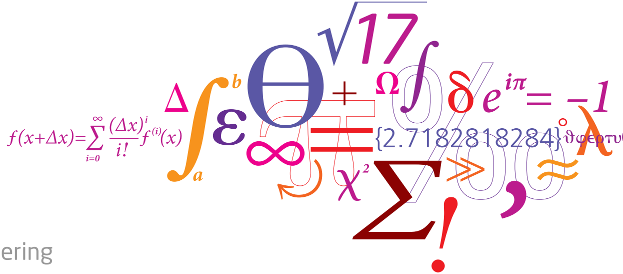
Gaussian processes

Filipe Rodrigues

Francisco Pereira



DTU Management Engineering
Department of Management Engineering

Outline

- Introduction
- Gaussian processes

Regression

- Consider models of the inputs $\mathbf{x} \in \mathbb{R}^D$ for continuous response variables $y \in \mathbb{R}$ of the form

$$y = f(\mathbf{x}) + \epsilon\,, \qquad \epsilon \sim \mathcal{N}(\epsilon|0, \sigma^2)$$

- Previously, we assumed f to be a **linear parametric** function of the inputs $\mathbf{x}$

$$f(\mathbf{x}) = \mathbf{w}^\mathsf{T}\mathbf{x}$$

- $\mathbf{w}$ was a D-dimensional vector of parameters (one weight per input dimension)
- We could therefore write the likelihood for a dataset $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^N$ as

$$p(\mathbf{y}|\mathbf{w}, \mathbf{X}) = \prod_{n=1}^{N} \mathcal{N}(y_n|\mathbf{w}^\mathsf{T}\mathbf{x}_n, \sigma^2)$$

where $\mathbf{y} = \{y_1, \ldots, y_N\}$ and $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$

Frequentist vs Bayesian approach

- Likelihood given by

$$p(\mathbf{y}|\mathbf{w}, \mathbf{X}) = \prod_{n=1}^{N} \mathcal{N}(y_n|\mathbf{w}^\mathsf{T}\mathbf{x}_n, \sigma^2)$$

- In a **frequentist approach**, we find the parameters **w** that maximize the (log) likelihood

$$\hat{\mathbf{w}}_{\mathsf{ML}} = \arg\max_{\mathbf{w}} \left(\sum_{n=1}^{N} \log p(y_n|\mathbf{w}, \mathbf{x}_n) \right)$$

- This is called **maximum likelihood (ML)** estimation
- We can make predictions for new test inputs $\mathbf{x}_*$ by plugging in the estimate $\hat{\mathbf{w}}_{\mathsf{ML}}$

$$p(y_*|\hat{\mathbf{w}}_{\mathsf{ML}}, \mathbf{x}_*)$$

- Point prediction given by

$$\hat{y}_* = (\hat{\mathbf{w}}_{\mathsf{ML}})^\mathsf{T}\mathbf{x}_*$$

Frequentist vs Bayesian approach

- Likelihood given by

$$p(\mathbf{y}|\mathbf{w},\mathbf{X}) = \prod_{n=1}^{N} \mathcal{N}(y_n|\mathbf{w}^\mathsf{T}\mathbf{x}_n, \sigma^2)$$

- We can further consider a prior on **w**: $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda\mathbf{I})$

$$\hat{\mathbf{w}}_{\text{MAP}} = \arg\max_{\mathbf{w}} \left(\sum_{n=1}^{N} \log p(y_n|\mathbf{w},\mathbf{x}_n) + \log p(\mathbf{w}) \right)$$

- This is called **maximum-a-posteriori (MAP)** estimation
- Term $\log p(\mathbf{w})$ acts as a penalty term - **regularization**
- As before, we make predictions by plugging in the estimate $\hat{\mathbf{w}}_{\text{MAP}}$

$$p(y_*|\hat{\mathbf{w}}_{\text{MAP}},\mathbf{x}_*)$$

- Point prediction given by

$$\hat{y}_* = (\hat{\mathbf{w}}_{\text{MAP}})^\mathsf{T}\mathbf{x}_*$$

Frequentist vs Bayesian approach

- Likelihood given by

$$p(\mathbf{y}|\mathbf{w}, \mathbf{X}) = \prod_{n=1}^{N} \mathcal{N}(y_n|\mathbf{w}^\mathsf{T}\mathbf{x}_n, \sigma^2)$$

- Prior given by $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \lambda\mathbf{I})$
- In a **Bayesian approach**, we treat **w** as a latent variable and do inference on it

$$p(\mathbf{w}|\mathbf{y}, \mathbf{X}) = \frac{p(\mathbf{y}|\mathbf{w}, \mathbf{X})\, p(\mathbf{w})}{p(\mathbf{y}|\mathbf{X})}$$

- We obtain a **full posterior distribution** on **w** rather than a point estimate!
- We can make predictions for new test input $\mathbf{x}_*$ by **averaging** over the values of **w**

$$p(y_*|\mathbf{x}_*, \mathbf{y}, \mathbf{X}) = \int p(y_*|\mathbf{w}, \mathbf{x}_*)\, p(\mathbf{w}|\mathbf{y}, \mathbf{X})\, d\mathbf{w}$$

- Marginal likelihood given by

$$p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|\mathbf{w}, \mathbf{X})\, p(\mathbf{w})\, d\mathbf{w}$$

Weight-space vs. function-space view

- Consider a dataset of target variables $\mathbf{y} = \{y_1, \ldots, y_N\}$ and their corresponding inputs $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$
- In Bayesian linear regression, we assumed that

$$y = f(\mathbf{x}) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

  where $f(\mathbf{x}) = \mathbf{w}^\mathsf{T}\mathbf{x}$

- We placed a prior on the weights $p(\mathbf{w})$ and performed inference to compute its posterior distribution $p(\mathbf{w}|\mathbf{y}, \mathbf{X})$
- We can consider this in vector-form

$$\mathbf{y} = \mathbf{f} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$$

  where $\mathbf{f} = f(\mathbf{X}) = \mathbf{w}^\mathsf{T}\mathbf{X}$

- Can we avoid $\mathbf{w}$ altogether and model $p(\mathbf{f})$ directly?
- Instead of working with weights $\mathbf{w}$, can we work with the functions $f(\mathbf{x})$?
  I.e. put a prior on $\mathbf{f}$ and perform inference on it?

Playtime!

- Jupyter notebook: “13 - Gaussian processes.ipynb"
- Part 1: From multivariate Gaussians to Gaussian processes

From multivariate Gaussians to Gaussian processes

- **Definition**: a Gaussian process (GP) is a collection of random variables, any finite number of which have (consistent) joint Gaussian distributions
- Consider a model of the form

$$y = f(\mathbf{x}) + \epsilon$$

- Now consider a multivariate (joint) Gaussian distribution over the N-dimensional vector $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))^\mathsf{T}$

$$\mathbf{f} \sim \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- A multivariate Gaussian distribution is fully specified by a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$
- A GP is a **stochastic process** fully specified by a mean function $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and a positive definite covariance function $k(\mathbf{x}, \mathbf{x}') = \mathrm{cov}[f(\mathbf{x}), f(\mathbf{x}')]$
- Therefore, a GP is a **generalization** of a multivariate Gaussian distribution to infinitely many variables

Gaussian processes

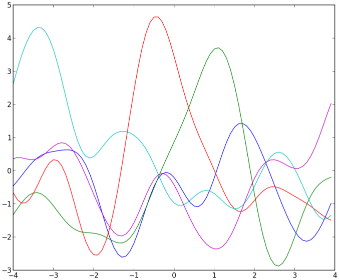
- A GP is a **stochastic process** fully specified by a mean function $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and a positive definite covariance function $k(\mathbf{x}, \mathbf{x}') = \text{cov}[f(\mathbf{x}), f(\mathbf{x}')]$
- Mean function $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ determines the mean of any arbitrary point $\mathbf{x}$ in the input space
    - Commonly assumed to be a zero-value vector, i.e. $m(\mathbf{x}) = 0$
- Covariance function $k(\mathbf{x}, \mathbf{x}') = \text{cov}[f(\mathbf{x}), f(\mathbf{x}')]$ determines how any two points in the input space covary (often called *kernels*)
    - Specifies basic aspects of the process such as smoothness, periodicity, stationarity and isotropy
- If we loosely see a function as a infinitely long vector $\mathbf{f}$, then we can think of a GP as a **probability distribution over functions**!
- **Main idea**: we place a GP prior over the function values $\mathbf{f}$; together with some likelihood function, we compute the GP posterior (we will return to this later...)
- GPs are **Bayesian non-parametric** models!

Covariance functions

- Most common choice is the **squared exponential (SE)**

$$k_{\mathsf{SE}}(\mathbf{x}, \mathbf{x}') = \exp\left(-\sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{2l^2}\right)$$

- Also called Gaussian kernel, RBF kernel, exponentiated quadratic, etc.
- Parameter l defining the characteristic length-scale
- Goes to unity as $\mathbf{x}$ becomes closer to $\mathbf{x}'$
- Nearby points are more likely to covary!
- GP prior with a SE covariance function prefers **smooth functions**



Covariance functions

- Other popular covariance functions:
  - Periodic (PER) covariance function

    $$k_{\mathsf{PER}}(\mathbf{x}, \mathbf{x}') = h^2 \exp\left(- \frac{1}{2\ell^2} \sin^2 \left(\frac{\pi}{p} \sum_{d=1}^{D} (x_d - x'_d) \right) \right)$$

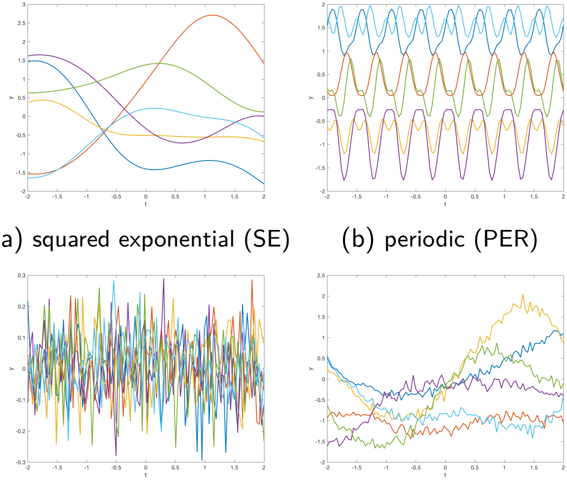
    where h controls the amplitude and p is the period
  - White noise (WN) covariance function (with variance σ^2)

    $$k_{\mathsf{WN}}(\mathbf{x}, \mathbf{x}') = \sigma^2 \, \delta(\mathbf{x}, \mathbf{x}')$$

    where $\delta(\mathbf{x}, \mathbf{x}')$ is the Kronecker delta function (1 when $\mathbf{x} = \mathbf{x}'$, 0 otherwise)

- Sums and products of proper covariance function are also valid covariance functions!

Covariance functions



(a) squared exponential (SE)

(b) periodic (PER)

(c) white noise (WN)

(d) SE+WN

Figure: Samples from Gaussian processes with different covariance functions.

Constructing a GP

- Given a dataset $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^N$
- Define GP prior for function values **f**: $\mathbf{f} \sim \mathcal{GP}(m(\mathbf{x}) = 0, k(\mathbf{x}, \mathbf{x}'))$
- Build covariance matrix **K**, where $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$
- Specifies a multivariate Gaussian distribution on **f**

$$\mathbf{f} \sim \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})$$

- This is our prior distribution over **f**
- We can use it to sample from the GP prior!

Playtime!

- Jupyter notebook: “13 - Gaussian processes.ipynb"
- Part 2: Sampling from a GP with different covariance functions

Inference with a GP

- Given a dataset $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$
- Define GP prior for function values $\mathbf{f}$: $\mathbf{f} \sim \mathcal{GP}(m(\mathbf{x}) = 0, k(\mathbf{x}, \mathbf{x}'))$
- Build covariance matrix $\mathbf{K}$, where $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$
- Specifies a multivariate Gaussian distribution on $\mathbf{f}$

$$\mathbf{f} \sim \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})$$

- This is our prior distribution over $\mathbf{f}$, $p(\mathbf{f})$, but we **need a likelihood too**!
- For continuous outputs $y \in \mathbb{R}$, an obvious choice is a Gaussian likelihood:

$$p(y_n|f_n) = \mathcal{N}(y_n|f_n, \sigma^2)$$

- Using Bayes rule, we can compute the posterior over $\mathbf{f}$ (exact inference)

$$p(\mathbf{f}|\mathbf{y}, \mathbf{X}) = \frac{p(\mathbf{y}|\mathbf{f})\, p(\mathbf{f})}{p(\mathbf{y}|\mathbf{X})}$$

- Compare this equation with the posterior for $\mathbf{w}$ in Bayesian linear regression

Marginal likelihood

- Making use of marginalization property for Gaussian distributions (see slide 4 of lecture 10), the marginal distribution of **y** is given by

$$p(\mathbf{y}|\mathbf{X}) = \int \underbrace{p(\mathbf{y}|\mathbf{f})}_{\text{likelihood}} \, \underbrace{p(\mathbf{f}|\mathbf{X})}_{\text{GP prior}} \, d\mathbf{f}$$

$$= \int \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I}) \, \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}) \, d\mathbf{f}$$

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

- We can use p(**y**|**X**) to optimize the parameters $\boldsymbol{\theta}$ of the covariance function!

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \left(\log \mathcal{N}(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}) \right)$$

Note

This another example of maximum marginal likelihood (also called type-II maximum likelihood, or empirical Bayes) from lecture 12.

Making predictions

- Our aim is to make a prediction y_* for a new input $\mathbf{x}_*$
- The joint distribution over $y_*, y_1, ..., y_N$ is simply given by

$$p(y_*, \mathbf{y}|\mathbf{x}_*, \mathbf{X}) = \mathcal{N}(y_*, \mathbf{y}|\mathbf{0}, \mathbf{V})$$

  with

$$\mathbf{V} = \begin{pmatrix} \sigma^2\mathbf{I} + \mathbf{K} & \mathbf{k}_* \\ \mathbf{k}_*^\mathsf{T} & \sigma^2 + k_{**} \end{pmatrix}$$

  where $\mathbf{k}_* = k(\mathbf{x}, x_*)$ and $k_{**} = k(x_*, x_*)$

- From the joint distribution, we can now determine the distribution of y_*, i.e. the predictive distribution:

$$p(y_*|\mathbf{y}, \mathbf{x}_*, \mathbf{X}) = \mathcal{N}(y_*|\mathbf{k}_*^\mathsf{T}(\sigma^2\mathbf{I} + \mathbf{K})^{-1}\mathbf{y},\, k_{**} + \sigma^2 - \mathbf{k}_*^\mathsf{T}(\sigma^2\mathbf{I} + \mathbf{K})^{-1}\mathbf{k}_*)$$

- Note: this is a direct application of the conditional probability for Gaussians (see slide 5 from lecture 10)

Playtime!

- Jupyter notebook: “13 - Gaussian processes.ipynb"
- Part 3: Inference and maximum marginal likelihood optimization

GP classification

- Given a dataset $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^N$
- What if y_n is discrete?
- Define GP prior for function values **f**: $\mathbf{f} \sim \mathcal{GP}(m(\mathbf{x}) = 0, k(\mathbf{x}, \mathbf{x}'))$, such that

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})$$

- For binary outputs $y_n \in \{0, 1\}$, a possible choice is the Probit function $\Phi(f_n)$:

$$p(y_n|f_n) = \Phi(f_n) = \int_{-\infty}^{f_n} \mathcal{N}(u|0,1)\, du$$

- Exact inference is no longer tractable; must resort to approximate methods

$$p(\mathbf{f}|\mathbf{y}, \mathbf{X}) = \frac{\left(\prod_{n=1}^N \Phi(f_n)\right) p(\mathbf{f})}{p(\mathbf{y}|\mathbf{X})}$$

- Similar approaches can be used to handle other types of outputs
  - Real, binary, categorical, positive real, positive integer or ordinal responses

Learning more about GPs

- A Visual Exploration of Gaussian Processes - **This notebook is a must!**
  `https://distill.pub/2019/visual-exploration-gaussian-processes/`
- Videolecture: Gaussian Processes, C. Rasmussen.
  `http://videolectures.net/mlss09uk_rasmussen_gp/`
- Book: Gaussian Processes for Machine Learning, C. Rasmussen and C. Williams. **Free!** `http://www.gaussianprocess.org/gpml/`
- Book: Pattern Recognition and Machine Learning, C. Bishop.