Chapter 1

Introduction

> "I only work on intractable nonparametrics - Gaussian processes don't count."
>
> Sinead Williamson, personal communication

1.1 Regression

The general problem of regression consists of learning a function f mapping from some input space $\mathcal{X}$ to some output space $\mathcal{Y}$. We would like an expressive language which can represent both simple parametric forms of f such as linear, polynomial, etc. and also complex nonparametric functions specified in terms of properties such as smoothness, periodicity, etc. Fortunately, Gaussian processes (GPs) provide a very general and analytically tractable way of capturing both simple and complex functions.

1.2 Gaussian process models

Gaussian processes are a flexible and tractable prior over functions, useful for solving regression and classification tasksRasmussen and Williams (2006). The kind of structure which can be captured by a GP model is mainly determined by its *kernel*: the covariance function. One of the main difficulties in specifying a Gaussian process model is in choosing a kernel which can represent the structure present in the data. For small to medium-sized datasets, the kernel has a large impact on modeling efficacy.

Gaussian processes are distributions over functions such that any finite subset of function evaluations, $(f(x_1), f(x_2), \ldots f(x_N))$, have a joint Gaussian distribution (Rasmussen and Williams, 2006). A GP is completely specified by its mean function, $\mu(x) = \mathbb{E}(f(x))$

and kernel (or covariance) function $k(x, x') = \text{Cov}(f(x), f(x'))$. 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. The structure of the kernel captures high-level properties of the unknown function, f, which in turn determines how the model generalizes or extrapolates to new data. We can therefore define a language of regression models by specifying a language of kernels.

1.2.1 Useful properties of Gaussian process models

- **Tractable inference** Given a kernel function, the 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 - for example, neural nets.
- **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, (⋆) Carl's reinforcement learning work.

Figure 1.1 shows a Gaussian process posterior. Typically, it's rendered with the mean and +- 2SD, but there's nothing special about mean.

1.2.2 Gaussian process models in practice

The class of models that could be called Gaussian processes is extremely broad. Examples of commonly used models not usually cast as GPs are linear regression, splines, some forms of generalized additive models, and Kalman filters.

Applying GP regression as a non-expert is a daunting task. The first problem one encounters when attempting to apply GPs to a particular task is that the choice of kernel is very important, but it's usually not clear which covariance function is appropriate.

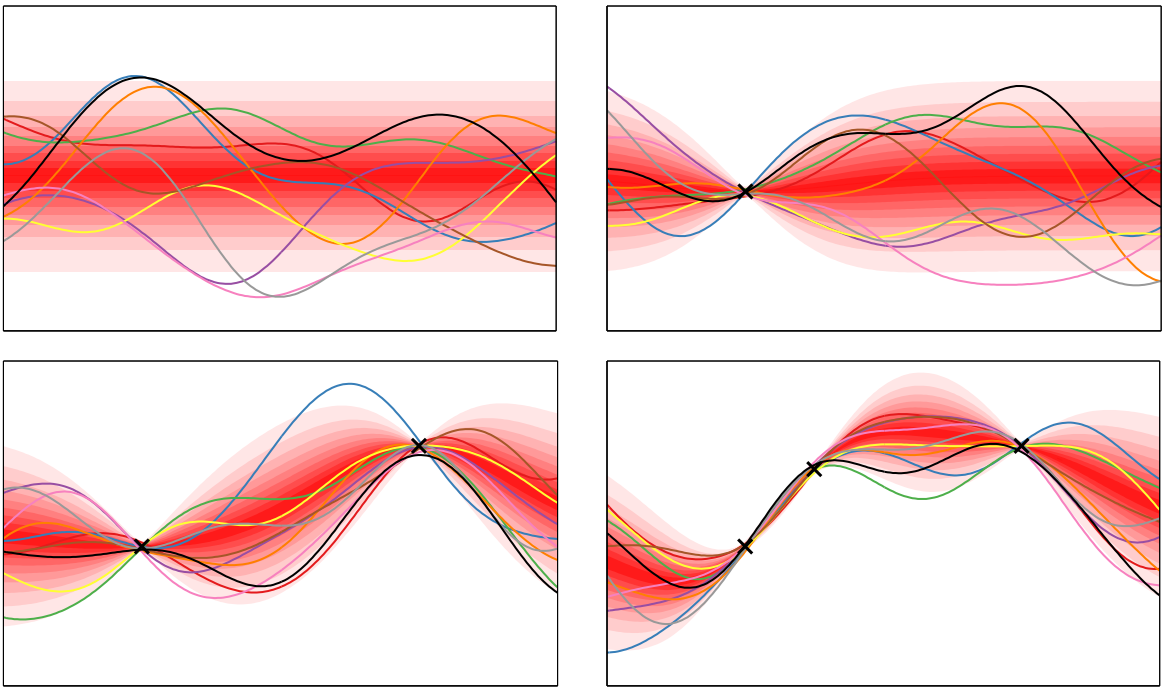

Fig. 1.1 A visual representation of a one-dimensional Gaussian process posterior. Red isocountours show the marginal density at each input location. Coloured lines are samples from the posterior.

In some instances, using a 'default' kernel yields acceptable performance. Many frequentist methods assume a characteristic kernel, such as the squared-exp. This choice is motivated by the fact that, in the limit of infinite data, and a shrinking lengthscale, the estimate of the function will converge asymptotically to the truth. [citation needed]

As we shall see in this thesis, in the small-to-medium data range, the choice of of kernel is extremely important. This is especially true when one wishes to do extrapolation.

When I first used the GPML package, I was stymied by the lack of a default kernel. Faced with a bewildering array of possible kernels to try, I simply tried each one, and chose the one with the best predictive performance on a held-out test set.

Why not simply automate this process as part of the software package? After strarting my Ph.D.,

Discovering interpretable structure Besides allowing faster learning and extrapolation, learning a more structured kernel sometimes has the added benefit of making the resulting model more intepretable. This is a similar motivation as for the use of

sparsity-inducing methods: on many real datasets, the signal can be well-predicted by some small subset of the inputs. Identifying this subset allows both better generalization, and a more interpretable model.

1.2.3 Why assume zero mean?

In literature, as well as in practice, it is common to construct GP priors with a zero mean function. This might seem strange, since it is presumably a good place to put prior information, or if we are comparing models, to express since marginalizing over an unknown mean function can be equivalently expressed as a different GP with zero-mean, and another term added to the kernel. Specifically, if we wish to model an unknown function $f(\mathbf{x})$ with known mean $m(\mathbf{x})$, (with unknown magnitude $c \sim \mathcal{N}(0, \sigma_c^2)$), we can equivalently express this model using another GP with zero mean:

$$f \sim \mathcal{GP}(cm(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \quad c \sim \mathcal{N}\left(0, \sigma_c^2\right) \iff f \sim \mathcal{GP}\left(\mathbf{0}, c^2 m(\mathbf{x}) m(\mathbf{x}') + k(\mathbf{x}, \mathbf{x}')\right) \tag{1.1}$$

By moving the mean function into the covariance function, we get the same model, but we can integrate over the magnitude of the mean function at no additional cost. This is one advantage of moving as much structure as possible into the covariance function. In fact, we can view GP regression as simply implicitly integrating over the magnitudes of (possibly uncountably many) different mean functions all summed together.

1.2.4 Why not simply learn the mean function?

One might ask: besides integrating over the magnitude, what is the advantage of moving the mean function into the covariance function? After all, mean functions are certainly more interpretable than a posterior distribution over functions.

Instead of searching over a large class of covariance functions, which seems strange and unnatural, we might consider simply searching over a large class of structured mean functions, assuming a simple i.i.d. noise model. This is the approach taken by practically every other regression technique: neural networks, decision trees, boosting, etc. . If we could integrate over a wide class of possible mean functions, we would have a very powerful learning an inference method. The problem faced by all of these methods is the well-known problem *overfitting*. If we are forced to choose just a single function with which to make predictions, we must carefully control the flexibility of the model we

learn, generally prefering "simple" functions, or to choose a function from a restricted set.

If, on the other hand, we are allowed to keep in mind many possible explanations of the data, *there is no need to penalize complexity.* [cite Occam's razor paper?] The power of putting structure into the covariance function is that doing so allows us to implictly integrate over many functions, maintaining a posterior distribution over infinitely many functions, instead of choosing just one. In fact, each of the functions being considered can be infinitely complex, without causing any form of overfitting. For example, each of the samples shown in figure 1.1 varies randomly over the whole real line, never repeating, each one requiring an infinite amount of information to describe. Choosing the one function which best fits the data will almost certainly cause overfitting. However, if we integrate over many such functions, we will end up with a posterior putting mass on only those functions which are compatible with the data. In other words, the parts of the function that we can determine from the data will be predicted with certainty, but the parts which are still uncertain will give rise to a wide range of predictions.

To repeat: *there is no need to assume that the function being modeled is simple, or to prefer simple explanations* in order to avoid overfitting, if we integrate over many possible explanations rather than choosing just the one.

1.3 Outline and Contributions

This thesis aims to present 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 ?? 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.

Chapter ?? shows how to construct a general, open-ended language over kernels - which implies a corresponding language over models. 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 ?? shows that, for the particular language of models constructed in chapter **??**, 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 **??**, 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 ?? 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.

Chapter ?? examines the prior over functions obtained by composing GP priors to form *deep Gaussian processes*, and relates them to existing neep neural network architecures. 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.

1.4 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 **??**, 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 **??** 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). [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 **??** was written in collaboration with James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, Zoubin Ghahramani, and was published in (Lloyd et al., 2014), in which James Lloyd is joint first author. 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 **??** 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)

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