CS480/680
Lecture 12: June 17, 2019

Gaussian Processes

[B] Section 6.4 [M] Chap. 15 [HTF] Sec. 8.3

Gaussian Process Regression

- Idea: distribution over functions

Bayesian Linear Regression

- Setting: $f(\boldsymbol{x}) = \underset{\substack{\downarrow \\ \text{unknown}}}{\boldsymbol{w}^T} \phi(\boldsymbol{x})$ and $y = f(\boldsymbol{x}) + \underset{\substack{\downarrow \\ N(0,\sigma^2)}}{\epsilon}$

- Weight space view:
  - Prior: $\Pr(\boldsymbol{w})$
  - Posterior: $\underset{\substack{\downarrow \\ \text{Gaussian}}}{\Pr(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y})} = k \underset{\substack{\downarrow \\ \text{Gaussian}}}{\Pr(\boldsymbol{w})} \underset{\substack{\downarrow \\ \text{Gaussian}}}{\Pr(\boldsymbol{y}|\boldsymbol{w},\boldsymbol{X})}$

Bayesian Linear Regression

- Setting: $\underset{\substack{\big\downarrow \\ \text{unknown}}}{f}(\boldsymbol{x}) = \boldsymbol{w}^T\phi(\boldsymbol{x})$ and $y = f(\boldsymbol{x}) + \underset{\substack{\big\downarrow \\ N(0,\sigma^2)}}{\epsilon}$

- Function space view:
  - Prior: $\underset{\substack{\big\downarrow \\ \text{Gaussian}}}{\Pr\big(f(\boldsymbol{x}_*)\big)} = \int_{\boldsymbol{w}} \underset{\substack{\big\downarrow \\ \text{Deterministic}}}{\Pr(f|\boldsymbol{w},\boldsymbol{x}_*)}\, \underset{\substack{\big\downarrow \\ \text{Gaussian}}}{\Pr(\boldsymbol{w})}\, d\boldsymbol{w}$
  - Posterior: $\underset{\substack{\big\downarrow \\ \text{Gaussian}}}{\Pr(f(\boldsymbol{x}_*)|\boldsymbol{X},\boldsymbol{y})} = \int_{\boldsymbol{w}} \underset{\substack{\big\downarrow \\ \text{Deterministic}}}{\Pr(f|\boldsymbol{w},\boldsymbol{x}_*)}\, \underset{\substack{\big\downarrow \\ \text{Gaussian}}}{\Pr(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y})}\, d\boldsymbol{w}$

Gaussian Process

- According to the function view, there is a Gaussian at $f(\boldsymbol{x}_*)$ for every $\boldsymbol{x}_*$. Those Gaussians are correlated through w.

- What is the general form of $\Pr(f)$ (i.e., distribution over functions)?

- Answer: **Gaussian Process** (infinite dimensional Gaussian distribution)

Gaussian Process

- Distribution over functions:

$$f(\boldsymbol{x}) \sim GP\big(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')\big) \ \forall \boldsymbol{x}, \boldsymbol{x}'$$

- Where $m(\boldsymbol{x}) = E(f(\boldsymbol{x}))$ is the mean and $k(\boldsymbol{x}, \boldsymbol{x}') = E((f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x}') - m(\boldsymbol{x}'))$ is the kernel covariance function

Mean function $m(\boldsymbol{x})$

- Compute the mean function $m(\boldsymbol{x})$ as follows:

- Let $f(\boldsymbol{x}) = \phi(\boldsymbol{x})^T \boldsymbol{w}$
  with $\boldsymbol{w} \sim N(\boldsymbol{0}, \alpha^{-1}\boldsymbol{I})$

- Then $$\begin{aligned} m(\boldsymbol{x}) &= E(f(\boldsymbol{x})) \\ &= E(\boldsymbol{w})^T \phi(\boldsymbol{x}) \\ &= \boldsymbol{0} \end{aligned}$$

Kernel covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$

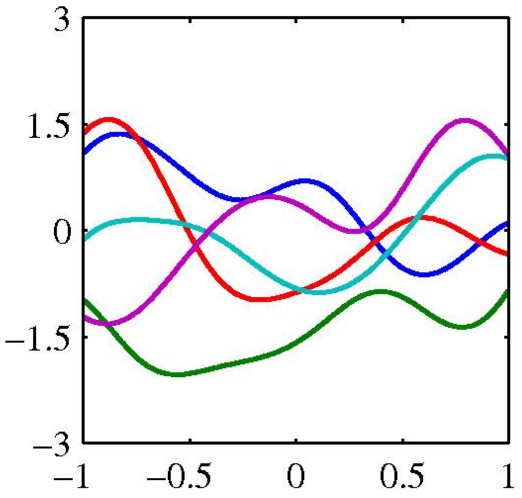
- Compute kernel covariance $k(\boldsymbol{x}, \boldsymbol{x}')$ as follows:

- $$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{x}') &= E(f(\boldsymbol{x})f(\boldsymbol{x}')) \\ &= \phi(\boldsymbol{x})^T E\left(\boldsymbol{w}\boldsymbol{w}^{\boldsymbol{T}}\right)\phi(\boldsymbol{x}') \\ &= \phi(\boldsymbol{x})^T \frac{I}{\alpha} \phi(\boldsymbol{x}') \\ &= \frac{\phi(\boldsymbol{x})^T \phi\left(\boldsymbol{x}'\right)}{\alpha} \end{aligned}$$

- In some cases we can use domain knowledge to specify k directly.

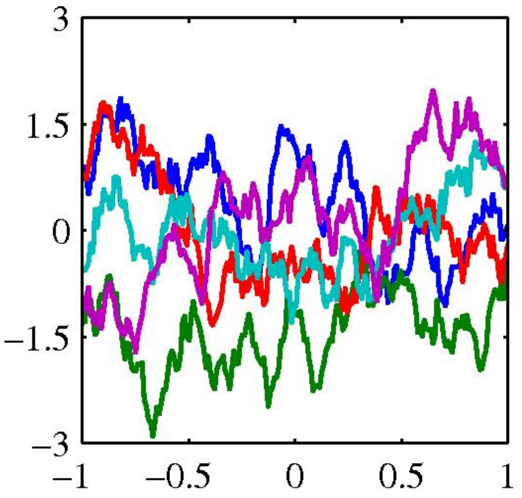
Examples

- Sampled functions from a Gaussian Process



Gaussian kernel

$$k(\boldsymbol{x},\boldsymbol{x}') = e^{-\frac{\left|\left|\boldsymbol{x}-\boldsymbol{x}'\right|\right|^2}{2\sigma^2}}$$



Exponential kernel
(Brownian motion)

$$k(\boldsymbol{x},\boldsymbol{x}') = e^{-\theta|\boldsymbol{x}-\boldsymbol{x}'|}$$

Gaussian Process Regression

- Gaussian Process Regression corresponds to kernelized Bayesian Linear Regression
- Bayesian Linear Regression:
  - Weight space view
  - Goal: $\Pr(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y})$ (posterior over $\boldsymbol{w}$)
  - Complexity: cubic in # of basis functions
- Gaussian Process Regression:
  - Function space view
  - Goal: $\Pr(f|\boldsymbol{X},\boldsymbol{y})$ (posterior over f)
  - Complexity: cubic in # of training points

Recap: Bayesian Linear Regression

- Prior: $\Pr(\boldsymbol{w}) = N(\boldsymbol{0}, \boldsymbol{\Sigma})$
- Likelihood: $\Pr(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) = N\left(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{\Phi}, \sigma^2\boldsymbol{I}\right)$

- Posterior: $\Pr(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}) = N\left(\overline{\boldsymbol{w}}, \boldsymbol{A}^{-\boldsymbol{1}}\right)$

  where $\overline{\boldsymbol{w}} = \sigma^{-2}\boldsymbol{A}^{-\boldsymbol{1}}\boldsymbol{\Phi}\boldsymbol{y}$ and $\boldsymbol{A} = \sigma^{-2}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\boldsymbol{T}} + \boldsymbol{\Sigma}^{-1}$
- Prediction:

  $\Pr(y_*|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) = N(\sigma^{-2}\phi(\boldsymbol{x}_*)^T\,\boldsymbol{A}^{-1}\boldsymbol{\Phi}\boldsymbol{y}, \sigma^2 + \phi(\boldsymbol{x}_*)^T\boldsymbol{A}^{-1}\phi(\boldsymbol{x}_*))$

- Complexity: inversion of $\boldsymbol{A}$ is cubic in # of basis functions

Gaussian Process Regression

- Prior: $\Pr(f(\cdot)) = N(m(\cdot), k(\cdot,\cdot))$
- Likelihood: $\Pr(\boldsymbol{y}|\boldsymbol{X}, f) = N(f(\boldsymbol{X}), \sigma^2\boldsymbol{I})$
- Posterior: $\Pr(f(\cdot)|\boldsymbol{X}, \boldsymbol{y}) = N\left(\bar{f}(\cdot), k'(\cdot,\cdot)\right)$

  where $\bar{f}(\cdot) = k(\cdot, \boldsymbol{X})(\boldsymbol{K} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}$ and

  $k'(\cdot,\cdot) = k(\cdot,\cdot) + \sigma^2\boldsymbol{I} - k(\cdot, \boldsymbol{X})(\boldsymbol{K} + \sigma^2\boldsymbol{I})^{-1}k(\boldsymbol{X},\cdot)$
- Prediction: $\Pr(y_*|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) = N\left(\bar{f}(\boldsymbol{x}_*), k'(\boldsymbol{x}_*, \boldsymbol{x}_*)\right)$

- Complexity: inversion of $\boldsymbol{K} + \sigma^2\boldsymbol{I}$ is cubic in # of training points

Infinite Neural Networks

- Recall: neural networks with a single hidden layer (that contains sufficiently many hidden units) can approximate any function arbitrarily closely

- Neal 94: The limit of an infinite single hidden layer neural network is a Gaussian Process

Bayesian Neural Networks

- Consider a neural network with J hidden units and a single identity output unit y_k:

$$y_k = f(\boldsymbol{x};\boldsymbol{w}) = \sum_{j=1}^{J} w_{kj} h\left(\sum_i w_{ji} x_i + w_{j0}\right) + w_{k0}$$

- Bayesian learning: express prior over the weights
  - Weight space view:

$$\Pr\left(w_{kj}\right) \text{ where } E\left(w_{kj}\right) = 0, Var\left(w_{kj}\right) = \frac{\alpha}{J} \quad \forall j,$$
$$\Pr(w_{k0}) \text{ where } E(w_{k0}) = 0, Var(w_{k0}) = \sigma^2 \forall ji$$
$$\text{Type equation here.}$$

  - Function space view: when $J \rightarrow \infty$, by the central limit theorem, an infinite sum of i.i.d. (identically and independently distributed) variables yields a Gaussian

$$\Pr\left(f(\boldsymbol{x})\right) = N(f(\boldsymbol{x})|0, \alpha E[h(\boldsymbol{x})h(\boldsymbol{x}')] + \sigma^2)$$

Mean Derivation

- Calculation of the mean function:
- $E[f(\boldsymbol{x})] = \sum_{j=1}^{J} E[w_{kj}h(\boldsymbol{x})] + E[w_{k0}]$

  $= \sum_{j=1}^{J} E\left[w_{kj}\right]E[h(\boldsymbol{x})] + E[w_{k0}]$

  $= \sum_{j=1}^{J} 0\ E[h(\boldsymbol{x})] + 0$

  $= 0$

Covariance Derivation

- $Cov[f(\boldsymbol{x}), f(\boldsymbol{x}')]$

$$= E[f(\boldsymbol{x})f(\boldsymbol{x}')] - E[f(\boldsymbol{x})]E[f(\boldsymbol{x}')]$$

$$= E[f(\boldsymbol{x})f(\boldsymbol{x}')]$$

$$= E\left[\left(\sum_j w_{kj}h_j(\boldsymbol{x}) + w_{k0}\right)\left(\sum_j w_{kj}h_j(\boldsymbol{x}') + w_{k0}\right)\right]$$

$$= \sum_{j=1}^{J} E\left[w_{kj}h_j(\boldsymbol{x})w_{kj}h_j(\boldsymbol{x}')\right] + E[w_{k0}w_{k0}]$$

$$= \sum_{j=1}^{J} E\left[w_{kj}^2\right]E\left[h_j(\boldsymbol{x})h_j(\boldsymbol{x}')\right] + E[w_{k0}^2]$$

$$= \sum_{j=1}^{J} Var\left[w_{kj}\right]E[h(\boldsymbol{x})h(\boldsymbol{x}')] + Var(w_{k0})$$

$$= \sum_{j=1}^{J} \frac{\alpha}{J} E[h(\boldsymbol{x})h(\boldsymbol{x}')] + \sigma^2$$

$$= \alpha E[h(\boldsymbol{x})h(\boldsymbol{x}')] + \sigma^2$$

Bayesian Neural Networks

- When # of hidden units $J \to \infty$, then Bayesian neural net is equivalent to a Gaussian Process $\Pr\big(f(\cdot)\big) = GP(f(\cdot)|0, \alpha E[h(\cdot)h(\cdot)] + \sigma^2)$

- Note: this works for
  - Any activation function h
  - Any i.i.d. prior over the weights with mean 0

Case Study: AIBO Gait Optimization

Gait Optimization

- Problem: find best parameter setting of the gait controller to maximize walking speed
  - Why?: Fast robots have a better chance of winning in robotic soccer

- Solutions:
  - Stochastic hill climbing
  - **Gaussian Processes**
    - Lizotte, Wang, Bowling, Schuurmans (2007) Automatic Gait Optimization with Gaussian Processes, *International Joint Conferences on Artificial Intelligence (IJCAI)*.

Search Problem

- Let $\boldsymbol{x} \in \Re^{15}$, be a vector of 15 parameters that defines a controller for gait
- Let $f: \boldsymbol{x} \rightarrow \Re$ be a mapping from controller parameters to gait speed

- Problem: find parameters $\boldsymbol{x}^*$ that yield highest speed.

$$\boldsymbol{x}^* \leftarrow argmax_{\boldsymbol{x}}\, f(\boldsymbol{x})$$

But f is unknown…

Approach

- Picture

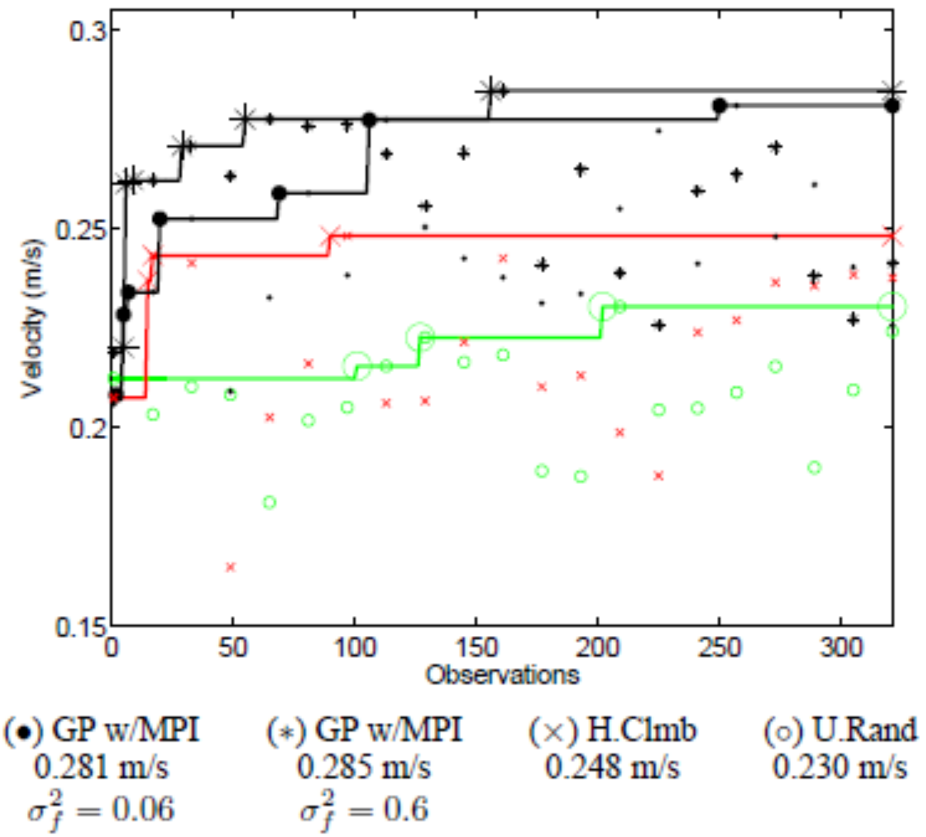
Approach

- Initialize $f(\cdot) \sim GP(m(\cdot), k(\cdot,\cdot))$
- Repeat:
  - Select new $\boldsymbol{x}$:

    $$\boldsymbol{x}_{new} \leftarrow argmax_{\boldsymbol{x}} \frac{k(\boldsymbol{x},\boldsymbol{x})}{\max_{\boldsymbol{x'} \in X} f(\boldsymbol{x'}) - m(\boldsymbol{x})}$$

  - Evaluate $f(\boldsymbol{x_{new}})$ by observing speed of robot with parameters set to $\boldsymbol{x}_{new}$
  - Update Gaussian process:
    - $\boldsymbol{X} \leftarrow \boldsymbol{X} \cup \{\boldsymbol{x_{new}}\}$ and $\boldsymbol{y} \leftarrow \boldsymbol{y} \cup f(\boldsymbol{x_{new}})$
    - $m(\cdot) \leftarrow k(\cdot, \boldsymbol{X})(\boldsymbol{K} + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{y}$
    - $k(\cdot,\cdot) \leftarrow k(\cdot,\cdot) + \sigma^2 \boldsymbol{I} - k(\cdot, \boldsymbol{X})(\boldsymbol{K} + \sigma^2 \boldsymbol{I})^{-1} k(\boldsymbol{X},\cdot)$

Results



Gaussian kernel:

$$k(\boldsymbol{x},\boldsymbol{x}') = \sigma_f^2 e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}')^T S(\boldsymbol{x}-\boldsymbol{x}')}$$