Gaussian processes

*A hands-on tutorial*

**Slides and code:**  https://github.com/paraklas/GPTutorial

Paris Perdikaris
Massachusetts Institute of Technology, Department of Mechanical Engineering

**Web:**  http://web.mit.edu/parisp/www/

**Email:**  parisp@mit.edu

ICERM
Providence, RI
June 5th, 2017
<table>
<thead>
<tr>
<th>Time</th>
<th>Description</th>
<th>Speaker</th>
<th>Location</th>
<th>Abstracts</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00 - 9:45</td>
<td>Probabilistic Dimensionality Reduction</td>
<td>Neil Lawrence, University of Sheffield and Amazon Research Cambridge</td>
<td>11th Floor Lecture Hall</td>
<td>Bayesian Calibration of Simulators with Structured Discretization Uncertainty</td>
</tr>
<tr>
<td>10:30 - 11:15</td>
<td>Bayesian optimization for automating model selection</td>
<td>Roman Garnett, Washington University in St. Louis</td>
<td>11th Floor Lecture Hall</td>
<td>Numerical Gaussian Processes for Time-dependent and Non-Linear Partial Differential Equations</td>
</tr>
<tr>
<td>11:30 - 12:15</td>
<td>Variational Reformulation of the Uncertainty Propagation Problem in Linear Partial Differential Equations</td>
<td>Ilias Bilionis, Purdue University</td>
<td>11th Floor Lecture Hall</td>
<td></td>
</tr>
<tr>
<td>10:00 - 10:30</td>
<td>Coffee Break</td>
<td></td>
<td>11th Floor Lecture Hall</td>
<td>Coffee/Tea Break</td>
</tr>
<tr>
<td>3:30 - 4:00</td>
<td>Coffee/Tea Break</td>
<td></td>
<td>11th Floor Lecture Hall</td>
<td>Coffee/Tea Break</td>
</tr>
<tr>
<td>4:00 - 4:45</td>
<td>Bayesian Probabilistic Numerical Methods. (Part II)</td>
<td>Jon Cockayne, University of Warwick</td>
<td>11th Floor Lecture Hall</td>
<td>Bayesian Probabilistic Numerical Methods. (Part II)</td>
</tr>
</tbody>
</table>

GPs will be mentioned in ~50% of the workshop talks!
**Data-driven modeling with Gaussian processes**

"The linear algebra of computation under uncertainty"

**Priors over functions:** \( f \sim \mathcal{GP}(\mu(x), K(x, x'; \theta)) \)

**Marginalization:**
\[
p(f_A, f_B) \sim \mathcal{N}(\mu, K).
\]
Then:
\[
p(f_A) = \int_{f_B} p(f_A, f_B)df_B = \mathcal{N}(\mu_A, K_{AA})
\]

**Posterior is also Gaussian:**
\[
p(f_A, f_B) \sim \mathcal{N}(\mu, K).
\]
Then:
\[
p(f_A|f_B) = \mathcal{N}((\mu_A + K_{AB}K_{BB}^{-1}(f_B - \mu_B), K_{AA} - K_{AB}K_{BB}^{-1}K_{BA})
\]

---

Rasmussen, C. E. Gaussian processes for machine learning (2006)
Data-driven modeling with Gaussian processes

\[ y = f(x) + \epsilon \quad f \sim \mathcal{GP}(0, k(x, x'; \theta)) \]

**Training via maximizing the marginal likelihood**

\[
\log p(y | X, \theta) = -\frac{1}{2} \log |K + \sigma^2 I| - \frac{1}{2} y^T (K + \sigma^2 I)^{-1} y - \frac{N}{2} \log 2\pi
\]

**Prediction via conditioning on available data**

\[
p(f_\ast | y, X, x_\ast) = \mathcal{N}(f_\ast | \mu_\ast, \sigma^2_\ast),
\]

\[
\mu_\ast(x_\ast) = k_\ast_N (K + \sigma^2 I)^{-1} y,
\]

\[
\sigma^2_\ast(x_\ast) = k_{\ast\ast} - k_{\ast N} (K + \sigma^2 I)^{-1} k_{N\ast},
\]

---

Rasmussen, C. E. Gaussian processes for machine learning (2006)
1.) Model specification: Choosing a prior
\[ f \sim \mathcal{GP}(0, k(x, x'; \theta)) \]

2.) Training the model: Inference algorithm
\[
\log p(y|X, \theta) = -\frac{1}{2} \log |K + \sigma^2 \epsilon I| - \frac{1}{2} y^T (K + \sigma^2 \epsilon I)^{-1} y - \frac{N}{2} \log 2\pi
\]

3.) Obtain predictions & quantify uncertainty
\[
p(f_*|y, X, x_*) = \mathcal{N}(f_*|\mu_*, \sigma_*^2),
\]
\[
\mu_*(x_*) = k_* N (K + \sigma^2 \epsilon I)^{-1} y,
\]
\[
\sigma_*^2(x_*) = k_{**} - k_* N (K + \sigma^2 \epsilon I)^{-1} k_{N*},
\]

4.) Data acquisition
Multi-fidelity modeling

Number of runs is limited by time and computational resources

We cannot compute at all \((x; \xi)\)

Prediction of \(Z_i(x) = \mathbb{E}[f(Y_i(x; \xi))]\) is a problem of statistical inference.
Multi-fidelity modeling

Number of runs is limited by time and computational resources

We cannot compute at all $\{x; \xi\}$

Prediction of $Z_i(x) = \mathbb{E}[f(Y_i(x; \xi))]$ is a problem of statistical inference
Multi-fidelity modeling

Number of runs is limited by time and computational resources

We cannot compute at all \((x; \xi)\)

Prediction of \(Z_i(x) = \mathbb{E}[f(Y_i(x; \xi))]\) is a problem of statistical inference

- **Exact high fidelity**
- **Single-fidelity GP prediction**
- **Two standard deviations band**
- **High-fidelity training data (4 points)**

- **Exact high fidelity**
- **Exact low fidelity**
- **Multi-fidelity GP prediction**
- **Two standard deviations band**
- **High-fidelity training data (4 points)**
- **Low-fidelity training data (7 points)**
Multi-fidelity modeling

Number of runs is limited by time and computational resources

We cannot compute at all \((x; \xi)\)

Prediction of \(Z_i(x) = E[f(Y_i(x; \xi))]\) is a problem of statistical inference
Predicting the Output from a Complex Computer Code When Fast Approximations Are Available

M. C. Kennedy; A. O’Hagan


Auto-regressive model: AR1

\[ f_t(x) = \rho_{t-1}(x)f_{t-1}(x) + \delta_t(x) \]

\[ t = 1, \ldots, s \]

Predictive posterior

\[ p(f_*|y, X, x_*) = \mathcal{N}(f_*|\mu_*, \sigma_*^2), \]

\[ \mu_*(x_*) = k_{*N}(K + \sigma_e^2 I)^{-1}y, \]

\[ \sigma_*^2(x_*) = k_{**} - k_{*N}(K + \sigma_e^2 I)^{-1}k_{N*}, \]

M.C Kennedy, and A. O’Hagan. Predicting the output from a complex computer code when fast approximations are available, 2000.
Bayesian optimization

**Goal:** Estimate the global minimum of a function: \( \mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathbb{R}^d} g(\mathbf{x}) \) (potentially intractable)

**Setup:** \( g(x) \) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

**Idea:** Approximate \( g(x) \) using a GP surrogate: \( y = f(x) + \epsilon, \quad f \sim \mathcal{GP}(f \mid 0, k_{x, x'}(\theta)) \)

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

\[
\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_{\theta} \mathbb{E}_{\nu} [ \mathbf{x}, \theta [U(\mathbf{x}, \nu, \theta)]
\]

The optimization problem is transformed to:

\[
\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)
\]

**Remark:**
Acquisition functions aim to balance the trade-off between exploration and exploitation.

*e.g. sample at the locations that maximize the expected improvement*

Bayesian optimization

**Goal:** Estimate the global minimum of a function: \( \mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^d} g(\mathbf{x}) \) (potentially intractable)

**Setup:** \( g(\mathbf{x}) \) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

**Idea:** Approximate \( g(\mathbf{x}) \) using a GP surrogate: \( y = f(\mathbf{x}) + \epsilon, \quad f \sim \mathcal{GP}(f \mid 0, k(\mathbf{x}, \mathbf{x'}; \theta)) \)

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

\[
\alpha(\mathbf{x}; D_n) = \mathbb{E}_{\theta} \mathbb{E}_{\nu} \mathbb{E}_{\mathbf{x}, \theta} [ U(\mathbf{x}, \nu, \theta) ] 
\]

The optimization problem is transformed to:

\[
\mathbf{x}_{n+1} = \arg \max_{\mathbf{x}} \alpha(\mathbf{x}; D_n)
\]

**Remark:** Acquisition functions aim to balance the trade-off between exploration and exploitation.

Bayesian optimization

**Goal:** Estimate the global minimum of a function: \( x^* = \arg \min_{x \in \mathbb{R}^d} g(x) \) (potentially intractable)

**Setup:** \( g(x) \) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

**Idea:** Approximate \( g(x) \) using a GP surrogate: \( y = f(x) + \epsilon, \ f \sim \mathcal{GP}(f|0, k(x, x'; \theta)) \)

![Bayesian Optimization Diagram](image)

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

\[
\alpha(x; D_n) = \mathbb{E}_\theta \mathbb{E}_v [x, \theta]\{U(x, v, \theta)\}
\]

The optimization problem is transformed to:

\[
x_{n+1} = \arg \max_x \alpha(x; D_n)
\]

**Remark:**

Acquisition functions aim to balance the trade-off between exploration and exploitation.

---

Bayesian optimization

**Goal:** Estimate the global minimum of a function:  
\[ x^* = \arg \min_{x \in \mathbb{R}^d} g(x) \]  
(potentially intractable)

**Setup:** \( g(x) \) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

**Idea:** Approximate \( g(x) \) using a GP surrogate:  
\[ y = f(x) + \epsilon, \quad f \sim \mathcal{GP}(f|0, k(x, x'; \theta)) \]

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

\[ \alpha(x; D_n) = \mathbb{E}_\theta [\mathbb{E}_{\mathbf{v}|\mathbf{x}, \theta} [U(x, \mathbf{v}, \theta)]] \]

The optimization problem is transformed to:

\[ x_{n+1} = \arg \max_x \alpha(x; D_n) \]

**Remark:**
Acquisition functions aim to balance the trade-off between exploration and exploitation.

**Bayesian optimization**

**Goal:** Estimate the global minimum of a function: \( \mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathbb{R}^d} g(\mathbf{x}) \) (potentially intractable)

**Setup:** \( g(x) \) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

**Idea:** Approximate \( g(x) \) using a GP surrogate: \( y = f(x) + \epsilon, \quad f \sim \mathcal{GP}(f|0, k(x, x'; \theta)) \)

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

\[
\alpha(\mathbf{x}; D_n) = \mathbb{E}_\theta \mathbb{E}_v | \mathbf{x}, \theta [U(\mathbf{x}, v, \theta)]
\]

The optimization problem is transformed to:

\[
\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; D_n)
\]

**Remark:**

Acquisition functions aim to balance the trade-off between exploration and exploitation.

---

**Goal:** Estimate the global minimum of a function: \( \mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathbb{R}^d} g(\mathbf{x}) \) (potentially intractable)

**Setup:** \( g(\mathbf{x}) \) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

**Idea:** Approximate \( g(\mathbf{x}) \) using a GP surrogate: \( y = f(\mathbf{x}) + \epsilon, \quad f \sim \mathcal{GP}(f|0, k(\mathbf{x}, \mathbf{x}'; \theta)) \)

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

\[
\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_\nu | \mathbf{x}, \theta [U(\mathbf{x}, \nu, \theta)]
\]

The optimization problem is transformed to:

\[
\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)
\]

**Remark:**
Acquisition functions aim to balance the trade-off between exploration and exploitation.

**Bayesian optimization**

**Goal:** Estimate the global minimum of a function: \[ x^* = \arg \min_{x \in \mathbb{R}^d} g(x) \] (potentially intractable)

**Setup:** \( g(x) \) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

**Idea:** Approximate \( g(x) \) using a GP surrogate: \[ y = f(x) + \epsilon, \quad f \sim \mathcal{GP}(f|0, k(x, x'; \theta)) \]

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

\[ \alpha(x; \mathcal{D}_n) = \mathbb{E}_{\theta} \mathbb{E}_{\epsilon}[x, \theta][U(x, \epsilon, \theta)] \]

The optimization problem is transformed to:

\[ x_{n+1} = \arg \max_x \alpha(x; \mathcal{D}_n) \]

**Remark:**
Acquisition functions aim to balance the trade-off between exploration and exploitation.

Some software packages

<table>
<thead>
<tr>
<th>Gaussian processes:</th>
<th>Bayesian optimization:</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://github.com/SheffieldML/GPy">https://github.com/SheffieldML/GPy</a></td>
<td><a href="https://github.com/HIPS/Spearmint">https://github.com/HIPS/Spearmint</a></td>
</tr>
<tr>
<td><a href="https://github.com/GPflow/GPflow">https://github.com/GPflow/GPflow</a></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Automatic differentiation:</th>
<th>Probabilistic programming:</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://github.com/HIPS/autograd">https://github.com/HIPS/autograd</a></td>
<td><a href="http://edwardlib.org">http://edwardlib.org</a></td>
</tr>
<tr>
<td><a href="https://www.tensorflow.org">https://www.tensorflow.org</a></td>
<td><a href="http://mc-stan.org">http://mc-stan.org</a></td>
</tr>
<tr>
<td><a href="http://pytorch.org">http://pytorch.org</a></td>
<td></td>
</tr>
</tbody>
</table>