Introduction to Gaussian Processes- Regression DSA2019 Addis Ababa

Charles I. Saidu [1], Michael Mayhew [2]

African University of Science and Technology, Nigeria and Baze University, Nigeria [1]
Inflammatrix, USA [2]

isaidu at aust dot edu dot ng [1]
mmayhew at inflammatix dot com [2]

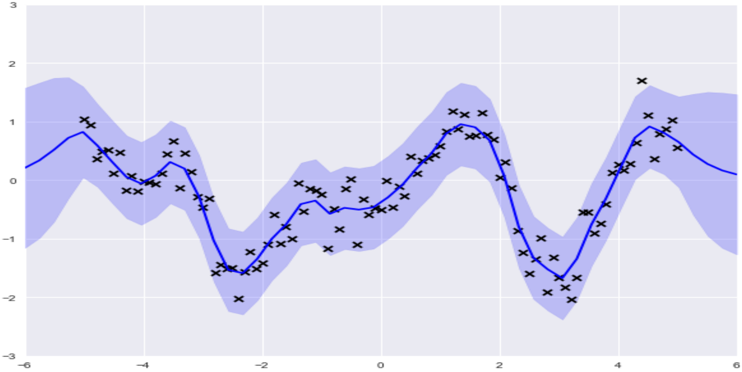
June 5, 2019

Introduction Gaussian Processes(GP)- Regression

1. Data Modelling (Pre-introduction to Gaussian Processes-regression):Regression problem
2. Parametric Models : Motivation
3. Non-Parametric Models, and the Gaussian Process
4. Gaussian Processes
5. Additional Resources

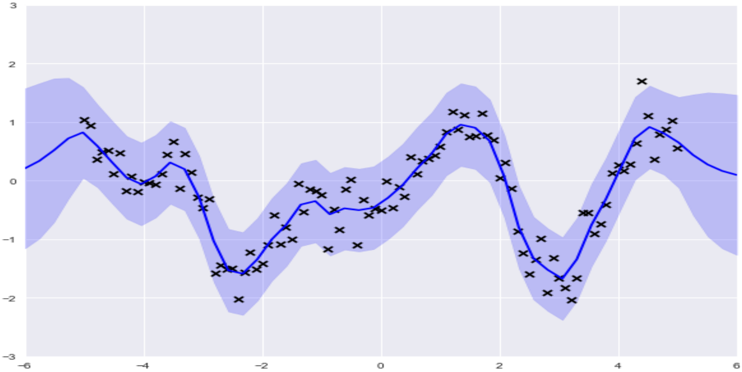
Data Modelling: Regression

- Let say we have data $\mathbf{D} = \{X, y\}$
- We are interested in finding the function $\mathbf{f}$, such that $y = f(x) + e$ describes the behaviour of data $\mathbf{D}$ with some error bars $\mathbf{e}$



Data Modelling: Regression

- Possible approaches towards finding **f** will be:
  - Parametric Approach: Neural Networks family,
  - Non-Parametric Approach - KNN, SVMs, **Gaussian Processes**

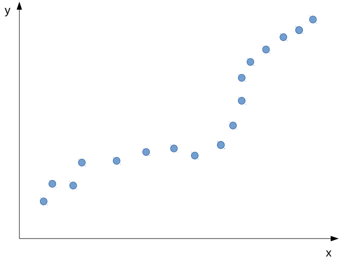


Data Modelling: Regression - Parametric Models

- We have Data $\mathbf{D} = \{X, y\}$
- Parametric Approach:
  - Choose a function class $f(x)$ or a mapping - With **FIXED NUMBER** of parameters Θ
  - Learn the **PARAMETERS** Θ^* of the model $f(x)$.
- Parameter Estimations
  - **Maximum Likelihood Estimation (MLE)**: Find the optimal value Θ^*
  - **Maximum A-Posterior Distribution (MAP)** $\rightarrow$ learn a point estimate (**Mode** of posterior distribution) of the FIXED Θ^*, using a prior over Θ^*. **Roburst to overfitting**
  - **Full Bayesian Methods**: Learn **plausable/feasible distribution over parameters** Θ^*. Using approximation methods: Variational Methods, MCMC, Laplace

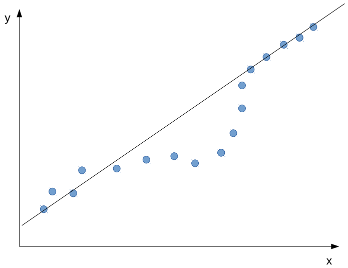
Data Modelling: Graphical Intuition - Motivation: Parametric models (Point Estimates vs Distribution)

- Consider the data scatter plot below
- How would you fit this model $f(x)$?



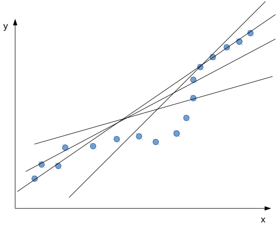
Data Modelling: Graphical Intuition - Motivation: Linear Parametric models - Parametric models (Point Estimates vs Distribution)

- Linear Model $f(x) = \theta_1 x + \theta_2$ - fitted using MLE or MAP
- Optimal parameters $\Theta^* = \{\theta_1, \theta_2\}$
- **NOTE**: Fixed Number of Parameters (2-parameters in this example)



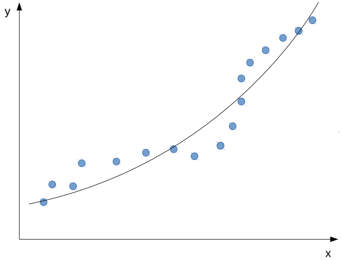
Data Modelling: Graphical Intuition - Motivation: Linear Parametric models

- Linear Model $f(x) = \theta_1 x + \theta_2$
- **Full Bayesian** - Distribution of the parameters Θ^*,
- Optimal parameters $\Theta^* = \theta_1, \theta_2$
- NOTE: **Still Fixed Number** of Parameters (2-parameters in this example)



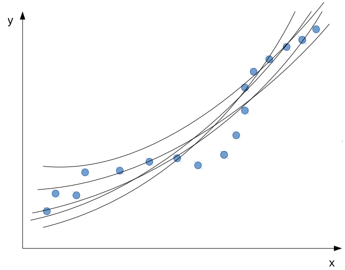
Data Modelling: Graphical Intuition - Motivation: Quadratic Parametric models - MLE

- Quadratic Model $f(x) = \theta_1 x^2 + \theta_2 x + \theta_3$ - fitted using MLE or MAP
- Optimal parameters $\Theta^* = \{\theta_1, \theta_2, \theta_3\}$ NOTE: Fixed Number of Parameters (3-parameters in this example)



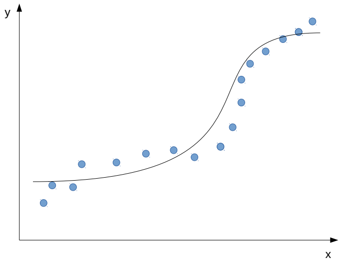
Data Modelling: Graphical Intuition - Motivation: Quadratic Parametric models - MAP

- Quadratic Model $f(x) = \theta_1 x^2 + \theta_2 x + \theta_3$
- **Full Bayesian** - Distribution of the parameters Θ^*
- Optimal parameters $\Theta^* = \{\theta_1, \theta_2, \theta_3\}$
- **NOTE**: **Still Fixed Number** of Parameters (3-parameters in this example)



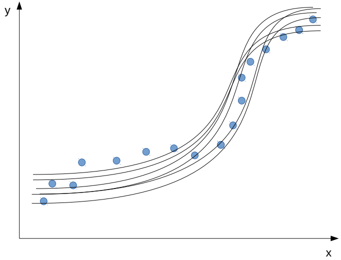
Data Modelling: Graphical Intuition - Motivation: Cubic Parametric models - MLE

- Quadratic Model $f(x) = \theta_1 x^3 + \theta_2 x^2 + \theta_3 x + \theta_4$ - fitted using MLE or MAP
- Optimal parameters $\Theta^* = \{\theta_1, \theta_2, \theta_3, \theta_4\}$
- **NOTE**: Fixed Number of Parameters (4-parameters in this example)



Data Modelling: Graphical Intuition - Motivation: Cubic Parametric models - MAP

- Quadratic Model $f(x) = \theta_1 x^3 + \theta_1 x^2 + \theta_2 x + \theta_3$
- **Full Bayesian**. Distribution of the parameters Θ^*
- Optimal parameters $\Theta^* = \{\theta_1, \theta_2, \theta_3, \theta_4\}$
- NOTE: Fixed Number of Parameters (4-parameters in this example)



Data Modeling: PUNCH-LINE

- What if we dont want to specify the number of parameters upfront in our model?
- Also what if we want to consider a distribution over **plausable** functions that describe our data, such that these functions complexity/parameters scale with the data
- Also we might want our model to be able to handle missing **Missing** data: aka **Generative Model**
- How???

Non Parametric models

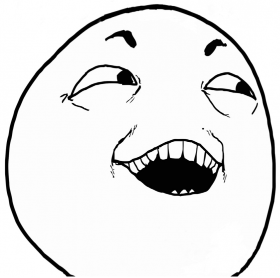
What is a **Non-parametric model**?

- No! It does **NOT** mean the model has no parameters
- Simply means the models's number of parameters is **NOT** fixed or determined upfront like in the previous examples - parametric models
- When you hear nonparametric, think models whose parameters scale with amount/complexity of data

Non Parametric models -GPs

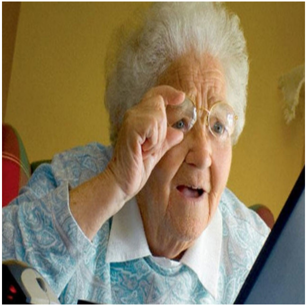
- So we want a model whose parameters scale with data/complexity
- We also want to model plausable functions $f(x)$ that describes our data
- Consequently, we want is a distribution over these functions

Sounds Cool!!!



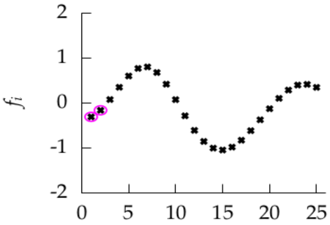
Non Parametric models -GPs

- **But How??**



Non Parametric models -Gaussian Processes

- Lets define a vector of function values evaluated at n points for $x_i \in \mathcal{X}$ as $\mathbf{f} = (f(x_1), f(x_2), .., f(x_n))$
- Lets also assume the notion of **smoothness** of $\mathbf{f}$ to mean points $(f(x_i), f(x_{i+1}))$ that are closer in space are highly **correlated**.



(a) A 25 dimensional correlated random variable (values ploted against index)

$$\begin{bmatrix} 1 & 0.96587 \\ 0.96587 & 1 \end{bmatrix}$$

(b) correlation between f_1 and f_2.

Figure: Smoothness Assumption. Source: Neil Lawrence

Non Parametric models -Gaussian Processes

Definition: Gaussian Process:

- Guassian processes GPs assume neighbouring points x_i, x_{i+1} are correlated and function values f_i, f_{i+1} are distributed multivariate gaussian
- Hence, GPs are parameterized by $\mu(x)$ and covariance function or kernel $K(x_i, x_{i+1})$

$$p(f_i, f_{i+1}) = \mathbf{GP}(\mu, K) \tag{1}$$

$$\mu = \begin{bmatrix} \mu(x_i) \\ \mu(x_{i+1}) \end{bmatrix}, K = \begin{bmatrix} K(x_i, x_i) & K(x_i, x_{i+1}) \\ K(x_{i+1}, x_i) & K(x_{i+1}, x_{i+1}) \end{bmatrix} \tag{2}$$

Non Parametric models - Gaussian Processes

Similarly $p(\mathbf{f}) = p(f(x_1), f(x_2), ..., f(x_n))$ is also multivariate guassian given by

$$p(\mathbf{f}) = \mathbb{N}(\mu, K) \tag{3}$$

where

$$\mu = \begin{bmatrix} \mu(x_1) \\ \mu(x_1) \\ . \\ . \\ \mu(x_n) \end{bmatrix}, K = \begin{bmatrix} K(x_1, x_1) & K(x_1, x_2) & .. & K(x_1, x_n) \\ K(x_2, x_1) & K(x_2, x_2) & .. & K(x_2, x_n) \\ ..... \\ K(x_n, x_1) & K(x_n, x_2) & .. & K(x_n, x_n) \end{bmatrix} \tag{4}$$

Note:

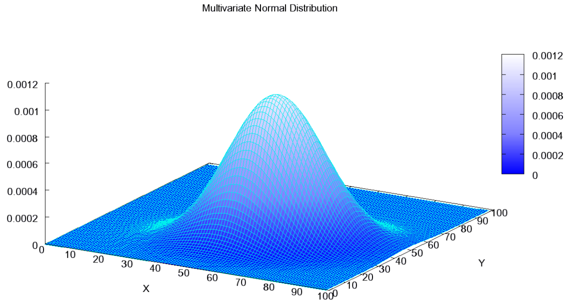
- function **K** generates the covariance matrix Σ
- Σ must be **positive definite** functions/matrices
- Note also that **f** could easily be infinite dimension as n tend to infinitiny

Brief Note on Multivariate Norm

Multivariate Normal - Statistic's swiss army knife

$$\overline{X}|\mu, \Sigma \sim \text{MVNorm}(\mu, \Sigma)$$

- A highly useful *joint* distribution for *continous, vector-valued* observations
- Parameterized by mean vector μ and covariance matrix Σ



Properties Multivariate Norm

Theorem

Suppose $x = (x_1, x_2)$ is jointly Gaussian with parameters

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \ \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}, \ \Lambda = \Sigma^{-1} = \begin{pmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{pmatrix}$$

Then the marginals are also Gaussians given by

$$\begin{aligned} p(\mathbf{x}_1) &= \mathcal{N}(\mathbf{x}_1 | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11}) \\ p(\mathbf{x}_2) &= \mathcal{N}(\mathbf{x}_2 | \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22}) \end{aligned}$$

Theorem - continues

The posterior is also gaussian given by

$$
\begin{aligned}
p(\mathbf{x}_1|\mathbf{x}_2) &= \mathcal{N}(\mathbf{x}_1|\boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}) \\
\boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \\
&= \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{11}^{-1}\boldsymbol{\Lambda}_{12}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \\
&= \boldsymbol{\Sigma}_{1|2}\left(\boldsymbol{\Lambda}_{11}\boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{12}(\mathbf{x}_2 - \boldsymbol{\mu}_2)\right) \\
\boldsymbol{\Sigma}_{1|2} &= \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} = \boldsymbol{\Lambda}_{11}^{-1}
\end{aligned}
$$

Non Parametric models - Gaussian Processes

- GPs **HAVE** parameters: they are parameterized by μ and class of kernel function $K(x_i, x_j)$:
- However, parameters scale with complexity/data
- An example of a **Kernel** function is

$$K(x_i, x_j|\Theta) = \theta_0 exp\Big[-\frac{||x_i - x_j||^2}{2\ell^2}\Big] \tag{5}$$

Hyper parameters $= [\theta_0, \ell]$ - parameter vector

- ℓ is the **lengthscale**,
- θ_0 is known as the amplitude

Some kernel functions

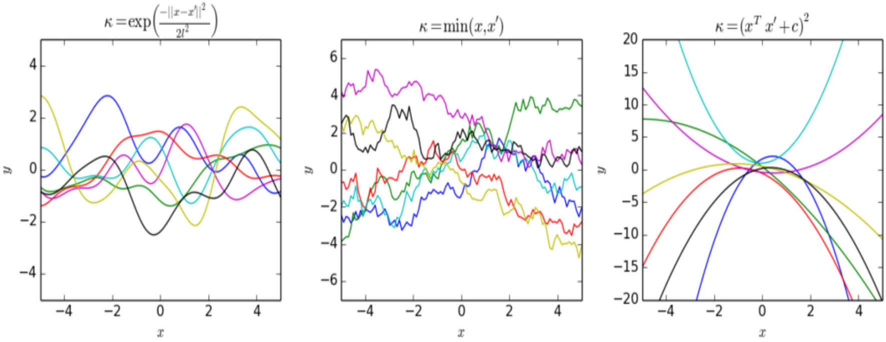


Figure: Effect on choosing different kernels on the prior function distribution. Source: wikipedia

Non Parametric models - Gaussian Processes

Once we design on our kernel function
Gaussian processes can thus be used for bayesian regression:

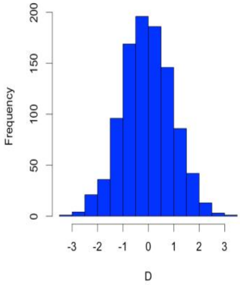
$$p(\mathbf{f}|D) = \frac{p(D|\mathbf{f})p(\mathbf{f})}{p(D)} \tag{6}$$

Where $p(\mathbf{f})$ represents our prior before of the functions
$p(D|\mathbf{f}$ is our likelihood of the Data D given the functions
$p(\mathbf{f}|D)$ is our posterior after observing the data D

Recap: Bayes' Theorem/What's a likelihood?

Posterior Likelihood Prior

$$\Pr(\theta|D) = \frac{\Pr(D|\theta)\Pr(\theta)}{\Pr(D)}$$



$$\Pr(D|\theta) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

NOTE: A likelihood is your model for how the data was generated

Non Parametric models - Where does the likelihood come from?

- All probability modeling starts with a preliminary analysis or visual inpsection of the data
  - Called **E**xploratory **D**ata **A**nalysis (EDA)
  - Motivates choice/formulation of the likelihood

NOTE

- Carrying out EDA doesnt violate spirit of prior specification unless the prior is engineered to look exactly like whats in the data
- This is why we tend to
  - elicit priors from third-party experts
  - use *flat*, non-informative priors

GPs - Regression Prediction

- We have training data $D = \{X, y\}$
- We want to predict y_* given points X_*
- Our model is
  - $y_n = f_n + e_n$
  - $f \sim GP(0, K)$
- Then we can make predictions by combining the likelihood and posterior **theoretically** as

$$p(y_*|X_*, D) = \int p(y_*|X_*, f, D)p(f|D)df \tag{7}$$

Non-Parametric Models - Gaussian Processes -Regression Prediction

- If we assume Gaussian noise: $y_n = f_n + e_n$, where $e \sim N(0, \sigma^2)$
- **Likelihood** is gaussian : IID samples
- Predictive distribution has Gaussian **Analytical solution** as

Gaussian Process

$$p(y_*|X_*, D) \sim \mathbb{N}(f|\mu_*, \Sigma_*) \quad (8)$$

$$\mu_* = K_*^T K_y^{-1} y \quad (9)$$

$$\Sigma_* = K_{**} - K_y^{-1} K_* \quad (10)$$

Non-Parametric Models - Gaussian Processes -Regression Prediction

Gaussian Process

$$
\begin{aligned}
p(y_*|X_*, D) &\sim \mathbb{N}(f|\mu_*, \Sigma_*) && (11)\\
\mu_* &= K_*^T K_y^{-1} y && (12)\\
\Sigma_* &= K_{**} - K_y^{-1} K_* && (13)
\end{aligned}
$$

Where

- $K_y = K + \sigma\mathbb{I}$
- K - is a kernel function covariance matrix of of $x_1, x_2, ..., x_n$
- K_* correlation between $x_1, x_2, ..., x_n$ and X_* - the test points
- K_{**} correlation between X_*

Non-Parametric Models - Gaussian Processes -Regression Prediction

How do we choose hyper-parameters

- Optimizations to find hyperparameters

What about NON-Gaussian Likelihood functions

- For **NON Gaussian Likelihood**, The posterior does **NOT** have analytical form. **NO SUMMARISING STATISTICS**. Hence, we obtain posterior via
  - Sampling
  - Analytic approximations

Why Gaussian Processes

What are they good for

- Good for time series data
- Directly captures model uncertainty
- Work very well no so large datasets
- Ability to be able to encode prior information of the model
- handles model complexity and scalability quite well

Some limitations

- Not so great for large dataset (time/space complexity). However, parallelization, Sparse GPs and other techniques tries to solve this
- May not be your number one go-to option for classification problems

- Notebook on coding GPs using the equations above using python numpy included (Just the intuition).
- Practical handson using GPy Coming up!

Additional resources

- C. E., Rasmussen and C. K. I. Williams (2006) Gaussian Processes for Machine Learning
- Lecture Notes Neil Lawrence - `http://inverseprobability.com`
- Lecture Notes Lehel Csato - `http://www.cs.ubbcluj.ro/~csatol/`
- Lecture Notes Nando de Freitas Video - `http://www.cs.ox.ac.uk/people/nando.defreitas/`
- Gaussian processes website - `http://www.gaussianprocess.org/`

Thank you: Questions?