Chapter 1

Introduction

> "All models are wrong, but yours are stupid too."
>
> @ML_Hipster (2013)

Prediction, extrapolation, and induction can all be expressed as learning a function from data. A general recipe for learning from data is to perform *inference* - to choose a hyopthesis or to weight a set of hypotheses, based on how compatible they are with the data. To do inference, we start with a weighted set of hypotheses - a *model.* To be able to learn a wide variety of types of functions, we'd like to have an expressive language of models of functions, which can represent both simple parametric functions, such as linear or polynomial, and also complex nonparametric functions specified in terms of properties such as smoothness or periodicity. Fortunately, Gaussian processes (GPs) provide a very general and analytically tractable way of learning many different classes of functions. This chapter will introduce the basic properties of GPs.

Once we have a rich enough language for expressing functions, the remaining question becomes: Which particular model will work well on my problem? What sort of structure should I put in my model? The next chapter will describe the many types of functions that we know how to model using GPs.

1.1 Gaussian Process Models

Gaussian processes are a simple and general class of nonparametric models of functions. To be precise, a GP is any distribution over functions such that any finite subset of function evaluations $f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots f(\mathbf{x}_N)$, have a joint Gaussian distribution (Rasmussen and Williams, 2006). A GP model, before conditioning on data, is completely specified

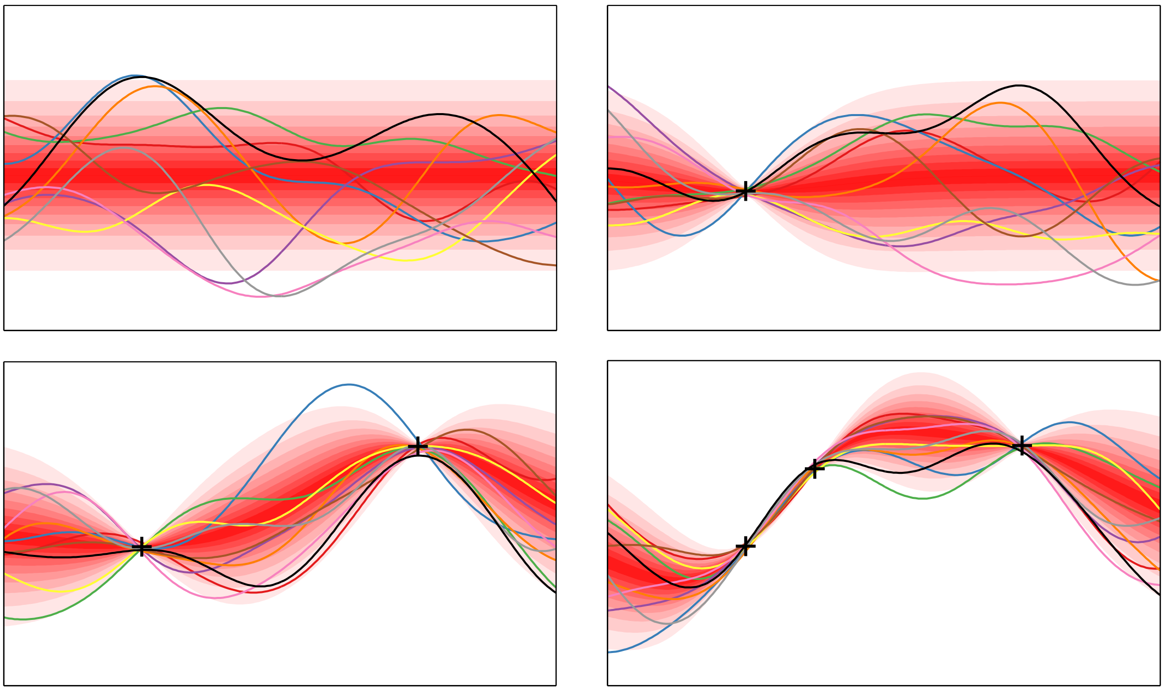

Fig. 1.1 A visual representation of a one-dimensional Gaussian process posterior. Different shades of red correspond to deciles of the predictive density at each input location. Coloured lines show samples from the process. Top left: A GP not conditioned on any datapoints. The remaining plots show the posterior after conditioning on different amounts of data.

by its mean function,

$$\mathbb{E}\left[f(\mathbf{x})\right] = \mu(\mathbf{x}) \tag{1.1}$$

and its covariance function, also called the *kernel*:

$$\text{Cov}(f(\mathbf{x}), f(\mathbf{x}')) = k(\mathbf{x}, \mathbf{x}') \tag{1.2}$$

It is common practice to assume zero mean, since marginalizing over an unknown mean function can be equivalently expressed as a zero-mean GP with a new kernel, or by subtracting the mean function from the data.

After accounting for the mean, the kind of structure which can be captured by a GP model is entirely determined by its kernel, which in turn determines how the model generalizes, or extrapolates to new data.

There are many possible choices of covariance function, and we can specify a wide range of models just by specifying the kernel. Examples of commonly used models not usually cast as GPs are linear regression, splines, and Kalman filters. However, these model classes barely scratch the surface of the wide variety of possible GP models. One of the main difficulties in using GPs is constructing a kernel which represents the structure present in the data.

Gaussian processes can be seen as a dual representation of Bayesian linear regression. (⋆) Cite By moving into this dual space, we pay a price in computational complexity, but gain the ability to model functions to any desired level of detail. Crucially, the marginal likelihood allows us to automatically discover the appropriate amount of detail to use, by Bayesian Occam's razor (MacKay, 2003; Rasmussen and Ghahramani, 2001).

To be concrete, here's the marginal likelihood under a GP of observing a set of function values $[f(\mathbf{x}_1), f(\mathbf{x}_2), \dots f(\mathbf{x}_N)] = \boldsymbol{f}(\mathbf{X})$ at locations given by the rows of $\mathbf{X}$:

$$\begin{aligned} p(\boldsymbol{f}(\mathbf{X})|\mathbf{X}, \mu(\cdot), k(\cdot,\cdot)) &= \mathcal{N}(\boldsymbol{f}(\mathbf{X})|\mu(\mathbf{X}), k(\mathbf{X},\mathbf{X})) \\ &= (2\pi)^{-\frac{N}{2}} |k(\mathbf{X},\mathbf{X})|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} \left(\boldsymbol{f}(\mathbf{X}) - \boldsymbol{\mu}(\mathbf{X})\right)^{\mathsf{T}} k(\mathbf{X},\mathbf{X})^{-1} \left(\boldsymbol{f}(\mathbf{X}) - \boldsymbol{\mu}(\mathbf{X})\right) \right\} \end{aligned} \tag{1.3}$$

This Gaussian likelihood is referred to as the *marginal* likelihood because it implicitly integrates over all possible functions values $\boldsymbol{f}(\bar{\mathbf{X}})$, where $\bar{\mathbf{X}}$ is the set of all locations where we don't have any observations. However, even though we don't need to consider any other locations when computing the likelihood, we can still ask the model which function values are likely to occur at other locations, given that the observations we've seen. The predictive distribution at a test point $\mathbf{x}^\star$ has a simple form:

$$\begin{aligned} p(f(\mathbf{x}^\star)|\boldsymbol{f}(\mathbf{X}), \mathbf{X}, \mu(\cdot), k(\cdot,\cdot)) = \mathcal{N}\Big(& f(\mathbf{x}^\star)|\mu(\mathbf{x}^\star) + k(\mathbf{x}^\star,\mathbf{X})k(\mathbf{X},\mathbf{X})^{-1}\left(\boldsymbol{f}(\mathbf{X}) - \mu(\mathbf{X})\right) \\ & k(\mathbf{x}^\star,\mathbf{x}^\star) - k(\mathbf{x}^\star,\mathbf{X})k(\mathbf{X},\mathbf{X})^{-1}k(\mathbf{X},\mathbf{x}^\star)\Big) \end{aligned} \tag{1.4}$$

These equations may look complex, but only require a few matrix operations to evaluate.

Since the marginal likelihood depends on the kernel, we can select the form of the covariance function, or its parameters, by maximum likelihood, or through inference in a Bayesian model.

Sampling from a GP is also straightforward: a sample from a GP is just a single sample from a single multivariate Normal distribution (1.4). Figure 1.1 shows prior and posterior samples from a GP.

1.1.1 Useful Properties of Gaussian Processes

- **Analytic inference** Given a kernel function and some observations, the predictive posterior distribution can be computed exactly in closed form. This is a rare property for nonparametric models to have.

- **Expressivity** By choosing different covariance functions, we can express a very wide range of modeling assumptions.

- **Integration over hypotheses** The fact that a GP posterior lets us exactly integrate over a wide range of hypotheses means that overfitting is less of an issue than in comparable model classes. It also removes the need for sophisticated optimization schemes. In contrast, much of the neural network literature is devoted to techniques for regularization and optimization.

- **Marginal likelihood** A side benefit of being able to integrate over all hyoptheses is that we compute the *marginal likelihood* of the data given the model. This gives us a principled way of comparing different Gaussian process models.

- **Closed-form posterior** The posterior predictive distribution of a GP is another GP. This means that GPs can easily be composed with other models or decision procedures. For example, in reinforcement learning applications,

- **Easy to Analyze** It may seem unsatisfying to restrict ourselves to a limited model class. Shouldn't we instead learn to use some more flexible model class, such as the set of all computible functions? The answer is: simple models can be used as well-understood building blocks for constructing more interesting models in diverse settings.

  Consider linear models. Although they form an extremely limited model class, they are fast, simple, and easy to analyze. This makes them easy to incorporate into other models or procedures. Linear models may seem like a hopelessly simple model class, but they're arguably the most useful modeling tools in existence.

1.1.2 Limitations of Gaussian Processes

- **Slow inference** Computing the matrix inverse in (1.3) and (1.4) takes $\mathcal{O}(N^3)$ time. Fortunately, this problem can be addressed by approximate inference schemes, and most GP software packages implement several of these.

- **Light tails** We may wish to use non-Gaussian noise models, for instance in order to be robust to outliers, or to perform classification, or some other form of structured prediction. Using non-Gaussian noise models requires approximate inference schemes; fortunately, mature software packages exist which can automatically perform approximate inference for a wide variety of likelihoods.

- **The need to choose a kernel** In practice, the extreme flexibility of GP models means that we are also faced with the difficult task of choosing a kernel. In fact, choosing a useful kernel is equivalent to the problem of learning good features for the data. Typically, human experts choose from among a small set of standard kernels. In this thesis, we hope to go some way towards automating the construction and selection of useful kernels.

1.2 Outline and Contributions of Thesis

This thesis presents a set of related results about how the probabilistic nature of Gaussian process models allows them to be easily extended or composed with other models. Furthermore, the fact that the marginal likelihood is often available (or easily approximable) means that we can evaluate how much evidence the data provides for one structure over another.

Chapter 1.3 contains a wide-ranging overview of many of the types of structured priors on functions that can be easily expressed by constructing appropriate covariance functions. For example, in chapter 1.3, we'll see how GPs can be combined with latent variable models to produce models of nonparametric manifolds. By introducing structure into the kernels of those GPs, we can create manifolds with diverse topological structures, such as cylinders, torii and Möbius strips.

Given a wide variety of structures, plus the ability to evaluate the suitability of each one, it's straightforward to automatically search over models. **Chapter 1.3** shows how to construct a general, open-ended language over kernels - which implies a corresponding language over models. In chapter 1.3, we'll see how the marginal likelihood can guide us into automatically building structured models, and how capturing structure allows us to extrapolate, rather than simply interpolating.

Another benefit of using a relatively simple model class is that the resulting models are easy to understand. **Chapter 1.3** demonstrates how easy-to-understand the resulting models are, by demonstrating a simple system which automatically describes the structure discovered in a dataset by a search over GP models. This system automatically generates reports with graphs and english-language descriptions of GP models. Chapter 1.3 shows that, for the particular language of models constructed in chapter 1.3, it's relatively easy to automatically generate english-language descriptions of the models discovered. Augmented with interpretable plots decomposing the predictive posterior, we demonstrate how to automatically generate useful analyses of time-series. Combined with the automatic model search developed in chapter 1.3, this system represents the beginnings of an "automatic statistician". We discuss the advantages and potential pitfalls of automating the modeling process in this way.

Chapter 1.3 examines the model class obtained by performing dropout in GPs, finding them to have equivalent covariance to *additive Gaussian processes*, a model summing over exponentially-many GP models, each depending on a different subset of the input variables. An polynomial-time algorithm for doing inference in this model class is given, and the resulting model class is characterized and related to existing model classes.

Chapter ?? develops an extension of the GP-LVM in which the latent distribution is a mixture of Gaussians. This model gives rise to a Bayesian clustering model in the clusters have nonparametric shapes. Like the density manifolds learned by the GP-LVM, the shapes of the clusters learned by the iWMM follow the contours of the data density.

Besides having a dual representation as linear regression, GPs can also be derived as the limit of an infinitely-wide neural network. As an example of using GPs as a simple-to-understand building block, **Chapter ??** analyzes deep network models by characterizing the prior over functions obtained by composing GP priors to form *deep Gaussian processes.* We find that, as the number of layers in such models increases, the amount of information retained about the original input diminshes to a single degree of freedom. We show that a simple change to the network architecture fixes this pathology.

1.3 Attribution

This thesis was made possible (and enjoyable to produce) by the substantial contributions of the many co-authors I was fortunate to work with. In this section, I attempt to give proper credit to my tireless co-authors.

Structure through kernels Section **??** of chapter 1.3, describing how symmetries in the kernel of a GP-LVM give rise to priors on manifolds with interesting topologies, is based on a collaboration with David Reshef, Roger Grosse, Josh Tenenbaum, and Zoubin Ghahramani.

Structure Search The research upon which Chapter 1.3 is based was done in collaboration with James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani, and was published in (Duvenaud et al., 2013), where James Lloyd was joint first author. Myself, James Lloyd and Roger Grosse jointly developed the idea of searching through a grammar-based language of GP models, inspired by Grosse et al. (2012), and wrote the first versions of the code together. James Lloyd ran almost all of the experiments. Carl Rasmussen, Zoubin Ghahramani and Josh Tenenbaum provided many conceptual insights, as well as suggestions about how the resulting procedure could be most fruitfully applied.

Automatic Statistician The work appearing in chapter 1.3 was written in collaboration with James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, Zoubin Ghahramani, and was published in (Lloyd et al., 2014). The idea of the correspondence between kernels and adjectives grew out of discussions between James and myself. James Lloyd wrote most of the code to automatically generate reports, and ran all of the experiments. The text was written mainly by myself, James Lloyd, and Zoubin Ghahramani, with many helpful contributions and suggestions from Roger Grosse and Josh Tenenbaum.

Additive Gaussian processes The work in chapter 1.3 discussing additive GPs was done in collaboration with Hannes Nickisch and Carl Rasmussen, who developed a richly parameterized kernel which efficiently sums all possible products of input dimensions. My role in the project was to examine the properties of the resulting model, clarify the connections to existing methods, and to create all figures and run all experiments. This work was previously published in (Duvenaud et al., 2011).

Warped Mixtures The work comprising the bulk of chapter **??** was done in collaboration with Tomoharu Iwata and Zoubin Ghahramani, and appeared in (Iwata et al., 2013). Specifically, the main idea was borne out of a conversation between Tomo and myself, and together we wrote almost all of the code together as well as the paper. Tomo ran most of the experiments. Zoubin Ghahramani provided initial guidance, as well as many helpful suggestions throughout the project.

Deep Gaussian Processes The ideas contained in chapter **??** were developed through discussions with Oren Rippel, Ryan Adams and Zoubin Ghahramani, and appear in (Duvenaud et al., 2014). The derivations, experiments and writing were done mainly by myself, with many helpful suggestions by my co-authors.

References

David Duvenaud, Hannes Nickisch, and Carl Edward Rasmussen. Additive Gaussian processes. In *Advances in Neural Information Processing Systems 24*, pages 226–234, Granada, Spain, 2011. (page 7)

David Duvenaud, James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. Structure discovery in nonparametric regression through compositional kernel search. In *Proceedings of the 30th International Conference on Machine Learning*, June 2013. (page 7)

David Duvenaud, Oren Rippel, Ryan P. Adams, and Zoubin Ghahramani. Avoiding pathologies in very deep networks. Reykjavik, Iceland, April 2014. URL `http://arxiv.org/pdf/1402.5836.pdf`. (page 8)

Roger B. Grosse, Ruslan Salakhutdinov, William T. Freeman, and Joshua B. Tenenbaum. Exploiting compositionality to explore a large space of model structures. In *Uncertainty in Artificial Intelligence*, 2012. (page 7)

Tomoharu Iwata, David Duvenaud, and Zoubin Ghahramani. Warped mixtures for nonparametric cluster shapes. Bellevue, Washington, July 2013. URL `http://arxiv.org/pdf/1206.1846`. (page 7)

James Robert Lloyd, David Duvenaud, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. Automatic construction and natural-language description of nonparametric regression models. Technical Report arXiv:1402.4304 [stat.ML], 2014. (page 7)

David JC MacKay. *Information theory, inference, and learning algorithms*. Cambridge university press, 2003. (page 3)

@ML_Hipster. "…essentially, all models are wrong, but yours are stupid too." – G.E.P. Box in a less than magnanimous mood., 2013. URL `https://twitter.com/ML_Hipster/status/394577463990181888`. (page 1)

Carl Edward Rasmussen and Zoubin Ghahramani. Occam's razor. *Advances in neural information processing systems*, pages 294–300, 2001. (page 3)

C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*, volume 38. The MIT Press, Cambridge, MA, USA, 2006. (page 1)