Gaussian Process in Practice: Scalability and Uncertainty

Zhenwen Dai

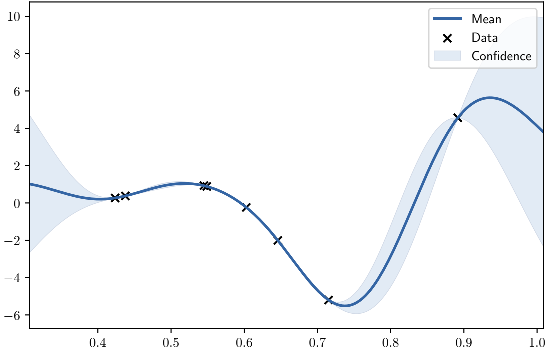
Amazon

2019-04-09

Gaussian process

$$\mathbf{y} = (y_1, \ldots, y_N), \quad \mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^\top$$

$$p(\mathbf{y}|\mathbf{f}) = \mathcal{N}\left(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I}\right), \quad p(\mathbf{f}|\mathbf{X}) = \mathcal{N}\left(\mathbf{f}|0, \mathbf{K}(\mathbf{X}, \mathbf{X})\right)$$

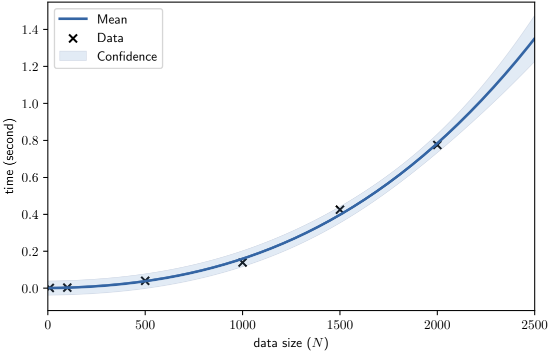


The computational time of GP

- The time complexity of Gaussian process is $O(N^3)$.
- Take 1D regression problem as an example:
  - The input and output dimensionality are both one.
  - We generate generate synthetic data.
  - Measure the time that it takes for computing the log-likelihood.

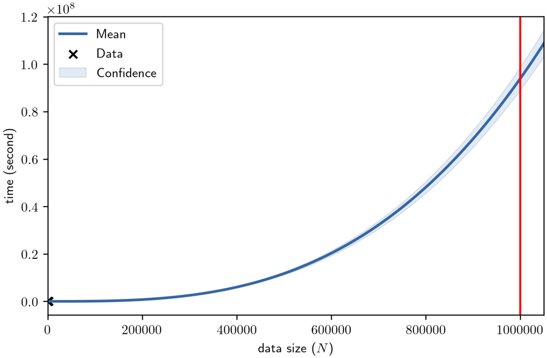
Empirical analysis of computational time

- I collect the run time for $N = \{10, 100, 500, 1000, 1500, 2000\}$.
- They take 1.3ms, 8.5ms, 28ms, 0.12s, 0.29s, 0.76s.



What if we have 1 million data points?

- The mean of predicted computational time is 9.4×10^7 seconds ≈ 2.98 years.



What about waiting for faster computers?

- Computational time $= \frac{\text{amount of work}}{\text{computer speed}}$
- If the computer speed increase at the pace of 20% year over year:
  - After 10 years, it will take about 176 days.
  - After 50 years, it will take about 2.9 hours.
- If we double the size of data, it takes 11.4 years to catch up.

What about parallel computing / GPU?

- Ongoing works about speeding up Cholesky decomposition with multi-core CPU or GPU.
- Main limitation: heavy communication and shared memory.

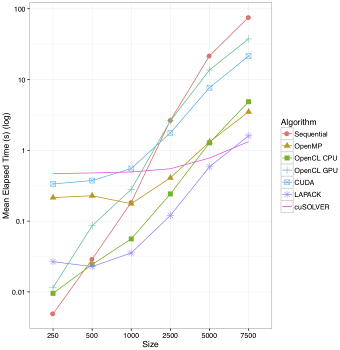


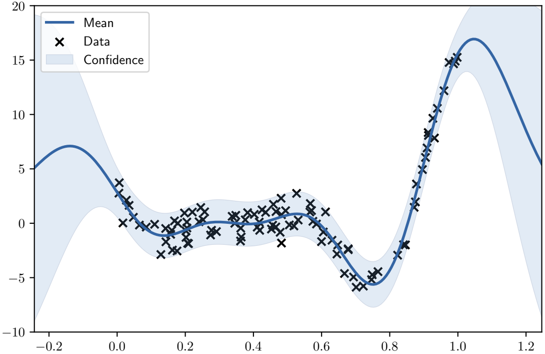
Figure 1: (RUSCHEL 2016)

Other approaches

- Apart from speeding up the exact computation, there have been a lot of works on approximation of GP inference.
- These methods often target at some specific scenario and provide good approximation for the targeted scenarios.
- Provide an overview about common approximations.

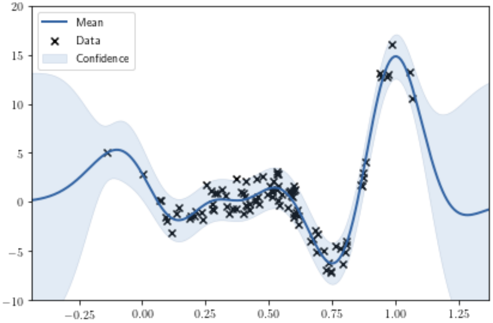
Big data (?)

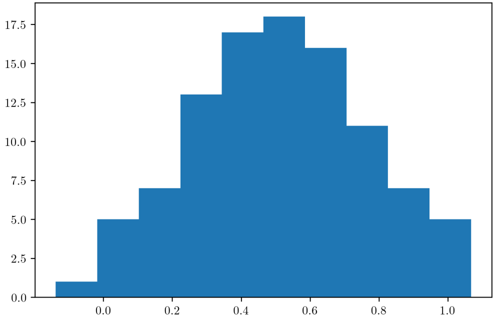
- lots of data $\neq$ complex function
- In real world problems, we often collect a lot of data for modeling relatively simple relations.



Data subsampling?

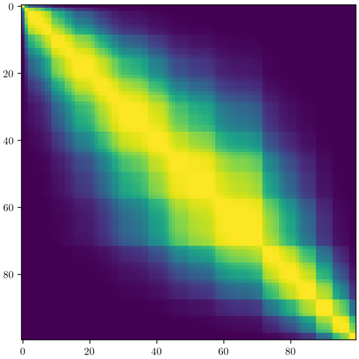
- Real data often do not evenly distributed.
- We tend to get a lot of data on common cases and very few data on rare cases.

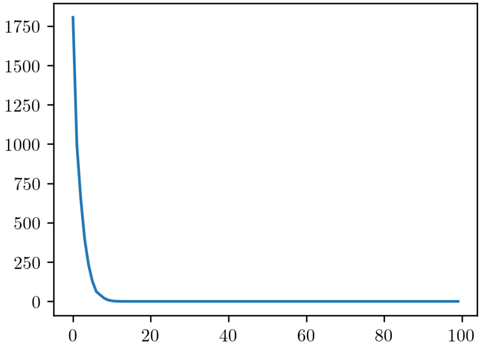




Covariance matrix of redundant data

- With redundant data, the covariance matrix becomes low rank.
- What about low rank approximation?





Low-rank approximation

- Let's recall the log-likelihood of GP:

$$\log p(\mathbf{y}|\mathbf{X}) = \log \mathcal{N}\left(\mathbf{y}|0, \mathbf{K} + \sigma^2\mathbf{I}\right),$$

  where $\mathbf{K}$ is the covariance matrix computed from $\mathbf{X}$ according to the kernel function $k(\cdot,\cdot)$ and σ^2 is the variance of the Gaussian noise distribution.
- Assume $\mathbf{K}$ to be low rank.
- This leads to Nyström approximation by Williams and Seeger (2001).

Nyström approximation (Williams and Seeger 2001)

- Let's randomly pick a subset from the training data: $\mathbf{Z} \in \mathbb{R}^{M \times Q}$.
- Approximate the covariance matrix $\mathbf{K}$ by $\tilde{\mathbf{K}}$:

$$\tilde{\mathbf{K}} = \mathbf{K}_z \mathbf{K}_{zz}^{-1} \mathbf{K}_z^\top,$$

  where $\mathbf{K}_z = \mathbf{K}(\mathbf{X}, \mathbf{Z})$ and $\mathbf{K}_{zz} = \mathbf{K}(\mathbf{Z}, \mathbf{Z})$.
- Note that $\tilde{\mathbf{K}} \in \mathbb{R}^{N \times N}$, $\mathbf{K}_z \in \mathbb{R}^{N \times M}$ and $\mathbf{K}_{zz} \in \mathbb{R}^{M \times M}$.
- The log-likelihood is approximated by

$$\log p(\mathbf{y}|\mathbf{X}, \theta) \approx \log \mathcal{N}\left(\mathbf{y}|0, \mathbf{K}_z \mathbf{K}_{zz}^{-1} \mathbf{K}_z^\top + \sigma^2 \mathbf{I}\right).$$

Efficient computation using Woodbury formula

- The naive formulation does not bring any computational benefits.

$$\tilde{\mathcal{L}} = -\frac{1}{2}\log|2\pi(\tilde{\mathbf{K}} + \sigma^2\mathbf{I})| - \frac{1}{2}\mathbf{y}^\top(\tilde{\mathbf{K}} + \sigma^2\mathbf{I})^{-1}\mathbf{y}$$

- Apply the Woodbury formula:

$$(\mathbf{K}_z\mathbf{K}_{zz}^{-1}\mathbf{K}_z^\top + \sigma^2\mathbf{I})^{-1} = \sigma^{-2}\mathbf{I} - \sigma^{-4}\mathbf{K}_z(\mathbf{K}_{zz} + \sigma^{-2}\mathbf{K}_z^\top\mathbf{K}_z)^{-1}\mathbf{K}_z^\top$$

- Note that $(\mathbf{K}_{zz} + \sigma^{-2}\mathbf{K}_z^\top\mathbf{K}_z) \in \mathbb{R}^{M\times M}$.
- The computational complexity reduces to $O(NM^2)$.

Gaussian process with pseudo data (1)

- Snelson and Ghahramani (2006) propose the idea of having pseudo data. This approach is later referred to as Fully Independent Training Conditional (FITC).
- Augment the training data $(\mathbf{X}, \mathbf{y})$ with pseudo data $\mathbf{u}$ at location $\mathbf{Z}$.

$$p\left(\begin{bmatrix}\mathbf{y}\\ \mathbf{u}\end{bmatrix} | \begin{bmatrix}\mathbf{X}\\ \mathbf{Z}\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{y}\\ \mathbf{u}\end{bmatrix} |0, \begin{bmatrix}\mathbf{K}_{ff} + \sigma^2\mathbf{I} & \mathbf{K}_{fu}\\ \mathbf{K}_{fu}^\top & \mathbf{K}_{uu}\end{bmatrix}\right)$$

where $\mathbf{K}_{ff} = \mathbf{K}(\mathbf{X}, \mathbf{X})$, $\mathbf{K}_{fu} = \mathbf{K}(\mathbf{X}, \mathbf{Z})$ and $\mathbf{K}_{uu} = \mathbf{K}(\mathbf{Z}, \mathbf{Z})$.

Gaussian process with pseudo data (2)

- Thanks to the marginalization property of Gaussian distribution,

$$p(\mathbf{y}|\mathbf{X}) = \int_{\mathbf{u}} p(\mathbf{y}, \mathbf{u}|\mathbf{X}, \mathbf{Z}).$$

- Further re-arrange the notation:

$$p(\mathbf{y}, \mathbf{u}|\mathbf{X}, \mathbf{Z}) = p(\mathbf{y}|\mathbf{u}, \mathbf{X}, \mathbf{Z})p(\mathbf{u}|\mathbf{Z})$$

where $p(\mathbf{u}|\mathbf{Z}) = \mathcal{N}\left(\mathbf{u}|0, \mathbf{K}_{uu}\right)$,
$p(\mathbf{y}|\mathbf{u}, \mathbf{X}, \mathbf{Z}) = \mathcal{N}\left(\mathbf{y}|\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{u}, \mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{fu}^{\top} + \sigma^2\mathbf{I}\right)$.

FITC approximation (1)

- So far, $p(\mathbf{y}|\mathbf{X})$ has not been changed, but there is no speed-up, $\mathbf{K}_{ff} \in \mathbb{R}^{N \times N}$ in $\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{fu}^{\top} + \sigma^2\mathbf{I}$.
- The FITC approximation assumes

$$\tilde{p}(\mathbf{y}|\mathbf{u}, \mathbf{X}, \mathbf{Z}) = \mathcal{N}\left(\mathbf{y}|\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{u}, \mathbf{\Lambda} + \sigma^2\mathbf{I}\right),$$

where $\mathbf{\Lambda} = (\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{fu}^{\top}) \circ \mathbf{I}$.

FITC approximation (2)

- Marginalize $\mathbf{u}$ from the model definition:

$$\tilde{p}(\mathbf{y}|\mathbf{X},\mathbf{Z}) = \mathcal{N}\left(\mathbf{y}|0, \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{fu}^\top + \mathbf{\Lambda} + \sigma^2\mathbf{I}\right)$$

- Woodbury formula can be applied in the sam way as in Nyström approximation:

$$(\mathbf{K}_z\mathbf{K}_{zz}^{-1}\mathbf{K}_z^\top + \mathbf{\Lambda} + \sigma^2\mathbf{I})^{-1} = \mathbf{A} - \mathbf{A}\mathbf{K}_z(\mathbf{K}_{zz} + \mathbf{K}_z^\top\mathbf{A}\mathbf{K}_z)^{-1}\mathbf{K}_z^\top\mathbf{A},$$

where $\mathbf{A} = (\mathbf{\Lambda} + \sigma^2\mathbf{I})^{-1}$.

FITC approximation (3)

- FITC allows the pseudo data not being a subset of training data.
- The inducing inputs $\mathbf{Z}$ can be optimized via gradient optimization.
- Like Nyström approximation, when taking all the training data as inducing inputs, the FITC approximation is equivalent to the original GP:

$$\tilde{p}(\mathbf{y}|\mathbf{X}, \mathbf{Z} = \mathbf{X}) = \mathcal{N}\left(\mathbf{y}|0, \mathbf{K}_{ff} + \sigma^2\mathbf{I}\right)$$

- FITC can be combined easily with expectation propagation (EP).
- (Bui, Yan, and Turner 2017) provides an overview and a nice connection with variational sparse GP.

Model Approximation vs. Approximate Inference

When the exact model/inference is intractable, typically there are two types of approaches:

- Approximate the original model with a simpler one such that inference becomes tractable, like Nyström approximation, FITC.
- Keep the original model but derive an approximate inference method which is often *not* able to return the true answer, like variational inference.

Model Approximation vs. Approximate Inference

A problem with model approximation is that

- when an approximated model requires some tuning, e.g., for hyper-parameters, it is unclear how to improve it based on training data.
- In the case of FITC, we know the model is correct if $\mathbf{Z} = \mathbf{X}$, however, optimizing $\mathbf{Z}$ will not necessarily lead to a better location.
- In fact, optimizing $\mathbf{Z}$ can lead to overfitting. (Quiñonero-Candela and Rasmussen 2005)

Variational Sparse Gaussian Process (1)

- (Titsias 2009) introduces a variational approach for sparse GP.
- It follows the same concept of pseudo data:

$$p(\mathbf{y}|\mathbf{X}) = \int_{\mathbf{f},\mathbf{u}} p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u},\mathbf{X},\mathbf{Z})p(\mathbf{u}|\mathbf{Z})$$

where $p(\mathbf{u}|\mathbf{Z}) = \mathcal{N}\left(\mathbf{u}|0,\mathbf{K}_{uu}\right)$,
$p(\mathbf{y}|\mathbf{u},\mathbf{X},\mathbf{Z}) = \mathcal{N}\left(\mathbf{y}|\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{u},\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{fu}^{\top} + \sigma^2\mathbf{I}\right)$.

Variational Sparse Gaussian Process (2)

- Instead of approximate the model, (Titsias 2009) derives a variational lower bound.
- Normally, a variational lower bound of a marginal likelihood, also known as evidence lower bound (ELBO), looks like

$$\begin{aligned}\log p(\mathbf{y}|\mathbf{X}) &= \log \int_{\mathbf{f},\mathbf{u}} p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u},\mathbf{X},\mathbf{Z})p(\mathbf{u}|\mathbf{Z}) \\ &\geq \int_{\mathbf{f},\mathbf{u}} q(\mathbf{f},\mathbf{u}) \log \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u},\mathbf{X},\mathbf{Z})p(\mathbf{u}|\mathbf{Z})}{q(\mathbf{f},\mathbf{u})}.\end{aligned}$$

Special Variational Posterior

- (Titsias 2009) defines an unusual variational posterior:

$$q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f}|\mathbf{u}, \mathbf{X}, \mathbf{Z})q(\mathbf{u}), \quad \text{where } q(\mathbf{u}) = \mathcal{N}\left(\mathbf{u}|\mu, \Sigma\right).$$

- Plug it into the lower bound:

$$\begin{aligned}
\mathcal{L} &= \int_{\mathbf{f},\mathbf{u}} p(\mathbf{f}|\mathbf{u}, \mathbf{X}, \mathbf{Z})q(\mathbf{u}) \log \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u}, \mathbf{X}, \mathbf{Z})p(\mathbf{u}|\mathbf{Z})}{p(\mathbf{f}|\mathbf{u}, \mathbf{X}, \mathbf{Z})q(\mathbf{u})} \\
&= \langle \log p(\mathbf{y}|\mathbf{f}) \rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X},\mathbf{Z})q(\mathbf{u})} - \mathrm{KL}\left(q(\mathbf{u}) \parallel p(\mathbf{u}|\mathbf{Z})\right) \\
&= \left\langle \log \mathcal{N}\left(\mathbf{y}|\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{u}, \sigma^2\mathbf{I}\right) \right\rangle_{q(\mathbf{u})} - \mathrm{KL}\left(q(\mathbf{u}) \parallel p(\mathbf{u}|\mathbf{Z})\right)
\end{aligned}$$

Tighten the Bound

- Find the optimal parameters of $q(\mathbf{u})$:

$$\mu^*, \Sigma^* = \arg\max_{\mu,\Sigma} \mathcal{L}(\mu, \Sigma).$$

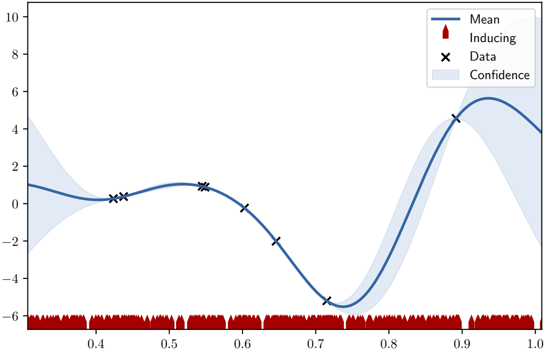
- Make the bound as tight as possible by plugging in μ^* and Σ^*:

$$\mathcal{L} = \log \mathcal{N}\left(\mathbf{y}|0, \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{fu}^\top + \sigma^2\mathbf{I}\right) - \frac{1}{2\sigma^2}\mathrm{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{fu}^\top\right).$$

- The overall complexity of the lower bound remains $O(NM^2)$.

Variational sparse GP

- Note that $\mathcal{L}$ is not a valid log-pdf, $\int_{\mathbf{y}} \exp(\mathcal{L}(\mathbf{y})) \leq 1$, due to the trace term.
- As inducing points are variational parameters, optimizing the inducing inputs $\mathbf{Z}$ always leads to a better bound.
- The model does not "overfit" with too many inducing points.



Are big covariance matrices always (almost) low-rank?

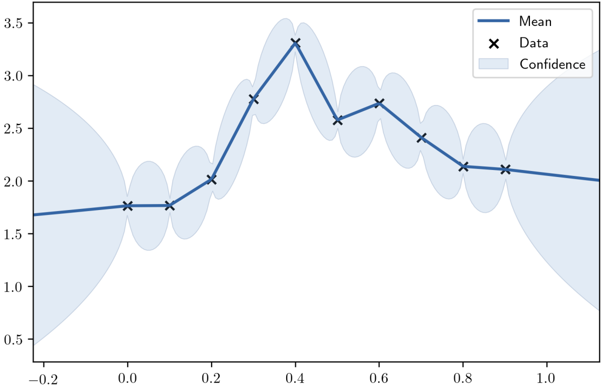
- Of course, not.
- A time series example

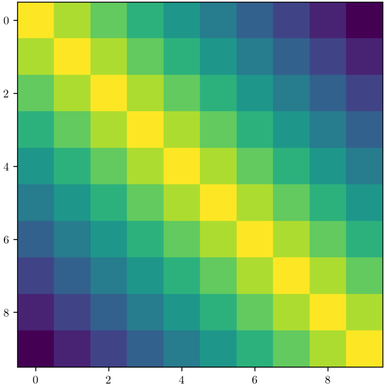
$$y = f(t) + \epsilon$$

- The data are collected with even time interval continuously.

A time series example: 10 data points

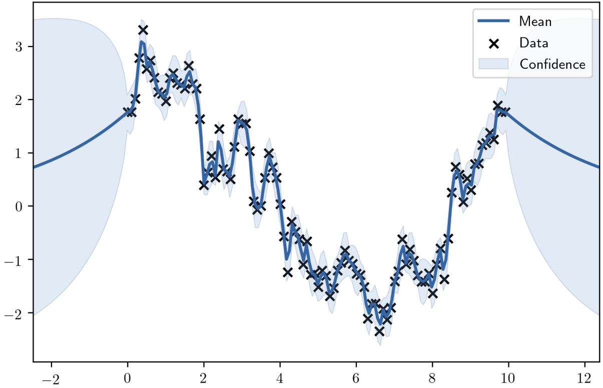
- When we observe until $t = 1.0$:

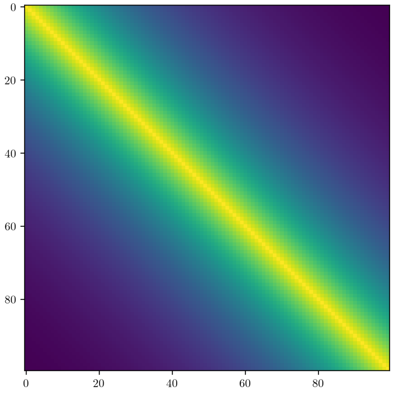




A time series example: 100 data points

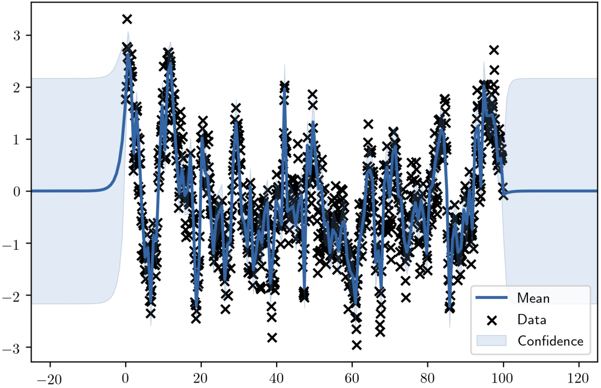
- When we observe until $t = 10.0$:

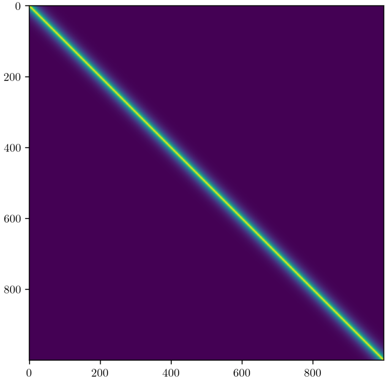




A time series example: 1000 data points

- When we observe until $t = 100.0$:





Banded precision matrix

- For the kernels like the Matern family, the precision matrix is banded.
- For example, given a Matern$\frac{1}{2}$ or known as exponential kernel: $k(x, x') = \sigma^2 \exp(-\frac{|x-x'|}{l^2})$.

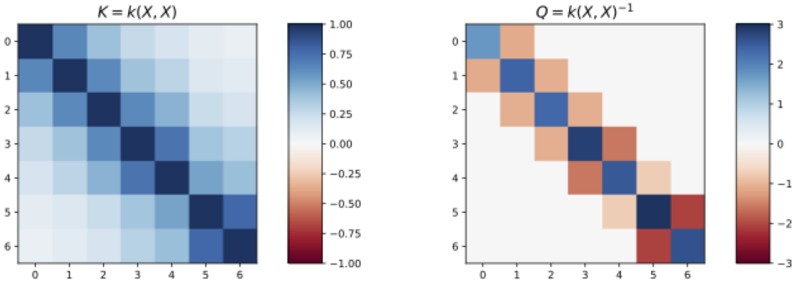


Figure 2: This slide is taken from Nicolas Durrande (2019)

Closed form precision matrix

- The precision matrix of Matern kernels can be computed in closed form.
- The lower triangular matrix from the Cholesky decomposition of the precision matrix is banded as well.

$$\log(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\log|2\pi(LL^\top)^{-1}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{y}\mathbf{y}^\top LL^\top\right)$$

  where L is the lower triangular matrix from the Cholesky decomposition of the precision matrix Q, $Q = LL^\top$.

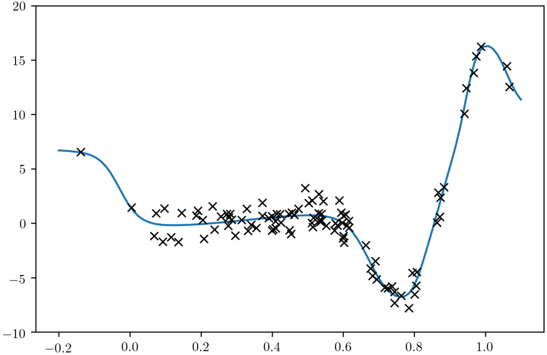
- The computational complexity becomes $O(N)$.

Other approximations

- deterministic/stochastic frequency approximation
- distributed approximation
- conjugate gradient methods for covariance matrix inversion

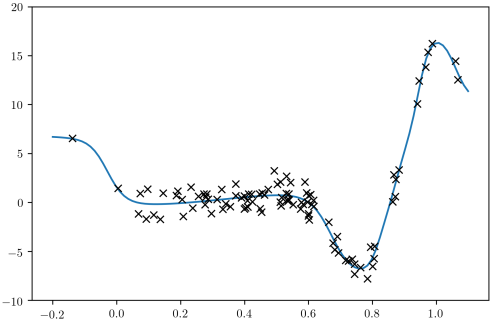
Why GP?

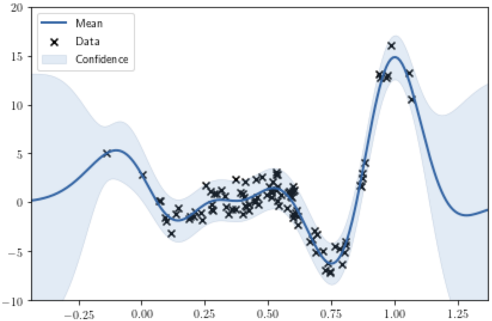
- Why do we spend all these energy on speeding up GP?
- Only for a non-parametric regressor?
- What about fitting a neural network?



What is the difference?

- The error bar!

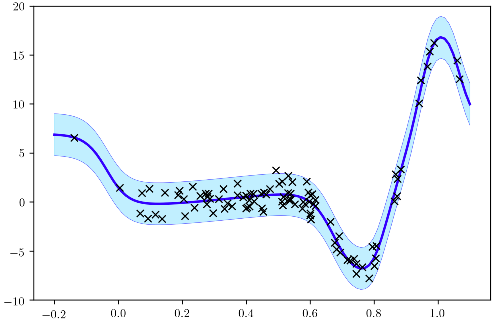




Can we learn an error bar with NN?

- Of course, we can. Let's add a likelihood to the neural network:

$$p(y|x) = \mathcal{N}\left(y|f_\theta(x), \sigma^2\right)$$



- Now, we have an error bar for our neural network. Are they the same?

Two types of uncertainty

- In our GP regression model, we have two “layers” of distributions:

$$p(\mathbf{y}|\mathbf{f}) = \mathcal{N}\left(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I}\right), \quad p(\mathbf{f}|\mathbf{X}) = \mathcal{N}\left(\mathbf{f}|0, \mathbf{K}(\mathbf{X}, \mathbf{X})\right)$$

- Aleatoric uncertainty: the uncertainty about the noise in individual data points
- Epistemic uncertainty: the uncertainty in the model

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

Automated decision making

- Under the model assumption, the epistemic uncertainty allows us to know what the model *does not* know.
- This enables us to trade-off among different choices with limited information.
- Example: Bayesian optimization

$$x^* = \arg\min_x f(x)$$

Bayesian optimization

A surrogate model guided search

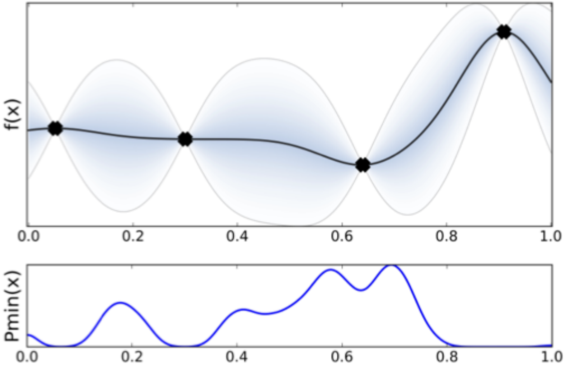


Figure 3: A 1D example

Possibility represented by uncertainty

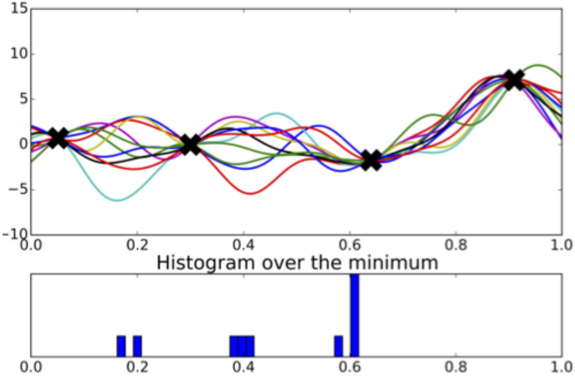


Figure 4: A 1D example

Balance exploitation and exploration

- **Exploitation**: Evaluate at the known best location will return the minimal value so far, but we learn nothing new.
- **Exploration**: Improve the understanding of the objective function, but may not be better than the current minimum.

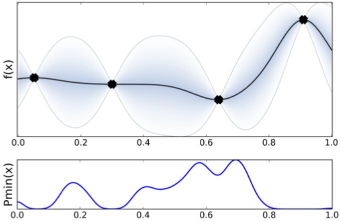


Figure 5: A 1D example

Acquisition function

- Formulate the policy of the exploitation and exploration tradeoff.
- The utility function about improvement:

$$u(f) = \max(0, f' - f)$$

- The expected improvement under our surrogate model:

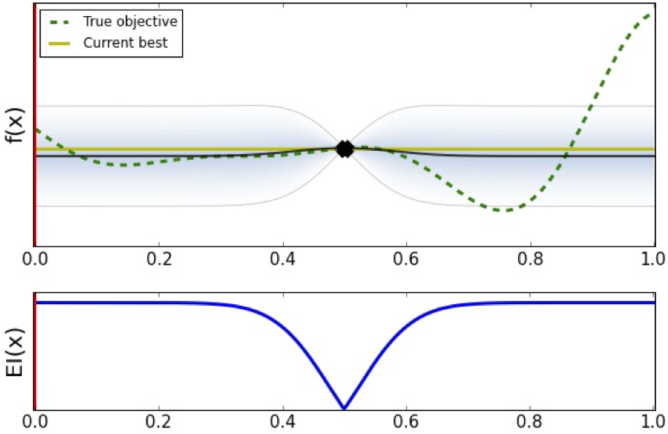
$$\begin{aligned} a_{\mathsf{EI}}(x) &= \int u(f) p(f|x, \mathcal{D}) \mathrm{d}f \\ &= \int \max(0, f' - f) \mathcal{N}\left(f|m(x), c(x)\right) \mathrm{d}f \end{aligned}$$

A BO algorithm

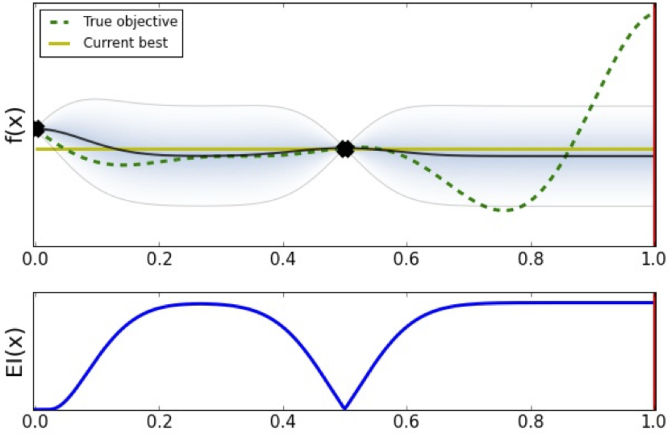
Loop of Bayesian optimization:

1. Evaluate the objective function.
2. Update the surrogate model.
3. Select to the next location according to the acquisition function.

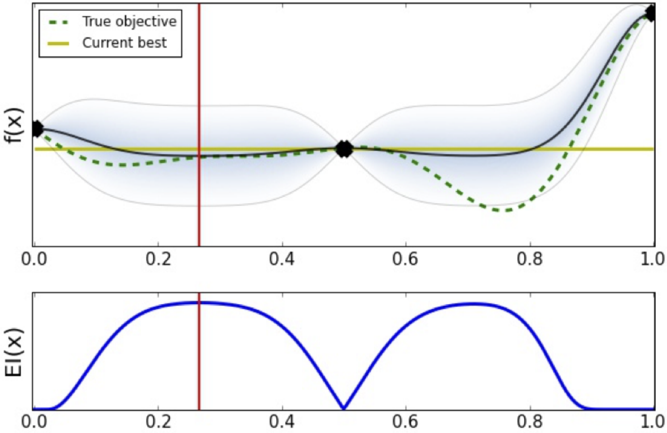
BO Example


True objective
Current best
f(x)
0.0
0.2
0.4
0.6
0.8
1.0
EI(x)
0.0
0.2
0.4
0.6
0.8
1.0

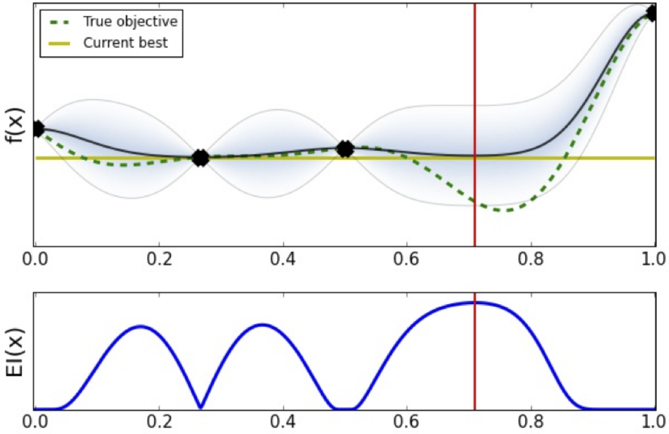
BO Example


True objective
Current best
f(x)
0.0
0.2
0.4
0.6
0.8
1.0
EI(x)
0.0
0.2
0.4
0.6
0.8
1.0

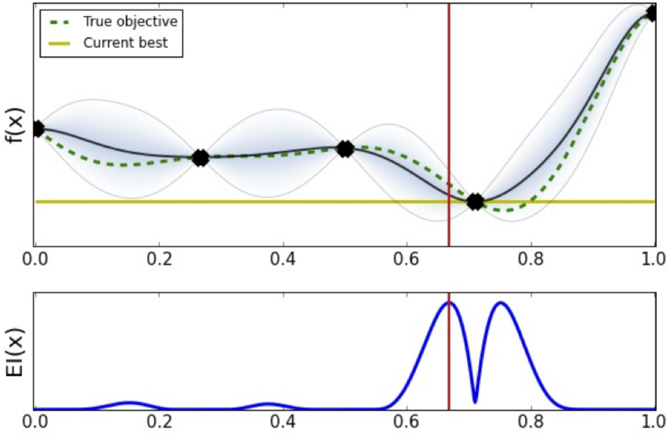
BO Example


True objective
Current best
f(x)
0.0
0.2
0.4
0.6
0.8
1.0
EI(x)
0.0
0.2
0.4
0.6
0.8
1.0

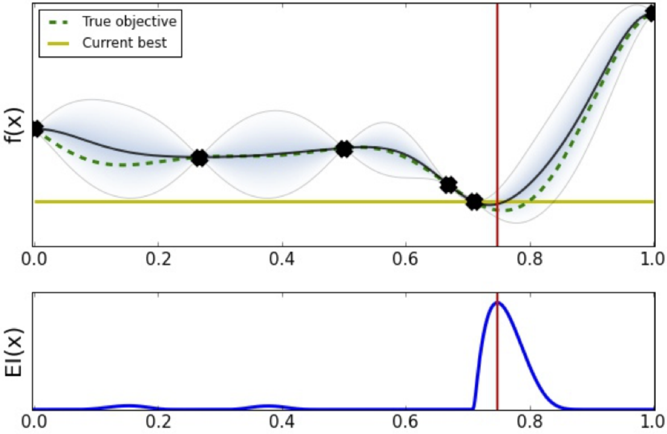
BO Example


True objective
Current best
f(x)
0.0
0.2
0.4
0.6
0.8
1.0
EI(x)
0.0
0.2
0.4
0.6
0.8
1.0

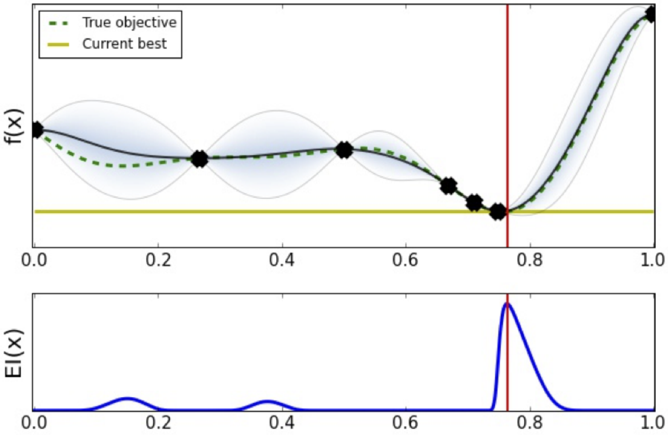
BO Example


True objective
Current best
f(x)
0.0
0.2
0.4
0.6
0.8
1.0
EI(x)
0.0
0.2
0.4
0.6
0.8
1.0

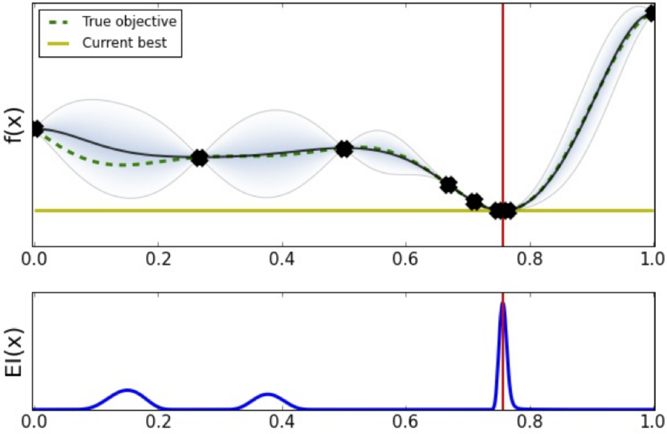
BO Example


True objective
Current best
f(x)
0.0
0.2
0.4
0.6
0.8
1.0
EI(x)
0.0
0.2
0.4
0.6
0.8
1.0

BO Example


True objective
Current best
f(x)
0.0
0.2
0.4
0.6
0.8
1.0
EI(x)
0.0
0.2
0.4
0.6
0.8
1.0

BO Example


True objective
Current best
f(x)
0.0
0.2
0.4
0.6
0.8
1.0
EI(x)
0.0
0.2
0.4
0.6
0.8
1.0

Challenges in Bayesian Optimization

- Optimizing the acquisition may be hard.
- With a high dimensional search problem, surrogate modeling becomes hard.
- Structured inputs can be hard to handle.
- Non-stationarity of an objective function
- Model mismatch
- Unknown safety constraints
- Warm-starting

- Thank you!

Lab session

- Please download the Jupyter notebook for the lab session from the following link:

http://gpss.cc/gpss18/labs/GPSS_Lab3_2018.ipynb

References I

Bui, Thang D, Josiah Yan, and Richard E Turner. 2017. “A Unifying Framework for Gaussian Process Pseudo-Point Approximations Using Power Expectation Propagation.” *Journal of Machine Learning Research* 18: 3649–3720.

Durrande, Nicolas, Vincent Adam, Lucas Bordeaux, Stefanos Eleftheriadis, and James Hensman. 2019. “Banded Matrix Operators for Gaussian Markov Models in the Automatic Differentiation Era.” *arXiv*.

Quiñonero-Candela, Joaquin, and Carl Edward Rasmussen. 2005. “A Unifying View of Sparse Approximate Gaussian Process Regression.” *Journal of Machine Learning Research* 6: 1939–59.

RUSCHEL, JOÃO PAULO TARASCONI. 2016. “Parallel Implementations of the Cholesky Decomposition on Cpus and Gpus.” Bachelor Thesis, Federal University of Rio Grande do Sul.

References II

Snelson, Edward, and Zoubin Ghahramani. 2006. "Sparse Gaussian Processes Using Pseudo-Inputs." In *Advances in Neural Information Processing Systems*, 1257–64.

Titsias, Michalis. 2009. "Variational Learning of Inducing Variables in Sparse Gaussian Processes." In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, 567–74.

Williams, Christopher K. I., and Matthias Seeger. 2001. "Using the Nyström Method to Speed up Kernel Machines." In *Advances in Neural Information Processing Systems*, 682–88.