Introduction to Gaussian Processes

Regression, Classification, Experimental Design and Bayesian Optimization.

Ruben Martinez-Cantin

Defense University Center
Zaragoza, Spain
rmcantin@unizar.es
Outline

What you will see here:
- Gaussian process hyperparameters
- Regression
- Binary classification
- Active learning and experimental design
- Submodularity
- Bayesian optimization
- Stochastic bandits
Outline

What you won’t see here:

- Multi-class classification (only binary)
- Full Bayesian inference
  - Only ML estimate of hyperparameters
- Active learning for GP hyperparameters
- Sparse Gaussian process
- Adversarial bandits or reinforcement learning with GPs
- ...

Ruben Martinez-Cantin
Introduction to Gaussian Processes
Gaussian processes

We have a function with noisy observations

\[ y = f(x) + \epsilon \quad \quad f(x) = \phi(x)^T w \]

\[ \epsilon \sim \mathcal{N}(0, \sigma_n^2) \quad \quad w \sim \mathcal{N}(0, \Sigma_p) \]

Remember: \( \phi(x) \) and \( w \) can be infinite dimensional.

Then

\[
p(f_*|x_*, x, y) = \int p(f_*|x_*, w)p(w|x, y) \, dw
\]

\[
= \int p(f_*|x_*, w) \frac{p(y|x, w)p(w)}{p(y|x)} \, dw
\]

Good news: Everything is linear-Gaussian!
After some linear algebra

Let us define

$$\phi_* = \phi(x_*) \quad \Phi = \phi(x)$$

Then, the predicted distribution is

$$\hat{f}_* | x_*, x, y = \phi_\ast^T \Sigma_p \Phi (\Phi^T \Sigma_p \Phi + \sigma_n^2 I)^{-1} y$$

$$\text{cov}(f_* | x_*, x, y) = \phi_\ast^T \Sigma_p \phi_\ast - \phi_\ast^T \Sigma_p \Phi (\Phi^T \Sigma_p \Phi + \sigma_n^2 I)^{-1} \Phi^T \Sigma_p \phi_\ast$$
Kernels come in

- Remember Bernard Schölkopf’s talk:
  \[
  k(x, x') = \langle \phi(x), \phi(x') \rangle
  \]

- Then we can write:
  \[
  \hat{f}_* | x, y = K(x_*, x)(K(x, x) + \sigma_n^2 I)^{-1} y
  \]
  \[
  \text{cov}(f_*, x, y) = K(x_*, x_*) - K(x_*, x)(K(x, x) + \sigma_n^2 I)^{-1} K(x, x_*)
  \]

- This can be rewritten as:
  \[
  \begin{bmatrix}
  y \\
  f_*
  \end{bmatrix}
  \sim \mathcal{N}
  \begin{pmatrix}
  0, & K(x, x) + \sigma_n^2 I & K(x, x_*) \\
  K(x_*, x) & K(x_*, x) \\
  \end{pmatrix}
  \]

- On top of this, you can add your favorite mean function.
Gaussian processes in action

- Distribution over functions
- Every subset of points follows a multi-variate Gaussian distribution
- Non-parametric:
  - Bad news: (Computational) Complexity increase with the number of data points.
  - Good news: (Model) Complexity increase with the number of data points.

Typically we plot the 95% of the predicted distribution.

\[ f_* \pm 2 \cdot \text{cov}(f_*) \]
Kernel/Covariance functions

- Squared exponential

\[ k(x, x') = \exp \left( -\frac{(x - x')^2}{2l^2} \right) \]

- Mattern-3

\[ k(x, x') = \left( 1 + \frac{\sqrt{3}|x - x'|}{l} \right) \exp \left( -\frac{\sqrt{3}|x - x'|}{l} \right) \]

- Linear

\[ k(x, x') = \sum_{d=1}^{D} \sigma_d^2 x_d x'_d \]

- and basically all the kernels from Bernard Schölkopf’s talk . . .
Hyperparameter learning

- We still depend on the hyperparameters of our model
  - Kernel: \( l, \sigma_d^2, \ldots \)
  - Likelihood: \( \sigma_n^2 \)
  - Mean function parameters.

- We can give then priors and compute the full posterior.

- However, in practice:
  - Set up by hand.
  - Maximum likelihood estimate, such as, conjugate gradient.