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 tasks**?**. 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 (**?**). A

GP is completely specified by its mean function, $\mu(x) = \mathbb{E}(f(x))$ and kernel (or covariance) function $k(x, x') = \mathrm{Cov}(f(x), f(x'))$. 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 Why assume zero-mean?

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.

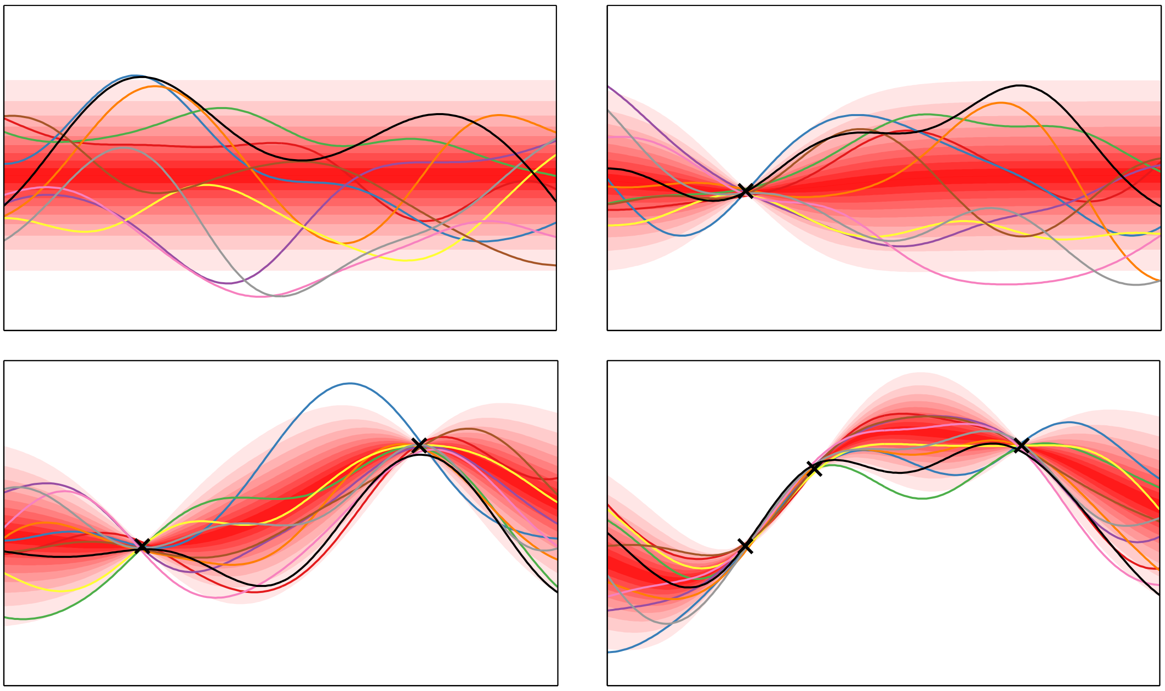


Fig. 1.1 A visual representation of a one-dimensional Gaussian process posterior. Red isocountours show the marginal density at each input location. Coloured lines are samples from the posterior.

1.3 Latent Variable Models

Besides being useful for modeling functions, a simple extension allows GPs to be useful for general density modeling.

Unfortunately, this extension causes many of the useful properties of the GP not to hold.

The GP-LVM can also be thought of as a method for modeling the covariance matrix between all rows of Y using a number of parameters which grows linearly with N.

1.4 Outline

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.

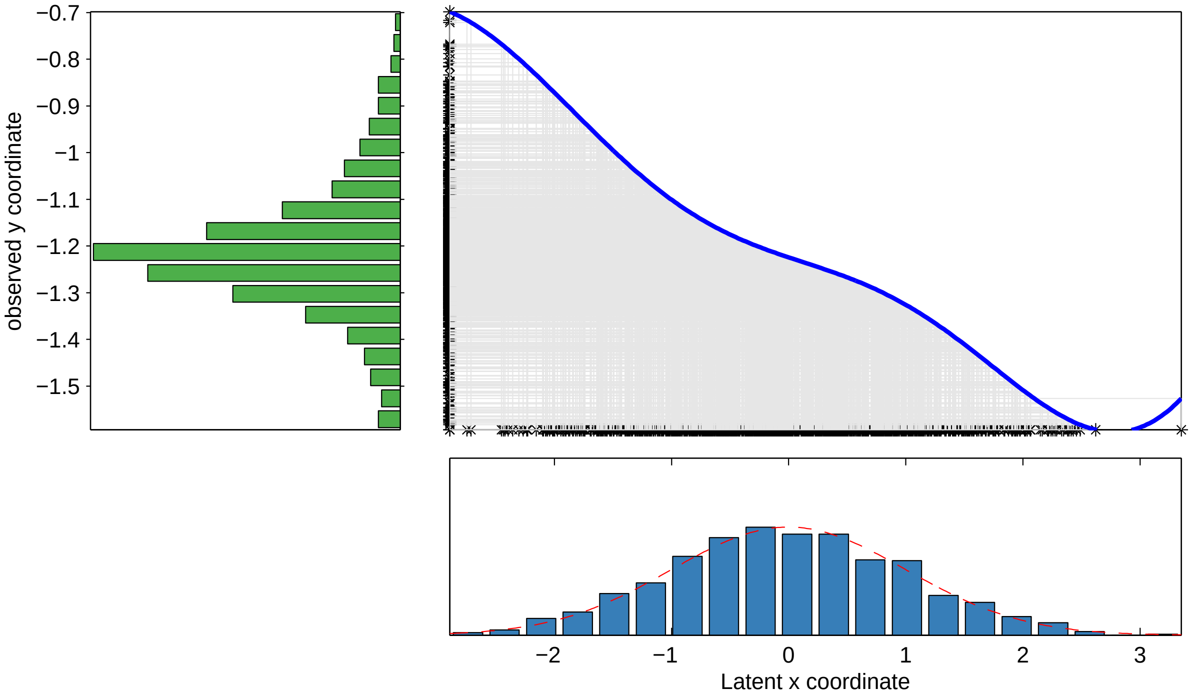
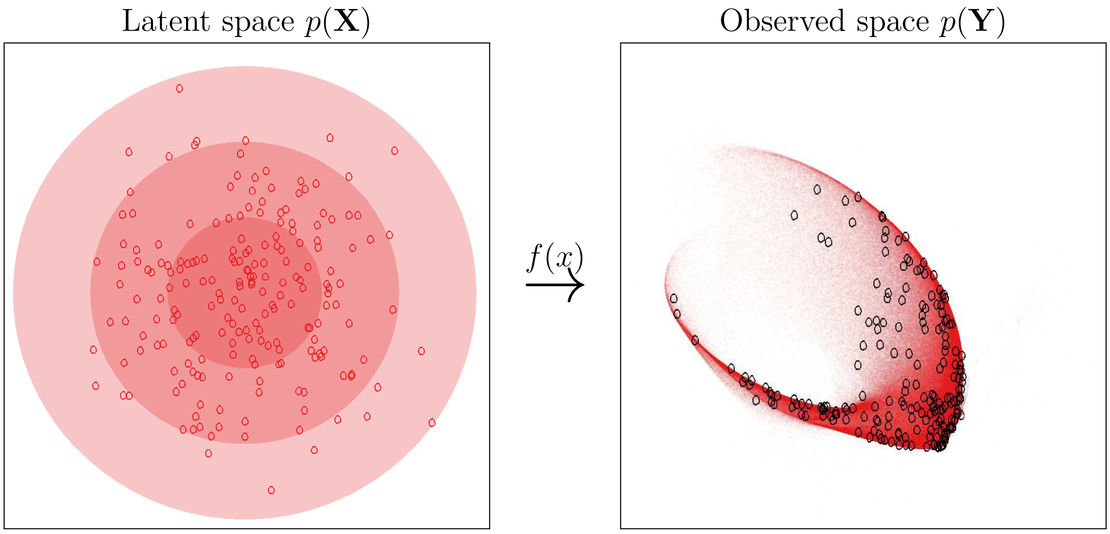


Fig. 1.2 A visual representation of the Gaussian process latent variable model. Bottom: density and samples from a 1D Gaussian, specifying the distribution $p(\mathbf{X})$ in the latent space. Top Right: A function drawn from a GP prior. Left: A nonparametric density defined by warping the latent density through the function drawn from a GP prior.

Chapter ?? describes, in detail, the ways in which different sorts of structure can be introduced into a GP model through the kernel.

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. Combined with


Latent space $p(\mathbf{X})$

Observed space $p(\mathbf{Y})$

Fig. 1.3 A visual representation of the Gaussian process latent variable model. Left: Isocontours and samples from a 2D Gaussian, specifying the distribution $p(\mathbf{X})$ in the latent space. Right: Density and samples from a nonparametric density defined by warping the latent density through a function drawn from a GP prior.

1.5 Contributions

My doctorate was made possible, and enjoyable by the company of the many co-authors I was fortunate to work with. In this section, I detail the novel contributions of this thesis, and attempt to give proper credit to my tireless co-authors.

Grammar on topologies Chapter **??** contains a section on describes, in detail, the ways in which different sorts of structure can be introduced into a GP model through the kernel.

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. Combined with