Journal of Machine Learning Research 18 (2017) 1-72 Submitted 11/16; Revised 10/17; Published 10/17

A Unifying Framework for Gaussian Process Pseudo-Point Approximations using Power Expectation Propagation

Thang D. Bui TDB40@CAM.AC.UK
Josiah Yan JOSIAH.YAN@GMAIL.COM
Richard E. Turner RET26@CAM.AC.UK
Computational and Biological Learning Lab, Department of Engineering
University of Cambridge, Trumpington Street, Cambridge, CB2 1PZ, UK

Editor: Neil Lawrence

Abstract

Gaussian processes (GPs) are flexible distributions over functions that enable high-level assumptions about unknown functions to be encoded in a parsimonious, flexible and general way. Although elegant, the application of GPs is limited by computational and analytical intractabilities that arise when data are sufficiently numerous or when employing non-Gaussian models. Consequently, a wealth of GP approximation schemes have been developed over the last 15 years to address these key limitations. Many of these schemes employ a small set of pseudo data points to summarise the actual data. In this paper we develop a new pseudo-point approximation framework using Power Expectation Propagation (Power EP) that unifies a large number of these pseudo-point approximations. Unlike much of the previous venerable work in this area, the new framework is built on standard methods for approximate inference (variational free-energy, EP and Power EP methods) rather than employing approximations to the probabilistic generative model itself. In this way all of the approximation is performed at 'inference time' rather than at 'modelling time', resolving awkward philosophical and empirical questions that trouble previous approaches. Crucially, we demonstrate that the new framework includes new pseudo-point approximation methods that outperform current approaches on regression and classification tasks.

Keywords: Gaussian process, expectation propagation, variational inference, sparse approximation

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 paper 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 paper 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 paper begins by reviewing several frameworks for sparse approximation focussing on the GP regression and classification setting (Section 2). It then lays out the new unifying framework and the relationship to existing techniques (Section 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 4.

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, $p(y_n|f, \mathbf{x}_n, \theta)$ which may also depend on hyperparameters θ. 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

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

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 data points 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.

2.1 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 $f = \{\mathbf{u}, \mathbf{f}, f_{\neq \mathbf{u},\mathbf{f}}\}$. Here $f_{\neq \mathbf{u},\mathbf{f}}$ denotes the function values which are not at the training inputs or pseudo-inputs. The GP prior is then decomposed using the product rule

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

Of central interest is the relationship between the pseudo-points and the training function values $p(\mathbf{f}|\mathbf{u},\theta) = \mathcal{N}(\mathbf{f}; \mathbf{K}_{\mathbf{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u}, \mathbf{D}_{\mathbf{ff}})$ where $\mathbf{D}_{\mathbf{ff}} = \mathbf{K}_{\mathbf{ff}} - \mathbf{Q}_{\mathbf{ff}}$ and $\mathbf{Q}_{\mathbf{ff}} = \mathbf{K}_{\mathbf{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{uf}}$. 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 $[\mathbf{K}_{\mathbf{uu}}]_{mm'} = k_\theta(\mathbf{z}_m, \mathbf{z}_{m'})$ and similarly for the covariance between the pseudo-input and data locations $[\mathbf{K}_{\mathbf{uf}}]_{mn} = k_\theta(\mathbf{z}_m, \mathbf{x}_n)$. Importantly, this term saddles inference and learning with a complexity cost that is cubic in the number of data points. 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,

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

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

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

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).

Before considering inference in the modified models, 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,

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

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

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

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 data points, 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

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

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 of understanding 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 hyperparameters, θ, to be learned via optimisation

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

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

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

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 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 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 the presentation in 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

$$q^{\mathrm{opt}}(f|\theta) = \mathcal{GP}(f;\mu_{f|\mathbf{y}},\Sigma_{f|\mathbf{y}}),\ \ \mu_{f|\mathbf{y}} = \mathbf{Q}_{f\mathbf{f}}\tilde{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y},\ \ \Sigma_{f|\mathbf{y}} = \mathbf{K}_{ff} - \mathbf{Q}_{f\mathbf{f}}\tilde{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{Q}_{\mathbf{f}f}, \tag{3}$$

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 Equation (1) as $\alpha \to 0$). In fact, DTC was originally derived using a related KL argument (Csató, 2002; Seeger et al., 2003). The optimised free-energy is

$$\mathcal{F}(q^{\mathrm{opt}},\theta) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\tilde{\mathbf{K}}_{\mathbf{ff}}| - \frac{1}{2}\mathbf{y}^\intercal\tilde{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y} - \frac{1}{2\sigma_y^2}\mathrm{trace}(\mathbf{K}_{\mathbf{ff}} - \mathbf{Q}_{\mathbf{ff}}). \tag{4}$$

Notice that the free-energy has an additional trace term as compared to the marginal likelihood obtained from the DTC generative model approach (see Equation (2) 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 $\mathbb{E}_{q(f_n)}\left[\log p(y_n|\mathrm{f}_n,\theta)\right]$ which may be evaluated using quadrature (Hensman et al., 2015).

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 of the underlying function is revealed, without altering the modelling assumptions (e.g. see Bui et al. (2017) 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 facts 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?

2.3 Sparse GP Approximation via Approximate Inference: EP

Expectation Propagation (EP) is a deterministic inference method (Minka, 2001) that is known to outperform VFE methods in GP classification when non-sparse fully-factored approximations $q(\mathbf{f}) = \prod_n q_n(\mathrm{f}_n)$ 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 know 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 (Kushner and Budhiraja, 2000; Ito and Xiong, 2000) was employed rather than full-blown EP.

2.4 Contributions

One of the primary contributions of this paper is to provide a clear computational exposition of Csató and Opper's EP procedure including an explicit form of the approximating distribution and full details about each step. In addition to bringing 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') recovers 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, an important open area of research.

Many of the new theoretical contributions described above are summarised in Figure 1 along with their relationship to previous work.

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.

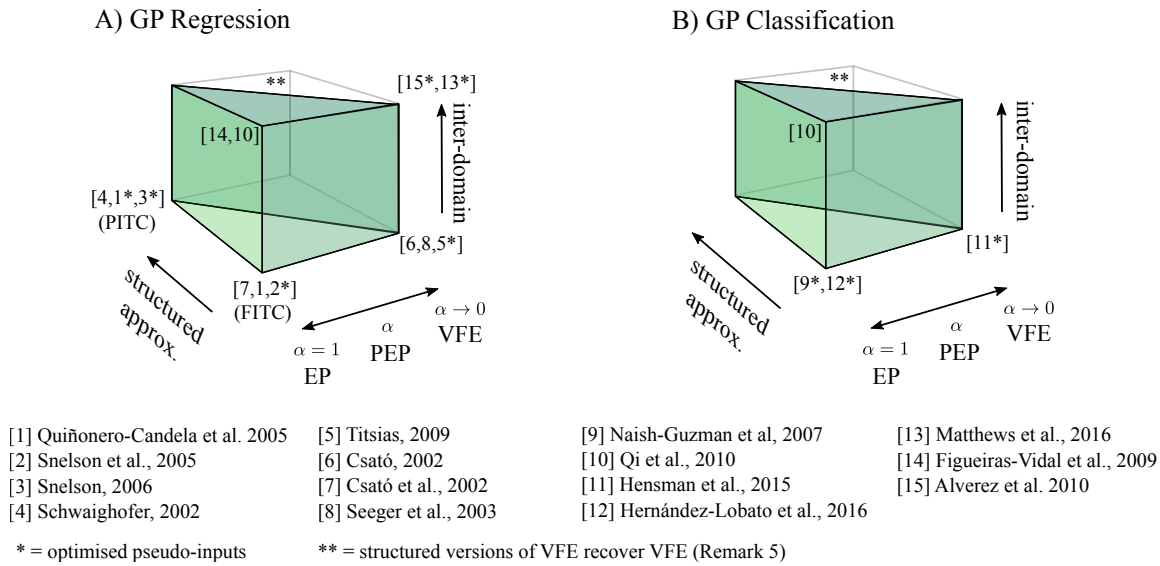

Figure 1: A unified view of pseudo-point GP approximations applied to A) regression, and B) classification. Every point in the algorithm polygons corresponds to a form of GP approximation. Previous algorithms correspond to labelled vertices. The new Power EP framework encompasses the three polygons, including their interior.

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 $q^*(f) \approx p(f, \mathbf{y}|\theta)$ 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 $q^*(f) = Zq(f)$ into a normalised component that approximates the posterior $q(f) \approx p(f|\mathbf{y}, \theta)$ 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, $q^*(f) = Z\, \mathcal{GP}(f; m_{\mathrm{f}}, V_{\mathrm{ff}'})$.

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

$$\overline{\mathrm{KL}}[q^*(f)||p(f,\mathbf{y}|\theta)] = \int q^*(f)\log\frac{q^*(f)}{p(f,\mathbf{y}|\theta)}\mathrm{d}f + \int (p(f,\mathbf{y}|\theta) - q^*(f))\,\mathrm{d}f. \tag{5}$$

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 $q^*(f) = Z_{\mathrm{VFE}}\, q(f)$ encourages the approximation to match both the posterior and marginal-likelihood, and it yields analytic solutions

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

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). A disadvantage is that optimisation of hyperparameters must logically proceed by optimising the marginal likelihood approximation, $Z^{\mathrm{opt}}_{\mathrm{VFE}}$, 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 the single objective $p(\mathbf{y}|\theta) - \overline{\mathrm{KL}}[q^*(f|\theta)||p(f,\mathbf{y}|\theta)]$ directly for both $q^*(f|\theta)$ and θ is equivalent and that this also recovers the standard VFE method (see Appendix A).

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,

$$p^*(f|\mathbf{y},\theta) = p(f|\mathbf{y},\theta)p(\mathbf{y}|\theta) = p(f|\theta)\prod_n p(y_n|f,\theta) \approx p(f|\theta)\prod_n t_n(\mathbf{u}) = q^*(f|\theta). \tag{6}$$

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. As the approximation retains the exact prior it explicitly depends on the hyperparameters θ. However, we will suppress this dependence to lighten the notation.

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 $\prod_n t_n(\mathbf{u}) = \mathcal{N}(\tilde{\mathbf{y}};\tilde{\mathbf{W}}\mathbf{u},\tilde{\Sigma})$ (assuming that the product of these factors is normalisable which may not be the case for heavy tailed likelihoods, for example). 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 $\tilde{\mathbf{y}} = \tilde{\mathbf{W}}\mathbf{u} + \tilde{\Sigma}^{1/2}\epsilon$. We note that the

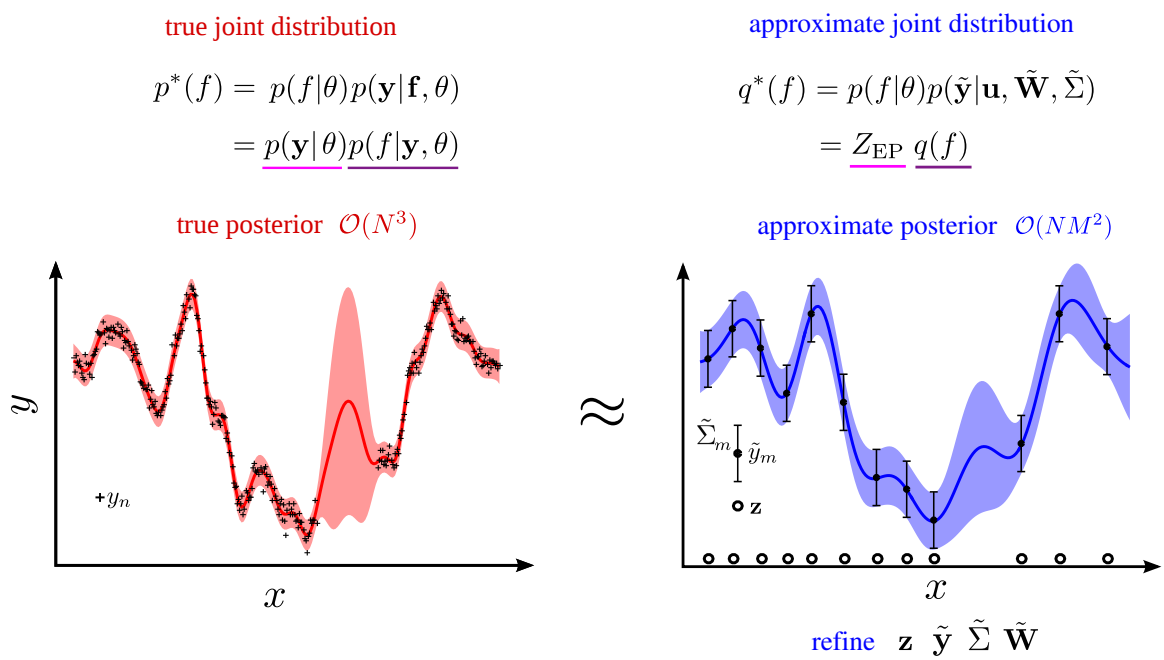

Figure 2: Perspectives on the approximating family. The true joint distribution over the unknown function f and the N data points $\mathbf{y}$ (top left) comprises the GP prior and an intractable likelihood function. This is approximated by a surrogate regression model with a joint distribution over the function f and M surrogate data points $\tilde{\mathbf{y}}$ (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 $\tilde{\Sigma}$.

pseudo-points $\mathbf{u}$ live on the latent function (or an inter-domain transformation thereof) and the surrogate observations $\tilde{\mathbf{y}}$ will not generally lie on the latent function. The surrogate observations and the pseudo-points are therefore analogous to the data $\mathbf{y}$ and the function values $\mathbf{f}$ in a normal Gaussian Process regression problem, respectively. To make the paper more specific on this point, we have defined parameters for the surrogate regression problem explicitly in Appendix H. The PEP algorithm will implicitly 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 Figure 2).

3.3 The EP Algorithm

One method for updating the approximate likelihood factors $t_n(\mathbf{u})$ would be 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),

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

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 the 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,

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

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.

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

$$q^*(f) \leftarrow \operatorname*{argmin}_{q^*(f)\in\mathcal{Q}} \overline{\mathrm{KL}}[\tilde{p}(f)||q^*(f)].$$

   Here $\mathcal{Q}$ is the set of allowed $q^*(f)$ defined by Equation (6).
3. **Update**: compute a new fraction of the approximate factor by dividing the new approximate posterior by the cavity, $t^\alpha_{n,\mathrm{new}}(\mathbf{u}) = q^*(f)/q^*_{\backslash n}(f)$, 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 guaranteed to converge in general, 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 without pseudo-data) (Seeger, 2008).

If Power EP converges, the fractional updates are equivalent to running the original EP procedure, but replacing the KL minimisation with an alpha-divergence minimisation (Zhu and Rohwer, 1995; Minka, 2005),

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

When $\alpha = 1$, the alpha-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 Appendix B for more details). In the current case, we will now show explicitly that these cases of Power EP recover FITC and Titsias's VFE solution respectively.

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 ground work for the equivalence relationships. The Appendix F 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 posterior, $q^*(f|\theta) = \mathcal{Z}_{\mathrm{PEP}}\mathcal{GP}(f; m_{\mathrm{f}}, V_{\mathrm{ff}'})$. The mean and covariance functions of the approximate posterior are

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

Deletion: 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 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 the following mean and covariance functions

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

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 *Due to the structure of the approximate posterior, $q^*(f) = p(f_{\neq\mathbf{u}}|\mathbf{u})q^*(\mathbf{u})$, the objective, $\overline{\mathrm{KL}}[\tilde{p}(f)||q^*(f)]$ 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'[2] provides a useful shortcut that only requires one integral to compute the log-normaliser of the tilted distribution, $\log \tilde{Z}_n = \log \mathbb{E}_{q^*_{\setminus n}(f)}[p^\alpha(y_n|\mathrm{f}_n)]$, before differentiating w.r.t. the cavity mean to give

$$\mathbf{m}_{\mathbf{u}} = \mathbf{m}_{\mathbf{u}}^{\setminus n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\setminus n} \frac{\mathrm{d}\log \tilde{Z}_n}{\mathrm{d}m_{\mathrm{f}_n}^{\setminus n}}; \qquad \mathbf{V}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}^{\setminus n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\setminus n} \frac{\mathrm{d}^2 \log \tilde{Z}_n}{\mathrm{d}(m_{\mathrm{f}_n}^{\setminus n})^2} \mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\setminus n}. \tag{7}$$

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

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

where we have defined the log-normaliser as the functional $\mathcal{G}(\tilde{\mathcal{N}}(\mathbf{u}; z, \mathbf{T}_1, \mathbf{T}_2)) = \log \int \tilde{\mathcal{N}}(\mathbf{u}; z, \mathbf{T}_1, \mathbf{T}_2)\mathrm{d}\mathbf{u}$. Remarkably, these results and Equation (7) reveals that $\mathbf{T}_{2,n,new}$ is a rank-1 matrix. As such, 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 a significant memory saving and $\mathcal{O}(NM^2)$ 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,

$$\log \mathcal{Z}_{\mathrm{PEP}} = \log \int p(f) \prod_n t_n(\mathbf{u})\mathrm{d}f = \mathcal{G}(q^*(\mathbf{u})) - \mathcal{G}(p^*(\mathbf{u})) + \sum_n \log z_n. \tag{8}$$

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

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:

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

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. 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 $q^*_{\setminus n}(\mathbf{u})$ become,

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

2. In this case, the dominated convergence theorem can be used to justify the interchange of integration and differentiation (see e.g. Brown, 1986).

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 $\int q^*_{\setminus n}(f)p^\alpha(y_n|\mathrm{f}_n)\mathrm{d}f_{\neq\mathbf{u}} = q^*_{\setminus n}(\mathbf{u})\int p(\mathrm{f}_n|\mathbf{u})p^\alpha(y_n|\mathrm{f}_n)\mathrm{d}\mathrm{f}_n \propto q^*(\mathbf{u})$. 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.

Remark 2 *When $\alpha = 1$ [EP], the Power EP posterior becomes the FITC posterior in Equation* (1) *and the Power EP approximate marginal likelihood becomes the FITC marginal likelihood in Equation* (2)*. 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 Equations* (3) *and* (4) *(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.*

Remark 4 *For fixed hyperparameters, a single pass of Power EP is sufficient for convergence in the regression case.*

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 2.1)

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 data points 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 2.1). Each PEP factor update will then approximate an entire block which will contain a set of data points, rather than just a single one. This is related to a form of EP approximation that has recently been used to distribute Monte Carlo algorithms across many machines (Gelman et al., 2014; 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,

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

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 5 *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 3.6.*

Remark 6 *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). This is a result of the equivalence of local and global exclusive KL divergence minimisation. See Appendix B for more details and Figure 1 for more relationships.*

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[3]. However, it involves a one-dimensional expectation of a non-linear function of a normally-distributed random variable and, therefore, can be approximated using numerical methods, e.g. Gauss-Hermite quadrature. This procedure gives an approximation to the expectation, resulting in 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 Figure 1 which also shows that inter-domain transformations and structured approximations have

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

not yet been fully explored 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., 2014; Xu et al., 2014). In addition, when $\alpha = 1$, $M = N$ and the pseudo-points are at the training inputs, the standard EP algorithm for GP classification is recovered (Rasmussen and Williams, 2005, sec. 3.6).

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).

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 data points (Hernández-Lobato and Hernández-Lobato, 2016). In particular, the Power EP update steps in Section 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 Equation (8) 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.

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 paper 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 or regression); the data set (synthetic data sets, 8 real world regression data sets and 6 classification data sets); 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 Appendix I (along with additional experiments). Python

and Matlab implementations are available at `http://github.com/thangbui/sparseGP_powerEP`.

4.1 Regression on Synthetic Data Sets

In the first experiment, we investigate the performance of the proposed Power EP method on toy regression data sets 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 data sets, each comprising 1000 data points 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). 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 Figure 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 $M = N = 500$. 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.

4.2 Regression on Real-world Data Sets

The experiment above was replicated on 8 UCI regression data sets, 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 is 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 VFE and EP and because settings around this value empirically performed the best on average across all data sets, splits, numbers of inducing points, and evaluation metrics.

In Figure 4A we plot (for each data set, 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

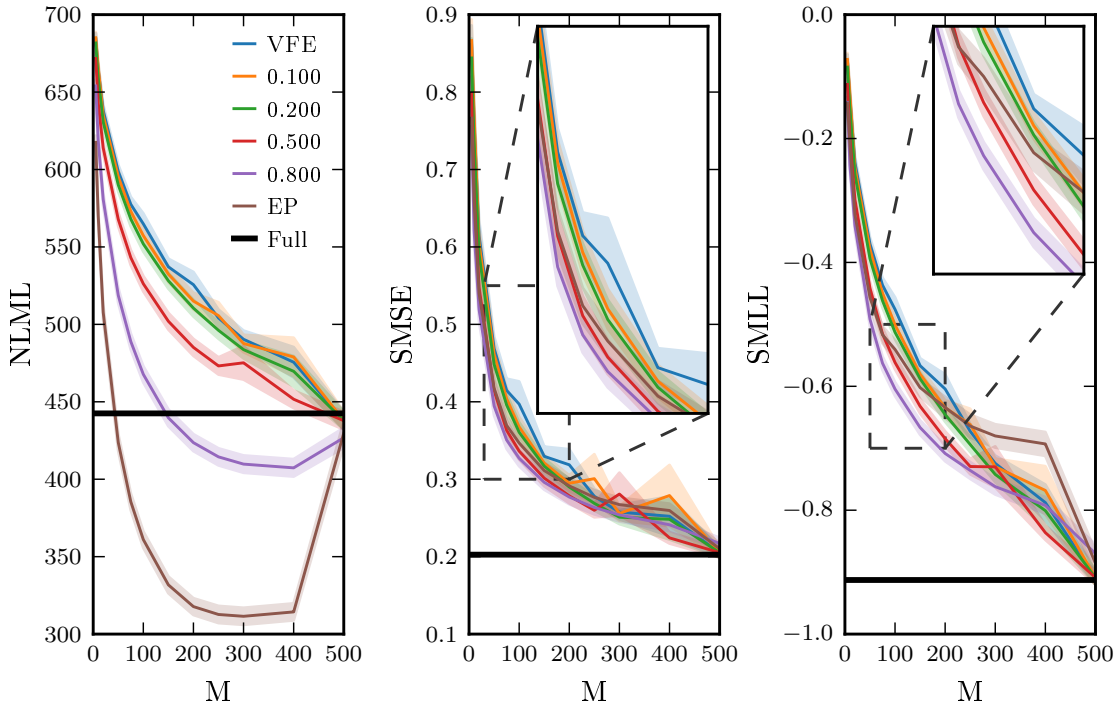

Figure 3: The performance of various α values averaged over 30 trials. See text for more details

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 Figure 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 worst.

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 Figure 4B were produced by sorting the performances of the nine different approximating

methods for each data set, split, and number of pseudo-points M and assigning a rank. These ranks are then averaged over all data sets 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 data sets, but the same general trends are clear: For MSE $\alpha = 0.5$ is better than VFE on 6/8 data sets and EP on 8/8 data sets, whilst VFE is better than EP on 3 data sets (the difference on the others being small). For SMLL EP is better than $\alpha = 0.5$ on 5/8 data sets and VFE on 7/8 data sets, whilst $\alpha = 0.5$ is better than VFE on 8/8 data sets. 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 data sets 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 SMLL we recommend using EP. It is possible that more fine grained recommendations are possible based upon details of the data set and the computational resources available for processing, but further work will be needed to establish this.

4.3 Binary Classification

We also evaluated the Power EP method on 6 UCI classification data sets, each has 20 train/test splits. The details of the data sets are included in Appendix I.3. The data sets are all roughly balanced, and the most imbalanced is `pima` with 500 positive and 267 negative data points. Again α was varied between 0 and 1, and M was varied between 10 and 100. We adopt the experimental protocol discussed in Section 3.9, including: (i) not waiting for Power EP to converge before making hyperparameter updates, (ii) using minibatches of data points 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 differences in the implementation—optimisation for VFE vs. the iterative PEP procedure, and we also note that each step of PEP only gets to see a tiny fraction of each data point when α is small which can slow the learning speed. 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 Figure 6, we plot (for each data set, each split and each setting of M) the evaluation scores using one inference algorithm against the score obtained using another [see Section 4.2 for a detailed explanation of the plots]. In contrast to the regression results in Section 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 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%)

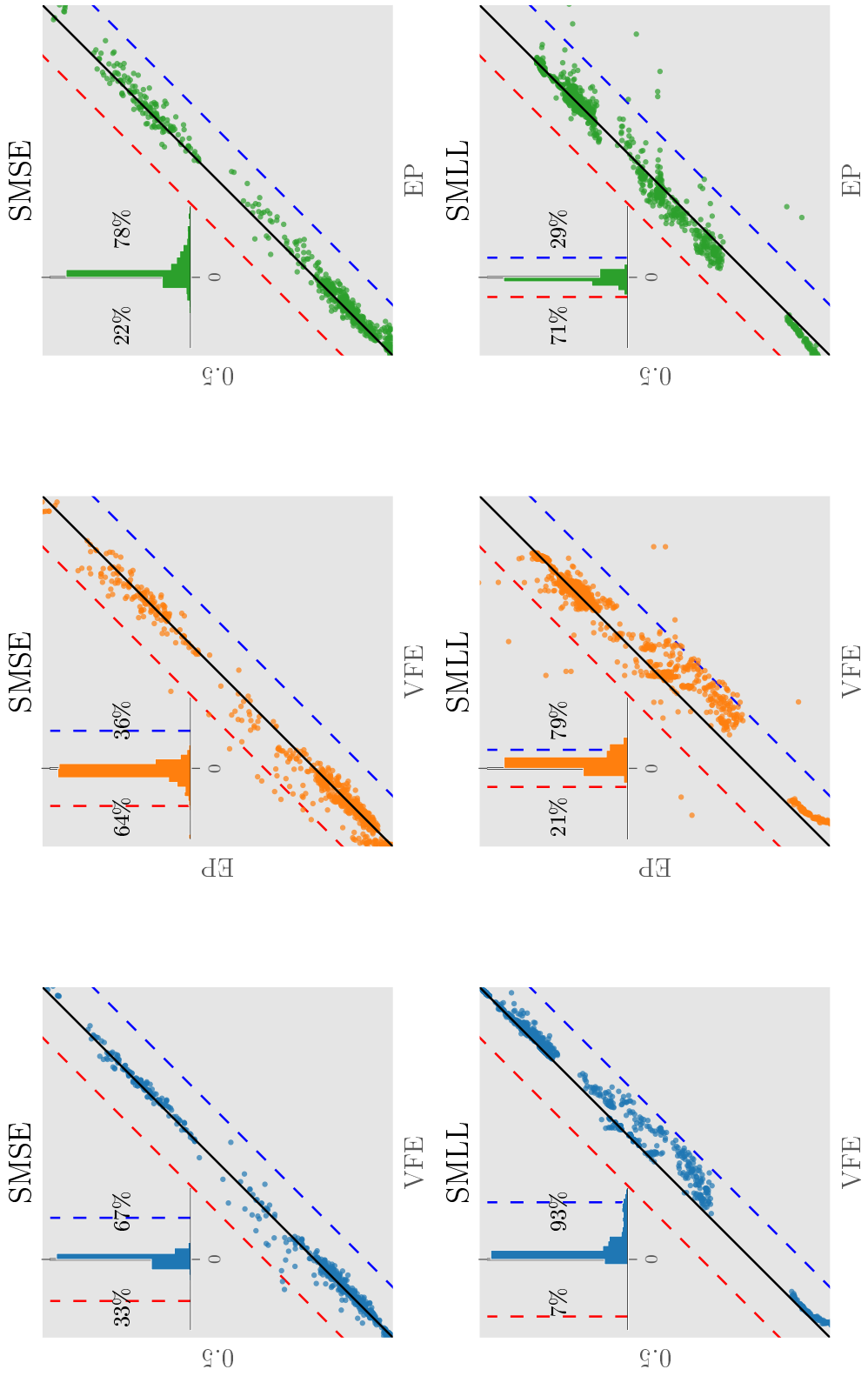

Figure 4: Pair-wise comparisons between Power EP with $\alpha = 0.5$, EP ($\alpha = 1$) and VFE ($\alpha \to 0$), evaluated on several regression data sets and various settings of M. Each coloured point is the result for one split. Points that are below the diagonal line illustrate the method on the 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 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.

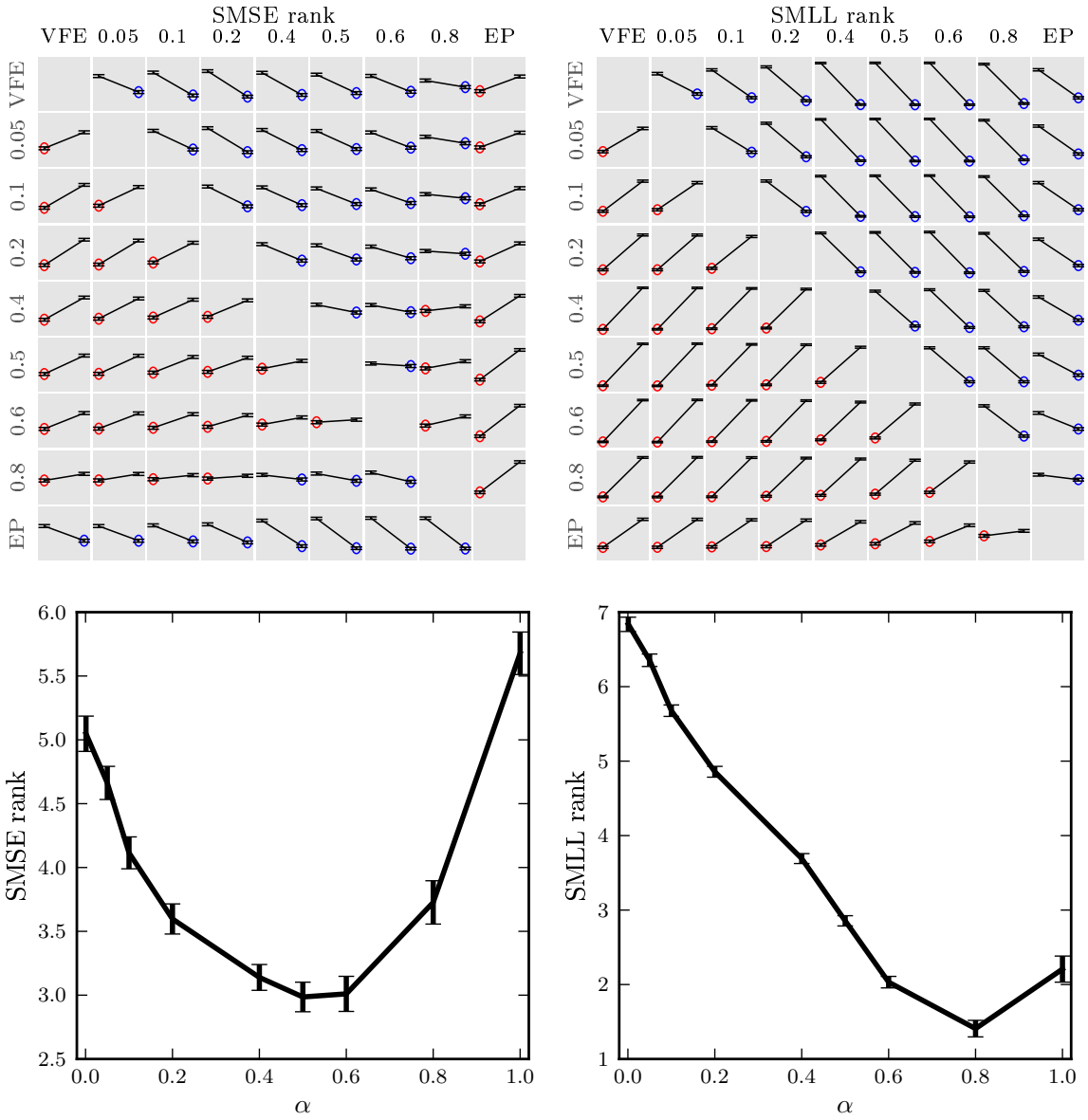

Figure 5: Average ranking of various α values in the regression experiment, lower is better. Top plots show the pairwise comparisons. Red circles denote rows being better than the corresponding columns, and blue circles mean vice versa. Bottom plots show the ranks of all methods when being compared together. Intermediate α values (not EP or VFE) are best on average.

We repeat the pairwise comparison above to all methods and show the results in Figure 7. The plots show that there is no conclusive winner on the test error metric with VFE and PEP $\alpha \leq 0.5$ performing similarly. On the NLL metric VFE, PEP $\alpha = 0.4$ and PEP $\alpha = 0.5$ have a slight edge over other α values. 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 Figure 7. However, the large errorbars suggest the difference between the methods is small in both metrics.

There is some variability between individual data sets, 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 data set and is better than EP on 3/6 data sets (the differences on the other data sets are small). VFE outperforms EP on 2/6 data sets, while EP beats VFE on only 1/6 data sets. For NLL, PEP $\alpha = 0.5$ only clearly outperforms VFE on 1/6 data set, but is worse compared to VFE on 1 data set (the other 4 data sets have no clear winner). PEP $\alpha = 0.5$ is better than EP on 5/6 data sets and EP is better on the remaining data set). EP is only better than VFE on 2/6 data sets, and is outperformed by VFE on the other 4/6 data sets. 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 data set and the computational resources available for processing, but further work will be needed to establish this.

5. Discussion

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,

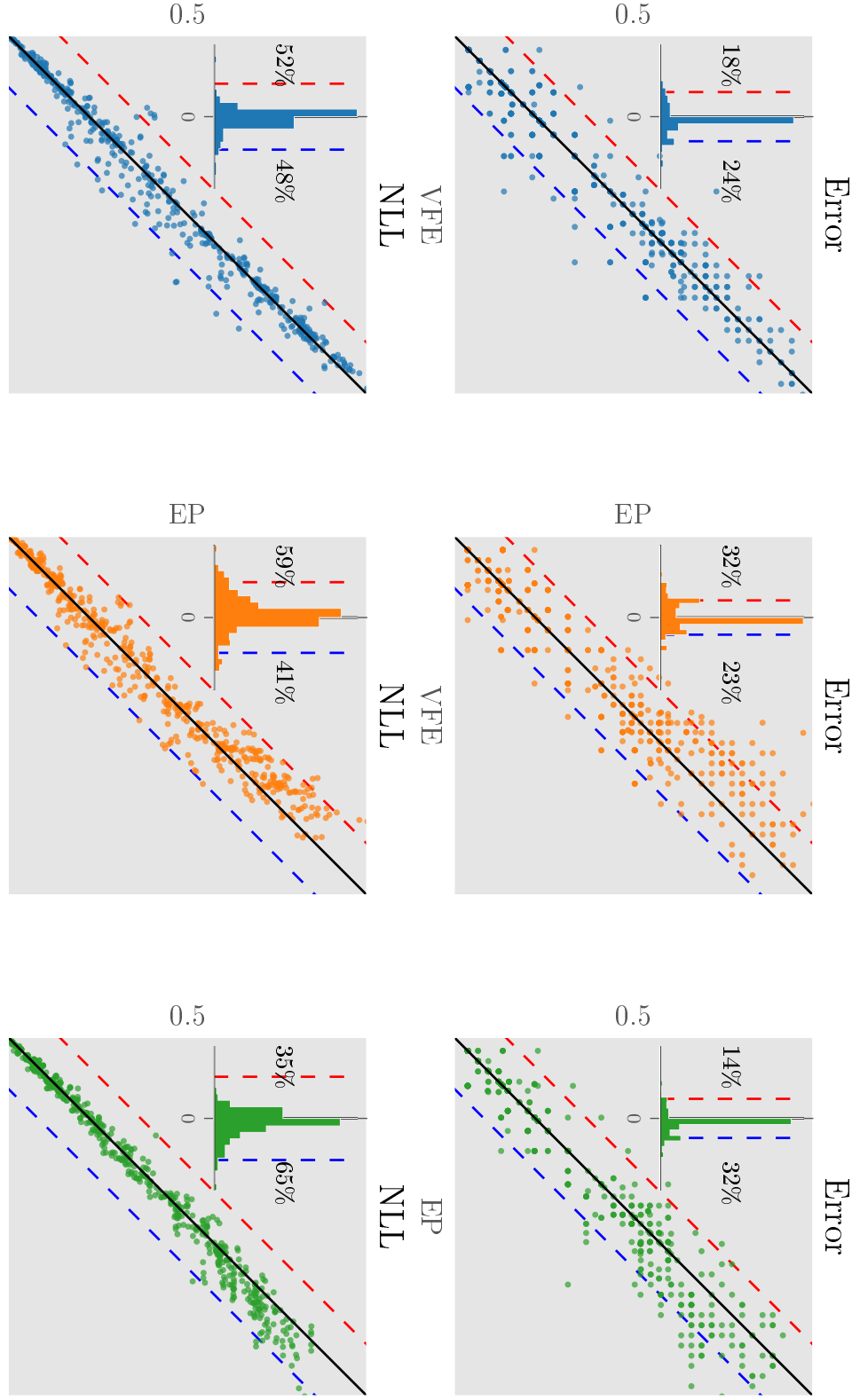

Figure 6: Pair-wise comparisons between Power EP with $\alpha = 0.5$, EP ($\alpha = 1$) and VFE ($\alpha \to 0$), evaluated on several classification data sets and various settings of M. Each coloured point is the result for one split. Points that are below the diagonal line illustrate the method on 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 pairwise ranking of the two methods. Positive differences means the y-axis method is better than the x-axis method and vice versa.

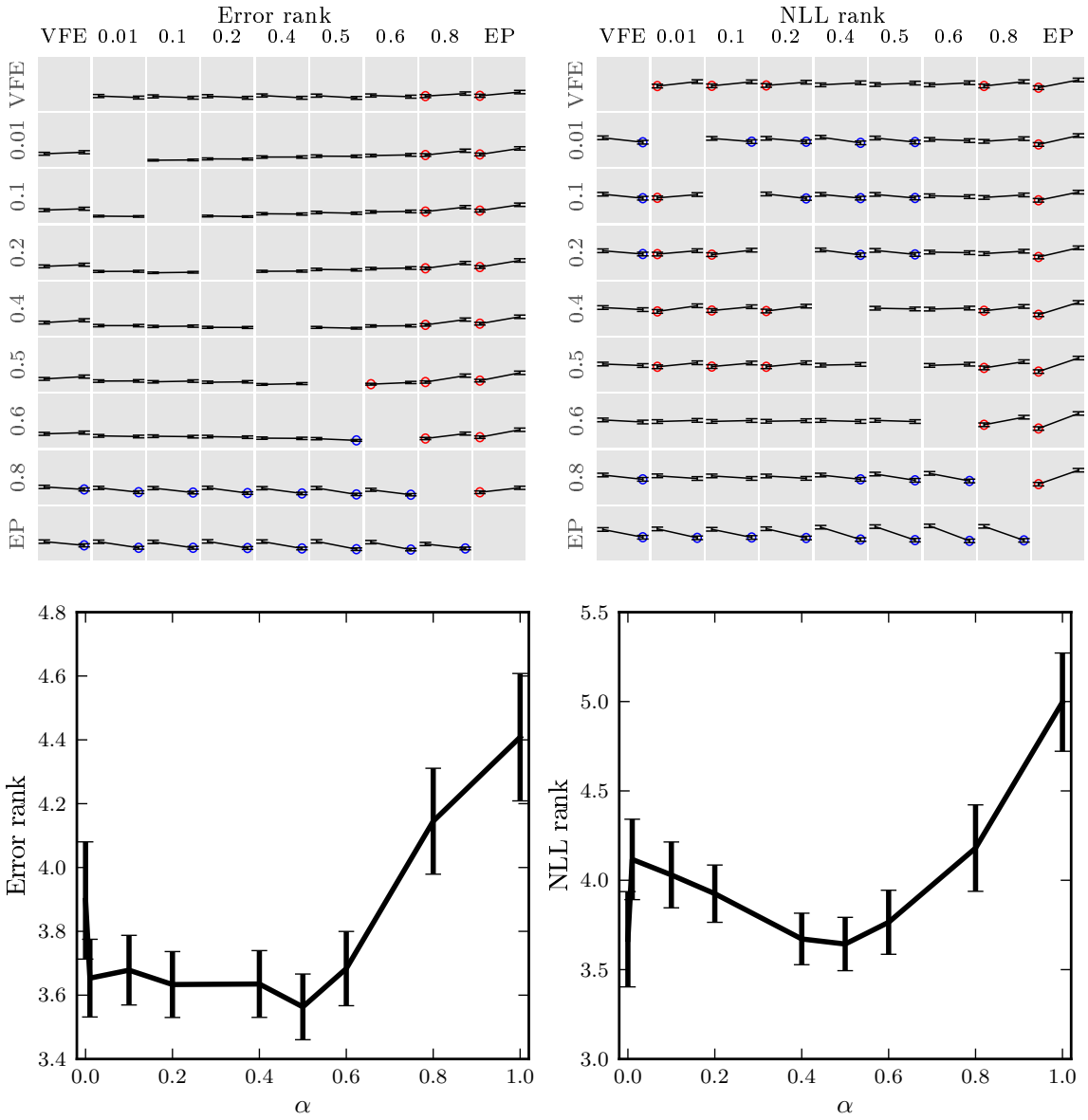

Figure 7: Average ranking of various α values in the classification experiment, lower is better. Top plots show the pairwise comparisons. Red circles denote rows being better than the corresponding columns, and blue circles mean vice versa. Bottom plots show the ranks of all methods when being compared together. Intermediate α values (not EP or VFE) are best on average.

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 or 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 (see e.g. Minka, 2005; Hernández-Lobato et al., 2016; Depeweg et al., 2016). In this case, the alpha-divergence interpretation of Power EP shows that it is minimising the Hellinger distance whose square root is a valid distance metric. Further experimental and theoretical work is required to clarify these issues.

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 data set. The SMSE and SMLL on the training and test sets, for example, are similar which is indicative of a well-fit model. It would be interesting to explore distributional hyperparameter estimates (see e.g. Piironen and Vehtari, 2017) that employ these pseudo-point approximations.

One of the features of the approximate generative models introduced in Section 2.1 for regression, is that they contain input-dependent noise, unlike the original model. Many data sets 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 data points (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 paper 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.

6. Conclusion

This paper 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.

Acknowledgments

The authors would like to thank Prof. Carl Edward Rasmussen, Nilesh Tripuraneni, Matthias Bauer, James Hensman, and Hugh Salimbeni for insightful comments and discussion. TDB thanks Google for funding his European Doctoral Fellowship. RET thanks EPSRC grants EP/G050821/1, EP/L000776/1 and EP/M026957/1.

Appendix A. 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

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

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,

$$\begin{aligned}\operatorname*{argmin}_{q^*(f|\theta)} \overline{\mathrm{KL}}[q^*(f|\theta)||p(f,\mathbf{y}|\theta)] &= \operatorname*{argmax}_{q^*(f|\theta)} \left(p(\mathbf{y}|\theta) - \overline{\mathrm{KL}}[q^*(f|\theta)||p(f,\mathbf{y}|\theta)]\right) \\ &= \operatorname*{argmax}_{q^*(f|\theta)} \left(\int q^*(f|\theta) \log \frac{p(f,\mathbf{y}|\theta)}{q^*(f|\theta)} \mathrm{d}f + \int q^*(f|\theta) \mathrm{d}f\right).\end{aligned}$$

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 $q^*(f|\theta) = Z_{\mathrm{VFE}}\, q(f|\theta)$:

$$\mathcal{G}(Z_{\mathrm{VFE}} q(f|\theta),\theta) = Z_{\mathrm{VFE}} \left(1 - \log Z_{\mathrm{VFE}} + \int q(f|\theta) \log \frac{p(f,\mathbf{y}|\theta)}{q(f|\theta)} \mathrm{d}f\right).$$

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

$$\mathcal{G}(Z_{\mathrm{VFE}}^{\mathrm{opt}} q(f|\theta),\theta) = Z_{\mathrm{VFE}}^{\mathrm{opt}} = \exp(\mathcal{F}(q(f|\theta),\theta)).$$

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.

Appendix B. 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,

$$\begin{aligned}
\mathrm{KL}[q(\theta)||p(\theta)] &= \int \mathrm{d}\theta q(\theta) \log \frac{q(\theta)}{p(\theta)} \\
&= \int \mathrm{d}\theta q(\theta) \log \frac{\prod_n t_n(\theta)}{\prod_n f_n(\theta)} \\
&= \int \mathrm{d}\theta q(\theta) \log \left[\frac{\prod_n t_n(\theta)}{\prod_n f_n(\theta)} \frac{\prod_n \prod_{i \neq n} t_i(\theta)}{\prod_n \prod_{i \neq n} t_i(\theta)} \right] \\
&= \int \mathrm{d}\theta q(\theta) \log \frac{\prod_n [\prod_i t_i(\theta)]}{\prod_n [f_n(\theta) \prod_{i \neq n} t_i(\theta)]} \\
&= \sum_n \int \mathrm{d}\theta q(\theta) \log \frac{\prod_i t_i(\theta)}{f_n(\theta) \prod_{i \neq n} t_i(\theta)} \\
&= \sum_n \mathrm{KL}[q(\theta)||\tilde{p}_n(\theta)],
\end{aligned}$$

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, $\mathrm{KL}[q(\theta)||p(\theta)]$.

Appendix C. Some Relevant Linear Algebra and Function Expansion Identities

The Woodbury matrix identity or Woodbury formula is:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}. \tag{9}$$

In general, C need not be invertible, we can use the Binomial inverse theorem,

$$(A + UCV)^{-1} = A^{-1} - A^{-1}UC(C + CVA^{-1}UC)^{-1}CVA^{-1}. \tag{10}$$

When C is an identity matrix and U and V are vectors, the Woodbury identity can be shortened and become the Sherman-Morrison formula,

$$(A + uv^\intercal)^{-1} = A^{-1} - \frac{A^{-1}uv^\intercal A^{-1}}{1 + v^\intercal A^{-1}u}. \tag{11}$$

Another useful identity is the matrix determinant lemma,

$$\det(A + uv^\intercal) = (1 + v^\intercal A^{-1}u)\det(A). \tag{12}$$

The above theorem can be extend for matrices U and V,

$$\det(A + UV^\intercal) = \det(\mathrm{I} + V^\intercal A^{-1}U)\det(A). \tag{13}$$

We also make use of the following Maclaurin series,

$$\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots, \tag{14}$$

$$\text{and} \quad \log(1 + x) = x - \frac{x^2}{2} + \frac{x^3}{3} + \cdots. \tag{15}$$

Appendix D. 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

$$q(f) = p(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u}) = p(f_{\neq \mathbf{u}}|\mathbf{u})\frac{\exp(\theta_{\mathbf{u}}^{\intercal}\phi(\mathbf{u}))}{\mathcal{Z}(\theta_{\mathbf{u}})},$$

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,

$$\begin{aligned}
\mathcal{F}_{\mathrm{KL}} &= \mathrm{KL}[\tilde{p}(f)||q(f)] \\
&= \int \tilde{p}(f)\log\frac{\tilde{p}(f)}{q(f)}\mathrm{d}f \\
&= \langle \log \tilde{p}(f)\rangle_{\tilde{p}(f)} - \langle \log p(f_{\neq \mathbf{u}}|\mathbf{u})\rangle_{\tilde{p}(f)} - \theta_{\mathbf{u}}^{\intercal}\langle \phi(\mathbf{u})\rangle_{\tilde{p}(f)} + \log \mathcal{Z}(\theta_{\mathbf{u}}).
\end{aligned}$$

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,

$$\begin{aligned}
0 = \frac{\mathrm{d}\mathcal{F}_{\mathrm{KL}}}{\mathrm{d}\theta_{\mathbf{u}}} &= -\langle \phi(\mathbf{u})\rangle_{\tilde{p}(f)} + \frac{\mathrm{d}\log \mathcal{Z}(\theta_{\mathbf{u}})}{\mathrm{d}\theta_{\mathbf{u}}} \\
&= -\langle \phi(\mathbf{u})\rangle_{\tilde{p}(f)} + \langle \phi(\mathbf{u})\rangle_{q(\mathbf{u})},
\end{aligned}$$

therefore, $\langle \phi(\mathbf{u})\rangle_{\tilde{p}(f)} = \langle \phi(\mathbf{u})\rangle_{q(\mathbf{u})}$. 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 inducing points* $\mathbf{u}$.[4]

Appendix E. 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,

$$\begin{aligned}
\mathbf{m}_{\mathbf{u}} &= \mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}f}^{\backslash n}\frac{\mathrm{d}\log \mathcal{Z}_{\mathrm{tilted},n}}{\mathrm{d}\mathbf{m}_{f}^{\backslash n}}, \\
\mathbf{V}_{\mathbf{u}} &= \mathbf{V}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}f}^{\backslash n}\frac{\mathrm{d}^2\log \mathcal{Z}_{\mathrm{tilted},n}}{\mathrm{d}\mathbf{m}_{f}^{\backslash n,2}}\mathbf{V}_{f\mathbf{u}}^{\backslash n},
\end{aligned}$$

4. We can show that this condition gives the minimum of $\mathcal{F}_{\mathrm{KL}}$ by computing the second derivative.

where $\mathcal{Z}_{\text{tilted},n}$ is the normaliser of the tilted distribution,

$$\begin{aligned}\mathcal{Z}_{\text{tilted},n} &= \int q^{\backslash n}(f)p(y_n|f)\mathrm{d}f \\ &= \int q^{\backslash n}(f)p(y_n|f_n)\mathrm{d}f \\ &= \int q^{\backslash n}(f_n)p(y_n|f_n)\mathrm{d}f_n.\end{aligned}$$

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,

$$\mathbf{m}_{\mathbf{u}} = \mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}f_n}^{\backslash n}\frac{\mathrm{d}\log\mathcal{Z}_{\text{tilted},n}}{\mathrm{d}m_{f_n}^{\backslash n}}, \tag{16}$$

$$\mathbf{V}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}f_n}^{\backslash n}\frac{\mathrm{d}^2\log\mathcal{Z}_{\text{tilted},n}}{\mathrm{d}m_{f_n}^{\backslash n,2}}\mathbf{V}_{f_n\mathbf{u}}^{\backslash n}. \tag{17}$$

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,

$$\begin{aligned}\frac{\mathrm{d}\log\mathcal{Z}_{\text{tilted},n}}{\mathrm{d}\mathbf{m}_{\mathbf{u}}^{\backslash n}} &= \frac{\mathrm{d}\log\mathcal{Z}_{\text{tilted},n}}{\mathrm{d}m_{f_n}^{\backslash n}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}f_n}, \\ \frac{\mathrm{d}\log\mathcal{Z}_{\text{tilted},n}}{\mathrm{d}\mathbf{V}_{\mathbf{u}}^{\backslash n}} &= \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}f_n}\frac{\mathrm{d}^2\log\mathcal{Z}_{\text{tilted},n}}{\mathrm{d}m_{f_n}^{\backslash n,2}}\mathbf{K}_{f_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}.\end{aligned}$$

Substituting these results back in Equations (16) and (17), we obtain

$$\mathbf{m}_{\mathbf{u}} = \mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}}^{\backslash n}\frac{\mathrm{d}\log\mathcal{Z}_{\text{tilted},n}}{\mathrm{d}\mathbf{m}_{\mathbf{u}}^{\backslash n}}, \tag{18}$$

$$\mathbf{V}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}}^{\backslash n}\frac{\mathrm{d}^2\log\mathcal{Z}_{\text{tilted},n}}{\mathrm{d}\mathbf{m}_{\mathbf{u}}^{\backslash n,2}}\mathbf{V}_{\mathbf{u}}^{\backslash n}. \tag{19}$$

Therefore, using Equations (16) and (17), or Equations (18) and (19) are equivalent in our approximation settings.

Appendix F. 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.

F.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,

$$m_{\mathrm{f}} = \mathbf{K}_{\mathrm{f}\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}_{2,\mathbf{u}}^{-1}\mathbf{T}_{1,\mathbf{u}} = \mathbf{K}_{\mathrm{f}\mathbf{u}}\gamma,$$
$$V_{\mathrm{ff}'} = \mathbf{K}_{\mathrm{ff}'} - \mathbf{Q}_{\mathrm{ff}'} + \mathbf{K}_{\mathrm{f}\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}_{2,\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}\mathrm{f}'} = \mathbf{K}_{\mathrm{ff}'} - \mathbf{K}_{\mathrm{f}\mathbf{u}}\beta\mathbf{K}_{\mathbf{u}\mathrm{f}'},$$

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

$$m_{\mathrm{f}}^{\backslash n} = \mathbf{K}_{\mathrm{f}\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}_{2,\mathbf{u}}^{\backslash n,-1}\mathbf{T}_{1,\mathbf{u}}^{\backslash n} = \mathbf{K}_{\mathrm{f}\mathbf{u}}\gamma^{\backslash n},$$
$$V_{\mathrm{ff}'}^{\backslash n} = \mathbf{K}_{\mathrm{ff}'} - \mathbf{Q}_{\mathrm{ff}'} + \mathbf{K}_{\mathrm{f}\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}\backslash n, -1_{2,\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}\mathrm{f}'} = \mathbf{K}_{\mathrm{ff}'} - \mathbf{K}_{\mathrm{f}\mathbf{u}}\beta^{\backslash n}\mathbf{K}_{\mathbf{u}\mathrm{f}'}.$$

Projection step: The central step in Power EP is the projection step. Obtaining the new approximate unnormalised 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,

$$\mathbf{m}_{\mathbf{u}} = \mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}\frac{\mathrm{d}\log\tilde{Z}_n}{\mathrm{d}m_{\mathrm{f}_n}^{\backslash n}}, \tag{20}$$

$$\mathbf{V}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}\frac{\mathrm{d}^2\log\tilde{Z}_n}{\mathrm{d}(m_{\mathrm{f}_n}^{\backslash n})^2}\mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}, \tag{21}$$

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,

$$\mathbf{T}_{1,n,\mathrm{new}} = \mathbf{V}_{\mathbf{u}}^{-1}\mathbf{m}_{\mathbf{u}} - \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{m}_{\mathbf{u}}^{\backslash n}, \tag{22}$$

$$\mathbf{T}_{2,n,\mathrm{new}} = \mathbf{V}_{\mathbf{u}}^{-1} - \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}, \tag{23}$$

$$z_n^\alpha = \tilde{Z}_n \exp(\mathcal{G}_{q_*^{\backslash n}(\mathbf{u})} - \mathcal{G}_{q^*(\mathbf{u})}),$$

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 Equation (9) and Equation (21), we have,

$$\mathbf{V}_{\mathbf{u}}^{-1} - \mathbf{V}_{\mathbf{u}}^{\backslash n,-1} = -\mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}\left[d_2^{-1} + \mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}\mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}\right]^{-1}\mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}\mathbf{V}_{\mathbf{u}}^{\backslash n,-1}. \tag{24}$$

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 Equation (24) and Equation (23) gives

$$\mathbf{T}_{2,n,\mathrm{new}} = \mathbf{w}_n\alpha v_n^{-1}\mathbf{w}_n^{\intercal}. \tag{25}$$

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,

$$\begin{aligned}
\mathbf{w}_n &= \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n} \\
&= (\mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uu}}\beta^{\backslash n}\mathbf{K}_{\mathbf{uu}})^{-1}(\mathbf{K}_{\mathbf{u}\mathrm{f}_n} - \mathbf{K}_{\mathbf{uu}}\beta^{\backslash n}\mathbf{K}_{\mathbf{u}\mathrm{f}_n}) \\
&= \mathbf{K}_{\mathbf{uu}}^{-1}(\mathrm{I} - \mathbf{K}_{\mathbf{uu}}\beta^{\backslash n})^{-1}(\mathrm{I} - \mathbf{K}_{\mathbf{uu}}\beta^{\backslash n})\mathbf{K}_{\mathbf{u}\mathrm{f}_n} \\
&= \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}\mathrm{f}_n}.
\end{aligned}$$

Using Equation (20) an Equation (25) gives,

$$\begin{aligned}
\mathbf{V}_{\mathbf{u}}^{-1}\mathbf{m}_{\mathbf{u}} &= (\mathbf{V}_{\mathbf{u}}^{\backslash n,-1} + \mathbf{w}_n\alpha v_n^{-1}\mathbf{w}_n^{\intercal})(\mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1) \\
&= \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{w}_n\alpha v_n^{-1}\mathbf{w}_n^{\intercal}\mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1 + \mathbf{w}_n\alpha v_n^{-1}\mathbf{w}_n^{\intercal}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1.
\end{aligned}$$

Substituting this result into Equation (22),

$$\begin{aligned}
\mathbf{T}_{1,n,\mathrm{new}} &= \mathbf{V}_{\mathbf{u}}^{-1}\mathbf{m}_{\mathbf{u}} - \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{m}_{\mathbf{u}}^{\backslash n} \\
&= \mathbf{w}_n\alpha v_n^{-1}\mathbf{w}_n^{\intercal}\mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1 + \mathbf{w}_n\alpha v_n^{-1}\mathbf{w}_n^{\intercal}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1 \\
&= \mathbf{w}_n\alpha v_n^{-1}\left(\mathbf{w}_n^{\intercal}\mathbf{m}_{\mathbf{u}}^{\backslash n} + d_1 v_n/\alpha + \mathbf{w}_n^{\intercal}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1\right).
\end{aligned}$$

Let $\mathbf{T}_{1,n,\mathrm{new}} = \mathbf{w}_n\alpha v_n^{-1}g_n$, we obtain,

$$g_n = -\frac{d_1}{d_2} + \mathbf{K}_{\mathrm{f}_n\mathbf{u}}\gamma^{\backslash n}.$$

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,

$$\begin{aligned}
t_n(\mathbf{u}) &= \tilde{\mathcal{N}}(\mathbf{u};z_n,\mathbf{T}_{1,n},\mathbf{T}_{2,n}) \\
&= z_n\exp(\mathbf{u}^{\intercal}\mathbf{T}_{1,n} - \frac{1}{2}\mathbf{u}^{\intercal}\mathbf{T}_{2,n}\mathbf{u}) \\
&= z_n\exp(\mathbf{u}^{\intercal}\mathbf{w}_n v_n^{-1}g_n - \frac{1}{2}\mathbf{u}^{\intercal}\mathbf{w}_n v_n^{-1}\mathbf{w}_n^{\intercal}\mathbf{u}).
\end{aligned}$$

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 $\mathbf{T}_{2,n}$.

F.2 Projection

We now recall the update equations in the projection step (Equations (20) and (21)):

$$\mathbf{m}_{\mathbf{u}} = \mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n} d_1,$$
$$\mathbf{V}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n} d_2 \mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}.$$

Note that:

$$\mathbf{m}_{\mathbf{u}} = \mathbf{K}_{\mathbf{uu}}\gamma,$$
$$\mathbf{V}_{\mathbf{u}} = \mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uu}}\beta\mathbf{K}_{\mathbf{uu}},$$

and

$$\mathbf{m}_{\mathbf{u}}^{\backslash n} = \mathbf{K}_{\mathbf{uu}}\gamma^{\backslash n},$$
$$\mathbf{V}_{\mathbf{u}}^{\backslash n} = \mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uu}}\beta^{\backslash n}\mathbf{K}_{\mathbf{uu}}.$$

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,

$$\begin{aligned}
\gamma &= \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m}_{\mathbf{u}} \\
&= \mathbf{K}_{\mathbf{uu}}^{-1}(\mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n} d_1) \\
&= \gamma^{\backslash n} + \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n} d_1, \text{ and} && (26) \\
\beta &= \mathbf{K}_{\mathbf{uu}}^{-1}(\mathbf{K}_{\mathbf{uu}} - \mathbf{V}_{\mathbf{u}})\mathbf{K}_{\mathbf{uu}}^{-1} \\
&= \mathbf{K}_{\mathbf{uu}}^{-1}(\mathbf{K}_{\mathbf{uu}} - \mathbf{V}_{\mathbf{u}}^{\backslash n} - \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n} d_2 \mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n})\mathbf{K}_{\mathbf{uu}}^{-1} \\
&= \beta^{\backslash n} - \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n} d_2 \mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}\mathbf{K}_{\mathbf{uu}}^{-1}. && (27)
\end{aligned}$$

F.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,

$$q^{\backslash n}(f) \propto \frac{q(f)}{t_n^{\alpha}(\mathbf{u})} = \frac{p(f_{\neq\mathbf{u}}|\mathbf{u})q(\mathbf{u})}{t_n^{\alpha}(\mathbf{u})} = p(f_{\neq\mathbf{u}}|\mathbf{u})q^{\backslash n}(\mathbf{u}).$$

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,

$$\begin{aligned}
&\text{Projection:} \quad q(f) \approx q^{\backslash n}(f)p(y_n|\mathrm{f}_n), \\
&\text{Deletion:} \quad q^{\backslash n}(f) \propto q(f)\frac{1}{t_n^{\alpha}(\mathbf{u})}.
\end{aligned}$$

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 (26) and (27), 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, $\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}$ to $\mathbf{V}_{\mathbf{u}\mathrm{f}_n}$, $\gamma^{\backslash n}$ 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

$$\log \tilde{Z}_n = \log \int q(f) \frac{1}{t_n^{\alpha}(\mathbf{u})} \mathrm{d}f.$$

Now, we note

$$\begin{aligned}
\frac{1}{t_n(\mathbf{u})} &\propto \frac{1}{\mathcal{N}^{\alpha}(\mathbf{w}_n^{\intercal}\mathbf{u}; g_n, v_n)} \\
&\propto \frac{1}{\exp\left(-\frac{\alpha}{2} v_n^{-1} \left(\mathbf{w}_n^{\intercal}\mathbf{u} - g_n\right)^2\right)} \\
&= \exp\left(\frac{1}{2}\alpha v_n^{-1} \left(\mathbf{w}_n^{\intercal}\mathbf{u} - g_n\right)^2\right) \\
&\propto \mathcal{N}(\mathbf{w}_n^{\intercal}\mathbf{u}; g_n, -v_n/\alpha).
\end{aligned}$$

Then we obtain the derivatives of $\log \tilde{Z}_n$

$$\begin{aligned}
\tilde{d}_2 &= \frac{\mathrm{d}^2 \log \tilde{Z}_n}{\mathrm{d}m_{\mathrm{f}_n}^2} = -\left[\mathbf{K}_{\mathrm{f}_n,\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathrm{f}_n} - \mathbf{K}_{\mathrm{f}_n,\mathbf{u}}\beta\mathbf{K}_{\mathbf{u},\mathrm{f}_n} - v_n/\alpha\right]^{-1}, \\
\tilde{d}_1 &= \frac{\mathrm{d}\log \tilde{Z}_n}{\mathrm{d}m_{\mathrm{f}_n}} = (\mathbf{K}_{\mathrm{f}_n,\mathbf{u}}\gamma - g_n)\tilde{d}_2.
\end{aligned}$$

Putting the above results together, we obtain,

$$\begin{aligned}
\gamma^{\backslash n} &= \gamma + \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}\tilde{d}_1, \quad \text{and} \\
\beta^{\backslash n} &= \beta - \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}\tilde{d}_2\mathbf{V}_{\mathrm{f}_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}.
\end{aligned}$$

F.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 $\beta^{\backslash n}$

$$\gamma^{\backslash n} = \gamma + \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}\tilde{d}_1, \text{ and}$$
$$\beta^{\backslash n} = \beta - \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}\tilde{d}_2\mathbf{V}_{\mathrm{f}_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}.$$

(b) Projection step: find γ and β

$$\gamma = \gamma^{\backslash n} + \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1,$$
$$\beta = \beta^{\backslash n} - \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_2\mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}\mathbf{K}_{\mathbf{uu}}^{-1}.$$

(c) Update step: find $g_{n,\text{new}}$ and $v_{n,\text{new}}$

$$g_{n,\text{new}} = -\frac{d_1}{d_2} + \mathbf{K}_{\mathrm{f}_n\mathbf{u}}\gamma^{\backslash n},$$
$$v_{n,\text{new}} = -d_2^{-1} - \mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}\mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n},$$

and parameters for the full factor,

$$v_n \leftarrow (v_{n,\text{new}}^{-1} + (1-\alpha)v_n^{-1})^{-1},$$
$$g_n \leftarrow v_n(g_{n,\text{new}}v_{n,\text{new}}^{-1} + (1-\alpha)g_n v_n^{-1}).$$

Appendix G. 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,

$$\mathcal{F} = \mathcal{G}(q_*(\mathbf{u})) - \mathcal{G}(p_*(\mathbf{u})) + \frac{1}{\alpha}\sum_n \left[\log \mathcal{Z}_{\text{tilted},n} + \mathcal{G}(q_*^{\backslash n}(\mathbf{u})) - \mathcal{G}(q_*(\mathbf{u}))\right],$$

where $\mathcal{G}(q_*(\mathbf{u}))$ is the log-normaliser of the approximate posterior, that is,

$$\begin{aligned}\mathcal{G}(q_*(\mathbf{u})) &= \log \int p(f_{\neq\mathbf{u}}|\mathbf{u})\exp(\theta_{\mathbf{u}}^\intercal\phi(\mathbf{u}))\mathrm{d}f_{\neq\mathbf{u}}\mathrm{d}\mathbf{u} \\ &= \log \int \exp(\theta_{\mathbf{u}}^\intercal\phi(\mathbf{u}))\mathrm{d}\mathbf{u} \\ &= \frac{M}{2}\log(2\pi) + \frac{1}{2}\log|\mathbf{V}| + \frac{1}{2}\mathbf{m}^\intercal\mathbf{V}^{-1}\mathbf{m}, \end{aligned} \tag{28}$$

where $\mathbf{m}$ and $\mathbf{V}$ are the mean and covariance of the posterior distribution over $\mathbf{u}$, respectively. Similarly,

$$\mathcal{G}(q_*^{\backslash n}(\mathbf{u})) = \frac{M}{2}\log(2\pi) + \frac{1}{2}\log|\mathbf{V}_{\text{cav},n}| + \frac{1}{2}\mathbf{m}_{\text{cav},n}^\intercal\mathbf{V}_{\text{cav},n}^{-1}\mathbf{m}_{\text{cav},n}, \tag{29}$$
$$\text{and } \mathcal{G}(p_*(\mathbf{u})) = \frac{M}{2}\log(2\pi) + \frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}|.$$

Finally, $\log \mathcal{Z}_{\text{tilted},n}$ is the log-normalising constant of the tilted distribution,

$$\begin{aligned}\log \mathcal{Z}_{\text{tilted}} &= \log \int q_{\text{cav}}(f) p^{\alpha}(y_n|f) \mathrm{d}f \\ &= \log \int p(f_{\neq \mathbf{u}}|\mathbf{u}) q_{\text{cav}}(\mathbf{u}) p^{\alpha}(y_n|f) \mathrm{d}f_{\neq \mathbf{u}} \mathrm{d}\mathbf{u} \\ &= \log \int p(f_n|\mathbf{u}) q_{\text{cav}}(\mathbf{u}) p^{\alpha}(y_n|f_n) \mathrm{d}f_n \mathrm{d}\mathbf{u}. \end{aligned} \tag{30}$$

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,

$$\mathbf{V}^{-1} = \mathbf{K}_{\mathbf{uu}}^{-1} + \sum_i \mathbf{w}_i \tau_i \mathbf{w}_i^{\intercal}, \tag{31}$$

$$\mathbf{V}^{-1}\mathbf{m} = \sum_i \mathbf{w}_i \tau_i \tilde{y}_i, \tag{32}$$

$$\mathbf{V}_{\text{cav},n}^{-1} = \mathbf{V}^{-1} - \alpha \mathbf{w}_n \tau_n \mathbf{w}_n^{\intercal}, \tag{33}$$

$$\mathbf{V}_{\text{cav},\mathbf{n}}{}^{-1}\mathbf{m}_{\text{cav},n} = \mathbf{V}^{-1}\mathbf{m} - \alpha \mathbf{w}_n \tau_n g_n. \tag{34}$$

Note that $\tau_i \coloneqq v_i^{-1}$. Using Equation (11) and Equation (33) gives,

$$\mathbf{V}_{\text{cav},n} = \mathbf{V} + \frac{\mathbf{V}\mathbf{w}_n \alpha \tau_n \mathbf{w}_n^{\intercal}\mathbf{V}}{1 - \mathbf{w}_n^{\intercal}\alpha\tau_n \mathbf{V}\mathbf{w}_n}. \tag{35}$$

Using Equation (12) and Equation (33) gives,

$$\log\det(\mathbf{V}_{\text{cav},n}) = \log\det(\mathbf{V}) - \log(1 - \mathbf{w}_n^{\intercal}\alpha\tau_n \mathbf{V}\mathbf{w}_n). \tag{36}$$

Substituting Equation (35) and Equation (36) back to Equation (29) results in,

$$\begin{aligned}\mathcal{G}(q_*^{\setminus n}(\mathbf{u})) = &\frac{M}{2}\log(2\pi) + \frac{1}{2}\log\det(\mathbf{V}) + \frac{1}{2}\mathbf{m}^{\intercal}\mathbf{V}^{-1}\mathbf{m} \\ &- \frac{1}{2}\log(1 - \mathbf{w}_n^{\intercal}\alpha\tau_n \mathbf{V}\mathbf{w}_n) + \frac{1}{2}\frac{\mathbf{m}^{\intercal}\mathbf{w}_n \alpha\tau_n \mathbf{w}_n^{\intercal}\mathbf{m}}{1 - \mathbf{w}_n^{\intercal}\alpha\tau_n \mathbf{V}\mathbf{w}_n} \\ &+ \frac{1}{2} g_n \alpha\tau_n \mathbf{w}_n^{\intercal}\mathbf{V}_{\text{cav},n}\mathbf{w}_n \alpha\tau_n g_n - g_n \alpha\tau_n \mathbf{w}_n^{\intercal}\mathbf{V}_{\text{cav},n}\mathbf{V}^{-1}\mathbf{m}. \end{aligned} \tag{37}$$

We now plug the above result back into the approximate marginal likelihood, yielding,

$$\begin{aligned}\mathcal{F} = &\frac{1}{2}\log|\mathbf{V}| + \frac{1}{2}\mathbf{m}^{\intercal}\mathbf{V}^{-1}\mathbf{m} - \frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}| + \frac{1}{\alpha}\sum_n \log \mathcal{Z}_{\text{tilted},n} \\ &+ \sum_n \Bigg[-\frac{1}{2\alpha}\log(1 - \mathbf{w}_n^{\intercal}\alpha\tau_n \mathbf{V}\mathbf{w}_n) + \frac{1}{2}\frac{\mathbf{m}^{\intercal}\mathbf{w}_n \tau_n \mathbf{w}_n^{\intercal}\mathbf{m}}{1 - \mathbf{w}_n^{\intercal}\alpha\tau_n \mathbf{V}\mathbf{w}_n} \\ &+ \frac{1}{2} g_n \tau_n \mathbf{w}_n^{\intercal}\mathbf{V}_{\text{cav},n}\mathbf{w}_n \alpha\tau_n g_n - g_n \tau_n \mathbf{w}_n^{\intercal}\mathbf{V}_{\text{cav},n}\mathbf{V}^{-1}\mathbf{m} \Bigg]. \end{aligned} \tag{38}$$

G.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 $\log \mathcal{Z}_{\text{tilted},n}$,

$$\log \mathcal{Z}_{\text{tilted},n} = -\frac{\alpha}{2}\log(2\pi\sigma_y^2) + \frac{1}{2}\log(\sigma_y^2) - \frac{1}{2}\log(\alpha v_n + \sigma_y^2) - \frac{1}{2}\frac{(y_n - \mu_n)^2}{v_n + \sigma_y^2/\alpha},$$

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,

$$\begin{aligned}
\mathcal{F} &= \frac{1}{2}\log|\mathbf{V}| + \frac{1}{2}\mathbf{m}^\intercal \mathbf{V}^{-1}\mathbf{m} - \frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}| \\
&\quad + \sum_n \left[-\frac{1}{2}\log(2\pi\sigma_y^2) + \frac{1}{2\alpha}\log\sigma_y^2 - \frac{1}{2\alpha}\log d_n - \frac{y_n^2}{2d_n} \right] \\
&= -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{D} + \mathbf{Q}_{\mathbf{ff}}| - \frac{1}{2}\mathbf{y}^T(\mathbf{D} + \mathbf{Q}_{\mathbf{ff}})^{-1}\mathbf{y} - \frac{1-\alpha}{2\alpha}\sum_n \log(\frac{d_n}{\sigma_y^2}),
\end{aligned}$$

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 likelihood,

$$\mathcal{F} = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{D} + \mathbf{Q}_{\mathbf{ff}}| - \frac{1}{2}\mathbf{y}^T(\mathbf{D} + \mathbf{Q}_{\mathbf{ff}})^{-1}\mathbf{y},$$

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,

$$\lim_{\alpha \to 0} \frac{1-\alpha}{2\alpha}\sum_n \log(\frac{d_n}{\sigma_y^2}) = \frac{1}{2}\sum_n \lim_{\alpha \to 0} \frac{\log(1 + \alpha\frac{g_n}{\sigma_y^2})}{\alpha} = \frac{\sum_n h_n}{2\sigma_y^2},$$

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,

$$\mathcal{F} = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\sigma_y^2 \mathrm{I} + \mathbf{Q}_{\mathbf{ff}}| - \frac{1}{2}\mathbf{y}^T(\sigma_y^2 \mathrm{I} + \mathbf{Q}_{\mathbf{ff}})^{-1}\mathbf{y} - \frac{\sum_n h_n}{2\sigma_y^2},$$

which is the variational lower bound of Titsias (Titsias, 2009).

G.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 Equation (38):

$$\lim_{\alpha\to 0} -\frac{1}{2\alpha}\log(1-\mathbf{w}_n^\intercal \alpha\tau_n \mathbf{V}\mathbf{w}_n) = \frac{1}{2}\mathbf{w}_n^\intercal \tau_n \mathbf{V}\mathbf{w}_n, \tag{39}$$

$$\frac{1}{2}\frac{\mathbf{m}^\intercal \mathbf{w}_n \tau_n \mathbf{w}_n^\intercal \mathbf{m}}{1-\mathbf{w}_n^\intercal \alpha\tau_n \mathbf{V}\mathbf{w}_n}\bigg|_{\alpha=0} = \frac{1}{2}\mathbf{m}^\intercal \mathbf{w}_n \tau_n \mathbf{w}_n^\intercal \mathbf{m}, \tag{40}$$

$$\frac{1}{2} g_n \tau_n \mathbf{w}_n^\intercal \mathbf{V}_{\text{cav},n}\mathbf{w}_n \alpha\tau_n g_n\bigg|_{\alpha=0} = 0, \tag{41}$$

$$-g_n\tau_n \mathbf{w}_n^\intercal \mathbf{V}_{\text{cav},n}\mathbf{V}^{-1}\mathbf{m}\bigg|_{\alpha=0} = -g_n\tau_n\mathbf{w}_n^\intercal \mathbf{m}. \tag{42}$$

We turn our attention to $\log \mathcal{Z}_{\text{tilted},n}$. First, we expand $p^\alpha(y_n|f_n)$ using Equation (14):

$$\begin{aligned} p^\alpha(y_n|f_n) &= \exp(\alpha \log p(y_n|f_n)) \\ &= 1 + \alpha\log p(y_n|f_n) + \xi(\alpha^2). \end{aligned}$$

Substituting this result back into $\log \mathcal{Z}\text{tilted}/\alpha$ gives,

$$\begin{aligned} \frac{1}{\alpha}\log \mathcal{Z}_{\text{tilted}} &= \frac{1}{\alpha}\log \int p(f_n|\mathbf{u})q_{\text{cav}}(\mathbf{u})p^\alpha(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u} \\ &= \frac{1}{\alpha}\log \int p(f_n|\mathbf{u})q_{\text{cav}}(\mathbf{u})[1+\alpha\log p(y_n|f_n)+\xi(\alpha^2)]\mathrm{d}f_n\mathrm{d}\mathbf{u} \\ &= \frac{1}{\alpha}\log\left[1+\alpha\int p(f_n|\mathbf{u})q_{\text{cav}}(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u}+\alpha^2\xi(1)\right] \\ &= \frac{1}{\alpha}\left[\alpha\int p(f_n|\mathbf{u})q_{\text{cav}}(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u}+\alpha^2\xi(1)\right] \\ &= \int p(f_n|\mathbf{u})q_{\text{cav}}(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u}+\alpha\xi(1). \end{aligned}$$

Therefore,

$$\lim_{\alpha\to 0}\frac{1}{\alpha}\log \mathcal{Z}_{\text{tilted}} = \int p(f_n|\mathbf{u})q(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u}. \tag{43}$$

Putting these results into Equation (38), we obtain,

$$\begin{aligned}
\mathcal{F} &= \frac{1}{2}\log|\mathbf{V}| + \frac{1}{2}\mathbf{m}^\intercal\mathbf{V}^{-1}\mathbf{m} - \frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}| \\
&\quad + \sum_n \frac{1}{2}\mathbf{w}_n^\intercal\tau_n\mathbf{V}\mathbf{w}_n + \frac{1}{2}\mathbf{m}^\intercal\mathbf{w}_n\tau_n\mathbf{w}_n^\intercal\mathbf{m} - g_n\tau_n\mathbf{w}_n^\intercal\mathbf{m} + \int p(f_n|\mathbf{u})q(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u} \\
&= \frac{1}{2}\log|\mathbf{V}| + \frac{1}{2}\mathbf{m}^\intercal\mathbf{V}^{-1}\mathbf{m} - \frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}| + \frac{1}{2}\mathbf{m}^\intercal(\mathbf{V}^{-1} - \mathbf{K}_{\mathbf{uu}}^{-1})\mathbf{m} - \mathbf{m}^\intercal\mathbf{V}^{-1}\mathbf{m} \\
&\qquad + \sum_n \frac{1}{2}\mathbf{w}_n^\intercal\tau_n\mathbf{V}\mathbf{w}_n + \int p(f_n|\mathbf{u})q(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u} \\
&= \frac{1}{2}\log|\mathbf{V}| - \frac{1}{2}\mathbf{m}^\intercal\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m} - \frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}| + \sum_n \frac{1}{2}\mathbf{w}_n^\intercal\tau_n\mathbf{V}\mathbf{w}_n \\
&\qquad + \sum_n \int p(f_n|\mathbf{u})q(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u}. \qquad (44)
\end{aligned}$$

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),

$$\mathcal{F}_{\mathrm{VFE}} = -\mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})] + \sum_n \int p(f_n|\mathbf{u})q(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u}, \qquad (45)$$

where

$$\begin{aligned}
-\mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})] &= -\frac{1}{2}\mathrm{trace}(\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{V}) - \frac{1}{2}\mathbf{m}^\intercal\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m} + \frac{M}{2} - \frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}| + \frac{1}{2}\log|\mathbf{V}| \\
&= -\frac{1}{2}\mathrm{trace}([\mathbf{V}^{-1} - \sum_n \mathbf{w}_n\tau_n\mathbf{w}_n]\mathbf{V}) - \frac{1}{2}\mathbf{m}^\intercal\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m} \\
&\qquad + \frac{M}{2} - \frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}| + \frac{1}{2}\log|\mathbf{V}| \\
&= \frac{1}{2}\mathrm{trace}(\sum_n \mathbf{w}_n\tau_n\mathbf{w}_n\mathbf{V}) - \frac{1}{2}\mathbf{m}^\intercal\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m} - \frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}| + \frac{1}{2}\log|\mathbf{V}|. \qquad (46)
\end{aligned}$$

Therefore, $\mathcal{F}_{\mathrm{VFE}}$ is identical to the limit of the approximate marginal likelihood provided by power-EP as shown in Equation (44).

Appendix H. The Surrogate Regression Viewpoint

It was written in the main text that it is instructive to view the approximation using pseudo-points as forming a surrogate exact Gaussian process regression problem such that the posterior and the marginal likelihood of this surrogate problem are close to that of the original intractable regression/classification problem. This approximation view is useful and could potentially be used for other intractable probabilistic model, despite that we have not used this view in the practical implementation of the algorithms/PEP procedure discussed in this paper. In this section, we detail the surrogate model and how the parameters of this model can be tuned to match the approximate posterior and approximate marginal likelihood.

We consider the exact GP regression problem with M surrogate observations $\tilde{\mathbf{y}}$ that are formed by linear combinations the pseudo-outputs and additive surrogate Gaussian noise, $\tilde{\mathbf{y}} = \tilde{\mathbf{W}}\mathbf{u} + \tilde{\Sigma}^{1/2}\epsilon$. The exact posterior and log marginal likelihood can be obtained for this model as follows,

$$\begin{aligned}
\tilde{p}(\mathbf{u}|\mathbf{y}) &= \mathcal{N}^{-1}(\mathbf{u}; \tilde{\mathbf{W}}\tilde{\Sigma}^{-1}\tilde{\mathbf{y}}, \mathbf{K}_{\mathbf{uu}}^{-1} + \tilde{\mathbf{W}}^{\intercal}\tilde{\Sigma}^{-1}\tilde{\mathbf{W}}), \\
\log p(\tilde{\mathbf{y}}) &= -\frac{M}{2}\log(2\pi) - \frac{1}{2}(\log|\mathbf{K}_{\mathbf{uu}}^{-1} + \tilde{\mathbf{W}}^{\intercal}\tilde{\Sigma}^{-1}\tilde{\mathbf{W}}| + \log|\mathbf{K}_{\mathbf{uu}}^{-1}| + \log|\tilde{\Sigma}|) \\
&\quad - \frac{1}{2}\tilde{\mathbf{y}}^{\intercal}\tilde{\Sigma}^{-1}\tilde{\mathbf{y}} - \frac{1}{2}\tilde{\mathbf{y}}^{\intercal}\tilde{\Sigma}^{-1}\tilde{\mathbf{W}}(\mathbf{K}_{\mathbf{uu}}^{-1} + \tilde{\mathbf{W}}^{\intercal}\tilde{\Sigma}^{-1}\tilde{\mathbf{W}})^{-1}\tilde{\mathbf{W}}^{\intercal}\tilde{\Sigma}^{-1}\tilde{\mathbf{y}},
\end{aligned}$$

where we have used the matrix inversion lemma and the matrix determinant lemma in the equations above, and that $\mathcal{N}^{-1}$ denotes the Gaussian distribution with natural parameters.

The aim is to show that we can use the above quantities is to match a given approximate posterior $q(\mathbf{u}) = \mathcal{N}^{-1}(\mathbf{u}; \mathbf{S}^{-1}\mathbf{m}, \mathbf{S}^{-1})$ and an approximate marginal likelihood $\mathcal{F}$, that is, $\tilde{p}(\mathbf{u}|\mathbf{y}) = q(\mathbf{u})$ and $\log p(\tilde{\mathbf{y}}) = \mathcal{F}$. Substituting the above results into the constraints leading to the following simplified constraints:

$$\begin{aligned}
\tilde{\mathbf{W}}\tilde{\Sigma}^{-1}\tilde{\mathbf{y}} &= \mathbf{m}, \\
\tilde{\mathbf{W}}^{\intercal}\tilde{\Sigma}^{-1}\tilde{\mathbf{W}} &= \mathbf{R} = \mathbf{K}_{\mathbf{uu}}^{-1} - \mathbf{S}^{-1}, \\
\tilde{\mathbf{y}}^{\intercal}\tilde{\Sigma}^{-1}\tilde{\mathbf{y}} + \log|\tilde{\Sigma}| &= c,
\end{aligned}$$

where c is a constant. Assume that $\mathbf{R}$ is invertible, we can simplified the above results further,

$$\begin{aligned}
\tilde{\Sigma}^{-1/2}\tilde{\mathbf{y}} &= \mathbf{R}^{-1/2}\mathbf{m}, \\
\tilde{\Sigma}^{-1/2}\tilde{\mathbf{W}} &= \mathbf{R}^{\intercal/2}, \\
\log|\tilde{\Sigma}| &= d,
\end{aligned}$$

where d is a constant. We can choose $\tilde{\Sigma}$, e.g. a diagonal matrix, that satisfies the third equality above. Given $\tilde{\Sigma}$, obtaining $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{W}}$ from the first two equalities is trivial.

Appendix I. Extra Experimental Results

I.1 Comparison Between Various α Values on a Toy Regression Problem

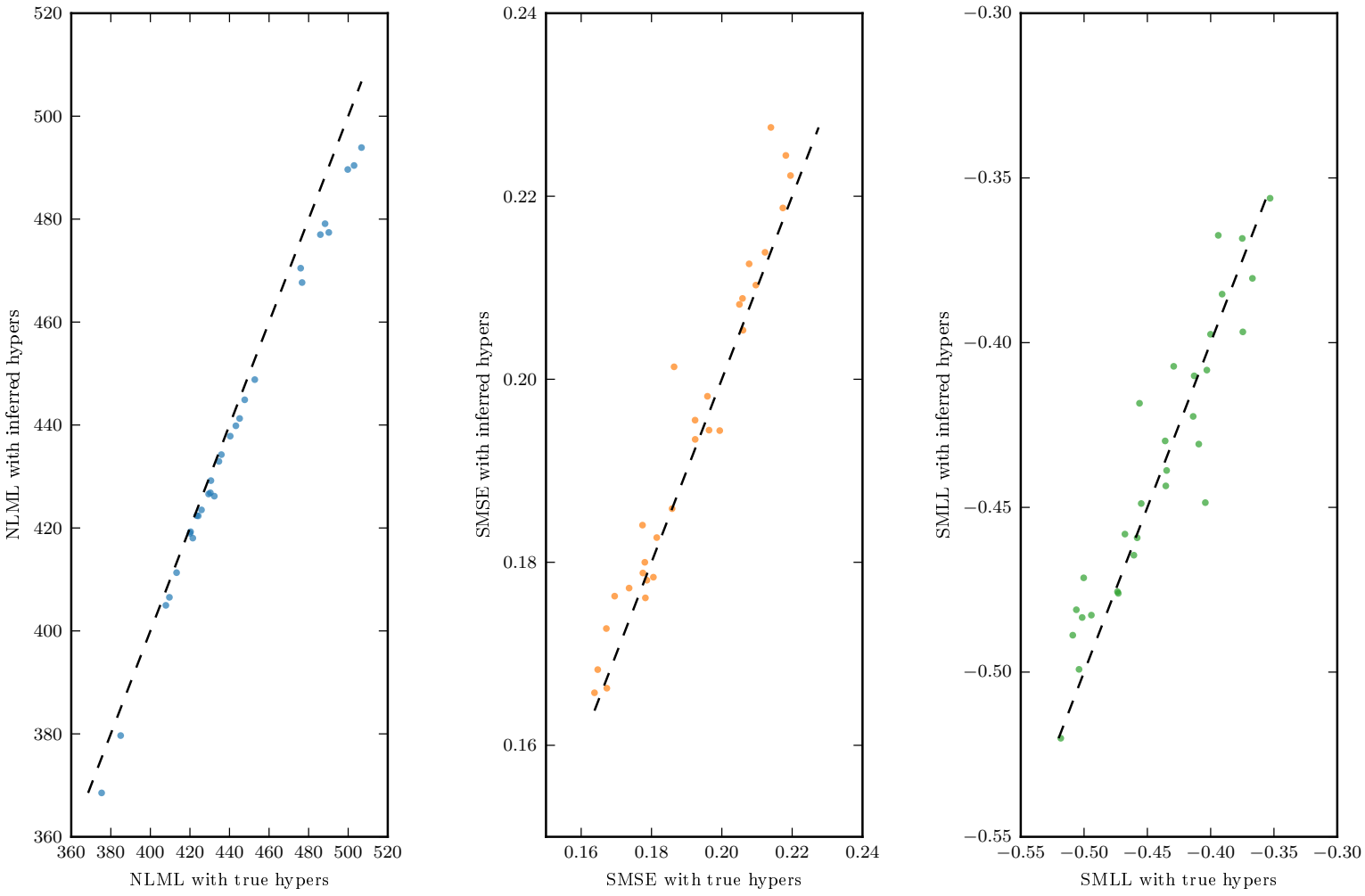

Figure 8: Results on a toy regression problem: Negative log-marginal likelihood, mean squared error and mean log-loss on the test set for full Gaussian process regression on synthetic data sets with *true* hyperparameters and hyperparameters obtained by type-2 ML. Each dot is one trial, i.e. one synthetic data set. The results demonstrate that type-2 maximum likelihood on hyperparameters works well, despite being a little confident on the log-marginal likelihood on the train set.

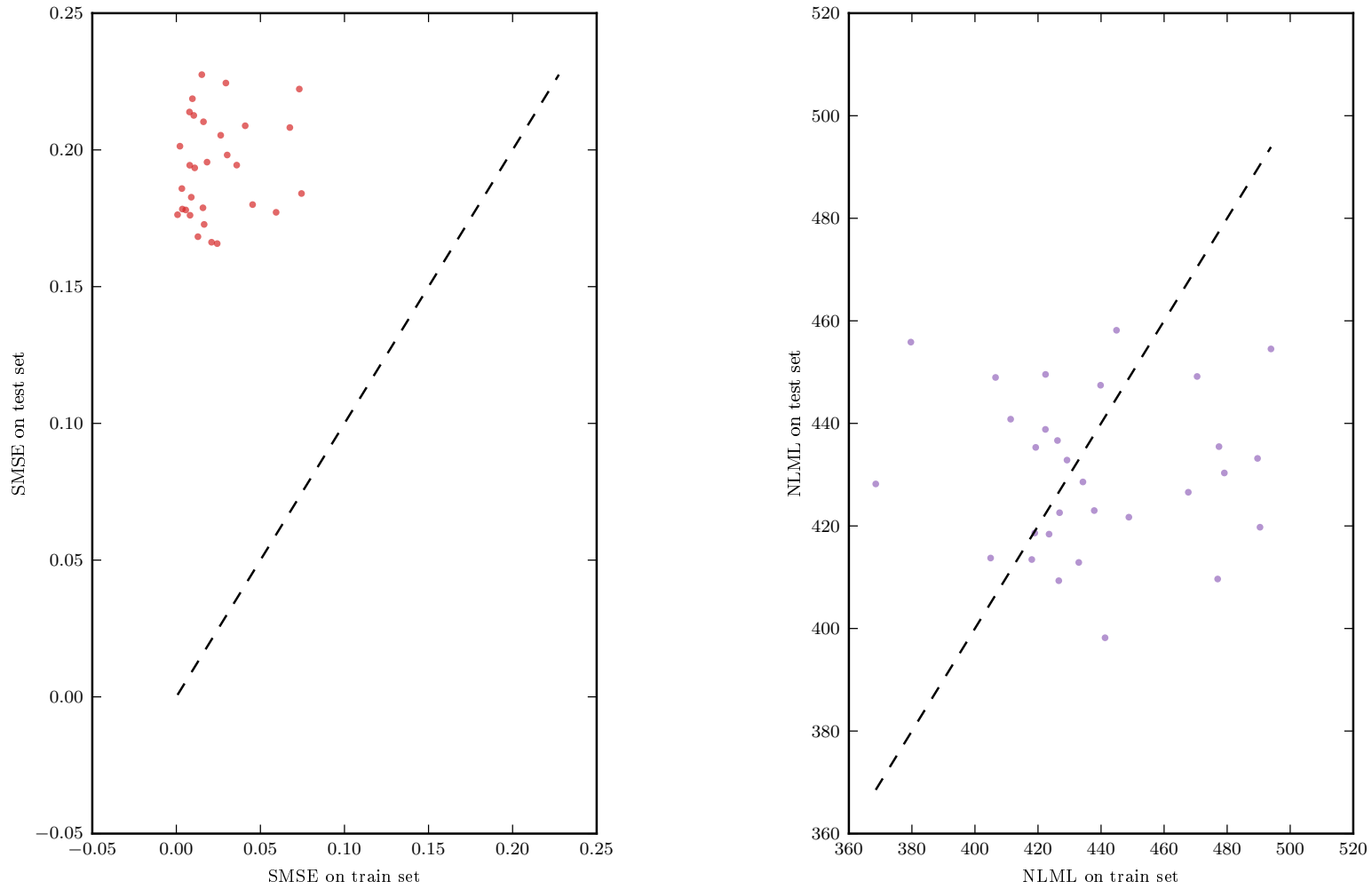

Figure 9: Results on a toy regression problem with 500 training points: Mean squared error and log-likelihood on train and test sets on synthetic data sets with hyperparameters obtained by type-2 ML. In this example, the test error is higher than the training error, as measured by the mean squared error, because the test points and training points are relatively far apart, making the prediction task on the training set easier (interpolation) than on the test set (extrapolation). This is consistent with the results with more training points, shown in Figure 10.

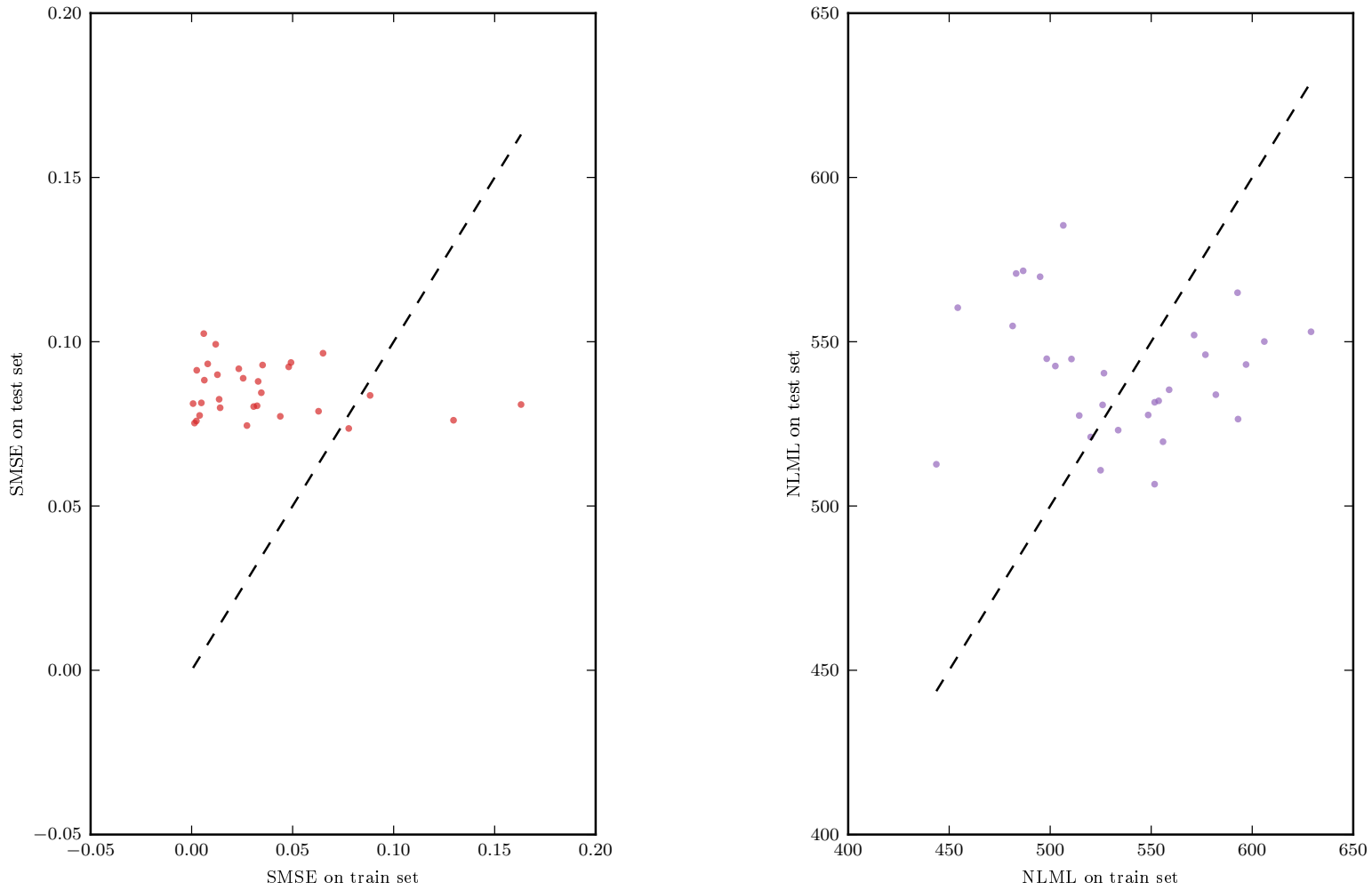

Figure 10: Results on a toy regression problem with 1000 training points: Mean squared error and log-likelihood on train and test sets on synthetic data sets with hyperparameters obtained by type-2 ML. See Figure 9 for a discussion.

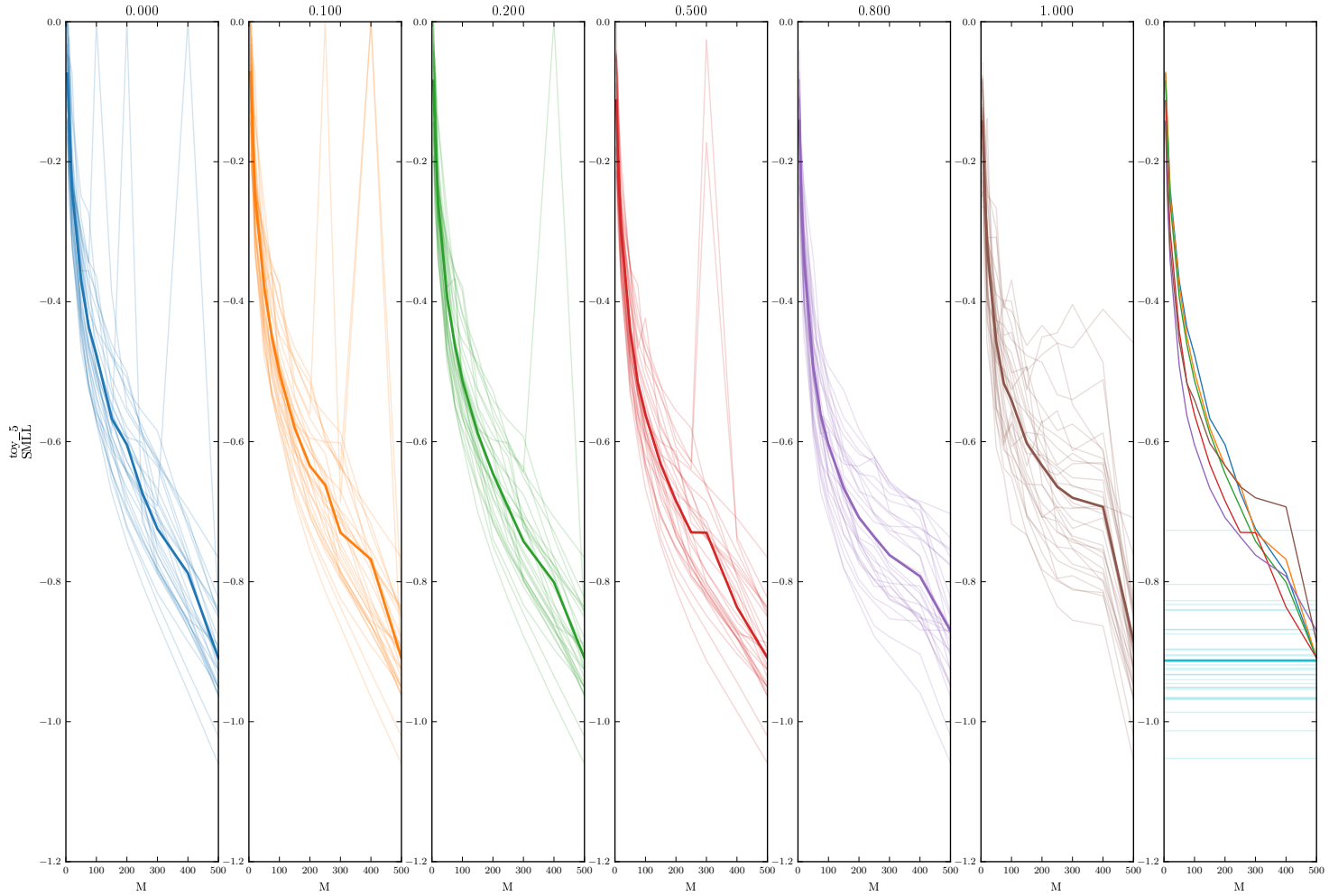

Figure 11: Results on a toy regression problem: Standardsised mean log-loss on the test set for various values of α and various number of pseudo-points M. Each trace is for one split, bold line is the mean. The rightmost figure shows the mean for various α, and the results using GP regression.

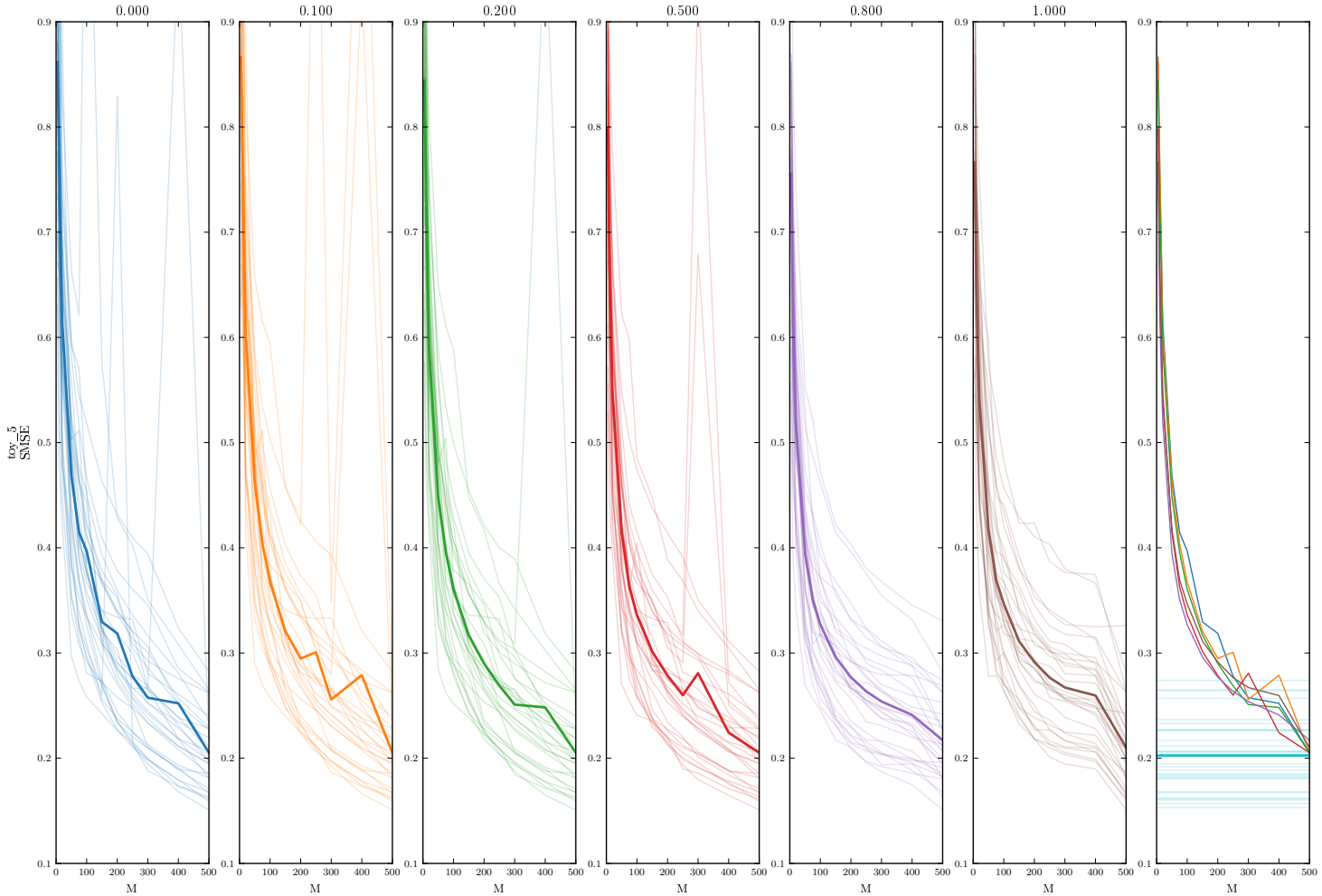

Figure 12: Results on a toy regression problem: Standardsised mean squared error on the test set for various values of α and various number of pseudo-points M. Each trace is for one split, bold line is the mean. The rightmost figure shows the mean for various α, and the results using GP regression.

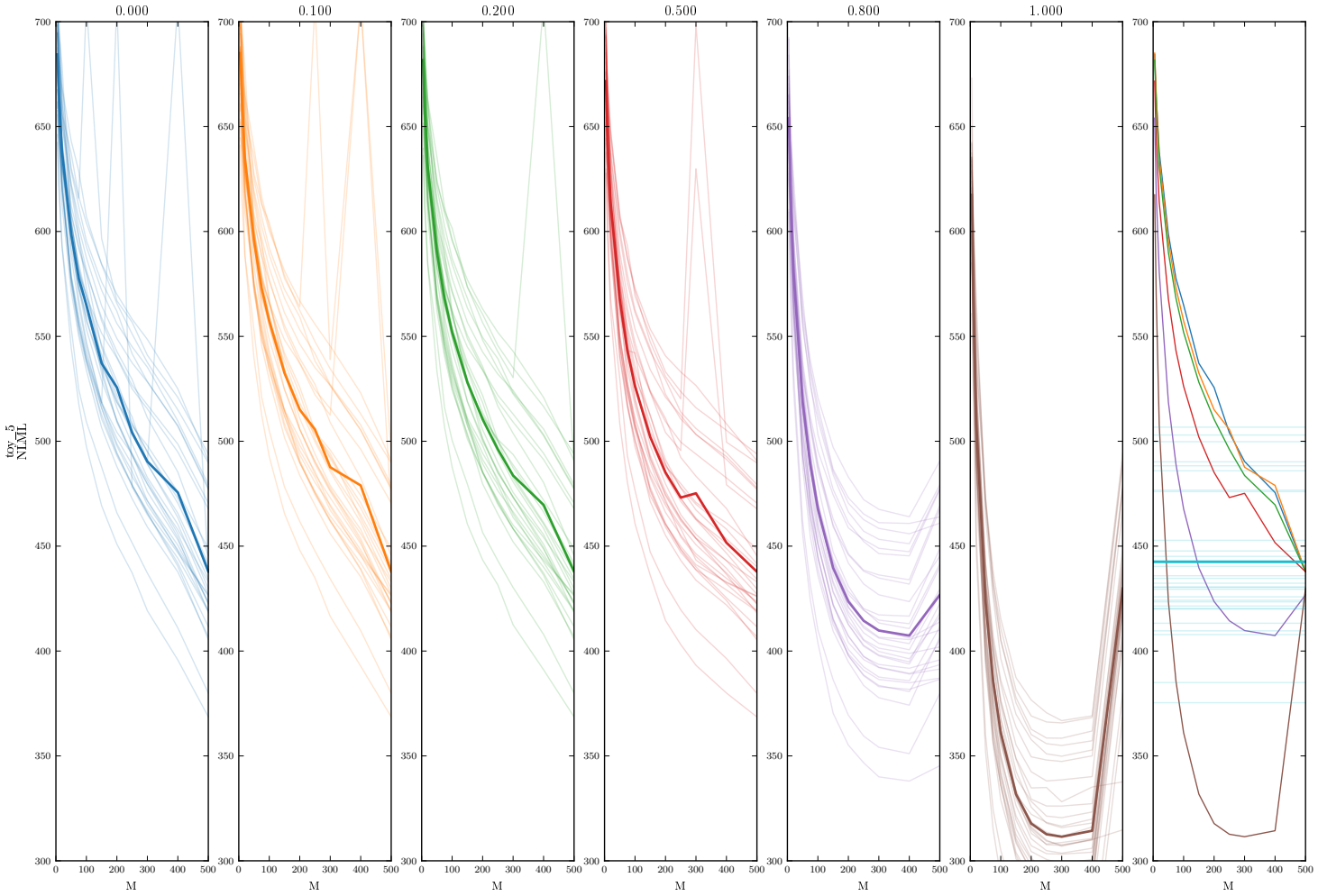

Figure 13: Results on a toy regression problem: The negative log marginal likelihood of the training set after training for various values of α and various number of pseudo-points M. Each trace is for one split, bold line is the mean. The rightmost figure shows the mean for various α, and the results using GP regression. Power EP with α close to 1 over-estimates the marginal-likelihood.

I.2 Real-world Regression

We include the details of the regression data sets in Table 1 and several comparisons of α values in Figures 17 to 22.

data set	N train/test	D
boston	455/51	14
concrete	927/103	9
energy	691/77	9
kin8nm	7373/819	9
naval	10741/1193	18
yacht	277/31	7
power	8611/957	5
red wine	1439/160	12

Table 1: Regression data sets

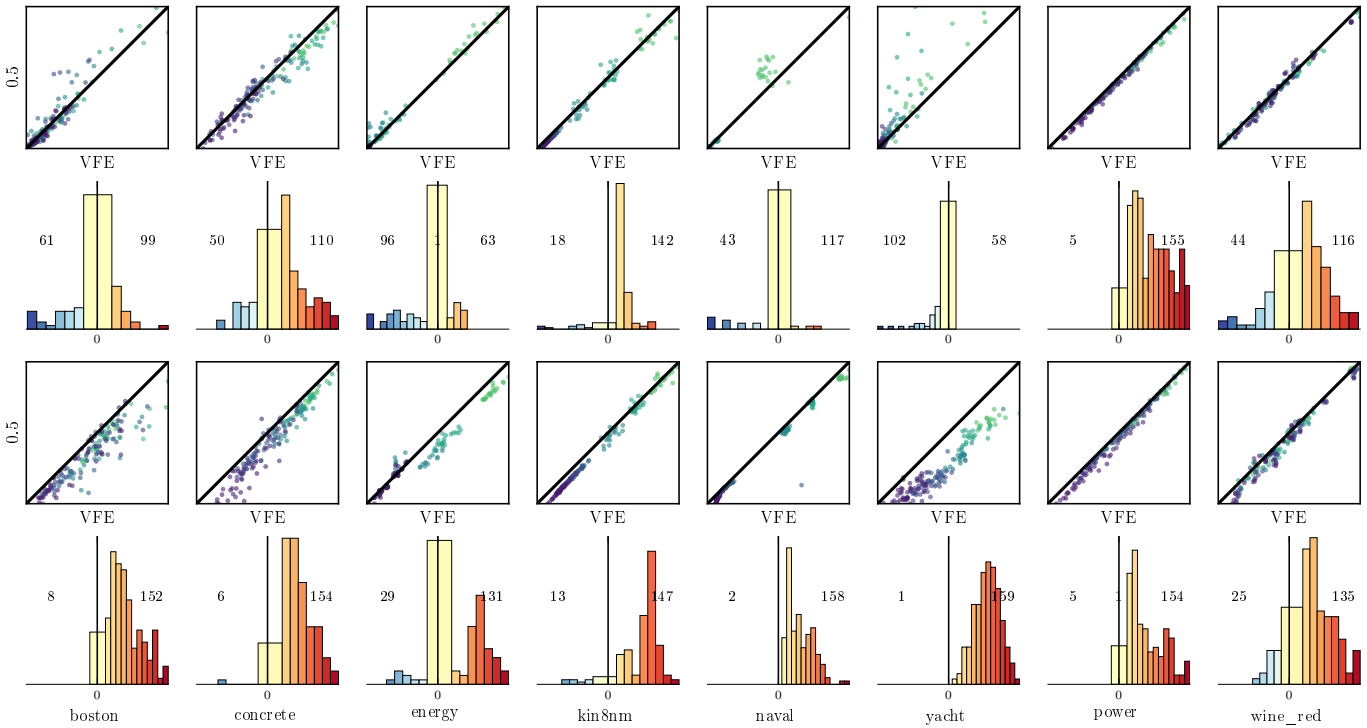

Figure 14: A comparison between Power-EP with $\alpha = 0.5$ and VFE on several regression data sets, on two metrics SMSE (top two rows) and SMLL (bottom two rows). The scatter plots show the performance of Power-EP ($\alpha = 0.5$) vs VFE. Each point is one split and points with lighter colours are runs with big M. Points that stay below the diagonal line show $\alpha = 0.5$ is better than VFE. The plots right underneath the scatter plots show the histogram of the difference between methods. Red means $\alpha = 0.5$ is better than VFE.

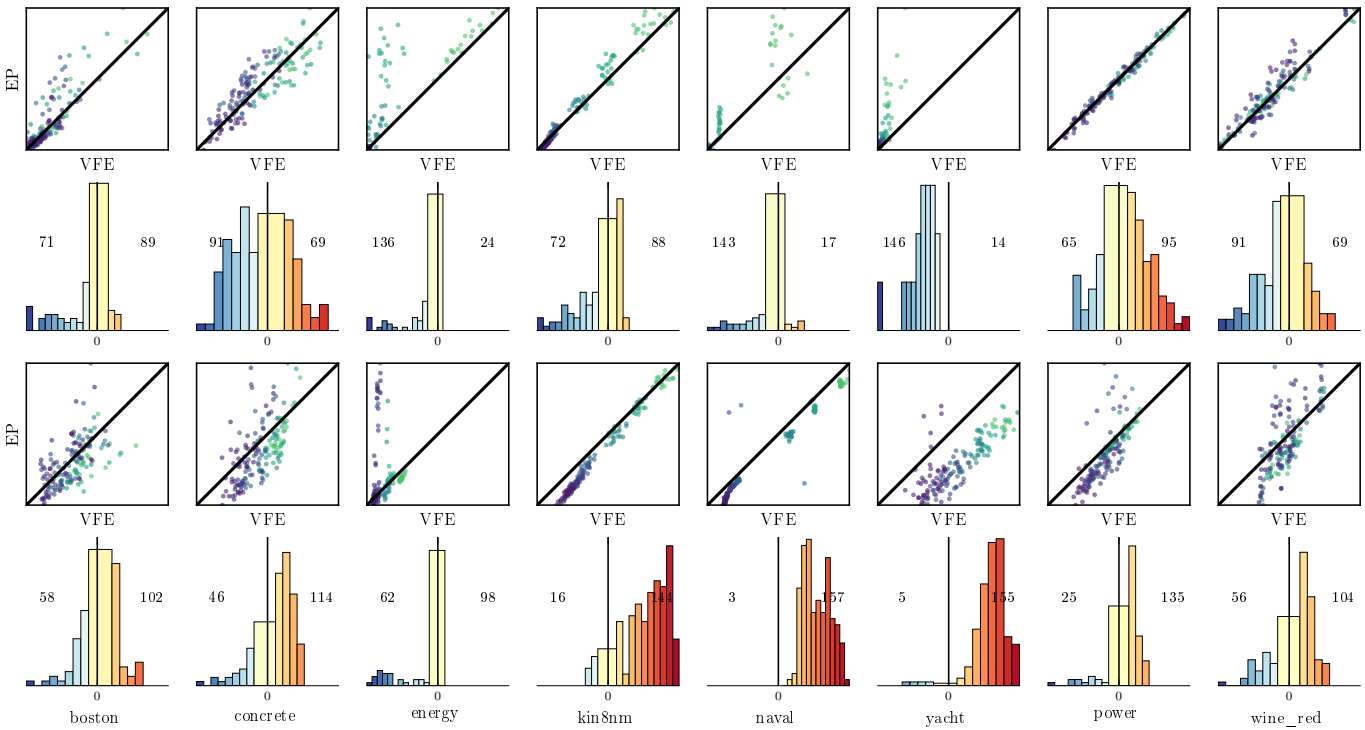

Figure 15: A comparison between EP and VFE on several regression data sets, on two metrics SMSE (top two rows) and SMLL (bottom two rows). See Figure 14 for more details about the plots.

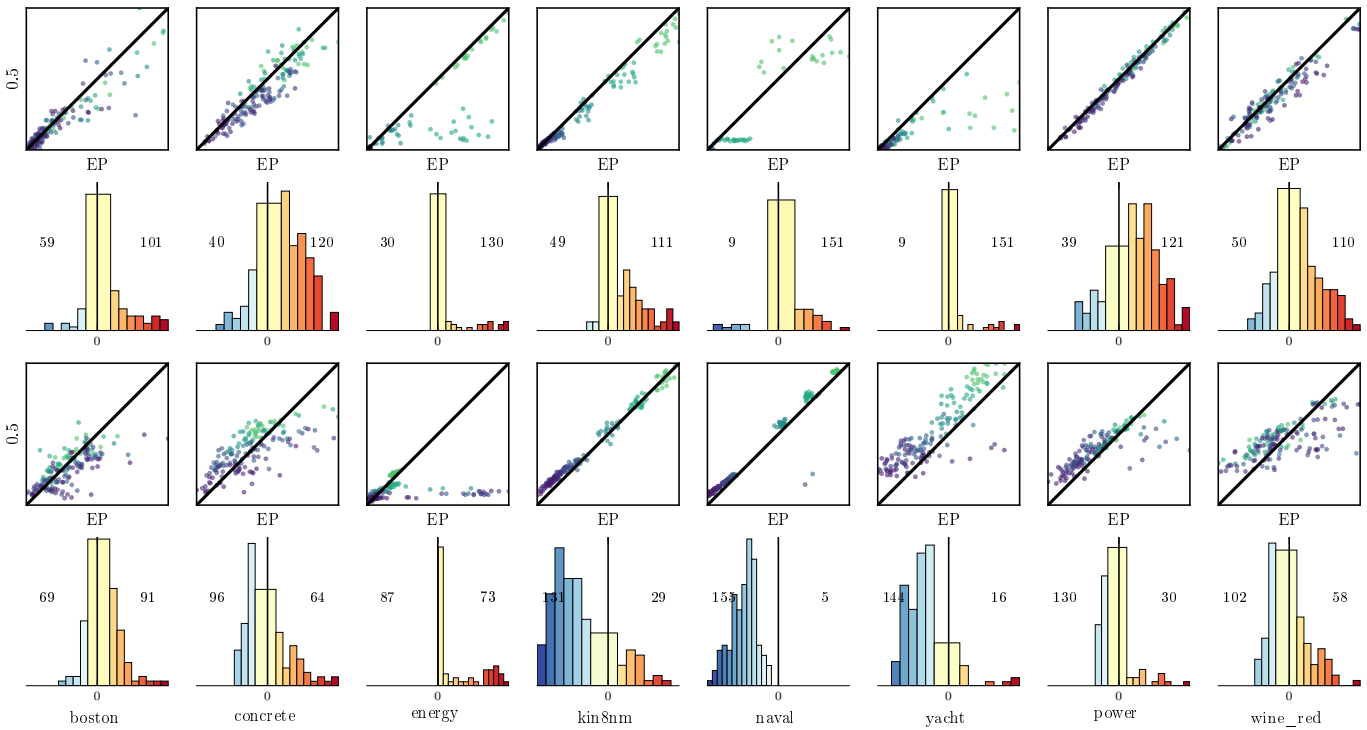

Figure 16: A comparison between Power-EP with $\alpha = 0.5$ and EP on several regression data sets, on two metrics SMSE (top two rows) and SMLL (bottom two rows). See Figure 14 for more details about the plots.

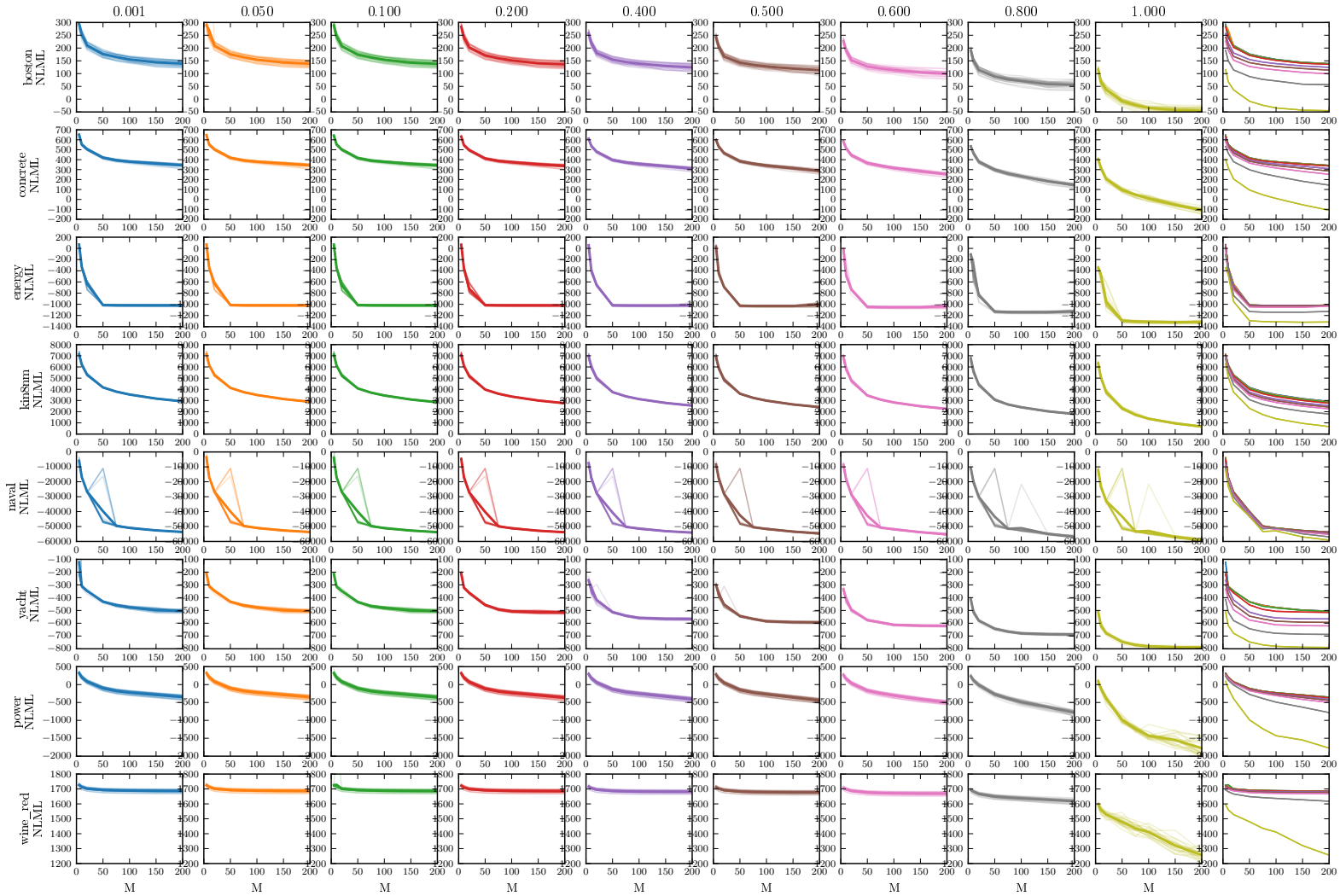

Figure 17: Results on real-world regression problems: Negative training log-marginal likelihood for different data sets, various values of α and various number of pseudo-points M. Each trace is for one split, bold line is the mean. The rightmost figures show the mean for various α for comparison. Lower is better [however, lower could mean overestimation].

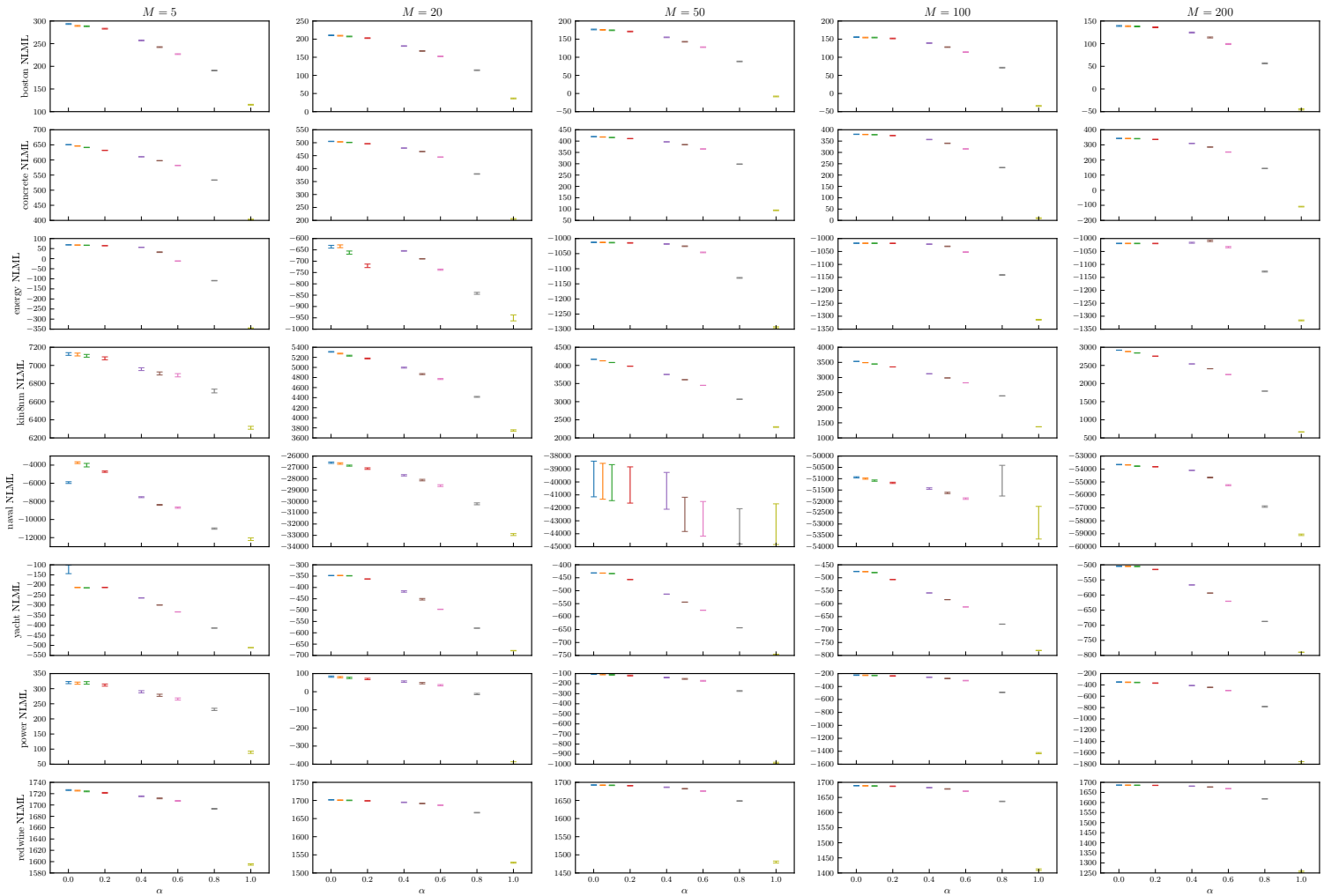

Figure 18: Results on real-world regression problems: Negative training log-marginal likelihood for different data sets, various values of α and various number of pseudo-points M, averaged over 20 splits. Lower is better [however, lower could mean overestimation].

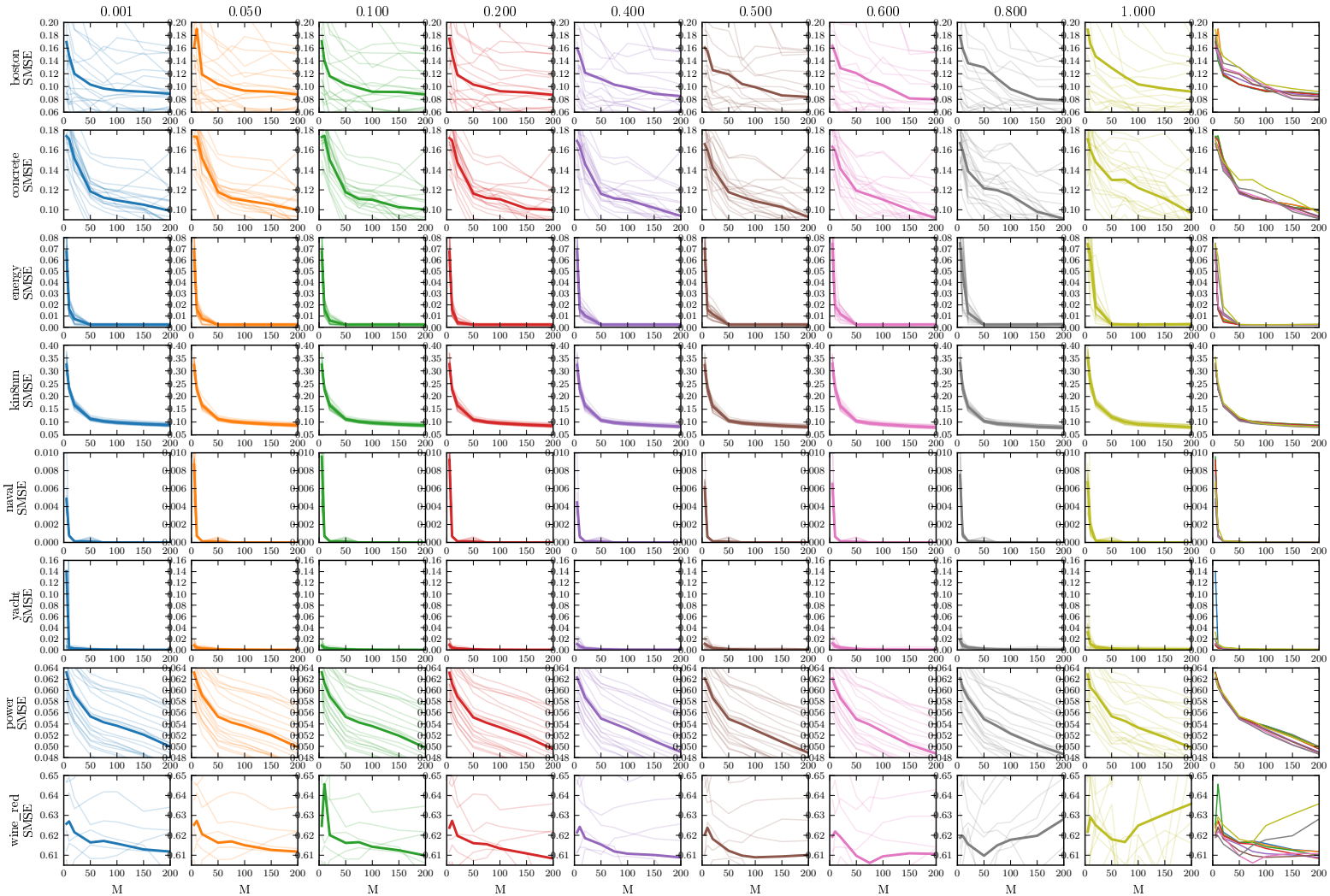

Figure 19: Results on real-world regression problems: Standardised mean squared error on the test set for different data sets, various values of α and various number of pseudo-points M. Each trace is for one split, bold line is the mean. The rightmost figures show the mean for various α for comparison. Lower is better.

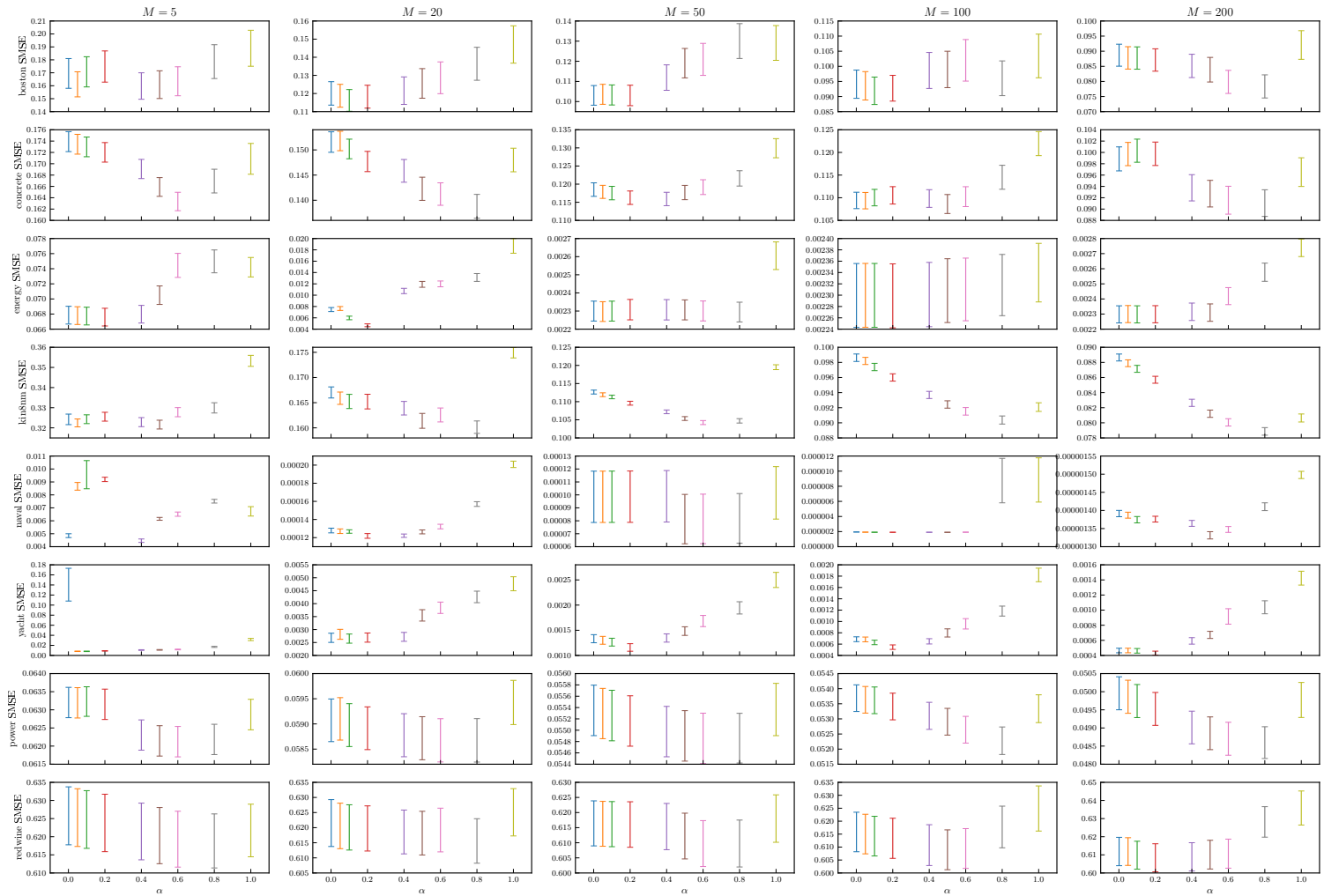

Figure 20: Results on real-world regression problems: Standardised mean squared error on the test set for different data sets, various values of α and various number of pseudo-points M, averaged over 20 splits. Lower is better.

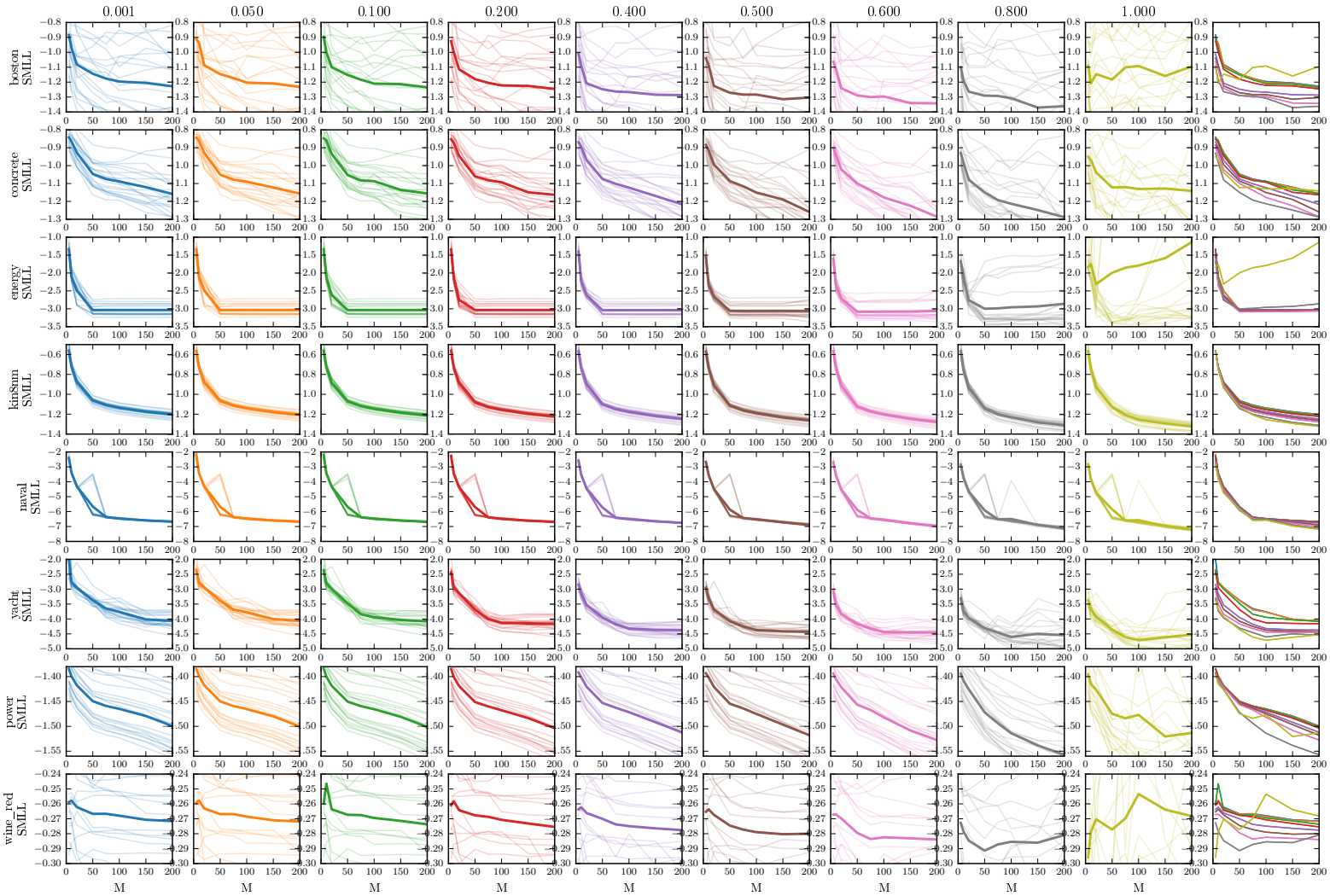

Figure 21: Results on real-world regression problems: Standardised mean log loss on the test set for different data sets, various values of α and various number of pseudo-points M. Each trace is for one split, bold line is the mean. The rightmost figures show the mean for various α for comparison. Lower is better.

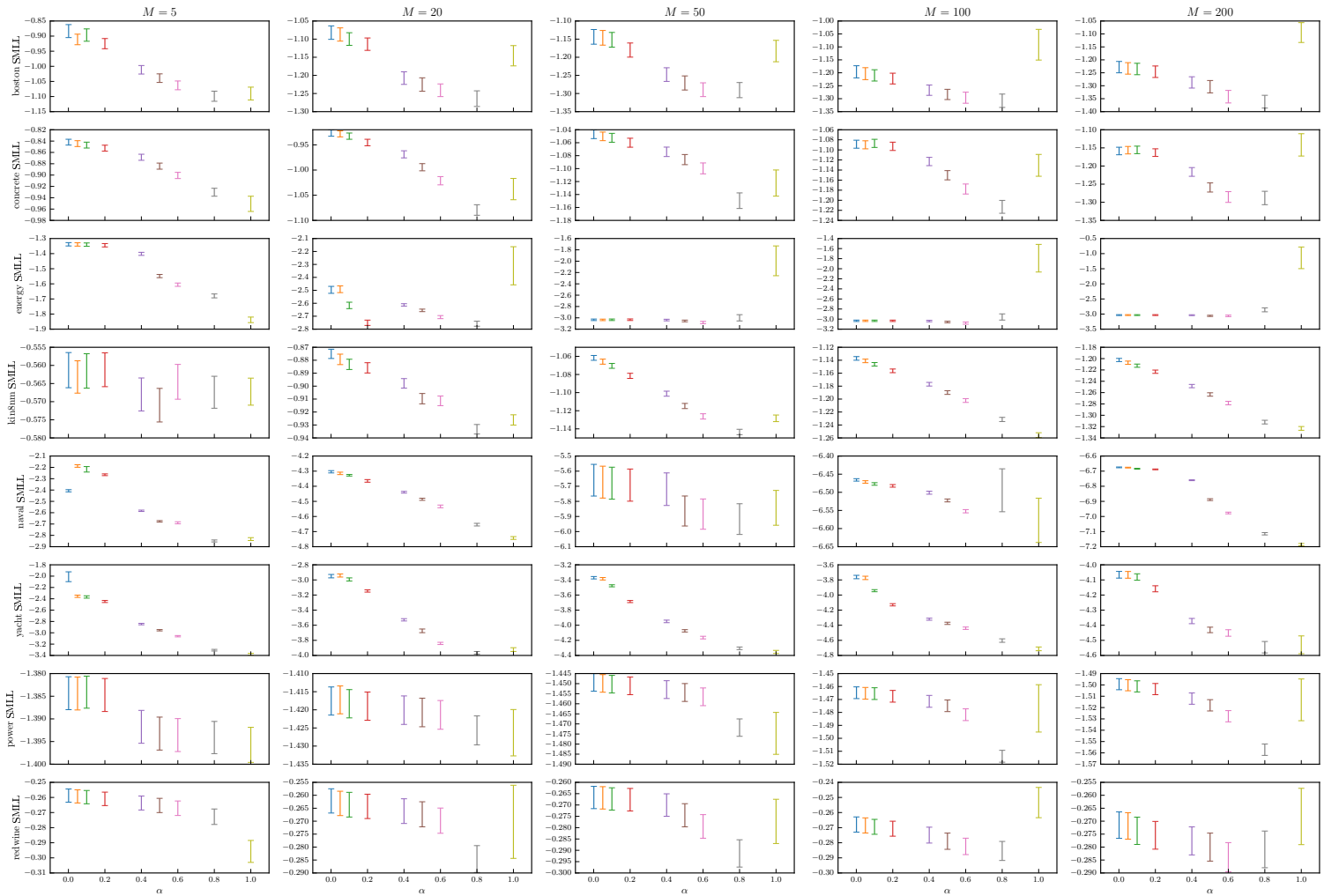

Figure 22: Results on real-world regression problems: Standardised mean log loss on the test set for different data sets, various values of α and various number of pseudo-points M, averaged over 20 splits. Lower is better.

I.3 Real-world Classification

It was demonstrated in (Hernández-Lobato and Hernández-Lobato, 2016; Hensman et al., 2015) that, once optimised, the pseudo points tend to concentrate around the decision boundary for VFE, and spread out to cover the data region in EP. Figure 23 illustrates the same effect as α goes from close to 0 (VFE) to 1 (EP).

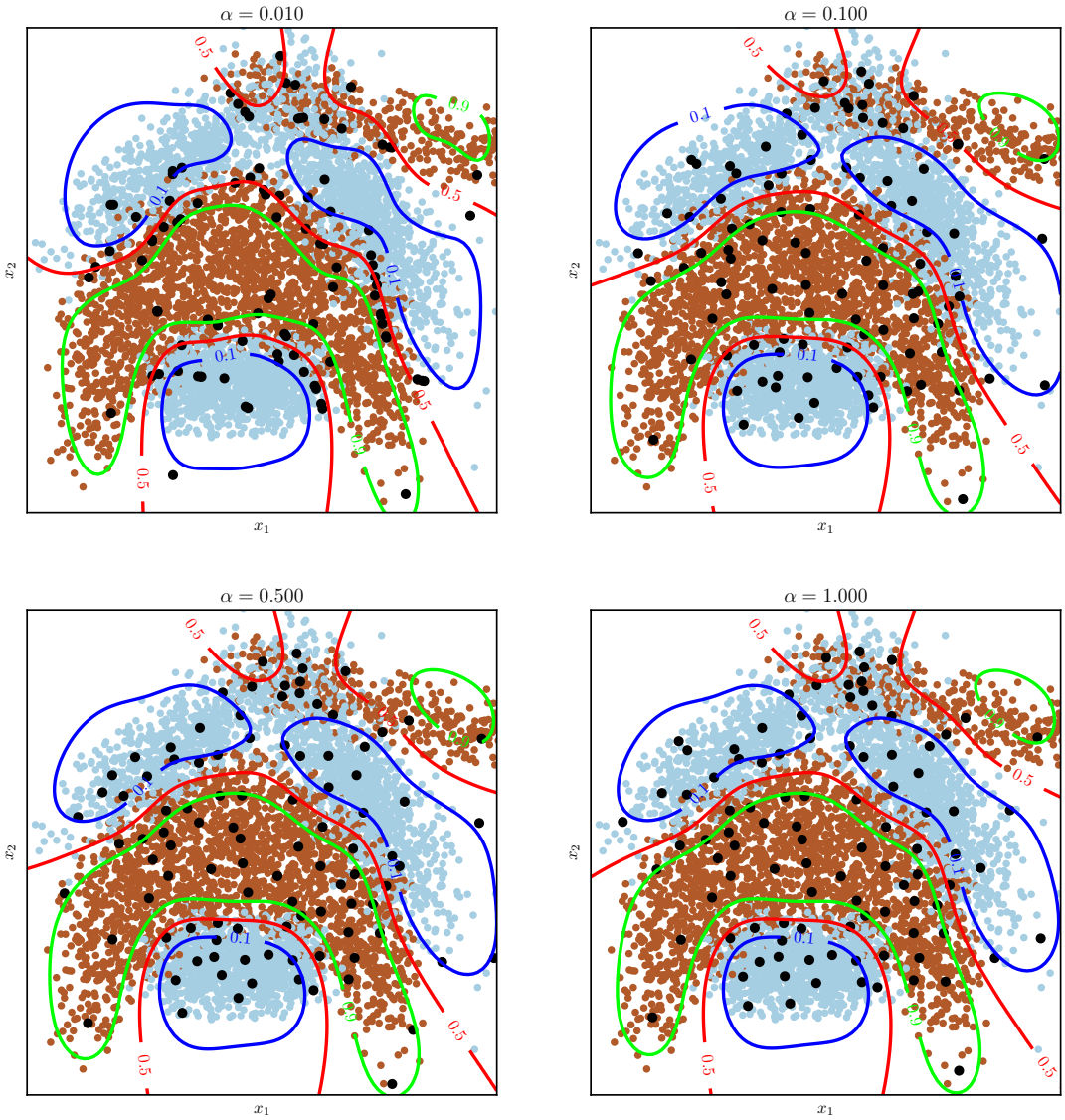

Figure 23: The locations of pseudo data points vary with α.. Best viewed in colour.

We include the details of the classification data sets in Table 2 and several comparisons of α values in Figures 27 to 30.

data set	N train/test	D	N positive/negative
australian	621/69	15	222/468
breast	614/68	11	239/443
crabs	180/20	7	100/100
iono	315/35	35	126/224
pima	690/77	9	500/267
sonar	186/21	61	111/96

Table 2: Classification data sets

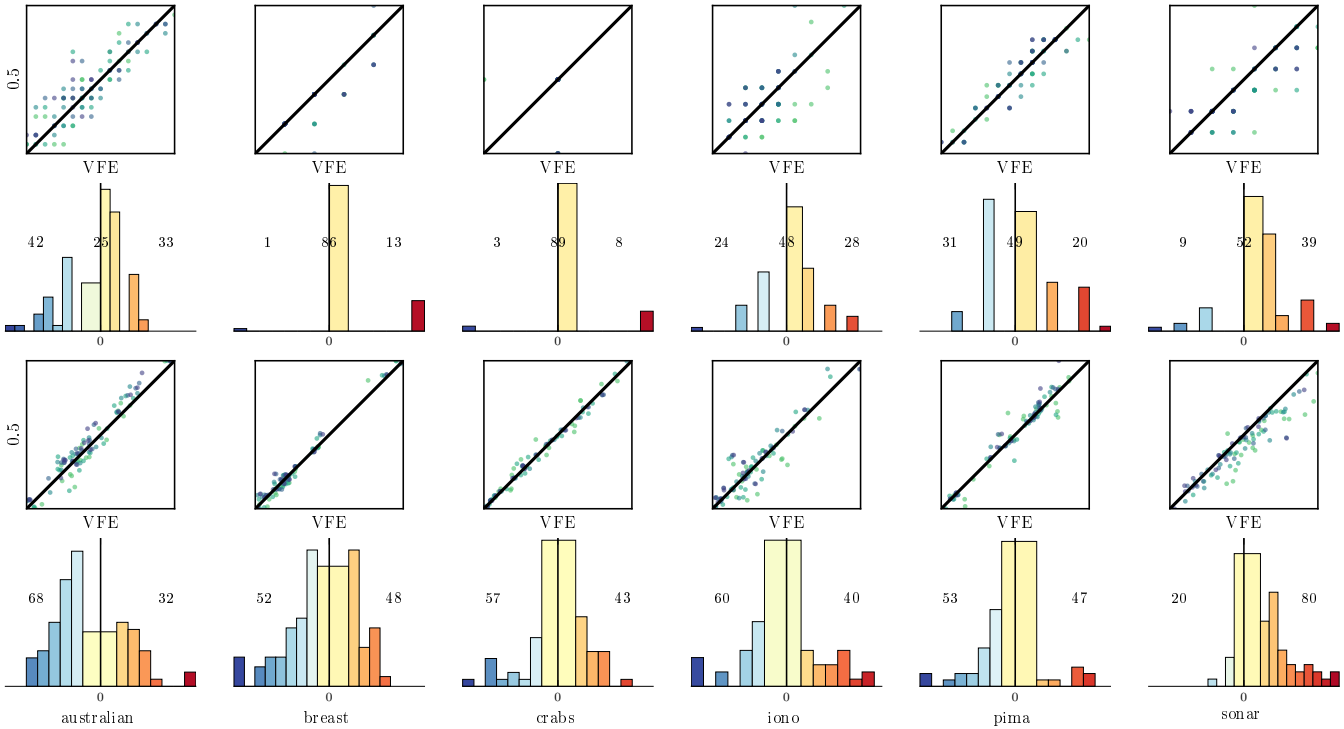

Figure 24: A comparison between Power-EP with $\alpha = 0.5$ and VFE on several classification data sets, on two metrics: classification error (top two rows) and NLL (bottom two rows). See Figure 14 for more details about the plots.

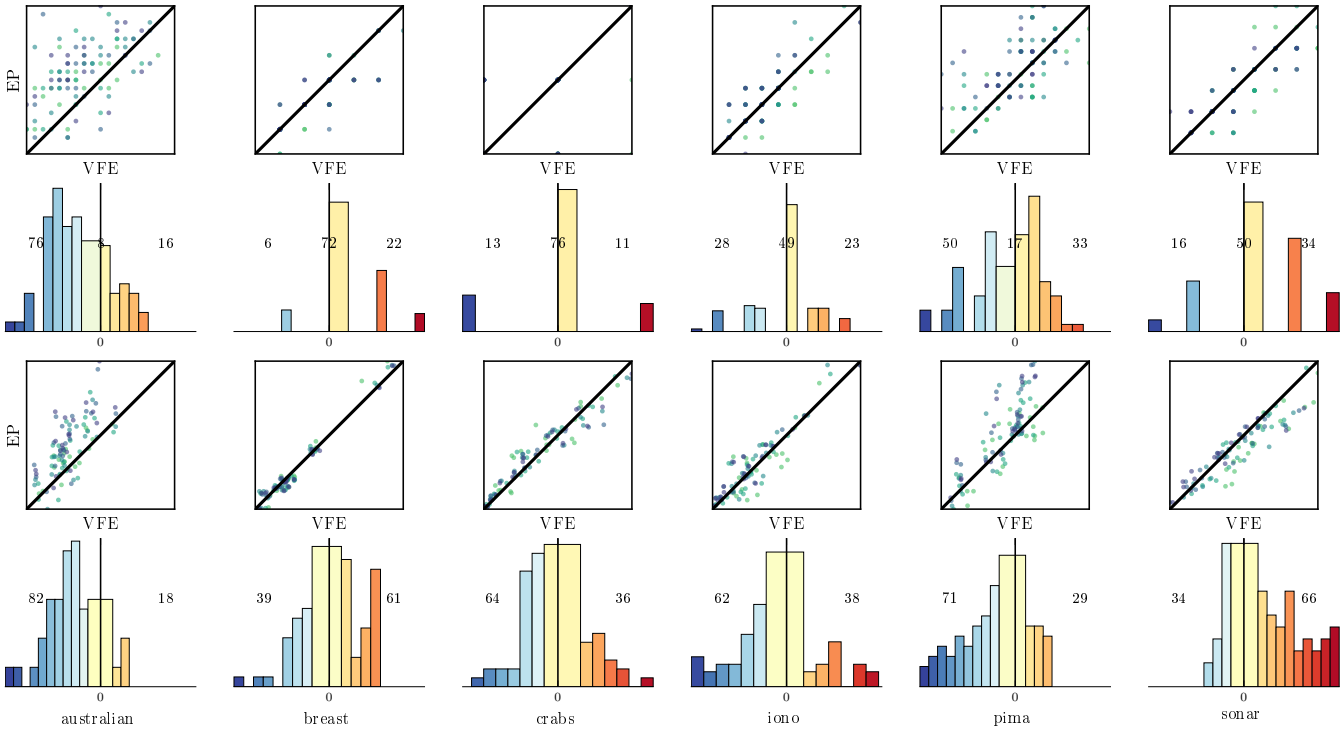

Figure 25: A comparison between EP and VFE on several classification data sets, on two metrics: classification error (top two rows) and NLL (bottom two rows). See Figure 14 for more details about the plots.

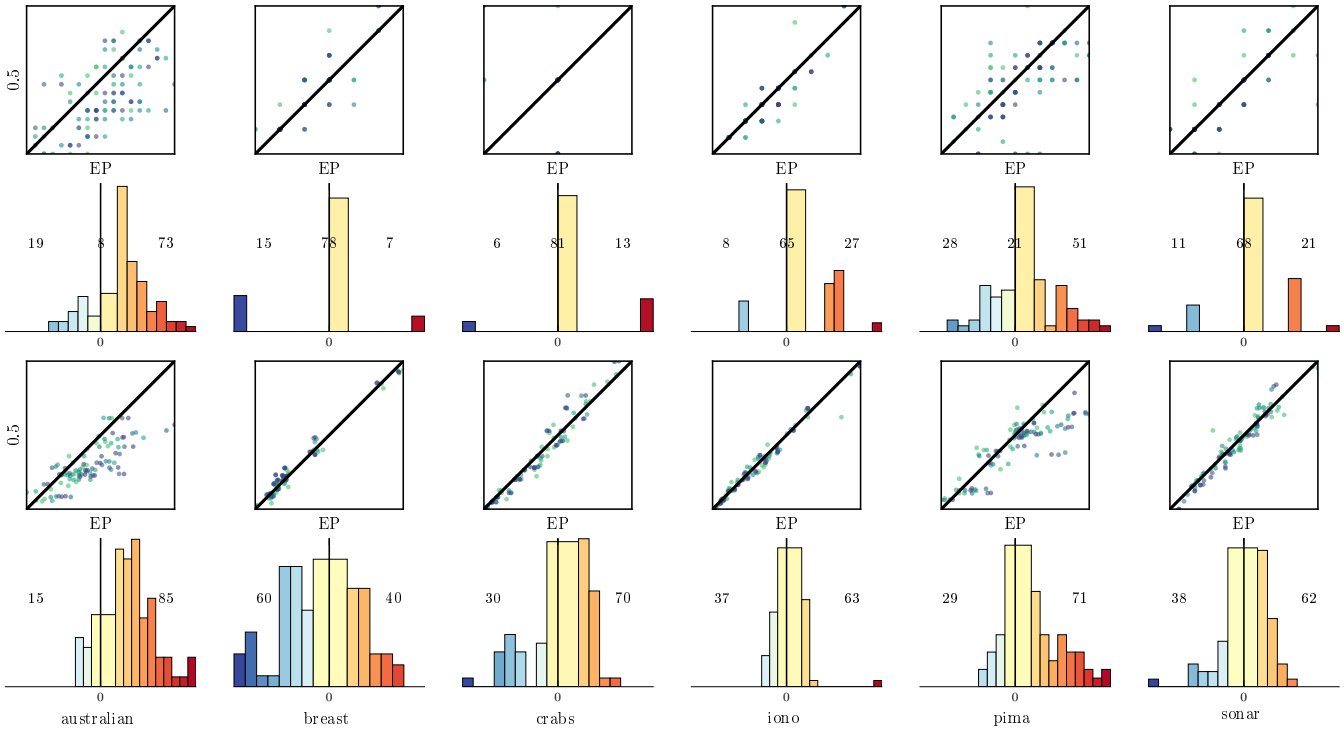

Figure 26: A comparison between Power-EP with $\alpha = 0.5$ and EP on several classification data sets, on two metrics: classification error (top two rows) and NLL (bottom two rows). See Figure 14 for more details about the plots.

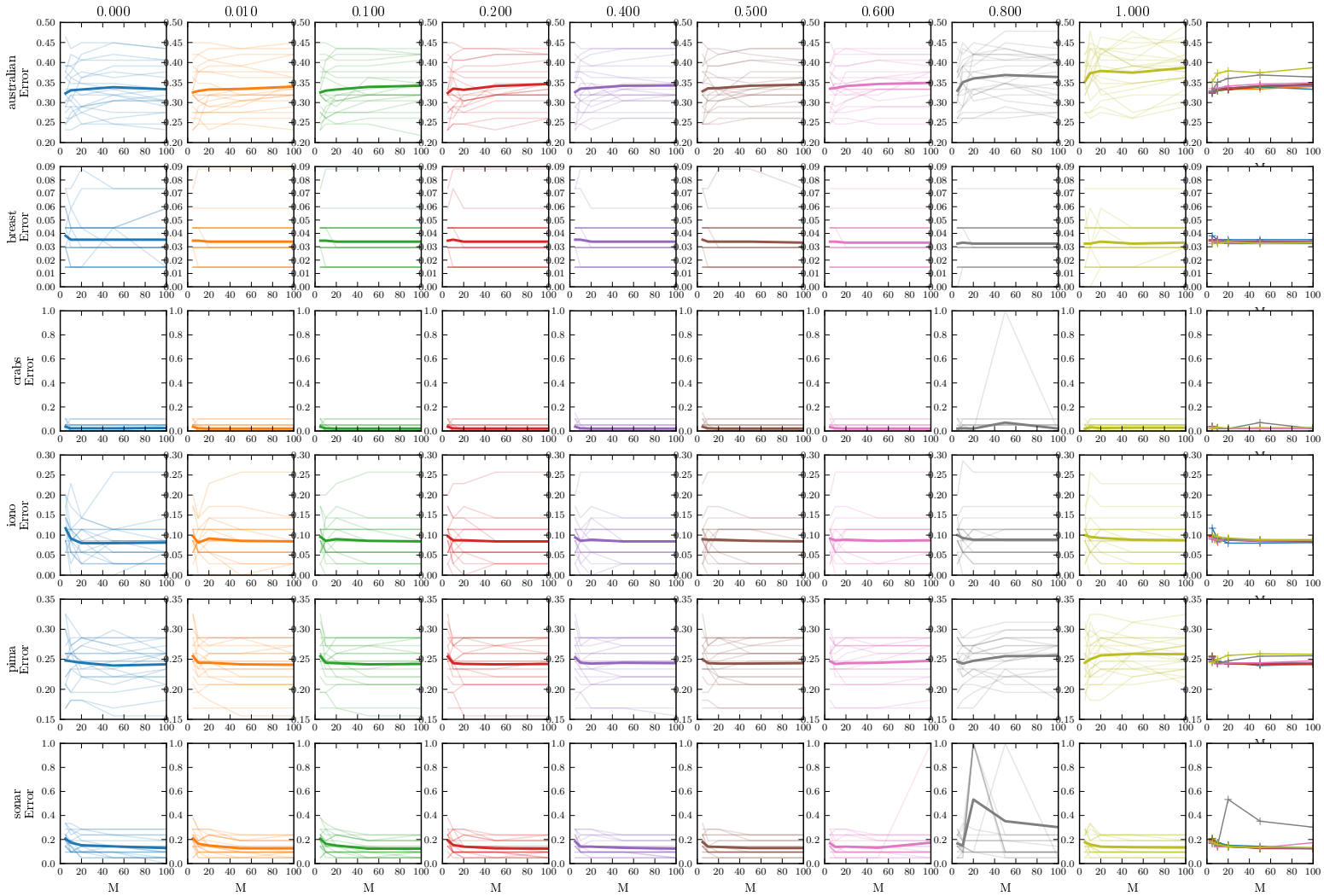

Figure 27: Results on real-world classification problems: Classification error rate on the test set for different data sets, various values of α and various number of pseudo-points M. Each trace is for one split, bold line is the mean. The rightmost figures show the mean for various α for comparison. Lower is better.

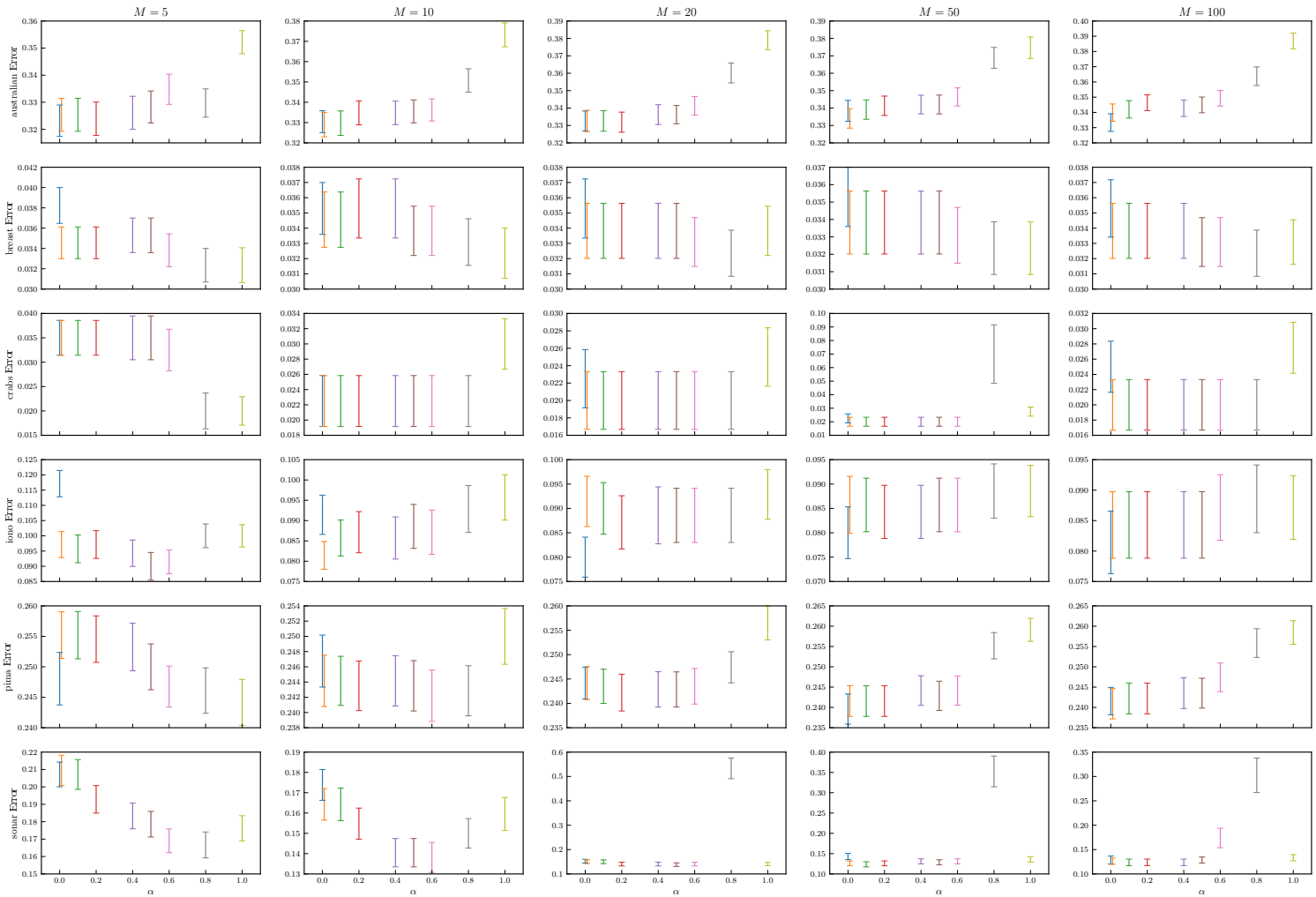

Figure 28: Results on real-world classification problems: Classification error rate on the test set for different data sets, various values of α and various number of pseudo-points M, averaged over 20 splits. Lower is better.

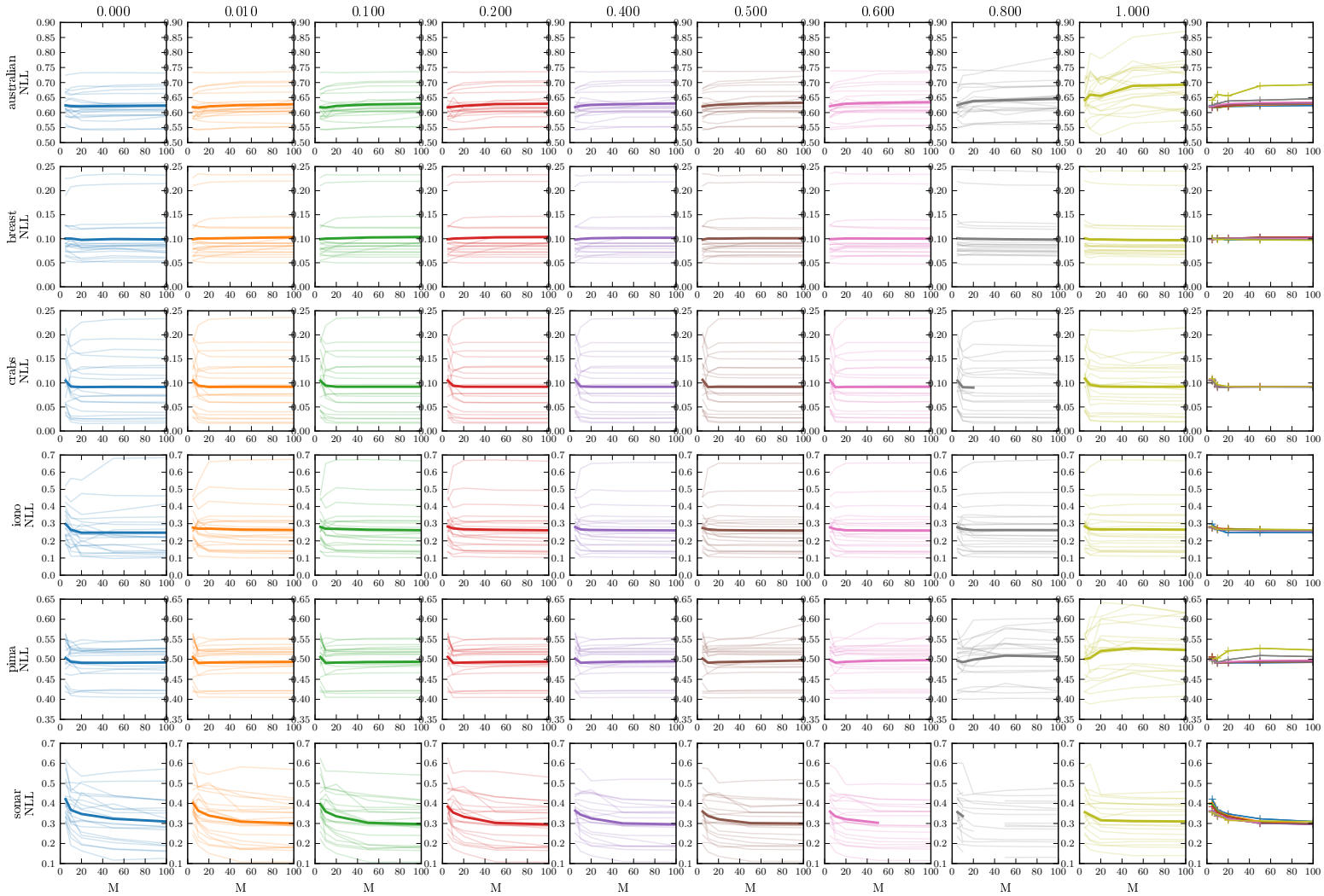

Figure 29: Results on real-world classification problems: Average negative log-likelihood on the test set for different data sets, various values of α and various number of pseudo-points M. Each trace is for one split, bold line is the mean. The rightmost figures show the mean for various α for comparison. Lower is better.

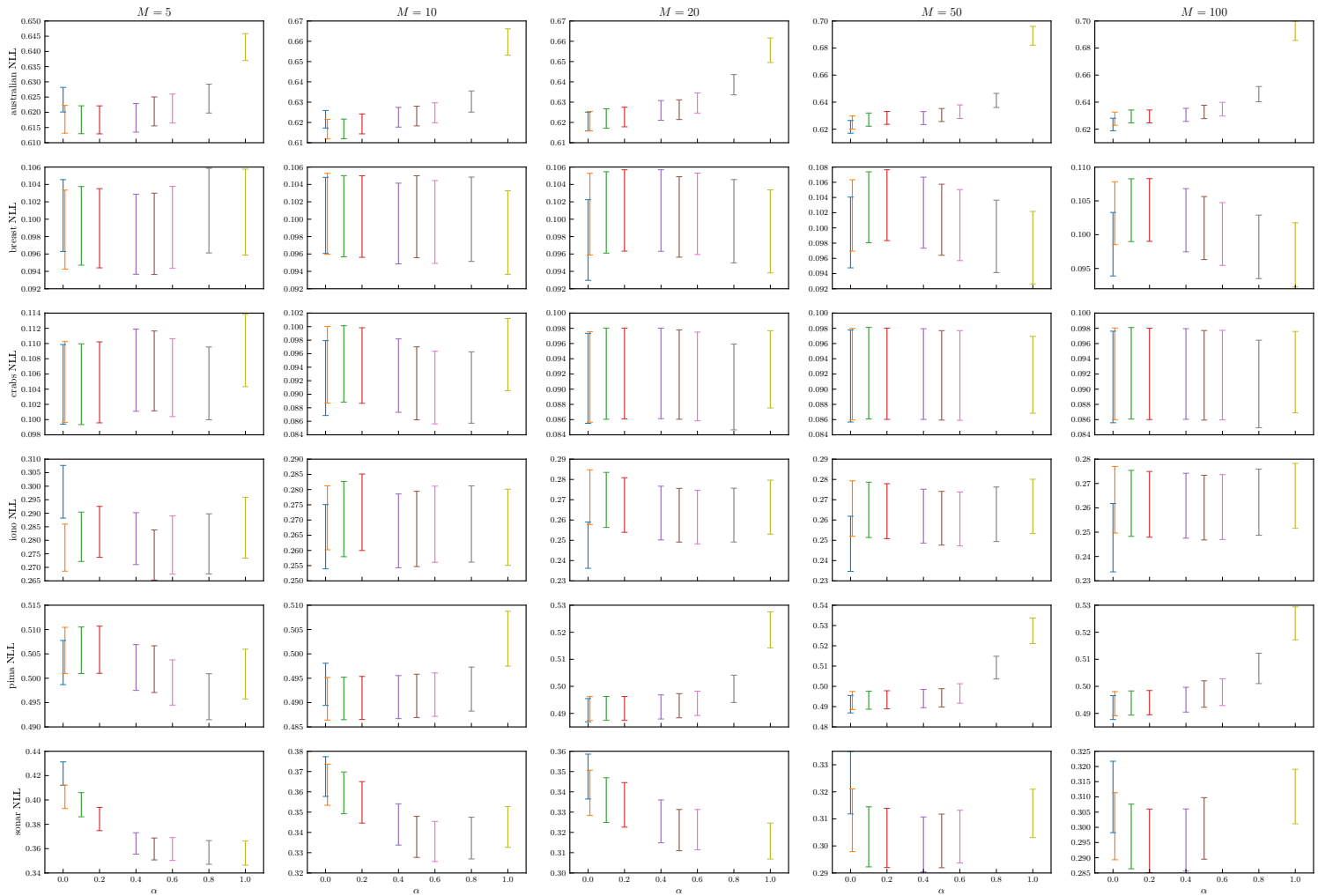

Figure 30: Results on real-world classification problems: Average negative log-likelihood on the test set for different data sets, various values of α and various number of pseudo-points M, averaged over 20 splits. Lower is better.

I.4 Binary Classification on Even/Odd MNIST Digits

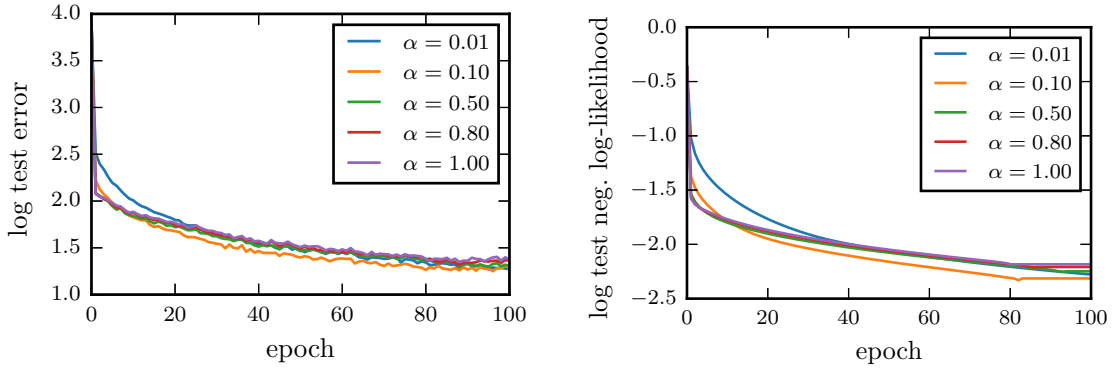

Figure 31: The test error and log-likelihood of the MNIST binary classification task (M=100).

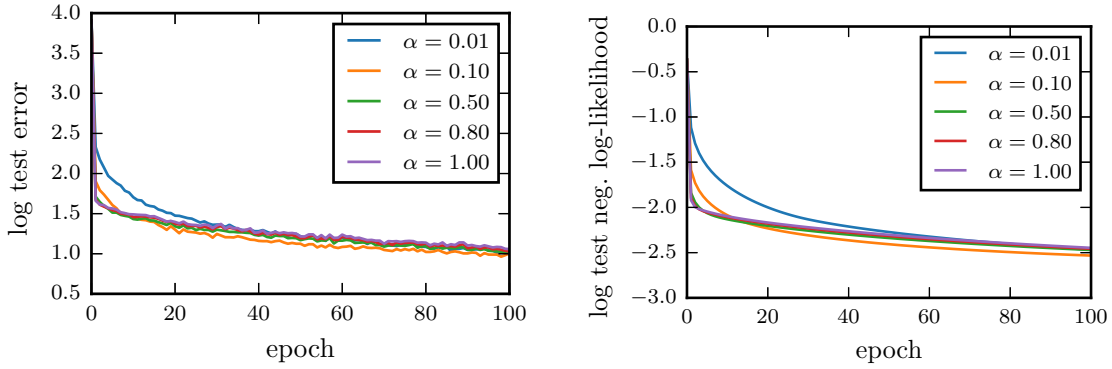

Figure 32: The test error and log-likelihood of the MNIST binary classification task (M=200).

I.5 When $M = N$ and $\alpha = 1$, Do We Recover EP for GPC (Rasmussen and Williams, 2005, sec. 3.6)?

The key difference between the EP method in this manuscript when $M = N$ and the pseudo-inputs and the training inputs are identical, and the standard EP method as described by (Rasmussen and Williams, 2005, sec. 3.6) is the factor representation. While Rasmussen and Williams (2005) used a one dimensional unnormalised Gaussian distribution that touches only *one* function value f_n to approximate each exact factor, the approximate factor used in the EP scheme described in the main text touches *all* M pseudo-points, hence *all* N function values when the pseudo-inputs are placed at the training inputs. However, in

practice both methods give virtually identical results. Figure 33 shows the approximate log marginal likelihood and the negative test log-likelihood, given by running the EP procedure described in the main text on the `ionosphere` data set. We note that these results are similar to that of the standard EP method (see Kuss and Rasmussen, 2005).

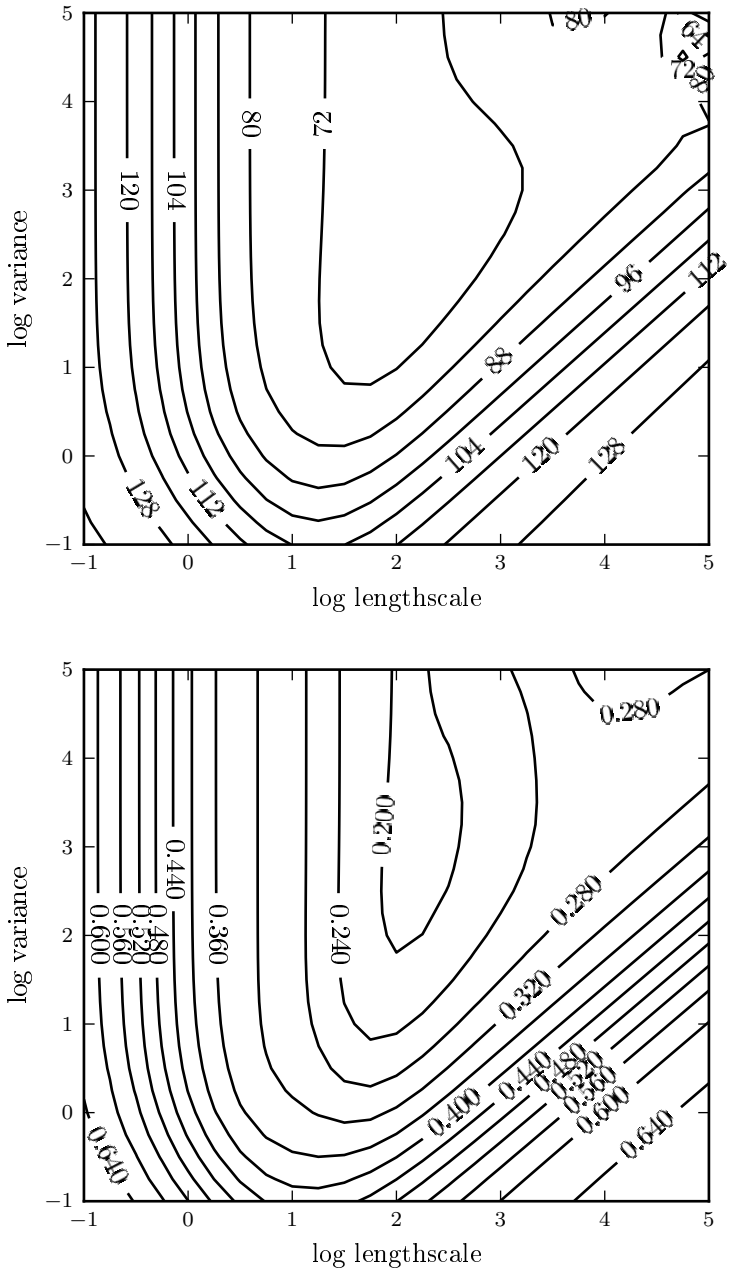

Figure 33: EP energy on the train set [TOP] and the average negative log-likelihood on the test set[BOTTOM] when $M = N$.

References

Mauricio A. Álvarez, David Luengo, Michalis K. Titsias, and Neil D. Lawrence. Efficient multioutput Gaussian processes through variational inducing kernels. In *13th International Conference on Artificial Intelligence and Statistics*, pages 25–32, 2010.

Matthias Bauer, Mark van der Wilk, and Carl E. Rasmussen. Understanding probabilistic sparse Gaussian process approximations. In *Advances in Neural Information Processing Systems 29*, pages 1525–1533, 2016.

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

Thang D. Bui and Richard E. Turner. Tree-structured Gaussian process approximations. In *Advances in Neural Information Processing Systems 27*, pages 2213–2221, 2014.

Thang D. Bui, Cuong V. Nguyen, and Richard E. Turner. Streaming sparse Gaussian process approximations. In *Advances in Neural Information Processing Systems 30*, 2017.

Lehel Csató. *Gaussian Processes — Iterative Sparse Approximations*. PhD thesis, Aston University, 2002.

Lehel Csató and Manfred Opper. Sparse online Gaussian processes. *Neural Computation*, 14(3):641–669, 2002.

Michael R.W. Dawson. *Understanding cognitive science*. Blackwell Publishing, 1998.

Marc P. Deisenroth. *Efficient Reinforcement Learning using Gaussian Processes*. PhD thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2010.

Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Learning and policy search in stochastic dynamical systems with Bayesian neural networks. In *4th International Conference on Learning Representations*, 2016.

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

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

Roger Frigola, Yutian Chen, and Carl E. Rasmussen. Variational Gaussian process state-space models. In *Advances in Neural Information Processing Systems 27*, pages 3680–3688, 2014.

Andrew Gelman, Aki Vehtari, Pasi Jylänki, Christian Robert, Nicolas Chopin, and John P Cunningham. Expectation propagation as a way of life. *arXiv preprint arXiv:1412.4869*, 2014.

James Hensman, Nicolo Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *29th Conference on Uncertainty in Artificial Intellegence*, pages 282–290, 2013.

James Hensman, Alexander G. D. G. Matthews, and Zoubin Ghahramani. Scalable variational Gaussian process classification. In *18th International Conference on Artificial Intelligence and Statistics*, pages 351–360, 2015.

Daniel Hernández-Lobato and José Miguel Hernández-Lobato. Scalable Gaussian process classification via expectation propagation. In *19th International Conference on Artificial Intelligence and Statistics*, pages 168–176, 2016.

José Miguel Hernández-Lobato, Yingzhen Li, Mark Rowland, Daniel Hernández-Lobato, Thang D Bui, and Richard E Turner. Black-box α-divergence minimization. In *33rd International Conference on International Conference on Machine Learning*, pages 1511–1520, 2016.

Trong Nghia Hoang, Quang Minh Hoang, and Bryan Kian Hsiang Low. A distributed variational inference framework for unifying parallel sparse Gaussian process regression models. In *33rd International Conference on Machine Learning*, pages 382–391, 2016.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

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

Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

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

Malte Kuss and Carl E. Rasmussen. Assessing approximate inference for binary Gaussian process classification. *The Journal of Machine Learning Research*, 6:1679–1704, 2005.

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

Yingzhen Li, José Miguel Hernández-Lobato, and Richard E. Turner. Stochastic expectation propagation. In *Advances in Neural Information Processing Systems 29*, pages 2323–2331, 2015.

Kian Hsiang Low, Jiangbo Yu, Jie Chen, and Patrick Jaillet. Parallel Gaussian process regression for big data: Low-rank representation meets Markov approximation. In *29th AAAI Conference on Artificial Intelligence*, pages 2821–2827, 2015.

Maren Mahsereci and Philipp Hennig. Probabilistic line searches for stochastic optimization. In *Advances in Neural Information Processing Systems 28*, pages 181–189, 2015.

Alexander G. D. G. Matthews, James Hensman, Richard E. Turner, and Zoubin Ghahramani. On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. In *19th International Conference on Artificial Intelligence and Statistics*, pages 231–239, 2016.

Andrew McHutchon. *Nonlinear modelling and control using Gaussian processes*. PhD thesis, University of Cambridge, 2014.

Thomas P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

Thomas P. Minka. Power EP. Technical report, Microsoft Research Cambridge, 2004.

Thomas P. Minka. Divergence measures and message passing. Technical report, Microsoft Research Cambridge, 2005.

Andrew Naish-Guzman and Sean B. Holden. The generalized FITC approximation. In *Advances in Neural Information Processing Systems 20*, pages 1057–1064, 2007.

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

Juho Piironen and Aki Vehtari. Comparison of Bayesian predictive methods for model selection. *Statistics and Computing*, 27(3):711–735, 2017.

Yuan Qi, Ahmed H. Abdel-Gawad, and Thomas P. Minka. Sparse-posterior Gaussian processes for general likelihoods. In *26th Conference on Uncertainty in Artificial Intelligence*, pages 450–457, 2010.

Joaquin Quiñonero-Candela and Carl E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

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

Alan D. Saul, James Hensman, Aki Vehtari, and Neil D. Lawrence. Chained Gaussian processes. In *19th International Conference on Artificial Intelligence and Statistics*, pages 1431–1440, 2016.

Anton Schwaighofer and Volker Tresp. Transductive and inductive methods for approximate Gaussian process regression. In *Advances in Neural Information Processing Systems 15*, pages 953–960, 2002.

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

Matthias Seeger and Michael I. Jordan. Sparse Gaussian process classification with multiple classes. Technical report, Department of Statistics, University of Berkeley, CA, 2004.

Matthias Seeger, Christopher K. I. Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *9th International Conference on Artificial Intelligence and Statistics*, 2003.

Edward Snelson. *Flexible and efficient Gaussian process models for machine learning*. PhD thesis, University College London, 2007.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 19*, pages 1257–1264, 2006.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25*, pages 2951–2959, 2012.

Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *12th International Conference on Artificial Intelligence and Statistics*, pages 567–574, 2009.

Michalis K. Titsias and Neil D. Lawrence. Bayesian Gaussian process latent variable model. In *13th International Conference on Artificial Intelligence and Statistics*, pages 844–851, 2010.

Felipe Tobar, Thang D. Bui, and Richard E. Turner. Learning stationary time series using Gaussian processes with nonparametric kernels. In *Advances in Neural Information Processing Systems 29*, pages 3501–3509, 2015.

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

Jack M. Wang, David J. Fleet, and Aaron Hertzmann. Gaussian process dynamical models. In *Advances in Neural Information Processing Systems 18*, pages 1441–1448, 2005.

Minjie Xu, Balaji Lakshminarayanan, Yee Whye Teh, Jun Zhu, and Bo Zhang. Distributed Bayesian posterior sampling via moment sharing. In *Advances in Neural Information Processing Systems 27*, pages 3356–3364, 2014.

Huaiyu Zhu and Richard Rohwer. Information geometric measurements of generalisation. Technical report, Aston University, 1995.

Huaiyu Zhu and Richard Rohwer. Measurements of generalisation based on information geometry. In *Mathematics of Neural Networks*, pages 394–398. 1997.