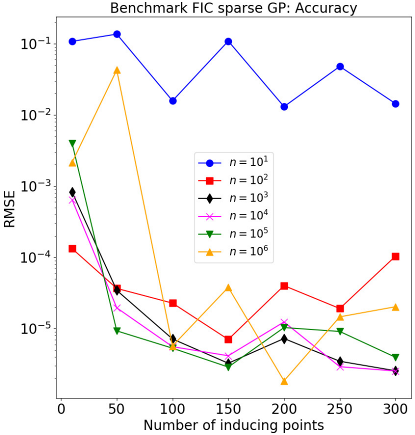


Figure: Benchmark of accuracy.

High-dimensional: Active subspace method

Formulations are derived by Constantine, Dow, and Wang 2014; Constantine 2015
Ideas:

- approximate high-dimensional function using gradients, $f : \mathcal{X} \subset \mathbb{R}^D \to \mathbb{R}$
- perform SVD on covariance of gradient vector with descending eigenvalues

$$\mathbb{E}[\nabla f(\mathbf{x}) \nabla f(\mathbf{x})^\top] = \mathbf{W} \text{Diag}[\lambda_1, \dots, \lambda_D] \mathbf{W}^\top \tag{31}$$

$$\text{Diag}[\lambda_1, \dots, \lambda_D] = \text{Diag}[\lambda_1, \dots, \lambda_d] \bigoplus \text{Diag}[\lambda_{d+1}, \dots, \lambda_D], \quad \mathbf{W} = [\mathbf{W}_1 \ \mathbf{W}_2] \tag{32}$$

- rotate the inputs $\mathbf{W}_1 \in \mathbb{R}^{D \times d}, \mathbf{W}_2 \in \mathbb{R}^{D \times (D-d)}$

$$f(\mathbf{x}) = f(\mathbf{W}\mathbf{W}^\top \mathbf{x}) = f(\mathbf{W}_1 \mathbf{W}_1^\top \mathbf{x} + \mathbf{W}_2 \mathbf{W}_2^\top \mathbf{x}) = f(\mathbf{W}_1 \mathbf{y} + \mathbf{W}_2 \mathbf{z}) \tag{33}$$

- if $\mathbf{z}$ invariant in an inactive subspace $\lambda_{d+1} = \cdots = \lambda_D = 0$, then $f(\mathbf{x}) = f(\mathbf{W}_1 \mathbf{y})$: reduce from D to d
- work great if gradients are readily available
- but what if gradients are not available? estimation by GP? constrained manifold optimization for $\mathbf{W}_1^\top$ besides the original optimization?

High-dimensional: Gaussian random projection

Mostly follow Wang et al. 2013; Wang et al. 2016. Main idea:

- choose (wisely) and optimize over $\mathcal{Z} \subset \mathbb{R}^d$
- embed and project onto high-dimensional space as $\mathbf{x} \leftarrow p_{\mathcal{X}}(\mathbf{A}z)$
- $\mathbf{A} \in \mathbb{R}^{D \times d}$: tall-and-skinny random matrix with standard normal component

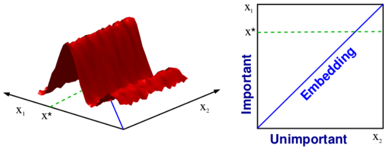


Figure: Photo courtesy of Wang et al Wang et al. 2016. Optimizing a 2d function (with 1d active subspace) via random embedding.

```
generate a random matrix
A ∈ R^{D×d} : a_ij ~ N(0,1)
choose the bounded region set Z ⊂ R^d
D_0 ← ∅
for i = 1, 2, ··· do
    locate next sampling point
    z_{i+1} ← argmax_{z∈Z} a(z) ∈ R^d
    query
    D_{i+1} ← D_i ∪ {z_{i+1}, f(p_X(A z_{i+1}))}
    update GP
end for
```

In math form: generate a random matrix $\mathbf{A} \in \mathbb{R}^{D \times d} : a_{ij} \sim \mathcal{N}(0,1)$; choose the bounded region set $\mathcal{Z} \subset \mathbb{R}^d$; $\mathcal{D}_0 \leftarrow \varnothing$; **for** $i = 1, 2, \cdots$ **do**: locate next sampling point $\mathbf{z}_{i+1} \leftarrow \operatorname{argmax}_{\mathbf{z} \in \mathcal{Z}} a(\mathbf{z}) \in \mathbb{R}^d$; query $\mathcal{D}_{i+1} \leftarrow \mathcal{D}_i \cup \{z_{i+1}, f(p_{\mathcal{X}}(\mathbf{A}z_{i+1}))\}$; update GP; **end for**

High-dimensional: Gaussian random projection

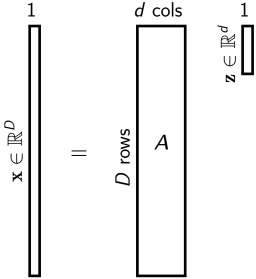


Figure: A random embedding or a random projection $\mathbf{x} = \mathbf{A}\mathbf{z}$ is built as a corollary from the Johnson-Lindenstrauss lemma, where $\mathbf{A}$ is a random normal matrix.

Theorem (Johnson-Lindenstrauss lemma (cf. Lemma 15 Mahoney 2016))

Given n points $\{\mathbf{x}_i\}_{i=1}^n$*, each of which is in* $\mathbb{R}^D$*,* $\mathbf{A} \sim \mathcal{MN}_{D\times d}(0, \mathbf{I}, \mathbf{I})$*, and let* $\mathbf{z} \in \mathbb{R}^d$ *defined as* $\mathbf{z} = \mathbf{A}^\top \mathbf{x}$*. Then, if* $d \geq \frac{9 \log n}{\varepsilon^2 - \varepsilon^3}$*, for some* $\varepsilon \in \left(0, \frac{1}{2}\right)$*, then with probability at least* $\frac{1}{2}$*, all pairwise distances are preserved, i.e. for all i, j, we have*

$$(1-\varepsilon)\|\mathbf{x}_i-\mathbf{x}_j\|_2^2 \leq \|\mathbf{z}_i-\mathbf{z}_j\|_2^2 \leq (1+\varepsilon)\|\mathbf{x}_i-\mathbf{x}_j\|_2^2 \tag{34}$$

High-dimensional: Gaussian random projection

Mostly follow Wang et al. 2013; Wang et al. 2016.

- D: original high dimensionality
- d_e: true effective dimensionality
- $d \geq d_e$: guess dimensionality
- theory says that if $\varepsilon = \frac{\log d}{\sqrt{d}}$
- which implies $\mathcal{Z} = [-\sqrt{d}, +\sqrt{d}]^d \subset \mathbb{R}^d$

Caveats (and modifications):

- may have to normalize the embedding to $\frac{1}{d}\mathbf{Az}$ instead of $\mathbf{Az}$
- need to translate from $[-\sqrt{d}, +\sqrt{d}]^D$ to $[\underline{\mathbf{x}}, \overline{\mathbf{x}}]$

Compared to the active subspace method:

- does not require the rotation matrix $\mathbf{W}_1^\top$ (hence avoid the manifold optimization constraint)
- comes at the cost of having $(1 - \varepsilon)$ successful rate for finding optimal
- could be reduced with multiple $\mathbf{A}$

Numerical benchmark: High-dimensional (with low effective dimensionality)

The modified ZDT1 function, which is defined on $[-1, 1]^D$, is

$$f_2(\mathbf{x}) = g\left(1 - \sqrt{\frac{x_1^2}{g}}\right), \qquad (35)$$

where $g = 1 + 9\left(\sum_{i=2}^{D} \frac{x_i}{D-1}\right)^2$.

- (non-unique) global minimizer $\mathbf{x}^* = [1, 0, \ldots, 0]$
- $f_2(\mathbf{x}^*) = 0$
- $D = 10^4$
- $d = 10$
- $d_e = 2$

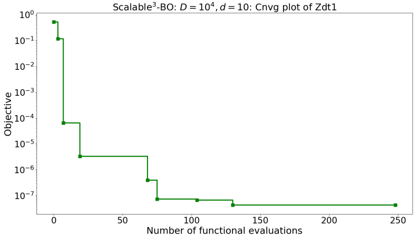


Figure: Convergence plot with $D = 10,000$, $d = 10$.

Numerical benchmark: High-dimensional (with low effective dimensionality)

The modified ZDT2 function, which is defined on $[-1, 1]^D$, is

$$f_2(\mathbf{x}) = g\left[1 - \left(\frac{x_1}{g}\right)^2\right], \qquad (36)$$

where $g = 1 + \left(9 \sum_{i=2}^{D} x_i\right)^2$.

- (non-unique) global minimizer $\mathbf{x}^* = [1, 0, \ldots, 0]$
- $f_2(\mathbf{x}^*) = 0$
- $D = 10^4$
- $d = 3$
- $d_e = 2$

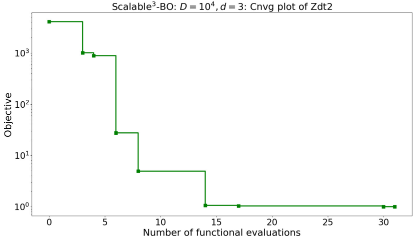


Figure: Convergence plot with $D = 10,000$, $d = 3$.

Asynchronous parallelism

Takeaway message:

- asynchronous scheduler reduces idle time for workers,
- benefit is maximized when simulator $f(\cdot)$ run-time vary (wildly).

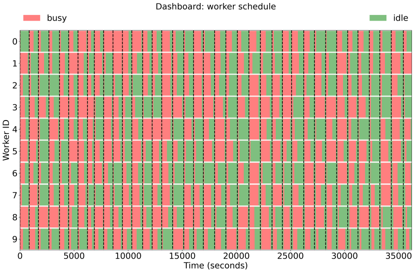


Figure: Batch-sequential parallel

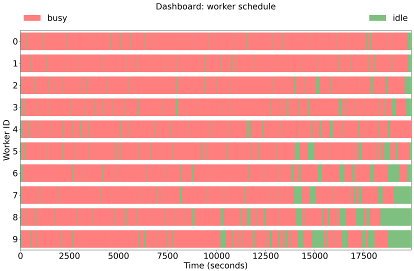
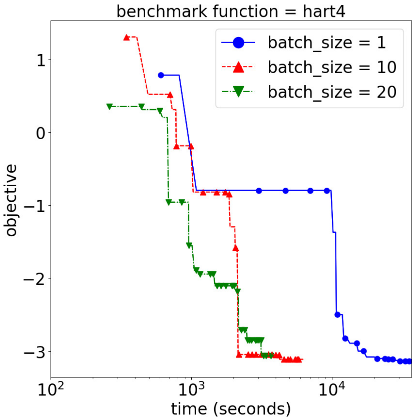


Figure: Asynchronous parallel

Numerical benchmark: parallelization

Hart4 function, $t \sim \mathcal{U}[30, 900]$s

$$f(\mathbf{x}) = \frac{1}{0.839}\left[1.1 - \sum_{i=1}^{4} \alpha_i \exp\left(-\sum_{j=4}^{3} A_{ij}(x_j - P_{ij})^2\right)\right], \qquad (37)$$



Scalable^3-BO algorithm

draw a random matrix $\mathbf{A} \in \mathbb{R}^{D \times d}$: $a_{ij} \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,1)$ ▷ $\mathbf{A}$ is a Gaussian random matrix
set $\mathcal{Z} \subset \mathbb{R}^d = [-\sqrt{d}, +\sqrt{d}]^d$
$\mathcal{D}_0 \leftarrow \varnothing$
while convergence criteria not met **do**
    while no available computational budget **do** ▷ threshold the computational budget
        wait and check periodically if there is any update
    end while
    update input, output, and status for all cases ▷ if not complete then hallucinate
    update dataset $\mathcal{D}_i$
    determine batch to fill ▷ exploit/explore or purely explore
    locate next sampling point: $\mathbf{z}_{i+1} = \text{argmax}_{\mathbf{z} \in \mathcal{Z}} \, a(\mathbf{z}) \in \mathbb{R}^d$
    embed, normalize, scale, and translate: $\mathbf{x}^*_{i+1} \leftarrow \underline{\mathbf{x}} + \frac{\frac{1}{d}\mathbf{A}\mathbf{z}_{i+1} + \sqrt{d}}{2(\sqrt{d})} \odot (\overline{\mathbf{x}} - \underline{\mathbf{x}})$ ▷ $\odot$ is Hadamard product
    project $\mathbf{x}_{i+1}$ to $\mathcal{X}$: $\mathbf{x}_{i+1} \leftarrow p_{\mathcal{X}}(\mathbf{x}^*_{i+1})$ ▷ $p_{\mathcal{X}}(\mathbf{z}) = \text{argmin}_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x} - \mathbf{z}\|_2$
    query $\mathcal{D}_{i+1} \leftarrow \mathcal{D}_i \cup \{\mathbf{z}_{i+1}, f(\mathbf{x}_{i+1})\}$
    hallucinate the sparse GP
    sample inducing inputs $\mathbf{Z}_\mathbf{u}$, where $|\mathbf{Z}_\mathbf{u}| = \min\{|\mathbf{X}|, m\}$ ▷ Latin hypercube sampling, $|\cdot|$ denotes cardinality
    update the sparse GP ▷ fully independent condition sparse GP
end while

Conclusion

In this talk, we

- replace classical GP with sparse GP for Big Data
- demonstrate scalability with 1M data points
- implement a random embedding based on Johnson-Lindenstrauss lemma for high-dimensional but low-effective-dimensional problems
- demonstrate with $D = 10,000$ but $d_e < 10$
- implement an asynchronous parallel feature to avoid downtime for computational workers
- demonstrate that larger batch means more effectiveness

Thank you for listening.

Any question?

Methodology:

- Anh Tran et al. (Aug. 2020d). “srMO-BO-3GP: A sequential regularized multi-objective constrained Bayesian optimization for design applications”. In: *Proceedings of the ASME 2020 IDETC/CIE*. vol. Volume 1: 40th Computers and Information in Engineering Conference. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers
- Anh Tran et al. (2020b). “aphBO-2GP-3B: A budgeted asynchronous-parallel multi-acquisition for known/unknown constrained Bayesian optimization on high-performing computing architecture”. In: *arXiv preprint arXiv:2003.09436*
- Anh Tran, Tim Wildey, and Scott McCann (2020). “sMF-BO-2CoGP: A sequential multi-fidelity constrained Bayesian optimization for design applications”. In: *Journal of Computing and Information Science in Engineering* 20.3, pp. 1–15
- Anh Tran, Tim Wildey, and Scott McCann (Aug. 2019). “sBF-BO-2CoGP: A sequential bi-fidelity constrained Bayesian optimization for design applications”. In: *Proceedings of the ASME 2019 IDETC/CIE*. vol. Volume 1: 39th Computers and Information in Engineering Conference. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. V001T02A073. American Society of Mechanical Engineers
- Anh Tran, Minh Tran, and Yan Wang (2019). “Constrained mixed-integer Gaussian mixture Bayesian optimization and its applications in designing fractal and auxetic metamaterials”. In: *Structural and Multidisciplinary Optimization*, pp. 1–24
- Anh Tran et al. (2019a). “pBO-2GP-3B: A batch parallel known/unknown constrained Bayesian optimization with feasibility classification and its applications in computational fluid dynamics”. In: *Computer Methods in Applied Mechanics and Engineering* 347, pp. 827–852

Applications:

- Anh Tran et al. (2020c). "Multi-fidelity machine-learning with uncertainty quantification and Bayesian optimization for materials design: Application to ternary random alloys". In: *The Journal of Chemical Physics* 153 (7), p. 074705
- Anh Tran et al. (2020a). "An active-learning high-throughput microstructure calibration framework for process-structure linkage in materials informatics". In: *Acta Materialia* 194, pp. 80–92
- Stefano Travaglino et al. (2020). "Computational optimization study of transcatheter aortic valve leaflet design using porcine and bovine leaflets". In: *Journal of Biomechanical Engineering* 142 (1)
- Anh Tran et al. (2019b). "WearGP: A computationally efficient machine learning framework for local erosive wear predictions via nodal Gaussian processes". In: *Wear* 422, pp. 9–26
- Anh Tran, Lijuan He, and Yan Wang (2018). "An efficient first-principles saddle point searching method based on distributed kriging metamodels". In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering* 4.1, p. 011006

References I

- Bauer, Matthias, Mark van der Wilk, and Carl Edward Rasmussen (2016). "Understanding probabilistic sparse Gaussian process approximations". In: *Advances in neural information processing systems* 29, pp. 1533–1541.
- Bonilla, Edwin V, Karl Krauth, and Amir Dezfouli (2019). "Generic inference in latent Gaussian process models.". In: *Journal of Machine Learning Research* 20.117, pp. 1–63.
- Burt, David R., Carl Edward Rasmussen, and Mark van der Wilk (2020). "Convergence of Sparse Variational Inference in Gaussian Processes Regression". In: *Journal of Machine Learning Research* 21.131, pp. 1–63.
- Chalupka, Krzysztof, Christopher KI Williams, and Iain Murray (2013). "A framework for evaluating approximation methods for Gaussian process regression". In: *Journal of Machine Learning Research* 14.Feb, pp. 333–350.

References II

- Constantine, Paul G (2015). *Active subspaces: Emerging ideas for dimension reduction in parameter studies*. SIAM.
- Constantine, Paul G, Eric Dow, and Qiqi Wang (2014). "Active subspace methods in theory and practice: applications to kriging surfaces". In: *SIAM Journal on Scientific Computing* 36.4, A1500–A1524.
- Diederik, P Kingma and Max Welling (2014). "Auto-encoding variational bayes". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. Vol. 1.
- Frigola, Roger, Yutian Chen, and Carl Edward Rasmussen (2014). "Variational Gaussian Process State-Space Models". In: *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc.
- Lawrence, Neil D (2016). "Introduction to gaussian processes". In: *MLSS* 8, p. 504. URL: inverseprobability.com/talks/slides/gp_mlss16.pdf.

References III

- Li, Mu, James Tin-Yau Kwok, and Baoliang Lü (2010). "Making large-scale Nyström approximation possible". In: *ICML 2010-Proceedings, 27th International Conference on Machine Learning*, p. 631.
- Mahoney, Michael W (2016). "Lecture notes on randomized linear algebra". In: *arXiv preprint arXiv:1608.04481*.
- Matthews, Alexander G de G et al. (2016). "On sparse variational methods and the Kullback-Leibler divergence between stochastic processes". In: *Artificial Intelligence and Statistics*. PMLR, pp. 231–239.
- Quiñonero-Candela, Joaquin and Lars Kai Hansen (2004). "Learning with uncertainty-Gaussian processes and relevance vector machines". In: *Technical University of Denmark, Copenhagen*.

References IV

- Quiñonero-Candela, Joaquin and Carl Edward Rasmussen (2005). "A unifying view of sparse approximate Gaussian process regression". In: *Journal of Machine Learning Research* 6.Dec, pp. 1939–1959.
- Quiñonero-Candela, Joaquin, Carl Edward Rasmussen, and Christopher KI Williams (2007). "Approximation methods for Gaussian process regression". In: *Large-scale kernel machines*, pp. 203–224.
- Rasmussen, Carl Edward (2006). *Gaussian processes in machine learning*. MIT Press.
- Titsias, Michalis (2009a). "Variational learning of inducing variables in sparse Gaussian processes". In: *Artificial Intelligence and Statistics*, pp. 567–574.
- Titsias, Michalis K (2009b). "Variational model selection for sparse Gaussian process regression". In: *Report, University of Manchester, UK*.

References V

- Tran, Anh, Lijuan He, and Yan Wang (2018). "An efficient first-principles saddle point searching method based on distributed kriging metamodels". In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering* 4.1, p. 011006.
- Tran, Anh, Minh Tran, and Yan Wang (2019). "Constrained mixed-integer Gaussian mixture Bayesian optimization and its applications in designing fractal and auxetic metamaterials". In: *Structural and Multidisciplinary Optimization*, pp. 1–24.

References VI

- Tran, Anh, Tim Wildey, and Scott McCann (Aug. 2019). "sBF-BO-2CoGP: A sequential bi-fidelity constrained Bayesian optimization for design applications". In: *Proceedings of the ASME 2019 IDETC/CIE*. Vol. Volume 1: 39th Computers and Information in Engineering Conference. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. V001T02A073. American Society of Mechanical Engineers.
- – (2020). "sMF-BO-2CoGP: A sequential multi-fidelity constrained Bayesian optimization for design applications". In: *Journal of Computing and Information Science in Engineering* 20.3, pp. 1–15.

References VII

- Tran, Anh et al. (2019a). "pBO-2GP-3B: A batch parallel known/unknown constrained Bayesian optimization with feasibility classification and its applications in computational fluid dynamics". In: *Computer Methods in Applied Mechanics and Engineering* 347, pp. 827–852.
- Tran, Anh et al. (2019b). "WearGP: A computationally efficient machine learning framework for local erosive wear predictions via nodal Gaussian processes". In: *Wear* 422, pp. 9–26.
- Tran, Anh et al. (2020a). "An active-learning high-throughput microstructure calibration framework for process-structure linkage in materials informatics". In: *Acta Materialia* 194, pp. 80–92.
- Tran, Anh et al. (2020b). "aphBO-2GP-3B: A budgeted asynchronous-parallel multi-acquisition for known/unknown constrained Bayesian optimization on high-performing computing architecture". In: *arXiv preprint arXiv:2003.09436*.

References VIII

Tran, Anh et al. (2020c). "Multi-fidelity machine-learning with uncertainty quantification and Bayesian optimization for materials design: Application to ternary random alloys". In: *The Journal of Chemical Physics* 153 (7), p. 074705.

Tran, Anh et al. (Aug. 2020d). "srMO-BO-3GP: A sequential regularized multi-objective constrained Bayesian optimization for design applications". In: *Proceedings of the ASME 2020 IDETC/CIE*. Vol. Volume 1: 40th Computers and Information in Engineering Conference. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers.

Travaglino, Stefano et al. (2020). "Computational optimization study of transcatheter aortic valve leaflet design using porcine and bovine leaflets". In: *Journal of Biomechanical Engineering* 142 (1).

References IX

- Vanhatalo, Jarno et al. (2012). “Bayesian modeling with Gaussian processes using the GPstuff toolbox”. In: *arXiv preprint arXiv:1206.5754*.
- – (2013). “GPstuff: Bayesian modeling with Gaussian processes”. In: *Journal of Machine Learning Research* 14.Apr, pp. 1175–1179.
- Wang, Ziyu et al. (2013). “Bayesian optimization in high dimensions via random embeddings”. In: AAAI Press/International Joint Conferences on Artificial Intelligence.
- Wang, Ziyu et al. (2016). “Bayesian optimization in a billion dimensions via random embeddings”. In: *Journal of Artificial Intelligence Research* 55, pp. 361–387.
- Williams, Christopher and Matthias Seeger (2001). “Using the Nyström method to speed up kernel machines”. In: *Advances in neural information processing systems* 13, pp. 682–688.