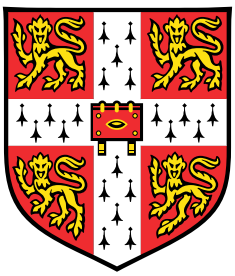
Efficient Deterministic Approximate Bayesian Inference for Gaussian Process models



Thang Duc Bui

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Trinity College

September 2017

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Thang Duc Bui
September 2017

Acknowledgements

First, I would like to thank my supervisor Richard E. Turner, for his guidance through the last four years. I still vividly remember our first, probably *ten*, meetings in which Rich patiently taught me about Gaussian processes and the FITC approximation, or the times Rich carefully went through the slides of my research talks and made sure he was at the talks to listen and give me feedback about my presentations afterwards. Rich has taught me how to think, how to do research to the highest standards of rigour, and how to communicate ideas clearly. It has been an intellectually rewarding experience to know and work with him.

While my name may be alone on the front cover of this thesis, I am by no means its sole contributor. I have been fortunate to have worked with many talented collaborators – Daniel Hernández-Lobato, José Miguel Hernández-Lobato, Yingzhen Li, Cuong Nguyen, Mark Rowland, Felipe Tobar, Josiah Yan, and of course Richard E. Turner – who deserve much of the credit. I would like to thank Nilesh Tripuraneni for many interesting discussions and his helpful feedback on the thesis. I would also like to thank my examiners, Carl Rasmussen and Manfred Opper, for an enjoyable viva and the contructive feedback. Of course, any remaining errors are my own.

I am also grateful to the Computational and Biological Learning Lab for its fantastic academic environment.

On a personal note, I would like to thank my family, particularly Bố Đá, Mẹ Vĩnh, Ba Chính, and Mẹ Thu, who have given me every opportunity.

Finally, there is nothing I could do to sufficiently express my gratitude to my wife, Mai, for her love, support, and patience. Not a single word in this thesis would exist without her.

Abstract

Gaussian processes are powerful nonparametric distributions over continuous functions that have become a standard tool in modern probabilistic machine learning. However, the applicability of Gaussian processes in the large-data regime and in hierarchical probabilistic models is severely limited by analytic and computational intractabilities. It is, therefore, important to develop practical approximate inference and learning algorithms that can address these challenges. To this end, this dissertation provides a comprehensive and unifying perspective of pseudo-point based deterministic approximate Bayesian learning for a wide variety of Gaussian process models, which connects previously disparate literature, greatly extends them and allows new state-of-the-art approximations to emerge.

We start by building a posterior approximation framework based on Power-Expectation Propagation for Gaussian process regression and classification. This framework relies on a structured approximate Gaussian process posterior based on a small number of pseudo-points, which is judiciously chosen to summarise the actual data and enable tractable and efficient inference and hyperparameter learning. Many existing sparse approximations are recovered as special cases of this framework, and can now be understood as performing approximate posterior inference using a common approximate posterior. Critically, extensive empirical evidence suggests that new approximation methods arisen from this unifying perspective outperform existing approaches in many real-world regression and classification tasks.

We explore the extensions of this framework to Gaussian process state space models, Gaussian process latent variable models and deep Gaussian processes, which also unify many recently developed approximation schemes for these models. Several mean-field and structured approximate posterior families for the hidden variables in these models are studied. We also discuss several methods for approximate uncertainty propagation in recurrent and deep architectures based on Gaussian projection, linearisation, and simple Monte Carlo. The benefit of the unified inference and learning frameworks for these models are illustrated in a variety of real-world state-space modelling and regression tasks.

Table of contents

Chapter 1

Introduction

This chapter aims to set the context for the remainder of this thesis. Several concepts central to this thesis such as Bayesian nonparametrics and Gaussian process regression are briefly introduced.

1.1 Probabilistic machine learning

Probabilistic modelling is a cornerstone of modern machine learning toolkits. It provides a principled framework for making coherent inferences, learning from observations and handling uncertainty, through the language of probability theory. A probabilistic model uses probability distributions to define the subjective belief or uncertainty of unknown quantities in the model via the *prior*, and how they are related to the observed data via the *likelihood*. Probabilistic inference, or Bayesian inference, then turns the prior belief into a posterior probability distribution of the unknown variables (the *posterior*), representing the belief about the unknown variables upon observing the data. As an example, probabilistic inference in a parametric model with a finite-dimensional parameter θ and observed data $\mathcal{D}$ allows us to obtain the posterior $p(\theta|\mathcal{D})$ from the prior $p(\theta)$ and the likelihood $p(\mathcal{D}|\theta)$ by using,

$$p(\theta|\mathcal{D}) = \frac{p(\theta)p(\mathcal{D}|\theta)}{p(\mathcal{D})}, \tag{1.1}$$

where $p(\mathcal{D})$ is the marginal likelihood of the model or the model hyperparameters,

$$p(\mathcal{D}) = \int p(\theta)p(\mathcal{D}|\theta)\mathrm{d}\theta. \tag{1.2}$$

Unlike non-probabilistic schemes which produce a single best parameter estimate, Bayesian inference gives a probability density over θ, that is, there are many (potentially infinite) parameter values that are plausible given the observed data, but some are more plausible than others. This posterior object can then be used as the prior when new data arrive, or to

predict unseen data $\mathcal{D}^*$,

$$p(\mathcal{D}^*|\mathcal{D}) = \int p(\mathcal{D}^*|\theta)p(\theta|\mathcal{D})\mathrm{d}\theta, \tag{1.3}$$

where we assume $\mathcal{D}$ and $\mathcal{D}^*$ are conditionally independent given θ. For more comprehensive reviews of Bayesian inference and its application to data analysis and machine learning, see Jaynes (2003); Gelman et al. (2014a); Ghahramani (2013, 2015).

Despite being conceptually and intuitively simple, exact Bayesian inference is often computationally and analytically intractable. The intractability can come from one or multiple sources including non-conjugacy, large data set size, and high dimensional parameter space. There are, however, many approximation schemes that can give asymptotically exact solutions such as Markov Chain Monte Carlo methods, or approximate solutions such as Laplace's method, variational free-energy method, and expectation propagation (e.g. MacKay, 2003; Neal, 2011; Jordan et al., 1999; Wainwright and Jordan, 2008; Minka, 2001b). Developing rich, accurate and general approximate Bayesian inference and learning methods for many probabilistic models is an active research area. Falling under this research theme, this thesis develops a series of generic deterministic approximation methods based on power expectation propagation (Minka, 2004) for a variety of Gaussian process probabilistic models.

1.2 Nonparametric models

Defining a suitable and flexible model for the data at hand is key to good performance, regardless of how the inference process is carried out (probabilistic or non-probabilistic).[1] In the case of Bayesian inference for a model with a finite-dimensional parameter space above, the flexibility of the model is constrained by the capacity of the parameter. In particular, the parameter is of a fixed size that is independent of the training set, and it is the *bottleneck* between the training the training data and the test data, as shown in eq. (1.3). This is arguably inflexible as the training size can grow and the fixed size parameter can then become a limited *information channel* from the data to the prediction (Ghahramani, 2013).

There are several strategies to expand the model capacity, including i. build a parametric model, but with a massive number of parameters (e.g. a large neural network with millions of weights) and ii. explicitly build a nonparametric module in the model, that is a component with an infinite-dimensional parameter. In this section and what follows, we focus on the approach of building nonparametric models. One question naturally arises, which is how we can represent and manipulate such a *big* parameter on a computer. Fortunately, this parameter can be mathematically represented as a function and, as a result, inference is now

[1] There is an issue with overfitting when the model is too flexible, i.e. the model that copies and hence perfectly explains the training data but does not generalise to test data, but the Bayesian paradigm is often robust to such behaviour, since the parameter(s) are averaged out. The discussion about overfitting, underfitting, model averaging and generalisation is, however, beyond the scope of this introduction.

performed over the function space instead of a finite-dimensional vector space. Additionally, in non-parametric models, the complexity actually grows with the size of data available and does not require an infinite amount of computation.

Combining the Bayesian paradigm with nonparametric modelling, which is often called Bayesian nonparametrics, has been a flourishing research area of machine learning and statistics (Hjort et al., 2010; Orbanz and Teh, 2011; Ghahramani, 2013). Several prime examples are Gaussian processes, (hierarchical) Dirichlet processes, and Indian Buffet Processes. In this thesis, we consider Gaussian processes (GPs) as a nonparametric component in building regression, classification, state space (recurrent), and hierarchical (deep) models. An example how GPs might be used for regression is shown in fig. 1.1. In this case, the infinite-dimension object is the non-linear function, mapping from one-dimensional input x to one-dimensional output y. We show several functions drawn from the GP prior and several functions drawn from the GP posterior upon observing three datapoints. Note that there are infinitely many functions that are consistent with the observed data.

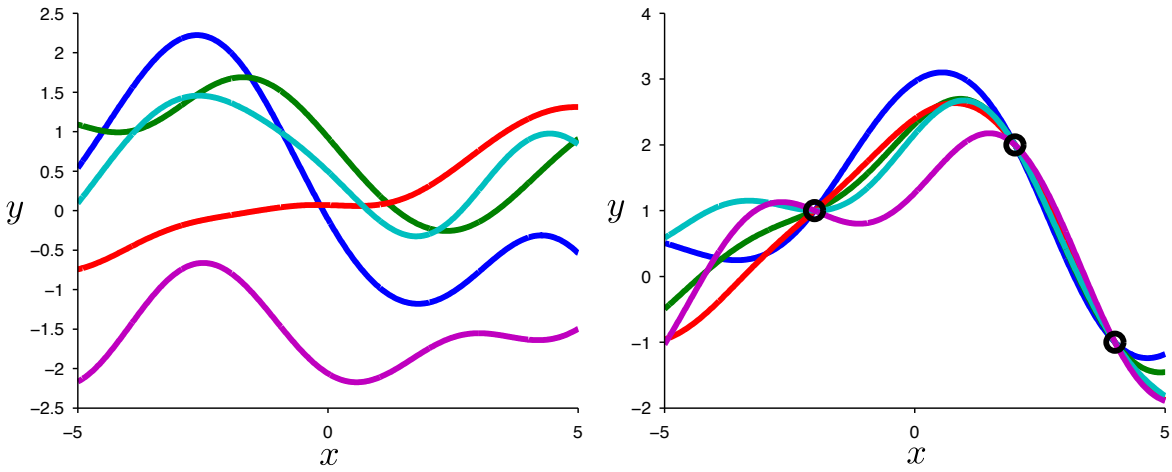


Fig. 1.1 Left: functions drawn from a GP prior with a zero mean function and an exponentiated quadratic covariance function, Right: functions drawn from the posterior GP, after conditioning on several observations (shown as black circles).

1.3 A refresher on Gaussian process regression

As the remainder of this thesis will dive deeper into how to use GPs as a nonparametric component of probabilistic models and how to perform inference in such models, we attempt to give a brief introduction to GPs and define some important terminology. Readers who are familiar with GPs might skip this section, while interested readers who want to learn more about GPs after reading this section might want to consult the excellent monograph by Rasmussen and Williams (2005).

In this section, we consider the function-space view of GPs. In particular, we consider a non-linear mapping, $f(\mathbf{x})$, from (potentially high-dimensional and structured) input, $\mathbf{x}$, to one-dimensional real-valued output. The function value, $f_i \coloneqq f(\mathbf{x}_i)$, at a particular input, $\mathbf{x}_i$, is a random variable and a GP is a collection of infinite random variables, any finite number of which have a joint Gaussian distribution (Rasmussen and Williams, 2005). A GP is fully specified by its mean function, $m_\theta(\mathbf{x})$, and its covariance function or kernel, $k_\theta(\mathbf{x}, \mathbf{x}')$,

$$f(\mathbf{x}) \sim \mathcal{GP}(m_\theta(\mathbf{x}), k_\theta(\mathbf{x}, \mathbf{x}')) \tag{1.4}$$

where $\mathbf{x}$ and $\mathbf{x}'$ denote the input locations at which the function $f(\cdot)$ is evaluated, and θ is a small set of hyperparameters of the mean and covariance functions. Following the definition of GPs above, we can write down the distribution of a finite collection of function values as follows,

$$p\left(\begin{bmatrix} f_i \\ f_j \\ f_r \\ \vdots \end{bmatrix} \middle| \theta\right) = \mathcal{N}\left(\begin{bmatrix} f_i \\ f_j \\ f_r \\ \vdots \end{bmatrix}; \begin{bmatrix} m_{f_i} \\ m_{f_j} \\ m_{f_r} \\ \vdots \end{bmatrix}, \begin{bmatrix} k_{f_i f_i} & k_{f_i f_j} & k_{f_i f_r} & \cdots \\ k_{f_j f_i} & k_{f_j f_j} & k_{f_j f_r} & \cdots \\ k_{f_r f_i} & k_{f_r f_j} & k_{f_r f_r} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}\right). \tag{1.5}$$

where the entries of the mean vector in the distribution above are the values of the mean function evaluated at the input values, $m_{f_i} = m_\theta(\mathbf{x}_i)$, and similarly, the entries of the covariance matrix are the values of covariance function evaluated at pairs of input values, $k_{f_i f_j} = k_\theta(\mathbf{x}_i, \mathbf{x}_j)$. Note that we use the function values, e.g. f_i, as subscripts to show that m and k are the mean and covariance of the distribution over the function values. Typically, the mean function is assumed to be a zero since the *prior* knowledge about the function $f(\cdot)$ can be encapsulated in the form of the covariance function and its hyperparameters θ. The family of the covariance function is selected based on prior knowledge about the function, e.g. smooth, rough, wiggly or periodic. A popular covariance function is the exponentiated quadratic or squared exponential kernel with automatic relevance determination,

$$k_\theta(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{(x_d - x'_d)^2}{l_d^2}\right), \tag{1.6}$$

where l_d is the private lengthscale for the d-th input dimension and σ_f^2 is the kernel variance, and in this case $\theta = (\{l_d\}_{d=1}^D, \sigma_f)$ are the kernel hyperparameters.

Gaussian processes are best known perhaps for their application to regression and classification. We briefly summarise the formulation for the regression case. Suppose we have a training set comprising N D-dimensional input vectors $\{\mathbf{x}_n\}_{n=1}^N$ and corresponding real valued scalar observations $\{y_n\}_{n=1}^N$. Typical regression models assume that each observation y_n is formed from an unknown function $f(\cdot)$, evaluated at input $\mathbf{x}_n$, which is corrupted by

additive, independent Gaussian noise,

$$y_n = f(\mathbf{x}_n) + \epsilon_n, \tag{1.7}$$

where $p(\epsilon_n) = \mathcal{N}(\epsilon_n; 0, \sigma_n^2)$. A GP as defined above can be used to specify a prior over function $f(\cdot)$ and the corresponding probabilistic model is as follows,

$$f|\theta \sim \mathcal{GP}(0, k_\theta(\cdot,\cdot)), \tag{1.8}$$

$$p(\mathbf{y}|f, \sigma_n^2) = \prod_n \mathcal{N}(y_n; f(\mathbf{x}_n), \sigma_n^2), \tag{1.9}$$

where $\mathbf{y}$ is a vector comprising of all training outputs.

In regression problems, the tasks typically involve predicting the function value f_* at some unseen input $\mathbf{x}_*$. This task also bears many names such as interpolation, forecasting, or missing data imputation. When $\mathbf{x}_*$ is at one of the training input points, the task now is to denoise the observation to estimate the true underlying function value f_*. The generative model described above provides a framework to obtain the predictive distribution of the target function values. Because the joint distribution between the training observations and the (test) latent functions is a multivariate normal distribution, the posterior can be obtained using the conditional Gaussian distribution property. It is also a GP with the following mean and covariance functions,

$$\hat{m}(\mathbf{x}) = k_{f\mathbf{f}}(\mathbf{K}_{\mathbf{ff}} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}, \tag{1.10}$$

$$\hat{k}(\mathbf{x}, \mathbf{x}') = k_{ff'} - k_{f\mathbf{f}}(\mathbf{K}_{\mathbf{ff}} + \sigma_n^2\mathbf{I})^{-1}k_{\mathbf{f}f'}, \tag{1.11}$$

where $\mathbf{f}$ is a vector whose elements are the function values at the training inputs, $\{f_n \coloneqq f(\mathbf{x}_n)\}_{n=1}^N$, and $k_{f\mathbf{f}}$ and $\mathbf{K}_{\mathbf{ff}}$ are the covariance matrices between the test function values and training function values, and the training function values and themselves, respectively. This means that the predictive distribution of functions at unseen inputs or the denoising distribution at training inputs are Gaussian, specified by evaluating the posterior mean and covariance functions in eqs. (1.10) and (1.11) at the corresponding test inputs. Figure 1.1 shows some functions drawn from a GP prior and the GP posterior after conditioning on some training points.

The procedure above allows us to obtain the GP posterior and make predictions at test inputs, with a *fixed* set of kernel hyperparameters θ and noise variance σ_n^2. However, these are often not known in advance and are usually difficult to select manually. The fully Bayesian approach can be used, that is, one can specify prior over the hyperparameters and obtain the joint posterior $p(\mathbf{f}, \theta, \sigma_n^2|\mathbf{y})$. However, this procedure is often not analytically available and requires approximation techniques such as MCMC. It is, therefore, a common practice to only obtain one set of the hyperparameters $\{\theta, \sigma_n^2\}$ by maximising the *marginal likelihood* of

the hyperparameters and use them to obtain the posterior or for prediction. In the regression case, the marginal likelihood can be conveniently obtained in closed-form and the log of which is,

$$\begin{aligned}\mathcal{L}(\theta, \sigma_n^2) &= \log p(\mathbf{y}|\theta, \sigma_n^2) \\ &= \log \int p(\mathbf{y}|f, \sigma_n^2) p(f|\theta) \mathrm{d}f \\ &= \log \mathcal{N}(\mathbf{y}; \mathbf{0}, \mathbf{K_{ff}} + \sigma_n^2 \mathbf{I}) \\ &= -\frac{1}{2}\mathbf{y}^\intercal (\mathbf{K_{ff}} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y} - \frac{1}{2}\log|\mathbf{K_{ff}} + \sigma_n^2 \mathbf{I}| - \frac{n}{2}\log 2\pi. \end{aligned} \tag{1.12}$$

The first term in eq. (1.12) is the only term that touches the observed outputs and hence controls how well the model fits the data. The second term controls the complexity of the model. As a result, optimising the above objective will result in hyperparameters that balance the data-fit quality of the model and the model complexity (Rasmussen and Williams, 2005). Therefore, this procedure is often said to be robust to overfitting, though the optimisation routine can get stuck at local maxima. Additionally, the (log) marginal likelihood can be used to select a kernel family that is appropriate for the data at hand (Duvenaud et al., 2013).

The computational complexity of hyperparameter learning and prediction is largely dominated by the cost to invert the matrix $\mathbf{K_{ff}} + \sigma_n^2 \mathbf{I}$, as seen in eqs. (1.10) to (1.12). This inversion costs $\mathcal{O}(N^3)$ and the learning requires repeating this operation multiple times. Once this has been performed, a subsequent prediction at a test input can be made in $\mathcal{O}(N^2)$.

1.4 Thesis overview

Deploying Gaussian processes in practice is challenging due to the cumbersome computational complexity (as described in the last section), and the analytical intractability when the posterior or marginal densities cannot be obtained in closed-form. This thesis is concerned with addressing these two challenges by developing novel deterministic approximate Bayesian schemes with structured posterior approximations. In detail,

- Chapter 2 develops a novel unifying framework of sparse approximations for Gaussian process regression and classification. The new framework unifies many existing approximations and develops new approximations, by viewing them as performing approximate Bayesian inference using power expectation propagation. This view is a complementary and orthogonal perspective to the popular approximate model view of sparse GPs of Quiñonero-Candela and Rasmussen (2005). This chapter is a joint work with Josiah Yan and Richard E. Turner.

- Chapter 3 extends the unifying framework in chapter 2 to GP latent variable and state space models. This chapter provides a comprehensive review of existing literature

on approximate inference for these models and develops novel inference and learning algorithm based on (approximate) power expectation propagation. In the case of the GP state space model, we discuss several mean-field and structured approximations for the hidden variables, and three approximate uncertainty propagation techniques based on linearisation, Gaussian projection, and simple Monte Carlo. This chapter is a joint work with Richard E. Turner.

- Chapter 4 unifies and greatly extends several structured approximations for inference and learning in deep GPs. The earlier version of this chapter, published in (Bui et al., 2016), is a joint work with José Miguel Hernández-Lobato, Yingzhen Li, Daniel Hernández-Lobato and Richard E. Turner. This chapter, however, significantly extends the earlier work in light of the results in chapter 2. This chapter also clearly connects many recently developed approaches for inference and learning in deep GPs, that were previously understood to be very different, by viewing them as special cases of power expectation propagation using a common structured posterior approximation.

While each of these chapters has been written such that a chapter can be read fairly independently of other chapters, one would be better served reading chapter 2 first before chapters 3 and 4. We conclude and suggest several future directions in chapter 5.

Chapter 2

Sparse approximations for Gaussian process regression and classification

This chapter is based on the JMLR paper, "A Unifying Framework for Sparse Gaussian Process Approximation using Power Expectation Propagation", which is a joint work with Josiah Yan and Richard E. Turner. Section 2.6 is new and has not been discussed in the paper.

2.1 Introduction

Gaussian processes (GPs) are powerful nonparametric distributions over continuous functions that are routinely deployed in probabilistic modelling for applications including regression and classification (Rasmussen and Williams, 2005), representation learning (Lawrence, 2005), state space modelling (Wang et al., 2005), active learning (Houlsby et al., 2011), reinforcement learning (Deisenroth, 2010), black-box optimisation (Snoek et al., 2012), and numerical methods (Mahsereci and Hennig, 2015). GPs have many elegant theoretical properties, but their use in probabilistic modelling is greatly hindered by analytic and computational intractabilities. A large research effort has been directed at this fundamental problem, resulting in the development of a plethora of sparse approximation methods that can sidestep these intractabilities (Csató, 2002; Csató and Opper, 2002; Schwaighofer and Tresp, 2002; Seeger et al., 2003; Quiñonero-Candela and Rasmussen, 2005; Snelson and Ghahramani, 2006; Snelson, 2007; Naish-Guzman and Holden, 2007; Titsias, 2009; Figueiras-Vidal and Lázaro-Gredilla, 2009; Álvarez et al., 2010; Qi et al., 2010; Bui and Turner, 2014; Frigola et al., 2014; McHutchon, 2014; Hensman et al., 2015; Hernández-Lobato and Hernández-Lobato, 2016; Matthews et al., 2016).

This chapter develops a general sparse approximate inference framework based upon Power Expectation Propagation (PEP) (Minka, 2004) that unifies many of these approximations, extends them significantly, and provides improvements in practical settings. In this way, the

chapter provides a complementary perspective to the seminal review of Quiñonero-Candela and Rasmussen (2005), viewing sparse approximations through the lens of approximate *inference*, rather than approximate *generative models*.

The chapter begins by reviewing several frameworks for sparse approximation focussing on the GP regression and classification setting (section 2.2). It then lays out the new unifying framework and the relationship to existing techniques (section 2.3). Readers whose focus is to understand the new framework might want to move directly to this section. Finally, a thorough experimental evaluation is presented in section 2.4.

2.2 Pseudo-point approximations for GP regression and classification

This section provides a concise introduction to GP regression and classification and then reviews several pseudo-point based sparse approximation schemes for these models. For simplicity, we first consider a supervised learning setting in which the training set comprises N D-dimensional input and scalar output pairs $\{\mathbf{x}_n, y_n\}_{n=1}^N$ and the goal is to produce probabilistic predictions for the outputs corresponding to novel inputs. A non-linear function, $f(\mathbf{x})$, can be used to parameterise the probabilistic mapping between inputs and outputs, $p(y_n|f, \mathbf{x}_n, \theta)$. Typical choices for the probabilistic mapping are Gaussian $p(y_n|f, \mathbf{x}_n, \theta) = \mathcal{N}(y_n; f(\mathbf{x}_n), \sigma_y^2)$ for the regression setting ($y_n \in \mathbb{R}$) and Bernoulli $p(y_n|f, \mathbf{x}_n, \theta) = \mathcal{B}(y_n; \Phi(f(\mathbf{x}_n)))$ with a sigmoidal link function $\Phi(f)$ for the binary classification setting ($y_n \in \{0, 1\}$). Whilst it is possible to specify the non-linear function f via an explicit parametric form, a more flexible and elegant approach employs a GP prior over the functions directly, $p(f|\theta) = \mathcal{GP}(f; 0, k_\theta(\cdot, \cdot))$, here assumed without loss of generality to have a zero mean-function and a covariance function $k_\theta(\mathbf{x}, \mathbf{x}')$. This class of probabilistic models has a joint distribution

$$p(f, \mathbf{y}|\theta) = p(f|\theta) \prod_{n=1}^{N} p(y_n|f(\mathbf{x}_n), \theta) \tag{2.1}$$

where we have collected the observations into the vector $\mathbf{y}$ and suppressed the inputs on the left hand side to lighten the notation.

This model class contains two potential sources of intractability. First, the possibly non-linear likelihood function can introduce analytic intractabilities that require approximation. Second, the GP prior entails an $\mathcal{O}(N^3)$ complexity that is computationally intractable for many practical problems. These two types of intractability can be handled by combining standard approximate inference methods with pseudo-point approximations that summarise the full Gaussian process via M pseudo datapoints leading to an $\mathcal{O}(NM^2)$ cost. The main approaches of this sort can be characterised in terms of two parallel frameworks that are described in the following sections.

2.2.1 Sparse GP approximation via approximate generative models

The first framework begins by constructing a new generative model that is similar to the original, so that inference in the new model might be expected to produce similar results, but which has a special structure that supports efficient computation. Typically, this approach involves approximating the Gaussian process prior as it is the origin of the cubic cost. If there are analytic intractabilities in the approximate model, as will be the case in e.g. classification or state-space models, then these will require approximate inference to be performed in the approximate model.

The seminal review by Quiñonero-Candela and Rasmussen (Quiñonero-Candela and Rasmussen, 2005) reinterprets a family of approximations in terms of this unifying framework. The GP prior is approximated by identifying a small set of $M \leq N$ pseudo-points $\mathbf{u}$, here assumed to be disjoint from the training function values $\mathbf{f}$ so that $f = \{\mathbf{u}, \mathbf{f}, f_{\neq \mathbf{u},\mathbf{f}}\}$. The GP prior is then decomposed using the product rule

$$p(f|\theta) = p(\mathbf{u}|\theta)p(\mathbf{f}|\mathbf{u},\theta)p(f_{\neq \mathbf{u},\mathbf{f}}|\mathbf{f},\mathbf{u},\theta). \tag{2.2}$$

Of central interest is the relationship between the pseudo-points and the training function values $p(\mathbf{f}|\mathbf{u},\theta) = \mathcal{N}(\mathbf{f}; \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u}, \mathbf{D_{ff}})$ where $\mathbf{D_{ff}} = \mathbf{K_{ff}} - \mathbf{Q_{ff}}$ and $\mathbf{Q_{ff}} = \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}$. Here we have introduced matrices corresponding to the covariance function's evaluation at the pseudo-input locations $\{\mathbf{z}_m\}_{m=1}^M$, so that $[\mathbf{K_{uu}}]_{mm'} = k_\theta(\mathbf{z}_m, \mathbf{z}_{m'})$ and similarly for the covariance between the pseudo-input and data locations $[\mathbf{K_{uf}}]_{mn} = k_\theta(\mathbf{z}_m, \mathbf{x}_n)$. Importantly, this term saddles learning with a cubic complexity cost. Computationally efficient approximations can be constructed by simplifying these dependencies between the pseudo-points and the data function values $q(\mathbf{f}|\mathbf{u},\theta) \approx p(\mathbf{f}|\mathbf{u},\theta)$. In order to benefit from these efficiencies at prediction time as well, a second approximation is made whereby the pseudo-points form a bottleneck between the data function values and test function values $p(f_{\neq \mathbf{u},\mathbf{f}}|\mathbf{u},\theta) \approx p(f_{\neq \mathbf{u},\mathbf{f}}|\mathbf{f},\mathbf{u},\theta)$. Together, the two approximations result in an approximate prior process,

$$q(f|\theta) = p(\mathbf{u}|\theta)q(\mathbf{f}|\mathbf{u},\theta)p(f_{\neq \mathbf{u},\mathbf{f}}|\mathbf{u},\theta). \tag{2.3}$$

We can now compactly summarise a number of previous approaches to GP approximation as special cases of the choice

$$q(\mathbf{f}|\mathbf{u},\theta) = \prod_{b=1}^{B} \mathcal{N}(\mathbf{f}_b; \mathbf{K}_{\mathbf{f}_b,\mathbf{u}}\mathbf{K_{uu}^{-1}}\mathbf{u}, \alpha\mathbf{D}_{\mathbf{f}_b,\mathbf{f}_b}) \tag{2.4}$$

where b indexes B disjoint blocks of data-function values. The Deterministic Training Conditional (DTC) approximation uses $\alpha \to 0$; the Fully Independent Training Conditional (FITC) approximation uses $\alpha = 1$ and $B = N$; the Partially Independent Training Conditional

(PITC) approximation uses $\alpha = 1$ (Quiñonero-Candela and Rasmussen, 2005; Schwaighofer and Tresp, 2002).

In a moment we will consider inference in the modified models, before doing so we note that it is possible to construct more flexible modified prior processes using the inter-domain approach that places the pseudo-points in a different domain from the data, defined by a linear integral transform $g(z) = \int w(z, z')f(z')\mathrm{d}z'$. Here the window $w(z, z')$ might be a Gaussian blur or a wavelet transform. The pseudo-points are now placed in the new domain $g = \{\mathbf{u}, \mathbf{g}_{\neq \mathbf{u}}\}$ where they induce a potentially more flexible Gaussian process in the old domain f through the linear transform (see Figueiras-Vidal and Lázaro-Gredilla (2009) for FITC). The expressions in this section still hold, but the covariance matrices involving pseudo-points are modified to take account of the transform,

$$[\mathbf{K_{uu}}]_{mm'} = \int w(\mathbf{z}_m, \mathbf{z})k_\theta(\mathbf{z}, \mathbf{z}')w(\mathbf{z}', \mathbf{z}_{m'})\mathbf{dzdz}', \quad [\mathbf{K_{uf}}]_{mn} = \int w(\mathbf{z}_m, \mathbf{z})k_\theta(\mathbf{z}, \mathbf{x}_n)\mathbf{dz}. \quad (2.5)$$

Having specified modified prior processes, these can be combined with the original likelihood function to produce a new generative models. In the case of point-wise likelihoods, we have

$$q(\mathbf{y}, f|\theta) = q(f|\theta)\prod_{n=1}^{N} p(y_n|f(\mathbf{x}_n), \theta). \quad (2.6)$$

Inference and learning can now be performed using the modified model using standard techniques. Due to the form of the new prior process, the computational complexity is $\mathcal{O}(NM^2)$ (for testing, N becomes the number of test datapoints, assuming dependencies between the test-points are not computed).[1] For example, in the case of regression, the posterior distribution over function values f (necessary for inference and prediction) has a simple analytic form

$$q(f|\mathbf{y}, \theta) = \mathcal{GP}(f; \mu_{f|\mathbf{y}}, \Sigma_{f|\mathbf{y}}), \quad \mu_{f|\mathbf{y}} = \mathbf{Q}_{\mathbf{f}\mathbf{f}}\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y}, \quad \Sigma_{f|\mathbf{y}} = \mathbf{K}_{ff} - \mathbf{Q}_{\mathbf{f}\mathbf{f}}\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{Q}_{\mathbf{f}f} \quad (2.7)$$

where $\overline{\mathbf{K}}_{\mathbf{ff}} = \mathbf{Q}_{\mathbf{ff}} + \mathrm{blkdiag}(\{\alpha_b \mathbf{D}_{\mathbf{f}_b\mathbf{f}_b}\}_{b=1}^{B}) + \sigma_y^2\mathrm{I}$ and blkdiag builds a block-diagonal matrix from its inputs. One way to understand the origin of the computational gains is that the new generative model corresponds to a form of factor analysis in which the M pseudo-points determine the N function values at the observed data (as well as at potential test locations) via a linear Gaussian relationship. This results in low rank (sparse) structure in $\overline{\mathbf{K}}_{\mathbf{ff}}$ that can be exploited through the matrix inversion and determinant lemmas. In the case of regression, the new model's marginal likelihood also has an analytic form that allows the

[1] It is assumed that the maximum size of the blocks is not greater than the number of pseudo-points $\dim(\mathbf{f}_b) \leq M$.

hyperparameters, θ, to be learned via optimisation

$$\log q(\mathbf{y}|\theta) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\overline{\mathbf{K}}_{\mathbf{ff}}| - \frac{1}{2}\mathbf{y}^\intercal\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y}. \tag{2.8}$$

The approximate generative model framework has attractive properties. The cost of inference, learning, and prediction has been reduced from $\mathcal{O}(N^3)$ to $\mathcal{O}(NM^2)$ and in many cases, accuracy can be maintained with a relatively small number of pseudo-points. The pseudo-point input locations can be optimised by maximising the new model's marginal likelihood (Snelson and Ghahramani, 2006). When $M = N$ and the pseudo-points and observed data inputs coincide, then FITC and PITC are exact which appears reassuring. However, the framework is philosophically challenging as the elegant separation of model and (approximate) inference has been lost. Are we allowed in an online inference setting, for example, to add new pseudo-points as more data are acquired and the complexity of the underlying function is revealed? This seems sensible, but effectively changes the modelling assumptions as more data are seen. Devout Bayesians might then demand that we perform model averaging for coherence. Similarly, if the pseudo-input locations are optimised, the principled non-parametric model has suddenly acquired MD parameters and with them all of the concomitant issues of parametric models including overfitting and optimisation difficulties (Bauer et al., 2016). As the pseudo-inputs are considered part of the model, the Bayesians might then suggest that we place priors over the pseudo-inputs and to perform full-blown probabilistic inference over them.

These awkward questions arise because the generative modelling interpretation of pseudo-data entangles the assumptions made about the data with the approximations required to perform inference. Instead, the modelling assumptions (which encapsulate prior understanding of the data) should remain decoupled from inferential assumptions (which leverage structure in the posterior for tractability). In this way, pseudo-data should be introduced when we seek to perform computationally efficient approximate inference, leaving the modelling assumptions unchanged as we refine and improve approximate inference. Indeed, even under the generative modelling perspective, for analytically intractable likelihood functions, an additional approximate inference step is required, begging the question; why not handle computational and analytic intractabilities together at inference time?

2.2.2 Sparse GP approximation via approximate inference: VFE

The approximate generative model framework for constructing sparse approximations is philosophically troubling. In addition, learning pseudo-point input locations via optimisation of the model likelihood can perform poorly e.g. for DTC it is prone to overfitting even for $M \ll N$ (Titsias, 2009). This motivates a more direct approach that commits to the true generative model and performs all of the necessary approximation at inference time.

Perhaps the most well known approach in this vein is Titsias's beautiful sparse variational free-energy (VFE) method (Titsias, 2009). The original presentation of this work employs finite variable sets and an augmentation trick that arguably obscures its full elegance. Here instead we follow Matthews et al. (2016) and lower bound the marginal likelihood using a distribution $q(f)$ over the entire infinite dimensional function,

$$\log p(\mathbf{y}|\theta) = \log \int p(\mathbf{y}, f|\theta) \mathrm{d}f \geq \int q(f) \log \frac{p(\mathbf{y}, f|\theta)}{q(f)} \mathrm{d}f = \mathbb{E}_{q(f)} \left[\log \frac{p(\mathbf{y}, f|\theta)}{q(f)} \right] = \mathcal{F}(q, \theta).$$

The VFE bound can be written as the difference between the model log-marginal likelihood and the KL divergence between the variational distribution and the true posterior $\mathcal{F}(q, \theta) = \log p(\mathbf{y}|\theta) - \mathrm{KL}(q(f)||p(f|\mathbf{y}, \theta))$. The bound is therefore saturated when $q(f) = p(f|\mathbf{y}, \theta)$, but this is intractable. Instead, pseudo-points are made explicit, $f = \{\mathbf{u}, f_{\neq \mathbf{u}}\}$, and an approximate posterior distribution used of the following form $q(f) = q(\mathbf{u}, f_{\neq \mathbf{u}}|\theta) = p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta) q(\mathbf{u})$. Under this approximation, the set of variables $f_{\neq \mathbf{u}}$ do not experience the data directly, but rather only through the pseudo-points, as can be seen by comparison to the true posterior $p(f|\mathbf{y}, \theta) = p(f_{\neq \mathbf{u}}|\mathbf{y}, \mathbf{u}, \theta) p(\mathbf{u}|\mathbf{y}, \theta)$. Importantly, the form of the approximate posterior causes a cancellation of the prior conditional term, which gives rise to a bound with $\mathcal{O}(NM^2)$ complexity,

$$\begin{aligned} \mathcal{F}(q, \theta) &= \mathbb{E}_{q(f|\theta)} \left[\log \frac{p(\mathbf{y}|f, \theta) \cancel{p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)} p(\mathbf{u}|\theta)}{\cancel{p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)} q(\mathbf{u})} \right] \\ &= \sum_n \mathbb{E}_{q(f|\theta)} \left[\log p(y_n|\mathrm{f}_n, \theta) \right] - \mathrm{KL}(q(\mathbf{u})||p(\mathbf{u}|\theta)). \end{aligned}$$

For regression with Gaussian observation noise, the calculus of variations can be used to find the optimal approximate posterior Gaussian process over pseudo-data $q^{\mathrm{opt}}(f|\theta) = p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta) q^{\mathrm{opt}}(\mathbf{u})$ which has the form

$$q^{\mathrm{opt}}(f|\theta) = \mathcal{GP}(f; \mu_{f|\mathbf{y}}, \Sigma_{f|\mathbf{y}}), \quad \mu_{f|\mathbf{y}} = \mathbf{Q}_{f\mathbf{f}} \tilde{\mathbf{K}}_{\mathbf{ff}}^{-1} \mathbf{y}, \quad \Sigma_{f|\mathbf{y}} = \mathbf{K}_{ff} - \mathbf{Q}_{f\mathbf{f}} \tilde{\mathbf{K}}_{\mathbf{ff}}^{-1} \mathbf{Q}_{\mathbf{f}f} \tag{2.9}$$

where $\tilde{\mathbf{K}}_{\mathbf{ff}} = \mathbf{Q}_{\mathbf{ff}} + \sigma_y^2 \mathrm{I}$. This process is identical to that recovered when performing exact inference under the DTC approximate regression generative model (Titsias, 2009) (see eq. (2.7)). In fact, DTC was originally derived using the same KL argument (Csató, 2002; Csató et al., 2002; Seeger et al., 2003). However, this fact is not well-known in the literature, perhaps because these articles considered only the optimal approximate posterior and not the free-energy, and that the optimal approximate posterior was later reinterpreted as an exact posterior in an approximate model (Quiñonero-Candela and Rasmussen, 2005). The optimised free-energy is

$$\mathcal{F}(q^{\mathrm{opt}}, \theta) = -\frac{N}{2} \log(2\pi) - \frac{1}{2} \log |\tilde{\mathbf{K}}_{\mathbf{ff}}| - \frac{1}{2} \mathbf{y}^\intercal \tilde{\mathbf{K}}_{\mathbf{ff}}^{-1} \mathbf{y} - \frac{1}{2\sigma_y^2} \mathrm{trace}(\mathbf{K}_{\mathbf{ff}} - \mathbf{Q}_{\mathbf{ff}}). \tag{2.10}$$

Notice that the free-energy has an additional trace term as compared to the marginal likelihood obtained from the DTC generative model approach (see eq. (2.8) as $\alpha \to 0$). The trace term is proportional to the sum of the variances of the training function values given the pseudo-points, $p(\mathbf{f}|\mathbf{u})$, it thereby encourages pseudo-input locations that explain the observed data well. This term acts as a regulariser that prevents overfitting which plagues the generative model formulation of DTC.

The VFE approach can be extended to non-linear models including classification (Hensman et al., 2015), latent variable models (Titsias and Lawrence, 2010) and state space models (Frigola et al., 2014; McHutchon, 2014) by restricting $q(\mathbf{u})$ to be Gaussian and optimising its parameters. Indeed, this uncollapsed form of the bound can be beneficial in the context of regression too as it is amenable to stochastic optimisation (Hensman et al., 2013). Additional approximation is sometimes required to compute any remaining intractable non-linear integrals, but these are often low-dimensional. For example, when the likelihood depends on only one latent function value, as is typically the case for regression and classification, the bound requires only 1D integrals $\mathbb{E}_{q(f_n)}\left[\log p(y_n|\mathrm{f}_n, \theta)\right]$ that can be evaluated using quadrature (Hensman et al., 2015), for example.

The VFE approach can also be extended to employ inter-domain variables (Álvarez et al., 2010; Tobar et al., 2015; Matthews et al., 2016). The approach considers the augmented generative model $p(f, g|\theta)$ where to remind the reader the auxiliary process is defined by a linear integral transformation, $g(z) = \int w(z, z')f(z')\mathrm{d}z'$. Variational inference is now performed over both latent processes $q(f, g) = q(f, \mathbf{u}, g_{\neq \mathbf{u}}|\theta) = p(f, g_{\neq \mathbf{u}}|\mathbf{u}, \theta)q(\mathbf{u})$. Here the pseudo-data have been placed into the auxiliary process with the idea being that they can induce richer dependencies in the original domain that model the true posterior more accurately. In fact, if the linear integral transformation is parameterised then the transformation can be learned so that it approximates the posterior more accurately.

A key concept underpinning the VFE framework is that the pseudo-input locations (and the parameters of the inter-domain transformation, if employed) are purely parameters of the approximate posterior, hence the name 'variational parameters'. This distinction is important as it means, for example, that we are free to add pseudo-data as more structure is revealed the underlying function without altering the modelling assumptions (e.g. see Bui et al. (2017a) for an example in online inference). Moreover, since the pseudo-input locations are variational parameters, placing priors over them is unnecessary in this framework. Unlike the model parameters, optimisation of variational parameters is automatically protected from overfitting as the optimisation is minimising the KL divergence between the approximate posterior and the true posterior. Indeed, although the DTC posterior is recovered in the regression setting, as we have seen the free-energy is *not* equal to the log-marginal likelihood of the DTC generative model, containing an additional term that substantially improves the quality of the optimised pseudo-point input locations.

The fact that the form of the DTC approximation can be recovered from a direct approximate inference approach and that this new perspective leads to superior pseudo-input optimisation, raises the question; can this also be done for FITC and PITC?

2.2.3 Sparse GP approximation via approximate inference: EP

Expectation Propagation (EP) is a deterministic inference method (Minka, 2001b) that is known to outperform VFE methods in GP classification when unsparsified, fully-factored Gaussian approximations are used (Nickisch and Rasmussen, 2008). Motivated by this observation, EP has been combined with the approximate generative modelling approach to handle non-linear likelihoods (Naish-Guzman and Holden, 2007; Hernández-Lobato and Hernández-Lobato, 2016). This begs the question: can the sparsification and the non-linear approximation be handled in a single EP inference stage, as for VFE? Astonishingly, Csató and Opper not only developed such a method in 2002 (Csató and Opper, 2002), predating much of the work mentioned above, they showed that it is equivalent to applying the FITC approximation and running EP if further approximation is required. In our view, this is a central result, but it appears to have been largely overlooked by the field. Snelson was made aware of it when writing his thesis (Snelson, 2007), briefly acknowledging Csató and Opper's contribution. Qi et al. (2010) extended Csató and Opper's work to utilise inter-domain pseudo-points and they additionally recognised that the EP energy function at convergence is equal to the FITC log-marginal likelihood approximation. Interestingly, no additional term arises as it does when the VFE approach generalised the DTC generative model approach. We are unaware of other work in this vein.

It is hard to be known for certain why these important results are not widely known, but a contributing factor is that the exposition in these papers is largely at Marr's algorithmic level (Dawson, 1998), and does not focus on the computational level making them challenging to understand. Moreover, Csató and Opper's paper was written before EP was formulated in a general way and the presentation, therefore, does not follow what has become the standard approach. In fact, as the focus was online inference, Assumed Density Filtering was employed rather than full-blown EP. One of the main contributions of this chapter is to provide a clear computational exposition including an explicit form of the approximating distribution and full details about each step of the EP procedure. In addition, to bring clarity we make the following novel contributions:

- We show that a generalisation of EP called Power EP can subsume the EP and VFE approaches (and therefore FITC and DTC) into a single unified framework. More precisely, the fixed points of Power EP yield the FITC and VFE posterior distribution under different limits and the Power EP marginal likelihood estimate (the negative 'Power EP energy') recover the FITC marginal likelihood and the VFE too. Critically, the connection to the VFE method leans on the new interpretation of Titsias's approach

(Matthews et al., 2016) outlined in the previous section that directly employs the approximate posterior over function values (rather than augmenting the model with pseudo-points). The connection therefore also requires a formulation of power EP that involves KL divergence minimisation between stochastic processes.

- We show how versions of PEP that are intermediate between the existing VFE and EP approaches can be derived, as well as mixed approaches that treat some data variationally and others using EP. We also show how PITC emerges from the same framework and how to incorporate inter-domain transforms. For regression with Gaussian observation noise, we obtain analytical expressions for the fixed points of Power EP in a general case that includes all of these extensions as well as the form of the Power EP marginal likelihood estimate at convergence that is useful for hyperparameter and pseudo-input optimisation.
- We consider (Gaussian) regression and probit classification as canonical models on which to test the new framework and demonstrate through exhaustive testing that versions of PEP intermediate between VFE and EP perform substantially better on average. The experiments also shed light on situations where VFE is to be preferred to EP and vice versa which is an important open area of research.

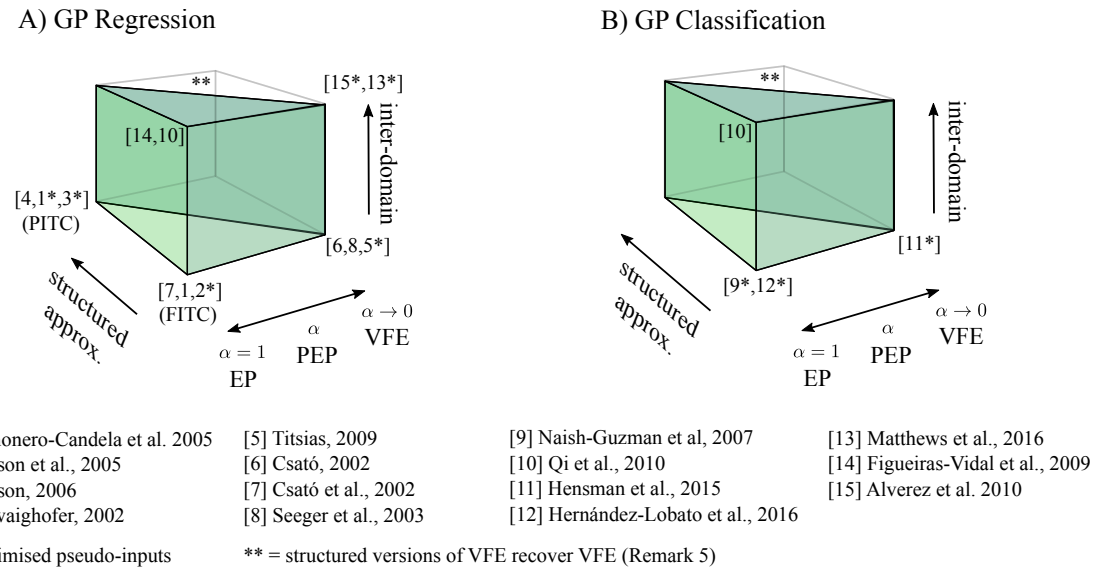
Many of the new theoretical contributions described above are summarised in fig. 2.1 along with their relationship to previous work.

2.3 A new unifying view using Power Expectation Propagation

In this section, we provide a new unifying view of sparse approximation using Power Expectation Propagation (PEP or Power EP) (Minka, 2004). We review Power EP, describe how to apply it for sparse GP regression and classification, and then discuss its relationship to existing methods.

2.3.1 The joint-distribution view of approximate inference and learning

One way of understanding the goal of distributional inference approximations, including the VFE method, EP and Power EP, is that they return an approximation of a tractable form to the model *joint-distribution* evaluated on the observed data. In the case of GP regression and classification, this means $q^*(f|\theta) \approx p(f, \mathbf{y}|\theta)$ where $*$ is used to denote an unnormalised process. Why is the model joint-distribution a sensible object of approximation? The joint distribution can be decomposed into the product of the posterior distribution and the marginal likelihood, $p(f, \mathbf{y}|\theta) = p^*(f|\mathbf{y}, \theta) = p(f|\mathbf{y}, \theta)p(\mathbf{y}|\theta)$, the two inferential objects of interest. A tractable approximation to the joint can therefore be similarly decomposed $q^*(f|\theta) = Zq(f|\theta)$



[1] Quiñonero-Candela et al. 2005
[2] Snelson et al., 2005
[3] Snelson, 2006
[4] Schwaighofer, 2002
[5] Titsias, 2009
[6] Csató, 2002
[7] Csató et al., 2002
[8] Seeger et al., 2003
[9] Naish-Guzman et al, 2007
[10] Qi et al., 2010
[11] Hensman et al., 2015
[12] Hernández-Lobato et al., 2016
[13] Matthews et al., 2016
[14] Figueiras-Vidal et al., 2009
[15] Alverez et al. 2010

* = optimised pseudo-inputs
** = structured versions of VFE recover VFE (Remark 5)

Fig. 2.1 A unified view of pseudo-point GP approximations applied to A) regression and B) classification. Every point in the algorithm polygons corresponds to a form of GP approximation. Previous algorithms correspond to labelled vertices. The new Power EP framework encompasses the three polygons, including their interior.

into a normalised component that approximates the posterior $q(f|\theta) \approx p(f|\mathbf{y},\theta)$ and the normalisation constant which approximates the marginal likelihood $Z \approx p(\mathbf{y}|\theta)$. In other words, the approximation of the joint simultaneously returns approximations to the posterior and marginal likelihood. In the current context tractability of the approximating family means that it is analytically integrable and that this integration can be performed with an appropriate computational complexity. We consider the approximating family comprising unnormalised GPs, $q^*(f|\theta) = Z\mathcal{GP}(f; m_{\mathrm{f}}, V_{\mathrm{ff}'})$.

The VFE approach can be reformulated in the new context using the unnormalised KL divergence (Zhu and Rohwer, 1997) to measure the similarity between the approximation and the joint distribution

$$\overline{\mathrm{KL}}(q^*(f|\theta)||p(f,\mathbf{y}|\theta)) = \int q^*(f) \log \frac{q^*(f)}{p(f,\mathbf{y}|\theta)} \mathrm{d}f + \int \left(p(f,\mathbf{y}|\theta) - q^*(f)\right) \mathrm{d}f. \tag{2.11}$$

The unnormalised KL divergence generalises the KL divergence to accommodate unnormalised densities. It is always non-negative and collapses back to the standard form when its arguments are normalised. Minimising the unnormalised KL with respect to $q^*(f|\theta) = Z_{\mathrm{VFE}} q(f)$ encourages the approximation to match both the posterior and marginal-likelihood, and it

yields analytic solutions

$$q^{\text{opt}}(f) = \underset{q(f)\in\mathcal{Q}}{\operatorname{argmin}} \operatorname{KL}(q(f)||p(f|\mathbf{y},\theta)), \quad \text{and} \quad Z^{\text{opt}}_{\text{VFE}} = \exp(\mathcal{F}(q^{\text{opt}}(f),\theta)). \tag{2.12}$$

That is, the standard variational free-energy approximation to the posterior and marginal likelihood is recovered. One of the pedagogical advantages of framing VFE in this way is that approximation of the posterior and marginal likelihood are committed to upfront, in contrast to the traditional derivation which begins by targeting approximation of the marginal likelihood, but shows that approximation of the posterior emerges as an essential part of this scheme (see section 2.2.2). A disadvantage is that optimisation of hyperparameters must logically proceed by optimising the marginal likelihood approximation, $Z^{\text{opt}}_{\text{VFE}}$, and at first sight therefore appears to necessitate different objective functions for $q^*(f|\theta)$ and θ (unlike the standard view which uses a single objective from the beginning). However, it is easy to show that maximising $p(\mathbf{y}|\theta) - \overline{\text{KL}}(q^*(f|\theta)||p(f,\mathbf{y}|\theta))$ directly for both $q^*(f|\theta)$ and θ is equivalent (see section A.1).

2.3.2 The approximating distribution employed by Power EP

Power EP also approximates the joint-distribution employing an approximating family whose form mirrors that of the target,

$$p^*(f|\mathbf{y},\theta) = p(f|\mathbf{y},\theta)p(\mathbf{y}|\theta) = p(f|\theta)\prod_n p(y_n|f,\theta) \approx p(f|\theta)\prod_n t_n(\mathbf{u}) = q^*(f|\theta). \tag{2.13}$$

Here, the approximation retains the exact prior, but each likelihood term in the exact posterior, $p(y_n|\mathrm{f}_n,\theta)$, is approximated by a simple factor $t_n(\mathbf{u})$ that is assumed Gaussian. These simple factors will be iteratively refined by the PEP algorithm such that they will capture the effect that each true likelihood has on the posterior.

Before describing the details of the PEP algorithm, it is illuminating to consider an alternative interpretation of the approximation. Together, the approximate likelihood functions specify an unnormalised Gaussian over the pseudo-points that can be written $\prod_n t_n(\mathbf{u}) = \mathcal{N}(\tilde{\mathbf{y}}; \tilde{\mathbf{W}}\mathbf{u}, \tilde{\Sigma})$.

The approximate posterior above can therefore be thought of as the (exact) GP posterior resulting from a surrogate regression problem with surrogate observations $\tilde{\mathbf{y}}$ that are generated from linear combinations of the pseudo-points and additive surrogate noise $\tilde{\mathbf{y}} = \tilde{\mathbf{W}}\mathbf{u} + \tilde{\Sigma}^{1/2}\epsilon$. The PEP algorithm will iteratively refine $\{\tilde{\mathbf{y}}, \tilde{\mathbf{W}}, \tilde{\Sigma}\}$ such that exact inference in the simple surrogate regression model returns a posterior and marginal likelihood estimate that is 'close' to that returned by performing exact inference in the intractable complex model (see fig. 2.2).

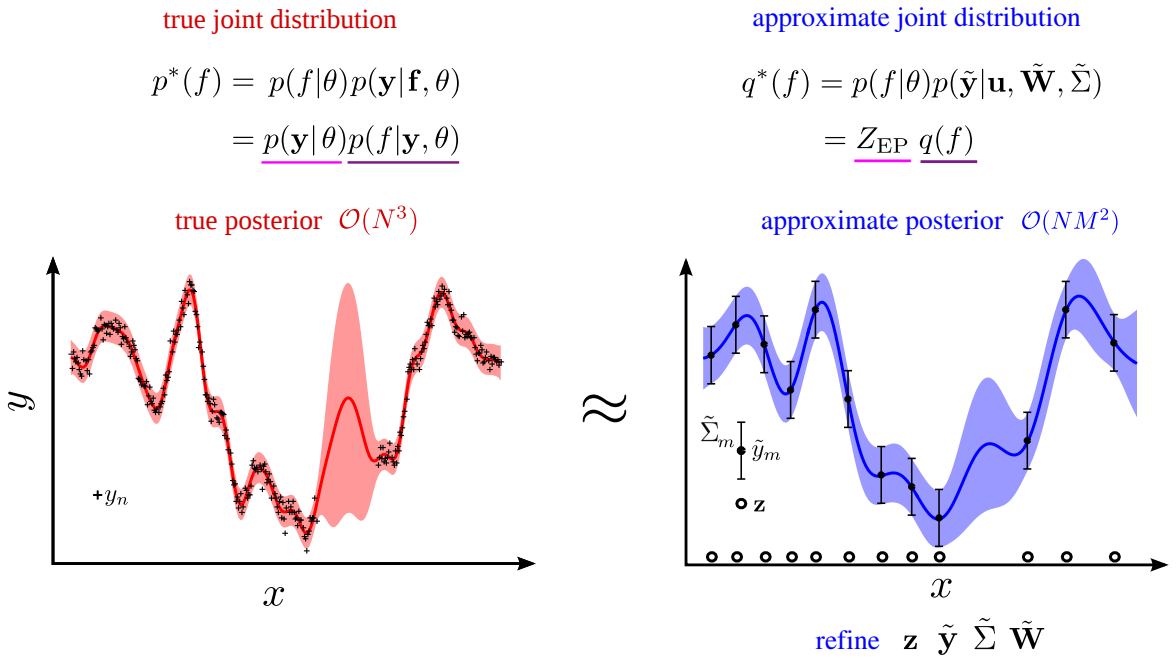


Fig. 2.2 Perspectives on the approximating family. The true joint distribution over the unknown function f and the N datapoints $\mathbf{y}$ (top left) comprises the GP prior and an intractable likelihood function. This is approximated by a surrogate regression model with a joint distribution over the function f and M surrogate datapoints $\tilde{\mathbf{y}}$ (top right). The surrogate regression model employs the same GP prior, but uses a Gaussian likelihood function $p(\tilde{\mathbf{y}}|\mathbf{u},\tilde{\mathbf{W}},\tilde{\Sigma}) = \mathcal{N}(\tilde{\mathbf{y}};\tilde{\mathbf{W}}\mathbf{u},\tilde{\Sigma})$. The intractable true posterior (bottom left) is approximated by refining the surrogate data $\tilde{\mathbf{y}}$ their input locations $\mathbf{z}$ and the parameters of the surrogate model $\tilde{\mathbf{W}}$ and $\tilde{\Sigma}$.

2.3.3 The EP algorithm

One method for updating the approximate likelihood factors $t_n(\mathbf{u})$ is to minimise the unnormalised KL Divergence between the joint distribution and each of the distributions formed by replacing one of the likelihoods by the corresponding approximating factor (Li et al., 2015),

$$\underset{t_n(\mathbf{u})}{\operatorname{argmax}}\ \overline{\mathrm{KL}}\left[p(f,\mathbf{y}|\theta)\middle\|\frac{p(f,\mathbf{y}|\theta)t_n(\mathbf{u})}{p(y_n|\mathrm{f}_n,\theta)}\right] = \underset{t_n(\mathbf{u})}{\operatorname{argmax}}\ \overline{\mathrm{KL}}[p^*_{\backslash n}(f)p(y_n|\mathrm{f}_n,\theta)||p^*_{\backslash n}(f)t_n(\mathbf{u})]. \quad (2.14)$$

Here we have introduced the leave-one-out joint $p^*_{\backslash n}(f) = p(f,\mathbf{y}|\theta)/p(y_n|\mathrm{f}_n,\theta)$ which makes clear that minimisation will cause the approximate factors to approximate the likelihoods in the context of the leave-one-out joint. Unfortunately, such an update is still intractable. Instead, EP approximates this idealised procedure by replacing the exact leave-one-out joint on both sides of the KL by the approximate leave-one-out joint (called the cavity) $p^*_{\backslash n}(f) \approx q^*_{\backslash n}(f) = q^*(f)/t_n(\mathbf{u})$. Not only does this improve tractability, but it also means that the new procedure effectively refines the approximating distribution directly at each

stage, rather than setting the component parts in isolation,

$$\overline{\mathrm{KL}}([q^*_{\backslash n}(f)p(y_n|\mathrm{f}_n,\theta)||q^*_{\backslash n}(f)t_n(\mathbf{u})] = \overline{\mathrm{KL}}([q^*_{\backslash n}(f)p(y_n|\mathrm{f}_n,\theta)||q^*(f)]. \tag{2.15}$$

However, the updates for the approximating factors are now coupled and so the updates must now be iterated, unlike in the idealised procedure. In this way, EP iteratively refines the approximate factors or surrogate likelihoods so that the GP posterior of the surrogate regression task best approximates the posterior of the original regression/classification problem.

2.3.4 The Power EP algorithm

Power EP is, algorithmically, a mild generalisation of the EP algorithm that instead removes (or includes) a fraction α of the approximate (or true) likelihood functions in the following steps:

1. **Deletion**: compute the cavity distribution by removing a fraction of one approximate factor, $q^{\backslash n}(f|\theta) \propto q^*(f|\theta)/t_n^\alpha(\mathbf{u})$.
2. **Projection**: first, compute the tilted distribution by incorporating a corresponding fraction of the true likelihood into the cavity, $\tilde{p}(f) = q^{\backslash n}(f|\theta)p^\alpha(y_n|\mathrm{f}_n)$. Second, project the tilted distribution onto the approximate posterior using the KL divergence for unnormalised densities,

$$q^*(f|\theta) \leftarrow \underset{q^*(f|\theta)\in\mathcal{Q}}{\operatorname{argmin}} \overline{\mathrm{KL}}(\tilde{p}(f)||q^*(f|\theta)). \tag{2.16}$$

   Here $\mathcal{Q}$ is the set of allowed $q^*(f|\theta)$ defined by eq. (2.13).
3. **Update**: compute a new fraction of the approximate factor by dividing the new approximate posterior by the cavity, $t^\alpha_{n,\mathrm{new}}(\mathbf{u}) = q^*(f|\theta)/q^{\backslash n}(f|\theta)$, and incorporate this fraction back in to obtain the updated factor, $t_n(\mathbf{u}) = t^{1-\alpha}_{n,\mathrm{old}}(\mathbf{u})t^\alpha_{n,\mathrm{new}}(\mathbf{u})$.

The above steps are iteratively repeated for each factor that needs to be approximated. Notice that the procedure only involves one likelihood factor to be handled at a time. In the case of analytically intractable likelihood functions, this often requires only low dimensional integrals to be computed. In other words, PEP has transformed a high dimensional intractable integral that is hard to approximate into a set of low dimensional intractable integrals that are simpler to approximate. The procedure is not, in general guaranteed to converge but we did not observe any convergence issues in our experiments. Furthermore, it can be shown to be numerically stable when the factors are log-concave (as in GP regression and classification) (Seeger, 2008). If Power EP converges, the fractional updates are equivalent to running the original EP procedure, but replacing the KL minimisation with an α-divergence minimisation

(Zhu and Rohwer, 1995; Minka, 2005),

$$\overline{\mathrm{D}}_\alpha[p^*(f)||q^*(f)] = \frac{1}{\alpha(1-\alpha)}\int \left[\alpha p^*(f) + (1-\alpha)q^*(f) - p^*(f)^\alpha q^*(f)^{1-\alpha}\right] \mathrm{d}f. \tag{2.17}$$

When $\alpha = 1$, the α-divergence is the inclusive KL divergence $\overline{\mathrm{D}}_1[p^*(f)||q^*(f)] = \overline{\mathrm{KL}}[p^*(f)||q^*(f)]$ recovering EP as expected from the PEP algorithm. As $\alpha \to 0$ the exclusive KL divergence is recovered, $\overline{\mathrm{D}}_{\to 0}[p^*(f)||q^*(f)] = \overline{\mathrm{KL}}[q^*(f)||p^*(f)]$, and since minimising a set of local exclusive KL divergences is equivalent to minimising a single global exclusive KL divergence (Minka, 2005), the Power EP solution is the minimum of a variational free-energy (see section A.2 for more details). In the current case, we will now show that these cases of Power EP recover FITC and Titsias's VFE solution respectively.

2.3.5 General results for Gaussian process Power EP

This section describes the Power EP steps in finer detail showing the complexity is $\mathcal{O}(NM^2)$ and laying the groundwork for the equivalence relationships. The section A.6 includes a full derivation.

We start by defining the approximate factors to be in natural parameter form, making it simple to combine and delete them, $t_n(\mathbf{u}) = \tilde{\mathcal{N}}(\mathbf{u}; z_n, \mathbf{T}_{1,n}, \mathbf{T}_{2,n}) = z_n \exp(\mathbf{u}^\intercal \mathbf{T}_{1,n} - \frac{1}{2}\mathbf{u}^\intercal \mathbf{T}_{2,n}\mathbf{u})$. We consider full rank $\mathbf{T}_{2,n}$, but will show that the optimal form is rank 1. The parameterisation means the approximate posterior over the pseudo-points has natural parameters $\mathbf{T}_{1,\mathbf{u}} = \sum_n \mathbf{T}_{1,n}$ and $\mathbf{T}_{2,\mathbf{u}} = \mathbf{K}_{\mathbf{uu}}^{-1} + \sum_n \mathbf{T}_{2,n}$ inducing an approximate GP posterior, $\mathcal{GP}(f; m_\mathrm{f}, V_{\mathrm{ff}'})$ with mean and covariance function,

$$m_\mathrm{f} = \mathbf{K}_{\mathrm{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}_{2,\mathbf{u}}^{-1}\mathbf{T}_{1,\mathbf{u}}; \quad V_{\mathrm{ff}'} = \mathbf{K}_{\mathrm{ff}'} - \mathbf{Q}_{\mathrm{ff}'} + \mathbf{K}_{\mathrm{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}_{2,\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathrm{u}f'}. \tag{2.18}$$

Deletion: The cavity for datapoint n, $q^{\backslash n}(f) \propto q^*(f)/t_n^\alpha(\mathbf{u})$, has a similar form to the posterior, but the natural parameters are modified by the deletion step, $\mathbf{T}_{1,\mathbf{u}}^{\backslash n} = \mathbf{T}_{1,\mathbf{u}} - \alpha\mathbf{T}_{1,n}$ and $\mathbf{T}_{2,\mathbf{u}}^{\backslash n} = \mathbf{T}_{2,\mathbf{u}} - \alpha\mathbf{T}_{2,n}$, yielding a new mean and covariance function

$$m_\mathrm{f}^{\backslash n} = \mathbf{K}_{\mathrm{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}_{2,\mathbf{u}}^{\backslash n,-1}\mathbf{T}_{1,\mathbf{u}}^{\backslash n}; \quad V_{\mathrm{ff}'}^{\backslash n} = \mathbf{K}_{\mathrm{ff}'} - \mathbf{Q}_{\mathrm{ff}'} + \mathbf{K}_{\mathrm{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}_{2,\mathbf{u}}^{\backslash n,-1}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathrm{u}f'}. \tag{2.19}$$

Projection: The central step in Power EP is the projection. Obtaining the new approximate unnormalised posterior $q^*(f)$ by minimising $\overline{\mathrm{KL}}(\tilde{p}(f)||q^*(f))$ would naïvely appear intractable. Fortunately,

Remark 1. *Because of the structure of the approximate posterior, $q(f) = p(f_{\neq\mathbf{u}}|\mathbf{u})q(\mathbf{u})$, the objective, $\mathrm{KL}(\tilde{p}(f)||q^*(f))$ is minimised when $\mathbb{E}_{\tilde{p}(f)}[\phi(\mathbf{u})] = \mathbb{E}_{q^*(\mathbf{u})}[\phi(\mathbf{u})]$, where $\phi(\mathbf{u}) = [\mathbf{u}, \mathbf{u}\mathbf{u}^\intercal]$ are the sufficient statistics, that is when the moments at the pseudo-inputs are matched.*

This is the central result from which computational savings are derived. Furthermore, this moment matching condition would appear to necessitate computation of a set of integrals to find the zeroth, first and second moments. However, the technique known as 'differentiation under the integral sign' (see e.g. Brown, 1986) provides a useful shortcut that only requires one integral to compute the log-normaliser of the tilted distribution, $\log \tilde{Z}_n = \log \mathbb{E}_{q^{\backslash n}(f)}[p^\alpha(y_n|\mathrm{f}_n)]$, before differentiating w.r.t. the cavity mean to give

$$\mathbf{m}_\mathbf{u} = \mathbf{m}_\mathbf{u}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n} \frac{\mathrm{d} \log \tilde{Z}_n}{\mathrm{d} m_{\mathrm{f}_n}^{\backslash n}}; \qquad \mathbf{V}_\mathbf{u} = \mathbf{V}_\mathbf{u}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n} \frac{\mathrm{d}^2 \log \tilde{Z}_n}{\mathrm{d}(m_{\mathrm{f}_n}^{\backslash n})^2} \mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}. \tag{2.20}$$

Update: Having computed the new approximate posterior, the approximate factor $t_{n,\mathrm{new}}(\mathbf{u}) = q^*(f)/q^{\backslash n}(f)$ can be straightforwardly obtained, resulting in,

$$\mathbf{T}_{1,n,\mathrm{new}} = \mathbf{V}_\mathbf{u}^{-1}\mathbf{m}_\mathbf{u} - (\mathbf{V}_\mathbf{u}^{\backslash n})^{-1}\mathbf{m}_\mathbf{u}^{\backslash n},\ \mathbf{T}_{2,n,\mathrm{new}} = \mathbf{V}_\mathbf{u}^{-1} - (\mathbf{V}_\mathbf{u}^{\backslash n})^{-1},\ z_n^\alpha = \tilde{Z}_n \mathrm{e}^{\mathcal{G}(q_*^{\backslash n}(\mathbf{u})) - \mathcal{G}(q^*(\mathbf{u}))},$$

where we have defined the log-normaliser $\mathcal{G}(\tilde{\mathcal{N}}(\mathbf{u}; z, \mathbf{T}_1, \mathbf{T}_2)) = \log \int \tilde{\mathcal{N}}(\mathbf{u}; z, \mathbf{T}_1, \mathbf{T}_2)\mathrm{d}\mathbf{u}$. Remarkably, these results and eqs. 2.20 reveals that $\mathbf{T}_{2,n,new}$ is a rank-1 matrix. As a result, the minimal and simplest way to parameterise the approximate factor is $t_n(\mathbf{u}) = z_n \mathcal{N}(\mathbf{K}_{\mathrm{f}_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u}; g_n, v_n)$, where g_n and v_n are scalars, resulting in significant memory saving and an $\mathcal{O}(NM^2)$ cost.

In addition to providing the approximate posterior after convergence, Power EP also provides an approximate log-marginal likelihood for model selection and hyperparameter optimisation,

$$\log \mathcal{Z}_{\mathrm{PEP}}(\theta) = \log \int p(f|\theta) \prod_n t_n(\mathbf{u}) \mathrm{d}f = \mathcal{G}(q^*(\mathbf{u})) - \mathcal{G}(p^*(\mathbf{u})) + \sum_n \log z_n. \tag{2.21}$$

Armed with these general results, we now consider the implications for Gaussian Process regression.

2.3.6 Gaussian regression case

When the model contains Gaussian likelihood functions, closed-form expressions for the Power EP approximate factors at convergence can be obtained and hence the approximate posterior:

$$t_n(\mathbf{u}) = \mathcal{N}(\mathbf{K}_{\mathrm{f}_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u}; y_n, \alpha D_{\mathrm{f}_n\mathrm{f}_n} + \sigma_y^2), \quad q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{K}_{\mathbf{uf}}\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y}, \mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uf}}\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{K}_{\mathbf{fu}})$$

where $\overline{\mathbf{K}}_{\mathbf{ff}} = \mathbf{Q}_{\mathbf{ff}} + \alpha \mathrm{diag}(\mathbf{D}_{\mathbf{ff}}) + \sigma_y^2 \mathrm{I}$ and $\mathbf{D}_{\mathbf{ff}} = \mathbf{K}_{\mathbf{ff}} - \mathbf{Q}_{\mathbf{ff}}$ as defined in section 2.2. These analytic expressions can be rigorously proven to be the stable fixed point of the Power EP procedure using remark 1. Briefly, assuming the factors take the form above, the natural

parameters of the cavity $q^{\backslash n}(\mathbf{u})$ become,

$$\mathbf{T}_{1,\mathbf{u}}^{\backslash n} = \mathbf{T}_{1,\mathbf{u}} - \alpha\gamma_n y_n \mathbf{K}_{\mathrm{f}_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}, \quad \mathbf{T}_{2,\mathbf{u}}^{\backslash n} = \mathbf{T}_{2,\mathbf{u}} - \alpha\gamma_n \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}\mathrm{f}_n}\mathbf{K}_{\mathrm{f}_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}, \tag{2.22}$$

where $\gamma_n^{-1} = \alpha D_{\mathrm{f}_n\mathrm{f}_n} + \sigma_y^2$. The subtracted quantities in the equations above are exactly the contribution the likelihood factor makes to the cavity distribution (see remark 1) so $\int q^{\backslash n}(f)p^\alpha(y_n|\mathrm{f}_n)\mathrm{d}f_{\neq\mathbf{u}} = q^{\backslash n}(\mathbf{u})\int p(\mathrm{f}_n|\mathbf{u})p^\alpha(y_n|\mathrm{f}_n)\mathrm{d}\mathrm{f}_n \propto q(\mathbf{u})$. Therefore, the posterior approximation remains unchanged after an update and the form for the factors above is the fixed point. Moreover, the approximate log-marginal likelihood is also analytically tractable,

$$\log \mathcal{Z}_{\mathrm{PEP}} = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\overline{\mathbf{K}}_{\mathbf{ff}}| - \frac{1}{2}\mathbf{y}^\intercal\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y} - \frac{1-\alpha}{2\alpha}\sum_n \log\left(1 + \alpha D_{\mathrm{f}_n\mathrm{f}_n}/\sigma_y^2\right).$$

We now look at special cases and the correspondence to the methods discussed in section 2.2.

Remark 2. *When $\alpha = 1$ [EP], the Power EP posterior becomes the FITC posterior in eq.* (2.7) *and the Power EP approximate marginal likelihood becomes the FITC marginal likelihood in eq.* (2.8)*. In other words, the FITC approximation for GP regression is, surprisingly, equivalent to running an EP algorithm for sparse GP posterior approximation to convergence.*

Remark 3. *As $\alpha \to 0$ the approximate posterior and approximate marginal likelihood are identical to that of the VFE approach in eqs.* (2.9) *and* (2.10) *(Titsias, 2009). This result uses the limit: $\lim_{x\to 0} x^{-1}\log(1+x) = 1$. So FITC and Titsias's VFE approach employ the same form of pseudo-point approximation, but refine it in different ways.*

2.3.7 Extensions: structured, inter-domain and multi-power Power EP approximations

The framework can now be generalised in three orthogonal directions:

1. enable structured approximations to be handled that retain more dependencies in the spirit of PITC (see section 2.2.1)
2. incorporate inter-domain pseudo-points thereby adding further flexibility to the form of the approximate posterior
3. employ different powers α for each factor (thereby enabling e.g. VFE updates to be used for some datapoints and EP for others).

Given the groundwork above, these three extensions are straightforward. In order to handle structured approximations, we take inspiration from PITC and partition the data into B disjoint blocks $\mathbf{y}_b = \{y_n\}_{n\in\mathcal{B}_b}$ (see section 2.2.1). Each PEP factor update will then approximate an entire block which will contain a set of datapoints, rather than just a single

one. This is a style of EP approximation that has recently been used to distribute Monte Carlo algorithms across many machines (Gelman et al., 2014b; Xu et al., 2014).

In order to handle inter-domain variables, we define a new domain via a linear transform $g(\mathbf{x}) = \int \mathrm{d}\mathbf{x}' W(\mathbf{x}, \mathbf{x}') f(\mathbf{x}')$ which now contains the pseudo-points $g = \{g_{\neq \mathbf{u}}, \mathbf{u}\}$. Choices for $W(\mathbf{x}, \mathbf{x}')$ include Gaussians or wavelets. These two extensions mean that the approximation becomes,

$$p(f, g|\theta) \prod_b p(\mathbf{y}_b|f, \theta) \approx p(f, g|\theta) \prod_b t_b(\mathbf{u}) = q^*(f|\theta). \tag{2.23}$$

Power EP is then performed using private powers α_b for each data block, which is the third generalisation mentioned above. Analytic solutions are again available (covariance matrices now incorporate the inter-domain transform)

$$t_b(\mathbf{u}) = \mathcal{N}(\mathbf{K}_{\mathbf{f}_b\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u}; \mathbf{y}_b, \alpha_b\mathbf{D}_{\mathbf{f}_b\mathbf{f}_b} + \sigma_y^2\mathrm{I}), \quad q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{K}_{\mathbf{uf}}\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y}, \mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uf}}\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{K}_{\mathbf{fu}})$$

where $\overline{\mathbf{K}}_{\mathbf{ff}} = \mathbf{Q}_{\mathbf{ff}} + \mathrm{blkdiag}(\{\alpha_b\mathbf{D}_{\mathbf{f}_b\mathbf{f}_b}\}_{b=1}^B) + \sigma_y^2\mathrm{I}$ and blkdiag builds a block-diagonal matrix from its inputs. The approximate log-marginal likelihood can also be obtained in closed-form,

$$\log \mathcal{Z}_{\mathrm{PEP}} = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\overline{\mathbf{K}}_{\mathbf{ff}}| - \frac{1}{2}\mathbf{y}^\intercal\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y} + \sum_b \frac{1-\alpha_b}{2\alpha_b}\log\left(\mathrm{I} + \alpha_b\mathbf{D}_{\mathbf{f}_b\mathbf{f}_b}/\sigma_y^2\right).$$

Remark 4. *When $\alpha_b = 1$ and $W(\mathbf{x}, \mathbf{x}') = \delta(\mathbf{x} - \mathbf{x}')$ the structured Power EP posterior becomes the PITC posterior and the Power EP approximate marginal likelihood becomes the PITC marginal likelihood. Additionally, when $B = N$ we recover FITC as discussed in section 2.3.6.*

Remark 5. *When $\alpha_b \to 0$ and $W(\mathbf{x}, \mathbf{x}') = \delta(\mathbf{x} - \mathbf{x}')$ the structured Power EP posterior and approximate marginal likelihood becomes identical to the VFE approach (Titsias, 2009). (See fig. 2.1 for more relationships.)*

2.3.8 Classification

For classification, the non-Gaussian likelihood prevents an analytic solution. As such, the iterative Power EP procedure is required to obtain the approximate posterior. The projection step requires computation of the log-normaliser of the tilted distribution, $\log \tilde{Z}_n = \log \mathbb{E}_{q^{\setminus n}(f)}[p^\alpha(y_n|f)] = \log \mathbb{E}_{q^{\setminus n}(\mathrm{f}_n)}[\Phi^\alpha(y_n\mathrm{f}_n)]$. For general α, this quantity is not available in closed form[2]. However, it involves a one-dimensional expectation of a non-linear function of a normally-distributed random variable and, therefore, can be approximated using Gauss-Hermite quadrature. This procedure gives an approximation to the expectation, resulting in

[2]except for special cases, e.g. when $\alpha = 1$ and $\Phi(x)$ is the probit inverse link function, $\Phi(x) = \int_{-\infty}^{x} \mathcal{N}(a; 0, 1)\mathrm{d}a$.

an approximate update for the posterior mean and covariance. The approximate log-marginal likelihood can also be obtained and used for hyperparameter optimisation. As $\alpha \to 0$, it becomes the variational free-energy used in (Hensman et al., 2015) which employs quadrature for the same purpose. These relationships are shown in fig. 2.1 which also shows that inter-domain transformations and structured approximations have not yet been employed in the classification setting. In our view, the inter-domain generalisation would be a sensible one to pursue and it is mathematically and algorithmically straightforward. The structured approximation variant is more complicated as it requires multiple non-linear likelihoods to be handled at each step of EP. This will require further approximation such as using Monte Carlo methods (Gelman et al., 2014b; Xu et al., 2014).

Since the proposed Power EP approach is general, an extension to other likelihood functions is as simple as for VFE methods (Dezfouli and Bonilla, 2015). For example, the multinomial probit likelihood can be handled in the same way as the binary case, where the log-normaliser of the tilted distribution can be computed using a C-dimensional Gaussian quadrature [C is the number of classes] (Seeger and Jordan, 2004) or nested EP (Riihimäki et al., 2013).

2.3.9 Complexity

The computational complexity of all the regression and classification methods described in this section is $\mathcal{O}(NM^2)$ for training, and $\mathcal{O}(M^2)$ per test point for prediction. The training cost can be further reduced to $\mathcal{O}(M^3)$, in a similar vein to the uncollapsed VFE approach (Hensman et al., 2013, 2015), by employing stochastic updates of the posterior and stochastic optimisation of the hyperparameters using minibatches of datapoints (Hernández-Lobato and Hernández-Lobato, 2016). In particular, the Power EP update steps in section 2.3.2 are repeated for only a small subset of training points and for only a small number of iterations. The approximate log-marginal likelihood in eq. (2.21) is then computed using this minibatch and optimised as if the Power EP procedure has converged. This approach results in a computationally efficient training scheme, at the cost of returning noisy hyperparameter gradients. In practice, we find that the noise can be handled using stochastic optimisers such as Adam (Kingma and Ba, 2015). In summary, given these advances, the general PEP framework is as scalable as variational inference.

2.4 Experiments

The general framework described above lays out a large space of potential inference algorithms suggesting many exciting directions for innovation. The experiments considered in the chapter will investigate only one aspect of this space; how do algorithms that are intermediate between VFE ($\alpha = 0$) and EP/FITC ($\alpha = 1$) perform? Specifically, we will investigate how the

performance of the inference scheme varies as a function of α and whether this depends on; the type of problem (classification, regression or state-space modelling); the dataset (synthetic datasets, 8 real-world regression datasets and 6 classification datasets); the performance metric (we compare metrics that require point-estimates to those that are uncertainty sensitive). An important by-product of the experiments is that they provide a comprehensive comparison between the VFE and EP approaches which has been an important area of debate in its own right.

The results presented below are compact summaries of a large number of experiments full details of which are included in the appendix of Bui et al. (2017b).

2.4.1 Regression on synthetic datasets

In the first experiment, we investigate the performance of the proposed Power EP method on toy regression datasets where ground truth is known. We vary α (from 0 VFE to 1 EP/FITC) and the number of pseudo-points (from 5 to 500). We use thirty datasets, each comprising 1000 datapoints with five input dimensions and one output dimension, that were drawn from a GP with an Automatic Relevance Determination squared exponential kernel. A 50:50 train/test split was used. The hyperparameters and pseudo-inputs were found by optimising the PEP energy using L-BFGS with a maximum of 2000 function evaluations. The performances are compared using two metrics: standardised mean squared error (SMSE) and standardised mean log loss (SMLL) as described in (Rasmussen and Williams, 2005, page 23). The approximate negative log-marginal likelihood (NLML) for each experiment is also computed. The mean performance using Power EP with different α values and full GP regression is shown in fig. 2.3. The results demonstrate that as M increases, the SMLL and SMSE of the sparse methods approach that of full GP. Power EP with $\alpha = 0.8$ or $\alpha = 1$ (EP) overestimates the log-marginal likelihood when intermediate numbers of pseudo-points are used, but the overestimation is markedly less when $M = N = 500$. The jump from a large overestimation to a small overestimation in fig. 2.3 is consistent across multiple random seeds and various pseudo-input initialisations. Importantly, however, an intermediate value of α in the range 0.5-0.8 seems to be best for prediction on average, outperforming both EP and VFE.

2.4.2 Regression on real-world datasets

The experiment above was replicated on 8 UCI regression datasets, each with 20 train/test splits. We varied α between 0 and 1, and M was varied between 5 and 200. Full details of the experiments along with extensive additional analysis are presented in the appendices. Here we concentrate on several key aspects. First we consider pairwise comparisons between VFE ($\alpha \to 0$), Power EP with $\alpha = 0.5$ and EP/FITC ($\alpha = 1$) on both the SMSE and SMLL evaluation metrics. Power EP with $\alpha = 0.5$ was chosen because it is the mid-point between

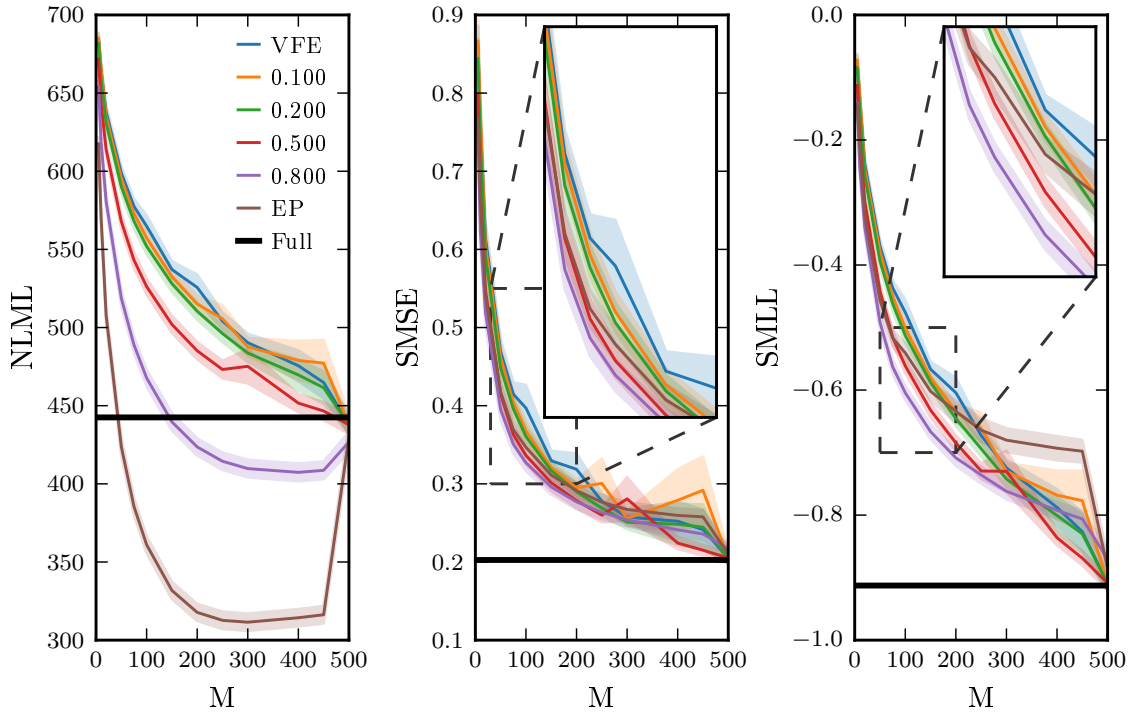


Fig. 2.3 The performance of various α values averaged over 30 trials. See text for more details

VFE and EP and because settings around this value empirically performed the best on average across all datasets, splits, numbers of pseudo-points, and evaluation metrics.

In fig. 2.4A we plot (for each dataset, each split and each setting of M) the evaluation scores obtained using one inference algorithm (e.g. PEP $\alpha = 0.5$) against the score obtained using another (e.g. VFE $\alpha = 0$). In this way, points falling below the identity line indicate experiments where the method on the y-axis outperformed the method on the x-axis. These results have been collapsed by forming histograms of the difference in the performance of the two algorithms, such that mass to the right of zero indicates the method on the y-axis outperformed that on the x-axis. The proportion of mass on each side of the histogram, also indicated on the plots, shows in what fraction of experiments one method returns a more accurate result than the other. This is a useful summary statistic, linearly related to the average rank, that we will use to unpack the results. The average rank is insensitive to the magnitude of the performance differences and readers might worry that this might give an overly favourable view of a method that performs the best frequently, but only by a tiny margin, and when it fails it does so catastrophically. However, the histograms indicate that the methods that win most frequently tend also to 'win big' and 'lose small', although EP is a possible exception to this trend (see the outliers below the identity line on the bottom right-hand plot).

A clear pattern emerges from these plots. First PEP $\alpha = 0.5$ is the best performing approach on the SMSE metric, outperforming VFE 67% of the time and EP 78% of the time. VFE is better than EP on the SMSE metric 64% of the time. Second, EP performs the best on the SMLL metric, outperforming VFE 93% of the time and PEP $\alpha = 0.5$ 71% of the time. PEP $\alpha = 0.5$ outperforms VFE in terms of the SMLL metric 93% of the time.

These pairwise rank comparisons have been extended to other values of α in fig. 2.5A. Here, each row of the figure compares one approximation with all others. Horizontal bars indicate that the methods have equal average rank. Upward sloping bars indicate the method shown on that row has lower average rank (better performance), and downward sloping bars indicate higher average rank (worse performance). The plots show that PEP $\alpha = 0.5$ outperforms all other methods on the SMSE metric, except for PEP $\alpha = 0.6$ which is marginally better. EP is outperformed by all other methods, and VFE only outperforms EP on this metric. On the other hand, EP is the clear winner on the SMLL metric, with performance monotonically decreasing with α so that VFE is the least favourable.

The same pattern of results is seen when we simultaneously compare all of the methods, rather than considering sets of pairwise comparisons. The average rank plots shown in fig. 2.4B were produced by sorting the performances of the 8 different approximating methods for each dataset, split, and number of pseudo-points M and assigning a rank. These ranks are then averaged over all datasets and their splits, and settings of M. PEP $\alpha = 0.5$ is the best for the SMSE metric, and the two worst methods are EP and VFE. PEP $\alpha = 0.8$ is the best for the SMLL metric, with EP and PEP $\alpha = 0.6$ not far behind (when EP performs poorly it can do so with a large magnitude, explaining the discrepancy with the pairwise ranks).

There is some variability between individual datasets, but the same general trends are clear: For MSE, $\alpha = 0.5$ is better than VFE on 6/8 datasets and EP on 8/8 datasets, whilst VFE is better than EP on 3 datasets (the difference on the others being small). For NLL, EP is better than $alpha = 0.5$ on 5/8 datasets and VFE on 7/8 datasets, whilst $alpha = 0.5$ is better than VFE on 8/8 datasets. Performance tends to increase for all methods as a function of the number of pseudo-points M. The interaction between the choice of M and the best performing inference method is often complex and variable across datasets making it hard to give precise advice about selecting α in an M dependent way.

In summary, we make the following recommendations based on these results for GP regression problems. For a MSE loss, we recommend using $\alpha = 0.5$. For a NLL, we recommend using EP. It is possible that more fine-grained recommendations are possible based upon details of the dataset and the computational resources available for processing, but further work will be needed to establish this.

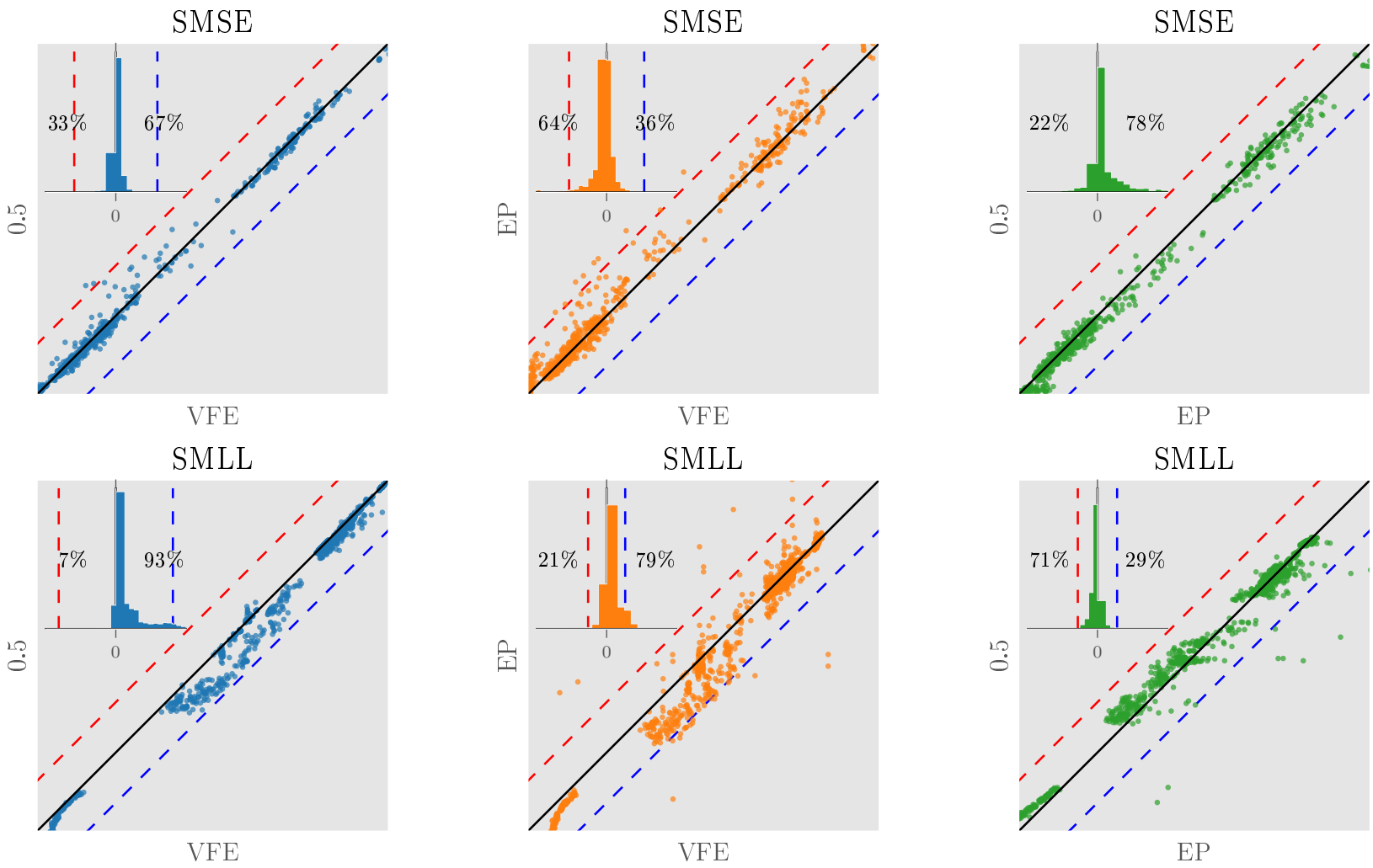


Fig. 2.4 Pair-wise comparisons between Power EP with $\alpha = 0.5$, EP ($\alpha = 1$) and VFE ($\alpha \to 0$), evaluated on several regression datasets and various settings of M. Each coloured point is the result of one split. Points that are below the diagonal line illustrate the method on the y-axis is better than the method on the x-axis. The inset diagrams show the histograms of the difference between methods (x-value $-$ y-value), and the counts of negative and positive differences. Note that this indicates the pairwise ranking of the two methods. Positive differences mean the y-axis method is better than the x-axis method and vice versa. For example, the middle, bottom plot shows EP is on average better than VFE.

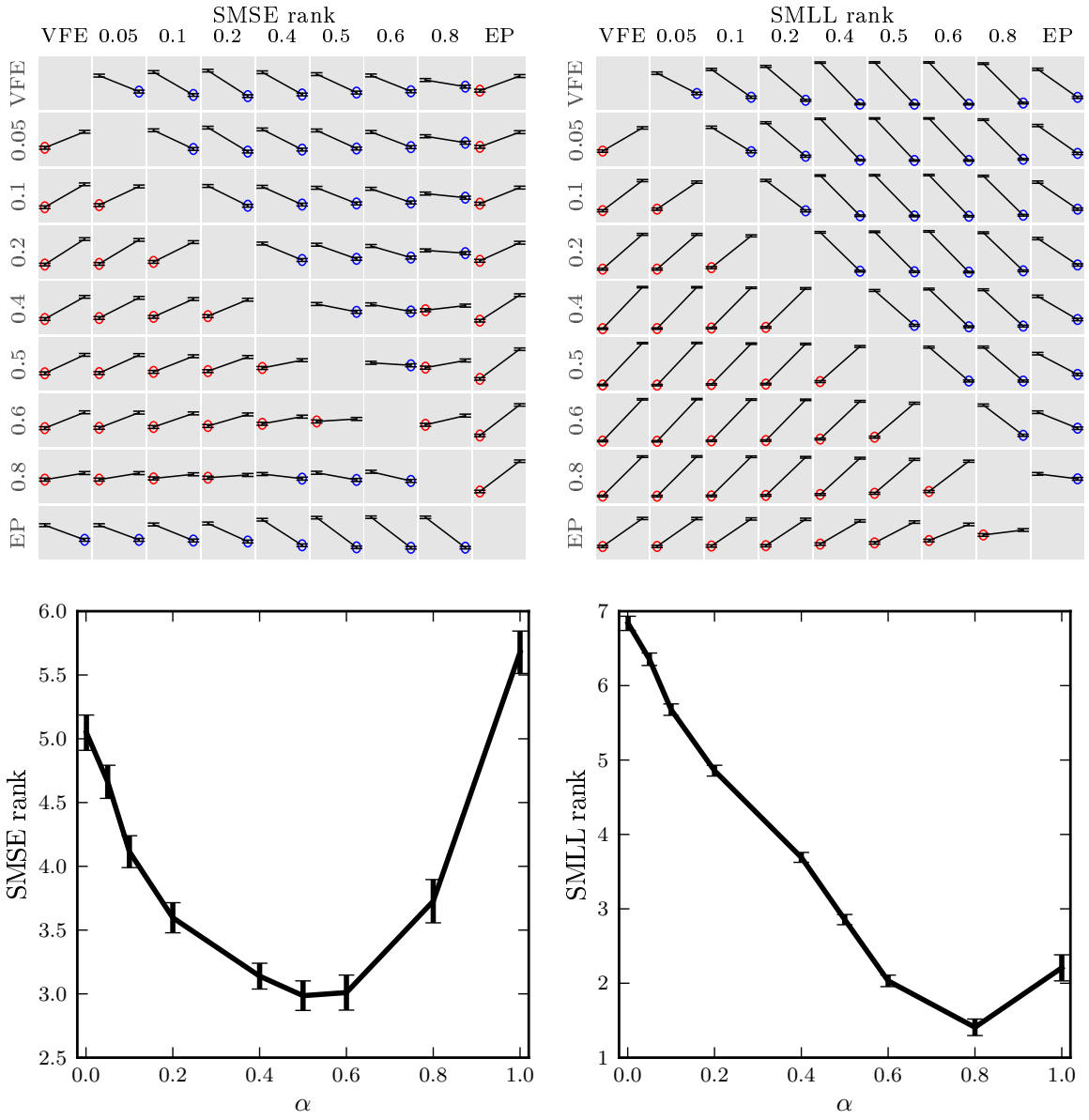


Fig. 2.5 Average ranking of various α values in the regression experiment, lower is better. Top plots show the pairwise comparisons. Red circles denote rows being better than the corresponding columns, and blue circles mean vice versa. Bottom plots show the ranks of all methods when being compared together. Intermediate α values (not EP or VFE) are best on average.

2.4.3 Binary classification

We also evaluated the Power EP method on 6 UCI classification datasets, each has 20 train/test splits. The details of the datasets are included in the appendix of Bui et al.

(2017b).. The datasets are all roughly balanced, and the most imbalanced is `pima` with 500 positive and 267 negative datapoints. Again α was varied between 0 and 1, and M was varied between 10 and 100. We adopt the experimental protocol discussed in section 2.3.9, including (i) not waiting for Power EP to converge before making hyperparameter updates, (ii) using minibatches of datapoints for each Power EP sweep, (iii) parallel factor updates. The Adam optimiser was used with default hyperparameters to handle the noisy gradients produced by these approximations (Kingma and Ba, 2015). We also implemented the VFE approach of Hensman et al. (2015) and include this in the comparison to the PEP methods. The VFE approach should be theoretically identical to PEP with small α, however, we note that the results can be slightly different due to the difference in practical implementations – optimisation for VFE vs. iterative procedure and each step only gets to see a tiny fraction of each datapoint when α is small for PEP. Similar to the regression experiment, we compare the methods using the pairwise ranking plots on the test error and negative log-likelihood (NLL) evaluation metrics.

In fig. 2.6, we plot (for each dataset, each split and each setting of M) the evaluation scores using one inference algorithm against the score obtained using another [see section 2.4.2 for a detailed explanation of the plots]. In contrast to the regression results in section 2.4.2, there are no clear-cut winners among the methods. The test error results show that PEP $\alpha = 0.5$ is marginally better than VFE and EP, while VFE slightly edges EP out in this metric. Similarly, all methods perform comparably on the NLL scale, except with PEP $\alpha = 0.5$ outperforming EP by a narrow margin (65% of the time vs. 35%)

We repeat the pairwise comparison above to all methods and show the results in fig. 2.7. The plots show that there is no conclusive winner on the test error metric, and VFE, PEP $\alpha = 0.4$ and PEP $\alpha = 0.5$ have a slight edge over other α values on the NLL metric. Notably, methods corresponding to bigger α values, such as PEP $\alpha = 0.8$ and EP, are outperformed by all other methods. Similar to the regression experiment, we observe the same pattern of results when all methods are simultaneously compared, as shown in fig. 2.7. However, the big errorbars suggest the difference between the methods is small in both metrics.

There is some variability between individual datasets, but the general trends are clear and consistent with the pattern noted above. For test error, PEP $\alpha = 0.5$ is better than VFE on 1/6 dataset and is better than EP on 3/6 datasets (the differences on the other datasets are small). VFE outperforms EP on 2/6 datasets, while EP beats VFE on only 1/6 datasets. For NLL, PEP $\alpha = 0.5$ only clearly outperforms VFE on 1/6 dataset but is worse compared to VFE on 1 dataset (the other 4 datasets have no clear winner). PEP $\alpha = 0.5$ is better than EP on 5/6 datasets and EP is better on the remaining dataset). EP is only better than VFE on 2/6 datasets and is outperformed by VFE on the other 4/6 datasets. The finding that PEP and VFE are slightly better than EP on the NLL metric is surprising as we expected EP perform the best on the uncertainty sensitive metric (just as was discovered in the regression case). The full results are included in the appendices (see figs 25, 26 and 27). Similar to the

regression case, we observe that as M increases, the performance tends to be better for all methods and the differences between the methods tend to become smaller, but we have not found evidence for systematic sensitivity to the nature of the approximation.

In summary, we make the following recommendations based on these results for GP classification problems. For a raw test error loss and for NLL, we recommend using $\alpha = 0.5$ (or $\alpha = 0.4$). It is possible that more fine-grained recommendations are possible based upon details of the dataset and the computational resources available for processing, but further work will be needed to establish this.

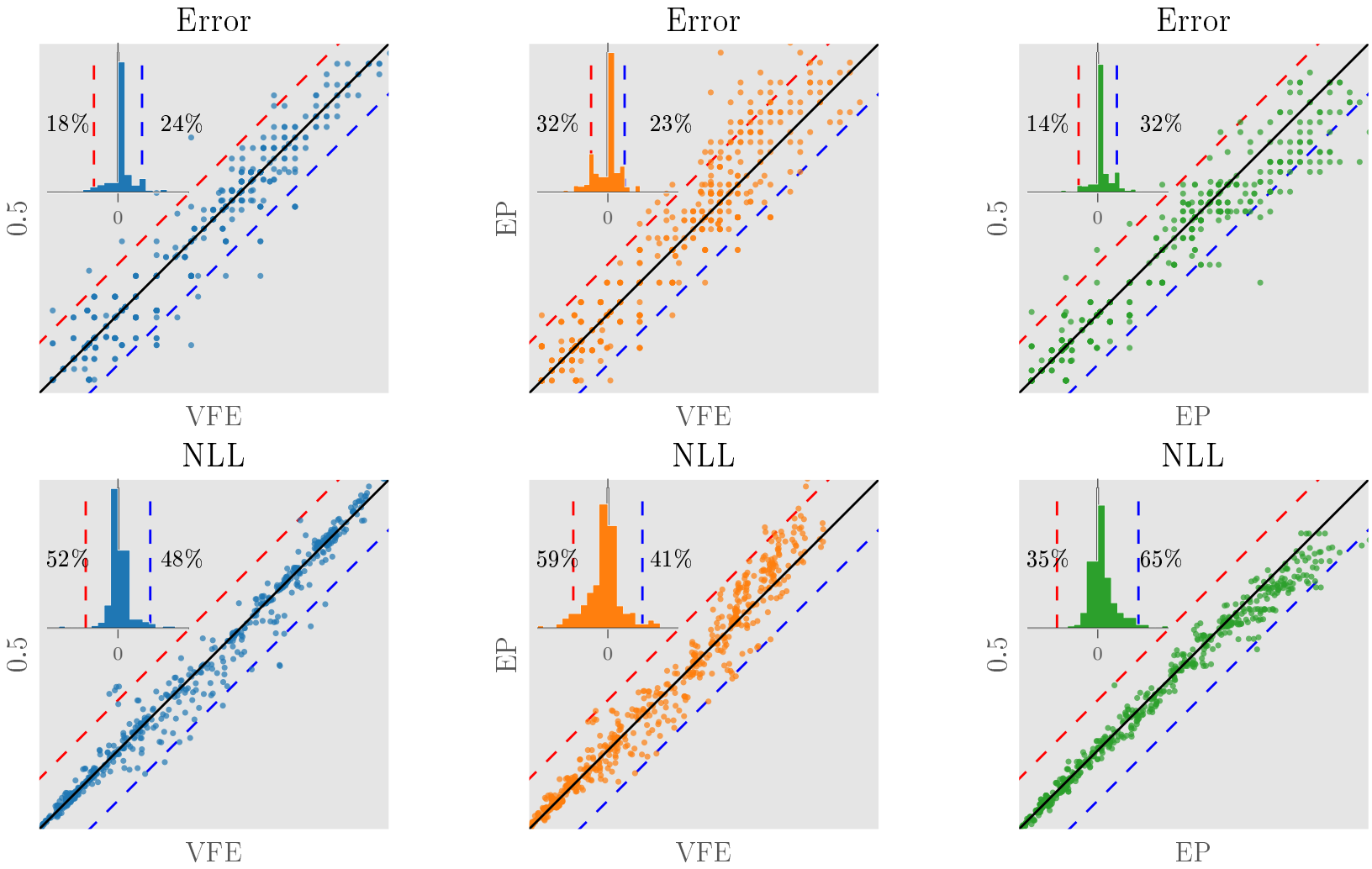


Fig. 2.6 Pair-wise comparisons between Power EP with $\alpha = 0.5$, EP ($\alpha = 1$) and VFE ($\alpha \to 0$), evaluated on several classification datasets and various settings of M. Each coloured point is the result of one split. Points that are below the diagonal line illustrate the method on the y-axis is better than the method on the x-axis. The inset diagrams show the histograms of the difference between methods (x-value $-$ y-value), and the counts of negative and positive differences. Note that this indicates the pairwise ranking of the two methods. Positive differences mean the y-axis method is better than the x-axis method and vice versa.

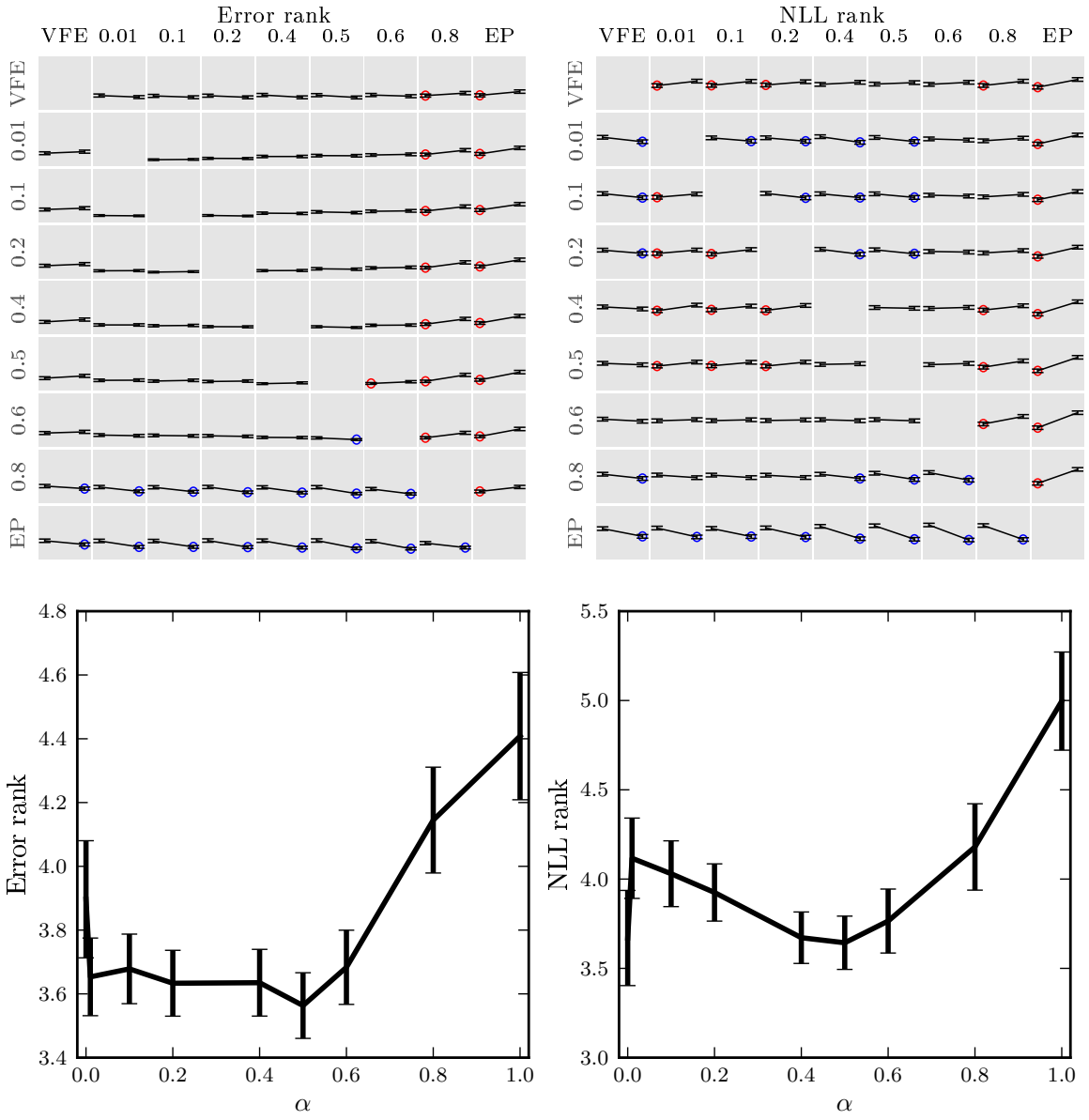


Fig. 2.7 Average ranking of various α values in the classification experiment, lower is better. Top plots show the pairwise comparisons. Red circles denote rows being better than the corresponding columns, and blue circles mean vice versa. Bottom plots show the ranks of all methods when being compared together. Intermediate α values (not EP or VFE) are best on average.

2.5 Discussion

The results presented above employed (approximate) type-II maximum likelihood fitting of the hyperparameters. This estimation method is known in some circumstances to overfit the

data. It is therefore conceivable therefore that pseudo-point approximations, which have a tendency to encourage under-fitting due to their limited representational capacity, could be beneficial due to them mitigating overfitting. We do not believe that this is a strong effect in the experiments above. For example, in the synthetic data experiments the NLML, SMSE and SMLL obtained from fitting the unapproximated GP were similar to those obtained using the GP from which the data were generated, indicating that overfitting is not a strong effect (see fig. 9 in the appendix). It is true that EP and $\alpha = 0.8$ over-estimates the marginal likelihood in the synthetic data experiments, but this is a distinct effect from over-fitting which would, for example, result in overconfident predictions on the test dataset. The SMSE and SMLL on the training and test sets, for example, are similar which is indicative of a well-fit model.

It is difficult to identify precisely where the best approximation methods derive their advantages, but here we will speculate. Since the negative variational free-energy is a lower-bound on the log-marginal likelihood, it has the enviable theoretical guarantee that pseudo-input optimisation is always guaranteed to improve the estimate of the log marginal likelihood and the posterior (as measured by the inclusive KL). The negative EP energy, in contrast, is not generally a lower bound which can mean that pseudo-input optimisation drives the solution to the point where the EP energy over-estimates the log marginal likelihood the most, rather than to the point where the marginal likelihood and/or posterior estimate is best. For this reason, we believe that variational methods are likely to be better than EP if the goal is to derive accurate marginal likelihood estimates, or accurate predictive distributions, for fixed hyperparameter settings. For hyperparameter optimisation, things are less clear-cut since variational methods are biased away from the maximal marginal likelihood, towards hyperparameter settings for which the posterior approximation is accurate. Often this bias is severe and also creates local-optima Turner and Sahani (2011). So, although EP will generally also be biased away from the maximal marginal likelihood and potentially towards areas of over-estimation, it can still outperform variational methods. Superposed onto these factors, is a general trend for variational methods to minimise MSE / classification error-rate and EP methods to minimise negative log-likelihood, due to the form of their respective energies (the variational free-energy includes the average training MSE in the regression case, for example). Intermediate methods will blend the strengths and weaknesses of the two extremes. It is interesting that values of α around a half are arguably the best performing on average. Similar empirical conclusions have been made elsewhere Minka (2005); Hernández-Lobato et al. (2016); Depeweg et al. (2016a). In this case, the α-divergence interpretation of Power EP shows that it is minimising the Hellinger distance whose square root is a valid distance metric. Further experiments and theoretical work are required to clarify these issues.

One of the features of the approximate generative models introduced in section 2.2.1 for regression, is that they contain input-dependent noise, unlike the original model. Many datasets contain noise of this sort and so approximate models like FITC and PITC, or models

in which the observation noise is explicitly modelled are arguably more appropriate than the original unapproximated regression model (Snelson, 2007; Saul et al., 2016). Motivated by this train of reasoning, Titsias (2009) applied the variational free-energy approximation to the FITC generative model an approach that was later generalised by Hoang et al. (2016) to encompass a more general class of input dependent noise, including Markov structure (Low et al., 2015). Here the insight is that the resulting variational lower bound separates over datapoints (Hensman et al., 2013) and is, therefore, amenable to stochastic optimisation using minibatches, unlike the marginal likelihood. In a sense, these approaches unify the approximate generative modelling approach, including the FITC and PITC variants, with the variational free-energy methods. Indeed, one approach is to posit the desired form of the optimal variational posterior, and to work backwards from this to construct the generative model implied (Hoang et al., 2016). However, these approaches are quite different from the one described in this chapter where FITC and PITC are shown to emerge in the context of approximating the original unapproximated GP regression model using Power EP. Indeed, if the goal really is to model input dependent noise, it is not at all clear that generative models like FITC are the most sensible. For example, FITC uses a single set of hyperparameters to describe the variation of the underlying function and the input dependent noise.

2.6 The approximate Power EP approach using tied factors

Power EP is a general and flexible framework for approximate inference and learning, and more importantly, intermediate α values have been shown in section 2.4 to be advantageous compared to VFE and EP. However, in the case when the optimal approximate posterior is not analytically tractable, e.g. probit classification, this flexibility does come at a cost:

- Hyperparameter updates and posterior inference need to be interleaved during learning, that is, there is no single objective function or procedure for learning both the hyperparameters and approximate posterior at the same time. Optimising the Power EP energy to obtain the approximate posterior alone (instead of running the iterative procedure) is also not straightforward, as non-standard, double-loop schemes needed to be deployed (Heskes and Zoeter, 2002). The VFE approach, on the other hand, provides a lower bound to the marginal likelihood, which can be optimised to concurrently learn both the hyperparameters and the approximate posterior. There are ways to side-step this problem, for example, not waiting for Power EP to converge before performing an update for the hyperparameters (Hernández-Lobato and Hernández-Lobato, 2016). However, this remains as an arguably major reason why Power EP is not used more widely in practice when hyperparameter optimisation is required.

- The sequential update nature of Power EP is problematic for large data sets, as multiple passes over the training data are needed for convergence. Parallel updates can be used

instead, but are prone to numerical problems. Techniques such as damping or skipping can be used (see e.g. Minka and Lafferty, 2002), but they are not sufficient for all cases.

- When the number of pseudo-points is large, the memory required to parameterise all the approximate factors is high and could be out of reach. Techniques such as average or stochastic EP (Li et al., 2015; Dehaene and Barthelmé, 2015) can significantly reduce this memory complexity. However, though this memory limitation is not a major focus of this chapter, it turns out that the trick employed in stochastic EP to reduce the memory constraint can be used to sidestep other problems.

As mentioned above, stochastic EP greatly reduces the memory complexity of Power EP. This is achived by using the same parameterisation for similar factors, enforcing an identical contribution from each factor to the posterior. The approximate posterior in eq. (2.13) becomes,

$$p^*(f|\mathbf{y},\theta) = p(f|\mathbf{y},\theta)p(\mathbf{y}|\theta) = p(f|\theta)\prod_n p(y_n|f,\theta) \approx p(f|\theta)\prod_n t(\mathbf{u}) = p(f|\theta)t^N(\mathbf{u}) = q^*_{\mathrm{APEP}}(f|\theta).$$

Each common factor $t(\mathbf{u})$ could be thought of as the average contribution from each datapoint towards the posterior. Having described an approximate factorisation, one could proceed to run the Power EP iterative procedure (Li et al., 2015), and then perform hyperparameter optimisation as an outer loop as with Power EP. However, Hernández-Lobato et al. (2016) noticed that the factor tying approximation above turns the original minimax Power EP energy optimisation problem into a minimisation problem. In other words, in a similar fashion to the VFE approach, the approximate Power EP energy can now be optimised using standard optimisation techniques, to find *both* the approximate posterior and the hyperparameters.

We assess the performance of directly optimising the approximate Power-EP (APEP), suggested by Hernández-Lobato et al. (2016), on the same regression and classification tasks considered in section 2.4. We compare the performance of APEP with PEP that uses the same α-value, and show the pairwise comparison between APEP and PEP in figs. 2.8 and 2.9 for the regression and classification cases, respectively.

In the regression case, we have shown in section 2.3.6 that the optimal approximate posterior can be obtained for fixed hyperparameters and as a result, the *collapsed* Power-EP energy can be directly optimised to obtain the hyperparameters and the pseudo-inputs. On the other hand, when the approximate Power-EP energy with the tied factor constraint is minimised, it is not clear what the optimal approximate posterior would be and that the average factor would be rank-one. Therefore the approximate energy needs to be optimised with respect to the hyperparameters, the pseudo-inputs, *and* the average factor (or to be more precise, the parameters used to parameterise the mean and the covariance of the average factor). Figure 2.8 demonstrates that optimising the collapsed Power-EP energy is superior to using the approximate uncollapsed Power-EP energy. This trend is also consistent across

different α values. The results here suggest that if a collapsed bound/energy is available, one ought to use it instead of the uncollapsed version.

In the classification case, the differences between PEP and APEP are i. the form of the approximate factors (PEP uses rank-one factors while APEP uses a full-rank average factor), and ii. how the approximate factors are obtained (PEP uses its iterative procedure while APEP employs direct optimisation of the energy). In contrast to the regression case, fig. 2.9 shows that there is no marked difference between APEP and PEP in both error and NLL metrics and across different α values. This suggests that for classification tasks, one could use the APEP method to enjoy the practical advantages it provides, without degrading the predictive performance when compared to the exact PEP solution.

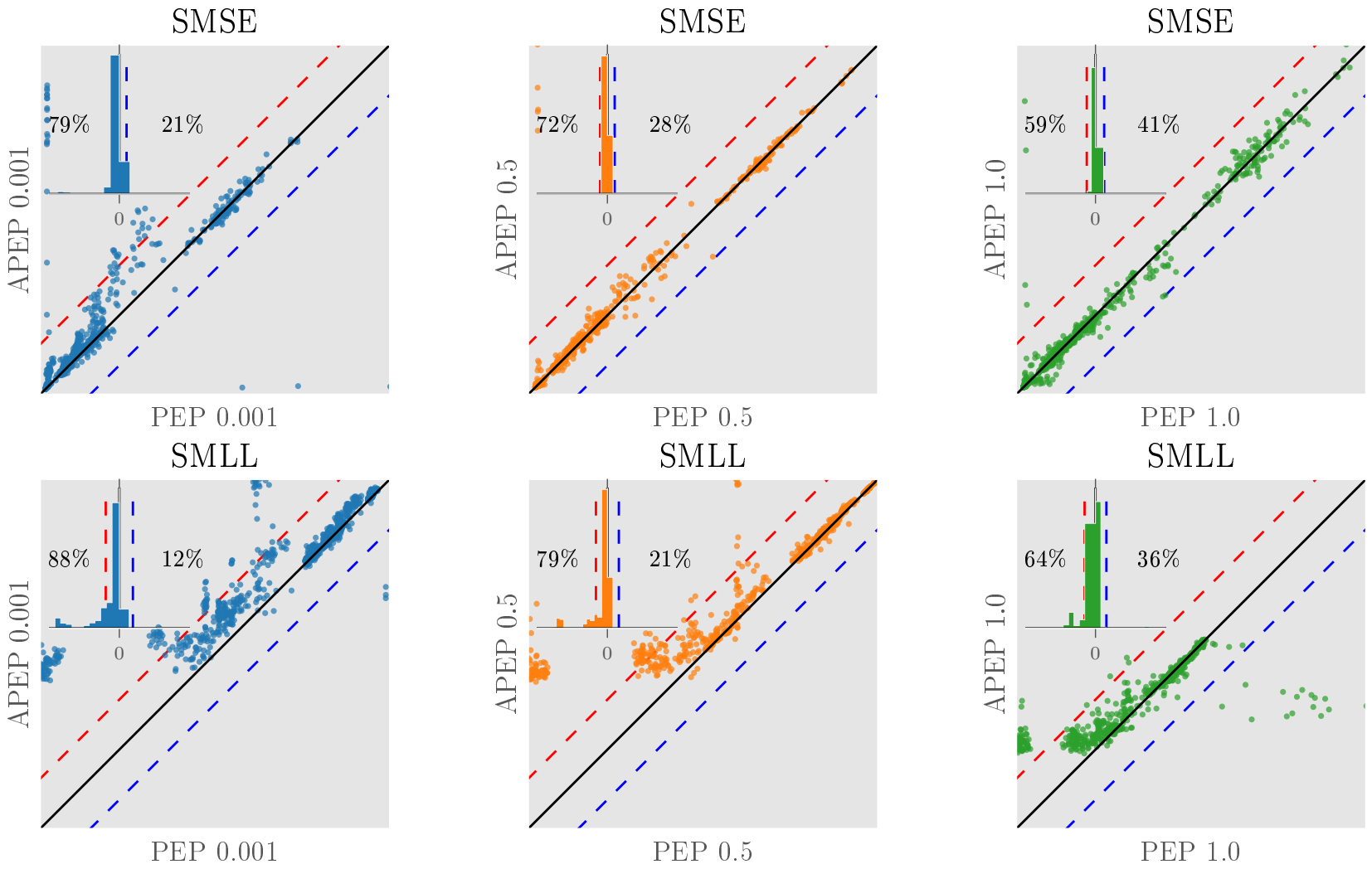


Fig. 2.8 Comparisons between Power EP with $\alpha = 0.001$, $\alpha = 0.5$, and EP (Power EP $\alpha = 1$), with their approximate counterpart (APEP) using the same α value, evaluated on several regression datasets and various settings of M. Each coloured point is the result of one split. Points that are below the diagonal line illustrate the method on the y-axis is better than the method on the x-axis. The inset diagrams show the histograms of the difference between methods (x-value $-$ y-value), and the counts of negative and positive differences. Note that this indicates the pairwise ranking of the two methods. Positive differences mean the y-axis method is better than the x-axis method and vice versa. For example, the middle, bottom plot shows PEP with $\alpha = 0.5$ is on average better than APEP with $\alpha = 0.5$.

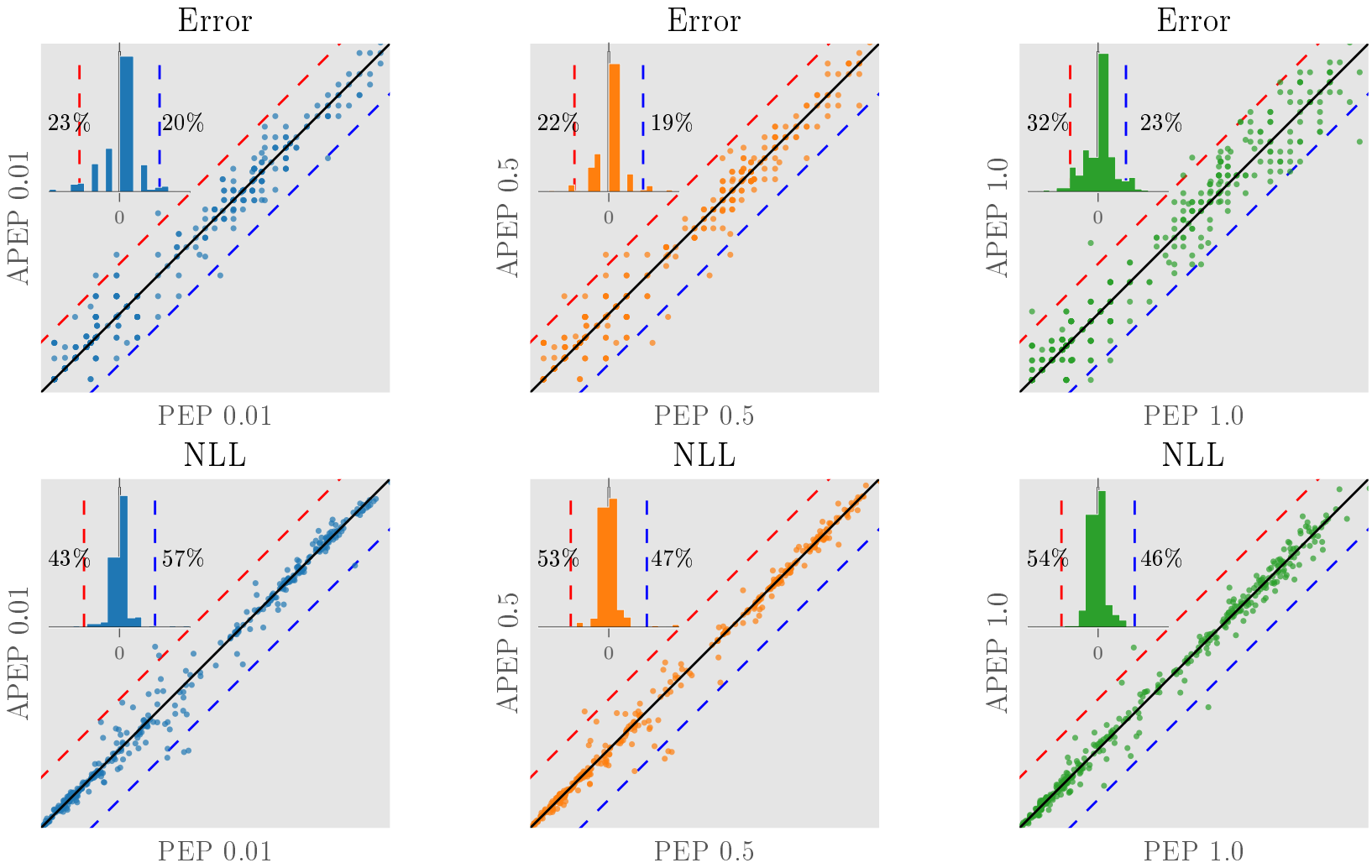


Fig. 2.9 Comparisons between Power EP with $\alpha = 0.01$, $\alpha = 0.5$, and EP (Power EP $\alpha = 1$), with their approximate counterpart using the same α value, evaluated on several classification datasets and various settings of M. Each coloured point is the result of one split. Points that are below the diagonal line illustrate the method on the y-axis is better than the method on the x-axis. The inset diagrams show the histograms of the difference between methods (x-value $-$ y-value), and the counts of negative and positive differences. Note that this indicates the pairwise ranking of the two methods. Positive differences mean the y-axis method is better than the x-axis method and vice versa.

2.7 Summary

This chapter provided a new unifying framework for GP pseudo-point approximations based on Power EP that subsumes many previous approaches including FITC, PITC, DTC, Titsias's VFE method, Qi et al's EP method, and inter-domain variants. It provided a clean computational perspective on the seminal work of Csató and Opper that related FITC to EP, before extending their analysis significantly to include a closed form Power EP marginal likelihood approximation for regression, connections to PITC, and further results on classification and GPSSMs. The new framework was used to devise new algorithms for GP regression and GP classification. Extensive experiments indicate that intermediate values of Power EP with the power parameter set to $\alpha = 0.5$ often outperform the state-of-the-art EP and VFE approaches. The new framework suggests many interesting directions for future work in this area that we have not explored, for example, extensions to online inference, combinations with special structured matrices (e.g. circulant and Kronecker structure), Bayesian hyperparameter learning, and applications to richer models. The current work has only scratched the surface, but we believe that the new framework will form a useful theoretical foundation for the next generation of GP approximation schemes.

Chapter 3

Sparse approximations for Gaussian process state space and latent variable models

3.1 Introduction

Unsupervised learning of underlying latent parameters and variables governing observed data is central to many machine learning settings. This type of learning typically involves building a generative probabilistic model for observed data and performing learning and inference over the unobserved or latent variables. Such a procedure allows us to define a density model for the observations and from a probabilistic perspective, learning in this model is equivalent to finding a model that maximises the probability of the data. Importantly, such a model of the data enables us to handle tasks in the data space such as denoising noisy observations, synthesising new data or imputing missing data, or to compress high-dimensional data into low-dimensional latent factors for summarisation, analysis and visualisation purposes (for example identifying style and content latent factors).

Inference in this class of models is ill-posed, as there are potentially many ways to explain all the patterns and ambiguity in the observed data. As such, a good probabilistic model needs to be flexible to allow a rich generative power to explain the data.[1] Importantly, a good learning and inference algorithm needs to produce accurate predictions of unseen data with reliable and calibrated uncertainty estimates. Unfortunately, the gold standard exact Bayesian learning and inference paradigm is analytically and computationally intractable in many interesting and complicated generative models, forcing us to consider approximation techniques. This chapter considers a class of generative models based on continuous latent variables, in which the relationship between the latent and the observed variables, or between

[1] However, at the same time it cannot be too flexible since we then end up with delta functions at each observed sample and the model will not be able to generalise to unseen data

the latent variables themselves are flexibly modelled by a Gaussian process, and derives a practical and tractable deterministic approximate Bayesian inference and learning framework for these models.

The first part of this chapter reviews the Gaussian process state space model (GPSSM) to model time-series data and an existing variational free-energy (VFE) approximation to learning and inference. This is followed by a derivation of a novel approximation framework based on Power Expectation Propagation (Power EP), and an approximation to this framework allowing a tractable implementation and deployment in practice. A special case of this model class to non-time-series data, namely the Gaussian process latent variable model (GPLVM), will be discussed. Several applications including non-linear system identification, and structure discovery and visualisation for neuroscience data are used as evaluation testbeds.

3.2 The Gaussian process state space model

Complex non-linear dynamics arise in many fields of science and engineering, and are abundant in many machine learning tasks such as sequence modelling, control, and quantitative finance. The ability to model and learn complex time series dynamics directly from observations is a key problem in many disciplines. A key difference to other types of data, and also a challenge, is that observed data or measurements are potentially explicitly dependent on past observations. While this dependency can be implicitly encoded by simpler techniques like regression[2], building an explicit model to reflect our belief about this dependency is arguably a better choice.

There are a plethora of existing approaches to time series modelling, notably autoregressive models and state space models. Autoregressive models effectively are regression models, taking past observations as inputs to predict the current observations. While these models are simple and often the first port of call for many practitioners, they suffer from the problem of mixing input and measurement noises. State space models are probabilistic models describing the evolution of a state space or latent variables through time, and how the measurements at each time step are related to these variables. These latter models are potentially more flexible, as the learnt latent space could be viewed as a compressed state of high-dimensional observations, or as unobserved factors that potentially govern the underlying dynamics. Importantly, this class of models separates the measurement noise from the potentially noisy underlying dynamics, unlike autoregressive models in which input noise is not typically explicitly model (see Frigola, 2015, sec. 2.4). This flexibility, however, does come at a cost of less tractable inference and learning, especially for non-linear dynamics. In contrast, standard regression techniques can be readily deployed for inference and learning in the autoregressive variants.

[2]for example: in GP regression, the observations are correlated when the latent function is marginalised out.

We are interested in a state-space model subclass in which latent variables are continuous and the transition between these variables is non-linear. Due to the non-linearity in the dynamics, inference and learning, which involve finding the state transition as well as the latent states themselves, are difficult. The presence of transition and measurement noise, and a typically small number of measurements in practice make the inference task even more challenging. This model subclass can be compactly represented as follows,

$$p(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0; \mu_0, \Sigma_0), \tag{3.1}$$

$$p(\mathbf{x}_t|f, \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; f(\mathbf{x}_{t-1}), \mathbf{Q}_x), \quad \text{for } t = 1:T, \tag{3.2}$$

$$p(\mathbf{y}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{y}_t; \mathbf{U}\mathbf{x}_t, \mathbf{R}_y), \quad \text{for } t = 1:T, \tag{3.3}$$

where the dynamical noise is assumed Gaussian and the measurement or emission model is assumed to be linear and Gaussian, $\mathbf{x}$ and $\mathbf{y}$ are the latent variables and the measurements respectively, f is a continuous non-linear transition function, and T is the number of measurement steps. The idea of using non-linear function f to model the dynamic is not new, for example, classic work by Ito and Xiong (2000); Kushner and Budhiraja (2000) derived approximations for filtering with generic non-linear dynamics, Ghahramani and Roweis (1998) employed an RBF network, and Valpola and Karhunen (2002) used a (Bayesian) neural network.

In this chapter, we assume the non-linear transition function f is a draw from a Gaussian process, or in other words, we place a flexible nonparametric Gaussian process prior over this function, $p(f) = \mathcal{GP}(m(.), k(.,.))$. We refer to the resulting model as the Gaussian process state-space model (GPSSM). This model belongs to a plethora of closely related dynamical systems which describe how the state and emission variables evolve using Gaussian processes (Frigola, 2015, Chapter 2). For simplicity, we assume a linear Gaussian emission model, however, the inference techniques described in this chapter can be applied for non-Gaussian and non-linear likelihoods. In fact, a variant that uses a non-linear GP emission model is used later in the experiment.

The joint probablity of all variables and observations involved can be written as follows,

$$p(\mathbf{y}_{1:T}, \mathbf{x}_{0:T}, f|\theta) = p(\mathbf{x}_0)p(f|\theta)\prod_{t=1}^{T} p(\mathbf{x}_t|f, \mathbf{x}_{t-1}, \theta)p(\mathbf{y}_t|\mathbf{x}_t, \theta), \tag{3.4}$$

where θ includes the kernel hyperparameters, the transition noise, and the emission parameters. These hyperparameters can be found by performing model selection using the log marginal likelihood:

$$\mathcal{L}(\theta) = \log p(\mathbf{y}_{1:T}|\theta) = \log \int \mathrm{d}f \mathrm{d}\mathbf{x}_{0:T} p(\mathbf{y}_{1:T}, \mathbf{x}_{0:T}, f|\theta). \tag{3.5}$$

Unfortunately, the above marginalisation is analytically intractable due to the non-linearity in the model. An intertwined and equally difficult problem is to obtain the posterior over the latent transition and state variables,

$$p(\mathbf{x}_{0:T}, f|\mathbf{y}_{1:T}, , \theta) = \frac{p(\mathbf{y}_{1:T}, \mathbf{x}_{0:T}, f|\theta)}{p(\mathbf{y}_{1:T}|\theta)}. \tag{3.6}$$

The difficult task of inferring the latent transition function f and the latent variables $\mathbf{x}$ and performing model selection as explained above has been previously studied and is an active research area. Wang et al. (2005) obtains the *maximum-a-posterior* solutions for both f and $\mathbf{x}$ and, as a consequence, does not present a proper treatment of the uncertainty in the model. Frigola et al. (2013) attempted to remedy this issue in a fully Bayesian scheme based on particle MCMC, at the cost of added computational complexity. More recently, Frigola et al. (2014) introduced an inference scheme mixing variational method and sequential Monte Carlo, the core of which is based on earlier work on sparse GP regression by Titsias (2009). A fully variational treatment has then been studied in McHutchon (2014), which was further extended by using recent developments in neural network based recognition models by Eleftheriadis et al. (2017).

In this section, we first review the variational free-energy approaches by Frigola et al. (2014); McHutchon (2014), and then present a more general approximate inference scheme based on Power Expectation Propagation (Power EP), capable of learning both the transition function f and latent variables $\mathbf{x}$. EP has been considered for learning and inference for GPSSMs. For example, the inference approach introduced in Deisenroth and Mohamed (2012) assumes a known transition dynamic f, and only infers an approximate posterior over $\mathbf{x}$. This EP scheme was later used as an E-step in an EM approach to learning both f and $\mathbf{x}$ (McHutchon, 2014). Crucially, the lack of an additional approximation for the latent function f in this EP scheme results in a prohibitive computational complexity of $\mathcal{O}(T^3)$. The approach proposed in this section, in contrast, employs Power EP to provide approximate Bayesian estimates for *both* f and $\mathbf{x}$ simultaneously in a computationally and analytically tractable manner. Importantly, Power EP offers a flexible approximate inference framework which has EP and structured variational inference (VI) as special cases. This framework generalises the Power EP framework for regression and classification discussed in the last chapter and can be extended to other non-GP based state space models. We attempt to describe the relationship of the methods described in this chapter by the poster approximation and the approximate inference scheme that each method uses, in table 3.1.

3.3 The VFE approach

The difficulty of exact inference and learning described above can be sidestepped by using deterministic approximations. One example is the variational free energy method, which turns

Table 3.1 Different approximations discussed in this chapter, categorised by the approximate posterior and the inference method used. In all cases (except MAP), $q(\mathbf{u})$ is assumed Gaussian, $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{m}, \mathbf{S})$.

Sec.	Inf. method	$q(\mathbf{x}_{0:T})$	Notes and references
—	MAP	MAP	Wang et al. (2005)
—	VFE + SMC	sample-based	sandwiching VFE updates for $q(\mathbf{u})$ and SMC for $q(\mathbf{x}_{0:T})$, Frigola et al. (2014)
3.3	VFE	Gaussian, (block)-diagonal covariance	$q(\mathbf{u})$ collapsable
3.4	Power EP	Gaussian, (block)-diagonal covariance	factors can be tied, $\alpha \to 0$ gives VFE
3.3	VFE	Gaussian, tri-(block)-diagonal precision	$q(\mathbf{u})$ collapsable, McHutchon (2014), Eleftheriadis et al. (2017) used inf. networks to parameterise $q(\mathbf{x}_{0:T})$
3.4	Power EP	Gaussian, tri-(block)-diagonal precision	$\alpha \to 0$ gives VFE
3.10.1	VFE and Power EP	$q(\mathbf{x}_{0:T}\|f) = q(\mathbf{x}_0)\prod_{t=1}^{T} p(\mathbf{x}_t\|f, \mathbf{x}_{t-1})$	inspired by Salimbeni and Deisenroth (2017) and section 4.4

the intractable inference problem into a simpler optimisation problem (Frigola et al., 2014; McHutchon, 2014). In the context of the GPSSMs, this approach introduces a variational approximation for the latent function f and the latent variables $\mathbf{x}_{0:T}$, $q(f, \mathbf{x}_{0:T})$, to lower-bound the log marginal likelihood in eq. (3.5) using Jensen's inequality as follows,

$$\mathcal{L}(\theta) = \log \int \mathrm{d}f \mathrm{d}\mathbf{x}_{0:T} p(\mathbf{y}_{1:T}, \mathbf{x}_{0:T}, f|\theta) \tag{3.7}$$

$$= \log \int \mathrm{d}f \mathrm{d}\mathbf{x}_{0:T} \frac{q(f, \mathbf{x}_{0:T})}{q(f, \mathbf{x}_{0:T})} p(\mathbf{y}_{1:T}, \mathbf{x}_{0:T}, f|\theta) \tag{3.8}$$

$$\geq \int \mathrm{d}f \mathrm{d}\mathbf{x}_{0:T} q(f, \mathbf{x}_{0:T}) \log \frac{p(\mathbf{y}_{1:T}, \mathbf{x}_{0:T}, f|\theta)}{q(f, \mathbf{x}_{0:T})} =: \mathcal{F}_{\mathrm{vfe}}(q(.), \theta), \tag{3.9}$$

where $F(q(.), \theta)$ is the negative variational free-energy or the variational lower bound to the marginal likelihood. The gap between the exact marginal likelihood and this bound can be shown as the KL-divergence from the exact posterior to the variational approximation, $\mathrm{KL}(q(f, \mathbf{x}_{0:T})||p(f, \mathbf{x}_{0:T}|\mathbf{y}, \theta))$. This gap becomes zero and the inequality above becomes equality when $q(f, \mathbf{x}_{0:T}) = p(f, \mathbf{x}_{0:T}|\mathbf{y}, \theta)$; however, this is intractable. Instead, a variational approximation is chosen from a simpler and restricted family and the resulting variational free-energy is maximised with respect to this approximation so that it gets closer to the intractable posterior (as the KL divergence is minimised). Most importantly, the variational distribution must be chosen to be rich and expressive to approximate the ground truth well, and to allow tractable computations. Note that, as the variational free energy approximates the log marginal likelihood, it can be optimised to obtain the model hyperparameters θ.

Inspired by the form of the structured variational approximation used in sparse approximations for GP regression and GPLVMs by Titsias (2009); Titsias and Lawrence (2010), Frigola et al. (2014) employed the following variational approximation,

$$q(f, \mathbf{x}_{0:T}) = p(f_{\neq \mathbf{u}}|\mathbf{u}) q(\mathbf{u}) q(\mathbf{x}_{0:T}). \tag{3.10}$$

This assumes a mean-field approximation between $\mathbf{x}_{0:T}$ and f, and mirrors the form of the prior on the latent function f,

$$p(f) = p(f_{\neq \mathbf{u}}|\mathbf{u}) p(\mathbf{u}). \tag{3.11}$$

This elegantly allows a cancellation of $p(f_{\neq \mathbf{u}}|\mathbf{u})$, leading to a tractable negative variational free-energy as follows,

$$\mathcal{F}_{\text{vfe}}(.) = \int \mathrm{d}f \mathrm{d}\mathbf{x}_{0:T} q(f, \mathbf{x}_{0:T}) \log \frac{p(\mathbf{y}_{1:T}|\mathbf{x}_{0:T}, f) p(\mathbf{x}_{0:T}, f)}{q(f, \mathbf{x}_{0:T})} \tag{3.12}$$

$$= \int \mathrm{d}f \mathrm{d}\mathbf{x}_{0:T} q(f, \mathbf{x}_{0:T}) \log \frac{p(\mathbf{y}_{1:T}|\mathbf{x}_{1:T}) p(\mathbf{x}_{1:T}|f, \mathbf{x}_0) \cancel{p(f_{\neq \mathbf{u}}|\mathbf{u})} p(\mathbf{u}) p(\mathbf{x}_0)}{\cancel{p(f_{\neq \mathbf{u}}|\mathbf{u})} q(\mathbf{u}) q(\mathbf{x}_{0:T})} \tag{3.13}$$

$$= -\mathrm{KL}(q(\mathbf{u})||p(\mathbf{u})) + \mathcal{H}(q(\mathbf{x}_{0:T})) + \int \mathrm{d}\mathbf{x}_0 q(\mathbf{x}_0) \log p(\mathbf{x}_0)$$
$$+ \sum_{t=1}^{T} \int \mathrm{d}\mathbf{x}_t q(\mathbf{x}_t) \log p(\mathbf{y}_t|\mathbf{x}_t) + \sum_{t=1}^{T} \int \mathrm{d}\mathbf{x}_{t-1,t} \mathrm{d}f q(\mathbf{x}_{t-1,t}) q(f) \log p(\mathbf{x}_t|f, \mathbf{x}_{t-1}), \tag{3.14}$$

where $\mathcal{H}(q)$ is the entropy of the density q, and note that we have used the model description in eqs. (3.2) and (3.3) to decompose parts of the energy into sums across time steps.

3.3.1 Obtaining an optimal $q(\mathbf{u})$

In the derivations above, we have not made any assumptions about the form or family of $q(\mathbf{u})$ and $q(\mathbf{x}_{0:T})$. In fact, it could be shown that the optimal form for $q(\mathbf{u})$ is a Gaussian density whose parameters depend on $q(\mathbf{x}_{0:T})$. In detail, taking the functional derivative of $\mathcal{F}_{\text{vfe}}(.)$ w.r.t. $q(\mathbf{u})$ gives us,

$$\frac{\delta \mathcal{F}_{\text{vfe}}(.)}{\delta q(\mathbf{u})} = -\log \frac{q(\mathbf{u})}{p(\mathbf{u})} - 1 + \sum_{t=1}^{T} \underbrace{\int \mathrm{d}\mathbf{x}_{t-1,t} \mathrm{d}f q(\mathbf{x}_{t-1,t}) p(f_{\neq \mathbf{u}}|\mathbf{u}) \log p(\mathbf{x}_t|f(\mathbf{x}_{t-1}))}_{\mathcal{G}_t}. \tag{3.15}$$

Setting eq. (3.15) to zero and making sure that $q(\mathbf{u})$ is normalised results in the following optimal variational distribution,

$$q^*(\mathbf{u}) = \frac{p(\mathbf{u}) \exp(\sum_{t=1}^{T} \mathcal{G}_t)}{\mathcal{Z}_{\mathbf{u}}}, \text{ where } \mathcal{Z}_{\mathbf{u}} = \int \mathrm{d}\mathbf{u} p(\mathbf{u}) \exp\left(\sum_{t=1}^{T} \mathcal{G}_t\right). \tag{3.16}$$

The integral over the entire function f in $\mathcal{G}_t$ can be simplified to only over a single function value, $f(\mathbf{x}_{t-1})$, since the transition log likelihood, $p(\mathbf{x}_t|f(\mathbf{x}_{t-1}))$, only depends on this term. The log of this likelihood term, $\log p(\mathbf{x}_t|f(\mathbf{x}_{t-1}))$, and the conditional distribution $p(f(\mathbf{x}_{t-1})|\mathbf{u})$ can be expanded as follows,

$$\log p(\mathbf{x}_t|f(\mathbf{x}_{t-1})) = \log \mathcal{N}(\mathbf{x}_t; f(\mathbf{x}_{t-1}), \mathbf{Q}) \tag{3.17}$$

$$= -\frac{D}{2} \log(2\pi) - \frac{1}{2} \log |\mathbf{Q}| - \frac{1}{2} (\mathbf{x}_t - f(\mathbf{x}_{t-1}))^\intercal \mathbf{Q}^{-1} (\mathbf{x}_t - f(\mathbf{x}_{t-1})), \tag{3.18}$$

$$p(f(\mathbf{x}_{t-1})|\mathbf{u}) = \mathcal{N}(f(\mathbf{x}_{t-1}); \mathbf{A}_t \mathbf{u}; \mathbf{B}_t), \tag{3.19}$$

where $\mathbf{A}_t = \mathbf{K}_{\mathrm{f}_t\mathbf{u}}\mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1}$, $\mathbf{B}_t = \mathbf{K}_{\mathrm{f}_t\mathrm{f}_t} - \mathbf{K}_{\mathrm{f}_t\mathbf{u}}\mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u}\mathrm{f}_t}$, $\mathbf{K}_{\mathrm{f}_t\mathbf{u}}$ is the covariance matrix between the function values at the previous latent state $\mathbf{x}_{t-1}$ and the pseudo-inputs $\mathbf{z}$, and $K_{\mathrm{f}_t\mathrm{f}_t}$ the covariance between the function value at $\mathbf{x}_{t-1}$ and itself. As a result, $f(\mathbf{x}_{t-1})$ can be analytically integrated out as follows,

$$\begin{aligned}
\mathcal{G}_t &= \int \mathrm{d}\mathbf{x}_{t-1,t}\mathrm{d}f\, q(\mathbf{x}_{t-1,t})p(f_{\neq\mathbf{u}}|\mathbf{u})\log p(\mathbf{x}_t|f(\mathbf{x}_{t-1})) && (3.20)\\
&= \int \mathrm{d}\mathbf{x}_{t-1,t}q(\mathbf{x}_{t-1,t})\left[\int \mathrm{d}f(\mathbf{x}_{t-1})p(f(\mathbf{x}_{t-1})|\mathbf{u})\log p(\mathbf{x}_t|f(\mathbf{x}_{t-1}))\right] && (3.21)\\
&= \int \mathrm{d}\mathbf{x}_{t-1,t}q(\mathbf{x}_{t-1,t})\left[\log\mathcal{N}(\mathbf{x}_t;\mathbf{A}_t\mathbf{u},\mathbf{Q}) - \frac{1}{2}\operatorname{tr}(\mathbf{Q}^{-1}\mathbf{B}_t)\right] && (3.22)\\
&= -\frac{D}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{Q}| - \frac{1}{2}\operatorname{tr}\left(\mathbf{Q}^{-1}\left[\langle\mathbf{B}_t\rangle_{q(\mathbf{x}_{t-1})} + \langle\mathbf{x}_t^\intercal\mathbf{x}_t\rangle_{q(\mathbf{x}_t)}\right]\right)\\
&\quad - \frac{1}{2}\mathbf{u}^\intercal\langle\mathbf{A}_t^\intercal\mathbf{Q}^{-1}\mathbf{A}_t\rangle_{q(\mathbf{x}_{t-1})}\mathbf{u} + \mathbf{u}^\intercal\langle\mathbf{A}_t^\intercal\mathbf{Q}^{-1}\mathbf{x}_t\rangle_{q(\mathbf{x}_{t-1,t})}. && (3.23)
\end{aligned}$$

As $\mathcal{G}_t$ forms a quadratic in $\mathbf{u}$, substituting this in the optimal variational distribution $q(\mathbf{u})$ above gives a closed form Gaussian distribution $q^*(\mathbf{u}) = \mathcal{N}(\mathbf{u};\mathbf{m}_*,\mathbf{S}_*)$, where,

$$\mathbf{S}_*^{-1} = \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} + \sum_{t=1}^{T}\langle\mathbf{A}_t^\intercal\mathbf{Q}^{-1}\mathbf{A}_t\rangle_{q(\mathbf{x}_{t-1})}, \tag{3.24}$$

$$\mathbf{S}_*^{-1}\mathbf{m}_* = \sum_{t=1}^{T}\langle\mathbf{A}_t^\intercal\mathbf{Q}^{-1}\mathbf{x}_t\rangle_{q(\mathbf{x}_{t-1,t})} \tag{3.25}$$

Note that the parameters of this optimal variational distribution depend on the variational approximation $q(\mathbf{x}_{1:T})$, or more precisely, on the marginal approximations at each time step $q(\mathbf{x}_t)$ and the pair-wise marginal approximations at consecutive time steps $q(\mathbf{x}_{t-1,t})$. The analytic tractability now depends on being able to compute the expectations in eqs. (3.24) and (3.25). These expectations are available in closed-form when the approximation for the hidden variables is Gaussian and the covariance function used is an exponentiated quadratic kernel, linear kernel, or a linear mixture of these. Detailed derivation of these expectations can be found in the appendix of McHutchon (2014). For covariance functions that do not admit tractable expectation computation, approximations such as simple Monte Carlo can be used. Additionally, the log of the normalising constant $\mathcal{Z}_\mathbf{u}$ in eq. (3.16) can also be obtain analytically,

$$\begin{aligned}
\log\mathcal{Z}_\mathbf{u} = &-\frac{DT}{2}\log(2\pi) - \frac{T}{2}\log|\mathbf{Q}| - \frac{1}{2}\operatorname{tr}\left(\mathbf{Q}^{-1}\sum_{t=1}^{T}\left[\langle\mathbf{B}_t\rangle_{q(\mathbf{x}_{t-1})} + \langle\mathbf{x}_t^\intercal\mathbf{x}_t\rangle_{q(\mathbf{x}_t)}\right]\right)\\
&- \frac{1}{2}\log|\mathbf{K}_{\mathbf{u}\mathbf{u}}| + \frac{1}{2}\log|\mathbf{S}_*| + \frac{1}{2}\mathbf{m}_*^\intercal\mathbf{S}_*^{-1}\mathbf{m}_*. && (3.26)
\end{aligned}$$

3.3.2 Choosing a variational family for $q(\mathbf{x}_{0:T})$

Having shown that the optimal variational distribution $q(\mathbf{u})$ takes a Gaussian form, we now turn our attention to finding an optimal form for $q(\mathbf{x}_{0:T})$. Following a similar procedure as for $q(\mathbf{u})$, we can find the functional derivative of the free-energy w.r.t. $q(\mathbf{x}_{0:T})$ as follows,

$$\frac{\delta \mathcal{F}_{\text{vfe}}(.)}{\delta q(\mathbf{x}_{0:T})} = -1 - \log q(\mathbf{x}_{0:T}) + \int \mathrm{d}f q(f) \log \frac{p(\mathbf{y}_{1:T}|\mathbf{x}_{1:T})p(\mathbf{x}_{1:T}|f,\mathbf{x}_0)p(\mathbf{u})p(\mathbf{x}_0)}{q(\mathbf{u})} \tag{3.27}$$

$$= \mathrm{C} - \log q(\mathbf{x}_{0:T}) + \log p(\mathbf{x}_0) + \log p(\mathbf{y}_t|\mathbf{x}_t) + \sum_{t=1}^{T} \int \mathrm{d}f q(f) \log p(\mathbf{x}_t|f,\mathbf{x}_{t-1}), \tag{3.28}$$

where C is a constant that can be folded into the normaliser of $q(\mathbf{x}_{0:T})$. The integral in the last term above can be computed in closed-form,

$$\int \mathrm{d}f q(f) \log p(\mathbf{x}_t|f,\mathbf{x}_{t-1}) = \int \mathrm{d}f(\mathbf{x}_{t-1}) \mathrm{d}\mathbf{u} p(f(\mathbf{x}_{t-1})|\mathbf{u}) q(\mathbf{u}) \log p(\mathbf{x}_t|f,\mathbf{x}_{t-1}) \tag{3.29}$$

$$= \log \mathcal{N}(\mathbf{x}_t; \mathbf{A}_t\mathbf{m}, \mathbf{Q}) - \frac{1}{2}\operatorname{tr}(\mathbf{Q}^{-1}[\mathbf{B}_t + \mathbf{A}_t\mathbf{S}\mathbf{A}_t^\intercal]), \tag{3.30}$$

where $\mathbf{m}$ and $\mathbf{S}$ are the mean and covariane of $q(\mathbf{u})$. Substituting the above result into the gradient and setting this gradient to zero lead to the following optimal variational approximation for the latent variables,

$$q^*(\mathbf{x}_{0:T}) \propto p(\mathbf{x}_0) \prod_{t=1}^{T} \left[p(\mathbf{y}_t|\mathbf{x}_t) \mathcal{N}(\mathbf{x}_t; \mathbf{A}_t\mathbf{m}, \mathbf{Q}) \exp(-\frac{1}{2}\operatorname{tr}(\mathbf{Q}^{-1}[\mathbf{B}_t + \mathbf{A}_t\mathbf{S}\mathbf{A}_t^\intercal])) \right]. \tag{3.31}$$

Note that $\mathbf{A}_t$ and $\mathbf{B}_t$ depends on $\mathbf{x}_{t-1}$ and as a result, the optimal form above possesses a Markovian structure despite a non-trivial structure in the exact and intractable posterior. However, since $\mathbf{A}_t$ and $\mathbf{B}_t$ have non-linear dependencies on $\mathbf{x}_{t-1}$, this optimal approximate posterior is also *not* analytically tractable. Approximation techniques can be used to sidestep this difficulty, for example: Frigola et al. (2014) used *Sequential Monte Carlo* (SMC) to obtain samples from this distribution, and McHutchon (2014) employed a deterministic Gauss-Markov structure $q(\mathbf{x}_{0:T}) = \mathcal{N}(\mathbf{x}_{0:T}; \mu, \Sigma)$ where Σ^{-1} is tridiagonal or block-tridiagonal. In this chapter, we experiment with the Markovian Gaussian approximation of McHutchon (2014), and a mean-field Gaussian approximation where Σ is diagonal or block-diagonal. Note that the computation of the variational free-energy requires several expectations w.r.t. the marginal densities over the latent variables, and the pair-wise densities at successive time steps, and a computation of the entropy of $q(\mathbf{x}_{0:T})$. We will discuss how the (block-)diagonal and structured Gassian approximations allow quick access to these densities and also permit efficient entropy computation.

In addition, the optimal form for $q(\mathbf{x}_{0:T})$ above depends on the parameters of the variational distribution over the pseudo-datapoints, $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{m}, \mathbf{S})$. We have shown earlier that the optimal $q(\mathbf{u})$ can also be obtained and depends on $q(\mathbf{x}_{0:T})$. This strong coupling suggests an alternating optimisation approach: optimise $q(\mathbf{u})$ while keeping $q(\mathbf{x}_{0:T})$ fixed, then optimise $q(\mathbf{x}_{0:T})$ while keeping $q(\mathbf{u})$ fixed, and repeat. This strategy was employed in Frigola et al. (2014), in which SMC sampling steps for $q(\mathbf{x}_{0:T})$ and optimal update steps for $q(\mathbf{u})$ based on eqs. (3.24) and (3.25) are interleaved. Instead, we follow McHutchon (2014) and explicitly parameterise a Gaussian variational approximation $q(\mathbf{x}_{0:T})$, and consider two ways to deal with $q(\mathbf{u})$:

- substitute the optimal $q(\mathbf{u})$ back to the variational free-energy and hence remove an explicit dependency on $q(\mathbf{u})$, the free-energy is then optimsed w.r.t the parameters of $q(\mathbf{x}_{0:T})$, and
- explicitly parameterise the mean and covariance of $q(\mathbf{u})$ in addition to that of $q(\mathbf{x}_{0:T})$ and optimise both distributions jointly using the variational free-energy.

We will refer to these two strategies as collapsed and uncollapsed, respectively.

3.3.3 Collapsed and uncollapsed variational free-energies

The uncollapsed variational free-energy is the original variational free-energy in eq. (3.14), as approximate variational distributions over the latent states and the pseudo-points are explicitly parameterised. For completeness, the free-energy can be written in closed-form as follows,

$$\mathcal{F}_{\text{vfe}}(.) = -\text{KL}(q(\mathbf{u})||p(\mathbf{u})) + \mathcal{H}(q(\mathbf{x}_{0:T})) + \langle \log p(\mathbf{x}_0) \rangle_{q(\mathbf{x}_0)} + \sum_{t=1}^{T} (\mathcal{F}_{\text{vfe, dyn, } t} + \mathcal{F}_{\text{vfe, emi, } t}),$$

where

$$\begin{aligned}
\mathcal{F}_{\text{vfe, dyn, } t} &= \int \mathrm{d}\mathbf{x}_{t-1,t}\mathrm{d}f q(\mathbf{x}_{t-1,t})q(f)\log p(\mathbf{x}_t|f,\mathbf{x}_{t-1}) \\
&= -\frac{1}{2}\log[(2\pi)^{D_\mathbf{x}}|\mathbf{Q}|] - \frac{1}{2}\operatorname{tr}\left(\mathbf{Q}^{-1}\left[\mathbf{m}_{\mathbf{x}_t}^\intercal\mathbf{m}_{\mathbf{x}_t} + \mathbf{S}_{\mathbf{x}_t}\right]\right) + \mathbf{m}_\mathbf{u}^\intercal\langle\mathbf{A}_t^\intercal\mathbf{Q}^{-1}\mathbf{x}_t\rangle_{q(\mathbf{x}_{t-1:t})} \\
&\quad - \frac{1}{2}\operatorname{tr}\left(\mathbf{Q}^{-1}\left[\langle\mathbf{B}_t\rangle_{q(\mathbf{x}_{t-1})} + \langle\mathbf{A}_t[\mathbf{S}_\mathbf{u} + \mathbf{m}_\mathbf{u}\mathbf{m}_\mathbf{u}^\intercal]\mathbf{A}_t^\intercal\rangle_{q(\mathbf{x}_{t-1})}\right]\right), \qquad (3.32) \\
\mathcal{F}_{\text{vfe, emi, } t} &= \int \mathrm{d}\mathbf{x}_t q(\mathbf{x}_t)\log p(\mathbf{y}_t|\mathbf{x}_t) \\
&= -\frac{1}{2}\log[(2\pi)^{D_\mathbf{y}}|\mathbf{R}_\mathbf{y}|] - \frac{1}{2}\mathbf{y}_t^\intercal\mathbf{R}_\mathbf{y}^{-1}\mathbf{y}_t + \mathbf{y}_t\mathbf{R}_\mathbf{y}^{-1}\mathbf{U}\mathbf{m}_{\mathbf{x}_t} \\
&\quad - \frac{1}{2}\operatorname{tr}\left(\mathbf{U}^\intercal\mathbf{R}_\mathbf{y}^{-1}\mathbf{U}[\mathbf{S}_{\mathbf{x}_t} + \mathbf{m}_{\mathbf{x}_t}\mathbf{m}_{\mathbf{x}_t}^\intercal]\right), \qquad (3.33) \\
\mathrm{KL}(q(\mathbf{u})||p(\mathbf{u})) &= \mathrm{KL}(\mathcal{N}(\mathbf{u};\mathbf{m}_\mathbf{u},\mathbf{S}_\mathbf{u})||\mathcal{N}(\mathbf{u};\mathbf{0},\mathbf{K}_{\mathbf{uu}})) \\
&= \frac{1}{2}\left[\operatorname{tr}(\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{S}_\mathbf{u}) + \mathbf{m}_\mathbf{u}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m}_\mathbf{u}^\intercal - M + \log|\mathbf{K}_{\mathbf{uu}}| - \log|\mathbf{S}_\mathbf{u}|\right], \qquad (3.34) \\
\langle\log p(\mathbf{x}_0)\rangle_{q(\mathbf{x}_0)} &= -\frac{1}{2}\log[(2\pi)^{D_\mathbf{x}}|\Sigma_0|] - \frac{1}{2}\operatorname{tr}(\Sigma_0[\mathbf{S}_{\mathbf{x}_0} + \mathbf{m}_{\mathbf{x}_0}\mathbf{m}_{\mathbf{x}_0}^\intercal]). \qquad (3.35)
\end{aligned}$$

While this requires an optimisation over more variational parameters, the objective is amenable to stochastic optimisation. In particular, the free-energy involves two sums over the measurement steps and as a result a noisy unbiased estimate of the energy can be obtained using a random subsequence of the entire training sequence as follows,

$$\mathcal{F}_{\text{vfe}}(.) \approx -\mathrm{KL}(q(\mathbf{u})||p(\mathbf{u})) + \mathcal{H}(q(\mathbf{x}_{0:T})) + \langle\log p(\mathbf{x}_0)\rangle_{q(\mathbf{x}_0)} + \frac{T}{B}\sum_{b=1}^{B}(\mathcal{F}_{\text{vfe, dyn, } b} + \mathcal{F}_{\text{vfe, emi, } b}),$$

where B is the length of the subsequence. This is computationally useful when the number of training steps is large, and an efficient computation of the entropy $\mathcal{H}(q(\mathbf{x}_{0:T}))$ is available.

When the variational free-energy is optimally maximised w.r.t. $q(\mathbf{u})$, we only need to maintain a variational approximation over the latent variables $q(\mathbf{x}_{0:T})$. In detail, substituting

the optimal $q(\mathbf{u})$ in eq. (3.16) into eq. (3.14) leads to,

$$\begin{aligned}
\mathcal{F}_{\text{vfe}}(.) &= -\int \mathrm{d}\mathbf{u} q(\mathbf{u}) \log \frac{p(\mathbf{u}) \exp(\sum_{t=1}^{T} \mathcal{G}_t)}{\mathcal{Z}_{\mathbf{u}} p(\mathbf{u})} + \mathcal{H}(q(\mathbf{x}_{0:T})) + \int \mathrm{d}\mathbf{x}_0 q(\mathbf{x}_0) \log p(\mathbf{x}_0) \\
&\quad + \sum_{t=1}^{T} \int \mathrm{d}\mathbf{x}_t q(\mathbf{x}_t) \log p(\mathbf{y}_t|\mathbf{x}_t) + \sum_{t=1}^{T} \int \mathrm{d}\mathbf{x}_{t-1,t} \mathrm{d}f q(\mathbf{x}_{t-1,t}) q(f) \log p(\mathbf{x}_t|f, \mathbf{x}_{t-1}) \\
&= -\int \mathrm{d}\mathbf{u} q(\mathbf{u}) \log \frac{p(\mathbf{u})}{p(\mathbf{u})} + \log \mathcal{Z}_{\mathbf{u}} - \sum_{t=1}^{T} \int \mathrm{d}\mathbf{u} q(\mathbf{u}) \mathcal{G}_t + \mathcal{H}(q(\mathbf{x}_{0:T})) + \int \mathrm{d}\mathbf{x}_0 q(\mathbf{x}_0) \log p(\mathbf{x}_0) \\
&\quad + \sum_{t=1}^{T} \int \mathrm{d}\mathbf{x}_t q(\mathbf{x}_t) \log p(\mathbf{y}_t|\mathbf{x}_t) + \sum_{t=1}^{T} \int \mathrm{d}\mathbf{x}_{t-1,t} \mathrm{d}f q(\mathbf{x}_{t-1,t}) q(f) \log p(\mathbf{x}_t|f, \mathbf{x}_{t-1}) \\
&= \log \mathcal{Z}_{\mathbf{u}} + \mathcal{H}(q(\mathbf{x}_{0:T})) + \int \mathrm{d}\mathbf{x}_0 q(\mathbf{x}_0) \log p(\mathbf{x}_0) + \sum_{t=1}^{T} \int \mathrm{d}\mathbf{x}_t q(\mathbf{x}_t) \log p(\mathbf{y}_t|\mathbf{x}_t). \quad (3.36)
\end{aligned}$$

Note that a closed-form $\log \mathcal{Z}_{\mathbf{u}}$ is available in eq. (3.26). The collapsed variational free-energy above is, however, not amenable to stochastic optimisation, as $\log \mathcal{Z}_{\mathbf{u}}$ has a non-trivial dependency on the latent variables at all measurement steps.

3.3.4 Diagonal and Markovian Gaussian parameterisations for $q(\mathbf{x}_{0:T})$

The simplest and perhaps most inaccurate approximation for $q(\mathbf{x}_{0:T})$ is a block-diagonal Gaussian, which assumes the latent variables across time steps are not correlated a posteriori, $q(\mathbf{x}_{0:T}) = \prod_t q(\mathbf{x}_t) = \prod_t \mathcal{N}(\mathbf{x}_t; \mu_t, \Sigma_t)$. Let $D_{\mathbf{x}}$ be the dimensions of the latent states, the number of parameters needed for this block-diagonal approximation is $(T+1)D_{\mathbf{x}}$ and $TD_{\mathbf{x}}(D_{\mathbf{x}}+1)/2$ for the mean and covariance respectively.[3] This approximation is less damaging when the observations are more informative about the latent variables, or when the ground truth posterior over the latent variables is not strongly correlated. However, when the hyperparameters are concurrently optimised, this property means the variational objective will potentially bias the hyperparameters towards a region in which an uncorrelated approximate posterior is a good approximation, for example by learning a simpler latent function or higher noise (Turner and Sahani, 2011).

An alternative to improve over this block-diagonal approximation is to use a fully-correlated Gaussian approximation, $q(\mathbf{x}_{0:T}) = \mathcal{N}(\mathbf{x}_{0:T}; \mu, \Sigma)$. However, this is computationally expensive for long sequences due to the computation of the entropy of $q(\mathbf{x}_{0:T})$, a $(T+1)D_{\mathbf{x}}$-dimensional Gaussian density. Additionally, this requires a number of parameters of order $\mathcal{O}(T^2)$, specfically $(T+1)D_{\mathbf{x}}$ and $(T+1)D_{\mathbf{x}}((T+1)D_{\mathbf{x}}+1)/2$ for the mean and covariance respectively.

[3] The number of parameters for the covariance is further reduced to $(T+1)D_{\mathbf{x}}$ when the approximation is fully diagonal.

Observing that the optimal $q(\mathbf{x}_{0:T})$ possesses a Markovian structure, we attempt to incorporate this structure into the Gaussian approximation. The resulting structured Gauss-Markov approximation offers a nice trade-off between approximation quality and computational complexity: i. it explicitly retains correlations across measurement steps, and ii. it allows quick computation of the marginal and pair-wise densities as well as the entropy term. In detail, the Gauss-Markov property allows the following decomposision of the joint approximation,

$$q(\mathbf{x}_{0:T}) = q(\mathbf{x}_0) \prod_{t=1}^{T} \bar{q}_t(\mathbf{x}_{t-1}, \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_0; \mu_0, \Sigma_0) \prod_{t=1}^{T} \mathcal{N}\Bigg(\begin{bmatrix} \mathbf{x}_{t-1} \\ \mathbf{x}_t \end{bmatrix}; \begin{bmatrix} \mu_{t-1}^t \\ \mu_t^t \end{bmatrix}, \begin{bmatrix} \Sigma_{t-1}^t & \Sigma_{t-1,t}^t \\ \Sigma_{t,t-1}^t & \Sigma_t^t \end{bmatrix} \Bigg),$$

which enforces conditional independence between non-adjacent hidden variables given all other hidden variables in between. Due to this property, the precision matrix of this variational approximation is tri-block-diagonal. Note that $\bar{q}_t(\mathbf{x}_{t-1}, \mathbf{x}_t)$ denotes a factor that involves two variables, $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$, and is not the pairwise marginal of the variational distribution, $q(\mathbf{x}_{t-1}, \mathbf{x}_t)$. The latter quantity, $q(\mathbf{x}_{t-1}, \mathbf{x}_t)$, can be computed as follows,

$$q(\mathbf{x}_{t-1}, \mathbf{x}_t) = \bar{q}_{t-1}(\mathbf{x}_{t-1}) \bar{q}_t(\mathbf{x}_{t-1}, \mathbf{x}_t) \bar{q}_{t+1}(\mathbf{x}_t). \tag{3.37}$$

Similarly, the marginal distribution, $q(\mathbf{x}_t)$, can also be efficiently computed,

$$q(\mathbf{x}_t) = \bar{q}_{t-1}(\mathbf{x}_t) \bar{q}_t(\mathbf{x}_t). \tag{3.38}$$

These formulations mean that the computation of all pairwise and single-site marginals over successive time steps can be done in one sweep over the training sequence, i.e. linear in T. In fact, this parameterisation allows the marginal computation based on local parameters, which means computing the pairwise and single-site marginals on any subsequence can be done in time linear in the subsequence length, and does not require a loop over the entire training sequence. This result is important when subsequence based stochastic learning is required. This parameteristaion of the Gauss-Markov structure has been used by McHutchon (2014), and is strictly more general than the autoregressive structure employed by Eleftheriadis et al. (2017).[4]

[4]In addition, the autoregressive structure in Eleftheriadis et al. (2017), $q(\mathbf{x}_{0:T}) = q(\mathbf{x}_0) \prod_{t=1}^{T} q(\mathbf{x}_t | \mathbf{x}_{t-1})$ does require a loop over the entire training sequence to compute the marginal $q(\mathbf{x}_T)$, and therefore is arguably unsuited for subsequence based stochastic training.

The entropy term $\mathcal{H}(q(\mathbf{x}_{0:T}))$ can also be found in a closed-form that involves only the aforementioned marginal and pair-wise densities,

$$\mathcal{H}(q(\mathbf{x}_{0:T})) = -\int \mathrm{d}\mathbf{x}_{0:T} q(\mathbf{x}_{0:T}) \log q(\mathbf{x}_{0:T}) \tag{3.39}$$

$$= -\int \mathrm{d}\mathbf{x}_{0:T} q(\mathbf{x}_{0:T}) \log \left[q(\mathbf{x}_0) \prod_{t=1}^{T} \frac{q(\mathbf{x}_t, \mathbf{x}_{t-1})}{q(\mathbf{x}_{t-1})} \right] \tag{3.40}$$

$$= \frac{(T+1)D_x}{2} \log(2\pi e) + \frac{1}{2} \sum_{t=1}^{T} \log |\Sigma_{t-1:t}| - \frac{1}{2} \sum_{t=1}^{T-1} \log |\Sigma_t|. \tag{3.41}$$

The number of parameters used for this parameterisation is only linear in the sequence length T, specifically $(2T+1)D_{\mathbf{x}}$ for the mean parameters, $D_{\mathbf{x}}(D_{\mathbf{x}}+1)/2$ for the covariance of $q(\mathbf{x}_0)$, and $TD_{\mathbf{x}}(2D_{\mathbf{x}}+1)$ for the covariance of $\{\bar{q}(\mathbf{x}_{t-1}, \mathbf{x}_t)\}_{t=1}^{T}$. This requirement is significantly smaller than that of the fully correlated parameterisation above, and is about four times as many parameters as the mean-field parameterisation.

Next, we discuss a more general deterministic approximate inference framework based on Power EP, and additional approximations required for tractable inference and learning in GP state space models.

3.4 The Power EP approach

It has been shown in the previous chapter that the variational free-energy approach for regression and classification is a special case of a general inference and learning framework based on Power EP. In a similar fashion, we present a unifying framework for learning and inference in GPSSMs based on Power EP. Similar to the previous chapter, we will view Power EP as approximating the joint density, which provides both the approximate posterior and the approximate marginal likelihood. In particular, the joint density of the latent variables, latent dynamics and the observed measurements is

$$p(\mathbf{y}_{1:T}, \mathbf{x}_{0:T}, f|\theta) = p(\mathbf{x}_0)p(f|\theta) \prod_{t=1}^{T} p(\mathbf{x}_t|f, \mathbf{x}_{t-1}, \theta)p(\mathbf{y}_t|\mathbf{x}_t, \theta), \tag{3.42}$$

and the approximate joint density takes the following form,

$$q(\mathbf{x}_{0:T}, f) = p(\mathbf{x}_0)p(f|\theta) \prod_{t=1}^{T} [\phi_t(\mathbf{x}_{t-1}, \mathbf{x}_t)h_t(\mathbf{u})][\gamma_t(\mathbf{x}_t)], \tag{3.43}$$

where the product $\phi_t(\mathbf{x}_{t-1}, \mathbf{x}_t)h_t(\mathbf{u})$ is introduced to approximate the transition factor $p(\mathbf{x}_t|f, \mathbf{x}_{t-1}, \theta)$, and similarly $\gamma_t(\mathbf{x}_t)$ for the emission $p(\mathbf{y}_t|\mathbf{x}_t, \theta)$. This approximation can be illustrated using the factor graphs in fig. 3.1(B), which schematically shows how Power EP operates at the local factor level by breaking up the difficult factors into simpler factors that

we can approximate. Note that the h factors only depend on a small number of function values $\mathbf{u}$.

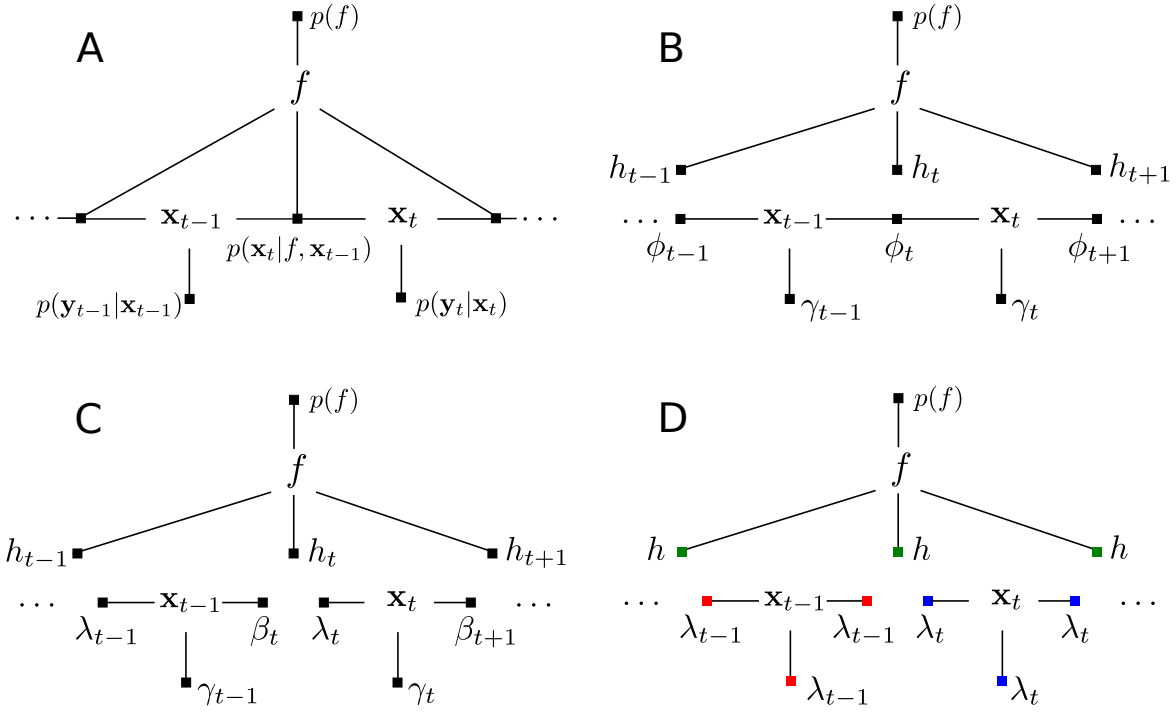


Fig. 3.1 A: a factor graph of the Gaussian process state space model. Note that while there are many factor graphs that can be used to show the same model, our approximation is based on this representation. B: a factor graph showing approximate factors and the variables that each factor involves. This factor graph assumes a correlated structure over the hidden variables. C: another approximation to the original factor graph, but this assumes a mean-field structure over the hidden variables. D: a factor graph showing how factors in C are tied, factors coloured using the same colour are identical. Best viewed in colour.

The approximate posterior above can be rewritten by observing that factors that touch a common variable can be grouped,

$$q(\mathbf{x}_{0:T}, f) = p(f|\theta)\left[\prod_{t=1}^{T} h_t(\mathbf{u})\right]\left[p(\mathbf{x}_0)\phi_0(\mathbf{x}_0)\prod_{t=1}^{T}\phi_t(\mathbf{x}_{t-1},\mathbf{x}_t)\gamma_t(\mathbf{x}_t)\right] \tag{3.44}$$

$$= p(f_{\neq\mathbf{u}}|\mathbf{u},\theta)\left[p(\mathbf{u})\prod_{t=1}^{T} h_t(\mathbf{u})\right]\left[p(\mathbf{x}_0)\phi_0(\mathbf{x}_0)\prod_{t=1}^{T}\phi_t(\mathbf{x}_{t-1},\mathbf{x}_t)\gamma_t(\mathbf{x}_t)\right]. \tag{3.45}$$

The resulting form above resembles the global approximate posterior used for the VFE approach in eq. (3.10), except that here we have explicitly assume a Markovian structure for the latent variables. We can further impose a mean-field structure for the hidden variables, by assuming that $\phi_t(\mathbf{x}_{t-1},\mathbf{x}_t) = \lambda_t(\mathbf{x}_t)\beta_t(\mathbf{x}_{t-1})$, as illustrated in fig. 3.1(C). The resulting global approximation posterior is identical to that used in the mean-field (diagonal Gaussian)

VFE approach discussed in the last section. As the correlated structure in eq. (3.45) is more general, we will use this form in the next sections.[5]

Having described the form of the approximate joint density, we will next show how the approximate local factors are found using the Power EP procedure. Similar to the regression and classification cases in chapter 2, the Power EP procedure iterative refines the approximate factors by finding the cavity distribution, matching the approximate posterior's moments to that of the tilted distribution and updating the factors, until it satisfies some convergence condition. However, unlike the regression and classification cases, the scheduling of the updates in the state-space model could affect the convergence. We have only experimented with updating factors in a chronological order, i.e. sequentially from time step 0 to time step T. This schedule could slow down convergence when the time series is long. Alternative update schedules, such as parallel updates or interleaved forward and backward updates, are left as future work. In the experiments included in section 3.8, the scheduling is not an issue as we do not run the Power-EP iterative procedure and instead, the approximate Power-EP energy is directly optimised. Note that in what follows, the approximate factors are assumed to be Gaussian, which allows analytic computation; however, this assumption could be relaxed.

3.4.1 Dealing with the transition factor $p(\mathbf{x}_t|f, \mathbf{x}_{t-1})$

The goal is to update the factors $\phi_t(\mathbf{x}_{t-1}, \mathbf{x}_t)$ and $h_t(\mathbf{u})$ such that they approximate the contribution of $p(\mathbf{x}_{t-1}|f, \mathbf{x}_t)$ towards the true posterior. We first compute the cavity distributions by removing a fraction of the approximate factors, $\phi_t(\mathbf{x}_{t-1}, \mathbf{x}_t)$ and $h_t(\mathbf{u})$, from the posterior as follows,

$$q^{\backslash t}(\mathbf{x}_{t-1}, \mathbf{x}_t) = \phi_{t-1}(\mathbf{x}_{t-1})\phi_t^{1-\alpha}(\mathbf{x}_{t-1}, \mathbf{x}_t)\phi_{t+1}(\mathbf{x}_t), \tag{3.46}$$

$$q^{\backslash t}(f) = p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)q^{\backslash t}(\mathbf{u}) = p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)p(\mathbf{u})h_t(\mathbf{u})^{1-\alpha}\prod_{i\neq t} h_i(\mathbf{u}), \tag{3.47}$$

where $\phi_{t-1}(\mathbf{x}_{t-1})$ and $\phi_{t+1}(\mathbf{x}_t)$ are the marginals of $\phi_{t-1}(\mathbf{x}_{t-2}, \mathbf{x}_{t-1})$ and $\phi_{t+1}(\mathbf{x}_t, \mathbf{x}_{t+1})$, respectively. The surrogate posterior or the tilted distribution is formed by multiplying a fraction of the transition factor with the cavity distribution:

$$\tilde{q}(f, \mathbf{x}_{t-1}, \mathbf{x}_t) = q^{\backslash t}(\mathbf{x}_{t-1}, \mathbf{x}_t)q^{\backslash t}(f)p^{\alpha}(\mathbf{x}_t|f, \mathbf{x}_{t-1}). \tag{3.48}$$

The central step of Power EP is the projection step, that is to find a new approximate posterior by minimising the unnormalised KL divergence, $\overline{\mathrm{KL}}(\tilde{q}(f, \mathbf{x}_{t-1}, \mathbf{x}_t)||q(f, \mathbf{x}_{t-1}, \mathbf{x}_t))$. This minimisation problem is equivalent to finding $q(f, \mathbf{x}_{t-1}, \mathbf{x}_t)$ whose zeroth, first and second order moments match that of the tilted distribution $\tilde{q}(f, \mathbf{x}_{t-1}, \mathbf{x}_t)$. Similar to the

[5] The mean-field results can be easily obtained from the results for the structured case by setting the off-diagonal components to zero.

regression and classification cases, this moment matching step is achieved by computing the log-normaliser of the tilted distribution, $\log \tilde{Z} = \log \mathrm{E}_{\tilde{q}(f,\mathbf{x}_{t-1},\mathbf{x}_t)}[1]$ and its gradients w.r.t. the cavity mean and covariance, and performing the following updates,

$$\mathbf{m}_\mathbf{u} = \mathbf{m}_\mathbf{u}^{\backslash t} + \mathbf{V}_\mathbf{u}^{\backslash t} \frac{\mathrm{d} \log \tilde{Z}}{\mathrm{d}\mathbf{m}_\mathbf{u}^{\backslash t}}; \qquad \mathbf{V}_\mathbf{u} = \mathbf{V}_\mathbf{u}^{\backslash t} - \mathbf{V}_\mathbf{u}^{\backslash t} \left[\frac{\mathrm{d} \log \tilde{Z}}{\mathbf{m}_\mathbf{u}^{\backslash t}} \left(\frac{\mathrm{d} \log \tilde{Z}}{\mathbf{m}_\mathbf{u}^{\backslash t}} \right)^\intercal - 2 \frac{\mathrm{d} \log \tilde{Z}}{\mathbf{V}_\mathbf{u}^{\backslash t}} \right] \mathbf{V}_\mathbf{u}^{\backslash t}.$$

The updates for $\{\mathbf{x}_{t-1}, \mathbf{x}_t\}$ take identical forms. In general, the quantity $\log \tilde{Z}$ and its gradients are not available in closed-form, as the normaliser of the tilted distribution is an intergrated product of a Gaussian with a non-Gaussian distribution as illustrated in fig. 3.2. In more detail,

$$\begin{aligned} \tilde{Z} &= \int \mathrm{d}\mathbf{x}_{t-1} \mathrm{d}\mathbf{x}_t \mathrm{d}f q^{\backslash t}(\mathbf{x}_{t-1}, \mathbf{x}_t) q^{\backslash t}(f) p^\alpha(\mathbf{x}_t | f, \mathbf{x}_{t-1}) \\ &= \int \mathrm{d}\mathbf{x}_{t-1} \mathrm{d}\mathbf{x}_t \mathrm{d}f \mathcal{N}(\mathbf{x}_{t-1:t}; \mathbf{m}_{\mathbf{x}_{t-1:t}}^{\backslash t}, \mathbf{V}_{\mathbf{x}_{t-1:t}}^{\backslash t}) p(f_{\neq \mathbf{u}} | \mathbf{u}) q^{\backslash t}(\mathbf{u}) \mathcal{N}^\alpha(\mathbf{x}_t; f(\mathbf{x}_{t-1}), \mathbf{Q}). \end{aligned} \tag{3.49}$$

The strategy we will follow next is to exactly integrate out f and $\mathbf{x}_t$, and then approximately integrate out $\mathbf{x}_{t-1}$. In detail, we can first integrate out all latent function values, except $f(\mathbf{x}_{t-1})$, as follows,

$$\begin{aligned} q_a(f(\mathbf{x}_{t-1})) &= \int \mathrm{d}f_{\neq f(\mathbf{x}_{t-1})} p(f_{\neq \mathbf{u}} | \mathbf{u}) q^{\backslash t}(\mathbf{u}) && (3.50) \\ &= \int \mathrm{d}\mathbf{u} \mathcal{N}(f(\mathbf{x}_{t-1}); \mathbf{A}_t \mathbf{u}, \mathbf{B}_t) q^{\backslash t}(\mathbf{u}) && (3.51) \\ &= \mathcal{N}(f(\mathbf{x}_{t-1}); \underbrace{\mathbf{A}_t \mathbf{m}_\mathbf{u}^{\backslash t}}_{\mathbf{m}_a}, \underbrace{\mathbf{B}_t + \mathbf{A}_t \mathbf{S}_\mathbf{u}^{\backslash t} \mathbf{A}_t^\intercal}_{\mathbf{v}_a}), && (3.52) \end{aligned}$$

where $\mathbf{A}_t = \mathbf{K}_{\mathrm{f}_t\mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1}$, $\mathbf{B}_t = \mathbf{K}_{\mathrm{f}_t\mathrm{f}_t} - \mathbf{K}_{\mathrm{f}_t\mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{K}_{\mathbf{u}\mathrm{f}_t}$. Note that this distribution is the prediction of function value $f(\mathbf{x}_{t-1})$ at an input $\mathbf{x}_{t-1}$ when the cavity distribution $q^{\backslash t}(\mathbf{u})$ is used as the posterior. As such, it depends on $\mathbf{x}_{t-1}$ in a complex manner, through the covariance matrices $\mathbf{K}_{\mathrm{f}_t\mathbf{u}}$ and $\mathbf{K}_{\mathrm{f}_t\mathrm{f}_t}$.

Notice that another quantity in eq. (3.49) that touches $f(\mathbf{x}_{t-1})$ is $\mathcal{N}^\alpha(\mathbf{x}_t; f(\mathbf{x}_{t-1}), \mathbf{Q})$, which is also a Gaussian in $f(\mathbf{x}_{t-1})$. Consequently, $f(\mathbf{x}_{t-1})$ can also be integrated out analytically as follows,

$$\begin{aligned} q_b(\mathbf{x}_t | \mathbf{x}_{t-1}) &= \int \mathrm{d}f(\mathbf{x}_{t-1}) q_a(f(\mathbf{x}_{t-1})) \mathcal{N}^\alpha(\mathbf{x}_t; f(\mathbf{x}_{t-1}), \mathbf{Q}) && (3.53) \\ &= \int \mathrm{d}f(\mathbf{x}_{t-1}) \mathcal{N}(f(\mathbf{x}_{t-1}); \mathbf{m}_a, \mathbf{v}_a) \mathcal{N}^\alpha(\mathbf{x}_t; f(\mathbf{x}_{t-1}), \mathbf{Q}) && (3.54) \\ &= \frac{(2\pi)^{D_\mathbf{x}/2} |\alpha^{-1}\mathbf{Q}|^{1/2}}{(2\pi)^{\alpha D_\mathbf{x}/2} |\mathbf{Q}|^{\alpha/2}} \mathcal{N}(\mathbf{x}_t; \mathbf{m}_a, \mathbf{v}_a + \alpha^{-1}\mathbf{Q}) && (3.55) \end{aligned}$$

Notice further that the cavity distribution $q^{\setminus t}(\mathbf{x}_{t-1}, \mathbf{x}_t)$ can be decomposed,

$$q^{\setminus t}(\mathbf{x}_{t-1}, \mathbf{x}_t) = q^{\setminus t}(\mathbf{x}_t|\mathbf{x}_{t-1})q^{\setminus t}(\mathbf{x}_{t-1}),$$

where

$$q^{\setminus t}(\mathbf{x}_{t-1}, \mathbf{x}_t) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_{t-1} \\ \mathbf{x}_t \end{bmatrix}; \begin{bmatrix} \mathbf{m}^{\setminus t}_{\mathbf{x}_{t-1}} \\ \mathbf{m}^{\setminus t}_{\mathbf{x}_t} \end{bmatrix}, \begin{bmatrix} \mathbf{V}^{\setminus t}_{\mathbf{x}_{t-1}} & \mathbf{V}^{\setminus t}_{\mathbf{x}_{t-1},\mathbf{x}_t} \\ \mathbf{V}^{\setminus t}_{\mathbf{x}_t,\mathbf{x}_{t-1}} & \mathbf{V}^{\setminus t}_{\mathbf{x}_t} \end{bmatrix} \right), \tag{3.56}$$

$$q^{\setminus t}(\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_{t-1}; \mathbf{m}^{\setminus t}_{\mathbf{x}_{t-1}}, \mathbf{V}^{\setminus t}_{\mathbf{x}_{t-1}}), \tag{3.57}$$

$$q^{\setminus t}(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \mathbf{c} + \mathbf{W}\mathbf{x}_{t-1}; \mathbf{H}) \tag{3.58}$$

$$\mathbf{W} = \mathbf{V}^{\setminus t}_{\mathbf{x}_t,\mathbf{x}_{t-1}} \mathbf{V}^{\setminus t,-1}_{\mathbf{x}_{t-1}} \tag{3.59}$$

$$\mathbf{c} = \mathbf{m}^{\setminus t}_{\mathbf{x}_t} - \mathbf{V}^{\setminus t}_{\mathbf{x}_t,\mathbf{x}_{t-1}} \mathbf{V}^{\setminus t,-1}_{\mathbf{x}_{t-1}} \mathbf{m}^{\setminus t}_{\mathbf{x}_{t-1}} \tag{3.60}$$

$$\mathbf{H} = \mathbf{V}^{\setminus t}_{\mathbf{x}_t} - \mathbf{V}^{\setminus t}_{\mathbf{x}_t,\mathbf{x}_{t-1}} \mathbf{V}^{\setminus t,-1}_{\mathbf{x}_{t-1}} \mathbf{V}^{\setminus t}_{\mathbf{x}_{t-1},\mathbf{x}_t} \tag{3.61}$$

This allows us to multiply $q^{\setminus t}(\mathbf{x}_t|\mathbf{x}_{t-1})$ with $q_b(\mathbf{x}_t|\mathbf{x}_{t-1})$, and integrate out $\mathbf{x}_t$ analytically,

$$\Gamma(\mathbf{c}|\mathbf{x}_{t-1}) = \int \mathrm{d}\mathbf{x}_t q^{\setminus t}(\mathbf{x}_t|\mathbf{x}_{t-1}) q_b(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{3.62}$$

$$= \frac{(2\pi)^{D_\mathbf{x}/2}|\alpha^{-1}\mathbf{Q}|^{1/2}}{(2\pi)^{\alpha D_\mathbf{x}/2}|\mathbf{Q}|^{\alpha/2}} \int \mathrm{d}\mathbf{x}_{t-1} \mathcal{N}(\mathbf{x}_t; \mathbf{c} + \mathbf{W}\mathbf{x}_{t-1}; \mathbf{H}) \mathcal{N}(\mathbf{x}_t; \mathbf{m}_a, \mathbf{v}_a + \alpha^{-1}\mathbf{Q}) \tag{3.63}$$

$$= \frac{(2\pi)^{D_\mathbf{x}/2}|\alpha^{-1}\mathbf{Q}|^{1/2}}{(2\pi)^{\alpha D_\mathbf{x}/2}|\mathbf{Q}|^{\alpha/2}} \mathcal{N}(\mathbf{c}; \mathbf{m}_a - \mathbf{W}\mathbf{x}_{t-1}, \mathbf{v}_a + \mathbf{H} + \alpha^{-1}\mathbf{Q}) \tag{3.64}$$

Finally, substituting $\Gamma(\mathbf{c}|\mathbf{x}_{t-1})$ into $\tilde{Z}$ in eq. (3.49) and noticing that $\mathbf{x}_{t-1}$ is the only remaining variable that needs to be integrated out, lead to,

$$\tilde{Z} = \frac{(2\pi)^{D_\mathbf{x}/2}|\alpha^{-1}\mathbf{Q}|^{1/2}}{(2\pi)^{\alpha D_\mathbf{x}/2}|\mathbf{Q}|^{\alpha/2}} \int \mathrm{d}\mathbf{x}_{t-1} \mathcal{N}(\mathbf{x}_{t-1}; \mathbf{m}^{\setminus t}_{\mathbf{x}_{t-1}}, \mathbf{V}^{\setminus t}_{\mathbf{x}_{t-1}}) \mathcal{N}(\mathbf{c}; \mathbf{m}_a - \mathbf{W}\mathbf{x}_{t-1}, \mathbf{v}_a + \mathbf{H} + \alpha^{-1}\mathbf{Q})$$

which is analytically intractable for general α, due to the non-linear dependencies of $\mathbf{m}_a$ and $\mathbf{v}_a$ in $\mathbf{x}_{t-1}$. Specifically, for each deterministic $\mathbf{x}_{t-1}$, the output $\mathbf{c}$ is Gaussian-distributed. Unfortunately, as $\mathbf{x}_{t-1}$ is a random variable that we have posited a variational distribution over, the output value when $\mathbf{x}_{t-1}$ is integrated out can be thought of as an infinite mixture of Gaussians and is not available in closed-form.[6] We will next detail three techniques that allow $\tilde{Z}$ to be approximately computed, based on simple Monte Carlo, Gaussian projection and linearisation. In control and signal processing literature, these techniques are often called Monte-Carlo unscented propagation, moment matching unscented propagation and scented

[6] When $\alpha \to 0$, the quantity $\alpha^{-1}\log \tilde{Z}$ is analytically tractable, as we will show in section 3.4.4.

propagation respectively. The technique we develop here generalise existing methods, and can be extended to other state-space models.

Simple Monte Carlo

For a single deterministic $\mathbf{x}_{t-1}$, $\Gamma(\mathbf{c}|\mathbf{x}_{t-1})$ is a Gaussian distribution. This suggests a Monte Carlo approach for evaluating $\tilde{Z}$ as follows,

$$\tilde{Z} \approx \frac{(2\pi)^{D_{\mathbf{x}}/2}|\alpha^{-1}\mathbf{Q}|^{1/2}}{(2\pi)^{\alpha D_{\mathbf{x}}/2}|\mathbf{Q}|^{\alpha/2}} \frac{1}{L} \sum_{l=1}^{L} \mathcal{N}(\mathbf{c}; \mathbf{A}_{t,l}\mathbf{m}_{\mathbf{u}}^{\backslash t} - \mathbf{W}\mathbf{x}_{t-1,l}, \mathbf{B}_{t,l} + \mathbf{A}_{t,l}\mathbf{S}_{\mathbf{u}}^{\backslash t}\mathbf{A}_{t,l}^{\intercal} + \mathbf{H} + \alpha^{-1}\mathbf{Q}), \tag{3.65}$$

where L samples, $\{\mathbf{x}_{t-1,l}\}_{l=1}^{L}$, are randomly drawn from $q^{\backslash t}(\mathbf{x}_{t-1})$. For low-dimensional state spaces, the location of the samples can be chosen according to many Gaussian quadrature rules, e.g. Gauss-Hermite, to reduce the variance of the estimate. The gradients w.r.t. the cavity mean and covariance are also available by differentiating through the above Monte Carlo estimate, and using the *reparameterisation trick* (Kingma and Welling, 2014; Salimans and Knowles, 2013; Rezende et al., 2014). Critically, though the estimation of $\tilde{Z}$ obtained from using this Monte-Carlo procedure is unbiased (indeed it is asymptotically unbiased), the computation of $\log \tilde{Z}$ using this estimate will become biased. This bias can be reduced when the number of sample points increases, and is often negliably smaller than the variance introduced by additional Monte Carlo approximations (such as minibatch-based Monte Carlo over training instances (Hernández-Lobato et al., 2016)).

Moment matching or Gaussian projection

It has been noted above that the integral, $\int \mathrm{d}\mathbf{x}_{t-1} q^{\backslash t}(\mathbf{x}_{t-1})\Gamma(\mathbf{c}|\mathbf{x}_{t-1})$ is intractable as the non-linear dependencies of the mean and covariance of $\Gamma(\mathbf{c}|\mathbf{x}_{t-1})$ on $\mathbf{x}_{t-1}$, i.e. the resulting distribution over $\mathbf{c}$ when $\mathbf{x}$ is integrated out is non-Gaussian. However, its mean and covariance can be computed in closed-form for widely used covariance functions such as exponentiated quadratic, linear or a more general class of spectral mixture kernels (Wilson and Adams, 2013). Following Girard et al. (2003); Deisenroth and Mohamed (2012), we can use the law of iterated conditionals to exactly find the mean and covariance of the distribution over the output values in eq. (3.64) as follows,

$$\overline{\mathbf{m}}_{\mathbf{c}} = \mathrm{E}_{q^{\backslash t}(\mathbf{x}_{t-1})}[m_{\Gamma(\mathbf{c}|\mathbf{x}_{t-1})}], \tag{3.66}$$

$$\overline{\mathbf{S}}_{\mathbf{c}} = \mathrm{E}_{q^{\backslash t}(\mathbf{x}_{t-1})}[v_{\Gamma(\mathbf{c}|\mathbf{x}_{t-1})}] + \mathrm{var}_{q^{\backslash t}(\mathbf{x}_{t-1})}[m_{\Gamma(\mathbf{c}|\mathbf{x}_{t-1})}] \tag{3.67}$$

which in words are the expected mean, and the sum of the expected variance and the variance of the mean, respectively, where the expectations are taken w.r.t. $q^{\backslash t}(\mathbf{x}_{t-1})$. Substituting the

mean and covariance of $\Gamma(c|\mathbf{x}_{t-1})$ in eq. (3.64) in the above results gives:

$$\overline{\mathbf{m}}_{\mathbf{c}} = \langle \mathbf{A}_t \rangle_{q^{\setminus t}(\mathbf{x}_{t-1})} \mathbf{m}_{\mathbf{u}}^{\setminus t} - \mathbf{W}\mathbf{m}_{\mathbf{x}_{t-1}}^{\setminus t}, \tag{3.68}$$

$$\begin{aligned}\overline{\mathbf{S}}_{\mathbf{c}} = &\langle \mathbf{B}_t \rangle_{q^{\setminus t}(\mathbf{x}_{t-1})} + \langle \mathbf{A}_t [\mathbf{S}_{\mathbf{u}}^{\setminus t} + \mathbf{m}_{\mathbf{u}}^{\setminus t} \mathbf{m}_{\mathbf{u}}^{\setminus t,\intercal}] \mathbf{A}_t^\intercal \rangle_{q^{\setminus t}(\mathbf{x}_{t-1})} + 2\mathbf{W} \langle \mathbf{x}_{t-1} \mathbf{m}_{\mathbf{u}}^{\setminus t} \mathbf{A}_t^\intercal \rangle_{q^{\setminus t}(\mathbf{x}_{t-1})} \\ &+ \mathbf{W}[\mathbf{V}_{\mathbf{x}_{t-1}}^{\setminus t} + \mathbf{m}_{\mathbf{x}_{t-1}}^{\setminus t} \mathbf{m}_{\mathbf{x}_{t-1}}^{\setminus t,\intercal}] \mathbf{W}^\intercal + \mathbf{H} + \alpha^{-1}\mathbf{Q} - \overline{\mathbf{m}}_{\mathbf{c}} \overline{\mathbf{m}}_{\mathbf{c}}^\intercal.\end{aligned} \tag{3.69}$$

Computing the above mean and covariance requires the expectations of the covariance matrices, as identically needed in the VFE approach. Critically, a Gaussian distribution of the same mean and covariance can be used as an approximation to the non-Gaussian and intractable distribution over $\mathbf{c}$ when $\mathbf{x}_{t-1}$ is integrated out, allowing analytic computation of an approximation to $\tilde{Z}$ as follows,

$$\tilde{Z} \approx \frac{(2\pi)^{D_{\mathbf{x}}/2} |\alpha^{-1}\mathbf{Q}|^{1/2}}{(2\pi)^{\alpha D_{\mathbf{x}}/2} |\mathbf{Q}|^{\alpha/2}} \mathcal{N}(\mathbf{c}; \overline{\mathbf{m}}_{\mathbf{c}}, \overline{\mathbf{S}}_{\mathbf{c}}). \tag{3.70}$$

This Gaussian projection procedure is illustrated in fig. 3.2, which shows that the approximation is accurate when the cavity distributions have tight (co)variances, and inaccurate, but desirably so, when these distributions are uncertain and the output values are multi-modal and heavy-tailed.

Linearisation

The core reason why the above approximations are needed is the propagation of the input distribution through a distribution or process over the latent *non-linear* function, whose covariance function depends on the input in a non-linear manner. This is no longer an issue when the underlying function is linear *and* the covariance function is a constant w.r.t. the input. Using this insight leads us to a local linearisation of the mapping from $\mathbf{x}_{t-1}$ to $\mathbf{c}$, around the mean $\mathbf{m}_{\mathbf{x}_{t-1}}^{\setminus t}$ of the cavity distribution $q^{\setminus t}(\mathbf{x}_{t-1})$ as follows,

$$\begin{aligned}&\Gamma(\mathbf{c}|\mathbf{x}_{t-1}) \propto \mathcal{N}(\mathbf{c}; \mathbf{m}_a - \mathbf{W}\mathbf{x}_{t-1}, \mathbf{v}_a + \mathbf{H} + \alpha^{-1}\mathbf{Q}) \approx \tilde{\Gamma}(\mathbf{c}|\mathbf{x}_{t-1}) \propto \mathcal{N}(\mathbf{c}; \tilde{\mathbf{m}}_\Gamma, \tilde{\mathbf{S}}_\Gamma), \\ &\text{where } \tilde{\Gamma}_{\mathbf{c}} = \left[\mathbf{A}_{t, \mathbf{m}_{\mathbf{x}_{t-1}}^{\setminus t}} + \sum_{d=1}^{D_{\mathbf{x}}} (x_{t-1,d} - m_{t-1,d}) \frac{\partial \mathbf{A}_t}{\partial x_{t-1,d}} \Big|_{\mathbf{m}_{\mathbf{x}_{t-1}}^{\setminus t}} \right] \mathbf{m}_{\mathbf{u}}^{\setminus t} + \mathbf{W}\mathbf{x}_{t-1}, \\ &\tilde{\mathbf{S}}_\Gamma = \mathbf{B}_{t, \mathbf{m}_{\mathbf{x}_{t-1}}^{\setminus t}} + \mathbf{A}_{t, \mathbf{m}_{\mathbf{x}_{t-1}}^{\setminus t}} \mathbf{S}_{\mathbf{u}}^{\setminus t} \mathbf{A}_{t, \mathbf{m}_{\mathbf{x}_{t-1}}^{\setminus t}}^\intercal + \mathbf{H} + \alpha^{-1}\mathbf{Q},\end{aligned}$$

and $\mathbf{A}_{t, \mathbf{m}_{\mathbf{x}_{t-1}}^{\setminus t}}$ and $\mathbf{B}_{t, \mathbf{m}_{\mathbf{x}_{t-1}}^{\setminus t}}$ are $\mathbf{A}_t$ and $\mathbf{B}_t$ evaluated at $\mathbf{m}_{\mathbf{x}_{t-1}}^{\setminus t}$, and $\{\frac{\partial \mathbf{A}_t}{\partial x_{t-1,d}}\}_{d=1}^{D_{\mathbf{x}}}$ are the gradients of $\mathbf{A}_t$ w.r.t. $\mathbf{x}_{t-1}$. The linearisation step above assumes the mapping from $\mathbf{x}_{t-1}$ to $\mathbf{c}$ is linear around the mean of $q^{\setminus t}(\mathbf{x}_{t-1})$, and it is accurate when this is the case. In addition, this approximation step does not rely on the covariance of $q^{\setminus t}(\mathbf{x}_{t-1})$, that is different input distributions with the same mean will lead to the same linearisation. However, the covariance

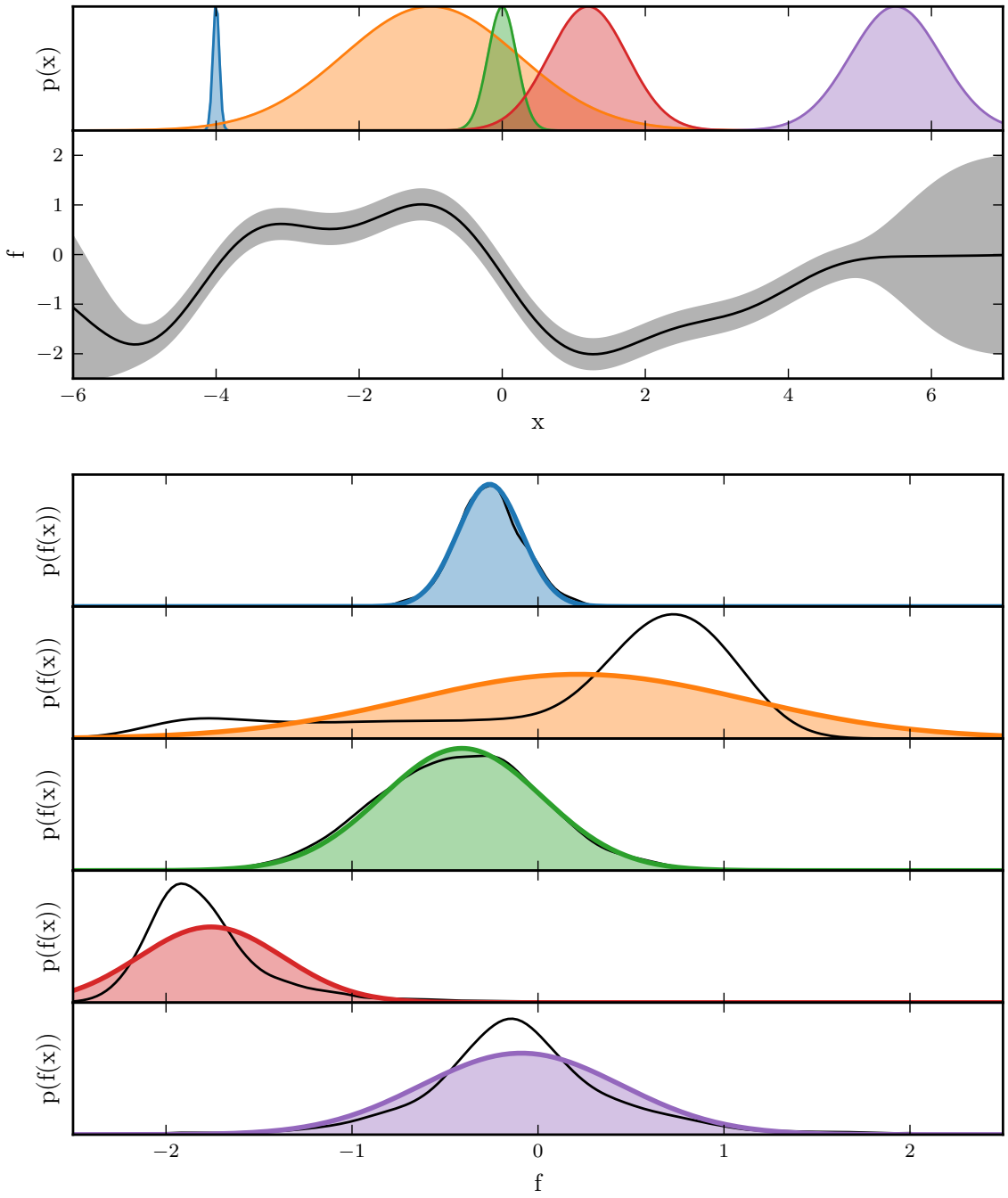


Fig. 3.2 An illustration of propagating a distribution over the inputs through a non-linear function, and the Gaussian projection approximation. The top plot shows the distributions over inputs, with different means and variances. The second plot shows a distribution of possible non-linear functions; these functions are the transition dynamics in the GPSSM case, and the input warping in the GPLVM case. The next five plots show the exact marginal distributions of the output values and the approximate Gaussian projections. Note that the exact marginals are an infinite mixture of Gaussians, and can be heavy-tailed or multi-modal — we approximate this by the simple Monte Carlo estimate using a large number of samples. The Gaussian approximations are uni-modal and possess the same means and variances as the exact non-analytic marginals.

will affect the distribution over the function ouputs, when the input distribution is passed through the linearised approximation. Specifically, passing the Gaussian cavity distribution over the input through the function above leads to a closed-form Gaussian distribution $\overline{q}(\mathbf{c}) \propto \mathcal{N}(\mathbf{c}; \overline{\mathbf{m}}_\mathbf{c}, \overline{\mathbf{S}}_\mathbf{c})$, where,

$$\overline{\mathbf{m}}_\mathbf{c} = \mathbf{A}_{t,\mathbf{m}_{\mathbf{x}_{t-1}}^{\backslash t}} \mathbf{m}_\mathbf{u}^{\backslash t} + \mathbf{W}\mathbf{m}_{\mathbf{x}_{t-1}}^{\backslash t}, \tag{3.71}$$

$$\overline{\mathbf{S}}_\mathbf{c} = \mathbf{B}_{t,\mathbf{m}_{\mathbf{x}_{t-1}}^{\backslash t}} + \mathbf{A}_{t,\mathbf{m}_{\mathbf{x}_{t-1}}^{\backslash t}} \mathbf{S}_\mathbf{u}^{\backslash t} \mathbf{A}_{t,\mathbf{m}_{\mathbf{x}_{t-1}}^{\backslash t}}^\intercal + \mathbf{d}_t \mathbf{S}_{\mathbf{x}_{t-1}}^{\backslash t} \mathbf{d}_t^\intercal + \mathbf{H} + \alpha^{-1}\mathbf{Q}, \tag{3.72}$$

where $\mathbf{d}_t$ is a vector whose elements are $\{\frac{\partial \mathbf{A}_t}{\partial x_{t-1,d}} \mathbf{m}_\mathbf{u}^{\backslash t}\}_{d=1}^{D_\mathbf{x}}$. This linearisation procedure is illustrated in fig. 3.3, which shows the approximation is often over-confident and potentially very damaging if the aproximate distribution is used for the next stage (such as computing $\log \tilde{Z}$). Similar to the moment matching approximation above, this analytic Gaussian approximation leads to an analytic approximation to to $\tilde{Z}$ as follows,

$$\tilde{Z} \approx \frac{(2\pi)^{D_\mathbf{x}/2} |\alpha^{-1}\mathbf{Q}|^{1/2}}{(2\pi)^{\alpha D_\mathbf{x}/2} |\mathbf{Q}|^{\alpha/2}} \mathcal{N}(\mathbf{c}; \overline{\mathbf{m}}_\mathbf{c}, \overline{\mathbf{S}}_\mathbf{c}). \tag{3.73}$$

3.4.2 Dealing with the emission factor $p(\mathbf{y}_t|\mathbf{x}_{t-1})$

Following the same Power EP steps as above, we can derive the iterative updates for $\gamma_t(\mathbf{x}_t)$ such that it approximates the effect of the emission likelihood $p(\mathbf{y}_t|\mathbf{x}_{t-1})$ on the posterior. In detail, the posterior over the latent variable $\mathbf{x}_t$, the cavity distribution when a fraction of $\gamma_t(\mathbf{x}_t)$ is removed, and the tilted distributions when a fraction of $p(\mathbf{y}_t|\mathbf{x}_t)$ is put back are as follows,

$$q(\mathbf{x}_t) = \phi_t(\mathbf{x}_t)\phi_{t+1}(\mathbf{x}_t)\gamma_t(\mathbf{x}_t) \tag{3.74}$$

$$q^{\backslash t}(\mathbf{x}_t) = \phi_t(\mathbf{x}_t)\phi_{t+1}(\mathbf{x}_t)\gamma_t^{1-\alpha}(\mathbf{x}_t) \tag{3.75}$$

$$\tilde{q}(\mathbf{x}_t) = q^{\backslash t}(\mathbf{x}_t)p^\alpha(\mathbf{y}_t|\mathbf{x}_t) \tag{3.76}$$

We have assumed in the previous section that the emission likelihood takes a Gaussian form, $p(\mathbf{y}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{y}_t; \mathbf{U}\mathbf{x}_t, \mathbf{R}_\mathbf{y})$, and that the approximate factor $\gamma_t(\mathbf{x}_t)$ is also a Gaussian over $\mathbf{x}_t$. As such, it is straightforward to show that in the Gaussian case, the parameters for $\gamma_t(\mathbf{x}_t)$ are identical to that of $p(\mathbf{y}_t|\mathbf{x}_t)$, when $p(\mathbf{y}_t|\mathbf{x}_t)$ is viewed as a Gaussian over $\mathbf{x}_t$,

$$\gamma_t(\mathbf{x}_t) \propto \exp(-\frac{1}{2}\mathbf{x}_t\mathbf{U}^\intercal\mathbf{R}_\mathbf{y}\mathbf{U}\mathbf{x}_t^\intercal + \mathbf{x}_t\mathbf{U}^\intercal\mathbf{R}_\mathbf{y}\mathbf{y}). \tag{3.77}$$

Note that this optimal form does not depend on the power parameter of Power EP, α. However, the log-normaliser of the tilted distribution and hence the contribution of the emission likelihood towards the approximate marginal likelihood does depend on α, as

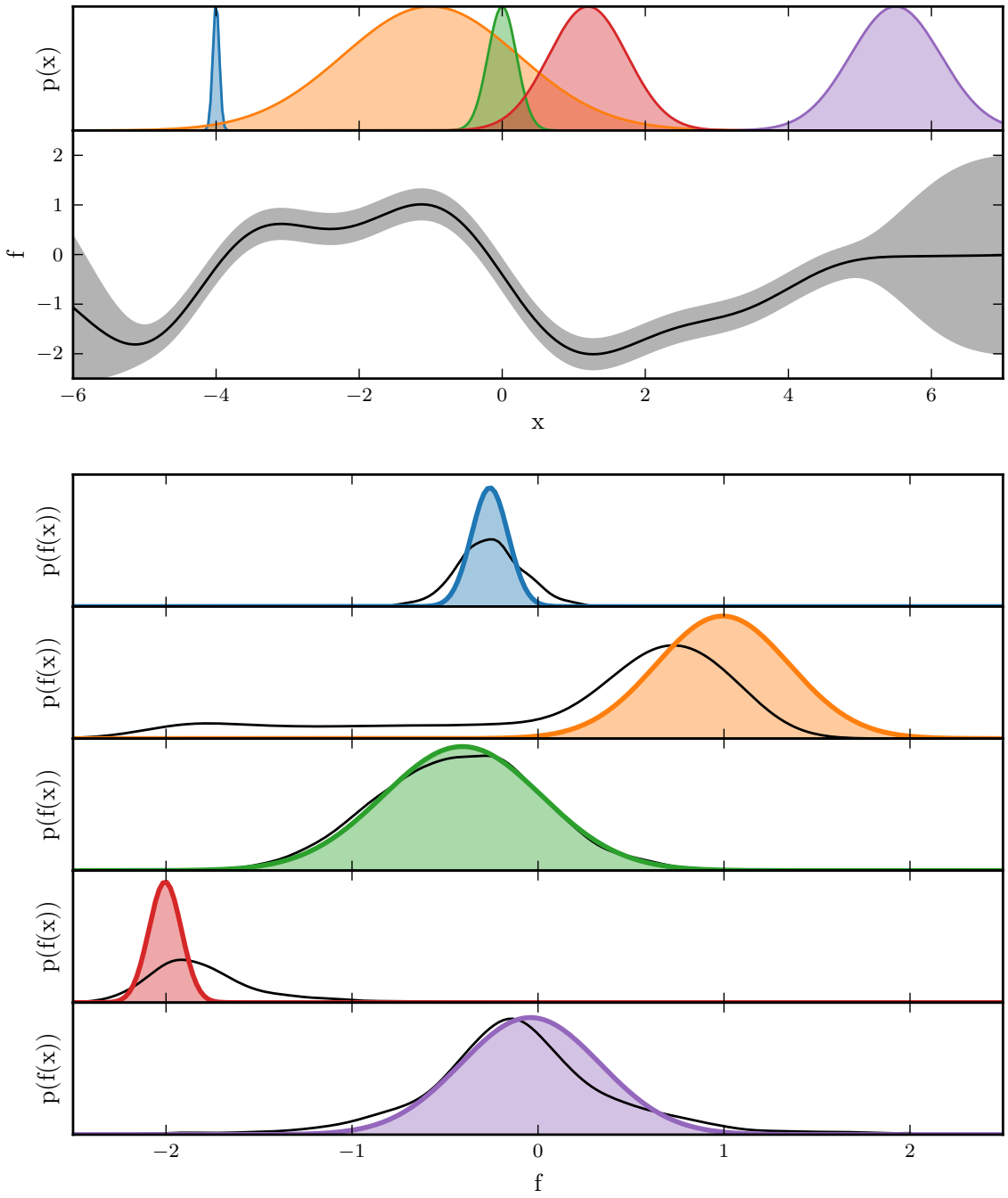


Fig. 3.3 An illustration of propagating a distribution over the inputs through a non-linear function and the linearisation approximation. The top plot shows the distributions over inputs, with different means and variances. The second plot shows a distribution of possible non-linear functions; these functions are the transition dynamics in the GPSSM case, and the input warping in the GPLVM case. The next five plots show the exact marginal distributions of the output values and the approximate Gaussian projections. Note that the exact marginals are an infinite mixture of Gaussians, and can be heavy-tailed or multi-modal — we approximate this by the simple Monte Carlo estimate using a large number of samples. The Gaussian approximations are formed by passing the input distributions through a linear approximation of the function around the input mean. This approximation is accurate in the region where the function is linear, and undesirably inaccurate at the wiggly peaks of the function.

follows,

$$\tilde{Z} = \int \mathrm{d}\mathbf{x}_t q^{\backslash t}(\mathbf{x}_t) p^{\alpha}(\mathbf{y}_t|\mathbf{x}_t) = \frac{(2\pi)^{D_\mathbf{y}/2}|\alpha^{-1}\mathbf{R}_\mathbf{y}|}{(2\pi)^{\alpha D_\mathbf{y}/2}|\mathbf{R}_\mathbf{y}|^{\alpha}} \mathcal{N}(\mathbf{y}_t; \mathbf{U}\mathbf{m}_{\mathbf{x}_t}^{\backslash t}; \mathbf{U}\mathbf{S}_{\mathbf{x}_t}^{\backslash t}\mathbf{U}^\intercal + \alpha^{-1}\mathbf{R}_\mathbf{y}). \quad (3.78)$$

3.4.3 Power EP energy and hyperparameter optimisation

The previous sections have detailed how the Power EP procedure iteratively updates and refines the approximate posterior. When this procedure converges, the resulting fixed point is a stationary point of an energy function, called the Power EP energy, which can be treated as an approximation to the negative marginal likelihood (Minka, 2001a). In contrast to the variational free-energy (which we will show to be the Power EP energy as $\alpha \to 0$), there is no guarantee for this energy to be an upper bound of the negative marginal likelihood for general α. This is a caveat when using the Power EP energy for hyperparameter optimisation, as a lower bound when minimised can go arbitrarily small. However, Power EP often works well in practice, and the energy is often close to the exact negative marginal likelihood in many models. For GPSSMs, the negative Power EP energy can be obtained in closed form after performing the Power EP iterative procedure, as follows,

$$\mathcal{F}_{\mathrm{pep}}(\theta) = \Phi[q(\mathbf{x}_{0:T}, \mathbf{u})] - \Phi[p(\mathbf{x}_0)] - \Phi[p(\mathbf{u})] + \frac{1}{\alpha}\sum_{t=1}^{T}\left(\mathcal{F}_{\mathrm{pep},\ \mathrm{dyn},\ t} + \mathcal{F}_t^{\mathrm{pep,emi}}\right), \quad (3.79)$$

$$\text{and } \mathcal{F}_t^{\mathrm{pep,dyn}} = \log \tilde{Z}_{\mathrm{dyn},\ t} - \Phi[q(\mathbf{x}_{t-1:t}, \mathbf{u})] + \Phi[q^{\backslash t}(\mathbf{x}_{t-1:t}, \mathbf{u})], \quad (3.80)$$

$$\mathcal{F}_t^{\mathrm{pep,emi}} = \log \tilde{Z}_{\mathrm{emi},\ t} - \Phi[q(\mathbf{x}_t)] + \Phi[q^{\backslash t}(\mathbf{x}_t)], \quad (3.81)$$

$$\Phi[\mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{S})] = \frac{1}{2}\mathbf{m}\mathbf{S}^{-1}\mathbf{m}^\intercal + \frac{1}{2}\log|\mathbf{S}|. \quad (3.82)$$

Note that the approximations to the log normaliser of the tilted distributions are already needed and computed for the Power EP iterative procedure, so the energy above can be computed at a very small extra cost of computing the log-partition function of the Gaussian cavity and posterior distributions.

The gradients of the energy w.r.t. the model hyperparameters can also be computed, allowing gradient-based optimisers to be deployed. Additionally, the energy in eq. (3.79) involves a sum over the time steps and is hence amenable to stochastic optimisation. In detail, similar to the stochastic treatment to the uncollapsed variational free-energy, a random subsequence can be used to obtain an unbiased, noisy estimate of the (approximate) Power EP energy as follows,

$$\mathcal{F}_{\mathrm{PEP}}(\theta) \approx \Phi[q(\mathbf{x}_{0:T}, \mathbf{u})] - \Phi[p(\mathbf{x}_0)] - \Phi[p(\mathbf{u})] + \frac{T}{B}\frac{1}{\alpha}\sum_{b=1}^{B}\left(\mathcal{F}_{\mathrm{pep},\ \mathrm{dyn},b} + \mathcal{F}_{\mathrm{pep},\ \mathrm{emi},b}\right). \quad (3.83)$$

3.4.4 When VFE is recovered, as $\alpha \to 0$?

We have shown in eq. (3.45) that the approximate posterior formed by the proposed factorisation used in this session is identical to the approximate posterior used in the VFE case, where we have assumed a mean-field approximation for the latent variables $\mathbf{x}_{0:T}$ (i.e. the diagonal Gaussian parameterisation for $q(\mathbf{x}_{0:T})$ presented in section 3.3). As $\alpha \to 0$, inference and learning using Power EP are equivalent to that using the VFE technique, and the exact Power EP energy is identical to the variational free-energy. However, unlike the variational free-energy, the Power EP for GPSSMs is not analytically available and requires additional approximations. The focus of this section is to examine how different approximations of the log-normaliser of the tilted distributions $\log \tilde{Z}_{\text{dyn},\,t}$ including simple Monte Carlo, Gaussian projection and linearisation affect the convergence to the variational free-energy.

As a reminder, the uncollapsed variational free-energy in eq. (3.14) for a mean-field Gaussian $q(\mathbf{x}_{0:T})$ requires no further approximation and is available in closed-form as follows,

$$\mathcal{F}_{\text{vfe}}(.) = -\text{KL}(q(\mathbf{u})||p(\mathbf{u})) + \sum_{t=0}^{T} \mathcal{H}(q(\mathbf{x}_t)) + \langle \log p(\mathbf{x}_0) \rangle_{q(\mathbf{x}_0)} + \sum_{t=1}^{T} \mathcal{F}_{\text{vfe, dyn},\,t} + \sum_{t=1}^{T} \mathcal{F}_{\text{vfe, emi},\,t},$$

where $\mathcal{F}_{\text{vfe, dyn},\,t}$, $\mathcal{F}_{\text{vfe, emi},\,t}$, $\text{KL}(q(\mathbf{u})||p(\mathbf{u}))$ and $\langle \log p(\mathbf{x}_0) \rangle_{q(\mathbf{x}_0)}$ are detailed in eqs. (3.32) to (3.35), and $\mathcal{H}(q(\mathbf{x}_t))$ is the entropy of the posterior $q(\mathbf{x}_t)$. In the case when no additional approximations for $\log \tilde{Z}_{\text{dyn},\,t}$ are needed, the following Maclaurin expansion can be done when α is small:

$$\begin{aligned}
\frac{1}{\alpha} \log \tilde{Z}_{\text{dyn},\,t} &= \frac{1}{\alpha} \log \int \mathrm{d}f \mathrm{d}\mathbf{x}_{t-1:t} q(f, \mathbf{x}_{t-1:t}) p^{\alpha}(\mathbf{x}_t|f, \mathbf{x}_{t-1}) && (3.84)\\
&= \frac{1}{\alpha} \log \int \mathrm{d}f \mathrm{d}\mathbf{x}_{t-1:t} q(f, \mathbf{x}_{t-1:t}) [1 + \alpha \log p(\mathbf{x}_t|f, \mathbf{x}_{t-1}) + \xi(\alpha^2)] && (3.85)\\
&= \frac{1}{\alpha} \log [1 + \alpha \int \mathrm{d}f \mathrm{d}\mathbf{x}_{t-1:t} q(f, \mathbf{x}_{t-1:t}) \log p(\mathbf{x}_t|f, \mathbf{x}_{t-1}) + \alpha^2 \xi(1)] && (3.86)\\
&= \int \mathrm{d}f \mathrm{d}\mathbf{x}_{t-1:t} q(f, \mathbf{x}_{t-1:t}) \log p(\mathbf{x}_t|f, \mathbf{x}_{t-1}) + \alpha \xi(1), && (3.87)
\end{aligned}$$

and thus,

$$\lim_{\alpha \to 0} \frac{1}{\alpha} \log \tilde{Z}_{\text{dyn},\,t} = \int \mathrm{d}f \mathrm{d}\mathbf{x}_{t-1:t} q(f, \mathbf{x}_{t-1:t}) \log p(\mathbf{x}_t|f, \mathbf{x}_{t-1}) = \mathcal{F}_{\text{vfe, dyn},\,t}. \tag{3.88}$$

However, because of the approximation schemes to $\log \tilde{Z}_{\text{dyn},\,t}$, the limit above is potentially no longer equal to $\mathcal{F}_{\text{vfe, dyn},\,t}$. We detail the limit for each approximation scheme below.

Simple Monte Carlo

The integration w.r.t. $q(\mathbf{x}_{t-1})$ in $\log \tilde{Z}_{\text{dyn},\,t}$ can be approximated using simple Monte Carlo, as we have shown in eq. (3.65). It is useful here to consider the raw form of the Monte Carlo

estimate,

$$\log \tilde{Z}_t^{\text{dyn,mc}} = \log \frac{1}{L} \sum_{l=1}^{L} \int \mathrm{d}f \mathrm{d}\mathbf{x}_t q(f, \mathbf{x}_t) p^{\alpha}(\mathbf{x}_t | f, \mathbf{x}_{t-1,l}), \tag{3.89}$$

where L is the number of samples drawn from $q(\mathbf{x}_{t-1})$, as an analysis for small α similar to the above can be done leading to,

$$\lim_{\alpha \to 0} \frac{1}{\alpha} \log \tilde{Z}^{\text{dyn,mc}_t} = \frac{1}{L} \sum_{l=1}^{L} \int \mathrm{d}f \mathrm{d}\mathbf{x}_t q(f, \mathbf{x}_t) \log p(\mathbf{x}_t | f, \mathbf{x}_{t-1,l}). \tag{3.90}$$

The right hand size of eq. (3.90) is exactly a Monte Carlo approximation of $\mathcal{F}_{\text{vfe, dyn, } t}$. Therefore, when the number of $\mathbf{x}_{t-1}$ samples is large and as $\alpha \to 0$, $\frac{1}{\alpha} \log \tilde{Z}_{\text{dyn, } t}$ tends exactly to $\mathcal{F}_{\text{vfe, dyn, } t}$. In this case, the bias introduced by applying a logarithmic operation to a Monte Carlo estimate tends to 0.

Moment matching

Taking the log of the approximation in eq. (3.70) gives,

$$\begin{aligned} \log \tilde{Z}_t^{\text{dyn,mm}} = &-\frac{\alpha}{2} \log[(2\pi)^{D_{\mathbf{x}}} |\mathbf{Q}|] - \frac{1}{2} \log[\mathrm{I} + \alpha \hat{\mathbf{S}}_{\mathbf{c}} \mathbf{Q}^{-1}] \\ &- \frac{\alpha}{2} (\mathbf{c} - \overline{\mathbf{m}}_{\mathbf{c}}) (\alpha \hat{\mathbf{S}}_{\mathbf{c}} + \mathbf{Q})^{-1} (\mathbf{c} - \overline{\mathbf{m}}_{\mathbf{c}})^{\intercal}, \end{aligned}$$

where $\hat{\mathbf{S}}_{\mathbf{c}} = \overline{\mathbf{S}}_{\mathbf{c}} - \alpha \mathbf{Q}$, which does not depend on α (see eq. (3.70)). Noting that as $\alpha \to 0$, the cavity distributions are the approximate posteriors, we can obtain,

$$\begin{aligned} \lim_{\alpha \to 0} \frac{1}{\alpha} \log \tilde{Z}_t^{\text{dyn,mm}} = &-\frac{1}{2} \log[(2\pi)^{D_{\mathbf{x}}} |\mathbf{Q}|] - \frac{1}{2} \operatorname{tr}[\hat{\mathbf{S}}_{\mathbf{c}} \mathbf{Q}^{-1}] \\ &- \frac{1}{2} (\mathbf{c} - \overline{\mathbf{m}}_{\mathbf{c}}) \mathbf{Q}^{-1} (\mathbf{c} - \overline{\mathbf{m}}_{\mathbf{c}})^{\intercal}. \end{aligned} \tag{3.91}$$

Substituting $\overline{\mathbf{m}}_b$ and $\overline{\mathbf{S}}_b$ from eq. (3.68) and eq. (3.69) into the equation above gives the result *identical* to eq. (3.32), that is $\mathcal{F}_t^{\text{vfe,dyn}}$. The reason for this surprising equality is that the variational free energy only requires the first and second moments of the non-Gaussian output distributions (after passing a Gaussian through a non-linearity), and the additional moment matching approximation presented above preserves these quantities exactly.

Linearisation

The derivation for the moment matching approximation above up to eq. (3.91) can be reused for the linearisation approximation, as eqs. (3.70) and (3.73) are identical. However, as $\overline{\mathbf{m}}_b$ and $\overline{\mathbf{S}}_b$ in eqs. (3.71) and (3.71) are different compared to the moment matching case, the

limit in eq. (3.91) is not equal to $\mathcal{F}_t^{\text{vfe,dyn}}$, and hence the approximate Power EP energy with linearisation will not tend to the variational free-energy when $\alpha \to 0$.

3.4.5 Short summary

The Power EP energy presentation above completes the recipe for approximate posterior inference and hyperparameter learning in GPSSMs based on Power EP: run the Power EP iterative procedure (using simple Monte Carlo, Gaussian projection, or linearisation to find the approximate updates) until convergence, perform a gradient step to minimise the Power EP w.r.t. the hyperparameters and repeat. The Power EP also enables these propagation approximations what were used previously in EP to be deployed in VFE and for intermediate α values. This may have computational and algorithmic advantages in some settings. Additionally, each factor can have its own private α value, which means multiple approximation schemes with different α powers can be used in a single algorithmic procedure.

3.5 The approximate Power EP approach

Power EP is a general and flexible framework for approximate inference and learning, as shown in chapter 2 for GP regression and classification and the previous section for GPSSMs. Importantly, intermediate α values have been shown to be advantageous compared to VFE and EP in certain model classes (e.g. see chapter 2). However, this flexibility does come at a cost, as previously discussed in section 2.6. For clarity, we list the points discussed in section 2.6 here and add a couple of GPSSM-specific points,

- Hyperparameter updates and posterior inference need to be interleaved during learning, that is, there is no single objective function or procedure for learning both the hyperparameters and approximate posterior at the same time. Optimising the Power EP energy to obtain the approximate posterior alone (instead of running the iterative procedure) is also not straightforward, as non-standard, double-loop schemes needed to be deployed (Heskes and Zoeter, 2002). The VFE approach, on the other hand, provides a lower bound to the marginal likelihood, which can be optimised to concurrently learn both the hyperparameters and the approximate posterior. There are ways to side-step this problem, for example, not waiting for Power EP to converge before performing an update for the hyperparameters. However, this remains as an arguably major reason why Power EP is not used more widely in practice when hyperparameter optimisation is required.

- Numerical stability is a known issue for Power EP and in particular when using Power EP for GPSSMs (McHutchon, 2014). The Power EP update equations do not guarantee the new posterior covariance to be positive definite (or positive if the variable is single

dimensional), forcing additional heuristics such as damping to be used. In VFE, as the approximate posterior's (co)variances are parameterised under a specific transformation (for example: for the structured Gaussian case, the covariance matrix is parameterised using its triangular Cholesky decomposition), they are guaranteed to be positive definite.

- The sequential update nature of Power EP is problematic for long time series, as multiple passes over the training data are needed for convergence. Parallel updates can be used instead, but are prone to further numerical problems. Techniques such as damping or skipping can be used (see e.g. Minka and Lafferty, 2002), but they are not sufficient for all cases.

- For a long time series, a high-dimensional latent space and a large number of pseudo-points, the memory required to parameterise all the approximate factors is high and could be out of reach. Techniques such as average or stochastic EP (Li et al., 2015; Dehaene and Barthelmé, 2015) can significantly reduce this memory complexity. However, though this memory limitation is not a major focus of this chapter, it turns out that the trick employed in stochastic EP to reduce the memory constraint can be used to sidestep other problems.

As mentioned above, stochastic EP greatly reduces the memory complexity of Power EP. This is achived by using the same parameterisation for similar factors, enforcing an identical contribution from each factor to the posterior. The application of this approximation to GPSSMs is illustrated in fig. 3.1(C,D). Each common factor could be thought of as the average effect each factor has on the posterior, for example, in fig. 3.1(D), λ_t in the approximate scheme is the average contribution towards the posterior from λ_t, β_{t+1} and γ_t in the original factorisation in fig. 3.1(C).

Having described an approximate factorisation, one could proceed to run the Power EP iterative procedure (Li et al., 2015), and then perform hyperparameter optimisation as an outer loop as with Power EP. However, Hernández-Lobato et al. (2016) noticed that the factor tying approximation above turns the original minimax Power EP energy optimisation problem into a minimisation problem. In other words, in a similar fashion to the VFE approach, the approximate Power EP energy can now be optimised using standard optimisation techniques, to find *both* the approximate posterior and the hyperparameters.

3.6 Predictions

Given the approximate posterior over the hidden variables and the non-linear dynamics, we wish to forecast or predict the future observations. First, we consider a one-step prediction

task in which the object of interest is,

$$\begin{aligned} p(\mathbf{y}^*_{T+1}|\mathbf{y}_{1:T}) &= \int p(\mathbf{y}^*_{T+1}|\mathbf{x}^*_{T+1})p(\mathbf{x}^*_{T+1}|\mathbf{y}_{1:T})\mathrm{d}\mathbf{x}^*_{T+1} \\ \text{where } \; p(\mathbf{x}^*_{T+1}|\mathbf{y}_{1:T}) &= \int p(\mathbf{x}^*_{T+1}|\mathbf{x}_T, f)p(\mathbf{x}_T, f|\mathbf{y}_{1:T})\mathrm{d}f\mathrm{d}\mathbf{x}_T \\ &\approx \int p(\mathbf{x}^*_{T+1}|\mathbf{x}_T, f)q(\mathbf{x}_T)q(f)\mathrm{d}f\mathrm{d}\mathbf{x}^*_T, \end{aligned} \tag{3.92}$$

and we have replaced the exact posterior by the approximate posterior in eq. (3.92). Note that the computation required in eq. (3.92) is nearly identical to that required for the log normaliser of the tilted distribution in eq. (3.49). In particular, we have to find a GP predictive distribution with a Gaussian input – this is analytically intractable. Fortunately, all approximate propagation techniques discussed in section 3.4, such as Gaussian projection and simple Monte Carlo, can be used here. These techniques approximates the predictive distribution $p(\mathbf{x}^*_{T+1}|\mathbf{y}_{1:T})$ by a single Gaussian distribution or by a set of samples, which can be mapped through the emission model to obtain the approximate one-step prediction.

By following the similar procedure, the approximate predictive distribution at the k-th step in the future can also be obtained as follows,

$$\begin{aligned} p(\mathbf{y}^*_{T+k}|\mathbf{y}_{1:T}) &= \int p(\mathbf{y}^*_{T+k}|\mathbf{x}^*_{T+k})p(\mathbf{x}^*_{T+k}|\mathbf{y}_{1:T})\mathrm{d}\mathbf{x}^*_{T+k} \\ \text{where } \; p(\mathbf{x}^*_{T+k}|\mathbf{y}_{1:T}) &\approx \int p(\mathbf{x}^*_{T+k}|\mathbf{x}^*_{T+k-1}, f)q(\mathbf{x}^*_{T+k-1})q(f)\mathrm{d}f\mathrm{d}\mathbf{x}^*_{T+k-1}, \end{aligned}$$

which means the approximate predictive distribution at one time step can be passed forwards to make prediction at the next step.

3.7 The Gaussian process latent variable model

In this section, we consider a class of models closely-related to the GPSSM, namely the Gaussian process latent variable model (GPLVM, Lawrence, 2005). In detail, given a set of N D-dimensional observations, $\mathbf{Y} = \{\mathbf{y}_n\}_{n=1}^N$, a GPLVM assumes that there are N Q-dimensional latent variables, $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$, one for each observation, and that the mapping from the latent variables to each dimension of the observed data is a GP. The generative model can be summarised as follows,

$$p(\mathbf{X}) = \prod_{n=1}^{N} p(\mathbf{x}_n), \tag{3.93}$$

$$p(f_d) = \mathcal{GP}(0, k(\cdot,\cdot)) \;\text{ for }\; d = 1:D \tag{3.94}$$

$$p(\mathbf{Y}|\mathbf{X}, f_{1:D}) = \prod_{n=1}^{N}\prod_{d=1}^{D} \mathcal{N}(y_{n,d}; f_d(\mathbf{x}_n), \sigma_y^2), \tag{3.95}$$

where we have assumed a factorised Gaussian likelihood, but other observation models can be easily accommodated (see e.g. Gal et al., 2015). Typically, the dimensionality of the latent variables is smaller than that of the data, and in this case, learning the latent variables given the observed data is a form of probabilistic nonlinear dimensionality reduction.

The GPLVM could be thought of as an extension to the GP regression model when the inputs are random variables, or a special case of the GPSSM when there are N time series, each with only one time step and the emission model is a δ-function. As a result, the approximate inference and learning procedure discussed for GPSSMs can be directly applied to the GPLVM case. As they are straightforward to derive and given the similarity to the previous sections, we will not provide a detailed derivation of each method here. We summarise the existing literature and their relationships in table 3.2. The form of the approximate posterior used in Power-EP and VFE are pictorially depicted in fig. 3.4[B], which is an approximation to the exact posterior/factor graph displayed in fig. 3.4[A]. Similar to the GPSSM case, the factors for the global variable (the GP mappings) can be tied (as shown in fig. 3.1[C]) to give an approximate Power-EP energy which can be directly optimised, as discussed in section 3.5. And similar to the GPSSM case, as $\alpha \to 0$, the VFE approach presented by Titsias and Lawrence (2010) is recovered.

Table 3.2 Approximation schemes that have been used for GPLVMs. In all cases (except MAP), $q(\mathbf{u})$ is assumed Gaussian, $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{m}, \mathbf{S})$.

Inf. method	$q(\mathbf{x}_{0:N})$	Notes and references
MAP	MAP	Lawrence (2005) Lawrence and Quiñonero-Candela (2006) used inf. networks to parameterise $\mathbf{x}_{0:N}$
VFE	Gaussian	Titsias and Lawrence (2010), $q(\mathbf{u})$ collapsable uncollapsed energy can be distributed (Gal et al., 2014) or parallelised (Dai et al., 2014) Gal et al. (2015) used simple MC to evaluate the energy for non-Gaussian likelihoods Bui and Turner (2015) used inf. networks to parameterise $q(\mathbf{x}_{0:N})$
Power EP	Gaussian	factors can be tied, $\alpha \to 0$ gives VFE, see figs. 3.4[B, C]

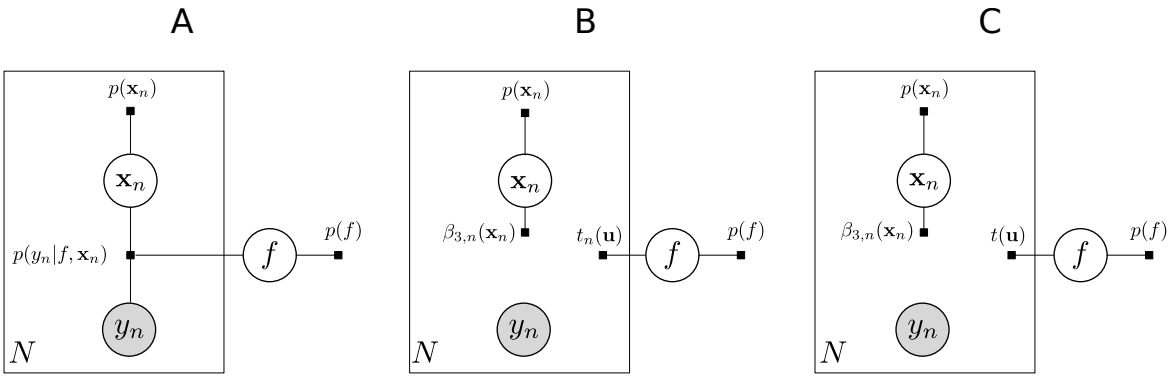


Fig. 3.4 A: a factor graph of the Gaussian process latent variable model. Note that while there are many factor graphs that can be used to show the same model, our approximation is based on this representation. B: a factor graph showing approximate factors and the variables that each factor involves. This factor graph assumes a mean-field structure between the latent variables and the latent GP mapping. C: a factor graph showing how factors involving the latent function can be tied. Best viewed in colour.

3.8 Experiments

In this section, we detail several experiments comparing the VFE approach (section 3.3) and the approximate Power EP approach (section 3.5) on learning one toy and one real-world non-linear dynamics. A mean-field diagonal Gaussian approximation over the hidden state space variables, as shown in section 3.3.4 and fig. 3.1, is used for both methods. The variational free-energy and the approximate Power EP energy are optimised using the Adam optimiser with its standard learning rate (Kingma and Ba, 2015).

3.8.1 Learning a one-dimensional non-linear system

In the first experiment, we compare the approximate negative log marginal likelihood that the VFE and approximate Power EP methods provide, and examine the quantitative performance of each method on a toy time series. This time series consists of 200 time steps, which are generated by simulating the `kink` non-linear system,

$$
\begin{aligned}
x_t &= f(x_{t-1}) + \sigma_x \epsilon_x \\
y_t &= x_t + \sigma_y \epsilon_y \\
\text{where} \qquad f(x_{t-1}) &= \begin{cases} x_{t-1} + 1 & \text{if} \quad x_{t-1} < 4 \\ -4x_{t-1} + 21 & \text{otherwise.} \end{cases} \\
\epsilon_x, \epsilon_y &\sim \mathcal{N}(0, 1),
\end{aligned}
$$

and $\sigma_x^2 = 0.2$, $\sigma_y^2 = 0.05$. To approximate the titled distribution, the Gaussian projection and simple Monte Carlo approximations are used. The number of pseudo-points used for all experiments here is 30.

Training curves and qualitative comparisons of learnt dynamics

The training curves for VFE and approximate Power EP with various α values are shown in fig. 3.5. Note that Power EP with $\alpha = 0.0001$ gives results nearly identical to the VFE approach, which confirms the convergence results discussed in previous sections. Similar to behaviours observed in the GP regression and classification, and deep GP cases, the Power EP energy tends to be smaller for bigger α. However, this does not always translate to better qualitative performance as shown in figs. 3.6 and 3.7. In particular, the VFE approach tends to give high dynamical noise variances, while the approximate Power EP approach with large α (e.g. $\alpha = 1$) learns very small noise variances (for both dynamical noise and observation noise), resulting in over-confident predictions. Additionally, there is no clear difference between the Gaussian projection and simple Monte Carlo approximations in this example.

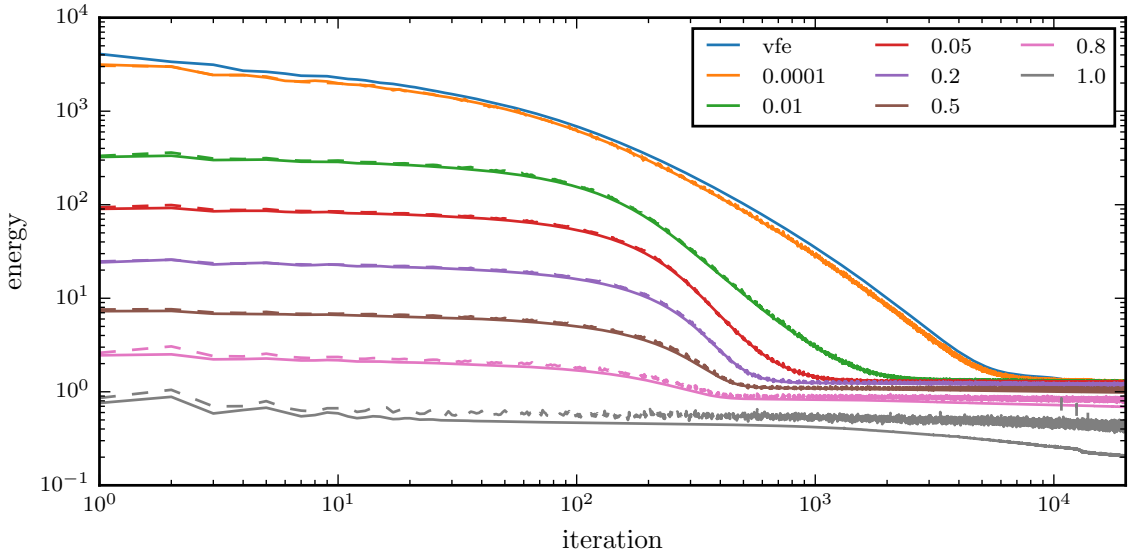
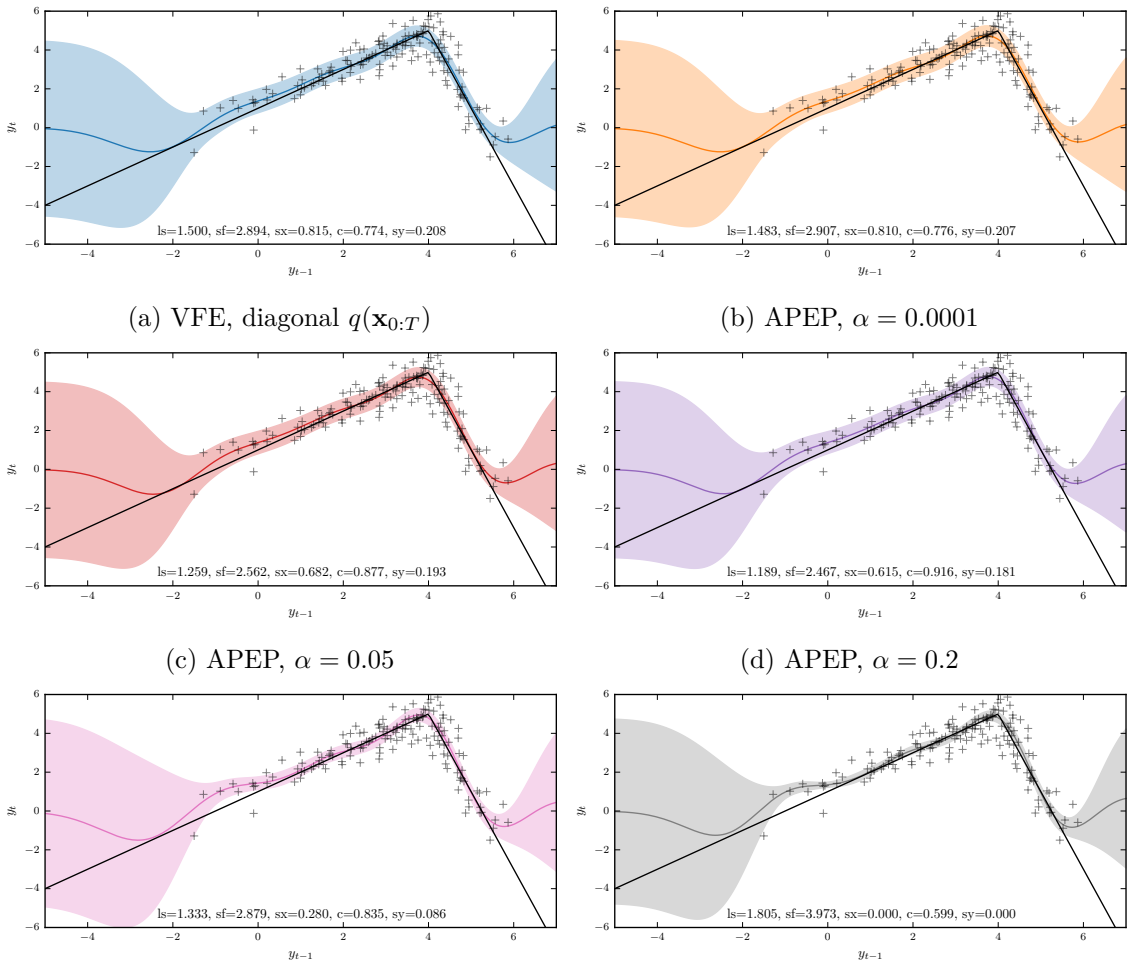


Fig. 3.5 The variational and approximate Power EP energies during training. Solid lines are results using the Gaussian projection to approximate the tilted distribution and dashed lines the simple Monte Carlo approximation with 20 samples. Note that Power EP with $\alpha = 0.0001$ gives results nearly identical to the VFE approach, which confirms the convergence results discussed in the text. Additionally, similar to behaviours observed in the GP regression and classification, and deep GP cases, the Power EP energy tends to be smaller for bigger α. However, this does not always translate to better qualitative performance as shown in figs. 3.6 and 3.7, or better quantitative performance as shown in fig. 3.9 . Additionally, there is no clear difference between the Gaussian projection and simple Monte Carlo approximations in this case. Best viewed in colour.



(a) VFE, diagonal $q(\mathbf{x}_{0:T})$

(b) APEP, $\alpha = 0.0001$

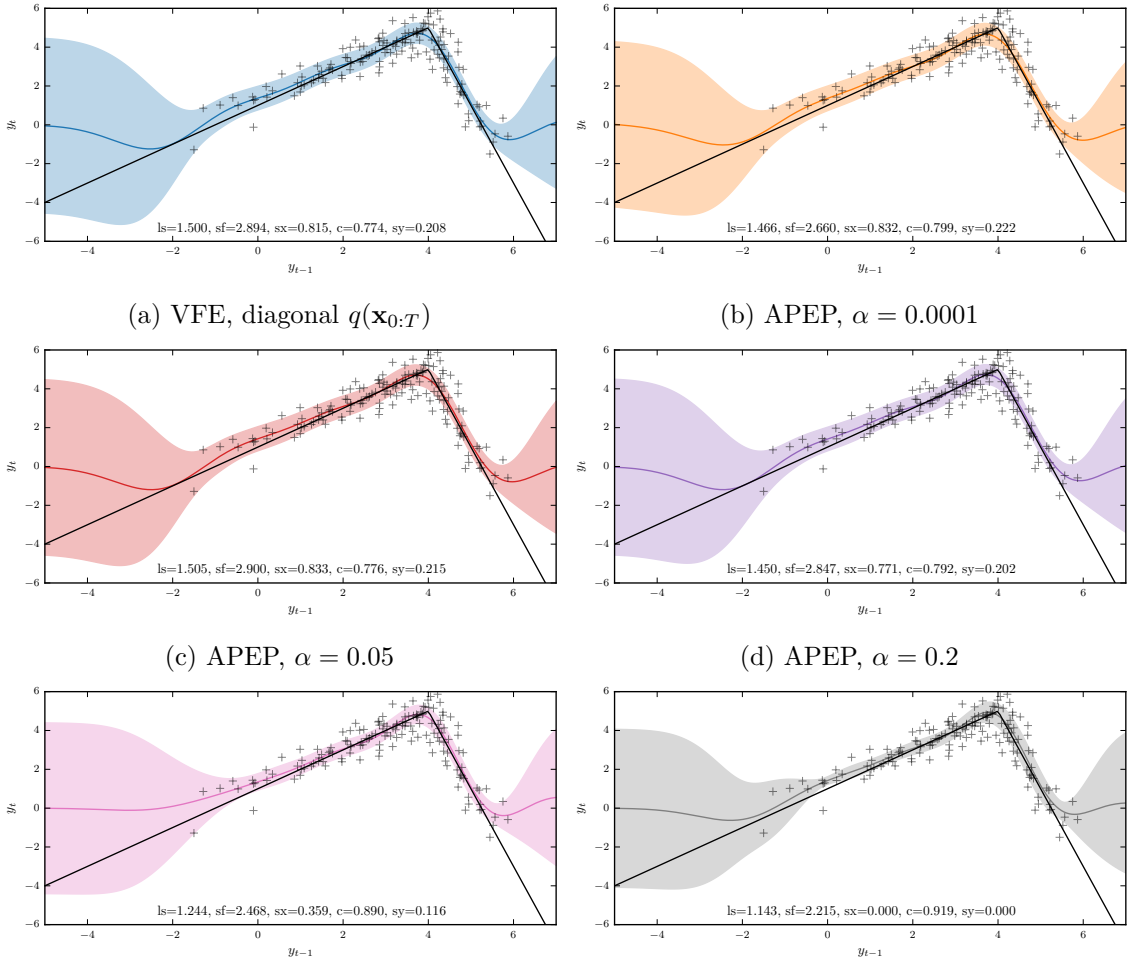
(c) APEP, $\alpha = 0.05$

(d) APEP, $\alpha = 0.2$

(e) APEP, $\alpha = 0.8$

(f) APEP, $\alpha = 1.0$

Fig. 3.6 The dynamics mapping from a time step to the next time step, learnt by the VFE approach and the approximate Power EP approach with the Gaussian projection/moment matching approximation. In each plot, the black line is the ground truth non-linear function, the black dots are the observed values, and the coloured line and shaded area are the mean and confidence interval of the output predictions. The hyperparameter values are included in the plot. Note that the VFE approach tends to give high dynamical noise variances, while the approximate Power EP approach with large α (e.g. $\alpha = 1$) learns very small noise variances for both dynamical noise and observation noise. Best viewed in colour.



(a) VFE, diagonal $q(\mathbf{x}_{0:T})$

(b) APEP, $\alpha = 0.0001$

(c) APEP, $\alpha = 0.05$

(d) APEP, $\alpha = 0.2$

(e) APEP, $\alpha = 0.8$

(f) APEP, $\alpha = 1.0$

Fig. 3.7 The dynamics mapping from a time step to the next time step, learnt by the VFE approach and the approximate Power EP approach with the simple Monte Carlo approximation. See fig. 3.6 for more details about the plots. Similar to the Gaussian projection case in fig. 3.6, the VFE approach tends to give high dynamical noise variances, while the approximate Power EP approach with large α (e.g. $\alpha = 1$) learns very small noise variances for both dynamical noise and observation noise. Best viewed in colour.

Using different propagation techniques for prediction

Having trained the model using various approximate inference and learning methods, we consider a task of predicting the outputs at future time steps. In particular, we use the models trained with approximate Power EP with different α values and make predictions using three different forward propagation techniques: Gaussian projection/moment matching (MM), linearisation (LIN) and simple Monte Carlo (MC). We plot in fig. 3.8 the prediction made by these propagation methods on a time series, as well as the predictive marginals at several future time steps for different α values used to train the model. We can observe that the simple Monte Carlo propagation is the best performing method for prediction as it provides predictive samples that are diverse and structurally similar to the training set, even after many future steps. In contrast, the predictions using the moment matching and linear propagation methods appear very uncertain after only a few steps. We also note that models trained with approximate Power EP with higher α values tend to give more confident predictions, and the Monte Carlo approximation, in this case, tends to produce multi-modal predictive distributions (see fig. 3.8, lower half, bottom right).

We attempt to quantitatively compare the difference between different α values and different propagation methods in fig. 3.9. We repeat the training procedure above for 100 different time-series generated from the `kink` system. The prediction quality is measured by the average log-likelihood the prediction at future time steps. The results shown in fig. 3.9 confirm that the simple Monte Carlo propagation outperforms other propagation methods. The closest competitor is the Gaussian project/moment matching method. Additionally, the results also show models trained with small α values are more competitive (e.g. see $\alpha = 0.001$ vs. $\alpha = 0.8$ in fig. 3.9).

3.8.2 Modelling action potential data generated by the Hodgkin–Huxley model

In this experiment, we attempt to learn a model from synthetic action potential data generated from the Hodgkin-Huxley model (Hodgkin and Huxley, 1952). The Hodgkin–Huxley model is a well-established mathematical model that describes how action potentials in neurons are initiated and propagated. This model can be thought of as a set of four ordinary differential equations driven by an external current I, with four state variables, action potential V and gating values m, h, n, that change with respect to time. The system is difficult to study because it is a non-linear system and cannot be solved analytically. We generate a four-dimensional time series from this system using a series of step and slope injected currents as shown in fig. 3.10. The system is known to have limit cycles when the input current is sufficiently large, as shown in fig. 3.11. Our goal here is to demonstrate the ability of the learning algorithm to learn a non-linear dynamics from data without using any biological insight.

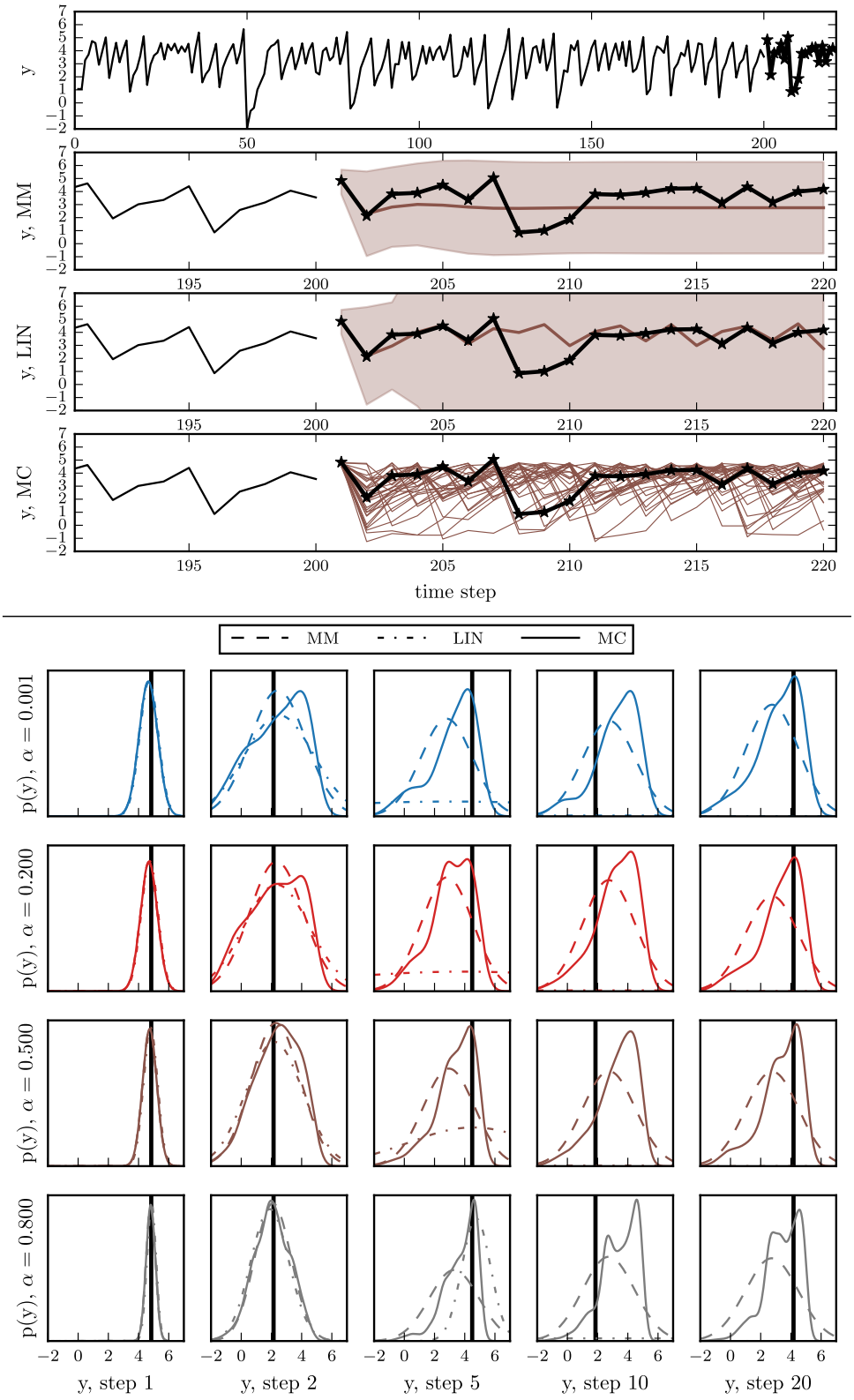


Fig. 3.8 Look-ahead predictions using Gaussian projection, linearisation, and simple Monte Carlo. TOP: a time series and ground truth observations (bold trajectory), and predictive means and variances for 20 future time steps obtained using different propagation methods, after training the model using approximate Power EP with $\alpha = 0.5$. BOTTOM: the marginal densities obtained by various methods using different models trained with various α values. The black lines are placed at the observed values. See text for more details. Best viewed in colour.

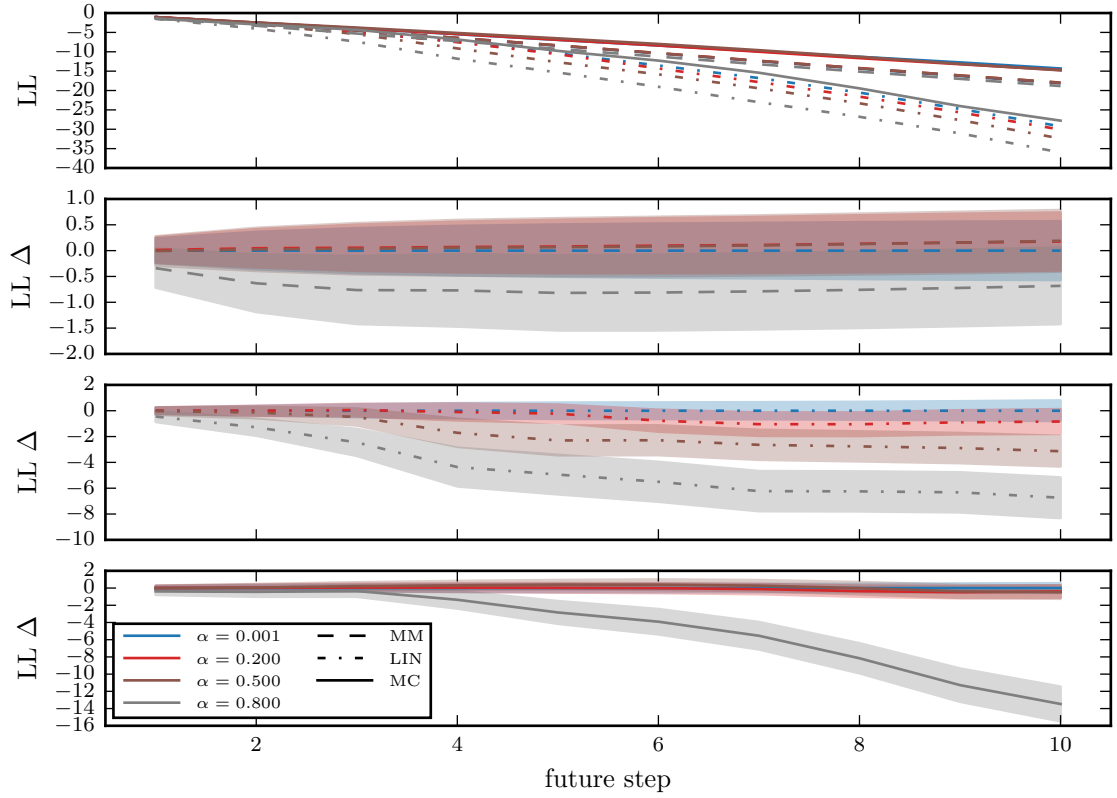


Fig. 3.9 The log likelihood (LL) of the look-ahead predictions given by various approximate propagation techniques on models trained with different α values [TOP], and the difference between the performance of various α values and $\alpha = 0.0001$ for each approximate propagation method [BOTTOM 3]. The higher the log-likelihood (LL), the better the prediction quality. LL Δ is the LL for a method less that of $\alpha = 0.0001$, that is a negative LL Δ means the considered method is worse than approximate Power EP with $\alpha = 0.0001$. The top plot demonstrates that the simple Monte Carlo approximation is the best performing method, and the approach based on linearisation performs poorly. The bottom three plots show a marked difference between the performance of $\alpha = 0.8$ and smaller α values, suggesting that smaller α values are better for training models which can give more accurate predictions at test time. Best viewed in colour.

To model the generated data shown in fig. 3.10, we use a GPSSM with two-dimensional state variables and a GP emission model, which results in 6 GPs to be learnt, 2 for the dynamics model and 4 for the emission model. There are reductions of the Hodgkin-Huxley model that use just two state variables, see e.g. Krinskiĭ and Kokoz (1973), indicating that a two-dimensional state space might be sufficient for modelling purposes. We use 30 pseudo-points for each GP, where the pseudo-inputs for the dynamic GPs are shared and the pseudo-inputs for the emission GPs are shared. In light of the results discussed in the last section, we train the model using approximate Power EP with a small α value, $\alpha = 0.2$, using the Gaussian projection method to approximate the tilted distribution. To evaluate

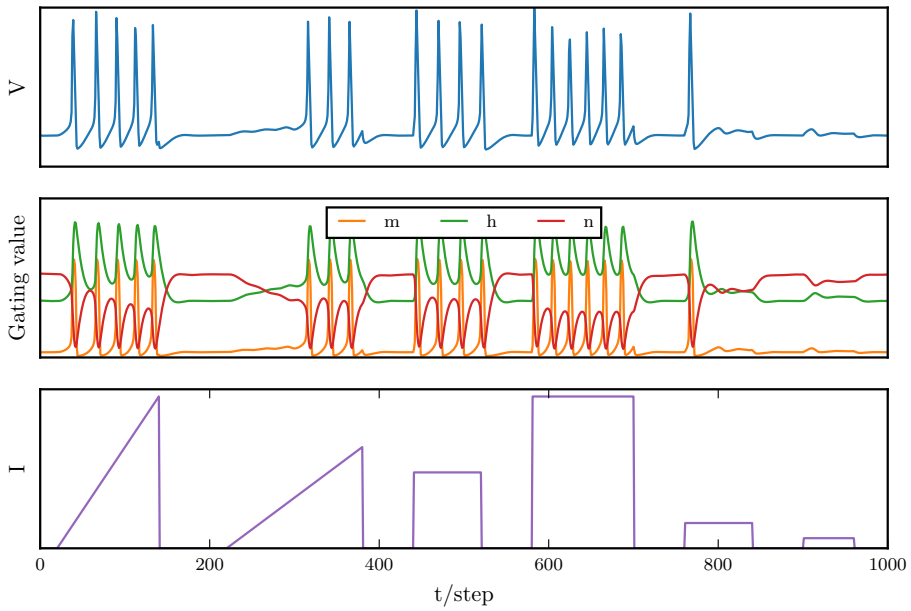


Fig. 3.10 The membrane action potentials, V, and gating channel values, m, h and n, generated by the Hodgkin-Huxley model, given a series of slope and step inject currents, I.

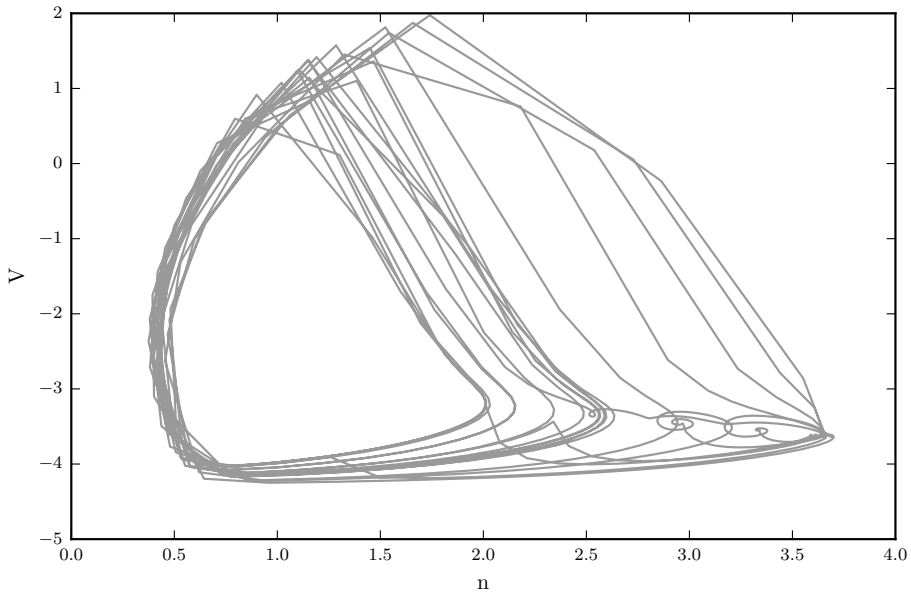


Fig. 3.11 A phase space plot the simulation in fig. 3.10 in terms of the potential V and the gating channel n. The Hodgkin-Huxley is known to have non-linear dynamics, for example the limit cycles shown in this plot.

the trained model, we compute the prediction from the current step given the next injected current for many future time steps. Note that the injected currents at test time are different

from that at training time. The predictions made by using the Gaussian projection (moment matching) propagation and the simple Monte Carlo propagation are shown in figs. 3.12 and 3.13, respectively. The simple Monte Carlo approach gives good predictions for many future time steps, and produces structurally reasonable samples even in the region of unseen control signal (see the quantitative comparison between the predictions and the ground truth in figs. 3.12 and 3.13). In contrast, the moment matching approach gives reasonable predictions only up to about 100 future steps and quickly reverts to underconfident predictions soon after.

3.9 Summary

This chapter reviewed existing approximations and proposed a unifying framework for approximate posterior inference and learning in GP state space models and latent variable models, based on Power EP. Several approximate uncertainty propagation methods in recurrent architectures, based on linearisation, moment matching/Gaussian projection, and simple Monte Carlo, are also discussed. The performance of the proposed framework was assessed and validated on several toy and real-world system identification tasks.

3.10 Extensions

In this section, we detail an alternative approximate posterior for inference and learning in GPSSMs, and a potential application of the proposed inference and learning scheme to active system identification. Early experiments indicate these are potentially promising extensions, but they are left as future work.

3.10.1 An alternative approximate posterior for GPSSMs

The ealier sections in this chapter have introduced several approximate Bayesian schemes for inference and learning in GPSSMs. In particular, a significant effort was spent addressing how to choose a variational distribution for the variables that can retain correlations between variables and admits efficient computation. Choosing such a distribution is not trivial and the previous sections had to resorts to approximations that assume a mean-field structure between the hidden variables and the latent function, and a mean-field or Markovian structure between the hidden variables. Whilst seemingly working well in practice, as shown in section 3.8, potential drawbacks of these approximations include the need to parameterise the variational approximation for the hidden variables and a question of how to sensible initialise this distribution. Inspired by recent work by Salimbeni and Deisenroth (2017) for deep GPs, we

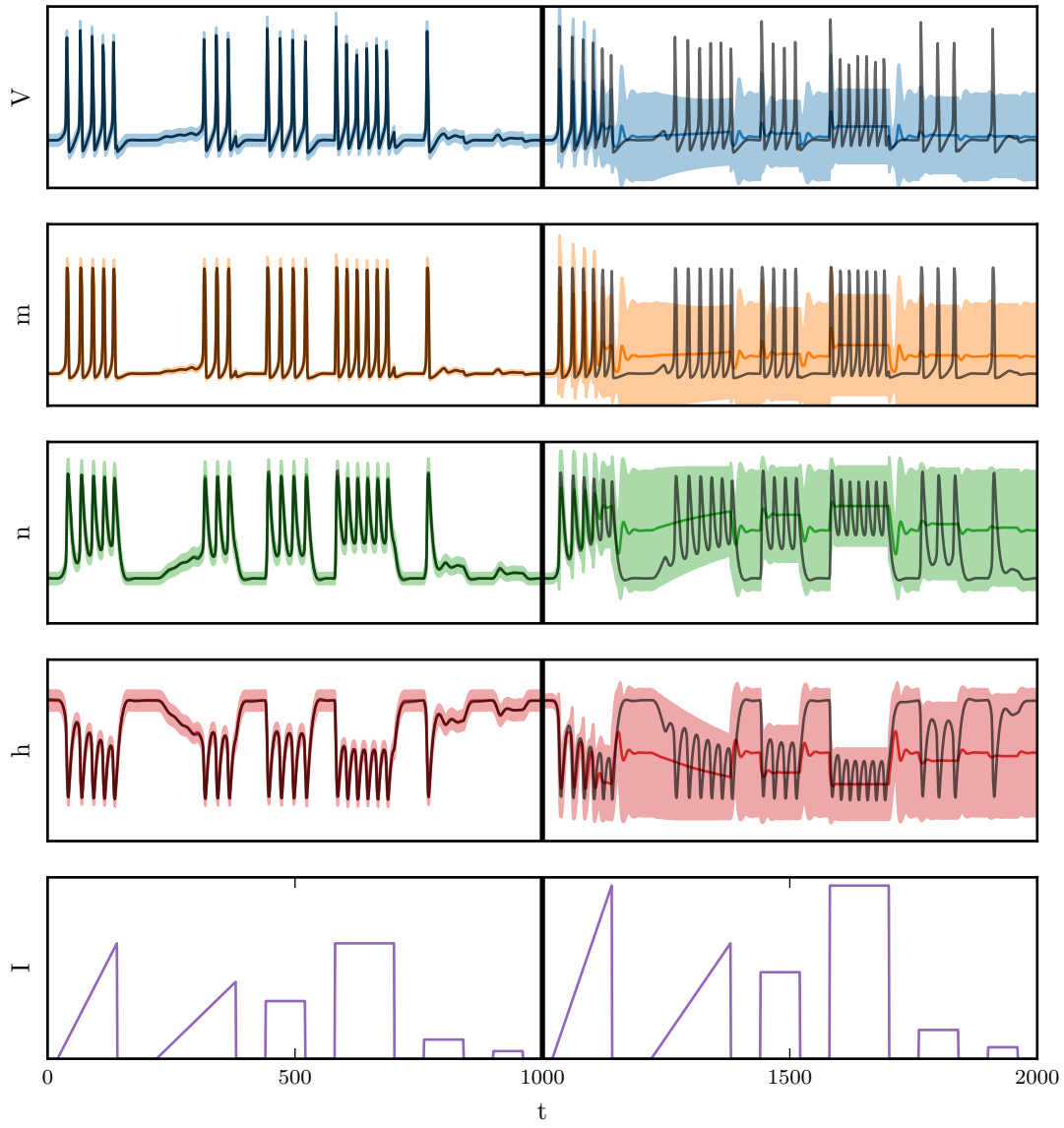


Fig. 3.12 Look-ahead forecasting of the action potentials and gating channels using the Gaussian projection propagation method. The left halves of the plots show the marginal posterior over the training points (the solid line is the mean and the shaded area shows four standard deviations). The right halves of the plots show the predictive marginals at future time steps (the solid line is the mean and the shaded area shows two standard deviations). The black traces show the ground truth simulated data. Best viewed in colour.

suggest the following approximate posterior that can circumvent many of these difficulties,

$$
\begin{aligned}
q(\mathbf{x}_{0:T}, f) &= p(f)p(\mathbf{x}_0) \prod_{t=1}^{T} \left[p(\mathbf{x}_t|\mathbf{x}_{t-1}, f) h_t(\mathbf{u})\right] \\
&= p(f_{\neq \mathbf{u}}|\mathbf{u}) \underbrace{p(\mathbf{u}) \left[\prod_{t=1}^{T} h_t(\mathbf{u})\right]}_{q(\mathbf{u})} \underbrace{\left[p(\mathbf{x}_0) \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}, f)\right]}_{p(\mathbf{x}_{0:T}|f)}.
\end{aligned}
$$

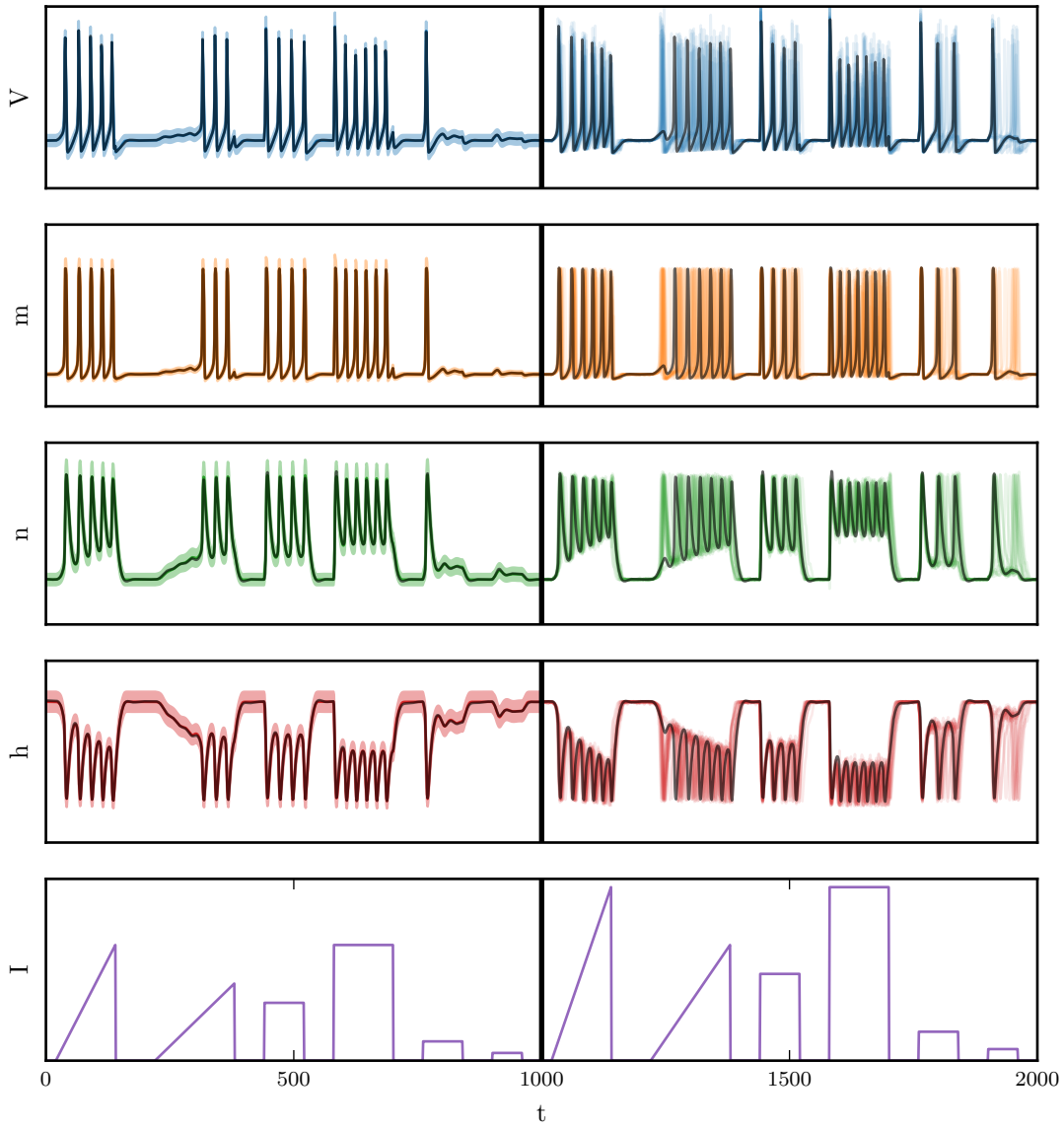


Fig. 3.13 Look-ahead forecasting of the action potentials and gating channels using the simple Monte Carlo propagation method. The left halves of the plots show the marginal posterior over the training points (the solid line is the mean and the shaded area shows four standard deviations). The right halves of the plots show 20 trajectories given by the Monte Carlo propagation. The black traces show the ground truth simulated data. Best viewed in colour.

Note that the exact joint density is,

$$p(\mathbf{x}_{0:T}, f, \mathbf{y}_{1:T}) = p(f)p(\mathbf{x}_0)\prod_{t=1}^{T}\left[p(\mathbf{x}_t|\mathbf{x}_{t-1}, f)p(\mathbf{y}_t|\mathbf{x}_t)\right],$$

which means the approximate posterior retains many terms in the original joint density and only introduces a variational approximation for the global variable (the GP mapping). Notably, unlike the approximations discussed in the previous sections, this approximation explicitly retains the correlations between the hidden variables and the global GP mapping, and between the hidden variables [when the global variable is integrated out, the hidden variables become correlated]. This approximate posterior can be employed in many deterministic schemes previously discussed including variational inference and Power EP. For example, we can write down the variational free-energy using the suggested approximate posterior as follows,

$$\begin{aligned}
\mathcal{F}_{\text{vfe}}(.) &= \int q(\mathbf{x}_{0:T}, f) \log \frac{p(\mathbf{x}_{0:T}, f, \mathbf{y}_{1:T})}{q(\mathbf{x}_{0:T}, f)} \mathrm{d}\mathbf{x}_{0:T} \mathrm{d}f \\
&= \int q(\mathbf{x}_{0:T}, f) \log \frac{\cancel{p(f_{\neq \mathbf{u}}|\mathbf{u})} p(\mathbf{u}) \cancel{\left[p(\mathbf{x}_0) \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}, f) \right]} \left[\prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_t) \right]}{\cancel{p(f_{\neq \mathbf{u}}|\mathbf{u})} q(\mathbf{u}) \cancel{\left[p(\mathbf{x}_0) \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}, f) \right]}} \mathrm{d}\mathbf{x}_{0:T} \mathrm{d}f \\
&= -\mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})] + \sum_{t=1}^{T} \int \mathrm{d}\mathbf{x}_t q(\mathbf{x}_t) \log p(\mathbf{y}_t|\mathbf{x}_t),
\end{aligned}$$

where $q(\mathbf{x}_t) = \int \prod_{i=1}^{t} p(\mathbf{x}_i|\mathbf{x}_{i-1}, f) p(f_{\neq \mathbf{u}}|\mathbf{u}) q(\mathbf{u}) \mathrm{d}\mathbf{x}_{0:t-1} \mathrm{d}f$, which can be obtained using a forward pass with nested simple Monte Carlo or Gaussian projections. Note that only a forward pass is needed to compute the free-energy due to the Markovian structure of the approximate posterior. However, to compute the gradients of the energy w.r.t. the parameters, a backward pass (back-propagation) is necessary. The final form of the free-energy seems much simpler compared to eq. (3.14), but two problems remain to be addressed: i. whether the forward pass can be implemented efficiently in practice, and ii. the efficacy of this approach on real-world state-space modelling tasks.

3.10.2 Active learning for data-efficient system identification

In this section, we suggest a sequential decision making task that makes use of the predictive uncertainty provided by the proposed approximate inference framework. In particular, we consider an active learning (optimal experimental design) task for non-linear system identification, that is, to sequentially select control variables such that the non-linear system can be identified in a data-efficient manner. Assume that a GPSSM was trained using a training time-series of length T, comprising of control inputs $\mathbf{c}_{0:T} \coloneqq \mathbf{c}$ and observed data $\mathbf{y}_{0:T} \coloneqq \mathbf{y}$. The active learning task seeks to find a new control signal $\hat{\mathbf{c}}_{0:\hat{T}} \coloneqq \hat{\mathbf{c}}$ such that the new data $\hat{\mathbf{y}}_{0:\hat{T}} \coloneqq \hat{\mathbf{y}}$ generated from the system using this control signal, together with the existing data can be used to update the trained GPSSM and result in a more accurate and more certain estimate of the non-linear dynamics. MacKay (1992); Houlsby et al. (2011) suggested expressing the aforementioned desideratum using the following information-theoretic

objective:

$$\mathcal{J}(\hat{\mathbf{c}}) = \mathcal{H}(p(f|\mathbf{c},\mathbf{y})) - \mathbb{E}_{p(\hat{\mathbf{y}}|\hat{\mathbf{c}},\mathbf{c},\mathbf{y})}[\mathcal{H}(p(f|\hat{\mathbf{c}},\hat{\mathbf{y}},\mathbf{c},\mathbf{y}))] \tag{3.96}$$

$$= \mathcal{I}(f,\hat{\mathbf{y}}|\hat{\mathbf{c}},\mathbf{c},\mathbf{y}) \tag{3.97}$$

$$= \mathcal{H}(p(\hat{\mathbf{y}}|\hat{\mathbf{c}},\mathbf{c},\mathbf{y})) - \mathbb{E}_{p(f|\mathbf{c},\mathbf{y})}[\mathcal{H}(p(\hat{\mathbf{y}}|\hat{\mathbf{c}},f))]. \tag{3.98}$$

Equation (3.98) is more advantageous compared to eq. (3.96), as it only involves the predictive densities whilst eq. (3.96) needs to access the posterior conditioned on an unknown control signal. However, both terms in eq. (3.98) are not analytically tractable, as the hidden variables for the new time series and the dynamics need to be integrated out, for example the first term requires,

$$p(\hat{\mathbf{y}}|\hat{\mathbf{c}},\mathbf{c},\mathbf{y}) = \int p(\hat{\mathbf{y}}|\hat{\mathbf{x}},f,\hat{\mathbf{c}})p(f|\mathbf{c},\mathbf{y})p(\hat{\mathbf{x}})\mathrm{d}\hat{\mathbf{x}}\mathrm{d}f \tag{3.99}$$

$$= \int \prod_{t=0}^{\hat{T}} [p(\hat{\mathbf{y}}_t|\hat{\mathbf{x}}_t)p(\hat{\mathbf{x}}_t|\hat{\mathbf{x}}_{t-1},f,\hat{\mathbf{c}}_t)]p(f|\mathbf{c},\mathbf{y})\mathrm{d}\hat{\mathbf{x}}\mathrm{d}f. \tag{3.100}$$

Similar to the approach suggested in Depeweg et al. (2016b), we can employ a simple Monte Carlo approach to evaluate the above integral,

$$p(\hat{\mathbf{y}}|\hat{\mathbf{c}},\mathbf{c},\mathbf{y}) \approx \frac{1}{K}\sum_{k=1}^{K} \int \prod_{t=0}^{\hat{T}} [p(\hat{\mathbf{y}}_t|\hat{\mathbf{x}}_t)p(\hat{\mathbf{x}}_t|\hat{\mathbf{x}}_{t-1},f_k,\hat{\mathbf{c}}_t)]\mathrm{d}\hat{\mathbf{x}}, \tag{3.101}$$

and to evaluate the second term of eq. (3.98),

$$\mathbb{E}_{p(f|\mathbf{c},\mathbf{y})}[\mathcal{H}(p(\hat{\mathbf{y}}|\hat{\mathbf{c}},f))] \approx \frac{1}{K}\sum_{k=1}^{K} \mathcal{H}(p(\hat{\mathbf{y}}|\hat{\mathbf{c}},f_k), \tag{3.102}$$

where $\{f_k\}_{k=1}^K$ are K functions drawn from the posterior $p(f|\mathbf{c},\mathbf{y})$. Additional approximations are needed to compute the entropies of the above Monte Carlo estimates, e.g. nonparametric nearest-neighbour based method. Early experimental results suggest this approach is tractable, however the performance heavily depends on the quality of the entropy estimator.

Chapter 4

Sparse Approximations for Deep Gaussian Processes

4.1 Introduction

Pseudo-point based sparse approximations for Gaussian processes (GPs), as reviewed and developed in chapters 2 and 3, have enabled practical and tractable inference and learning in a zoo of applications, ranging from regression and classification to latent variable modelling. However, many key challenges remain unaddressed in the existing literature. First, modelling real-world complex datasets often requires rich and hand-designed covariance functions. It is potentially challenging and time-consuming to design such functions without having prior knowledge about the underlying properties and structures of the data. Second, the functional mapping from inputs to outputs specified by a GP is Gaussian by nature, i.e. the distributions of the function values at any set of inputs is a Gaussian distribution. As a consequence, GPs are arguably unsuited to many use cases in which the empirical distributions of the function values are non-Gaussian, such as data from step-like functions, audio signals or finance time-series. Third, sparse approximations can be damaging for applications in which calibrated predictions are critical. As the computational complexity often scales quadratically in the number of pseudo-points M, increasing M to get a better approximation can quickly render an intractable computational cost in practice. For these reasons, searching for a richer probabilistic model to address these challenges and developing efficient and accurate inference methods for such model are an active area of research.

To this end, we study a multi-layer hierarchical generalisation of GPs or Deep Gaussian processes (DGPs) (Lawrence and Moore, 2007; Damianou and Lawrence, 2013), and investigate whether this class of model can tackle the aforementioned limitations of the shallow (sparse) Gaussian process models. It is well-established that a GP is equivalent to an infinitely wide neural network with single hidden layer (Neal, 1995), and similarly, it can be shown that a DGP is a multi-layer neural network with multiple infinitely wide hidden layers interleaved

with finite width hidden layers. The mapping between layers in this type of network is parameterised by a GP, and, as a result, DGPs retain useful theoretical properties of GPs such as nonparametric modelling power and well-calibrated predictive uncertainty estimates. In addition, DGPs employ a hierarchical structure of GP mappings and therefore are arguably more flexible, have a greater capacity to generalise, and are potentially able to provide better predictive performance (Damianou, 2015). This family of models is attractive as it can also potentially discover layers of increasingly abstract data representations, in much the same way as their deep parametric counterparts — deep neural networks, but it can also handle and propagate uncertainty in the hierarchy.

The addition of non-linear hidden layers can also potentially overcome the practical limitations of the *shallow* GP models mentioned above. First, DGPs can perform input warping or dimensionality compression or expansion, and automatically learn to construct a kernel that works well for the data at hand. The last GP layer in such network could be thought of as a nonparametric output warping layer, that can scale or squash the outputs to match the values of the observations. This is theoretically more flexible compared to alternative hand-chosen parametric warping functions (Snelson et al., 2004). Alternatively, you can view the initial layers as carrying out input warping before the final GP is applied. As a result, learning in this model provides a flexible form of Bayesian kernel design. Second, the functional mapping from inputs to outputs specified by a DGP is non-Gaussian which is a more general and flexible modelling choice. Third, DGPs can repair the damage done by sparse approximations to the representational power of each GP layer. For example, pseudo datapoint based approximation methods for DGP typically trade model complexity for a lower computational complexity of $\mathcal{O}(NLM^2)$ where N is the number of datapoints, L is the number of layers, and M is the number of pseudo datapoints. This complexity scales quadratically in M whereas the dependence on the number of layers L is only linear. Therefore, it can be cheaper to increase the representation power of the model by adding extra layers rather than by adding more pseudo datapoints.

The focus of this chapter is approximate Bayesian learning of DGPs, which involves inferring the posterior over the layer mappings and the hidden variables, and hyperparameter optimisation via the marginal likelihood. In particular, this chapter unifies previously disconnected literature for inference and learning based on variational inference (Damianou and Lawrence, 2013; Damianou, 2015; Hensman and Lawrence, 2014; Salimbeni and Deisenroth, 2017) and approximate expectation propagation (Bui et al., 2016), viewing them as performing (approximate) Power EP using the same variational distribution or a common approximate factor graph representation. This unifying perspective relies on the new approximate posterior inference view of sparse approximations developed in chapter 2, and, similar to chapters 2 and 3, allows a spectrum of new practical approximations to be applied to DGPs. The chapter is organised as follows: the DGP model and its related models are summarised in section 4.2; existing approximations are reviewed and greatly extended under three perspectives: one with

mean-field and parameterised variational distributions for the hidden variables (section 4.3) and one with structured and explicit variational distributions for these variables (section 4.4); and this is followed by some experiments on regression, classification and latent variable modelling in section 4.7.

4.2 Deep Gaussian processes

We first describe DGPs, its relationship with several models discussed in chapters 2 and 3, and briefly discuss existing literature on approximate inference and learning for DGPs. Suppose we have a training set comprising of N D-dimensional input and observation pairs $(\mathbf{x}_n, y_n)$. The probabilistic representation of a DGP comprising L layers can be written as follows,

$$p(f_l|\Theta_l) = \mathcal{GP}(f_l; \mathbf{0}, \mathbf{K}_l), \quad l = 1, \cdots, L \tag{4.1}$$

$$p(\mathbf{h}_l|f_l, \mathbf{h}_{l-1}, \sigma_l^2) = \prod_n \mathcal{N}(h_{l,n}; f_l(h_{l-1,n}), \sigma_l^2), \quad h_{1,n} = \mathbf{x}_n \tag{4.2}$$

$$p(\mathbf{y}|f_L, \mathbf{h}_{L-1}, \sigma_L^2) = \prod_n \mathcal{N}(y_n; f_L(h_{L-1,n}), \sigma_L^2) \tag{4.3}$$

where $h_{l,n}$ is the hidden variable which is the output of the l-th GP layer corresponding to the n-th datapoint, $\mathbf{h}_l \coloneqq h_{l,1:N}$, and f_l if the functions in the l-th layer. For ease of presentation, the outputs are assumed to be real-valued scalars and the observation likelihood is Gaussian, but pointwise likelihoods of the form $p(\mathbf{y}|f_L, \mathbf{h}_{L-1}) = \prod_n p(y_n|f_L(\mathbf{h}_{L-1,n}))$ are easy to accommodate in the approximate inference schemes described in this chapter. Additionally, the hidden variables in the intermediate layers are assumed to single dimensional, but they can and will generally have multiple dimensions.

More formally, we place a zero mean GP prior over the mapping f_l, that is, given the inputs to f_l any finite set of function values are distributed under the prior according to a multivariate Gaussian $p(\mathbf{f}_l) = \mathcal{N}(\mathbf{f}; \mathbf{0}, \mathbf{K_{ff}})$. Note that these function values and consequently the hidden variables are not marginally normally distributed, as the inputs are random variables. When $L = 1$, the model described above collapses back to GP regression. When the inputs $\{\mathbf{x}_n\}$ are unknown and random, the model becomes a DGP latent variable model, which has been studied by Lawrence and Moore (2007); Damianou and Lawrence (2013); Damianou (2015). Pictorially, a DGP with two hidden layers and three GP mapping layers ($L = 3$) is shown in fig. 4.2. It is worth noticing the similarity between this model class and the GPSSMs presented in chapter 3, c.f. fig. 3.1, which in turn leads to the similarity between the approximation techniques for DGPs discussed later in this chapter and those for GPSSMs in chapter 3.

An example of how the function mappings in a DGP might look when $L = 2$ and $\dim(h_{1,n}) = 2$ is shown in Figure 4.1. We train a network using an approximation introduced by Bui et al. (2016) to fit a value function of the mountain car problem (Sutton and Barto,

1998) from a small number of noisy evaluations. The mountain car is often used as a testbed for reinforcement learning agents, in which the agents must pick an action based on the current location and velocity states of the car to bring it up to the top of a mountain. The value function is particularly difficult for models such as GP regression with a standard exponentiated quadratic kernel due to a *steep value function cliff*, but is reasonably handled by a DGP with only two GP layers, as shown in 4.1. Interestingly the functions in the first layer are fairly simple and learn to explain different parts of the input space.

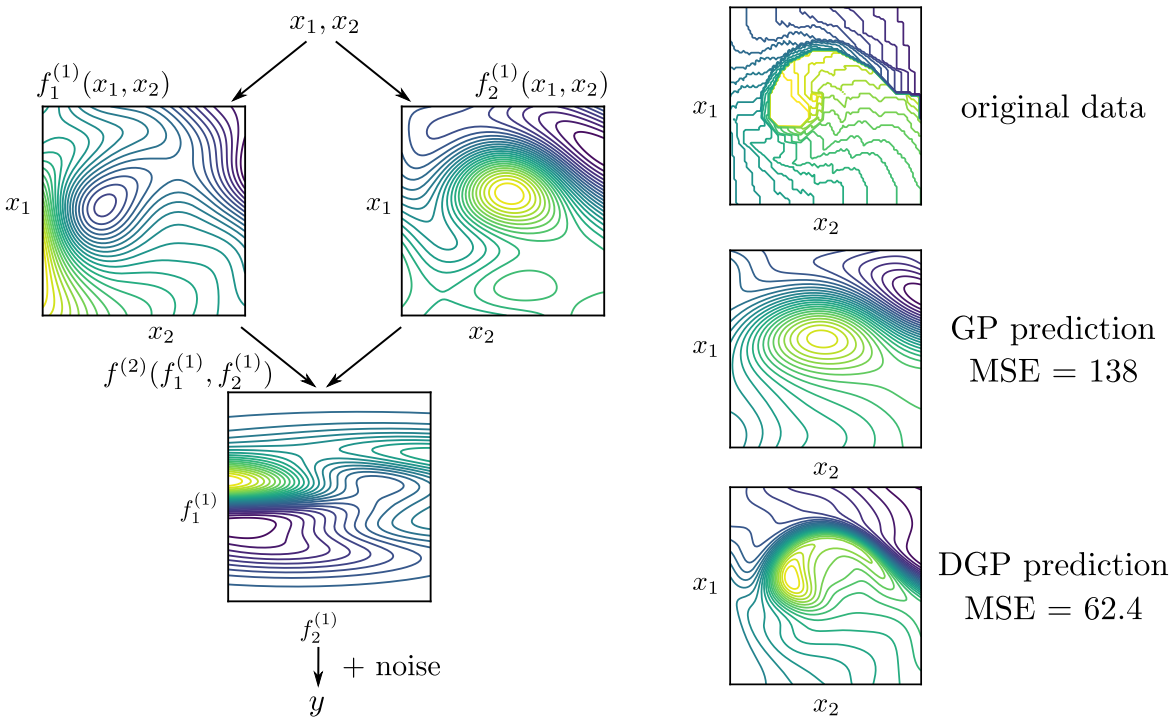


Fig. 4.1 A deep GP example that has two GP mapping layers and one 2-D hidden layer. The training output is the state value of the mountain car problem, and the training inputs are the location and velocity of the car. For illustration purpose, we do not perform on-line learning here, instead, we collect the function values and the states, and use them as inputs and outputs to a standard regression task. The left graphs show latent functions in each layer, two functions in the first layer and one in the second layer, learnt by using the proposed approach. The right graph shows the training data [top] and the predictions of the overall function mapping from inputs to outputs made by a GP [middle] and the DGP on the left [bottom].

We are interested in inferring the posterior distribution over the latent function mappings and the intermediate hidden variables, as well as obtaining a marginal likelihood estimate for hyperparameter tuning and model comparison. For simplicity but without the lack of generality, we take the DGP with two hidden layers described in fig. 4.2 as a running example

for this chapter. In detail, the joint density of all variables for this DGP is as follows,

$$p(\mathbf{y}, f_1, f_2, f_3, \mathbf{h}_1, \mathbf{h}_2|\mathbf{X}, \theta) = p(f_1)p(f_2)p(f_3)\prod_{n=1}^{N}\left[p(h_{1,n}|f_1, \mathbf{x}_1)p(h_{2,n}|f_2, h_{1,n})p(h_{3,n}|f_3, h_{2,n})\right],$$

where $\mathbf{h}_l = \{h_{l,n}\}_{n=1}^N$, $\mathbf{y} = \{y_n\}_{n=1}^N$ and $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$. The marginal likelihood of the model hyperparameters and the exact posterior of the unobserved variables are as follows,

$$p(\mathbf{y}|\mathbf{X}, \theta) = \int \mathrm{d}f_1 \mathrm{d}f_2 \mathrm{d}f_3 \mathrm{d}\mathbf{h}_1 \mathrm{d}\mathbf{h}_2 p(\mathbf{y}, f_1, f_2, f_3, \mathbf{h}_1, \mathbf{h}_2|\mathbf{X}) \tag{4.4}$$

$$p(f_1, f_2, f_3, \mathbf{h}_1, \mathbf{h}_2|\mathbf{y}, \mathbf{X}, \theta) = \frac{p(\mathbf{y}, f_1, f_2, f_3, \mathbf{h}_1, \mathbf{h}_2|\theta, \mathbf{X})}{p(\mathbf{y}|\mathbf{X}, \theta)}. \tag{4.5}$$

However, these quantities are analytically intractable. This is due to the non-linearity in the hierarchy, in particular, the non-linearity of each GP mapping and the stochasticity or randomness on the input of this mapping introduced by previous layers. As such, approximate inference is required. We briefly summarise several existing deterministic approximations here, and how they are currently presented and understood in the literature, before presenting a framework in the next sections pointing out a close connection between some of these methods.

The simplest approach is to obtain the *maximum a posteriori* estimate of the hidden variables (Lawrence and Moore, 2007). However, this procedure is prone to over-fitting and does not provide uncertainty estimates. Alternatively, a plethora of existing work, mostly based on the variational formulation and the seminal pseudo-point sparse approximation of Titsias (2009), can be used. Damianou and Lawrence (2013) introduced a variational approximation over both latent functions and hidden variables, which is chosen such that the variational free energy is both computationally and analytically tractable. Critically, as a variational distribution over the hidden variables is used in this approach, in addition to one over the pseudo-points, the number of variational parameters increases linearly with the number of training datapoints which hinders the use of this method for large-scale datasets. Furthermore, initialisation for this scheme is a known issue, even for a modest number of datapoints. An extension of this approach that has skip links from the inputs to every hidden layer in the network was proposed by Dai et al. (2016), based on suggestions provided in Duvenaud et al. (2014). To combat the large number of variational parameters required in Damianou and Lawrence (2013), Hensman and Lawrence (2014) introduce a *nested* variational scheme that only requires a variational distribution over the pseudo-outputs. Similarly, a recent work by Salimbeni and Deisenroth (2017) attempt to sidestep the parameter scaling problem of Damianou and Lawrence (2013) by a *doubly stochastic* variational approach. However, both approaches of Hensman and Lawrence (2014) and Salimbeni and Deisenroth (2017) are difficult to understand, as the forms of the variational approximations were not explicitly written in the original presentations. Two notable exceptions that do

not use variational inference are methods based (approximate) Power EP and the FITC approximation of Snelson and Ghahramani (2006), proposed by Bui et al. (2015, 2016). However, as pointed out in chapter 2, the presentation of these techniques based on the FITC method is philosophically troubling as FITC approximates each original GP layer by a parametric layer with finite capacity rather than being an approximation that is made at inference time. The complexity of the methods mentioned above is typically $\mathcal{O}(NLM^2)$ for computation, and $\mathcal{O}(NL + LM^2)$ or $\mathcal{O}(LM^2)$ for memory, for methods that use and do not use a parameterised variational distributions over the hidden variables, respectively.

A special case of DGPs when $L = 2$ and the sole hidden layer h_1 is only one-dimensional is warped GPs (Snelson et al., 2004; Lázaro-Gredilla, 2012). Lázaro-Gredilla (2012) proposed a variational approach, in a similar spirit to Titsias (2009) and Damianou and Lawrence (2013) to learn the latent functions. In contrast, the latent function in the second layer is assumed to be deterministic and parameterised by a small set of parameters by (Snelson et al., 2004), which can be learnt by maximising the analytically tractable marginal likelihood. However, the performance of warped GPs is often similar to a standard GP, most likely due to the narrow bottleneck that results from using a one-dimensional hidden layer.

Having summarised several existing approaches to inference and learning in DGPs, we will provide a review of Damianou and Lawrence (2013) and an extension using Power EP, an alternative presentation for the variational approaches proposed by Hensman and Lawrence (2014) and Salimbeni and Deisenroth (2017), a clearer exposition of the approaches proposed by Bui et al. (2015, 2016) based on (approximate) Power EP, and a unifying perspective relating these methods. We attempt to visualise their relationship and link to sections describing each method in table 4.1.

4.3 Approximate inference with parameterised approximations for hidden variables

The factor graph representation of the running regression example, shown in fig. 4.2(A), bears a resemblance to the factor graph of a GPSSM in fig. 3.1(A). Therefore, much of the approximation techniques for GPSSMs described in chapter 3 can be transfered and adapted to DGPs. In this section, we will discuss two structured approximations that uses parameterised Markovian and mean-field Gaussian variational distributions for the hidden variables, similar to that discussed in chapter 3. In particular, the original factor graph of the DGP is approximated by a structured factor graph, in which each original difficult factor involving multiple is broken up into a product of factors, each of which only touches one

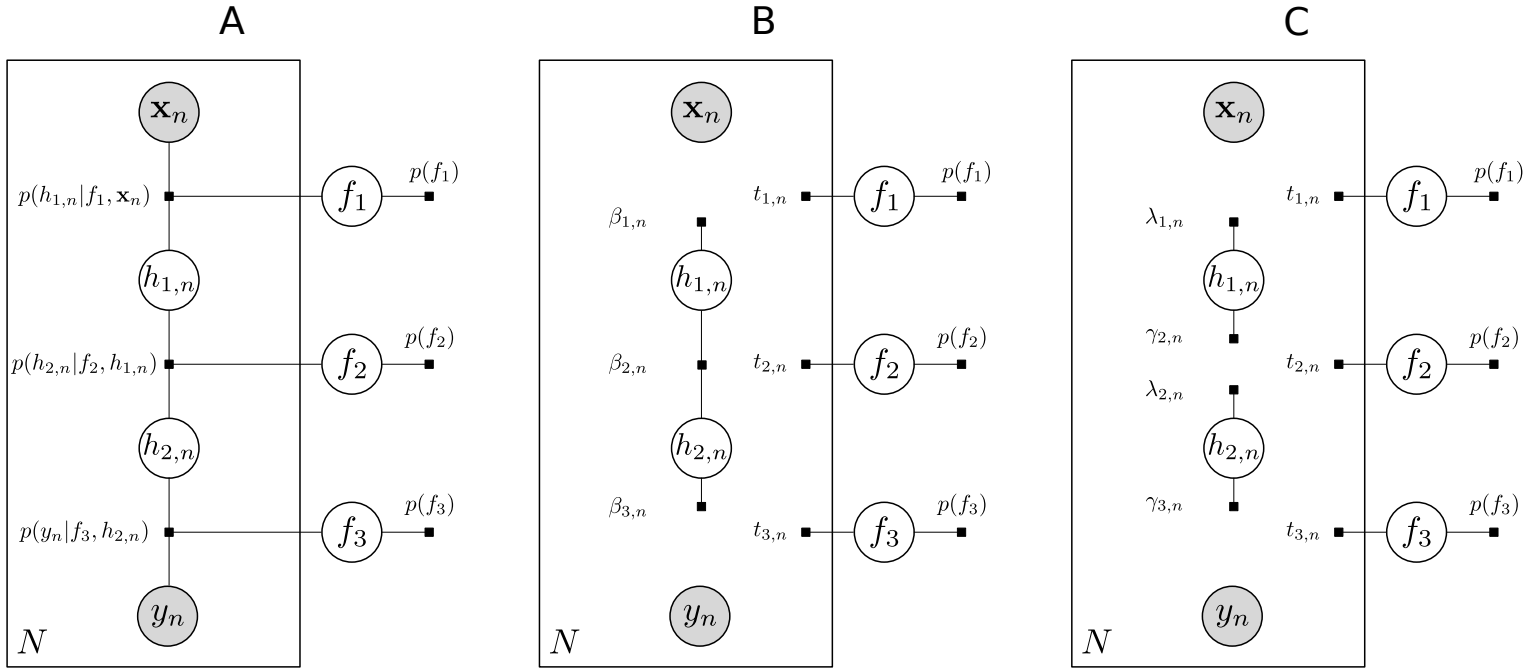


Fig. 4.2 Factor graphs showing a DGP with two hidden layers and three GP mapping layers when the inputs are observed (A), and an approximation to the original factor graph that has factors linking neighbouring hidden variables (B), and an approximation to the original factor graph that assumes a mean-field factorisation between the hidden variables [C].

Table 4.1 Different approximations discussed in this chapter, categorised by the approximate posterior and the inference method used. In all cases, $q(\mathbf{u}_l)$ is assumed Gaussian, $q(\mathbf{u}_l) = \mathcal{N}(\mathbf{u}; \mathbf{m}_l, \mathbf{S}_l)$.

Sec.	$q(\mathbf{h}_n)$	Notes and references
4.3	Gaussian, diagonal covariance	see fig. 4.2(C) VFE: $q(\mathbf{u})$ collapsable, Damianou and Lawrence (2013) PEP with $\alpha \to 0$ gives VFE of Damianou and Lawrence (2013)
4.3	Gaussian, tri-block-diagonal precision	see fig. 4.2(B) VFE: $q(\mathbf{u})$ collapsable, inspired by McHutchon (2014) Power EP with $\alpha \to 0$ gives VFE
4.4	$q(\mathbf{h}_n \vert f_{1:L}) = \prod_{l=1}^{L-1} p(h_{l,n} \vert f_l, h_{l-1,n})$	see section 4.4.1 for VFE and section 4.4.2 for Power EP Salimbeni and Deisenroth (2017) used VFE and simple MC Bui et al. (2015, 2016) used PEP with $\alpha = 1$ and moment maching PEP with $\alpha \to 0$ gives VFE
4.5	explicit $q(h_{l,n} \vert \mathbf{u}_l, h_{l-1,n})$	VFE, Hensman and Lawrence (2014) unclear how to use this for Power EP

variable or two neighbouring variables as follows,

$$p(\mathbf{y}, f_1, f_2, f_3, \mathbf{h}_1, \mathbf{h}_2|\mathbf{X}, \theta) = p(f_1)p(f_2)p(f_3) \prod_{n=1}^{N} \Big[p(h_{1,n}|f_1, \mathbf{x}_n) \; p(h_{2,n}|f_2, h_{1,n}) \; p(h_{3,n}|f_3, h_{2,n}) \Big], \tag{4.6}$$

$$q_{\text{Markovian}}(\mathbf{y}, f_1, f_2, f_3, \mathbf{h}_1, \mathbf{h}_2|\mathbf{X}, \theta) \propto p(f_1)p(f_2)p(f_3) \prod_{n=1}^{N} \Big[\beta_{1,n}(h_{1,n}) t_{1,n}(f_1) \; \beta_{2,n}(h_{1,n}, h_{2,n}) t_{2,n}(f_2) \; \beta_{3,n}(h_{2,n}) t_{3,n}(f_3) \Big], \tag{4.7}$$

$$q_{\text{mean-field}}(\mathbf{y}, f_1, f_2, f_3, \mathbf{h}_1, \mathbf{h}_2|\mathbf{X}, \theta) \propto p(f_1)p(f_2)p(f_3) \prod_{n=1}^{N} \Big[\beta_{1,n}(h_{1,n}) t_{1,n}(f_1) \; \gamma_{2,n}(h_{1,n}) \beta_{2,n}(h_{2,n}) t_{2,n}(f_2) \; \gamma_{3,n}(h_{2,n}) t_{3,n}(f_3) \Big], \tag{4.8}$$

where $p(.)$, $q_{\text{Markovian}}(.)$, and $q_{\text{mean-field}}(.)$ are the exact joint density and approximate joint densities with Markovian and mean-field approximations for the hidden variables, respectively. Note that we also employ pseudo-points approximations for each GP here, that is each t factor is assumed to involve only a small number of function values $\mathbf{u}$ [pseudo-outputs] at some pseudo-inputs, as follows,

$$t_{1,n}(f_1) = t_{1,n}(\mathbf{u}_1), \quad t_{2,n}(f_2) = t_{2,n}(\mathbf{u}_2), \quad t_{3,n}(f_3) = t_{3,n}(\mathbf{u}_3). \tag{4.9}$$

The approximate posteriors with Markovian and mean-field approximations for the hidden variables are pictorially depicted in fig. 4.2(B) and fig. 4.2(C), respectively. These illustrations clearly show the *local* nature of this approximation. However, a *global* form of the approximate posteriors can be formed by grouping the factors that share a common variable or a common

set of variables:

$$q_{\text{mean-field}}(.) \propto p(f_{1,\neq \mathbf{u}_1}|\mathbf{u}_1)p(f_{2,\neq \mathbf{u}_2}|\mathbf{u}_2)p(f_{3,\neq \mathbf{u}_3}|\mathbf{u}_3)\times$$
$$\overbrace{\left[p(\mathbf{u}_1)\prod_{n=1}^{N} t_{1,n}(\mathbf{u}_1)\right]}^{q(\mathbf{u}_1)}\overbrace{\left[p(\mathbf{u}_2)\prod_{n=1}^{N} t_{2,n}(\mathbf{u}_2)\right]}^{q(\mathbf{u}_2)}\overbrace{\left[p(\mathbf{u}_3)\prod_{n=1}^{N} t_{3,n}(\mathbf{u}_3)\right]}^{q(\mathbf{u}_3)}\times$$
$$\prod_{n=1}^{N}\underbrace{[\beta_{1,n}(h_{1,n})\gamma_{2,n}(h_{1,n})]}_{q(h_{1,n})}\prod_{n=1}^{N}\underbrace{[\beta_{2,n}(h_{2,n})\gamma_{3,n}(h_{2,n})]}_{q(h_{2,n})}, \tag{4.10}$$

$$q_{\text{Markovian}}(.) \propto p(f_{1,\neq \mathbf{u}_1}|\mathbf{u}_1)p(f_{2,\neq \mathbf{u}_2}|\mathbf{u}_2)p(f_{3,\neq \mathbf{u}_3}|\mathbf{u}_3)\times$$
$$\overbrace{\left[p(\mathbf{u}_1)\prod_{n=1}^{N} t_{1,n}(\mathbf{u}_1)\right]}^{q(\mathbf{u}_1)}\overbrace{\left[p(\mathbf{u}_2)\prod_{n=1}^{N} t_{2,n}(\mathbf{u}_2)\right]}^{q(\mathbf{u}_2)}\overbrace{\left[p(\mathbf{u}_3)\prod_{n=1}^{N} t_{3,n}(\mathbf{u}_3)\right]}^{q(\mathbf{u}_3)}\times$$
$$\prod_{n=1}^{N}\underbrace{[\beta_{1,n}(h_{1,n})\beta_{2,n}(h_{1,n},h_{2,n})\beta_{3,n}(h_{2,n})]}_{q(h_{1,n},h_{2,n})}, \tag{4.11}$$

It is clear from the equations above that both appoximations impose a structured approximation for the latent GPs, in a similar fashion to that in GP regression, classification and state space models. In addition, both approximations assumes a mean-field structure between the latent functions and the hidden variables. The key difference, however, is that eq. (4.10) posits a mean-field posterior [Gaussian with a diagonal covariance matrix] for the hidden variables, whilst eq. (4.11) explicitly encodes the pairwise interactions between neighbouring hidden variables in a Markovian posterior [Gaussian with a tri-diagonal precision matrix]. These approximate posteriors can be used in many deterministic approximation shemes such as variational inference or power expectation propagation, in a similar fashion to those discussed in chapter 3. In particular, the approximate factors are typically assumed Gaussian, and their means and covariances are parameterised and updated by using either the Power EP iterative procedure or by optimising the variational free-energy. When the variational free-energy approach with the appoximate posterior $q_{\text{mean-field}}(.)$ is used, we arrive at the approach of Damianou and Lawrence (2013). Similar to the GPSSM case, it can be shown that for the variational case, the optimal form for $q(\mathbf{h}) = q(\mathbf{h}_1, \mathbf{h}_2)$ is Markov and non-Gaussian, and the optimal form for $q(\mathbf{u})$ is conveniently a Gaussian that depends on $q(\mathbf{h})$. This leads to an analytic collapsed bound that only depends on variational parameters parameterising $q(\mathbf{h})$. The collapsed bound, however, is not amenable to stochastic optimisation. Alternatively, the iterative Power EP procedure can be used to update the factors, or the approximate Power EP energy can be minimised to obtain the approximate posterior when the factors are deliberately tied. As the algorithmic and computational procedures of these schemes, as well as additional approximations to enable tractability such as Moment Maching or simple

Monte Carlo, have been described at length in chapter 3, we refer interested readers to this chapter for more details.

One disadvantage of this class of approximate posterior is the need to parameterise the variational distribution over the local hidden variables (h_1 and h_2 in the running example). A clear consequence is the memory complexity that scales linearly in the number of datapoints, which is burdensome for large datasets. More importantly, initialisation of these distributions is problematic as numerical instability can arise if their initial parameters and those of the latent GPs are mismatched. Techniques such as inference networks can be useful to reduce the complexity, however, they do not resolve the initialisation issue.

This limitation begs consideration of whether the intermediate hidden variables can be integrated out so they can be avoided in the inference process. It can be shown that the intermediate Gaussian noise can be merged into the kernel of the GP layer, which means the local intermediate variables can be integrated out to arrive at a collapsed model, in which there are only global variables. However, the separation of the local variables and the global variables are key for the Titsias' trick (Titsias, 2009), as discussed in chapter 2, to be applicable. This rules out the above option of marginalising out the hidden variables. Fortunately, alternative variational approximations can be carefully chosen such that no explicitly parameterised forms for these variables are needed. We will discuss several choices of such approximations and how to use them for Power EP and variational inference in the next section.

4.4 Approximate inference with explicit conditional approximations for hidden variables

As described above, an approximation for the latent functions and the intermediate hidden variables can be used within the Power EP procedure or the variational inference scheme, to sidestep the intractabilies in inference and learning. However, a parameterised approximation over the hidden variables is a memory intensive constraint. In addition, a mean-field assumption between the latent functions and the hidden variables is arguably limited, as the hidden variables are the noisy function evaluations in the generation process, resulting in a potentially strong coupling in the posterior. In this section, we relax this mean-field

constraint and assume an explicit conditional distribution for the hidden variables as follows,

$$q(\cdot) \propto p(f_{1,\neq \mathbf{u}_1}|\mathbf{u}_1)p(f_{2,\neq \mathbf{u}_2}|\mathbf{u}_2)p(f_{3,\neq \mathbf{u}_3}|\mathbf{u}_3)\times \\ \overbrace{\left[p(\mathbf{u}_1)\prod_{n=1}^{N} t_{1,n}(\mathbf{u}_1)\right]}^{q(\mathbf{u}_1)}\overbrace{\left[p(\mathbf{u}_2)\prod_{n=1}^{N} t_{2,n}(\mathbf{u}_2)\right]}^{q(\mathbf{u}_2)}\overbrace{\left[p(\mathbf{u}_3)\prod_{n=1}^{N} t_{3,n}(\mathbf{u}_3)\right]}^{q(\mathbf{u}_3)}\times \\ \prod_{n=1}^{N} p(h_{1,n}; f_1, \mathbf{x}_n)\prod_{n=1}^{N} p(h_{2,n}|f_2, h_{1,n}). \tag{4.12}$$

In fact, this approximation retains in all layer the conditional distributions that map from the previous hidden variable through a latent function to the next hidden variable, $\{p(h_{1,n}; f_1, \mathbf{x}_n), p(h_{2,n}|f_2, h_{1,n})\}_{n=1}^{N}$, except for the last layer, in which the conditional distribution is the likelihood model for the data, $\{p(y_n|f_3, h_{2,n})\}_{n=1}^{N}$. This can be seen by comparing the exact joint density in eq. (4.6) with eq. (4.12). At this point, interested readers might find this approximate posterior rather strange, since there are no explicit approximations for $h_{1,n}$ and $h_{2,n}$, but instead, the approximation for each hidden variable are completely specified by the approximations for the GP at the current layer and the hidden variable at the previous layer, as $p(h_{1,n}; f_1, \mathbf{x}_n)$ and $p(h_{2,n}|f_2, h_{1,n})$ are not learnt. It is also not clear how each individual term in the joint density is approximated.

However, in this regression/classification setting, the ultimate task is to obtain good predictions at test time. The quantity required to obtain such a prediction is the approximate posterior over the GP mappings and not the hidden variables of the training points. It is therefore sensible to choose an approximation, like one chosen here, that devotes its approximation effort to the latent non-linear mappings, while providing an approximation for the training hidden variables for free. More importantly, the approximation discussed in this section addresses two major concerns when using the parameterised form in eqs. (4.10) and (4.11),

- the posterior dependencies between the hidden variables, and between the hidden variables and the latent functions are strictly enforced by the explicit conditional densities, as opposed to the mean-field assumptions in eqs. (4.10) and (4.11), and,
- there are no explicit factors or global approximations for the hidden variables that need to be parameterised and updated. This is a huge memory saving for big datasets, and at the same time, mitigates the initialisation issue that arose when using the parameterised Gaussian marginals in eqs. (4.10) and (4.11).

This approximation can now be understood to have been used for variational inference by Salimbeni and Deisenroth (2017), and (approximate) EP by Bui et al. (2015, 2016). Though the connection between these methods is obvious in hindsight given the explicit form of the approximate posterior, it was poorly understood due to the difference in the presentation

styles in the original publications and the fact that FITC was not understood as approximate inference. We will now detail this connection and show that these schemes are special cases of a more general Power EP framework that uses the same posterior approximation.

4.4.1 The VFE approach of Salimbeni and Deisenroth (2017)

In this section, we take the global view of the approximate posterior in eq. (4.12),

$$q(f_1, f_2, f_3, \mathbf{h}_1, \mathbf{h}_2) = p(f_{1,\neq \mathbf{u}_1}|\mathbf{u}_1)p(f_{2,\neq \mathbf{u}_2}|\mathbf{u}_2)p(f_{3,\neq \mathbf{u}_3}|\mathbf{u}_3)q(\mathbf{u}_1)q(\mathbf{u}_2)q(\mathbf{u}_3)\times \prod_{n=1}^{N}\Big[p(h_{1,n}; f_1, \mathbf{x}_n)p(h_{2,n}|f_2, h_{1,n})\Big]. \tag{4.13}$$

The joint density of the model in eq. (4.6) can be rewritten as follows,

$$p(\mathbf{y}, f_1, f_2, f_3, \mathbf{h}_1, \mathbf{h}_2|\mathbf{X}, \theta) = p(f_{1,\neq \mathbf{u}_1}|\mathbf{u}_1)p(f_{2,\neq \mathbf{u}_2}|\mathbf{u}_2)p(f_{3,\neq \mathbf{u}_3}|\mathbf{u}_3)p(\mathbf{u}_1)p(\mathbf{u}_2)p(\mathbf{u}_3)\times \prod_{n=1}^{N}\Big[p(h_{1,n}|f_1, \mathbf{x}_n)p(h_{2,n}|f_2, h_{1,n})p(y_n|f_3, h_{2,n})\Big]. \tag{4.14}$$

Following the standard variational Bayesian approach, we can arrive at the following negative variational free-energy, which is a lower bound of the log marginal likelihood,

$$\mathcal{F}_{\text{vfe}}(\cdot) = \int_{f_1,f_2,f_3,\mathbf{h}_1,\mathbf{h}_2} q(f_1, f_2, f_3, \mathbf{h}_1, \mathbf{h}_2) \log \frac{p(\mathbf{y}, f_1, f_2, f_3, \mathbf{h}_1, \mathbf{h}_2|\mathbf{X}, \theta)}{q(f_1, f_2, f_3, \mathbf{h}_1, \mathbf{h}_2)}. \tag{4.15}$$

The gap between the negative free-energy and the log marginal likelihood is the KL divergence from the exact posterior to the variational approximation,

$$\mathcal{F}_{\text{vfe}}(\cdot) = \log p(\mathbf{y}|\mathbf{X}, \theta) - \text{KL}(p(f_1, f_2, f_3, \mathbf{h}_1, \mathbf{h}_2|\mathbf{X}, \mathbf{y}, \theta)||q(f_1, f_2, f_3, \mathbf{h}_1, \mathbf{h}_2)), \tag{4.16}$$

which means the negative free-energy can be maximised w.r.t. the approximate posterior and the model hyperparameters, with a guarantee that the approximate posterior will get closer to the exact posterior as measured by the KL divergence. Importantly, the form of the approximation chosen above admits a tractable free energy as follows,

$$\mathcal{F}_{\text{vfe}}(\cdot) = -\sum_{l=1}^{3} \text{KL}(q(\mathbf{u}_l)||p(\mathbf{u}_l)) + \sum_{n=1}^{N} \underbrace{\langle \log p(y_n|f_3, h_{2,n})\rangle_{q(f_1,f_2,f_3,h_{1,n},h_{2,n})}}_{\mathcal{F}_{\text{vfe},2,n}}. \tag{4.17}$$

Note that the greatly simplified form above is the result of substituting the approximate posterior in eq. (4.13) and the joint density in eq. (4.14) to the negative free-energy in eq. (4.15), and cancelling out identical terms in the log. As in previous chapters and following what is typically done in the literature, we posit a Gaussian form for $q(\mathbf{u}_l)$, $q(\mathbf{u}_l) = \mathcal{N}(\mathbf{u}_l; \mathbf{m}_l, \mathbf{S}_l)$.

This allows analytic computation of the KL terms in eq. (4.17) since $p(\mathbf{u}_l)$ is also a Gaussian density. Notice further that the second term in eq. (4.17) decomposes as a sum of independent terms, one for each training instance, and that the data likelihood in the output layer only touches the last hidden variable $h_{2,n}$ and the last function mapping f_3, we can rewrite this term as,

$$\mathcal{F}_{\text{vfe},2,n} = \left\langle \log p(y_n|f_3, h_{2,n}) \right\rangle_{q(f_3)q(h_{2,n})}, \tag{4.18}$$

where $q(f_3) = p(f_{3,\neq\mathbf{u}_3}|\mathbf{u}_3)q(\mathbf{u}_3)$ and

$$q(h_{2,n}) = \int_{h_{1,n},f_1,f_2} p(h_{2,n}|f_2, h_{1,n})p(f_{2,\neq\mathbf{u}_2}|\mathbf{u}_2)q(\mathbf{u}_2)p(h_{1,n}|f_1, \mathbf{x}_n)p(f_{1,\neq\mathbf{u}_1}|\mathbf{u}_1)q(\mathbf{u}_1). \tag{4.19}$$

The focus is now on how to tractably compute the marginal $q(h_{2,n})$ in eq. (4.19) and the expectation of the log data likelihood in eq. (4.18). Due to the non-linearity in the network $q(h_{2,n})$ is in general non-Gaussian, which in turn results in an analytically intractable expectation in eq. (4.18). However, there are efficient procedures to approximate these quantities. The strategy is to approximate the integral in eq. (4.19) in a sequential order, from the input layer to the second last layer. In detail, notice f_1 can first be integrated out exactly,

$$\begin{aligned} q(h_{1,n}) &= \int_{f_1} p(h_{1,n}|f_1, \mathbf{x}_n)p(f_{1,\neq\mathbf{u}_1}|\mathbf{u}_1)q(\mathbf{u}_1) = \mathcal{N}(h_{1,n}; m_{h_{1,n}}, v_{h_{1,n}}), \qquad (4.20)\\ \text{where} \quad m_{h_{1,n}} &= \mathbf{k}_{n\mathbf{u},1}\mathbf{K}_{\mathbf{uu},\mathbf{1}}^{-1}\mathbf{m}_1,\\ v_{h_{1,n}} &= \mathbf{k}_{nn,1} - \mathbf{k}_{n\mathbf{u},1}\mathbf{K}_{\mathbf{uu},\mathbf{1}}^{-1}\mathbf{k}_{\mathbf{u}n,1} + \mathbf{k}_{n\mathbf{u},1}\mathbf{K}_{\mathbf{uu},\mathbf{1}}^{-1}\mathbf{S}_1\mathbf{K}_{\mathbf{uu},\mathbf{1}}^{-1}\mathbf{k}_{\mathbf{u}n,1} + \sigma_1^2, \end{aligned}$$

and the subscript 1 denotes the quantities in the first layer, e.g. $\mathbf{K}_{\mathbf{uu},\mathbf{1}}$ is the covariance between the first layer's pseudo-points. Note that $q(h_{1,n})$ is simply the predictive distribution at the input $\mathbf{x}_n$, given the posterior distribution of the latent GP grounded on the pseudo-points, and that $\mathbf{k}_{nn,1}$ and $\mathbf{k}_{n\mathbf{u},1}$ are evaluated at the input $\mathbf{x}_n$. Similarly, we can integrate f_2 out analytically for a given deterministic $h_{1,n}$,

$$\begin{aligned} q(h_{2,n}|h_{1,n}) &= \int_{f_2} p(h_{2,n}|f_2, h_{1,n})p(f_{2,\neq\mathbf{u}_2}|\mathbf{u}_2)q(\mathbf{u}_2) = \mathcal{N}(h_{2,n}; m_{h_{2,n}|h_{1,n}}, v_{h_{2,n}|h_{1,n}}),\\ \text{where} \quad m_{h_{2,n}|h_{1,n}} &= \mathbf{k}_{n\mathbf{u},2}\mathbf{K}_{\mathbf{uu},\mathbf{2}}^{-1}\mathbf{m}_2,\\ v_{h_{2,n}|h_{1,n}} &= \mathbf{k}_{nn,2} - \mathbf{k}_{n\mathbf{u},2}\mathbf{K}_{\mathbf{uu},\mathbf{2}}^{-1}\mathbf{k}_{\mathbf{u}n,2} + \mathbf{k}_{n\mathbf{u},2}\mathbf{K}_{\mathbf{uu},\mathbf{2}}^{-1}\mathbf{S}_2\mathbf{K}_{\mathbf{uu},\mathbf{2}}^{-1}\mathbf{k}_{\mathbf{u}n,2} + \sigma_2^2, \end{aligned}$$

and $\mathbf{k}_{nn,2}$ and $\mathbf{k}_{n\mathbf{u},2}$ are evaluated at the deterministic hidden value $h_{1,n}$. The above results simplify eq. (4.19) given both f_1 and f_2 have been exactly marginalised out,

$$q(h_{2,n}) = \int_{h_1} q(h_{2,n}|h_{1,n})q(h_{1,n}). \tag{4.21}$$

It is perhaps unsurprising that the integral above is exactly the central question we asked earlier in chapter 3: how to propagate a Gaussian distribution over the input through a GP posterior. The difficulty originates from the non-linearity of the GP predictive distribution w.r.t. the input, and the randomness in the input. As a result, $q(h_{2,n})$ is a non-trivial mixture of an infinite number of Gaussian densities, which can have heavy-tails and multiple modes. At this point, we can reuse a suite of approximation methods discussed in section 3.4.1 to get an approximate $q(h_{2,n})$. We will review two approximations: simple Monte Carlo and moment matching[1].

Nested simple Monte Carlo

The most naïve approach to approximate $q(h_{2,n})$ is by a mixture of a finite number of Gaussian densities,

$$q(h_{2,n}) \approx \frac{1}{R}\sum_{r=1}^{R} q(h_{2,n}|h_{1,n,r}) = \frac{1}{R}\sum_{r=1}^{R} \mathcal{N}(h_{2,n}; m_{h_{2,n}|h_{1,n,r}}, v_{h_{2,n}|h_{1,n,r}}), \tag{4.22}$$

where $\{h_{1,n,r}\}_{r=1}^{R}$ is a set of R independent samples, drawn from $q(h_{1,n})$. While this approximation is unbiased for a finite R and exact as $R \to \infty$, it can be poor when R is small and the dimension of $h_{1,n}$ is large. In the case when there are more than three GP layers, this mixture of Gaussians can be approximately propagated through another GP posterior using the same technique. This means the samples are now drawn from a mixture of uniformly weighted Gaussian distributions (instead of from a single Gaussian), and propagated through the GP mapping to form another mixture of Gaussians.

Given the mixture approximation to $q(h_{2,n})$, the expectation in eq. (4.18) can be approximated by,

$$\mathcal{F}_{\text{vfe},2,n} = \left\langle \log p(y_n|f_3, h_{2,n}) \right\rangle_{q(f_3)q(h_{2,n})} \approx \frac{1}{R}\sum_{r=1}^{R} \left\langle \log p(y_n|f_3, h_{2,n}) \right\rangle_{q(f_3)q(h_{2,n}|h_{1,n,r})}. \tag{4.23}$$

For a Gaussian observation model and for certain choices of the covariance function for the last GP mapping, this is available in closed-form[2]. However, it is analytically intractable for general likelihoods and requires additional approximations, e.g. another simple Monte Carlo integration (Gal et al., 2015). When the simple Monte Carlo method is used at this step, in addition to the Monte Carlo approximation for $q(h_{2,n})$, Salimbeni and Deisenroth (2017) called the overall method *doubly stochastic* due to the two sources of stochasticity in the estimation procedure.

[1] The linearisation approximation discussed in chapter 3 can be employed in this case too, but we suspect this approximation is poor for deep networks, as demonstrated when a nested version of this scheme performed poorly for GPSSM prediction in chapter 3.

[2] The approximate expectation is now a sum of expectations, each of which is the expectation computation required for variational inference in GPLVMs (Titsias and Lawrence, 2010).

One key implementation challenge is how to obtain the gradient of the estimated expectation w.r.t. the variational parameters (such as $\mathbf{m}_l$ and $\mathbf{S}_l$) and in particular, how to perform the backward pass through the Monte Carlo procedures. Fortunately, this can be addressed by employing the *reparameterisation trick* (Kingma and Welling, 2014; Salimans and Knowles, 2013), which is arguably the main factor leading to the recent revival of Monte Carlo based variational inference.

Nested Gaussian projection or moment matching

For certain choices of covariance functions in the network, it is possible to use an efficient and accurate approximation which propagates a Gaussian through the first layer of the network and projects this non-Gaussian distribution back to a moment matched Gaussian before propagating through the next layer and repeating the same steps, hence we call this approximation *nested Gaussian projection*. However, in the running example in this chapter, only one step of this approximation is needed, as illustrated in fig. 3.2 and detailed in chapter 3. We review the key results here.

Although the exact marginal $q(h_{2,n})$ is non-Gaussian and non-analytic, its mean and covariance can be obtained using the law of iterated conditionals (Girard et al., 2003; Deisenroth and Mohamed, 2012) as follows,

$$\begin{aligned} m_{h_{2,n}} &= \mathrm{E}_{q(h_{1,n})}[m_{h_{2,n}|h_{1,n}}] \\ v_{h_{2,n}} &= \mathrm{E}_{q(h_{1,n})}[v_{h_{2,n}|h_{1,n}}] + \mathrm{var}_{q(h_{1,n})}[m_{h_{2,n}|h_{1,n}}] \end{aligned}$$

Substituting the mean and covariance of $q(h_{1,n})$ in eq. (4.20) into the above results gives,

$$m_{h_{2,n}} = \langle \mathbf{A}_{2,n} \rangle_{q(h_{1,n})} \mathbf{m}_2, \tag{4.24}$$

$$v_{h_{2,n}} = \langle \mathbf{B}_{2,n} \rangle_{q(h_{1,n})} + \langle \mathbf{A}_{2,n} [\mathbf{S}_2 + \mathbf{m}_2 \mathbf{m}_2^{\mathsf{T}}] \mathbf{A}_{2,n}^{\mathsf{T}} \rangle_{q(h_{1,n})} + \sigma_2^2 - m_{h_{2,n}}^2. \tag{4.25}$$

where $\mathbf{A}_{2,n} = \mathbf{k}_{n\mathbf{u},2} \mathbf{K}_{\mathbf{uu},\mathbf{2}}^{-1}$ and $\mathbf{B}_{2,n} = \mathbf{k}_{nn,2} - \mathbf{k}_{n\mathbf{u},2} \mathbf{K}_{\mathbf{uu},\mathbf{2}}^{-1} \mathbf{k}_{\mathbf{u}n,2}$. The equations above require the expectations of the kernel matrices under a Gaussian distribution over the inputs, which are analytically tractable for widely used kernels such as exponentiated quadratic, linear or a more general class of spectral mixture kernels (Titsias and Lawrence, 2010; Wilson and Adams, 2013). We approximate $q(h_{2,n})$ by a Gaussian distribution with the same mean and covariance, $q(h_{2,n}) \approx \tilde{q}(h_{2,n}) \mathcal{N}(h_{2,n}; m_{h_{2,n}}, v_{h_{2,n}})$. In addition, this approximation above can be improved for networks that have multidimensional intermediate variables, by using a Gaussian with a non-diagonal covariance matrix (Deisenroth and Mohamed, 2012). This is, however, more computationally expensive so the diagonal approximation will be used here and for the rest of this chapter.

Given the moment-matched Gaussian approximation $\tilde{q}(h_2)$, eq. (4.18) can be approximated by substituting the Gaussian approximation in place of $q(h_2)$,

$$\mathcal{F}_{\text{vfe},2,n} = \left\langle \log p(y_n|f_3, h_{2,n}) \right\rangle_{q(f_3)q(h_{2,n})} \approx \left\langle \log p(y_n|f_3, h_{2,n}) \right\rangle_{q(f_3)\tilde{q}(h_{2,n})}. \tag{4.26}$$

The computation of the resulting approximate expectation is similar to that using the simple Monte Carlo approximation [eq. (4.23)]. This means eq. (4.26) can be computed analytically for certain choices of covariance functions and observation models, or can be approximated by simple Monte Carlo integration.

4.4.2 The Power EP approach

We have discussed the variational free-energy approach using the posterior approximation in eq. (4.12), for inference and learning in DGPs. As done in chapters 2 and 3, we will investigate the use of the same posterior approximation for Power EP, and establish the connection to VFE. A key difference compared to the VFE approach is that the local view of eq. (4.12) is used,

$$q(\cdot) \propto p(f_{1,\neq \mathbf{u}_1}|\mathbf{u}_1)p(f_{2,\neq \mathbf{u}_2}|\mathbf{u}_2)p(f_{3,\neq \mathbf{u}_3}|\mathbf{u}_3)p(\mathbf{u}_1)p(\mathbf{u}_2)p(\mathbf{u}_3)\times \\ \prod_{n=1}^{N}\Big[t_{1,n}(\mathbf{u}_1)t_{2,n}(\mathbf{u}_2)t_{3,n}(\mathbf{u}_3)p(h_{1,n}; f_1, \mathbf{x}_n)p(h_{2,n}|f_2, h_{1,n})\Big], \tag{4.27}$$

where the product $t_{1,n}(\mathbf{u}_1)t_{2,n}(\mathbf{u}_2)t_{3,n}(\mathbf{u}_3)p(h_{1,n}; f_1, \mathbf{x}_n)p(h_{2,n}|f_2, h_{1,n}) \triangleq \hat{g}_n$ could be thought of as an approximation to $p(h_{1,n}; f_1, \mathbf{x}_n)p(h_{2,n}|f_2, h_{1,n})p(y_n; f_3, h_{2,n}) \triangleq g_n$ in eq. (4.6). The standard iterative Power EP procedure can now be employed, which involves looping through the dataset multiple times and performing the following steps: 1. remove a fraction of $\hat{g}_n$ from the approximate posterior to form the cavity distribution, 2. incorporate a fraction of the exact factor g_n the cavity to form the tilted distribution, and moment match the approximate posterior to this distribution, and 3. update the approximate factor $\hat{g}_n$ using the new approximate posterior and the cavity distribution. We choose a Gaussian form for each approximate factor $\hat{g}_n$, and next discuss how to deal with these steps given a training datapoint $(\mathbf{x}_n, y_n)$, for a general α hyperparameter of Power EP.

Deletion step

The cavity distribution is formed by removing a fraction of $\hat{g}_n$ from $q(.)$,

$$q_{\text{cav},n}(\cdot) \propto q(\cdot)/\hat{g}_n^{\alpha} \propto p(f_{1,\neq \mathbf{u}_1}|\mathbf{u}_1)p(f_{2,\neq \mathbf{u}_2}|\mathbf{u}_2)p(f_{3,\neq \mathbf{u}_3}|\mathbf{u}_3)q_{\text{cav},n}(\mathbf{u}_1)q_{\text{cav},n}(\mathbf{u}_2)q_{\text{cav},n}(\mathbf{u}_3)\times \\ \Big[p(h_{1,n}; f_1, \mathbf{x}_n)p(h_{2,n}|f_2, h_{1,n})\Big]^{1-\alpha}\prod_{i=1, i\neq n}^{N}\Big[p(h_{1,i}; f_1, \mathbf{x}_i)p(h_{2,i}|f_2, h_{1,i})\Big],$$

where $q_{\text{cav},n}(\mathbf{u}_l) \triangleq p(\mathbf{u}_l)t_{l,n}^{1-\alpha}(\mathbf{u}_l)\prod_{i=1,i\neq n}^{N} t_{l,i}(\mathbf{u}_l) = \mathcal{N}(\mathbf{u}_l; \mathbf{m}_{\text{cav},l}, \mathbf{S}_{\text{cav},l})$.

Moment matching and update steps

In the ideal case, the approximate posterior can be found by minimising the KL divergence from it to the exact posterior, $\text{KL}(p(\cdot|\mathbf{y}, \mathbf{X})||q(\cdot))$. However, this is intractable. Instead, the Power EP procedure replaces the exact posterior by a surrogate posterior such that the resulting optimisation is tractable. This surrogate posterior, which is often called the tilted distribution, is formed by by multiplying the cavity distribution with a fraction of the exact factor, as follows,

$$\tilde{q}_n(\cdot) = q_{\text{cav},n}(\cdot)g_n^{\alpha} = p(f_{1,\neq\mathbf{u}_1}|\mathbf{u}_1)p(f_{2,\neq\mathbf{u}_2}|\mathbf{u}_2)p(f_{3,\neq\mathbf{u}_3}|\mathbf{u}_3)q_{\text{cav},n}(\mathbf{u}_1)q_{\text{cav},n}(\mathbf{u}_2)q_{\text{cav},n}(\mathbf{u}_3)\times \\ \Big[p(h_{1,n}; f_1, \mathbf{x}_n)p(h_{2,n}|f_2, h_{1,n})p^{\alpha}(y_n; f_3, h_{2,n})\Big]\prod_{i=1,i\neq n}^{N}\Big[p(h_{1,i}; f_1, \mathbf{x}_i)p(h_{2,i}|f_2, h_{1,i})\Big].$$

This surrogate posterior is then used to refine the approximate posterior, by minimising the KL divergence from the approximate posterior to the tilted distribution, $\text{KL}(\tilde{q}_n(\cdot)||q(\cdot))$. This minimisation problem is equivalent to matching the moments of $q(\cdot)$ to those of $\tilde{q}_n(\cdot)$. Computing the moments of $\tilde{q}_n(\cdot)$ is, however, analytically intractable as it involves propagating Gaussian processs through a non-linear hierarchy, resulting in a complex distribution over all random variables in the network. Fortunately, the structure of the approximate posterior comes to the rescue. As shown in chapter 2, since the approximate posterior is grounded on the pseudo-points, it is therefore sufficient to match zeroth, first and second order moments at the pseudo-points in the approximate posterior to that of the tilted distribution. This central result greatly simplifies the required computation and, since $\{q(\mathbf{u}_l)\}_{l=1}^{L}$ are assumed Gaussian, allows the following shortcut to obtain the mean and covarariance of the new approximate posterior at the pseudo-points,

$$\mathbf{m}_l^{\text{new}} = \mathbf{m}_l^{\backslash n} + \mathbf{S}_l^{\backslash n}\frac{\mathrm{d}\log\tilde{Z}_n}{\mathrm{d}\mathbf{m}_l^{\backslash n}}, \tag{4.28}$$

$$\mathbf{S}_l^{\text{new}} = \mathbf{S}_l^{\backslash n} - \mathbf{S}_l^{\backslash n}\left[\frac{\mathrm{d}\log\tilde{Z}_n}{\mathrm{d}\mathbf{m}_l^{\backslash n}}\left(\frac{\mathrm{d}\log\tilde{Z}_n}{\mathrm{d}\mathbf{m}_l^{\backslash n}}\right)^{\intercal} - 2\frac{\mathrm{d}\log\tilde{Z}_n}{\mathrm{d}\mathbf{S}_l^{\backslash n}}\right]\mathbf{S}_l^{\backslash n}, \tag{4.29}$$

where $\tilde{Z}_n$ is is normalising constant of $\tilde{q}_n(\cdot)$. Having the new mean and covariance, the approximate factor can be trivially updated by dividing the cavity distribution from the new approximate posterior. The inference scheme therefore reduces to evaluating $\tilde{Z}_n$ and its

gradient w.r.t. the cavity parameters. Note that,

$$\begin{aligned}\log \tilde{Z}_n &= \log \int_{f_1,f_2,f_3,\mathbf{h}_1,\mathbf{h}_2} \tilde{q}_n(f_1,f_2,f_3,\mathbf{h}_1,\mathbf{h}_2) \\ &= \log \int_{f_1,f_2,f_3,h_{1,n},h_{2,n}} q_{\text{cav},n}(f_1,f_2,f_3)\Big[p(h_{1,n};f_1,\mathbf{x}_n)p(h_{2,n}|f_2,h_{1,n})p^\alpha(y_n;f_3,h_{2,n})\Big] \\ &= \log \left\langle p^\alpha(y_n|f_3,h_{2,n})\right\rangle_{q_{\text{cav},n}(f_3)\tilde{q}(h_{2,n})}, \end{aligned} \tag{4.30}$$

where $q_{\text{cav},n}(f_l) = p(f_{l,\neq \mathbf{u}_l}|\mathbf{u}_l)q_{\text{cav},n}(\mathbf{u}_l)$ and

$$\tilde{q}(h_{2,n}) = \int_{h_{1,n},f_1,f_2} p(h_{2,n}|f_2,h_{1,n})q_{\text{cav},n}(f_2)p(h_{1,n}|f_1,\mathbf{x}_n)q_{\text{cav},n}(f_1). \tag{4.31}$$

Critically, the computation procedure required for eqs. (4.30) and (4.31) is identical to that required for the VFE approach (eqs. (4.18) and (4.19)). In particular, the tilted marginal, $\tilde{q}(h_{2,n})$, is analytically intractable for non-linear networks, but can be efficiently approximated by nesting simple Monte Carlo or Gaussian projection steps [see the previous section and chapter 3 for more details]. And similarly, the expectation in eq. (4.31) can be approximated by another layer of Gaussian projection or simple Monte Carlo.

When the Gaussian projection is used at each layer, computing $\log \tilde{Z}_n$ involves passing an approximate Gaussian distribution from the input layer to the output layer. As the mean and variance of the Gaussian approximation in each intermediate layer can be computed analytically, their gradients with respect to the mean and variance of the input distribution, as well as the cavity parameters of the current layers are also available. Since we require the gradients of the approximation to $\log \tilde{Z}_n$, we need to store these results in the forward propagation step, compute the approximate $\log \tilde{Z}_n$ and its gradients at the output layer and use the chain rule to pass the gradient information in the backward direction from the output layer to the input layer. This is reminiscent of the backpropagation algorithm in standard parametric neural networks, hence called the *probabilistic backpropagation* algorithm (Hernández-Lobato and Adams, 2015).

A particular case of the above procedure when $\alpha = 1$ has been described in (Bui et al., 2015, 2016), however the presentation and reinterpretation in this chapter allow other α values beyond $\alpha = 1$ to be used in the same tractable algorithmic procedure. Importantly, as $\alpha \to 0$, the VFE approach presented in the previous section is recovered. This result establishes the connection between the approached presented by Bui et al. (2015, 2016) and Salimbeni and Deisenroth (2017) as special cases of the same Power EP procedure, with different α values and different approximate uncertainty propagation methods in the network ($\alpha = 1$ and Gaussian projection, and $\alpha \to 0$ and simple Monte Carlo, respectively).

Approximate marginal likelihood and hyperparameter optimisation

The Power EP procedure is not guaranteed to converge in general, but if it does, the fixed points lie at the stationary points of the Power EP energy, and the negative of which can be used as an approximation to the log marginal likelihood,

$$\log p(\mathbf{y}) \approx \mathcal{F}_{\text{pep}} = \sum_{l=1}^{L} \Big[\Phi[q(\mathbf{u}_l)] - \Phi[p(\mathbf{u}_l)]\Big] + \frac{1}{\alpha}\sum_{n=1}^{N}\Big(\log \tilde{Z}_n + \sum_{l=1}^{L}\Big[\Phi[q^{\backslash n}(\mathbf{u}_l)] - \Phi[q(\mathbf{u}_l)]\Big]\Big),$$

where $\Phi[\mathcal{N}(\mathbf{x};\mathbf{m},\mathbf{S})] = \frac{1}{2}\mathbf{m}\mathbf{S}^{-1}\mathbf{m}^{\intercal} + \frac{1}{2}\log|\mathbf{S}|$. As $\alpha \to 0$, the Power EP energy tends to the variational free-energy. However, unlike the variational free-energy, there is no guarantee for the Power EP energy to be an upper bound of the negative marginal likelihood for a general α. This is a caveat when using the Power EP energy for hyperparameter optimisation, as a lower bound when minimised can go arbitrarily small. However, optimisation is found to often work well in practice, and the energy is often close to the exact negative marginal likelihood in many models.

As $\log \tilde{Z}_n$ is already required for the moment matching step in the Power EP iterative procedure, the approximate marginal likelihood above and its gradients w.r.t. the model hyperparameters can be computed at no additional cost. The overall computational complexity is $\mathcal{O}(NLM^2)$. Furthermore, Power EP requires the approximate factors to be stored in memory, which has a cost of $\mathcal{O}(NLM^2)$ for DGPs as we need to store the mean and the covariance matrix for each factor.

Direct EP energy minimisation with a tied factor constraint

In order to reduce the expensive memory footprint of Power EP, the data factor can be tied. That is the posterior of the pseudo-points is approximated by $q(\mathbf{u}_l) \propto p(\mathbf{u}_l)t(\mathbf{u}_l)^N$, where the factor $t(\mathbf{u})$ could be thought of as an *average* data factor that captures the average effect of a likelihood term on the posterior. Approximations of this form were recently used in the Stochastic EP algorithm (Li et al., 2015; Dehaene and Barthelmé, 2015) and although seemingly limited, in practice were found to perform almost as well as full Power EP while significantly reducing Power EP's memory requirement, from $\mathcal{O}(NLM^2)$ to $\mathcal{O}(LM^2)$ in our case.

The original Stochastic EP work devised modified versions of the EP updates appropriate for the new form of the approximate posterior. Originally Bui et al. (2015) applied this method to DGPs. However, Bui et al. (2016); Hernández-Lobato et al. (2016) later found an alternative approach that has superior performance, which is to optimise the Power EP energy directly to refine the approximating factors. The benefit is that the approximate Power EP energy can be jointly optimised for the hyperparameters, including the pseudo-inputs, at the same time. Normally, optimisation of the EP energy requires a double-loop algorithm,

which is computationally inefficient, however the use of tied factors simplifies the approximate marginal likelihood and allows direct optimisation. The approximate marginal likelihood becomes,

$$\begin{aligned}\mathcal{F}_{\text{pep}} &\approx \sum_{l=1}^{L}\Big[\Phi[q(\mathbf{u}_l)] - \Phi[p(\mathbf{u}_l)]\Big] + \frac{1}{\alpha}\sum_{n=1}^{N}\Big(\log\tilde{Z}_n + \sum_{l=1}^{L}\Big[\Phi[q^{\setminus 1}(\mathbf{u}_l)] - \Phi[q(\mathbf{u}_l)]\Big]\Big),\\ &= \sum_{l=1}^{L}\Big[(1-\frac{N}{\alpha})\Phi[q(\mathbf{u}_l) + \frac{N}{\alpha}\Phi[q^{\setminus 1}(\mathbf{u}_l)] - \Phi[p(\mathbf{u}_l)]\Big] + \frac{1}{\alpha}\sum_{n=1}^{N}\log\tilde{Z}_n, \qquad (4.32)\end{aligned}$$

since the cavity distribution $q^{\setminus n}(\mathbf{u}_l) \propto q(\mathbf{u})/\tilde{t}_n^{\alpha}(\mathbf{u}_l) = q(\mathbf{u}_l)/t^{\alpha}(\mathbf{u}_l) = q^{\setminus 1}(\mathbf{u}_l)$ is the same for all training points. This elegantly removes the need for a double-loop algorithm, since we can posit a form for the approximate posterior and optimise the above approximate marginal likelihood directly. However, it is important to note that, in general, optimising this objective will not give the same solution as optimising the full negative Power EP energy. Despite this difference, as $\alpha \to 0$, the above objective still becomes the exact negative variational free-energy in eq. (4.17).

Stochastic optimisation for scalable training

The propagation and moment-matching as described above costs $\mathcal{O}(LM^2)$ and needs to be repeated for all datapoints in the training set in batch mode, resulting in an overall complexity of $\mathcal{O}(NLM^2)$. Fortunately, the last term of the objective in the Power EP energy is a sum of independent terms, i.e. its computation can be distributed, resulting in a substantial decrease in computational cost. Furthermore, the objective is suitable for stochastic optimisation. In particular, an unbiased noisy estimate of the objective and its gradients can be obtained using a minibatch of training datapoints,

$$\mathcal{F}_{\text{pep}} = \sum_{l=1}^{L}\Big[(1-\frac{N}{\alpha})\Phi[q(\mathbf{u}_l) + \frac{N}{\alpha}\Phi[q^{\setminus 1}(\mathbf{u}_l)] - \Phi[p(\mathbf{u}_l)]\Big] + \frac{1}{\alpha}\frac{N}{|B|}\sum_{b=1}^{|B|}\log\tilde{Z}_b,$$

where $|B|$ denotes the minibatch size.

4.5 Alternative posterior approximations

Selecting a rich class of approximate posteriors that can be used for a wide class of approximation methods and that enables tractable inference and learning in DGPs is an active area of research. We have described two approximation families, one with mean-field parameterised Gaussian approximations and one with an explicit conditional distribution for the hidden variables, and how to use them for inference and learning using variational inference and Power EP. A clear advantage of the approximation with an explicit conditional distribution

shown in section 4.4 is that the posterior correlation between the latent functions and the latent variables is explicitly specified, as opposed to the mean-field assumption in section 4.3.

Alternative approximate posteriors that enable the same flexibility, while maintaining the same computational and memory complexities are possible. In this section, we review one such choice, a nested variational approach by Hensman and Lawrence (2014). The original presentation in this paper was rather convoluted and it is not intermediately clear what variational approximation was chosen, nor indeed whether the approach can be interpreted in terms of a a single variational distribution. In this section, we attempt to summarise this approach using a more standard route which shows a clear connection to the variational approximations and inference schemes reviewed in the previous sections. In particular, we interpret this approach as a variational free-energy scheme with a particular choice for the variational approximation that possesses an explicit conditional distributions for the hidden variables as follows,

$$q(f_1, f_2, f_3, h_1, h_2) = q(f_1)q(f_2)q(f_3)\prod_{n=1}^{N} q(h_{1,n}|\mathbf{u}_1)q(h_{2,n}|\mathbf{u}_2). \tag{4.33}$$

Note that $q(f_l) = p(f_{l,\neq\mathbf{u}_l}|\mathbf{u}_l)q(\mathbf{u}_l)$, and $q(\mathbf{u}_1)$, $q(\mathbf{u}_2)$ and $q(\mathbf{u}_3)$ are assumed Gaussian and will be parameterised and optimised. In addition, $q(h_1|\mathbf{u}_1)$, $q(h_2|\mathbf{u}_2)$ are explicitly defined and depend on $q(\mathbf{u}_1)$ and $q(\mathbf{u}_2)$ in a way that the resulting variational free-energy is simple to compute. Substituting the above distribution into the standard variational lower bound of the marginal likelihood leads to,

$$\mathcal{F}_{\text{vfe}} = \mathcal{F}_{\text{vfe},1} + \sum_{n=1}^{N} \mathcal{F}_{\text{vfe},2,n},$$

where $\mathcal{F}_{\text{vfe},1} - \text{KL}(q(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)||p(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3))$,

$$\mathcal{F}_{\text{vfe},2,n} = \Big\langle -\log q(h_{1,n}|\mathbf{u}_1) - \log q(h_{2,n}|\mathbf{u}_2) + \langle \log p(y_n|f_3, h_{2,n})\rangle_{p(f_{3,\neq\mathbf{u}_3})} + \langle \log p(h_{2,n}|f_2, h_{1,n})\rangle_{p(f_{2,\neq\mathbf{u}_2})} + \langle \log p(2, n|f_1, x)\rangle_{p(f_{1,\neq\mathbf{u}_1})} \Big\rangle_{q(\mathbf{u}_1,\mathbf{u}_2,\mathbf{u}_3,h_{1,n},h_{2,n})}$$

First, we can analytically compute the expected log likelihood for each layer in the equation above,

$$\text{E}_1 = \langle \log p(h_{1,n}|f_1, x)\rangle_{p(f_{1,\neq\mathbf{u}_1})} = \log \mathcal{N}(h_{1,n}; \mathbf{A}_{1,n}\mathbf{u}_1, \sigma_1^2) - \frac{1}{2\sigma_1^2}\mathbf{B}_{1,n} \tag{4.34}$$

$$\text{E}_2 = \langle \log p(h_{2,n}|f_2, h_{1,n})\rangle_{p(f_{2,\neq\mathbf{u}_2})} = \log \mathcal{N}(h_{2,n}; \mathbf{A}_{2,n}\mathbf{u}_2, \sigma_2^2) - \frac{1}{2\sigma_2^2}\mathbf{B}_{2,n} \tag{4.35}$$

$$\text{E}_3 = \langle \log p(y_n|f_3, h_{2,n})\rangle_{p(f_{3,\neq\mathbf{u}_3})} = \log \mathcal{N}(y_n; \mathbf{A}_{3,n}\mathbf{u}_3, \sigma_3^2) - \frac{1}{2\sigma_3^2}\mathbf{B}_{3,n} \tag{4.36}$$

where $\mathbf{A}_{l,n} = \mathbf{k}_{n\mathbf{u},l}\mathbf{K}_{\mathbf{uu},l}^{-1}$ and $\mathbf{B}_{l,n} = \mathbf{k}_{nn,l} - \mathbf{k}_{n\mathbf{u},l}\mathbf{K}_{\mathbf{uu},l}^{-1}\mathbf{k}_{\mathbf{u}n,l}$, where we have assumed a Gaussian observation likelihood so that E_3 is analytic, however, this is not a strict requirement. Now we judiciously choose

$$q(h_{1,n}|\mathbf{u}_1) = \mathcal{N}(h_{1,n}; \mathbf{A}_{1,n}\mathbf{u}_1, \sigma_1^2), \tag{4.37}$$

such that substituting this into the bound leads to the cancellation of $\log q(h_{1,n}|\mathbf{u}_1)$. Furthermore, $h_{1,n}$ can be marginalised out as follows,

$$\begin{aligned}
\mathcal{F}_{\text{vfe},2,n} &= \left\langle -\cancel{\log q(h_{1,n}|\mathbf{u}_1)} - \log q(h_{2,n}|\mathbf{u}_2) + \cancel{\log \mathcal{N}(h_{1,n}; \mathbf{A}_{1,n}\mathbf{u}_1, \sigma_1^2)} - \frac{1}{2\sigma_1^2}\mathbf{B}_{1,n} + \mathrm{E}_2 + \mathrm{E}_3 \right\rangle_{q(\cdot)} \\
&= \left\langle -\log q(h_{2,n}|\mathbf{u}_2) - \frac{1}{2\sigma_1^2}\mathbf{B}_{1,n} + \langle \mathrm{E}_2 \rangle_{q(h_{1,n})} + \mathrm{E}_3 \right\rangle_{q(\mathbf{u}_2,\mathbf{u}_3,h_{2,n})}
\end{aligned}$$

where

$$\begin{aligned}
\langle \mathrm{E}_2 \rangle_{q(h_{1,n})} &= \log \mathcal{N}(h_{2,n}; \langle \mathbf{A}_{2,n} \rangle_{q(h_{1,n})}\mathbf{u}_2, \sigma_2^2) - \frac{1}{2\sigma_2^2}\langle \mathbf{B}_{2,n} \rangle_{q(h_{1,n})} \\
&\quad + \frac{1}{2\sigma_2^2}\mathrm{tr}\left[\mathbf{u}_2\mathbf{u}_2^\intercal(\langle \mathbf{A}_{2,n}^\intercal \rangle_{q(h_{1,n})}\langle \mathbf{A}_{2,n} \rangle_{q(h_{1,n})} - \langle \mathbf{A}_{2,n}^\intercal \mathbf{A}_{2,n} \rangle_{q(h_{1,n})})\right], \text{ and} \\
q(h_{1,n}) &= \langle q(h_{1,n}|\mathbf{u}_1) \rangle_{q(\mathbf{u}_1)} = \mathcal{N}(h_{1,n}; \mathbf{A}_{1,n}\mathbf{m}_1, \mathbf{A}_{1,n}\mathbf{S}_1\mathbf{A}_{1,n}^\intercal + \sigma_1^2)
\end{aligned}$$

We can now judiciously choose $q(h_{2,n}|\mathbf{u}_2)$ in a similar fashion to $q(h_{1,n}|\mathbf{u}_1)$,

$$q(h_2|\mathbf{u}_2) = \mathcal{N}(h_{2,n}; \langle \mathbf{A}_{2,n} \rangle_{q(h_{1,n})}\mathbf{u}_2, \sigma_2^2), \tag{4.38}$$

leading to the cancellation of $q(h_{2,n}|\mathbf{u}_2)$ in the bound, as follows,

$$\begin{aligned}
\mathcal{F}_{\text{vfe},2,n} &= \left\langle -\cancel{\log q(h_{2,n}|\mathbf{u}_2)} - \frac{1}{2\sigma_1^2}\mathbf{B}_{1,n} + \cancel{\log \mathcal{N}(h_{2,n}; \langle \mathbf{A}_{2,n} \rangle_{q(h_{1,n})}\mathbf{u}_2, \sigma_2^2)} - \frac{1}{2\sigma_2^2}\langle \mathbf{B}_{2,n} \rangle_{q(h_{1,n})} \right. \\
&\quad \left. + \frac{1}{2\sigma_2^2}\mathrm{tr}\left[\mathbf{u}_2\mathbf{u}_2^\intercal(\langle \mathbf{A}_{2,n}^\intercal \rangle_{q(h_{1,n})}\langle \mathbf{A}_{2,n} \rangle_{q(h_{1,n})} - \langle \mathbf{A}_{2,n}^\intercal \mathbf{A}_{2,n} \rangle_{q(h_{1,n})})\right] + \mathrm{E}_3 \right\rangle_{q(\mathbf{u}_2,\mathbf{u}_3,h_{2,n})}, \\
&= \langle \mathrm{E}_3 \rangle_{q(\mathbf{u}_3,h_{2,n})} - \frac{1}{2\sigma_1^2}\mathbf{B}_{1,n} - \frac{1}{2\sigma_2^2}\langle \mathbf{B}_{2,n} \rangle_{q(h_{1,n})} \\
&\quad + \frac{1}{2\sigma_2^2}\mathrm{tr}\left[(\mathbf{S}_2 + \mathbf{m}_2\mathbf{m}_2^\intercal)(\langle \mathbf{A}_{2,n}^\intercal \rangle_{q(h_{1,n})}\langle \mathbf{A}_{2,n} \rangle_{q(h_{1,n})} - \langle \mathbf{A}_{2,n}^\intercal \mathbf{A}_{2,n} \rangle_{q(h_{1,n})})\right],
\end{aligned}$$

where we have placed $\langle \mathrm{E}_2 \rangle_{q(h_{1,n})}$ back in the bound and integrated out $\mathbf{u}_2$, and,

$$\begin{aligned}\langle \mathrm{E}_3 \rangle_{q(h_{2,n})} &= \log \mathcal{N}(y_n; \mathbf{A}_{3,n}\mathbf{u}_3, \sigma_3^2) - \frac{1}{2\sigma_3^2}\langle \mathbf{B}_{3,n} \rangle_{q(h_{2,n})} \\ &\quad + \frac{1}{2\sigma_3^2}\mathrm{tr}\left[\mathbf{u}_3\mathbf{u}_3^\intercal(\langle \mathbf{A}_{3,n}^\intercal \rangle_{q(h_{3,n})}\langle \mathbf{A}_{3,n} \rangle_{q(h_{2,n})} - \langle \mathbf{A}_{3,n}^\intercal \mathbf{A}_{3,n} \rangle_{q(h_{2,n})})\right], \text{ and} \\ q(h_{2,n}) &= \langle q(h_{2,n}|\mathbf{u}_2) \rangle_{q(\mathbf{u}_2)} = \mathcal{N}(h_{2,n}; \langle \mathbf{A}_{2,n} \rangle_{q(h_{1,n})}\mathbf{m}_2, \langle \mathbf{A}_{2,n} \rangle_{q(h_{1,n})}\mathbf{S}_2\langle \mathbf{A}_{2,n} \rangle_{q(h_{1,n})}^\intercal + \sigma_2^2)\end{aligned}$$

Finally, we can also integrate out $\mathbf{u}_3$ leading to,

$$\begin{aligned}\mathcal{F}_{\mathrm{vfe},2,n} &= \log \mathcal{N}(y_n; \mathbf{A}_{3,n}\mathbf{m}_3, \sigma_3^2) - \frac{1}{2\sigma_1^2}\mathbf{B}_{1,n} - \frac{1}{2\sigma_2^2}\langle \mathbf{B}_{2,n} \rangle_{q(h_{1,n})} - \frac{1}{2\sigma_3^2}\langle \mathbf{B}_{3,n} \rangle_{q(h_{2,n})} \\ &\quad + \frac{1}{2\sigma_2^2}\mathrm{tr}\left[(\mathbf{S}_2 + \mathbf{m}_2\mathbf{m}_2^\intercal)(\langle \mathbf{A}_{2,n}^\intercal \rangle_{q(h_{1,n})}\langle \mathbf{A}_{2,n} \rangle_{q(h_{1,n})} - \langle \mathbf{A}_{2,n}^\intercal \mathbf{A}_{2,n} \rangle_{q(h_{1,n})})\right] \\ &\quad + \frac{1}{2\sigma_3^2}\mathrm{tr}\left[(\mathbf{S}_3 + \mathbf{m}_3\mathbf{m}_3^\intercal)(\langle \mathbf{A}_{3,n}^\intercal \rangle_{q(h_{2,n})}\langle \mathbf{A}_{3,n} \rangle_{q(h_{2,n})} - \langle \mathbf{A}_{3,n}^\intercal \mathbf{A}_{3,n} \rangle_{q(h_{2,n})})\right] \\ &\quad - \frac{1}{2\sigma_3^2}\mathrm{tr}\left[\mathbf{S}_3\langle \mathbf{A}_{3,n}^\intercal \rangle_{q(h_{3,n})}\langle \mathbf{A}_{3,n} \rangle_{q(h_{2,n})}\right]. \end{aligned} \tag{4.39}$$

The above result can be trivially extend to networks with more layers, or with latent inputs.

Power EP using the variational approximation in eq. (4.33)?

In the previous sections, we have shown two classes of variational approximations that are general and can be used for both Power EP and VFE approaches. However, we have not been able to derive a tractable Power EP procedure that corresponds to the variational posterior in eq. (4.33).

4.6 Predictions

Given the approximate posterior and a new test input $\mathbf{x}^*$, we wish to make a prediction about the test output y^*. That is to find

$$p(y^*|\mathbf{x}^*, \mathbf{X}, \mathbf{y}) \approx \int_{f_1,f_2,f_3,h_1^*,h_2^*} p(y^*|f_3, h_2^*)p(h_2^*|f_2, h_1^*)p(h_1^*|f_1, \mathbf{x}^*) \prod_{l=1}^{3} p(f_{l,\neq\mathbf{u}}|\mathbf{u}_l)q(\mathbf{u}_l|\mathbf{X}, \mathbf{y}).$$

This predictive distribution is not analytically tractable, but fortunately, we can approximate it by a Gaussian sensity resulted from using the nested Gaussian projection sheme, or by a set of samples resulted from the nested simple Monte Carlo propagation – both of these techniques are discussed in sections 3.4 and 4.4.

4.7 Experiments

In this section, we compare the approximate Power EP schemes, i.e. direct optimisation of the approximate Power EP energy in eq. (4.32), on several toy regression problems, and compare the approximate Power EP scheme with $\alpha = 1$ to state-of-the-art methods for Bayesian neural networks on several real-world regression tasks.

4.7.1 Assessing different α values and various network sizes on toy datasets

We first evaluate the performance of the approximate Power EP scheme with different α values in training DGPs. These different variants are tested on two toy regression tasks, a noisy step function and a noisy periodic function as shown in figs. 4.4 and 4.11. DGPs with one and two GP layers, and various hidden dimensionalities are used. Each network was trained by optimising the approximate Power EP energy, using the Adam optimiser (Kingma and Ba, 2015) with 5000 iterations and learning rate 0.01. In particular, the variational approximation with an explicit conditional distribution for the hidden variables, as described in section 4.4, is used. The Gaussian projection step is used in both training and testing. Note that we do not directly compare to a VFE implementation here, but expect that the performance of Power EP with $\alpha = 0.001$ is close to that of the VFE approach.

Qualitative performance using different network sizes

In this experiment, we assess the performance of different network architectures when using the same approximation learning and inference scheme on the step function data. We use $\alpha = 0.001$ and $\alpha = 0.5$ and provide a qualitative evaluation for different network sizes in figs. 4.3 and 4.4. These figures show the Power EP energies during training and demonstrate that while deeper and wider networks take longer to train, they often yield better final training objectives and qualitatively better prediction.

Qualitative performance using different α values

We next compare the approximate energies and the predictions by using different α values on a fixed network size. Similar to the results in chapters 2 and 3, bigger α values often give lower energies, even when the variational approximations are identical after initialisation (see fig. 4.5). In the case of the step function, this difference leads to a better qualitative performance for bigger α values, as demonstrated in figs. 4.6 and 4.7. Notably, the noise variance tends to be smaller and the samples from the posterior exibit more non-Gaussian structure when bigger α values are used.

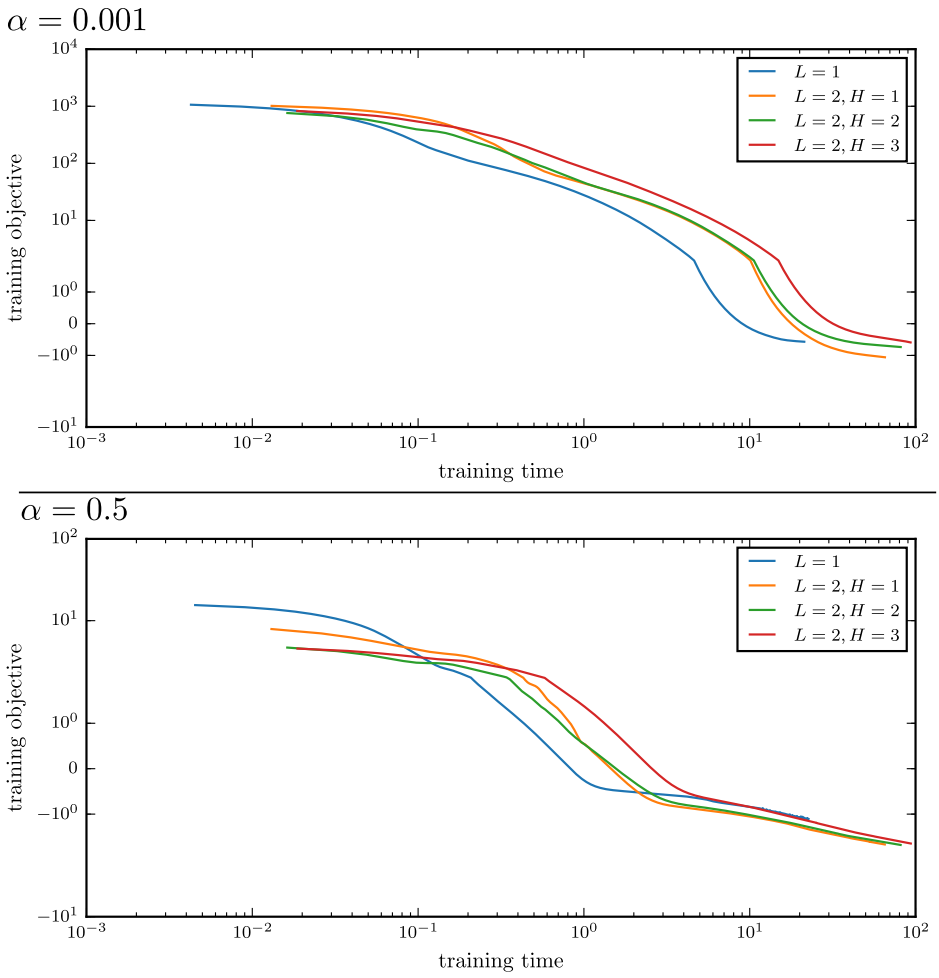


Fig. 4.3 Power EP energies on the training set during training, using different network architectures and two different α values: 0.001 and 0.5. L is the number of GP layers, and $L = 1$ means GP regression. H is the dimension of the hidden layer. Adding more layers and more hidden dimensions yield better final energies, though taking longer to train. This improvement often translates to better qualitative predictive performance as shown in fig. 4.4. Best viewed in colour.

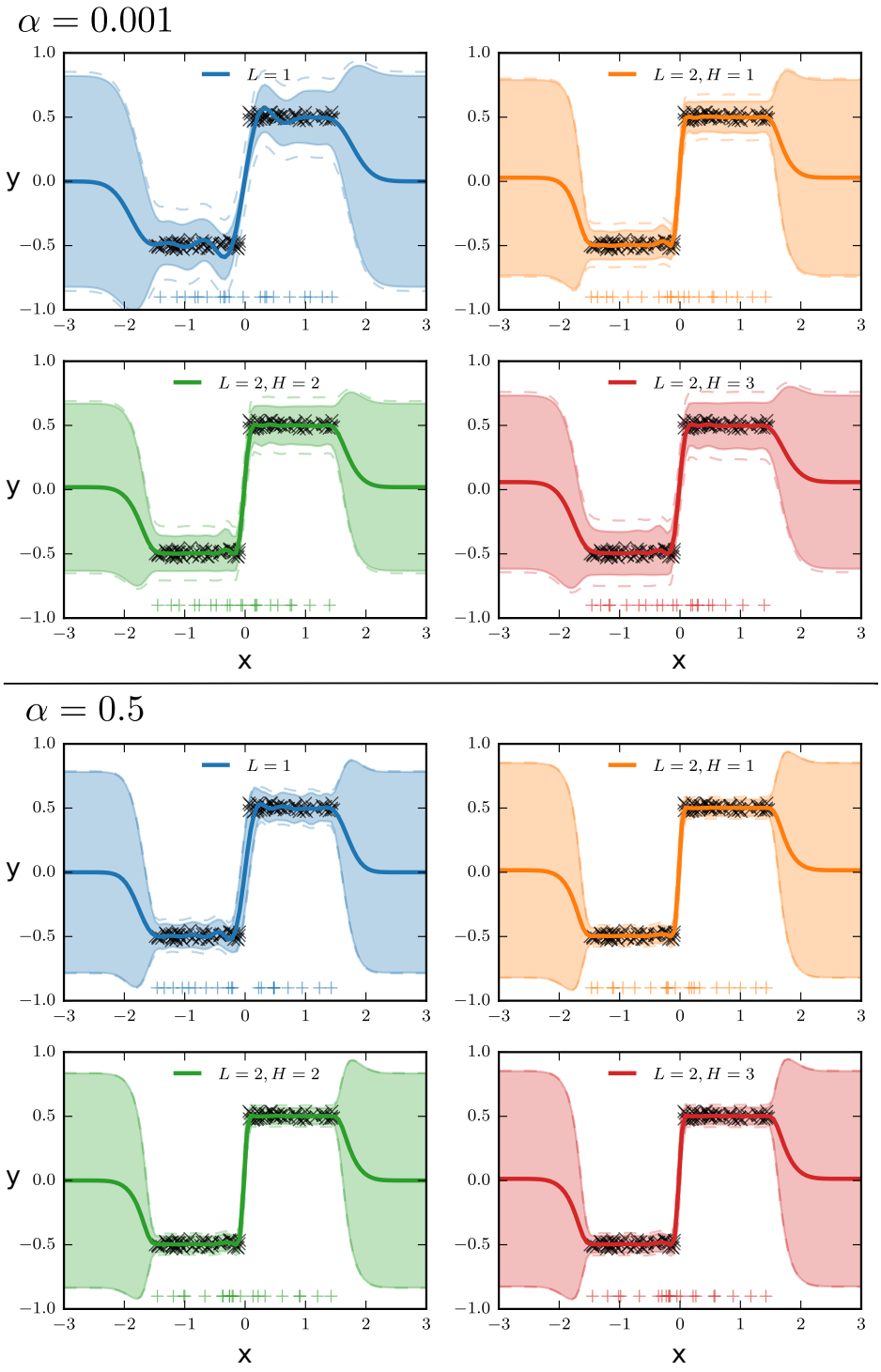


Fig. 4.4 Prediction after training on the step function data, using different network architectures and two different α values: 0.001 and 0.5. The predictive means and variances are obtained using the Gaussian projection approximation. L is the number of GP layers, and $L = 1$ means GP regression. H is the dimension of the hidden layer. Black markers are training points. Solid lines and shaded areas are the means and confidence intervals of the function values, respectively. The dashed lines show the confidence intervals of the noisy observations. The pluses show the locations of the pseudo points in the first layer. Adding more layers and more hidden dimensions yield qualitatively better fits. A quantitative evaluation is included in fig. 4.8. Best viewed in colour.

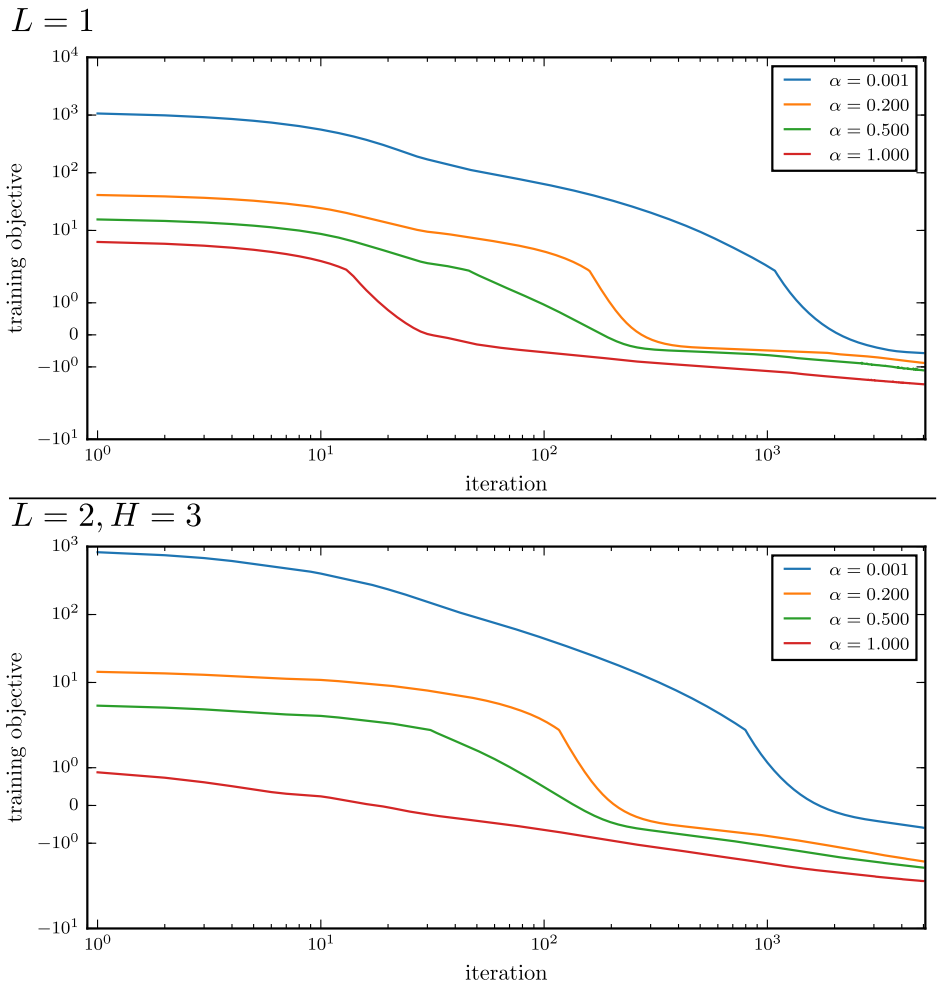


Fig. 4.5 Power EP energies on the training set during training, using different α values and two network architectures ($L = 1$ and $L = 2, H = 3$). The final energy after training is smaller, i.e. the approximate marginal likelihood is bigger, when α is bigger. This difference often translates to better qualitative predictive performance as shown in fig. 4.6, but there are cases that this does not hold (see figs. 4.10 and 4.11). Best viewed in colour.

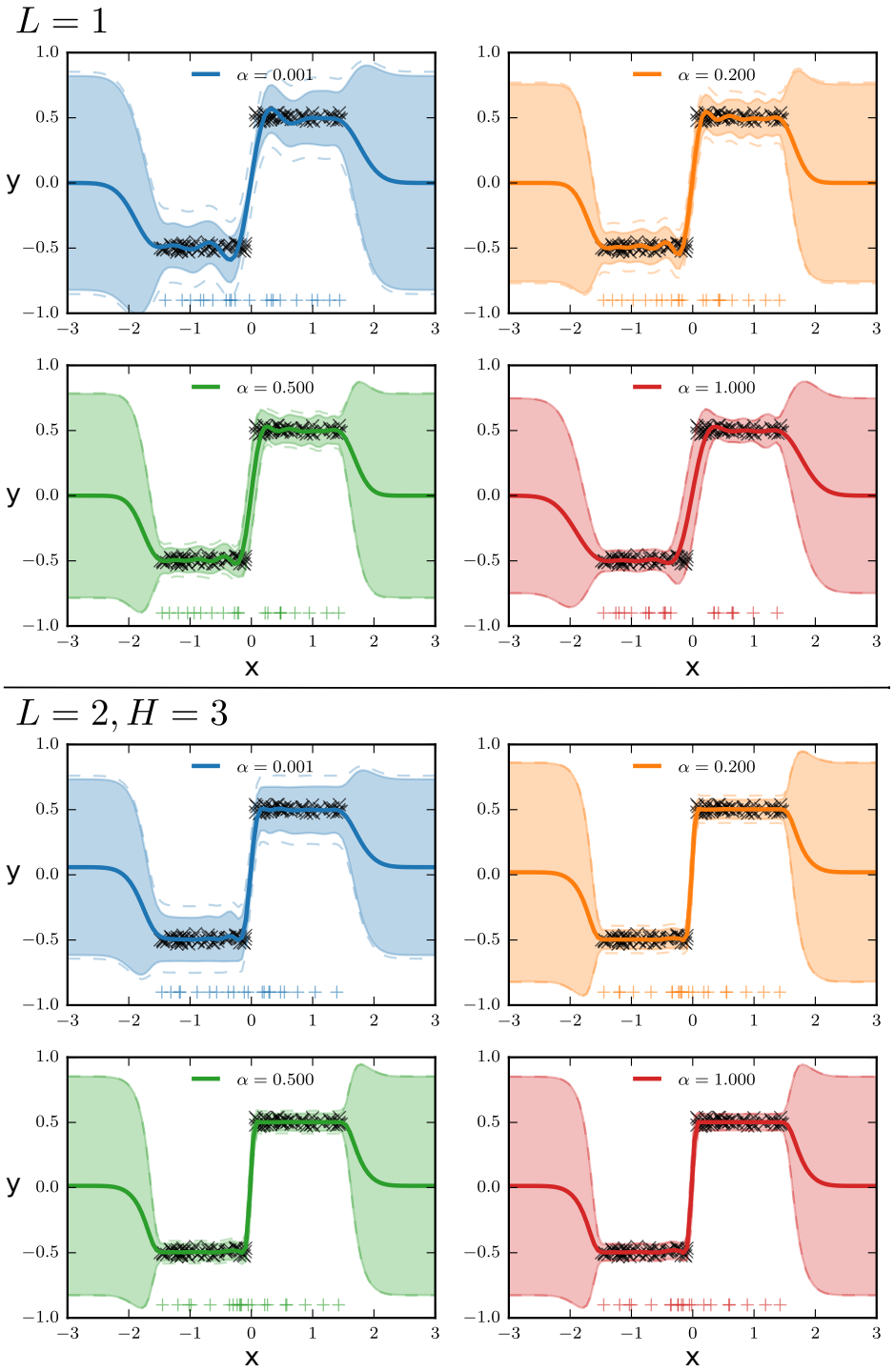


Fig. 4.6 Prediction after training on the step function data, using different α values and two network architectures ($L = 1$ and $L = 2, H = 3$). The predictive means and variances are obtained using the Gaussian projection approximation. Black markers are training points. Solid lines and shaded areas are the means and confidence intervals of the function values, respectively. The dashed lines show the confidence intervals of the noisy observations. The pluses show the locations of the pseudo points in the first layer. In this example, bigger α values give qualitatively better fit, and notably, learn a smaller observation noise. However, this trend does not hold in all cases (see figs. 4.10 and 4.11). Best viewed in colour.

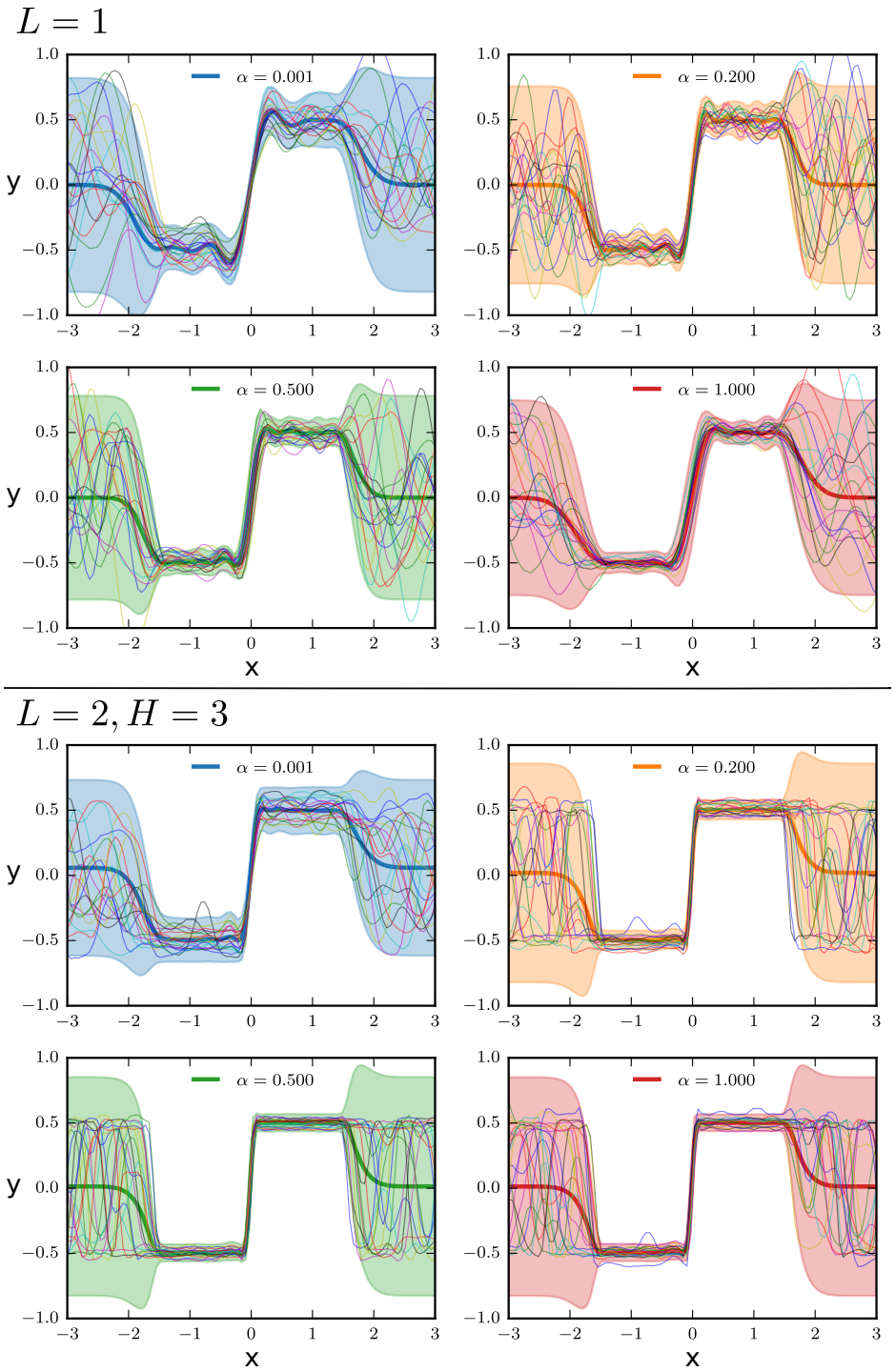


Fig. 4.7 Prediction after training on the step function data, using different α values and two network architectures ($L = 1$ and $L = 2, H = 3$). The predictive means and variances are obtained using the Gaussian projection approximation. This figure is identical to fig. 4.6, but we also include samples drawn from the posterior. Note that the prediction based on the Gaussian projection is generally very uncertain, while the sample functions exhibit far less uncertain non-Gaussian structure.

Quantitative performance using different network sizes and α values

We show the performance of various network sizes and α values on the train and test sets of the step function case, as evaluated by the average approximate log marginal likelihood and the average log predictive likelihood respectively, in fig. 4.8. There is a strong evidence that bigger and deeper networks outperform the shallow architecture (GP regression), and bigger α values generally outperform smaller values. Importantly, the approximate marginal likelihood given by Power EP can be used for model comparison, as a network with a higher approximate log marginal likelihood on the training set tends to perform better on the test set [see fig. 4.9]. However, these trends are not true in general for all datasets, and there are pitfalls when using a higher α and a bigger/deeper network.

A failure mode and potential workarounds

As shown in fig. 4.9, the approximate marginal likelihood provided by Power EP can be used for model comparison, and models with a higher log marginal likelihood tend to outperform ones that have smaller values. However, fig. 4.10 show that due to the *approximate* nature of the Power EP energy, having a higher negative energy can result in a poorer predictive performance. Figure 4.11 shows the predictions in the case $\alpha = 1$ has best final training objective, but learns a very small noise variance and gives poor uncertainty estimates.

More importantly, deeper networks trained using the approximate Power EP energy can perform poorly, compared to the standard shallow sparse GP regression. Note that these pitfalls have been shown in chapter 2 to also happen in the standard sparse GP regression case for high α values, e.g. $\alpha = 1$ (FITC) tends to prune out the pseudo-points, produce small observation noise, and explain wiggly functions using its heteroscedastic predictive variances. Recently, Díaz (2017) provided in-depth experiments on these issues and suggested that better initialisation schemes, for example, greedy layer-wise initialisation using variational sparse Gaussian processes, can potentially address the aforementioned pitfalls. An easier and faster workaround, in light of the results in fig. 4.11, is to use a smaller α value, e.g. $\alpha = 0.5$.

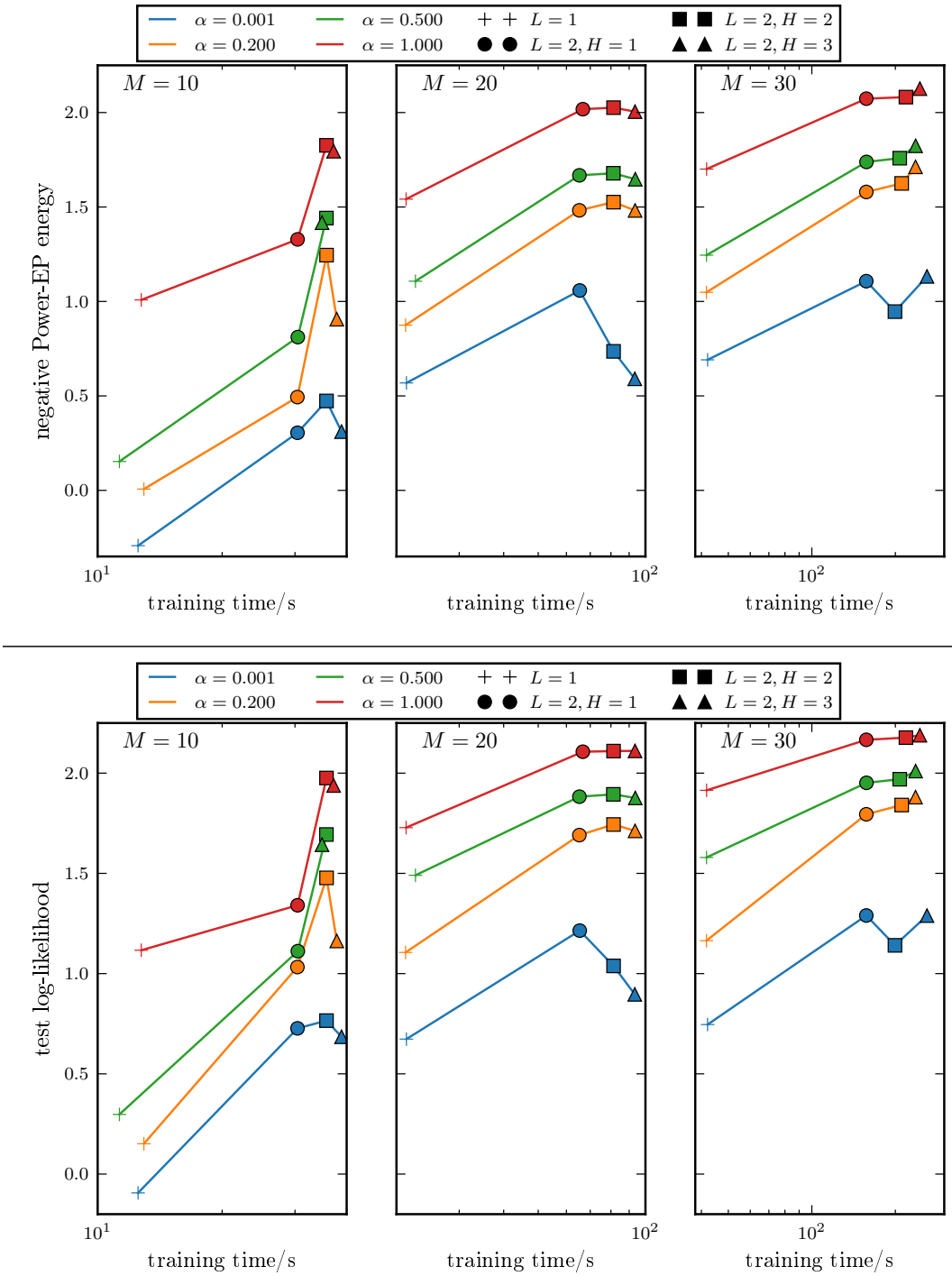


Fig. 4.8 The final negative Power EP energy on the training set [top] and the log-likelihood on the test set [bottom] vs. the total training time, on the step function data. In general, both the train approximate log marginal likelihood and the test log-likelihood are better for bigger α values, and for bigger networks. However, these trends are dataset-dependent, as there are cases when adding layers does not help or having a smaller α is better [as shown in figs. 4.10 and 4.11].

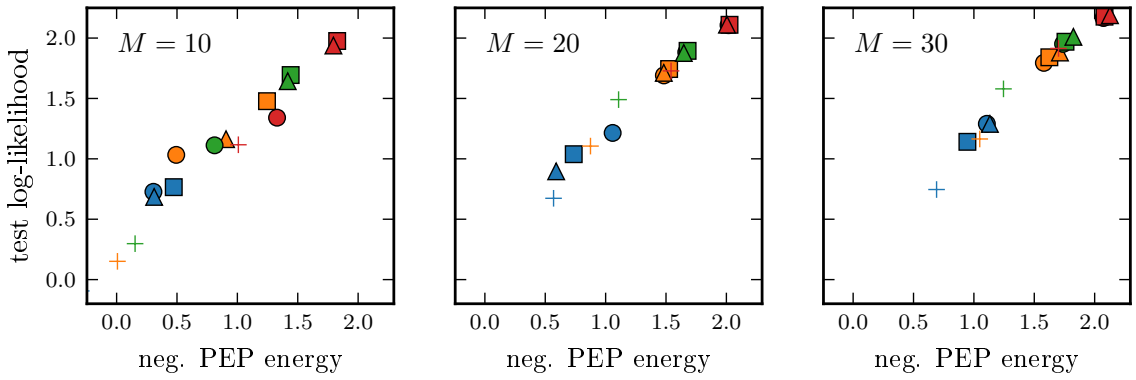


Fig. 4.9 The test log-likelihood vs the final approximate log marginal likelihood on the training set, on the step function data. See fig. 4.8 for more information about the markers and colours. In this case, there is a strong indication that having a more negative Power EP energy translates to better test performance, and bigger α values and bigger networks are generally better. However, a counter-example shown in fig. 4.10 demonstrates that these trends are not true in general.

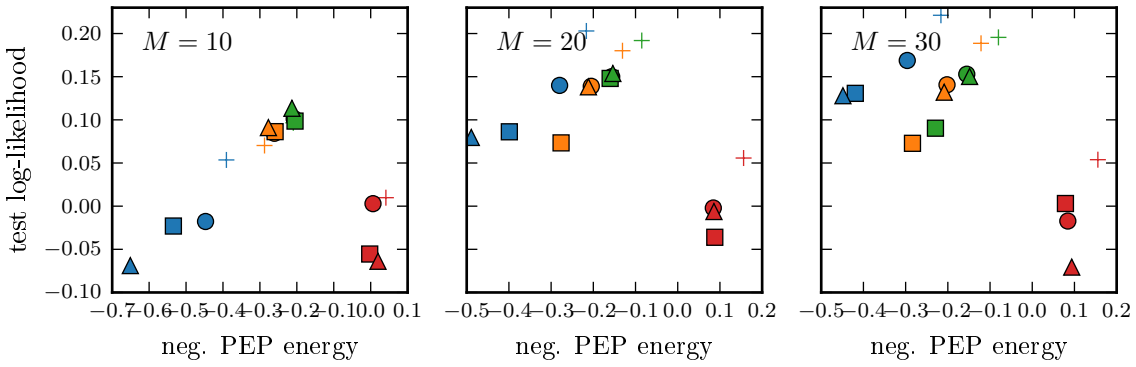


Fig. 4.10 The test log-likelihood vs the final approximate log marginal likelihood on the training set, on the periodic function data. See fig. 4.8 for more information about the markers and colours. In contrast to the results in fig. 4.9, bigger networks can perform poorly compared to the standard GP regression. In addition, having a larger negative Power EP energy does not translate to a better predictive performance on test set.

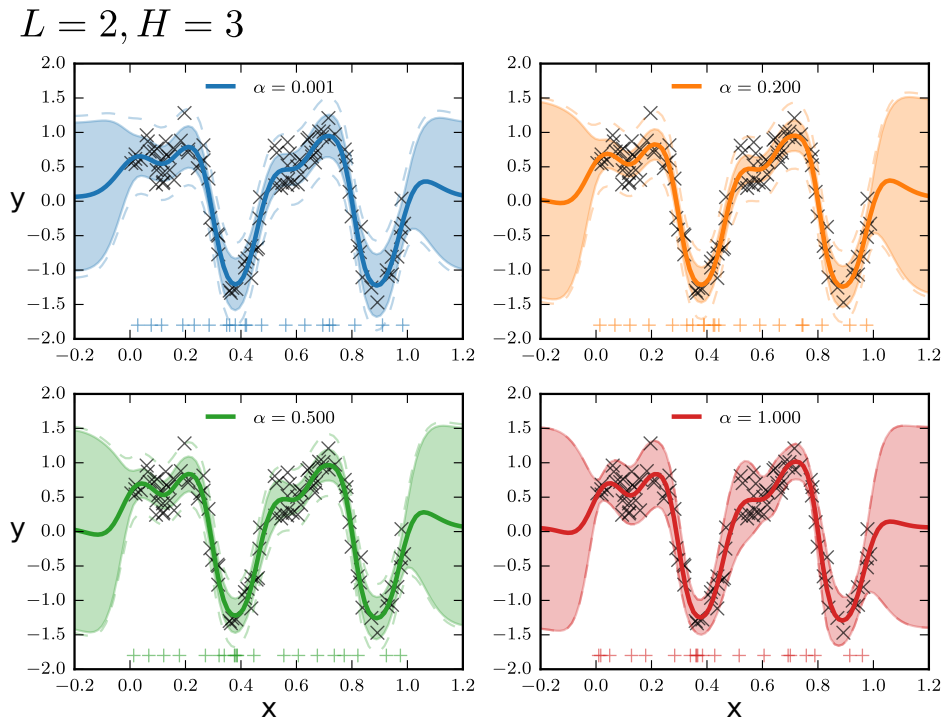


Fig. 4.11 Prediction after training on the periodic function data, using various α values. In this case, we use a DGP with one three-dimensional hidden layer. Despite having the highest approximate log marginal likelihood, $\alpha = 1$ gives a poor overall predictive performance. When $\alpha = 1$, the pseudo-points tend to pruned out or clumped together, the learnt noise is small, and the predictive variances are heteroscedastic even though the model does not explicitly model observation noise as input-dependent.

4.7.2 Regression on real-world datasets

In this experiment, the approximate Power EP scheme with $\alpha = 1$, i.e. direct optimisation of the approximate Power EP energy in eq. (4.32), for training DGPs is validated on several regression experiments using datasets from the UCI repository. In particular, we use the ten datasets and train/test splits used by Hernández-Lobato and Adams (2015) and Gal and Ghahramani (2016): 1 split for the *year* dataset [$N \approx 500000, D = 90$], 5 splits for the *protein* dataset [$N \approx 46000, D = 9$], and 20 for the others. In all the experiments reported here, we use Adam with the default learning rate (Kingma and Ba, 2015) for optimising our objective function. We use an exponentiated quadratic kernel with ARD lengthscales for each layer. The hyperparameters and pseudo point locations are different between functions in each layer. The lengthscales and pseudo-inputs of the first GP layer are sensibly initialised based on the median distance between datapoints in the input space and the k-means cluster centers respectively. We use long lengthscales and initial pseudo-inputs between $[-1, 1]$ for the higher layers to force them to start with an identity mapping. We parameterise the natural parameters of the average factor and initialise them with small random values. We

evaluate the predictive performance on the test set using two popular metrics: root mean squared error (RMSE) and mean log likelihood (MLL).

We compare our method (denoted by DGP) against sparse GP regression using the EP approximation (which has the same form as FITC, denoted by GP) and Bayesian neural network (denoted by BNN) regression using several state-of-the-art deterministic and sampling-based approximate inference techniques. As baselines, we include the results for BNNs reported in Hernández-Lobato and Adams (2015), BNN-VI(G)-1 and BNN-PBP-1, and in Gal and Ghahramani (2016), BNN-Dropout-1. The results reported for these methods are for networks with one hidden layer of 50 units (100 units for *protein* and *year*). Specifically, BNN-VI(G) uses a mean-field Gaussian approximation for the weights in the network and obtains the stochastic estimates of the bound and its gradient using a Monte Carlo approach (Graves, 2011). BNN-PBP employs Assumed Density Filtering and the probabilistic backpropagation algorithm to obtain a Gaussian approximation for the weights (Hernández-Lobato and Adams, 2015). BNN-Dropout is a recently proposed technique that employs *dropout* during training as well as at prediction time, that is to average over several predictions, each made by the entire network with a random proportion of the weights set to zero (Gal and Ghahramani, 2016). We implement other methods as follows,

- DGP: we evaluate three different architectures of DGPs, each with two GP layers and one hidden layer of one, two and three dimensions respectively (DGP-1, DGP-2, and DGP-3). We include the results for two settings of the number of pseudo-points, $M = 50$ and $M = 100$ respectively. Note that for the bigger datasets *protein* and *year*, we use $M = 100$ and $M = 200$ but do not annotate this in Figure 4.13. We choose these settings to ensure the run time for our method is smaller or comparable to that of other methods for BNNs.

- GP: we use the same number of pseudo-datapoints as in DGP (GP 50 and GP 100).

- BNN-VI(KW): this method, similar to (Graves, 2011), employs a mean-field Gaussian variational approximation but evaluates the variational free energy using the *reparameterisation trick* proposed by Kingma and Welling (2014). We use a diagonal Gaussian prior for the weights and fix the prior variance to 1. The noise variance of the Gaussian noise model is optimised together with the means and variances of the variational approximation using the variational free energy. We test two different network architectures with the rectified linear activation function, and one and two hidden layers, each of 50 units (100 for the two big datasets), denoted by VI(KW)-1 and VI(KW)-2 respectively.

- BNN-SGLD: we reuse the same networks with one and two hidden layers as with VI(KW) and approximately sample from the posterior over the weights using Stochastic Gradient Langevin Dynamics (SGLD) (Welling and Teh, 2011). We place a diagonal

Gaussian prior over the weights and parameterise the observation noise variance as $\sigma^2 = \log(1 + \exp(\kappa))$, a broad Gaussian prior over κ and sample κ using the same SGLD procedure. Two step sizes, one for the weights and one for κ, were manually tuned for each dataset. We use Autograd for the implementation of BNN-SGLD and BNN-VI(KW) (github.com/HIPS/autograd).

- BNN-HMC: We run Hybrid Monte Carlo (HMC) (Neal, 1993) using the MCMCstuff toolbox (Vanhatalo and Vehtari, 2006) for networks with one hidden layer. We place a Gaussian prior over the network weights and a broad inverse Gamma hyper-prior for the prior variance. We also assume an inverse Gamma prior over the observation noise variance. Note that this procedure takes a long time (e.g. 3 days for protein) and the *year* dataset is too large to be handled in this way.

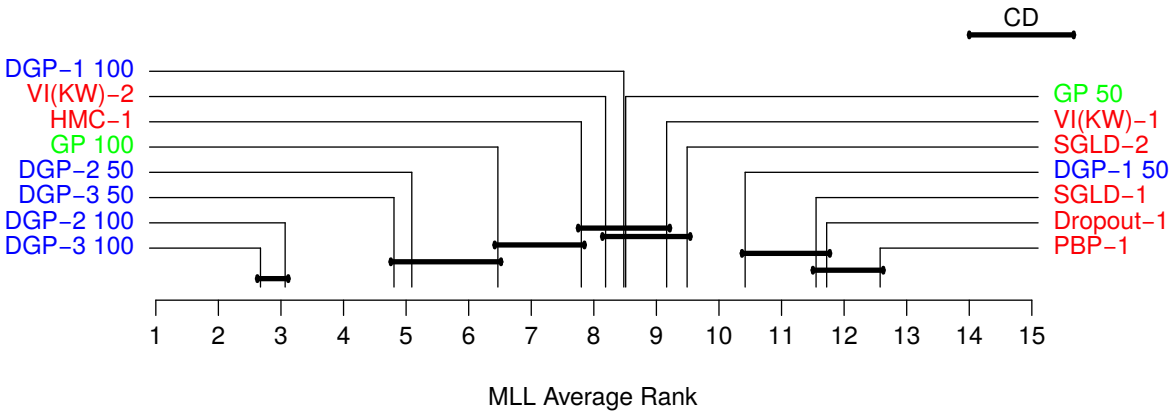


Fig. 4.12 The average rank of all methods across the datasets and their train/test splits, generated based on Demšar (2006). See the text for more details.

Figure 4.13 shows the average test log likelihood (MLL) for a subset of methods with their standard errors. We exclude methods that perform consistently poorly to improve readability. Full results and many more comparisons can be found in (Bui et al., 2016). We also evaluate the average rank of the MLL performance of all methods across the datasets and their train/test splits and include the results in Figure 4.12. This figure is generated using the comparison scheme provided by Demšar (2006), and shows statistical differences in the performance of the methods. More precisely, if the gap between the average ranks of any two methods is above the critical distance (shown on the top right), the two methods' performances are statistically significantly different. Methods that are not significantly different from each other are linked by a solid line. The rank result shows that DGPs with the approximate Power EP scheme are the best performing methods overall. Specifically, the DGP-3-100 architecture obtains the best performance on 6 out of 10 datasets and are competitive on the remaining four datasets. The performance of other DGP variants follows closely with the exception for DGP-1 which is a standard warped GP, the network with

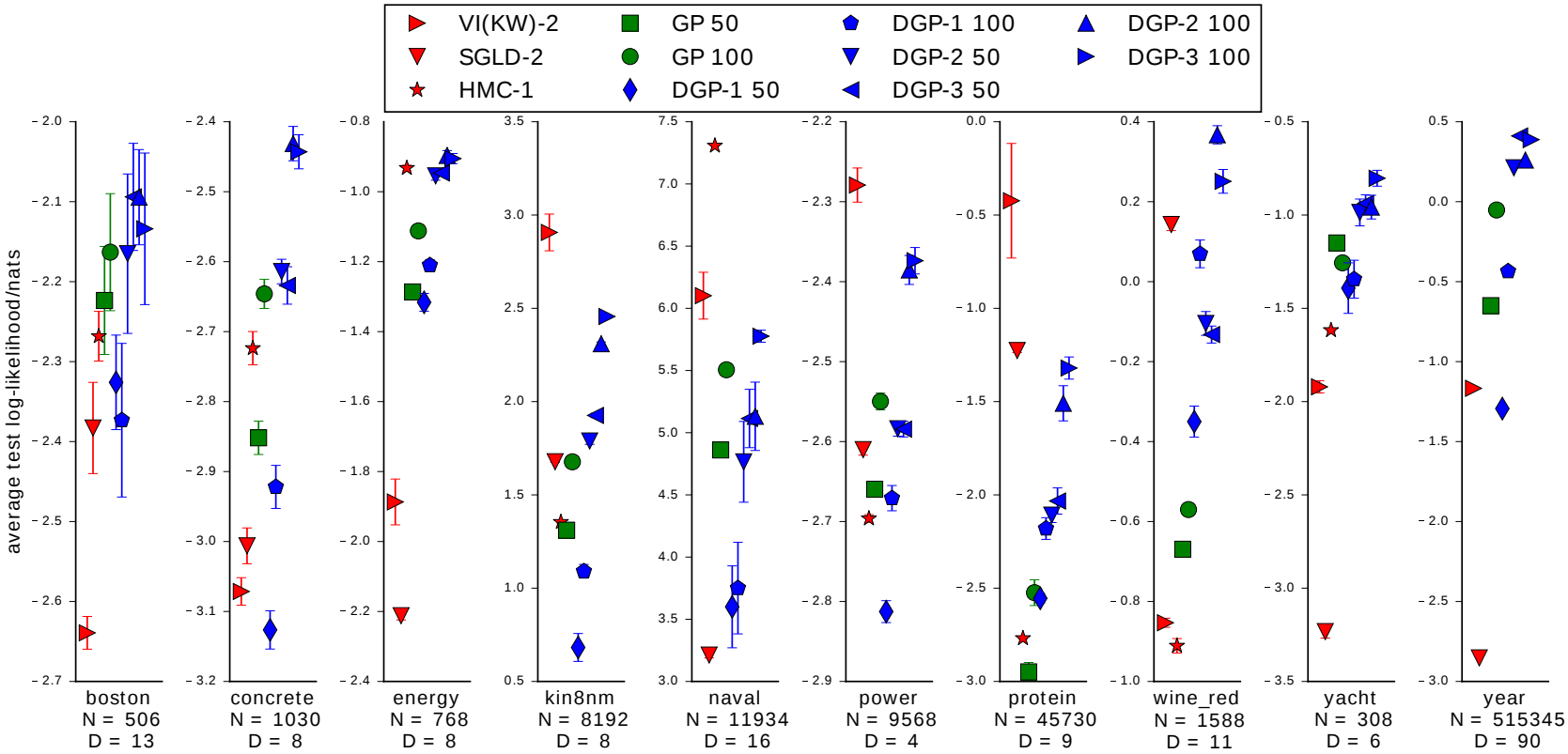


Fig. 4.13 Average predictive log likelihood of existing approaches for BNNs and GPs, and the proposed method for DGPs, across 10 datasets. The higher the better, and best viewed in colour. Full results are included in the supplementary material.

one-dimensional hidden layer. DGP-1 performs poorly compared to GP regression but is still competitive with several methods for BNNs. The results also strongly indicate that the predictive performance is almost always improved by adding extra hidden layers or extra hidden dimensions or extra pseudo-points.

The best non-GP method is BNN-VI(KW)-2 which obtains the best performance on three datasets. However, this method performs poorly on 6 out of 7 remaining datasets, pushing down the corresponding average rank. Despite this, VI(KW) is the best method of all deterministic approximations for BNNs with one or two hidden layers. Overall, the VI approach without the *reparameterisation trick* of Graves (2011), Dropout, and PBP perform poorly in comparison and give inaccurate predictive uncertainty.

Sampling-based methods such as SGLD and HMC obtain good predictive performance overall, but often require more tuning compared to other methods. In particular, HMC appears superior on one dataset, and competitive with DGPs on three other datasets; however, this method does not scale to large datasets.

The results for the RMSE metric follow the same trends with DGP-2 and DGP-3 performing as well or better compared to other methods. Interestingly, BNN-SGLD, despite being ranked relatively low according to the MLL metric, often provides good RMSE results perhaps unsurprisingly given the algorithm's similarity to stochastic gradient ascent.

4.8 Summary

This chapter has provided an extensive literature survey of approximation schemes for DGPs, and bridged the gap and provided a clear connection between many of these approximations, viewing them as special cases of performing approximate inference and learning using Power EP. We also considered several methods for propagating uncertainty in deep architectures and sidestepping a difficult marginalisation problem in the VFE/Power EP algorithm used. Some of the methods reviewed and proposed are evaluated on a range of toy and real-world and regression tasks. All of the approximations discussed can be extended and applied to classification tasks or when the inputs are random variables (unsupervised DGPs), however, we left this as future work. More experiments are also needed to assess the performance of the approximate posteriors in table 4.1, and whether EP, VFE or Power-EP with intermediate α values is best for training DGPs.

Chapter 5

Conclusions

5.1 Contributions

In this thesis, we have discussed a class of approximation schemes that allow practical and tractable approximate Bayesian inference and learning in a variety of Gaussian process models. In summary, this thesis unifies existing work and advances the frontiers of the following research themes:

Approximate Bayesian inference for Gaussian process models: We developed several unifying approximate inference and learning frameworks based on power expectation propagation for GP regression, classification, latent variable, state space, and hierarchical GP models. Critically, the new frameworks rely on *approximate inference* in the original, unmodified models, instead of (approximate/exact) inference in an *approximate model*. We also showed that the new frameworks allow state-of-the-art methods to emerge and that many existing techniques can be recovered as special cases.

Structured and correlated posterior approximations: This thesis considered a variety of structured approximate posteriors that admit efficient inference and tractable computation of the approximate marginal likelihood for hyperparameter learning. We also discussed several posterior approximations for the GP state space model and deep GPs, that explicitly retain the dependencies between the hidden variables and the global variables (GP mappings). Importantly, these approximations are judiciously chosen so that no parameterised approximations for the hidden variables are needed, whilst inference and learning can still be performed at no additional computational cost.

Principled uncertainty propagation in recurrent and deep architectures: We discussed several strategies for propagating probability densities for GP state space models and deep GPs based on linearisation, Gaussian projection/moment matching,

and simple Monte Carlo. These techniques can be applied at both training and test time, and are reminiscent of scented and unscented techniques in the signal processing and control literature.

5.2 Future work

The approximate inference and learning frameworks studied in this thesis allow many GP models to be tractably deployed in practice. We believe this thesis has only scratched the surface and there is a whole realm of possible future directions including,

more advanced structured posterior approximations that retain correlations between variables and, at the same time, enable computationally efficient and memory efficient inference and learning: for example we have suggested using an approximate posterior with an explicit conditional distribution for the hidden variables given the global variable for GPSSMs in section 3.10.1, in a similar fashion to section 4.4. These approximations could be applied to other hierarchical probabilistic models such as deep exponential families (Ranganath et al., 2015).

active learning of non-linear functions in models with hidden/latent variables: in this case, the hidden variables are not of interest in the active learning task and therefore need to be integrated out. For example, the information theoretic active learning objective of Houlsby et al. (2011), originally developed for GP classification, can be extended to the GPSSM as suggested in section 3.10.2.

models with multiple correlated outputs: we have only discussed sparse approximations for GP models in which multiple output dimensions are assumed conditionally independent given the input, however, these could be extended to models with explicitly correlated outputs given the input.

continual learning and transfer learning using streaming data and multiple related tasks: the training data in practice can come from multiple (related) sources/tasks and can arrive in chunks at different time points, hence it is important to devise schemes that can continually adjust the posterior approximations as data arrive, without the need to retrain from scratch. The approximations discussed in this thesis could be extended to these settings (see e.g. Bui et al., 2017a).

References

Álvarez, M. A., Luengo, D., Titsias, M. K., and Lawrence, N. D. (2010). Efficient multioutput Gaussian processes through variational inducing kernels. In *13th International Conference on Artificial Intelligence and Statistics*, pages 25–32. 9, 15

Bauer, M., van der Wilk, M., and Rasmussen, C. E. (2016). Understanding probabilistic sparse Gaussian process approximations. In *Advances in Neural Information Processing Systems 29*, pages 1525–1533. 13

Brown, L. D. (1986). *Fundamentals of Statistical Exponential Families with Applications in Statistical Decision Theory*. Institute of Mathematical Statistics, Hayward, CA. 23

Bui, T. D., Hernández-Lobato, J. M., Li, Y., Hernández-Lobato, D., and Turner, R. E. (2015). Training deep Gaussian processes using stochastic expectation propagation and probabilistic backpropagation. *arXiv preprint arXiv:1511.03405*. 92, 94, 98, 105, 106

Bui, T. D., Hernández-Lobato, J. M., Li, Y., Hernández-Lobato, D., and Turner, R. E. (2016). Deep Gaussian process for regression using approximate expectation propagation. In *33nd International Conference on Machine Learning*, pages 1472–1481. 7, 88, 89, 92, 94, 98, 105, 106, 122

Bui, T. D., Nguyen, C. V., and Turner, R. E. (2017a). Streaming sparse Gaussian process approximations. In *Advances in Neural Information Processing Systems 30*. 15, 126

Bui, T. D. and Turner, R. E. (2014). Tree-structured Gaussian process approximations. In *Advances in Neural Information Processing Systems 27*, pages 2213–2221. 9

Bui, T. D. and Turner, R. E. (2015). Stochastic variational inference for Gaussian process latent variable models using back constraints. In *NIPS Blackbox Learning and Inference workshop*. 72

Bui, T. D., Yan, J., and Turner, R. E. (2017b). A unifying framework for Gaussian process pseudo-point approximations using power expectation propagation. *The Journal of Machine Learning Research*, 18(104):1–72. 27, 31

Csató, L. (2002). *Gaussian Processes — Iterative Sparse Approximations*. PhD thesis, Aston University. 9, 14

Csató, L. and Opper, M. (2002). Sparse online Gaussian processes. *Neural Computation*, 14(3):641–669. 9, 16

Csató, L., Opper, M., and Winther, O. (2002). TAP Gibbs free energy, belief propagation and sparsity. In *Advances in Neural Information Processing Systems 15*, pages 657–663. 14

Dai, Z., Damianou, A., González, J., and Lawrence, N. D. (2016). Variational auto-encoded deep Gaussian processes. In *4th International Conference on Learning Representations*. 91

Dai, Z., Damianou, A., Hensman, J., and Lawrence, N. D. (2014). Gaussian process models with parallelization and GPU acceleration. *arXiv preprint arXiv:1410.4984*. 72

Damianou, A. (2015). *Deep Gaussian processes and variational propagation of uncertainty*. PhD thesis, University of Sheffield. 88, 89

Damianou, A. and Lawrence, N. D. (2013). Deep Gaussian processes. In *16th International Conference on Artificial Intelligence and Statistics*, pages 207–215. 87, 88, 89, 91, 92, 94, 96

Dawson, M. R. (1998). *Understanding cognitive science*. Blackwell Publishing. 16

Dehaene, G. and Barthelmé, S. (2015). Expectation propagation in the large-data limit. *arXiv preprint arXiv:1503.08060*. 38, 70, 106

Deisenroth, M. P. (2010). *Efficient Reinforcement Learning using Gaussian Processes*. PhD thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany. 9

Deisenroth, M. P. and Mohamed, S. (2012). Expectation propagation in Gaussian process dynamical systems. In *Advances in Neural Information Processing Systems 25*, pages 2609–2617. 46, 61, 102

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30. 122

Depeweg, S., Hernández-Lobato, J. M., Doshi-Velez, F., and Udluft, S. (2016a). Learning and policy search in stochastic dynamical systems with Bayesian neural networks. In *4th International Conference on Learning Representations*. 36

Depeweg, S., Hernández-Lobato, J. M., Doshi-Velez, F., and Udluft, S. (2016b). Uncertainty decomposition in Bayesian neural networks with latent variables. *arXiv preprint arXiv:1706.08495*. 86

Dezfouli, A. and Bonilla, E. V. (2015). Scalable inference for Gaussian process models with black-box likelihoods. In *Advances in Neural Information Processing Systems 28*, pages 1414–1422. 26

Díaz, S. P. (2017). Pathologies of deep Gaussian processes using approximate expectation propagation. Master's thesis, University of Cambridge. 117

Duvenaud, D., Lloyd, J., Grosse, R., Tenenbaum, J., and Ghahramani, Z. (2013). Structure discovery in nonparametric regression through compositional kernel search. In *30th International Conference on Machine Learning*, pages 1166–1174. 6

Duvenaud, D., Rippel, O., Adams, R. P., and Ghahramani, Z. (2014). Avoiding pathologies in very deep networks. In *17th International Conference on Artificial Intelligence and Statistics*. 91

Eleftheriadis, S., Nicholson, T. F., Deisenroth, M. P., and Hensman, J. (2017). Identification of Gaussian process state space models. In *Advances in Neural Information Processing Systems 30*. 46, 47, 55

Figueiras-Vidal, A. and Lázaro-Gredilla, M. (2009). Inter-domain Gaussian processes for sparse inference using inducing features. In *Advances in Neural Information Processing Systems 22*, pages 1087–1095. 9, 12

Frigola, R. (2015). *Bayesian Time Series Learning with Gaussian Processes*. PhD thesis, University of Cambridge. 44, 45

Frigola, R., Chen, Y., and Rasmussen, C. E. (2014). Variational Gaussian process state-space models. In *Advances in Neural Information Processing Systems 27*, pages 3680–3688. 9, 15, 46, 47, 48, 51, 52

Frigola, R., Lindsten, F., Schön, T. B., and Rasmussen, C. E. (2013). Bayesian inference and learning in Gaussian process state-space models with particle MCMC. In *Advances in Neural Information Processing Systems 26*, pages 3156–3164. 46

Gal, Y., Chen, Y., and Ghahramani, Z. (2015). Latent Gaussian processes for distribution estimation of multivariate categorical data. In *32nd International Conference on Machine Learning*, pages 645–654. 72, 101

Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *33rd International Conference on Machine Learning*. 120, 121

Gal, Y., van der Wilk, M., and Rasmussen, C. E. (2014). Distributed variational inference in sparse Gaussian process regression and latent variable models. In *Advances in Neural Information Processing Systems 27*, pages 3257–3265. 72

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2014a). *Bayesian data analysis*, volume 2. 2

Gelman, A., Vehtari, A., Jylänki, P., Robert, C., Chopin, N., and Cunningham, J. P. (2014b). Expectation propagation as a way of life. *arXiv preprint arXiv:1412.4869*. 25, 26

Ghahramani, Z. (2013). Bayesian non-parametrics and the probabilistic approach to modelling. *Phil. Trans. R. Soc. A*, 371(1984):20110553. 2, 3

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452. 2

Ghahramani, Z. and Roweis, S. T. (1998). Learning nonlinear dynamical systems using an EM algorithm. In *Advances in Neural Information Processing Systems 11*, pages 431–437. 45

Girard, A., Rasmussen, C. E., Quiñonero-Candela, J., and Murray-Smith, R. (2003). Gaussian process priors with uncertain inputs — application to multiple-step ahead time series forecasting. In *Advances in Neural Information Processing Systems 15*, pages 529–536. 61, 102

Graves, A. (2011). Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems 25*, pages 2348–2356. 121, 124

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *29th Conference on Uncertainty in Artificial Intellegence*, pages 282–290. 15, 26, 37

Hensman, J. and Lawrence, N. D. (2014). Nested variational compression in deep Gaussian processes. *arXiv preprint arXiv:1412.1370*. 88, 91, 92, 94, 108

Hensman, J., Matthews, A. G. D. G., and Ghahramani, Z. (2015). Scalable variational Gaussian process classification. In *18th International Conference on Artificial Intelligence and Statistics*, pages 351–360. 9, 15, 26, 32, 149

Hernández-Lobato, D. and Hernández-Lobato, J. M. (2016). Scalable Gaussian process classification via expectation propagation. In *19th International Conference on Artificial Intelligence and Statistics*, pages 168–176. 9, 16, 26, 37

Hernández-Lobato, J. M. and Adams, R. P. (2015). Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *32nd International Conference on Machine Learning*, pages 1861–1869. 105, 120, 121

Hernández-Lobato, J. M., Li, Y., Rowland, M., Hernández-Lobato, D., Bui, T. D., and Turner, R. E. (2016). Black-box α-divergence minimization. In *33rd International Conference on International Conference on Machine Learning*, pages 1511–1520. 36, 38, 61, 70, 106

Heskes, T. and Zoeter, O. (2002). Expectation propagation for approximate inference in dynamic Bayesian networks. In *18th Conference on Uncertainty in Artificial Intelligence*, pages 216–223. 37, 69

Hjort, N. L., Holmes, C., Müller, P., and Walker, S. G. (2010). *Bayesian nonparametrics*, volume 28. Cambridge University Press. 3

Hoang, T. N., Hoang, Q. M., and Low, B. K. H. (2016). A distributed variational inference framework for unifying parallel sparse Gaussian process regression models. In *33rd International Conference on Machine Learning*, pages 382–391. 37

Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544. 78

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. (2011). Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*. 9, 85, 126

Ito, K. and Xiong, K. (2000). Gaussian filters for nonlinear filtering problems. *IEEE Transactions on Automatic Control*, 45(5):910–927. 45

Jaynes, E. T. (2003). *Probability theory: The logic of science*. Cambridge University Press. 2

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233. 2

Kingma, D. P. and Ba, J. (2015). Adam: a method for stochastic optimization. In *3rd International Conference on Learning Representations*. 26, 32, 74, 111, 120

Kingma, D. P. and Welling, M. (2014). Stochastic gradient VB and the variational auto-encoder. In *2nd International Conference on Learning Representations*. 61, 102, 121

Krinskiĭ, V. and Kokoz, I. (1973). Analysis of the equations of excitable membranes. I. Reduction of the Hodgkins-Huxley equations to a 2d order system. *Biofizika*, 18(3):506. 80

Kushner, H. J. and Budhiraja, A. S. (2000). A nonlinear filtering algorithm based on an approximation of the conditional distribution. *IEEE Transactions on Automatic Control*, 45(3):580–585. 45

Lawrence, N. D. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *The Journal of Machine Learning Research*, 6:1783–1816. 9, 71, 72

Lawrence, N. D. and Moore, A. J. (2007). Hierarchical Gaussian process latent variable models. In *24th International Conference on Machine Learning*, pages 481–488. 87, 89, 91

Lawrence, N. D. and Quiñonero-Candela, J. (2006). Local distance preservation in the gp-lvm through back constraints. In *23rd International Conference on Machine learning*, pages 513–520. 72

Lázaro-Gredilla, M. (2012). Bayesian warped Gaussian processes. In *Advances in Neural Information Processing Systems 25*, pages 1619–1627. 92

Li, Y., Hernández-Lobato, J. M., and Turner, R. E. (2015). Stochastic expectation propagation. In *Advances in Neural Information Processing Systems 29*, pages 2323–2331. 20, 38, 70, 106

Low, K. H., Yu, J., Chen, J., and Jaillet, P. (2015). Parallel Gaussian process regression for big data: Low-rank representation meets Markov approximation. In *29th AAAI Conference on Artificial Intelligence*, pages 2821–2827. 37

MacKay, D. J. C. (1992). Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604. 85

MacKay, D. J. C. (2003). *Information theory, inference and learning algorithms*. Cambridge University Press. 2

Mahsereci, M. and Hennig, P. (2015). Probabilistic line searches for stochastic optimization. In *Advances in Neural Information Processing Systems 28*, pages 181–189. 9

Matthews, A. G. D. G., Hensman, J., Turner, R. E., and Ghahramani, Z. (2016). On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. In *19th International Conference on Artificial Intelligence and Statistics*, pages 231–239. 9, 14, 15, 17

McHutchon, A. (2014). *Nonlinear modelling and control using Gaussian processes*. PhD thesis, University of Cambridge. 9, 15, 46, 47, 48, 50, 51, 52, 55, 69, 94

Minka, T. P. (2001a). The EP energy function and minimization schemes. Technical report, Massachusetts Institute of Technology. 66

Minka, T. P. (2001b). *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology. 2, 16

Minka, T. P. (2004). Power EP. Technical report, Microsoft Research Cambridge. 2, 9, 17

Minka, T. P. (2005). Divergence measures and message passing. Technical report, Microsoft Research Cambridge. 22, 36

Minka, T. P. and Lafferty, J. (2002). Expectation-propagation for the generative aspect model. In *18th Conference on Uncertainty in Artificial Intelligence*, pages 352–359. 38, 70

Naish-Guzman, A. and Holden, S. B. (2007). The generalized FITC approximation. In *Advances in Neural Information Processing Systems 20*, pages 1057–1064. 9, 16

Neal, R. M. (1993). Bayesian learning via stochastic dynamics. In *Advances in Neural Information Processing Systems 6*, pages 475–482. 122

Neal, R. M. (1995). *Bayesian learning for neural networks*. PhD thesis, University of Toronto. 87

Neal, R. M. (2011). MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11). 2

Nickisch, H. and Rasmussen, C. E. (2008). Approximations for binary Gaussian process classification. *The Journal of Machine Learning Research*, 9(Oct):2035–2078. 16

Orbanz, P. and Teh, Y. W. (2011). Bayesian nonparametric models. In *Encyclopedia of Machine Learning*, pages 81–89. 3

Qi, Y., Abdel-Gawad, A. H., and Minka, T. P. (2010). Sparse-posterior Gaussian processes for general likelihoods. In *26th Conference on Uncertainty in Artificial Intelligence*, pages 450–457. 9, 16, 139, 143

Quiñonero-Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959. 6, 9, 10, 11, 12, 14

Ranganath, R., Tang, L., Charlin, L., and Blei, D. (2015). Deep exponential families. In *18th International Conference on Artificial Intelligence and Statistics*, pages 762–771. 126

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. The MIT Press. 3, 4, 6, 9, 27

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *31st International Conference on Machine Learning*, pages 1278–1286. 61

Riihimäki, J., Jylänki, P., and Vehtari, A. (2013). Nested expectation propagation for Gaussian process classification with a multinomial probit likelihood. *The Journal of Machine Learning Research*, 14(Jan):75–109. 26

Salimans, T. and Knowles, D. A. (2013). Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882. 61, 102

Salimbeni, H. and Deisenroth, M. P. (2017). Doubly stochastic variational inference for deep Gaussian processes. *arXiv preprint arXiv:1705.08933*. 47, 82, 88, 91, 92, 94, 98, 99, 101, 105

Saul, A. D., Hensman, J., Vehtari, A., and Lawrence, N. D. (2016). Chained Gaussian processes. In *19th International Conference on Artificial Intelligence and Statistics*, pages 1431–1440. 37

Schwaighofer, A. and Tresp, V. (2002). Transductive and inductive methods for approximate Gaussian process regression. In *Advances in Neural Information Processing Systems 15*, pages 953–960. 9, 12

Seeger, M. (2008). Bayesian inference and optimal design for the sparse linear model. *The Journal of Machine Learning Research*, 9(Apr):759–813. 21

Seeger, M. and Jordan, M. I. (2004). Sparse Gaussian process classification with multiple classes. Technical report, Department of Statistics, University of Berkeley, CA. 26

Seeger, M., Williams, C., and Lawrence, N. D. (2003). Fast forward selection to speed up sparse Gaussian process regression. In *9th International Conference on Artificial Intelligence and Statistics*. 9, 14

Snelson, E. (2007). *Flexible and efficient Gaussian process models for machine learning*. PhD thesis, University College London. 9, 16, 37

Snelson, E. and Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 19*, pages 1257–1264. 9, 13, 92

Snelson, E., Rasmussen, C. E., and Ghahramani, Z. (2004). Warped Gaussian processes. In *Advances in Neural Information Processing Systems 17*, pages 337–344, Cambridge, MA, USA. 88, 92

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25*, pages 2951–2959. 9

Sutton, R. S. and Barto, A. G. (1998). *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition. 89

Titsias, M. K. (2009). Variational learning of inducing variables in sparse Gaussian processes. In *12th International Conference on Artificial Intelligence and Statistics*, pages 567–574. 9, 13, 14, 24, 25, 37, 46, 48, 91, 92, 97, 147, 149

Titsias, M. K. and Lawrence, N. D. (2010). Bayesian Gaussian process latent variable model. In *13th International Conference on Artificial Intelligence and Statistics*, pages 844–851. 15, 48, 72, 101, 102

Tobar, F., Bui, T. D., and Turner, R. E. (2015). Learning stationary time series using Gaussian processes with nonparametric kernels. In *Advances in Neural Information Processing Systems 29*, pages 3501–3509. 15

Turner, R. E. and Sahani, M. (2011). Two problems with variational expectation maximisation for time-series models. In Barber, D., Cemgil, T., and Chiappa, S., editors, *Bayesian Time series models*, chapter 5, pages 109–130. Cambridge University Press. 36, 54

Valpola, H. and Karhunen, J. (2002). An unsupervised ensemble learning method for nonlinear dynamic state-space models. *Neural computation*, 14(11):2647–2692. 45

Vanhatalo, J. and Vehtari, A. (2006). MCMC methods for MLP-network and Gaussian process and stuff–a documentation for Matlab toolbox MCMCstuff. Technical report. 122

Wainwright, M. J. and Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305. 2

Wang, J. M., Fleet, D. J., and Hertzmann, A. (2005). Gaussian process dynamical models. In *Advances in Neural Information Processing Systems 18*, pages 1441–1448. 9, 46, 47

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *28th International Conference on Machine Learning*, pages 681–688. 121

Wilson, A. and Adams, R. P. (2013). Gaussian process kernels for pattern discovery and extrapolation. In *30th International Conference on Machine Learning*, pages 1067–1075. 61, 102

Xu, M., Lakshminarayanan, B., Teh, Y. W., Zhu, J., and Zhang, B. (2014). Distributed Bayesian posterior sampling via moment sharing. In *Advances in Neural Information Processing Systems 27*, pages 3356–3364. 25, 26

Zhu, H. and Rohwer, R. (1995). Information geometric measurements of generalisation. Technical report, Aston University. 22

Zhu, H. and Rohwer, R. (1997). Measurements of generalisation based on information geometry. In *Mathematics of Neural Networks*, pages 394–398. 18

Appendix A

Derivations for Chapter 2

A.1 A unified objective for unnormalised KL variational free-energy methods

Here we show that performing variational inference by optimising the unnormalised KL naturally leads to a single objective for both the approximation to the joint distribution, $q^*(f|\theta)$ and the hyperparameters θ.

The unnormalised KL is given by

$$\overline{\mathrm{KL}}(q^*(f|\theta)||p(f,\mathbf{y}|\theta)) = \int q^*(f|\theta)\log\frac{q^*(f|\theta)}{p(f,\mathbf{y}|\theta)}\mathrm{d}f + \int \left(p(f,\mathbf{y}|\theta) - q^*(f|\theta)\right)\mathrm{d}f. \tag{A.1}$$

This is intractable as it includes the marginal likelihood $p(\mathbf{y}|\theta) = \int p(f,\mathbf{y}|\theta)\mathrm{d}f$. However, since we are interested in minimising this objective with respect to $q^*(f|\theta)$ we can ignore the intractable term,

$$\operatorname*{argmin}_{q^*(f|\theta)} \overline{\mathrm{KL}}(q^*(f|\theta)||p(f,\mathbf{y}|\theta)) = \operatorname*{argmax}_{q^*(f|\theta)} \left(p(\mathbf{y}|\theta) - \overline{\mathrm{KL}}(q^*(f|\theta)||p(f,\mathbf{y}|\theta))\right) \tag{A.2}$$

$$= \operatorname*{argmax}_{q^*(f|\theta)} \left(\int q^*(f|\theta)\log\frac{p(f,\mathbf{y}|\theta)}{q^*(f|\theta)}\mathrm{d}f + \int q^*(f|\theta)\mathrm{d}f\right). \tag{A.3}$$

In other words, we have turned the unnormalised KL into a tractable lower-bound of the marginal likelihood $\mathcal{G}(q^*(f|\theta),\theta) = p(\mathbf{y}|\theta) - \overline{\mathrm{KL}}(q^*(f|\theta)||p(f,\mathbf{y}|\theta))$. The structure of this new lower-bound can be understood by decomposing the approximation to the joint distribution into a normalised posterior approximation $q(f|\theta)$ and an approximation to the marginal likelihood, Z_{VFE}, that is $q^*(f|\theta) = Z_{\mathrm{VFE}}q(f|\theta)$.

$$\mathcal{G}(Z_{\mathrm{VFE}}q(f|\theta),\theta) = Z_{\mathrm{VFE}}\left(1 - \log Z_{\mathrm{VFE}} + \int q(f|\theta)\log\frac{p(f,\mathbf{y}|\theta)}{q(f|\theta)}\mathrm{d}f\right) \tag{A.4}$$

We can see that optimising the lower-bound with respect to θ is equivalent to optimising the standard variational free-energy $\mathcal{F}(q(f|\theta), \theta) = \int q(f|\theta) \log \frac{p(f,\mathbf{y}|\theta)}{q(f|\theta)} \mathrm{d}f$. Moreover, optimising for Z_{VFE} recovers $Z_{\mathrm{VFE}}^{\mathrm{opt}} = \exp(\mathcal{F}(q(f|\theta), \theta))$. Substituting this back into the bound

$$\mathcal{G}(Z_{\mathrm{VFE}}^{\mathrm{opt}} q(f|\theta), \theta) = Z_{\mathrm{VFE}}^{\mathrm{opt}} = \exp(\mathcal{F}(q(f|\theta), \theta)). \tag{A.5}$$

In other words, the new collapsed bound is just the exponential of the original variational free-energy and optimising the collapsed bound for θ is equivalent to optimising the approximation to the marginal likelihood.

A.2 Global and local inclusive KL minimisations

In this section, we will show that optimising a single global inclusive KL-divergence, $\mathrm{KL}(q||p)$, is equivalent to optimising a sum of a set of local inclusive KL-divergence, $\mathrm{KL}(q||\tilde{p})$, where p, q and $\tilde{p}$ are the exact posterior, the approximate posterior and the tilted distribution accordingly. Without loss of generality, we assume that $p(\theta) = \prod_n f_n(\theta) \approx \prod_n t_n(\theta) = q(\theta)$, that is the exact posterior is a product of factors, $\{f_n(\theta)\}_n$, each of which is approximated by an approximate factor $t_n(\theta)$. Substituting these distributions into the global KL-divergence gives,

$$\begin{aligned}
\mathrm{KL}(q(\theta)||p(\theta)) &= \int \mathrm{d}\theta q(\theta) \log \frac{q(\theta)}{p(\theta)} \\
&= \int \mathrm{d}\theta q(\theta) \log \frac{\prod_n t_n(\theta)}{\prod_n f_n(\theta)} \\
&= \int \mathrm{d}\theta q(\theta) \log \left[\frac{\prod_n t_n(\theta)}{\prod_n f_n(\theta)} \frac{\prod_n \prod_{i \neq n} t_i(\theta)}{\prod_n \prod_{i \neq n} t_i(\theta)} \right] \\
&= \int \mathrm{d}\theta q(\theta) \log \frac{\prod_n [\prod_i t_i(\theta)]}{\prod_n [f_n(\theta) \prod_{i \neq n} t_i(\theta)]} \\
&= \sum_n \int \mathrm{d}\theta q(\theta) \log \frac{\prod_i t_i(\theta)}{[f_n(\theta) \prod_{i \neq n} t_i(\theta)} \\
&= \sum_n \mathrm{KL}(q(\theta)||\tilde{p}_n(\theta)),
\end{aligned} \tag{A.6}$$

which means running the EP procedure, where we use $\mathrm{KL}(q(\theta)||\tilde{p}_n(\theta))$ in place of $\mathrm{KL}(\tilde{p}_n(\theta)||q(\theta))$, is *equivalent* to the VFE approach which optimises a single global KL-divergence, $\mathrm{KL}(q(\theta)||p(\theta))$.

A.3 Some relevant linear algebra and function expansion identities

The Woodbury matrix identity or Woodbury formula is:

$$(A+UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1}+VA^{-1}U)^{-1}VA^{-1}. \tag{A.7}$$

In general, C need not be invertible, we can use the Binomial inverse theorem,

$$(A+UCV)^{-1} = A^{-1} - A^{-1}UC(C+CVA^{-1}UC)^{-1}CVA^{-1}. \tag{A.8}$$

When C is an identity matrix and U and V are vectors, the Woodbury identity can be shortened and become the Sherman-Morrison formula,

$$(A+uv^\intercal)^{-1} = A^{-1} - \frac{A^{-1}uv^\intercal A^{-1}}{1+v^\intercal A^{-1}u}. \tag{A.9}$$

Another useful identity is the matrix determinant lemma,

$$\det(A+uv^\intercal) = (1+v^\intercal A^{-1}u)\det(A). \tag{A.10}$$

The above theorem can be extend for matrices U and V,

$$\det(A+UV^\intercal) = \det(\mathrm{I}+V^\intercal A^{-1}U)\det(A). \tag{A.11}$$

We also make use of the following Maclaurin series,

$$\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots \tag{A.12}$$

$$\text{and} \quad \log(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} + \cdots . \tag{A.13}$$

A.4 KL minimisation between Gaussian processes and moment matching

The difficult step of Power EP is the projection step, that is how to find the posterior approximation $q(f)$ that minimises the KL divergence, $\mathrm{KL}(\tilde{p}(f)||q(f))$, where $\tilde{p}(f)$ is the tilted distribution. We have chosen the form of the approximate posterior

$$q(f) = p(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u}) = p(f_{\neq \mathbf{u}}|\mathbf{u})\frac{\exp(\theta_{\mathbf{u}}^\intercal \phi(\mathbf{u}))}{\mathcal{Z}(\theta_{\mathbf{u}})}, \tag{A.14}$$

where $\mathcal{Z}(\theta_{\mathbf{u}}) = \int \exp(\theta_{\mathbf{u}}^{\intercal}\phi(\mathbf{u}))\mathrm{d}\mathbf{u}$ to ensure normalisation. We can then write the KL minimisation objective as follows,

$$\begin{aligned}
\mathcal{F}_{\mathrm{KL}} &= \mathrm{KL}(\tilde{p}(f)||q(f)) && \text{(A.15)}\\
&= \int \tilde{p}(f)\log\frac{\tilde{p}(f)}{q(f)}\mathrm{d}f && \text{(A.16)}\\
&= \langle\log\tilde{p}(f)\rangle_{\tilde{p}(f)} - \langle\log p(f_{\neq\mathbf{u}}|\mathbf{u})\rangle_{\tilde{p}(f)} - \theta_{\mathbf{u}}^{\intercal}\langle\phi(\mathbf{u})\rangle_{\tilde{p}(f)} + \log\mathcal{Z}(\theta_{\mathbf{u}}). && \text{(A.17)}
\end{aligned}$$

Since $p(f_{\neq\mathbf{u}}|\mathbf{u})$ is the prior conditional distribution, the only free parameter that controls our posterior approximation is $\theta_{\mathbf{u}}$. As such, to find $\theta_{\mathbf{u}}$ that minimises F_{KL}, we find the gradient of F_{KL} w.r.t $\theta_{\mathbf{u}}$ and set it to zero,

$$\begin{aligned}
0 = \frac{\mathrm{d}\mathcal{F}_{\mathrm{KL}}}{\mathrm{d}\theta_{\mathbf{u}}} &= -\langle\phi(\mathbf{u})\rangle_{\tilde{p}(f)} + \frac{\mathrm{d}\log\mathcal{Z}(\theta_{\mathbf{u}})}{\mathrm{d}\theta_{\mathbf{u}}} && \text{(A.18)}\\
&= -\langle\phi(\mathbf{u})\rangle_{\tilde{p}(f)} + \langle\phi(\mathbf{u})\rangle_{q(u)}, && \text{(A.19)}
\end{aligned}$$

therefore, $\langle\phi(\mathbf{u})\rangle_{\tilde{p}(f)} = \langle\phi(\mathbf{u})\rangle_{q(u)}$. That is, though we are trying to perform the KL minimisation between two Gaussian processes, due to the special form of the posterior approximation, *it is sufficient to only match the moments at the pseudo-points* $\mathbf{u}$.[1]

A.5 Shortcuts to the moment matching equations

The most crucial step in Power EP is the moment matching step as discussed above. This step can be done analytically for the Gaussian case, as the mean and covariance of the approximate posterior can be linked to the cavity distribution as follows,

$$\mathbf{m}_{\mathbf{u}} = \mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}f}^{\backslash n}\frac{\mathrm{d}\log\mathcal{Z}_{\mathrm{tilted},n}}{\mathrm{d}\mathbf{m}_f^{\backslash n}}, \tag{A.20}$$

$$\mathbf{V}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}f}^{\backslash n}\frac{\mathrm{d}^2\log\mathcal{Z}_{\mathrm{tilted},n}}{\mathrm{d}\mathbf{m}_f^{\backslash n,2}}\mathbf{V}_{f\mathbf{u}}^{\backslash n}, \tag{A.21}$$

where $\mathcal{Z}_{\mathrm{tilted},n}$ is the normaliser of the tilted distribution,

$$\begin{aligned}
\mathcal{Z}_{\mathrm{tilted},n} &= \int q^{\backslash n}(f)p(y_n|f)\mathrm{d}f && \text{(A.22)}\\
&= \int q^{\backslash n}(f)p(y_n|f_n)\mathrm{d}f && \text{(A.23)}\\
&= \int q^{\backslash n}(f_n)p(y_n|f_n)\mathrm{d}f_n. && \text{(A.24)}
\end{aligned}$$

[1] We can show that this condition gives the minimum of $\mathcal{F}_{\mathrm{KL}}$ by computing the second derivative.

In words, $\mathcal{Z}_{\text{tilted},n}$ only depends on the marginal distribution of the cavity process, $q^{\backslash n}(f_n)$, simplifying the moment matching equations above,

$$\mathbf{m}_{\mathbf{u}} = \mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}f_n}^{\backslash n} \frac{\mathrm{d} \log \mathcal{Z}_{\text{tilted},n}}{\mathrm{d} m_{f_n}^{\backslash n}}, \tag{A.25}$$

$$\mathbf{V}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}f_n}^{\backslash n} \frac{\mathrm{d}^2 \log \mathcal{Z}_{\text{tilted},n}}{\mathrm{d} m_{f_n}^{\backslash n,2}} \mathbf{V}_{f_n\mathbf{u}}^{\backslash n}. \tag{A.26}$$

We can rewrite the cross-covariance $\mathbf{V}_{\mathbf{u}f_n}^{\backslash n} = \mathbf{V}_{\mathbf{u}}^{\backslash n} \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{K}_{\mathbf{u}f_n}$. We also note that, $m_{f_n}^{\backslash n} = \mathbf{K}_{f_n\mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{m}_{\mathbf{u}}^{\backslash n}$, resulting in,

$$\frac{\mathrm{d} \log \mathcal{Z}_{\text{tilted},n}}{\mathrm{d} \mathbf{m}_{\mathbf{u}}^{\backslash n}} = \frac{\mathrm{d} \log \mathcal{Z}_{\text{tilted},n}}{\mathrm{d} m_{f_n}^{\backslash n}} \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{K}_{\mathbf{u}f_n}, \tag{A.27}$$

$$\frac{\mathrm{d} \log \mathcal{Z}_{\text{tilted},n}}{\mathrm{d} \mathbf{V}_{\mathbf{u}}^{\backslash n}} = \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{K}_{\mathbf{u}f_n} \frac{\mathrm{d}^2 \log \mathcal{Z}_{\text{tilted},n}}{\mathrm{d} m_{f_n}^{\backslash n,2}} \mathbf{K}_{f_n\mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1}. \tag{A.28}$$

Substituting these results back in eqs. A.25 and A.26, we obtain

$$\mathbf{m}_{\mathbf{u}} = \mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}}^{\backslash n} \frac{\mathrm{d} \log \mathcal{Z}_{\text{tilted},n}}{\mathrm{d} \mathbf{m}_{\mathbf{u}}^{\backslash n}}, \tag{A.29}$$

$$\mathbf{V}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}}^{\backslash n} \frac{\mathrm{d}^2 \log \mathcal{Z}_{\text{tilted},n}}{\mathrm{d} \mathbf{m}_{\mathbf{u}}^{\backslash n,2}} \mathbf{V}_{\mathbf{u}}^{\backslash n}. \tag{A.30}$$

Therefore, using eqs. A.25 and A.26, or eqs. A.29 and A.30 are equivalent in our approximation settings.

A.6 Full derivation of the Power EP procedure

We provide the full derivation of the Power EP procedure in this section. We follow the derivation in (Qi et al., 2010) closely, but provide a clearer exposition and details how to get to each step used in the implementation, and how to handle powered/fractional deletion and update in Power EP.

A.6.1 Optimal factor parameterisation

We start by defining the approximate factors to be in natural parameter form as this makes it simple to combine and delete them, $t_n(\mathbf{u}) = \tilde{\mathcal{N}}(\mathbf{u}; z_n, \mathbf{T}_{1,n}, \mathbf{T}_{2,n}) = z_n \exp(\mathbf{u}^\intercal \mathbf{T}_{1,n} - \frac{1}{2}\mathbf{u}^\intercal \mathbf{T}_{2,n}\mathbf{u})$. We initially consider full rank $\mathbf{T}_{2,n}$, but will show that the optimal form is rank 1.

The next goal is to relate these parameters to the approximate GP posterior. The approximate posterior over the pseudo-outputs has natural parameters $\mathbf{T}_{1,\mathbf{u}} = \sum_n \mathbf{T}_{1,n}$ and

$\mathbf{T}_{2,\mathbf{u}} = \mathbf{K}_{\mathbf{uu}}^{-1} + \sum_n \mathbf{T}_{2,n}$. This induces an approximate GP posterior with mean and covariance function,

$$m_{\mathrm{f}} = \mathbf{K}_{\mathrm{f}\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}_{2,\mathbf{u}}^{-1}\mathbf{T}_{1,\mathbf{u}} = \mathbf{K}_{\mathrm{f}\mathbf{u}}\gamma \tag{A.31}$$

$$V_{\mathrm{ff}'} = \mathbf{K}_{\mathrm{ff}'} - \mathbf{Q}_{\mathrm{ff}'} + \mathbf{K}_{\mathrm{f}\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}_{2,\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}\mathrm{f}'} = \mathbf{K}_{\mathrm{ff}'} - \mathbf{K}_{\mathrm{f}\mathbf{u}}\beta\mathbf{K}_{\mathbf{u}\mathrm{f}'}. \tag{A.32}$$

where γ and β are likelihood-dependent terms we wish to store and update using PEP; γ and β fully specify the approximate posterior.

Deletion step: The cavity for data point n, $q^{\backslash n}(f) \propto q^*(f)/t_n^\alpha(\mathbf{u})$, has a similar form to the posterior, but the natural parameters are modified by the deletion, $\mathbf{T}_{1,\mathbf{u}}^{\backslash n} = \mathbf{T}_{1,\mathbf{u}} - \alpha\mathbf{T}_{1,n}$ and $\mathbf{T}_{2,\mathbf{u}}^{\backslash n} = \mathbf{T}_{2,\mathbf{u}} - \alpha\mathbf{T}_{2,n}$, yielding a new mean and covariance function

$$m_{\mathrm{f}}^{\backslash n} = \mathbf{K}_{\mathrm{f}\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}_{2,\mathbf{u}}^{\backslash n,-1}\mathbf{T}_{1,\mathbf{u}}^{\backslash n} = \mathbf{K}_{\mathrm{f}\mathbf{u}}\gamma^{\backslash n} \tag{A.33}$$

$$V_{\mathrm{ff}'}^{\backslash n} = \mathbf{K}_{\mathrm{ff}'} - \mathbf{Q}_{\mathrm{ff}'} + \mathbf{K}_{\mathrm{f}\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}\backslash n, -1_{2,\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}\mathrm{f}'} = \mathbf{K}_{\mathrm{ff}'} - \mathbf{K}_{\mathrm{f}\mathbf{u}}\beta^{\backslash n}\mathbf{K}_{\mathbf{u}\mathrm{f}'}. \tag{A.34}$$

Projection step: The central step in Power EP is the projection step. Obtaining the new approximate unormalised posterior $q^*(f)$ such that $\mathrm{KL}(\tilde{p}(f)||q^*(f))$ is minimised would naïvely appear intractable. Fortunately, as shown in the previous section, because of the structure of the approximate posterior, $q(f) = p(f_{\neq\mathbf{u}}|\mathbf{u})q(\mathbf{u})$, the objective, $\mathrm{KL}(\tilde{p}(f)||q^*(f))$ is minimised when $\mathbb{E}_{\tilde{p}(f)}[\phi(\mathbf{u})] = \mathbb{E}_{q(u)}[\phi(\mathbf{u})]$, where $\phi(\mathbf{u})$ are the sufficient statistics, that is when the moments at the pseudo-inputs are matched. This is the central result from which computational savings are derived. Furthermore, this moment matching condition would appear to necessitate computation of a set of integrals to find the zeroth, first and second moments. Using results from the previous section simplifies and provides the following shortcuts,

$$\mathbf{m}_{\mathbf{u}} = \mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}\frac{\mathrm{d}\log\tilde{Z}_n}{\mathrm{d}m_{\mathrm{f}_n}^{\backslash n}} \tag{A.35}$$

$$\mathbf{V}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}\frac{\mathrm{d}^2\log\tilde{Z}_n}{\mathrm{d}(m_{\mathrm{f}_n}^{\backslash n})^2}\mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}. \tag{A.36}$$

where $\log\tilde{Z}_n = \log\mathbb{E}_{q^{\backslash n}(f)}[p^\alpha(y_n|\mathrm{f}_n)]$ is the log-normaliser of the tilted distribution.

Update step: Having computed the new approximate posterior, the fractional approximate factor $t_{n,\mathrm{new}}(\mathbf{u}) = q^*(f)/q^{\backslash n}(f)$ can be straightforwardly obtained, resulting in,

$$\mathbf{T}_{1,n,\mathrm{new}} = \mathbf{V}_{\mathbf{u}}^{-1}\mathbf{m}_{\mathbf{u}} - \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{m}_{\mathbf{u}}^{\backslash n} \tag{A.37}$$

$$\mathbf{T}_{2,n,\mathrm{new}} = \mathbf{V}_{\mathbf{u}}^{-1} - \mathbf{V}_{\mathbf{u}}^{\backslash n,-1} \tag{A.38}$$

$$z_n^\alpha = \tilde{Z}_n\exp(\mathcal{G}_{q_*^{\backslash n}(\mathbf{u})} - \mathcal{G}_{q^*(\mathbf{u})}), \tag{A.39}$$

where $\mathcal{G}_{\tilde{\mathcal{N}}(\mathbf{u};z,\mathbf{T}_1,\mathbf{T}_2)} = \int \tilde{\mathcal{N}}(\mathbf{u};z,\mathbf{T}_1,\mathbf{T}_2)\mathrm{d}\mathbf{u}$. Let $d_1 = \frac{\mathrm{d}\log \tilde{Z}_n}{\mathrm{d}m_{\mathrm{f}_n}^{\backslash n}}$ and $d_2 = \frac{\mathrm{d}^2\log \tilde{Z}_n}{\mathrm{d}(m_{\mathrm{f}_n}^{\backslash n})^2}$. Using eq. (A.7) and eq. (A.36), we have,

$$\mathbf{V}_{\mathbf{u}}^{-1} - \mathbf{V}_{\mathbf{u}}^{\backslash n,-1} = -\mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}\left[d_2^{-1} + \mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}\mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}\right]^{-1}\mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}\mathbf{V}_{\mathbf{u}}^{\backslash n,-1} \tag{A.40}$$

Let $v_n = \alpha(-d_2^{-1} - \mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}\mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n})$, and $\mathbf{w}_n = \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}$. Combining eq. (A.40) and eq. (A.38) gives

$$\mathbf{T}_{2,n,\mathrm{new}} = \mathbf{w}_n\alpha v_n^{-1}\mathbf{w}_n^\intercal \tag{A.41}$$

At convergence, we have $t_n(\mathbf{u})^\alpha = t_{n,\mathrm{new}}(\mathbf{u})$, hence $\mathbf{T}_{2,n} = \mathbf{w}_n v_n^{-1}\mathbf{w}_n^\intercal$. In words, $\mathbf{T}_{2,n}$ is optimally a rank-1 matrix. Note that,

$$\begin{aligned}
\mathbf{w}_n &= \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n} && \text{(A.42)}\\
&= (\mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uu}}\beta^{\backslash n}\mathbf{K}_{\mathbf{uu}})^{-1}(\mathbf{K}_{\mathbf{u}\mathrm{f}_n} - \mathbf{K}_{\mathbf{uu}}\beta^{\backslash n}\mathbf{K}_{\mathbf{u}\mathrm{f}_n}) && \text{(A.43)}\\
&= \mathbf{K}_{\mathbf{uu}}^{-1}(\mathrm{I} - \mathbf{K}_{\mathbf{uu}}\beta^{\backslash n})^{-1}(\mathrm{I} - \mathbf{K}_{\mathbf{uu}}\beta^{\backslash n})\mathbf{K}_{\mathbf{u}\mathrm{f}_n} && \text{(A.44)}\\
&= \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}\mathrm{f}_n}. && \text{(A.45)}
\end{aligned}$$

Using eq. (A.35) an eq. (A.41) gives,

$$\begin{aligned}
\mathbf{V}_{\mathbf{u}}^{-1}\mathbf{m}_{\mathbf{u}} &= (\mathbf{V}_{\mathbf{u}}^{\backslash n,-1} + \mathbf{w}_n\alpha v_n^{-1}\mathbf{w}_n^\intercal)(\mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1) && \text{(A.46)}\\
&= \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{w}_n\alpha v_n^{-1}\mathbf{w}_n^\intercal\mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1 + \mathbf{w}_n\alpha v_n^{-1}\mathbf{w}_n^\intercal\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1 && \text{(A.47)}
\end{aligned}$$

Substituting this result into eq. (A.37),

$$\begin{aligned}
\mathbf{T}_{1,n,\mathrm{new}} &= \mathbf{V}_{\mathbf{u}}^{-1}\mathbf{m}_{\mathbf{u}} - \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{m}_{\mathbf{u}}^{\backslash n} && \text{(A.48)}\\
&= \mathbf{w}_n\alpha v_n^{-1}\mathbf{w}_n^\intercal\mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1 + \mathbf{w}_n\alpha v_n^{-1}\mathbf{w}_n^\intercal\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1 && \text{(A.49)}\\
&= \mathbf{w}_n\alpha v_n^{-1}\left(\mathbf{w}_n^\intercal\mathbf{m}_{\mathbf{u}}^{\backslash n} + d_1 v_n/\alpha + \mathbf{w}_n^\intercal\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1\right). && \text{(A.50)}
\end{aligned}$$

Let $\mathbf{T}_{1,n,\mathrm{new}} = \mathbf{w}_n\alpha v_n^{-1}g_n$, we obtain,

$$g_n = -\frac{d_1}{d_2} + \mathbf{K}_{\mathrm{f}_n\mathbf{u}}\gamma^{\backslash n}. \tag{A.51}$$

At convergence, $\mathbf{T}_{1,n} = \mathbf{w}_n v_n^{-1} g_n$. Re-writing the form of the approximate factor using $\mathbf{T}_{1,n}$ and $\mathbf{T}_{2,n}$ at convergence,

$$t_n(\mathbf{u}) = \tilde{\mathcal{N}}(\mathbf{u}; z_n, \mathbf{T}_{1,n}, \mathbf{T}_{2,n}) \tag{A.52}$$

$$= z_n \exp(\mathbf{u}^\intercal \mathbf{T}_{1,n} - \frac{1}{2}\mathbf{u}^\intercal \mathbf{T}_{2,n}\mathbf{u}) \tag{A.53}$$

$$= z_n \exp(\mathbf{u}^\intercal \mathbf{w}_n v_n^{-1} g_n - \frac{1}{2}\mathbf{u}^\intercal \mathbf{w}_n v_n^{-1} \mathbf{w}_n^\intercal \mathbf{u}) \tag{A.54}$$

As a result, the minimal and simplest way to parameterise the approximate factor is $t_n(\mathbf{u}) = \tilde{z}_n \mathcal{N}(\mathbf{w}_n^\intercal \mathbf{u}; g_n, v_n) = \tilde{z}_n \mathcal{N}(\mathbf{K}_{\mathrm{f}_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u}; g_n, v_n)$, where g_n and v_n are scalars, resulting in a significant memory saving compared to the parameterisation using $\mathbf{T}_{1,n}$ and $\mathbf{T}_{2,n}$.

A.6.2 Projection

We now recall the update equations in the projection step (eqns. A.35 and A.36):

$$\mathbf{m}_\mathbf{u} = \mathbf{m}_\mathbf{u}^{\setminus n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\setminus n} d_1, \tag{A.55}$$

$$\mathbf{V}_\mathbf{u} = \mathbf{V}_\mathbf{u}^{\setminus n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\setminus n} d_2 \mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\setminus n}. \tag{A.56}$$

Note that:

$$\mathbf{m}_\mathbf{u} = \mathbf{K}_{\mathbf{uu}}\gamma, \tag{A.57}$$

$$\mathbf{V}_\mathbf{u} = \mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uu}}\beta\mathbf{K}_{\mathbf{uu}}, \tag{A.58}$$

and

$$\mathbf{m}_\mathbf{u}^{\setminus n} = \mathbf{K}_{\mathbf{uu}}\gamma^{\setminus n}, \tag{A.59}$$

$$\mathbf{V}_\mathbf{u}^{\setminus n} = \mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uu}}\beta^{\setminus n}\mathbf{K}_{\mathbf{uu}}. \tag{A.60}$$

Using these results, we can convert the update for the mean and covariance, $\mathbf{m}_\mathbf{u}$ and $\mathbf{V}_\mathbf{u}$, into an update for γ and β,

$$\gamma = \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m}_{\mathbf{u}} \tag{A.61}$$
$$= \mathbf{K}_{\mathbf{uu}}^{-1}(\mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{uf}_n}^{\backslash n} d_1) \tag{A.62}$$
$$= \gamma^{\backslash n} + \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{V}_{\mathbf{uf}_n}^{\backslash n} d_1, \text{ and} \tag{A.63}$$
$$\beta = \mathbf{K}_{\mathbf{uu}}^{-1}(\mathbf{K}_{\mathbf{uu}} - \mathbf{V}_{\mathbf{u}})\mathbf{K}_{\mathbf{uu}}^{-1} \tag{A.64}$$
$$= \mathbf{K}_{\mathbf{uu}}^{-1}(\mathbf{K}_{\mathbf{uu}} - \mathbf{V}_{\mathbf{u}}^{\backslash n} - \mathbf{V}_{\mathbf{uf}_n}^{\backslash n} d_2 \mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n})\mathbf{K}_{\mathbf{uu}}^{-1} \tag{A.65}$$
$$= \beta^{\backslash n} - \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{V}_{\mathbf{uf}_n}^{\backslash n} d_2 \mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}\mathbf{K}_{\mathbf{uu}}^{-1} \tag{A.66}$$

A.6.3 Deletion step

Finally, we present how deletion might be accomplished. One direct approach to this step is to divide out the cavity from the cavity, that is,

$$q^{\backslash n}(f) \propto \frac{q(f)}{t_n^{\alpha}(\mathbf{u})} = \frac{p(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u})}{t_n^{\alpha}(\mathbf{u})} = p(f_{\neq \mathbf{u}}|\mathbf{u})q^{\backslash n}(\mathbf{u}). \tag{A.67}$$

Instead, we use an alternative using the KL minimisation as used in (Qi et al., 2010), by realising that doing this will result in an identical outcome as the direct approach since the factor and distributions are Gaussian. Furthermore, we can re-use results from the projection and inclusion steps, by simply swapping the quantities and negating the site approximation variance. In particular, we present projection and deletion side-by-side, to facilitate the comparison,

$$\text{Projection:} \quad q(f) \approx q^{\backslash n}(f)p(y_n|\mathrm{f}_n) \tag{A.68}$$
$$\text{Deletion:} \quad q^{\backslash n}(f) \propto q(f)\frac{1}{t_n^{\alpha}(\mathbf{u})} \tag{A.69}$$

The projection step minimises the KL between the LHS and RHS while moment matching, to get $q(f)$. We would like to do the same for the deletion step to find $q^{\backslash n}(f)$, and thus reuse the same moment matching results for γ and β with some modifications.

Our task will be to reuse Equations A.63 and A.66, the moment matching equations in γ and β. We have two differences to account for. Firstly, we need to change any uses of the parameters of the cavity distribution to the parameters of the approximate posterior, $\mathbf{V}_{\mathbf{uf}_n}^{\backslash n}$ to $\mathbf{V}_{\mathbf{uf}_n}$, $\gamma^{\backslash n}$ to γ and $\beta^{\backslash n}$ to β. This is the equivalent of re-deriving the entire projection operation, while swapping the symbols (and quantities) for the cavity and the full distribution. Secondly, the derivatives d_1 and d_2 are different here, as

$$\log \tilde{Z}_n = \log \int q(f)\frac{1}{t_n^{\alpha}(\mathbf{u})}\mathrm{d}f \tag{A.70}$$

Now, we note

$$\frac{1}{t_n(\mathbf{u})} \propto \frac{1}{\mathcal{N}^\alpha(\mathbf{w}_n^\intercal \mathbf{u}; g_n, v_n)} \tag{A.71}$$

$$\propto \frac{1}{\exp\left(-\frac{\alpha}{2} v_n^{-1} \left(\mathbf{w}_n^\intercal \mathbf{u} - g_n\right)^2\right)} \tag{A.72}$$

$$= \exp\left(\frac{1}{2}\alpha v_n^{-1} \left(\mathbf{w}_n^\intercal \mathbf{u} - g_n\right)^2\right) \tag{A.73}$$

$$\propto \mathcal{N}(\mathbf{w}_n^\intercal \mathbf{u}; g_n, -v_n/\alpha) \tag{A.74}$$

Then we obtain the derivatives of $\log \tilde{Z}_n$

$$\tilde{d}_2 = \frac{\mathrm{d}^2 \log \tilde{Z}_n}{\mathrm{d}m_{\mathrm{f}_n}^2} = -\left[\mathbf{K}_{\mathrm{f}_n,\mathbf{u}} \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{K}_{\mathbf{u},\mathrm{f}_n} - \mathbf{K}_{\mathrm{f}_n,\mathbf{u}} \beta \mathbf{K}_{\mathbf{u},\mathrm{f}_n} - v_n/\alpha\right]^{-1} \tag{A.75}$$

$$\tilde{d}_1 = \frac{\mathrm{d} \log \tilde{Z}_n}{\mathrm{d}m_{\mathrm{f}_n}} = (\mathbf{K}_{\mathrm{f}_n,\mathbf{u}} \gamma - g_n)\tilde{d}_2 \tag{A.76}$$

Putting the above results together, we obtain,

$$\gamma^{\backslash n} = \gamma + \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}_{\mathbf{u}\mathrm{f}_n} \tilde{d}_1, \text{ and} \tag{A.77}$$

$$\beta^{\backslash n} = \beta - \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}_{\mathbf{u}\mathrm{f}_n} \tilde{d}_2 \mathbf{V}_{\mathrm{f}_n\mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} \tag{A.78}$$

A.6.4 Summary of the PEP procedure

We summarise here the key steps and equations that we have obtained, that are used in the implementation:

1. Initialise the parameters: $\{g_n = 0\}_{n=1}^N$, $\{v_n = \infty\}_{n=1}^N$, $\gamma = \mathbf{0}_{M\times 1}$ and $\beta = \mathbf{0}_{M\times M}$
2. Loop through all data points until convergence:
   (a) Deletion step: find $\gamma^{\backslash n}$ and $\beta^{\backslash n}$

$$\gamma^{\backslash n} = \gamma + \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}_{\mathbf{u}\mathrm{f}_n} \tilde{d}_1, \text{ and} \tag{A.79}$$

$$\beta^{\backslash n} = \beta - \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}_{\mathbf{u}\mathrm{f}_n} \tilde{d}_2 \mathbf{V}_{\mathrm{f}_n\mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} \tag{A.80}$$

   (b) Projection step: find γ and β

$$\gamma = \gamma^{\backslash n} + \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n} d_1, \tag{A.81}$$

$$\beta = \beta^{\backslash n} - \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n} d_2 \mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n} \mathbf{K}_{\mathbf{uu}}^{-1} \tag{A.82}$$

(c) Update step: find $g_{n,\text{new}}$ and $v_{n,\text{new}}$

$$g_{n,\text{new}} = -\frac{d_1}{d_2} + \mathbf{K}_{\mathrm{f}_n\mathbf{u}}\gamma^{\backslash n}, \tag{A.83}$$

$$v_{n,\text{new}} = -d_2^{-1} - \mathbf{V}^{\backslash n}_{\mathrm{f}_n\mathbf{u}}\mathbf{V}^{\backslash n,-1}_{\mathbf{u}}\mathbf{V}^{\backslash n}_{\mathbf{u}\mathrm{f}_n} \tag{A.84}$$

and parameters for the full factor,

$$v_n \leftarrow (v^{-1}_{n,\text{new}} + (1-\alpha)v_n^{-1})^{-1} \tag{A.85}$$

$$g_n \leftarrow v_n(g_{n,\text{new}}v^{-1}_{n,\text{new}} + (1-\alpha)g_n v_n^{-1}) \tag{A.86}$$

A.7 Power EP energy for sparse GP regression and classification

The Power EP procedure gives an approximate marginal likelihood, which is the negative Power EP energy, as follows,

$$\mathcal{F} = \mathcal{G}(q_*(\mathbf{u})) - \mathcal{G}(p_*(\mathbf{u})) + \frac{1}{\alpha}\sum_n \left[\log \mathcal{Z}_{\text{tilted},n} + \mathcal{G}(q_*^{\backslash n}(\mathbf{u})) - \mathcal{G}(q_*(\mathbf{u}))\right] \tag{A.87}$$

where $\mathcal{G}(q_*(\mathbf{u}))$ is the log-normaliser of the approximate posterior, that is,

$$\mathcal{G}(q_*(\mathbf{u})) = \log \int p(f_{\neq\mathbf{u}}|\mathbf{u}) \exp(\theta_{\mathbf{u}}^\intercal \phi(\mathbf{u}))\mathrm{d}f_{\neq\mathbf{u}}\mathrm{d}\mathbf{u} \tag{A.88}$$

$$= \log \int \exp(\theta_{\mathbf{u}}^\intercal \phi(\mathbf{u}))\mathrm{d}\mathbf{u} \tag{A.89}$$

$$= \frac{M}{2}\log(2\pi) + \frac{1}{2}\log|\mathbf{V}| + \frac{1}{2}\mathbf{m}^\intercal\mathbf{V}^{-1}\mathbf{m}, \tag{A.90}$$

where $\mathbf{m}$ and $\mathbf{V}$ are the mean and covariance of the posterior distribution over $\mathbf{u}$, respectively. Similarly,

$$\mathcal{G}(q_*^{\backslash n}(\mathbf{u})) = \frac{M}{2}\log(2\pi) + \frac{1}{2}\log|\mathbf{V}_{\text{cav},n}| + \frac{1}{2}\mathbf{m}^\intercal_{\text{cav},n}\mathbf{V}^{-1}_{\text{cav},n}\mathbf{m}_{\text{cav},n}, \tag{A.91}$$

$$\text{and } \mathcal{G}(p_*(\mathbf{u})) = \frac{M}{2}\log(2\pi) + \frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}|. \tag{A.92}$$

Finally, $\log \mathcal{Z}_{\text{tilted},n}$ is the log-normalising constant of the tilted distribution,

$$\log \mathcal{Z}_{\text{tilted}} = \log \int q_{\text{cav}}(f) p^{\alpha}(y_n|f) \mathrm{d}f \tag{A.93}$$

$$= \log \int p(f_{\neq \mathbf{u}}|\mathbf{u}) q_{\text{cav}}(\mathbf{u}) p^{\alpha}(y_n|f) \mathrm{d}f_{\neq \mathbf{u}} \mathrm{d}\mathbf{u} \tag{A.94}$$

$$= \log \int p(f_n|\mathbf{u}) q_{\text{cav}}(\mathbf{u}) p^{\alpha}(y_n|f_n) \mathrm{d}f_n \mathrm{d}\mathbf{u} \tag{A.95}$$

Next, we can write down the form of the natural parameters of the approximate posterior and the cavity distribution, based on the approximate factor's parameters, as follows,

$$\mathbf{V}^{-1} = \mathbf{K}_{\mathbf{uu}}^{-1} + \sum_i \mathbf{w}_i \tau_i \mathbf{w}_i^{\intercal} \tag{A.96}$$

$$\mathbf{V}^{-1}\mathbf{m} = \sum_i \mathbf{w}_i \tau_i \tilde{y}_i \tag{A.97}$$

$$\mathbf{V}_{\text{cav},n}^{-1} = \mathbf{V}^{-1} - \alpha \mathbf{w}_n \tau_n \mathbf{w}_n^{\intercal} \tag{A.98}$$

$$\mathbf{V}_{\text{cav},\mathbf{n}}{}^{-1}\mathbf{m}_{\text{cav},n} = \mathbf{V}^{-1}\mathbf{m} - \alpha \mathbf{w}_n \tau_n g_n \tag{A.99}$$

Note that $\tau_i \coloneqq v_i^{-1}$. Using eq. (A.9) and eq. (A.98) gives,

$$\mathbf{V}_{\text{cav},n} = \mathbf{V} + \frac{\mathbf{V}\mathbf{w}_n \alpha \tau_n \mathbf{w}_n^{\intercal} \mathbf{V}}{1 - \mathbf{w}_n^{\intercal} \alpha \tau_n \mathbf{V} \mathbf{w}_n}. \tag{A.100}$$

Using eq. (A.10) and eq. (A.98) gives,

$$\log \det(\mathbf{V}_{\text{cav},n}) = \log \det(\mathbf{V}) - \log(1 - \mathbf{w}_n^{\intercal} \alpha \tau_n \mathbf{V} \mathbf{w}_n). \tag{A.101}$$

Subsituting eq. (A.100) and eq. (A.101) back to eq. (A.91) results in,

$$\begin{aligned} \mathcal{G}(q_*^{\setminus n}(\mathbf{u})) = \frac{M}{2}\log(2\pi) + \frac{1}{2}\log\det(\mathbf{V}) + \frac{1}{2}\mathbf{m}^{\intercal}\mathbf{V}^{-1}\mathbf{m} \\ - \frac{1}{2}\log(1 - \mathbf{w}_n^{\intercal}\alpha\tau_n\mathbf{V}\mathbf{w}_n) + \frac{1}{2}\frac{\mathbf{m}^{\intercal}\mathbf{w}_n\alpha\tau_n\mathbf{w}_n^{\intercal}\mathbf{m}}{1 - \mathbf{w}_n^{\intercal}\alpha\tau_n\mathbf{V}\mathbf{w}_n} \\ + \frac{1}{2}g_n\alpha\tau_n\mathbf{w}_n^{\intercal}\mathbf{V}_{\text{cav},n}\mathbf{w}_n\alpha\tau_n g_n - g_n\alpha\tau_n\mathbf{w}_n^{\intercal}\mathbf{V}_{\text{cav},n}\mathbf{V}^{-1}\mathbf{m} \end{aligned} \tag{A.102}$$

We now plug the above result back into the approximate marginal likelihood, yeilding,

$$\begin{aligned} \mathcal{F} = \frac{1}{2}\log|\mathbf{V}| + \frac{1}{2}\mathbf{m}^{\intercal}\mathbf{V}^{-1}\mathbf{m} - \frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}| + \frac{1}{\alpha}\sum_n \log \mathcal{Z}_{\text{tilted},n} \\ + \sum_n \Bigg[-\frac{1}{2\alpha}\log(1 - \mathbf{w}_n^{\intercal}\alpha\tau_n\mathbf{V}\mathbf{w}_n) + \frac{1}{2}\frac{\mathbf{m}^{\intercal}\mathbf{w}_n\tau_n\mathbf{w}_n^{\intercal}\mathbf{m}}{1 - \mathbf{w}_n^{\intercal}\alpha\tau_n\mathbf{V}\mathbf{w}_n} \\ + \frac{1}{2}g_n\tau_n\mathbf{w}_n^{\intercal}\mathbf{V}_{\text{cav},n}\mathbf{w}_n\alpha\tau_n g_n - g_n\tau_n\mathbf{w}_n^{\intercal}\mathbf{V}_{\text{cav},n}\mathbf{V}^{-1}\mathbf{m} \Bigg] \end{aligned} \tag{A.103}$$

A.7.1 Regression

We have shown in the previous section that the fixed point solution of the Power EP iterations can be obtained analytically for the regression case, $g_n = y_n$ and $\tau_n^{-1} = d_n = \alpha(K_{f_n f_n} - \mathbf{K}_{f_n \mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}f_n}) + \sigma_y^2$. Crucially, we can obtain a closed form expression for $\log \mathcal{Z}_{\text{tilted},n}$,

$$\log \mathcal{Z}_{\text{tilted},n} = -\frac{\alpha}{2}\log(2\pi\sigma_y^2) + \frac{1}{2}\log(\sigma_y^2) - \frac{1}{2}\log(\alpha v_n + \sigma_y^2) - \frac{1}{2}\frac{(y_n - \mu_n)^2}{v_n + \sigma_y^2/\alpha} \tag{A.104}$$

where $\mu_n = \mathbf{w}_n^\intercal \mathbf{m}_{\text{cav}} = \mathbf{w}_n^\intercal \mathbf{V}_{\text{cav}}(\mathbf{V}^{-1}\mathbf{m} - \mathbf{w}_n \alpha \tau_n y_n)$ and $v_n = \frac{d_n - \sigma_y^2}{\alpha} + \mathbf{w}_n^\intercal \mathbf{V}_{\text{cav}} \mathbf{w}_n$. We can therefore simplify the approximate marginal likelihood F further,

$$\begin{aligned}\mathcal{F} &= \frac{1}{2}\log|\mathbf{V}| + \frac{1}{2}\mathbf{m}^\intercal\mathbf{V}^{-1}\mathbf{m} - \frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}| + \sum_n \left[-\frac{1}{2}\log(2\pi\sigma_y^2) + \frac{1}{2\alpha}\log\sigma_y^2 - \frac{1}{2\alpha}\log d_n - \frac{y_n^2}{2d_n}\right] \\ &= -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{D} + \mathbf{Q}_{\mathbf{ff}}| - \frac{1}{2}\mathbf{y}^T(\mathbf{D} + \mathbf{Q}_{\mathbf{ff}})^{-1}\mathbf{y} - \frac{1-\alpha}{2\alpha}\sum_n \log(\frac{d_n}{\sigma_y^2}),\end{aligned} \tag{A.105}$$

where $\mathbf{Q}_{\mathbf{ff}} = \mathbf{K}_{\mathbf{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{uf}}$ and $\mathbf{D}$ is a diagonal matrix, $\mathbf{D}_{nn} = d_n$.

When $\alpha = 1$, the approximate marginal likelihood takes the same form as the FITC marginal likelhood,

$$\mathcal{F} = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{D} + \mathbf{Q}_{\mathbf{ff}}| - \frac{1}{2}\mathbf{y}^T(\mathbf{D} + \mathbf{Q}_{\mathbf{ff}})^{-1}\mathbf{y} \tag{A.106}$$

where $\mathbf{D}_{nn} = d_n = K_{f_n f_n} - \mathbf{K}_{f_n \mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}f_n} + \sigma_y^2$.

When α tends to 0, we have,

$$\lim_{\alpha\to 0}\frac{1-\alpha}{2\alpha}\sum_n \log(\frac{d_n}{\sigma_y^2}) = \frac{1}{2}\sum_n \lim_{\alpha\to 0}\frac{\log(1 + \alpha\frac{g_n}{\sigma_y^2})}{\alpha} = \frac{\sum_n h_n}{2\sigma_y^2}, \tag{A.107}$$

where $h_n = K_{f_n f_n} - \mathbf{K}_{f_n \mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}f_n}$. Therefore,

$$\mathcal{F} = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\sigma_y^2\mathrm{I} + \mathbf{Q}_{\mathbf{ff}}| - \frac{1}{2}\mathbf{y}^T(\sigma_y^2\mathrm{I} + \mathbf{Q}_{\mathbf{ff}})^{-1}\mathbf{y} - \frac{\sum_n h_n}{2\sigma_y^2}, \tag{A.108}$$

which is the variational lower bound of Titsias (Titsias, 2009).

A.7.2 Classification

In contrast to the regression case, the approximate marginal likelihood for classification cannot be simplified due to the non-Gaussian likelihood. Specifically, $\log \mathcal{Z}_{\text{tilted},n}$ is not analytically tractable, except when $\alpha = 1$ and the classification link function is the Gaussian

CDF. However, this quantity can be evaluated numerically, using sampling or Gauss-Hermite quadrature, since it only involves a one-dimensional integral.

We now consider the case when α tends to 0 and verify that in such case the approximate marginal likelihood becomes the variational lower bound. We first find the limits of individual terms in eq. (A.103):

$$\lim_{\alpha\to 0} -\frac{1}{2\alpha}\log(1-\mathbf{w}_n^\intercal \alpha\tau_n \mathbf{V}\mathbf{w}_n) = \frac{1}{2}\mathbf{w}_n^\intercal\tau_n\mathbf{V}\mathbf{w}_n \tag{A.109}$$

$$\frac{1}{2}\frac{\mathbf{m}^\intercal\mathbf{w}_n\tau_n\mathbf{w}_n^\intercal\mathbf{m}}{1-\mathbf{w}_n^\intercal\alpha\tau_n\mathbf{V}\mathbf{w}_n}\bigg|_{\alpha=0} = \frac{1}{2}\mathbf{m}^\intercal\mathbf{w}_n\tau_n\mathbf{w}_n^\intercal\mathbf{m} \tag{A.110}$$

$$\frac{1}{2}g_n\tau_n\mathbf{w}_n^\intercal\mathbf{V}_{\text{cav},n}\mathbf{w}_n\alpha\tau_n g_n\bigg|_{\alpha=0} = 0 \tag{A.111}$$

$$-g_n\tau_n\mathbf{w}_n^\intercal\mathbf{V}_{\text{cav},n}\mathbf{V}^{-1}\mathbf{m}\bigg|_{\alpha=0} = -g_n\tau_n\mathbf{w}_n^\intercal\mathbf{m}. \tag{A.112}$$

We turn our attention to $\log\mathcal{Z}_{\text{tilted},n}$. First, we expand $p^\alpha(y_n|f_n)$ using eq. (A.12):

$$p^\alpha(y_n|f_n) = \exp(\alpha\log p(y_n|f_n)) \tag{A.113}$$

$$= 1+\alpha\log p(y_n|f_n)+\xi(\alpha^2). \tag{A.114}$$

Substituting this result back into $\log\mathcal{Z}\text{tilted}/\alpha$ gives,

$$\frac{1}{\alpha}\log\mathcal{Z}_{\text{tilted}} = \frac{1}{\alpha}\log\int p(f_n|\mathbf{u})q_{\text{cav}}(\mathbf{u})p^\alpha(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u} \tag{A.115}$$

$$= \frac{1}{\alpha}\log\int p(f_n|\mathbf{u})q_{\text{cav}}(\mathbf{u})[1+\alpha\log p(y_n|f_n)+\xi(\alpha^2)]\mathrm{d}f_n\mathrm{d}\mathbf{u} \tag{A.116}$$

$$= \frac{1}{\alpha}\log\left[1+\alpha\int p(f_n|\mathbf{u})q_{\text{cav}}(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u}+\alpha^2\xi(1)\right] \tag{A.117}$$

$$= \frac{1}{\alpha}\left[\alpha\int p(f_n|\mathbf{u})q_{\text{cav}}(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u}+\alpha^2\xi(1)\right] \tag{A.118}$$

$$= \int p(f_n|\mathbf{u})q_{\text{cav}}(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u}+\alpha\xi(1). \tag{A.119}$$

Therefore,

$$\lim_{\alpha\to 0}\frac{1}{\alpha}\log\mathcal{Z}_{\text{tilted}} = \int p(f_n|\mathbf{u})q(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u}. \tag{A.120}$$

Putting these results into eq. (A.103), we obtain,

$$\begin{aligned}
\mathcal{F} &= \frac{1}{2}\log|\mathbf{V}| + \frac{1}{2}\mathbf{m}^\intercal\mathbf{V}^{-1}\mathbf{m} - \frac{1}{2}\log|\mathbf{K_{uu}}| \\
&\quad + \sum_n \frac{1}{2}\mathbf{w}_n^\intercal\tau_n\mathbf{V}\mathbf{w}_n + \frac{1}{2}\mathbf{m}^\intercal\mathbf{w}_n\tau_n\mathbf{w}_n^\intercal\mathbf{m} - g_n\tau_n\mathbf{w}_n^\intercal\mathbf{m} + \int p(f_n|\mathbf{u})q(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u} \\
&= \frac{1}{2}\log|\mathbf{V}| + \frac{1}{2}\mathbf{m}^\intercal\mathbf{V}^{-1}\mathbf{m} - \frac{1}{2}\log|\mathbf{K_{uu}}| + \frac{1}{2}\mathbf{m}^\intercal(\mathbf{V}^{-1} - \mathbf{K_{uu}^{-1}})\mathbf{m} - \mathbf{m}^\intercal\mathbf{V}^{-1}\mathbf{m} \\
&\quad + \sum_n \frac{1}{2}\mathbf{w}_n^\intercal\tau_n\mathbf{V}\mathbf{w}_n + \int p(f_n|\mathbf{u})q(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u} \\
&= \frac{1}{2}\log|\mathbf{V}| - \frac{1}{2}\mathbf{m}^\intercal\mathbf{K_{uu}^{-1}}\mathbf{m} - \frac{1}{2}\log|\mathbf{K_{uu}}| + \sum_n \frac{1}{2}\mathbf{w}_n^\intercal\tau_n\mathbf{V}\mathbf{w}_n + \sum_n \int p(f_n|\mathbf{u})q(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u}.
\end{aligned} \tag{A.121}$$

We now write down the evidence lower bound of the global variational approach of Titsias (Titsias, 2009), as applied to the classification case (Hensman et al., 2015),

$$\mathcal{F}_{\mathrm{VFE}} = -\mathrm{KL}(q(\mathbf{u})||p(\mathbf{u})) + \sum_n \int p(f_n|\mathbf{u})q(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u} \tag{A.122}$$

where

$$\begin{aligned}
-\mathrm{KL}(q(\mathbf{u})||p(\mathbf{u})) &= -\frac{1}{2}\mathrm{trace}(\mathbf{K_{uu}^{-1}}\mathbf{V}) - \frac{1}{2}\mathbf{m}^\intercal\mathbf{K_{uu}^{-1}}\mathbf{m} + \frac{M}{2} - \frac{1}{2}\log|\mathbf{K_{uu}}| + \frac{1}{2}\log|\mathbf{V}| \\
&= -\frac{1}{2}\mathrm{trace}([\mathbf{V}^{-1} - \sum_n \mathbf{w}_n\tau_n\mathbf{w}_n]\mathbf{V}) - \frac{1}{2}\mathbf{m}^\intercal\mathbf{K_{uu}^{-1}}\mathbf{m} + \frac{M}{2} - \frac{1}{2}\log|\mathbf{K_{uu}}| + \frac{1}{2}\log|\mathbf{V}| \\
&= \frac{1}{2}\mathrm{trace}(\sum_n \mathbf{w}_n\tau_n\mathbf{w}_n\mathbf{V}) - \frac{1}{2}\mathbf{m}^\intercal\mathbf{K_{uu}^{-1}}\mathbf{m} - \frac{1}{2}\log|\mathbf{K_{uu}}| + \frac{1}{2}\log|\mathbf{V}|.
\end{aligned} \tag{A.123}$$

Therefore, $\mathcal{F}_{\mathrm{VFE}}$ is identical to the limit of the approximate marginal likelihood provided by Power EP as shown in eq. (A.121).