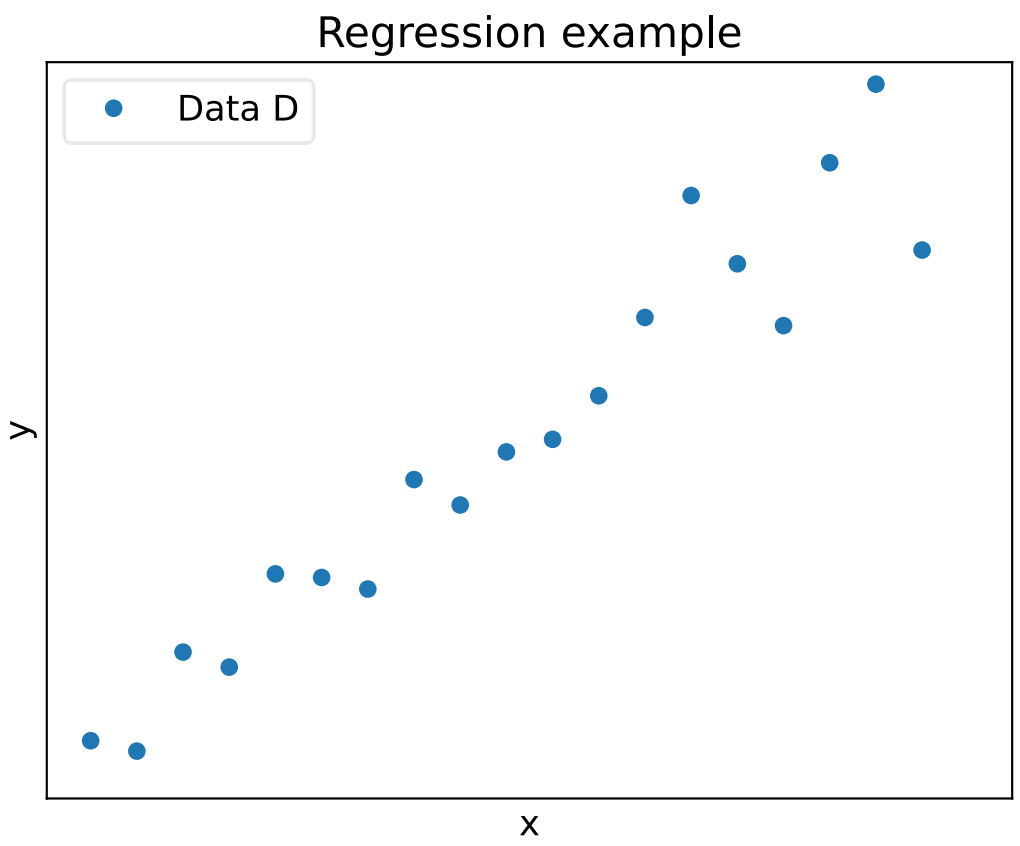
Gaussian Processes

Scientific Machine Learning

Sebastian Klein

Motivation – Regression problems [1]

- Data $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$
- Model $f(x) = w^T x + \epsilon$

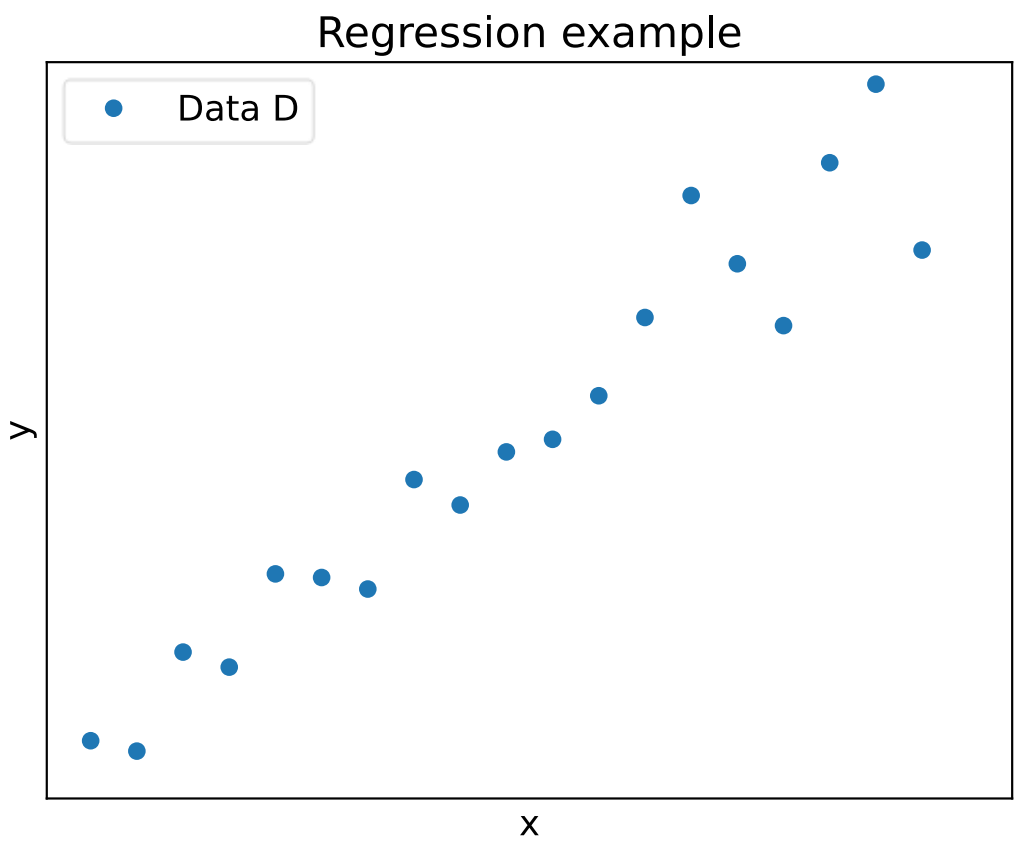


[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)

Motivation – Regression problems [1]

- Data $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$
- Model $f(x) = w^T x + \epsilon$
- MLE – Maximize $p(D|w)$

$$p(D|w) = \prod_{i=1}^{n} p(y_i|x_i, w)$$

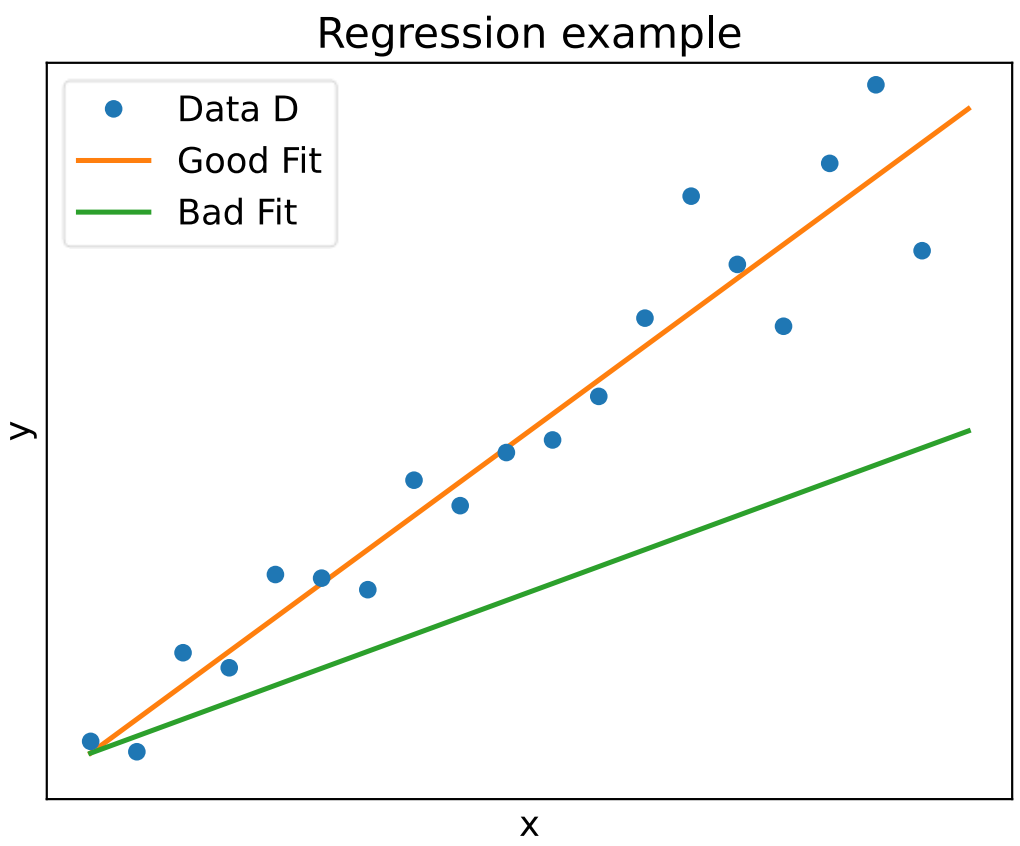


[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)

Motivation – Regression problems [1]

- Data $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$
- Model $f(x) = w^T x + \epsilon$
- MLE – Maximize $p(D|w)$

$$p(D|w) = \prod_{i=1}^{n} p(y_i|x_i, w)$$



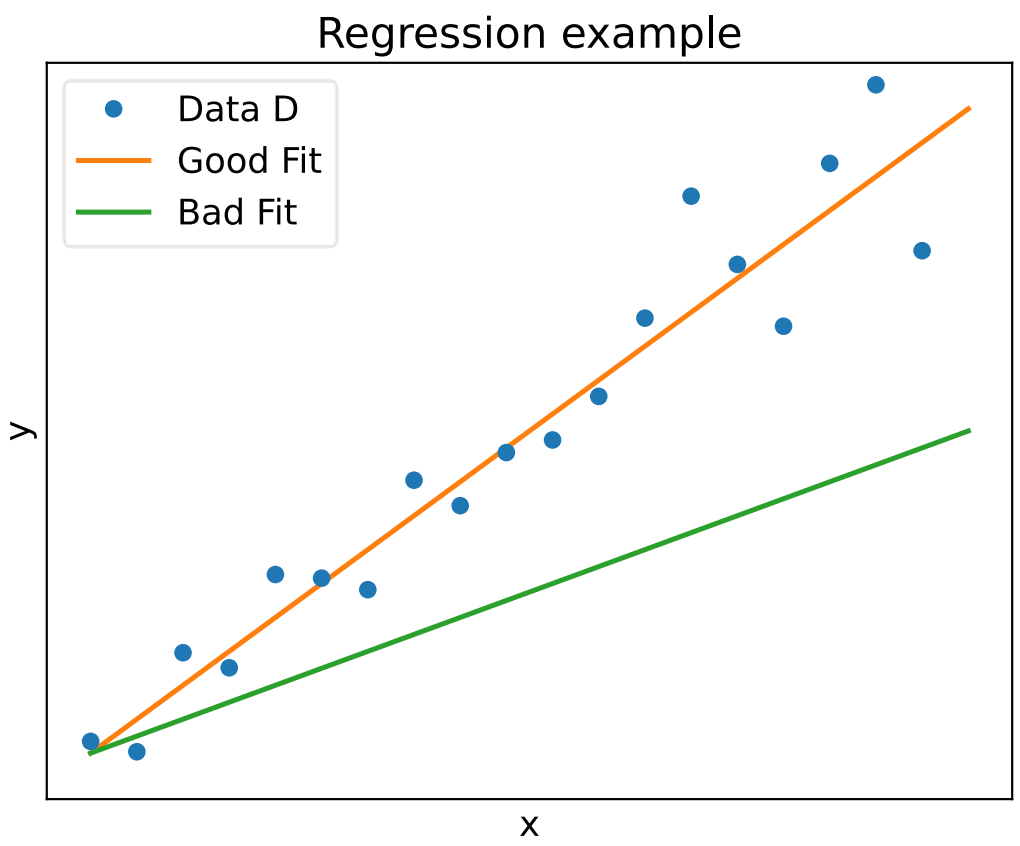
[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)

Motivation – Regression problems [1]

- Data $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$
- Model $f(x) = w^T x + \epsilon$
- MLE – Maximize $p(D|w)$

  $$p(D|w) = \prod_{i=1}^{n} p(y_i|x_i, w)$$
- MAP – Maximize $p(w|D)$
  - Bayes Theorem

    $$p(w|D) = p(D|w)p(w)/p(D)$$



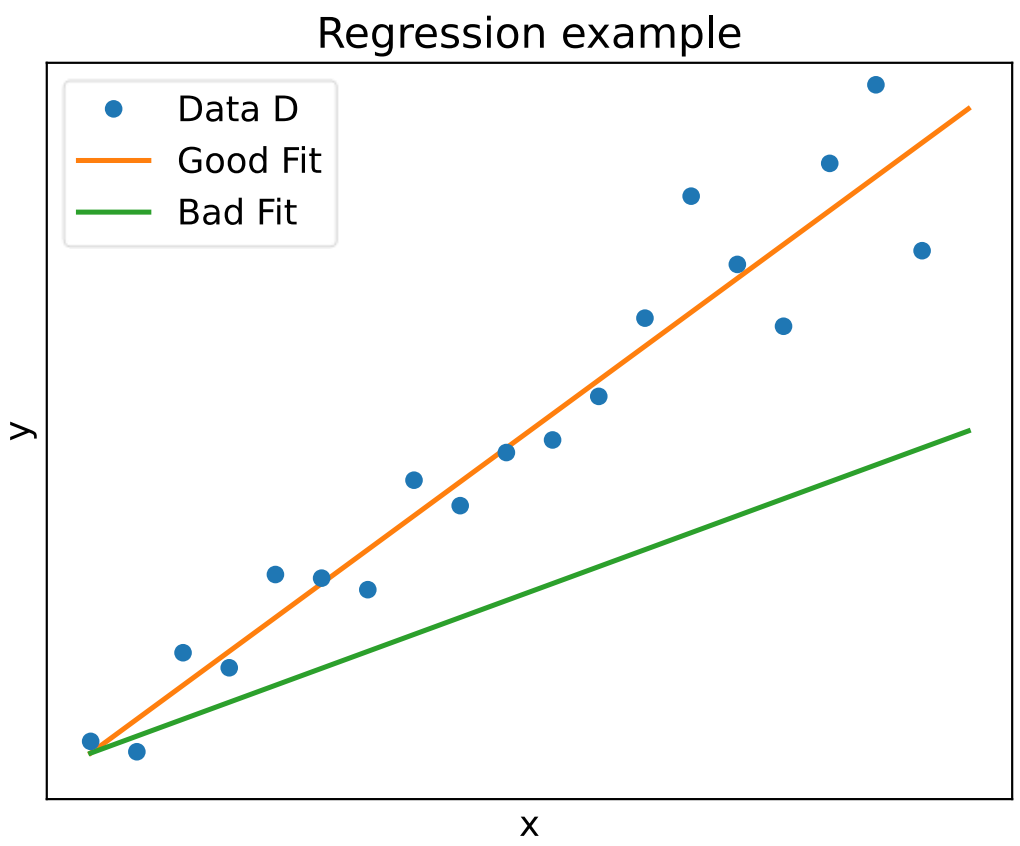
[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)

Motivation – Regression problems [1]

- Data $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$
- Model $f(x) = w^T x + \epsilon$
- MLE – Maximize $p(D|w)$

  $$p(D|w) = \prod_{i=1}^{n} p(y_i|x_i, w)$$
- MAP – Maximize $p(w|D)$
  - Bayes Theorem

    $$p(w|D) = p(D|w)p(w)/p(D)$$
- Assume every probability to be Gaussian



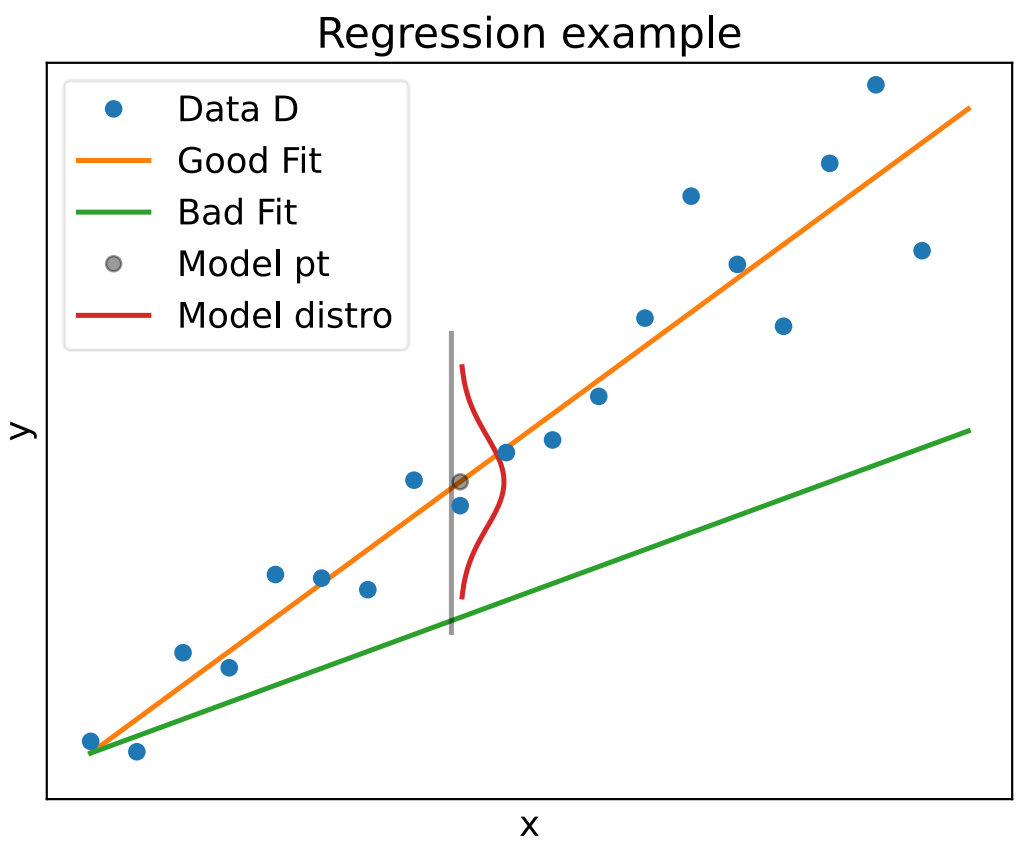
[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)

Motivation – Regression problems [1]

- Data $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$
- Model $f(x) = w^T x + \epsilon$
- MLE – Maximize $p(D|w)$

  $$p(D|w) = \prod_{i=1}^{n} p(y_i|x_i, w)$$
- MAP – Maximize $p(w|D)$
  - Bayes Theorem

    $$p(w|D) = p(D|w)p(w)/p(D)$$
- Assume every probability to be Gaussian -> $p(w|D) \sim \mathcal{N}(\mu, \Sigma)$



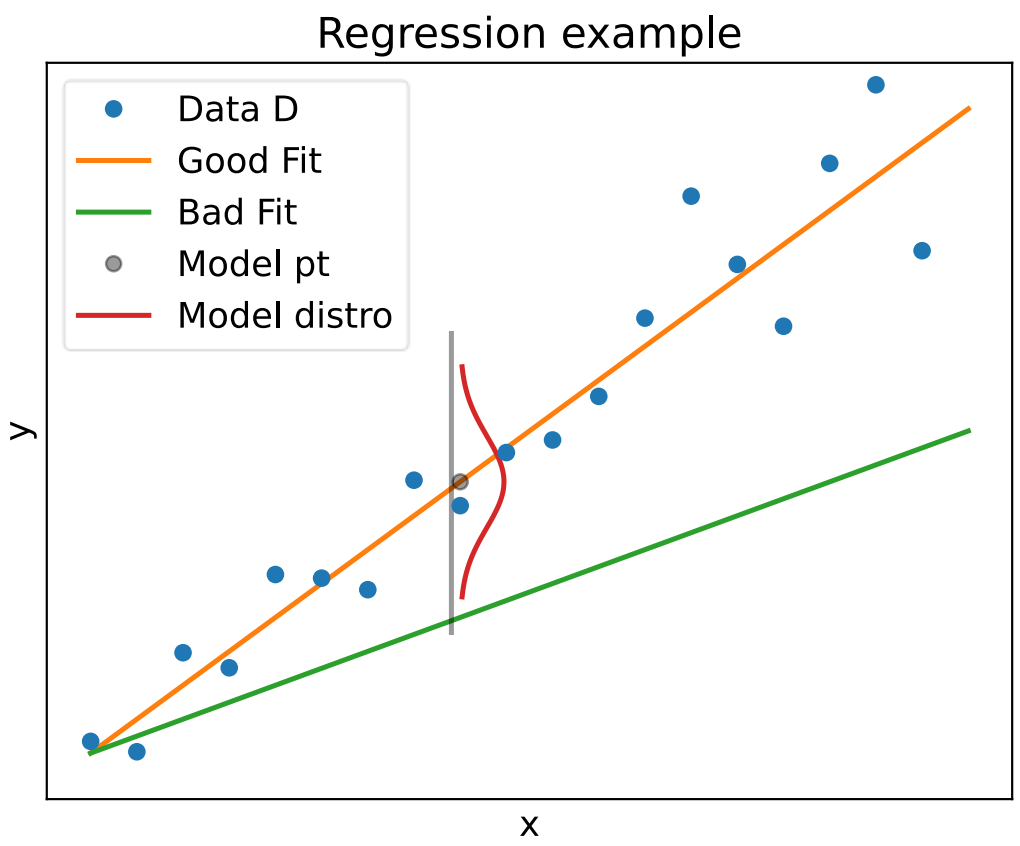
[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)

Motivation – Regression problems [1]

- Data $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$
- Model $f(x) = w^T x + \epsilon$
- MLE – Maximize $p(D|w)$

  $$p(D|w) = \prod_{i=1}^{n} p(y_i|x_i, w)$$
- MAP – Maximize $p(w|D)$
  - Bayes Theorem

    $$p(w|D) = p(D|w)p(w)/p(D)$$
- Assume every probability to be Gaussian -> $p(y|x, \mathrm{D}) \sim \mathcal{N}(\mu, \Sigma)$



[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)

Intermezzo – Multivariate Gaussian[2]

- Multivariat Gaussian distribution/ Normal distribution

$$\mathcal{N}(\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d \det \Sigma}} \exp\left(-\frac{1}{2}(\mathrm{x}-\mu)^T \Sigma^{-1} (\mathrm{x}-\mu)\right)$$

  - d dimensions, μ vector of mean values, Σ covariance matrix -> $\mathrm{diag}(\Sigma) = \sigma^2$
- Result of an operation with Gaussian distributions also Gaussian [1]
- With subsets X, Y

$$P_{X,Y} = \begin{bmatrix} X \\ Y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_X \\ \mu_Y \end{bmatrix}, \begin{bmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} = (\Sigma_{XY})^T & \Sigma_{YY} \end{bmatrix}\right)$$

[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)
[2] Görtler, et al., "A Visual Exploration of Gaussian Processes", Distill, 2019. DOI: https://doi.org/10.23915/distill.00017

Intermezzo – Marginalization [2]

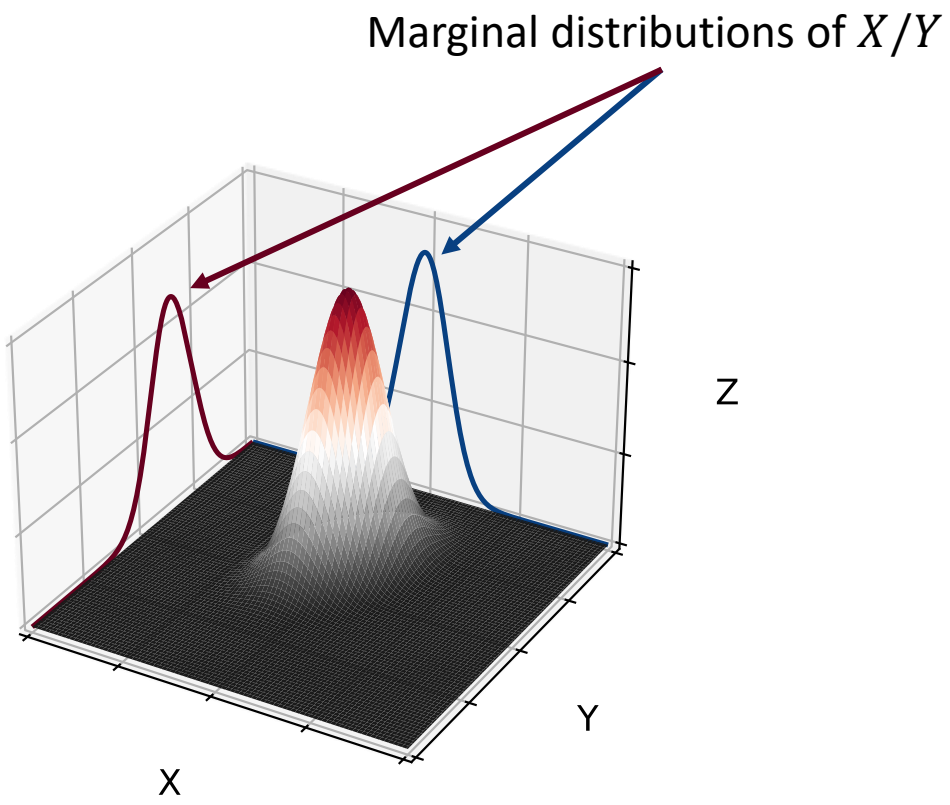
- Extract partial information from $P_{X,Y} = \begin{bmatrix} X \\ Y \end{bmatrix}$

$$p(x|X) = p_X(x) = \int_y p_{X,Y}(x,y)dy$$
$$= \int_y p_{X|Y}(x|y)p_Y(y)dy$$

- Gaussian distribution

$$X \sim \mathcal{N}(\mu_X, \Sigma_X) \mid Y \sim \mathcal{N}(\mu_Y, \Sigma_Y)$$

- X/Y only depending on corresponding μ_*/Σ_*
- Gets us $p(y|x, \mathrm{D}) \sim \mathcal{N}(\mu, \Sigma)$



[2] Görtler, et al., "A Visual Exploration of Gaussian Processes", Distill, 2019. DOI: https://doi.org/10.23915/distill.00017

Intermezzo – Conditioning [2]

- Determine probability of X dependent on Y

$$X|Y \sim \mathcal{N}(\mu_X + \Sigma_{XY}^T \Sigma_{YY}^{-1} (Y - \mu_Y), \quad \Sigma_{XX} - \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX})$$

$$= \mathcal{N}(\underbrace{\Sigma_{XY}^T \Sigma_{YY}^{-1} Y}_{\text{Mean}}, \quad \underbrace{\Sigma_{XX} - \Sigma_{YX}^T \Sigma_{YY}^{-1} \Sigma_{YX}}_{\text{Covariance Matrix}})$$

$$\Sigma = \begin{bmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{XY}^T & \Sigma_{YY} \end{bmatrix}$$

- For $Y|X$ switch all X above with Y
- Conditioning allows to implement Bayesian inference
  - Aka updating model as soon as new data Y is available

[2] Görtler, et al., "A Visual Exploration of Gaussian Processes", Distill, 2019. DOI: https://doi.org/10.23915/distill.00017

Regression as Gaussian Process (GP) [2,9]

Goal: Predict values

How to predict values?

Know $p(y|x, \mathrm{D}) \sim \mathcal{N}(\mu, \Sigma)$

What is a prediction here?

Sample drawn from $\mathcal{N}(\mu, \Sigma)$

How get $\mathcal{N}(\mu, \Sigma)$?

Learn with training data Y

Predict value for test data X

Characteristic of $\mathcal{N}(\mu, \Sigma)$ solely determined by Σ

How to determine μ and Σ?

Set $\mu = 0$

Simplifies conditioning to $X|\mathrm{Y} \sim \mathcal{N}(0, \Sigma_{XX} - \Sigma_{YX}^{T}\, \Sigma_{YY}^{-1}\, \Sigma_{\mathrm{YX}})$

[2] Görtler, et al., "A Visual Exploration of Gaussian Processes", Distill, 2019. DOI: https://doi.org/10.23915/distill.00017
[9] C. Fonnesbeck, Fitting Gaussian Process Models in Python (https://domino.ai/blog/fitting-gaussian-process-models-python)

Regression as Gaussian Process [2,9]

Last result

$$\mathcal{N}(0, \Sigma_{XX} - \Sigma_{YX}^{T}\, \Sigma_{YY}^{-1}\, \Sigma_{YX})$$

How to determine the Σ's?

$\Sigma = Cov(X, X')$ with $Cov(X, X') = k(x, x')$

Evaluate $k(x, x')$ pairwise

Covariance function $k(x, x')$ also called kernel

Can choose any kernel that is

Positive semidefinite

+

Symmetric

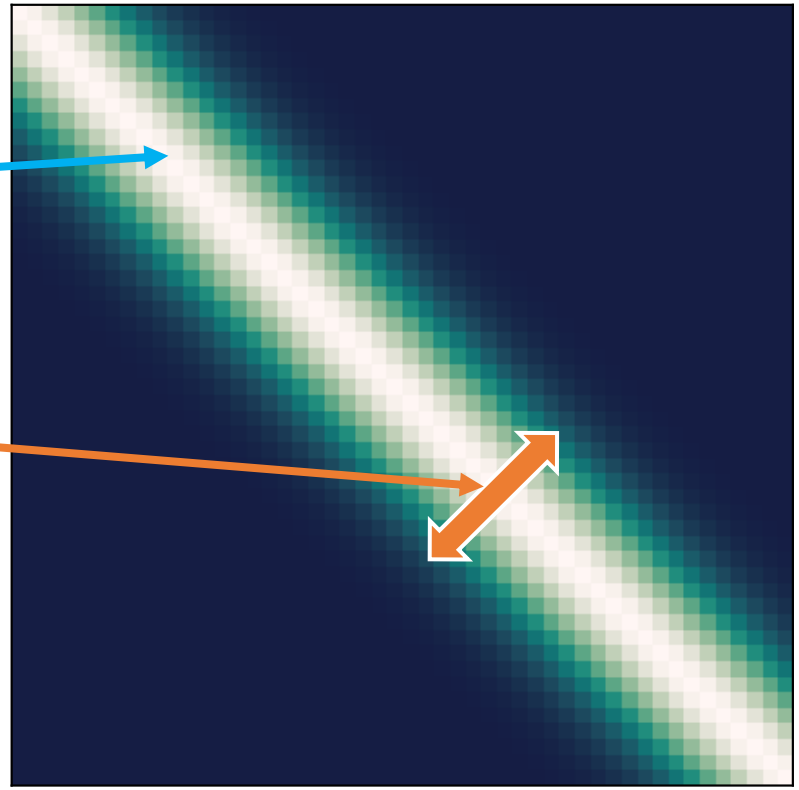
Combinations possible

[2] Görtler, et al., "A Visual Exploration of Gaussian Processes", Distill, 2019. DOI: https://doi.org/10.23915/distill.00017
[9] C. Fonnesbeck, Fitting Gaussian Process Models in Python (https://domino.ai/blog/fitting-gaussian-process-models-python)

Kernels – RBF [3]

- $\mathrm{k}(x_1, x_2) = \sigma^2 \exp\left(-\frac{|x_1 - x_2|^2}{2l^2}\right)$
- σ scale factor
  - "Intensity"
  - Deviation of function from mean
- l length scale parameter
  - Cannot extrapolate more than l units away from data [3]
- Universal [3]
- De-facto default kernel for GP [3]

RBF Kernel

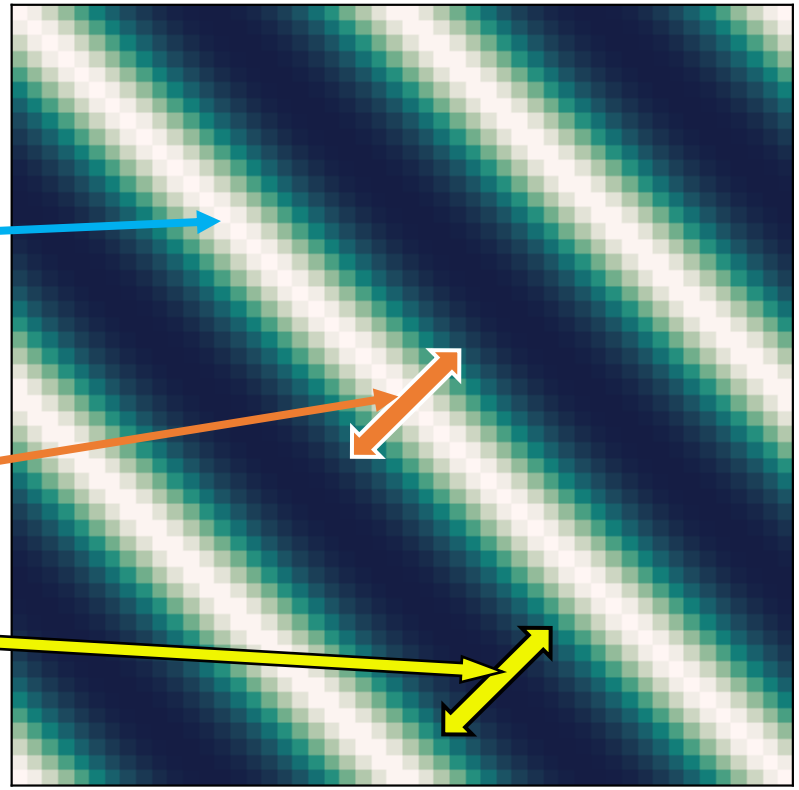


[3] D. K. Duvenaud, Dissertation, University of Cambridge, July 2014, (https: //www.cs.toronto.edu/~duvenaud/thesis.pdf)

Kernels – Periodic [3]

- $\mathrm{k}(x_1, x_2) =$
  $$\sigma^2 \exp\left(-\frac{2\ \sin^2(\pi|x_1 - x_2|/p)}{l^2}\right)$$
- σ scale factor
  - "Intensity"
  - Deviation of function from mean
- l length scale parameter
- p distance between repetitions

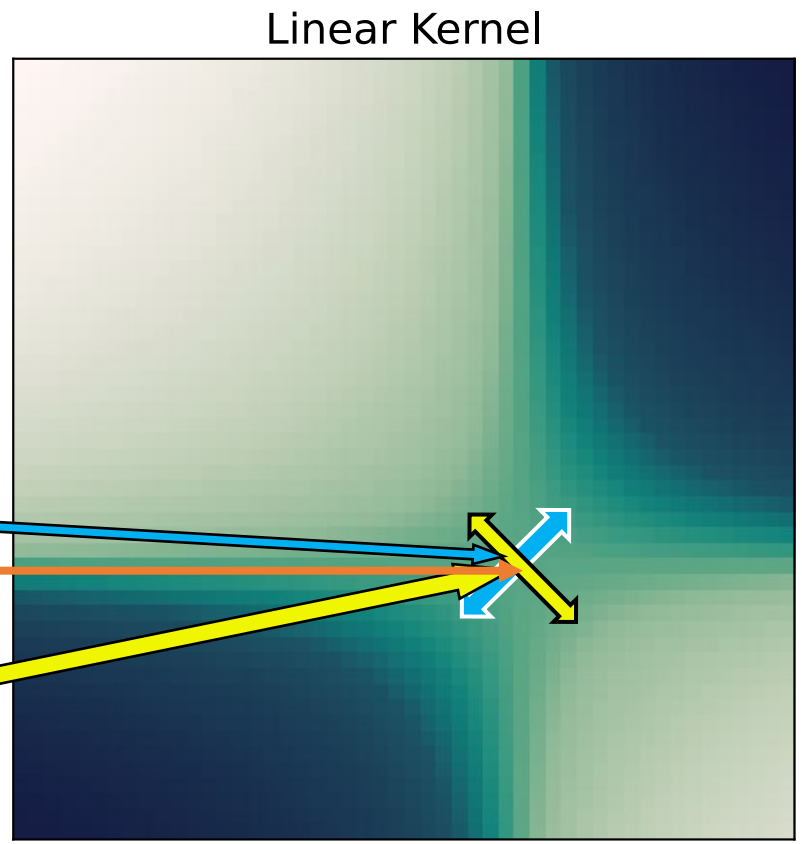
Periodic Kernel



[3] D. K. Duvenaud, Dissertation, University of Cambridge, July 2014, (https: //www.cs.toronto.edu/~duvenaud/thesis.pdf)

Kernels – Linear [3]

- $\mathrm{k}(x_1, x_2) =$
  $\sigma_b^2 + \sigma^2 (x_1 - c)(x_2 - c)$
- Non stationary kernel
  - Dependend on absolut position of input
- σ scale factor
- c offset ($\sigma = 0$)
  - Without noisy training data
- σ_b^2 how far from c $\sigma = 0$

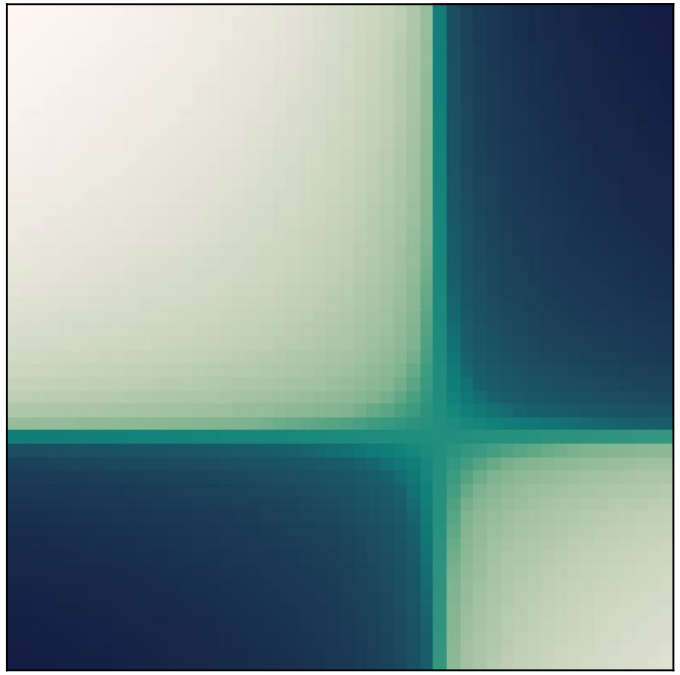


[3] D. K. Duvenaud, Dissertation, University of Cambridge, July 2014, (https: //www.cs.toronto.edu/~duvenaud/thesis.pdf)

Kernels – Linear

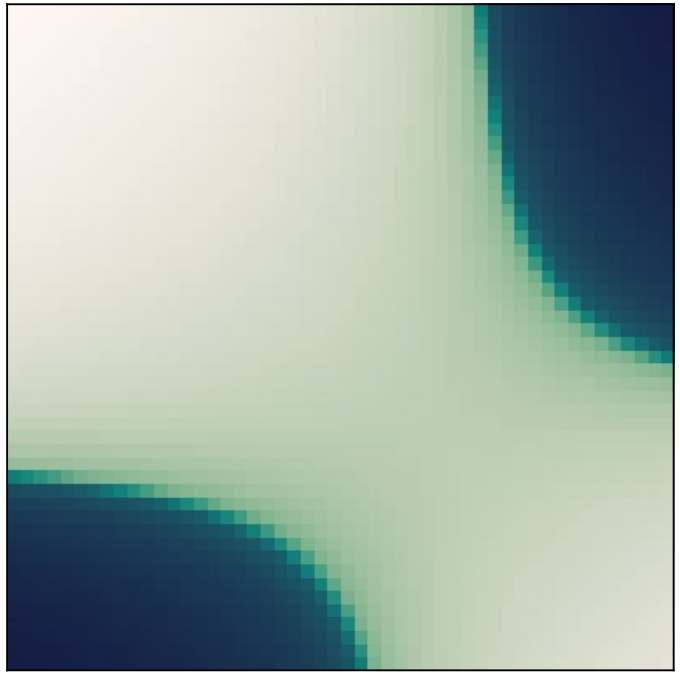
$$\boldsymbol{\sigma_b = 0, c = 3, \sigma = 1}$$

Linear Kernel



$$\boldsymbol{\sigma_b = 4, c = 3, \sigma = 1}$$

Linear Kernel



Regression as GP recipe [1,2]

- Choose appropriate kernel for data
- Determine prior distribution $\mathcal{N}(0, \Sigma_{YY})$ with training data Y
- Determine posterior distribution with conditioning X

$$\mathcal{N}(\Sigma_{XY}^T \Sigma_{YY}^{-1}\, Y, \Sigma_{XX} - \Sigma_{YX}^T\, \Sigma_{YY}^{-1}\, \Sigma_{\mathrm{YX}})$$

- Add noise σ of training data

$$\mathcal{N}(\Sigma_{XY}^T (\Sigma_{YY}^{-1} + \sigma^2 I) Y, \Sigma_{XX} - \Sigma_{YX}^T (\Sigma_{YY}^{-1} + \sigma^2 I) \Sigma_{\mathrm{YX}})$$

- With marginalization extract any μ_i / σ_i with $\sigma_i^2 = \Sigma_{ii}$
  - Got variance of prediction/ confidence of prediction

[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)
[2] Görtler, et al., "A Visual Exploration of Gaussian Processes", Distill, 2019. DOI: https://doi.org/10.23915/distill.00017

Live coding

https://github.com/sebastian-k-physics/SciML_Gaussian_Processes

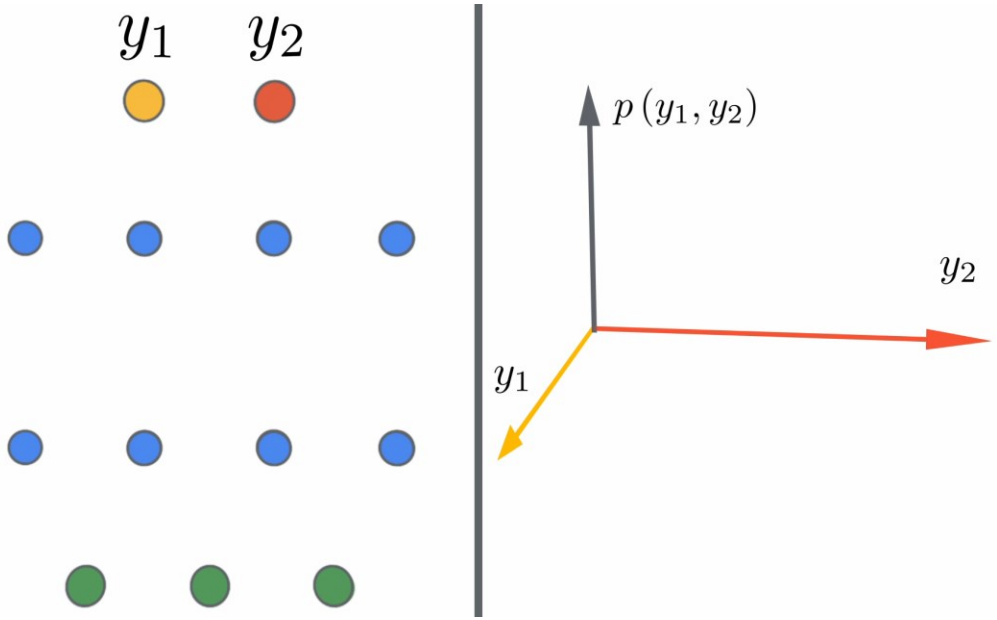
Deep Neural Networks as GP [6]

- Consider deep fully connected network with independent and identically distributed (i.i.d) random parameters
- Final layer sum of i.i.d. terms
  - Infinit width network
  - Central limit theorem (CLT)
  - Computed function drawn from Gaussian distribution
- Replace i.i.d prior over weights + biases with GP over functions
  - Exact bayesian inference
- Need to assume central limit theorem holds
- Specified GP called Neural Network Gaussian Process – NNGP

[6] Lee, Jaehoon; Bahri, Yasaman; Novak, Roman; Schoenholz, Samuel S.; Pennington, Jeffrey; Sohl-Dickstein, Jascha (2017). "Deep Neural Networks as Gaussian Processes". International Conference on Learning Representations. arXiv:1711.00165

Deep Neural Networks as GP – Single layer[6]

- $z_i^1(x) = b_i^1 + \sum_{j=1}^{N_1} W_{ij}^1 x_j(x)$
- $x_j^1(x) = \phi\left(b_j^0 + \sum_{k=1}^{d_{in}} W_{jk}^0 x_k\right)$
- Weight W + bias b i.i.d.
  - $x_j, x_{j'}$ i.i.d. for $j \neq j'$
- $N_1 \to \infty$ CLT $z_i^1(x) \sim \mathcal{N}(\mu, \sigma)$
- $\{z_i^1(x^\alpha), \dots, z_i^1(x^\alpha)\} \sim \mathcal{N}(\mu, \Sigma)$
- $z_i^1 \sim \mathcal{GP}(\mu^1, K^1)$
  - K covariance function



[7]

[6] Lee, Jaehoon; Bahri, Yasaman; Novak, Roman; Schoenholz, Samuel S.; Pennington, Jeffrey; Sohl-Dickstein, Jascha (2017). "Deep Neural Networks as Gaussian Processes". International Conference on Learning Representations. arXiv:1711.00165

[7] Samuel S. Schoenholz, Fast and Easy Infinitely Wide Networks with Neural Tangents https://blog.research.google/2020/03/fast-and-easy-infinitely-wide-networks.html

LETTER Communicated by Edward Snelson

Clustering Based on Gaussian Processes

Hyun-Chul Kim
grass@postech.ac.kr
Department of Computer Science, Yonsei University, 134 Shinchondong, Sudaimunku Seoul, 120-749, Korea

Jaewook Lee
jaewookl@postech.ac.kr
Department of Industrial and Management Engineering, Pohang University of Science and Technology, Pohang, Kyungbuk 797-784, Korea

In this letter, we develop a gaussian process model for clustering. The variances of predictive values in gaussian processes learned from a training data are shown to comprise an estimate of the support of a probability density function. The constructed variance function is then applied to construct a set of contours that enclose the data points, which correspond to cluster boundaries. To perform clustering tasks of the data points, an associated dynamical system is built, and its topological invariant property is investigated. The experimental results show that the proposed method works successfully for clustering problems with arbitrary shapes.

Int J Comput Vis (2010) 88: 169–188
DOI 10.1007/s11263-009-0268-3

Gaussian Processes for Object Categorization

Ashish Kapoor · Kristen Grauman · Raquel Urtasun · Trevor Darrell

Received: 22 July 2008 / Accepted: 1 July 2009 / Published online: 16 July 2009

Abstract Discriminative methods for visual object category recognition are typically non-probabilistic, predicting class labels but not directly providing an estimate of uncertainty. Gaussian Processes (GPs) provide a framework for deriving regression techniques with explicit uncertainty models; we show here how Gaussian Processes with covariance functions defined based on a Pyramid Match Kernel (PMK) can be used for probabilistic object category recognition. Our probabilistic formulation provides a principled way to learn hyperparameters, which we utilize to learn an optimal combination of multiple covariance functions. It also offers confidence estimates at test points, and naturally allows for an active learning paradigm in which points are optimally selected for interactive labeling. We show that with an appropriate combination of kernels a significant boost in classification performance is possible. Further, our experiments indicate the utility of active learning with probabilistic predictive models, especially when the amount of training data labels that may be sought for a category is ultimately very small.

Keywords Object recognition · Gaussian process · Kernel combination · Active learning

1 Introduction

Object categorization is a fundamental problem in image understanding. It remains a challenging learning task given both the variability of images that objects from the same class can produce, as well as the substantial expense of providing high quality image annotations needed to train accurate models. Discriminative methods for visual category learning have yielded promising results in recent years, including various approaches based on support vector machines or nearest neighbor classification (Grauman and Darrell 2005; Zhang et al. 2006; Wallraven et al. 2003; Nister and Stewenius 2006; Lazebnik et al. 2006; Varma and Ray 2007; Bosch et al. 2007; Frome et al. 2007; Kumar and Sminchisescu 2007). However, such methods typically are not explicitly probabilistic, which makes them inadequate when estimates of uncertainty are required. At the same time, probabilistic generative methods that attempt to

SCIENCE ADVANCES | RESEARCH ARTICLE

MATERIALS SCIENCE

Designing exceptional gas-separation polymer membranes using machine learning

J. Wesley Barnett[1]*, Connor R. Bilchak[1]*, Yiwen Wang[1†], Brian C. Benicewicz[2], Laura A. Murdock[2], Tristan Bereau[3], Sanat K. Kumar[1‡]

The field of polymer membrane design is primarily based on empirical observation, which limits discovery of new materials optimized for separating a given gas pair. Instead of relying on exhaustive experimental investigations, we trained a machine learning (ML) algorithm, using a topological, path-based hash of the polymer repeating unit. We used a limited set of experimental gas permeability data for six different gases in ~700 polymeric constructs that have been measured to date to predict the gas-separation behavior of over 11,000 homopolymers not previously tested for these properties. To test the algorithm's accuracy, we synthesized two of the most promising polymer membranes predicted by this approach and found that they exceeded the upper bound for CO_2/CH_4 separation performance. This ML technique, which is trained using a relatively small body of experimental data (and no simulation data), evidently represents an innovative means of exploring the vast phase space available for polymer membrane design.

PHYSICAL REVIEW D **105,** 036014 (2022)

Reconstructing QCD spectral functions with Gaussian processes

Jan Horak,[1] Jan M. Pawlowski,[1,2] José Rodríguez-Quintero,[3] Jonas Turnwald,[1] Julian M. Urban,[1,*] Nicolas Wink,[1] and Savvas Zafeiropoulos[4]

[1]*Institut für Theoretische Physik, Universität Heidelberg, Philosophenweg 16, D-69120 Heidelberg, Germany*
[2]*ExtreMe Matter Institute EMMI, GSI, Planckstr. 1, D-64291 Darmstadt, Germany*
[3]*Department of Integrated Sciences and Center for Advanced Studies in Physics, Mathematics and Computation, University of Huelva, E-21071 Huelva, Spain*
[4]*Aix Marseille Univ, Université de Toulon, CNRS, CPT, Marseille, France*

(Received 4 November 2021; accepted 8 February 2022; published 23 February 2022)

We reconstruct ghost and gluon spectral functions in $2+1$ flavor QCD with Gaussian process regression. This framework allows us to largely suppress spurious oscillations and other common reconstruction artifacts by specifying generic magnitude and length scale parameters in the kernel function. The Euclidean propagator data are taken from lattice simulations with domain wall fermions at the physical point. For the infrared and ultraviolet extensions of the lattice propagators as well as the low-frequency asymptotics of the ghost spectral function, we utilize results from functional computations in Yang-Mills theory and QCD. This further reduces the systematic error significantly. Our numerical results are compared against a direct real-time functional computation of the ghost and an earlier reconstruction of the gluon in Yang-Mills theory. The systematic approach presented in this work offers a promising route toward unveiling real-time properties of QCD.

DOI: 10.1103/PhysRevD.105.036014

Applications of Gaussian Processes at Extreme Lengthscales: From Molecules to Black Holes

Ryan-Rhys Griffiths

Supervisor: Dr. Alpha Lee

Department of Physics
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Wolfson College August 2022

Applications of Gaussian Processes at Extreme Lengthscales: From Molecules to Black Holes

Ryan-Rhys Griffiths

In many areas of the observational and experimental sciences data is scarce. Observation in high-energy astrophysics is disrupted by celestial occlusions and limited telescope time while laboratory experiments in synthetic chemistry and materials science are both time and cost-intensive. On the other hand, knowledge about the data-generation mechanism is often available in the experimental sciences, such as the measurement error of a piece of laboratory apparatus.

Both characteristics make Gaussian processes (GPs) ideal candidates for fitting such datasets. GPs can make predictions with consideration of uncertainty, for example in the virtual screening of molecules and materials, and can also make inferences about incomplete data such as the latent emission signature from a black hole accretion disc. Furthermore, GPs are currently the workhorse model for Bayesian optimisation, a methodology foreseen to be a vehicle for guiding laboratory experiments in scientific discovery campaigns.

The first contribution of this thesis is to use GP modelling to reason about the latent emission signature from the Seyfert galaxy Markarian 335, and by extension, to reason about the applicability of various theoretical models of black hole accretion discs. The second contribution is to deliver on the promised applications of GPs in scientific data modelling by leveraging them to discover novel and performant molecules. The third contribution is to extend the GP framework to operate on molecular and chemical reaction representations and to provide an open-source software library to enable the framework to be used by scientists. The fourth contribution is to extend current GP and Bayesian optimisation methodology by introducing a Bayesian optimisation scheme capable of modelling aleatoric uncertainty, and hence theoretically capable of identifying molecules and materials that are robust to industrial scale fabrication processes.

Results in Chemistry 4 (2022) 100508

Contents lists available at ScienceDirect

Results in Chemistry

journal homepage: www.sciencedirect.com/journal/results-in-chemistry

Comparison of Gaussian process regression, partial least squares, random forest and support vector machines for a near infrared calibration of paracetamol samples

Aminata Sow [a,*], Issiaka Traore [a], Tidiane Diallo [b,c], Mohamed Traore [d], Abdramane Ba [a]

[a] Laboratoire d'Optique, de Spectroscopie et des Sciences Atmosphériques (LOSSA), Faculté des Sciences et Techniques (FST), Université des Sciences, des Techniques et des Technologies de Bamako, Bamako, Mali
[b] Département des Sciences du Médicament, Faculté de Pharmacie, Université des Sciences, des Techniques et des Technologies de Bamako, Bamako, Mali
[c] Laboratoire National de la Santé (LNS), Bamako, Mali
[d] Ecole Nationale d'Ingénieurs Abderhamane Baba Touré, Bamako, Mali

ARTICLE INFO

Keywords:
Paracetamol
Near Infrared Spectroscopy
Data preprocessing
Nonlinear regression models
Linear regression techniques

ABSTRACT

In this article, we analyze the near-infrared (NIR) spectra of fifty-eight (58) commercial tablets of 500 mg of paracetamol from different origins (that is, with different batch numbers) in the local markets in Bamako. The NIR spectra were recorded in the spectral range 930 nm-1700 nm. The samples are divided into forty-eight (48) samples forming the set of calibration (training set) and ten (10) samples used as the validation or test set. To perform multivariate calibration, we apply-three nonlinear regression techniques (Gaussian processes regression (GPR), Random Forest (RF), Support vector machine (KSVM)), along with the traditional linear partial least-squares regression (PLSR) to several data pretreatments of the 58 samples. The results show that the three nonlinear regression calibrations have better prediction performance than PLS as far as RMSE is concerned. To decide the best regression model, we avoid R^2 since this quantity is not a good parameter for this purpose. We will instead consider RMSE when comparing the different multivariate models. Additionally, to assess the impact of data preprocessing, we apply the above regression techniques to the original data, Multi-scattering correction (MSC), standard variate normalization (SNV) correction, smoothing correction, first derivative (FD), and second derivative correction (SD). The overall results reveal that Gaussian Processes Regression (GPR) applied to smooth correction gives the lowest RMSEP = 2.303053e-06 for validation (prediction) and RMSEC = 2.112316e-06 for calibration. In our investigation, one also notices that the developed GPR model is more accurate and exhibits enhanced behavior no matter which data preprocessing is used. All in all, GPR can be seen as an alternative powerful regression tool for NIR spectra of paracetamol samples. The statistical parameters of the proposed model are compared to the results of some other models reported in the literature.

Conclusion – Caveats

- Book by Rasmussen + Lecture of Weinberger from Cornell University error in conditional Gaussian [1][5]

$$X|Y \sim \mathcal{N}(\Sigma_{XY}^T \Sigma_{YY}^{-1}\, Y, \boxed{\Sigma_{XX}} - \Sigma_{YX}^T \boxed{\Sigma_{YY}^{-1}} \Sigma_{\mathrm{YX}}) \checkmark$$

$$X|Y \sim \mathcal{N}(\Sigma_{XY}^T \Sigma_{YY}^{-1}\, Y, \boxed{\Sigma_{YY}} - \Sigma_{YX}^T \boxed{\Sigma_{XX}^{-1}} \Sigma_{\mathrm{YX}}) \times$$

$$\Sigma = \begin{bmatrix} \Sigma_{XX} & \Sigma_{\mathrm{XY}} \\ \Sigma_{XY}^T & \Sigma_{YY} \end{bmatrix}$$

[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)
[5] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning (The MIT Press, Nov. 2005) (https://doi.org/10.7551/mitpress/3206.001.0001).

Conclusion – Caveats

- Book by Rasmussen + Lecture of Weinberger from Cornell University error in conditional Gaussian [1][5]

$$X|Y \sim \mathcal{N}(\Sigma_{XY}^T \Sigma_{YY}^{-1}\, Y, \boxed{\Sigma_{XX}} - \Sigma_{YX}^T \boxed{\Sigma_{YY}^{-1}} \Sigma_{\mathrm{YX}}) \checkmark$$

$$X|Y \sim \mathcal{N}(\Sigma_{XY}^T \Sigma_{YY}^{-1}\, Y, \boxed{\Sigma_{YY}} - \Sigma_{YX}^T \boxed{\Sigma_{XX}^{-1}} \Sigma_{\mathrm{YX}}) \times$$

$$\Sigma = \begin{bmatrix} \Sigma_{XX} & \Sigma_{\mathrm{XY}} \\ \Sigma_{XY}^T & \Sigma_{YY} \end{bmatrix}$$

- Implementation troublesome for beginners (e.g. me)
  - Often differently used notation in literature -> always look at notation of Σ

[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)
[5] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning (The MIT Press, Nov. 2005) (https://doi.org/10.7551/mitpress/3206.001.0001).

Conclusion – Caveats

- Book by Rasmussen + Lecture of Weinberger from Cornell University error in conditional Gaussian [1][5]

$$X|Y \sim \mathcal{N}(\Sigma_{XY}^T \Sigma_{YY}^{-1}\, Y, \boxed{\Sigma_{XX}} - \Sigma_{YX}^T \boxed{\Sigma_{YY}^{-1}} \Sigma_{\mathrm{YX}}) \checkmark$$

$$X|Y \sim \mathcal{N}(\Sigma_{XY}^T \Sigma_{YY}^{-1}\, Y, \boxed{\Sigma_{YY}} - \Sigma_{YX}^T \boxed{\Sigma_{XX}^{-1}} \Sigma_{\mathrm{YX}}) \times$$

$$\Sigma = \begin{bmatrix} \Sigma_{XX} & \Sigma_{\mathrm{XY}} \\ \Sigma_{XY}^T & \Sigma_{YY} \end{bmatrix}$$

- Implementation troublesome for beginners (e.g. me)
- Understanding comes with implementation
  - Scikit-learn, GPflow, PyMC3, GPyTorch etc. only few commands – not too much insight in process “but easy to use”
  - Implementation as shown only found in Blogposts (e.g. [9])

[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)
[5] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning (The MIT Press, Nov. 2005) (https://doi.org/10.7551/mitpress/3206.001.0001).
[9] C. Fonnesbeck, Fitting Gaussian Process Models in Python (https://domino.ai/blog/fitting-gaussian-process-models-python)

Conclusion – Caveats

- Book by Rasmussen + Lecture of Weinberger from Cornell University error in conditional Gaussian [1][5]
- Inconsistent notation in literature (Blogposts etc.)
- Understanding comes with own implementation
- Linear Kernel does not work for unknown reasons
  - RuntimeWarning: covariance is not symmetric positive-semidefinite
  - RuntimeWarning: invalid value encountered in sqrt
    - Diagonal elements of $\Sigma < 0$
  - Need symmetric logarithmic normalization in plotting Σ to see any difference in hyperparameters changing

[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)
[5] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning (The MIT Press, Nov. 2005) (https://doi.org/10.7551/mitpress/3206.001.0001).

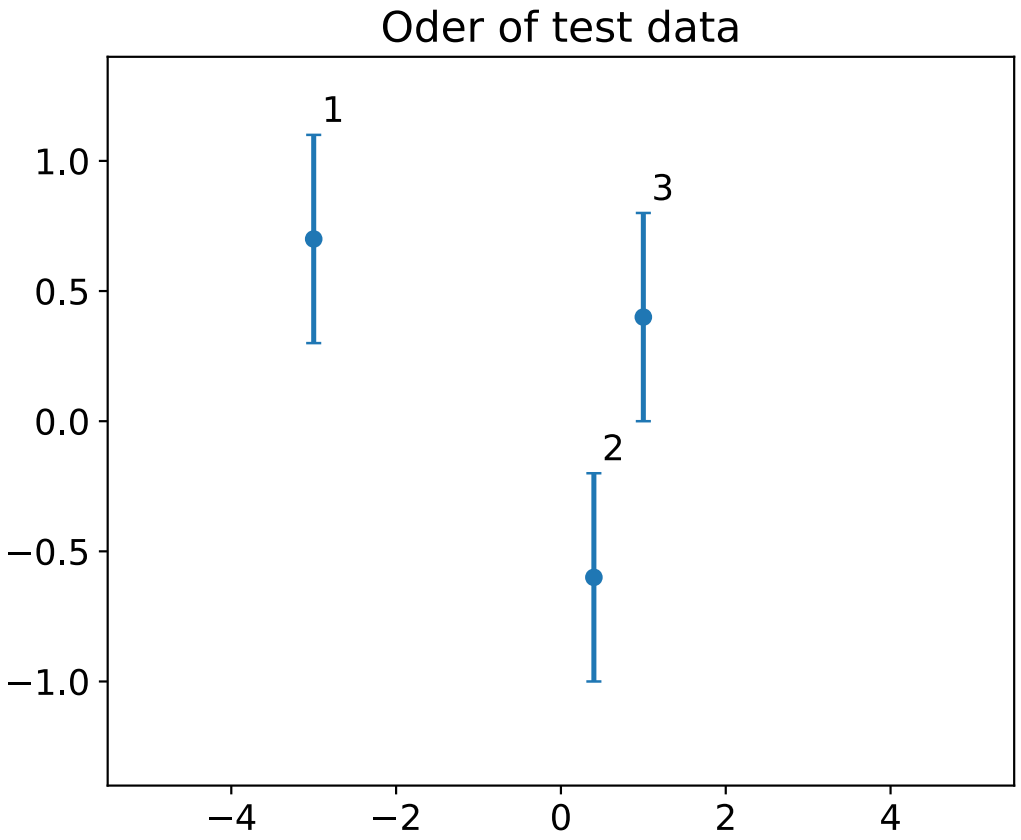
Conclusion – Caveats

- Book by Rasmussen + Lecture of Weinberger from Cornell University error in conditional Gaussian [1][5]
- Inconsistent notation in literature (Blogposts etc.)
- Understanding comes with own implementation
- Linear Kernel does not work for unknown reasons
- Concept of test / training data gets smeared due to Bayesian inference

[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)
[5] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning (The MIT Press, Nov. 2005) (https://doi.org/10.7551/mitpress/3206.001.0001).

Conclusion – Caveats – Test and Training Data

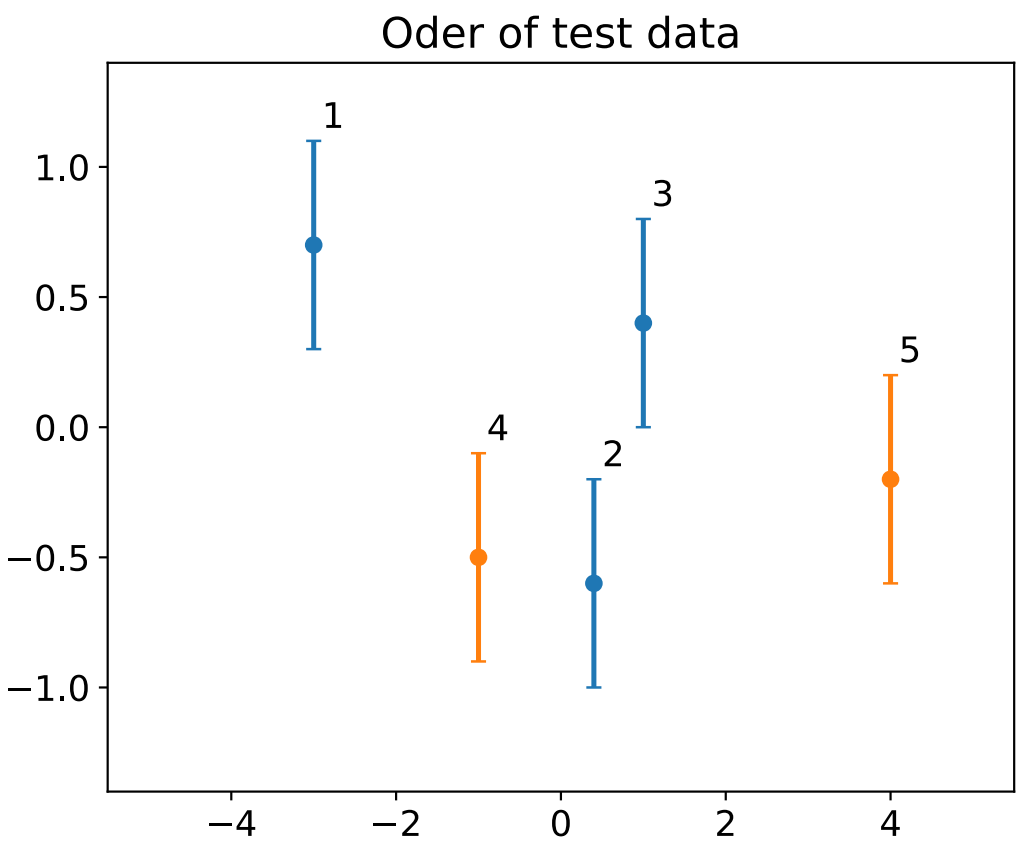
- Concept of test / training data gets smeared
- Adding more test data “old” test data gets training data



Conclusion – Caveats – Test and Training Data

- Concept of test / training data gets smeared
- Adding more test data “old” test data gets training data
- Importance of order of data
  - Effect in implementation unclear
  - In covariance matrix new elements added at end

$$\Sigma = \begin{bmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{XY}^T & \Sigma_{YY} \end{bmatrix}$$

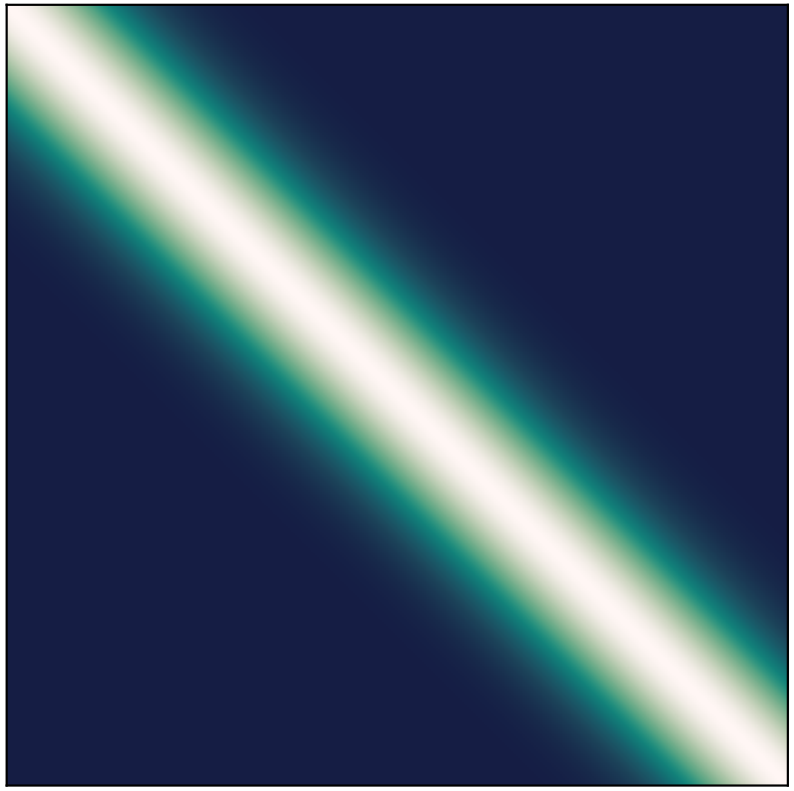


Conclusion – Caveats – Test and Training Data

- Concept of test / training data gets smeared
- Adding more test data “old” test data gets training data
- Importance of order of data
  - Effect in implementation unclear
  - In covariance matrix new elements added at end

$$\Sigma = \begin{bmatrix} \boxed{\Sigma_{XX}} & \Sigma_{\mathrm{XY}} \\ \Sigma_{XY}^{T} & \Sigma_{YY} \end{bmatrix}$$

RBF Kernel without test data

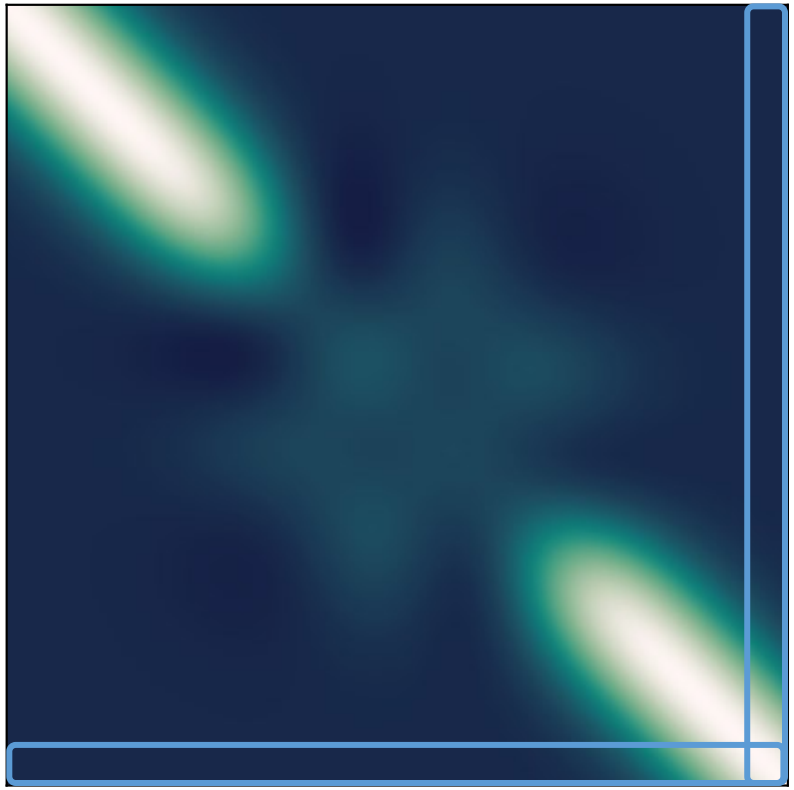


Conclusion – Caveats – Test and Training Data

- Concept of test / training data gets smeared
- Adding more test data “old” test data gets training data
- Importance of order of data
  - Effect in implementation unclear
  - In covariance matrix new elements added at end

$$\Sigma = \begin{bmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{XY}^T & \Sigma_{YY} \end{bmatrix}$$

RBF Kernel with test data



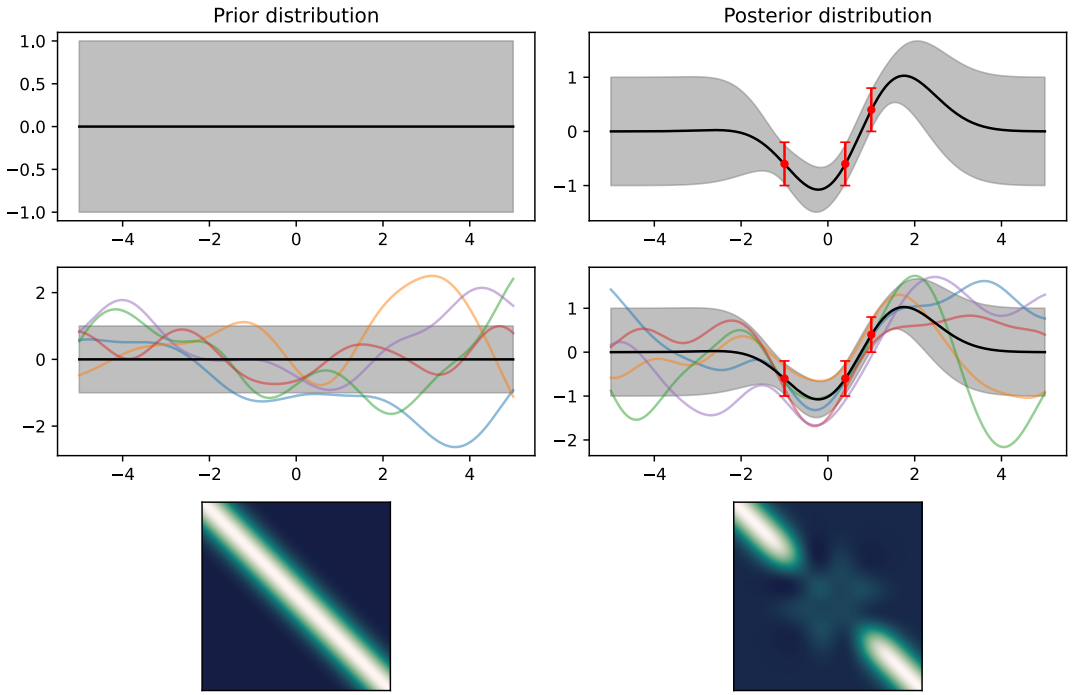
Conclusion – Caveats

- Book by Rasmussen + Lecture of Weinberger from Cornell University error in conditional Gaussian [1][5]
- Inconsistent notation in literature (Blogposts etc.)
- Understanding comes with own implementation
- Linear Kernel does not work for unknown reasons
- Concept of test / training data gets smeared due to Bayesian inference
- “Two approaches”?
  - 1) Start $\mu = 0$ (prior) -> add measurement data (posterior) -> predictions
  - 2) Measurement data -> optimize hyper parameters -> predictions

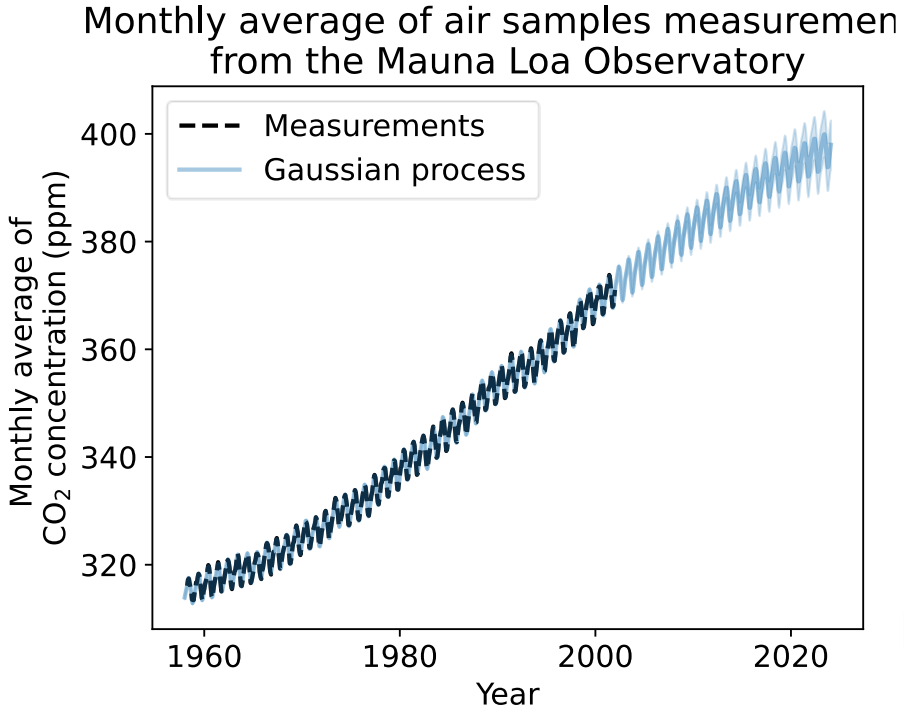
[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)
[5] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning (The MIT Press, Nov. 2005) (https://doi.org/10.7551/mitpress/3206.001.0001).

Conclusion – Caveats – “Two approaches”

Approach 1



Approach 2



[4]

[4] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011 (Figure modified)

Conclusion – Caveats

- Book by Rasmussen + Lecture of Weinberger from Cornell University error in conditional Gaussian [1][5]
- Inconsistent notation in literature (Blogposts etc.)
- Understanding comes with own implementation
- Linear Kernel does not work for unknown reasons
- Concept of test / training data gets smeared due to Bayesian inference
- “Two approaches”
  - Both the same – 1) just for illustration purposes

[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)
[5] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning (The MIT Press, Nov. 2005) (https://doi.org/10.7551/mitpress/3206.001.0001).

Conclusion – Caveats

- Book by Rasmussen + Lecture of Weinberger from Cornell University error in conditional Gaussian [1][5]
- Inconsistent notation in literature (Blogposts etc.)
- Understanding comes with own implementation
- Linear Kernel does not work for unknown reasons
- Concept of test / training data gets smeared due to Bayesian inference
- “Two approaches”
- Hypotheses testing [10]

[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)
[5] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning (The MIT Press, Nov. 2005) (https://doi.org/10.7551/mitpress/3206.001.0001).
[10] Alessio Benavoli, Francesca Mangili, PMLR 38:74-82, 2015. (https://proceedings.mlr.press/v38/benavoli15.html).

Conclusion

- Gained uncertainties of predictions [1]
- Full Bayesian inference implemented [8]
- Incorporate noise into model
- Applications many fields

- Running time $\mathcal{O}(n^3)$ [1][8]
- Kernel choice needs good knowledge of data [3]
- GPs struggle with high dimensional continuous input spaces [8]
  - Popular approach: VAEs [8]
- Assumes homoscedastic Gaussian noise [8]
  - All random variables have same variance
  - Need more data to operate effectively

[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)
[3] D. K. Duvenaud, Dissertation, University of Cambridge, July 2014, (https: //www.cs.toronto.edu/~duvenaud/thesis.pdf))
[8] Griffiths, R.-R. (2022). *Applications of Gaussian Processes at Extreme Lengthscales: From Molecules to Black Holes* [Apollo - University of Cambridge Repository]. https://doi.org/10.17863/CAM.93643

Conclusion

- Gained uncertainties of predictions [1]
- Full Bayesian inference implemented [8]
- Incorporate noise into model
- Applications many fields

- Running time $\mathcal{O}(n^3)$ [1][8]
- Kernel choice needs good knowledge of data [3]
- GPs struggle with high dimensional continuous input spaces [8]
- Assumes homoscedastic Gaussian noise [8]
- Caveats mentioned before

[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)
[3] D. K. Duvenaud, Dissertation, University of Cambridge, July 2014, (https: //www.cs.toronto.edu/~duvenaud/thesis.pdf))
[8] Griffiths, R.-R. (2022). *Applications of Gaussian Processes at Extreme Lengthscales: From Molecules to Black Holes* [Apollo - University of Cambridge Repository]. https://doi.org/10.17863/CAM.93643

Görtler, et al., "A Visual Exploration of Gaussian Processes“

https://distill.pub/2019/visual-exploration-gaussian-processes/

https://doi.org/10.23915/distill.00017

Sources

- Figures if not other stated made with a custom python script for this seminar talk
- K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)
  - The Multivariate gaussian distribution is written here with the normalization factor $\propto \det(\Sigma)^{-1}$ while it should be $\propto \det(\Sigma)^{-1/2}$
  - In notation of conditional normal distribution the formular for the covariance Matrix is wrong, K and K_** are switched
  - Lecture: https://youtu.be/R-NUdqxKjos?si=TKCLcN27yHUVgEpU
- Görtler, et al., "A Visual Exploration of Gaussian Processes", Distill, 2019. DOI: https://doi.org/10.23915/distill.00017
- C. Fonnesbeck, Fitting Gaussian Process Models in Python (https://domino.ai/blog/fitting-gaussian-process-models-python)
- Duvenaud, D. (2014). Automatic model construction with Gaussian processes. https://doi.org/10.17863/CAM.14087
  - Pdf at(https://www.cs.toronto.edu/~duvenaud/thesis.pdf)
  - For the kernel cookbook it selfe: https://www.cs.toronto.edu/~duvenaud/cookbook/
- Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011
- C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning (The MIT Press, Nov. 2005), isbn: 9780262256834, DOI 10.7551/mitpress/3206.001.0001, (https://doi.org/10.7551/mitpress/3206.001.0001).

Sources

- *Lee, Jaehoon; Bahri, Yasaman; Novak, Roman; Schoenholz, Samuel S.; Pennington, Jeffrey; Sohl-Dickstein, Jascha (2017). "Deep Neural Networks as Gaussian Processes". International Conference on Learning Representations. arXiv:1711.00165*
- Samuel S. Schoenholz, Fast and Easy Infinitely Wide Networks with Neural Tangents https://blog.research.google/2020/03/fast-and-easy-infinitely-wide-networks.html
- Kim H-C, Lee J. Clustering Based on Gaussian Processes. Neural computation. 2007;19(11):3088-3107.
- Kapoor, A., Grauman, K., Urtasun, R. et al. Gaussian Processes for Object Categorization. Int J Comput Vis 88, 169–188 (2010). https://doi.org/10.1007/s11263-009-0268-3
- J. Wesley Barnett et al., Designing exceptional gas-separation polymer membranes using machine learning. Sci. Adv.6,eaaz4301(2020).DOI: https://doi.org/10.1126/sciadv.aaz4301
- Griffiths, R.-R. (2022). Applications of Gaussian Processes at Extreme Lengthscales: From Molecules to Black Holes [Apollo - University of Cambridge Repository]. https://doi.org/10.17863/CAM.93643
- Horak J, Pawlowski JM, Rodríguez-Quintero J, et al. Reconstructing QCD spectral functions with Gaussian processes. Physical review. 2022;105:1-12. doi: https://doi.org/10.1103/PhysRevD.105.036014.
- Sow A, Traore I, Diallo T, Traore M, Ba A. Comparison of Gaussian process regression, partial least squares, random forest and support vector machines for a near infrared calibration of paracetamol samples. Results in Chemistry. 2022;4:100508.

Sources

- A. Benavoli, F. Mangili, presented at the Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, ed. by G. Lebanon, S. V. N. Vishwanathan, vol. 38, pp. 74–82, (https://proceedings.mlr.press/v38/benavoli15.html).
  - Used as: [10] Alessio Benavoli, Francesca Mangili, PMLR 38:74-82, 2015. (https://proceedings.mlr.press/v38/benavoli15.html).