

Understanding Probabilistic Sparse Gaussian Process Approximations

Matthias Stephan Bauer^{†‡} Mark van der Wilk[†] Carl Edward Rasmussen[†]

[†]Department of Engineering, University of Cambridge, Cambridge, UK

[‡]Max Planck Institute for Intelligent Systems, Tübingen, Germany

{msb55, mv310, cer54}@cam.ac.uk

Good sparse approximations are essential for practical inference in Gaussian Processes as the computational cost of exact methods is prohibitive for large datasets. The Fully Independent Training Conditional (FITC) and the Variational Free Energy (VFE) approximations are two recent popular methods. Despite superficial similarities, these approximations have surprisingly different theoretical properties and behave differently in practice. We thoroughly investigate the two methods for regression both analytically and through illustrative examples, and draw conclusions to guide practical application.

1. Introduction

Gaussian Processes (GPs) [1] are a flexible class of probabilistic models. Perhaps the most prominent practical limitation of GPs is that the computational requirement of an exact implementation scales as $O(N^3)$ time, and as $O(N^2)$ memory, where N is the number of training cases. Fortunately, recent progress has been made in developing *sparse approximations*, which retain the favourable properties of GPs but at a lower computational cost, typically $O(NM^2)$ time and $O(NM)$ memory for some chosen $M < N$. All sparse approximations rely on focussing inference on a small number of quantities which represent approximately the entire posterior over functions. These quantities can be chosen differently, e.g., it can be simply the value of the function at certain locations, properties of the spectral representations [2], more abstract representations [3]. Similar ideas are used in random feature expansions [4, 5].

Here we focus on methods which represent the approximate posterior using the function value at a set of M *inducing inputs* (also sometimes known as pseudo-inputs). These methods include the Deterministic Training Conditional (DTC) [6] and FITC [7], see [8] for a review, as well as the **Variational Free Energy (VFE) approximation** [9]. The methods differ both in terms of the theoretical approach in deriving the approximation, and also in terms of how the inducing inputs are handled. Broadly speaking, inducing inputs can either be chosen (e.g. at random) from the training set or be optimised over. In this paper we consider the latter, as this will generally allow for the best trade-off between accuracy and computational requirements. **Training the GP entails jointly optimizing over inducing inputs and hyperparameters.**

In this work, we aim to thoroughly investigate and characterise the difference in behaviour of the **FITC and VFE approximations**. We investigate the biases of the bounds when learning hyperparameters, where each method allocates its modelling capacity, and the optimisation behaviour. In Section 2 we briefly introduce inducing point methods and state the two algorithms using a unifying notation. In Section 3 we discuss properties of the two approaches, both theoretical and practical. Our aim is to understand the approximations in detail in order to know under which conditions each method is likely to succeed or fail in practice. We highlight issues which may arise in practical situations and how to diagnose and possibly avoid them. Some of the properties of the methods have been previously reported in the literature, our aim here is a more complete and comparative approach. We draw conclusions in Section 4.

2. Sparse Gaussian Processes

A Gaussian Process is a flexible distribution over functions, with many useful analytical properties. It is fully determined by its mean $m(\mathbf{x})$ and covariance $k(\mathbf{x}, \mathbf{x}')$ functions. We assume the mean to be zero, without loss of generality. The covariance function determines properties of the functions, like smoothness, amplitude, etc. A finite collection of function values at inputs $\{\mathbf{x}_i\}$ will have a Gaussian distribution $\mathcal{N}(\mathbf{f}; 0, K_{\mathbf{ff}})$, where $[K_{\mathbf{ff}}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

Here we revisit the Gaussian process model for regression [1]. We model the function of interest $f(\cdot)$ using a GP prior, and noisy observations at the input locations $X = \{\mathbf{x}_i\}$ are observed in the vector \mathbf{y} .

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{f}; 0, K_{\mathbf{ff}}) \quad p(\mathbf{y}|\mathbf{f}) = \prod_{n=1}^N \mathcal{N}(y_n; f_n, \sigma_n^2) \quad (1)$$

Throughout we will employ a squared exponential covariance function $k(x, x') = s_f^2 \exp(-\frac{1}{2}|x - x'|^2/\ell^2)$. The hyperparameter θ contains the signal variance s_f^2 , the lengthscale ℓ and the noise variance σ_n^2 , and is suppressed in the notation.

To make predictions, **we follow the common approach of first determining θ by optimising the marginal likelihood and then marginalising over the posterior of \mathbf{f} :**

$$\theta^* = \underset{\theta}{\operatorname{argmax}} p(\mathbf{y}|\theta) \quad p(y^*|\mathbf{y}) = \frac{p(y^*, \mathbf{y})}{p(\mathbf{y})} = \int p(y^*|f^*)p(f^*|\mathbf{f})p(\mathbf{f}|\mathbf{y})d\mathbf{f}df^* \quad (2)$$

While the marginal likelihood, the posterior and the predictive distribution all have closed-form Gaussian expressions, the cost of evaluating them scales as $\mathcal{O}(N^3)$ due **to the inversion of $K_{\mathbf{ff}} + \sigma_n^2 I$, which is impractical for many datasets**.

Over the years, the two inducing point methods **that have remained most influential** are FITC [7] and VFE [9]. Unlike previously proposed methods (see [6, 10, 8]), both FITC and VFE provide an approximation to the marginal likelihood which allows both the **hyperparameters and inducing inputs** to be **learned from the data through gradient based optimisation**. Both methods rely on the low rank matrix $Q_{\mathbf{ff}} = K_{\mathbf{fu}}K_{\mathbf{uu}}^{-1}K_{\mathbf{uf}}$ instead of the full rank $K_{\mathbf{ff}}$ to reduce the size of any matrix inversion to M . Note that for most covariance functions, **the eigenvalues of $K_{\mathbf{uu}}$ are not bounded away from zero**. Any practical implementation will have to address this to avoid numerical instability. We follow the common practice of adding **a tiny diagonal jitter term εI to $K_{\mathbf{uu}}$** before inverting.

2.1. Fully Independent Training Conditional (FITC)

Over the years, FITC has been formulated in several different ways. A form of FITC first appeared in an online learning setting by Csató and Oppé [11], derived from the viewpoint of approximating the full GP posterior. Snelson and Ghahramani [7] introduced FITC as approximate inference in a model with a modified likelihood and proposed using its marginal likelihood to train the hyperparameters and inducing inputs jointly. An alternate interpretation where the prior is modified, but exact inference is performed, was presented in [8], unifying it with other techniques. The latest interesting development came with the connection that FITC can be obtained by approximating the GP posterior using Expectation Propagation (EP) [12, 13].

Using the interpretation of modifying the prior to

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{f}; 0, Q_{\mathbf{ff}} + \text{diag}[K_{\mathbf{ff}} - Q_{\mathbf{ff}}]) \quad (3)$$

we obtain the objective function in Eq. (6). We would like to stress, however, that this modification gives *exactly* the same procedure as approximating the full GP posterior with EP. Regardless of the fact that that FITC *can* be seen as a completely different model, we aim to characterise it as an approximation to the full GP.

2.2. Variational Free Energy (VFE)

Variational inference can also be used to approximate the true posterior. We follow the derivation by Titsias [9] and bound the marginal likelihood, by instantiating extra function values on the latent Gaussian process \mathbf{u} at locations Z ,¹ followed by lower bounding the marginal likelihood. To ensure efficient calculation, $q(\mathbf{u}, \mathbf{f})$ is chosen to factorise as $q(\mathbf{u})p(\mathbf{f}|\mathbf{u})$. This removes terms with $K_{\mathbf{ff}}^{-1}$:

$$\log p(\mathbf{y}) \geq \int q(\mathbf{u}, \mathbf{f}) \log \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})p(\mathbf{u})}{p(\mathbf{f}|\mathbf{u})q(\mathbf{u})} d\mathbf{u} d\mathbf{f} \quad (4)$$

The optimal $q(\mathbf{u})$ can be found by variational calculus resulting in the lower bound:

$$\log p(\mathbf{y}) \geq \log \mathcal{N}(\mathbf{y}; 0, Q_{\mathbf{ff}} + \sigma_n^2 I) - \frac{1}{2\sigma_n^2} \text{tr}(K_{\mathbf{ff}} - Q_{\mathbf{ff}}) \quad (5)$$

2.3. Common notation

The objective functions for both VFE and FITC look very similar. In the following discussion we will refer to a common notation of their negative log marginal likelihood (NLML), which will be minimised to train the methods:

$$\mathcal{F} = \frac{N}{2} \log(2\pi) + \underbrace{\frac{1}{2} \log |Q_{\mathbf{ff}} + G|}_{\text{complexity penalty}} + \underbrace{\frac{1}{2} \mathbf{y}^\top (Q_{\mathbf{ff}} + G)^{-1} \mathbf{y}}_{\text{data fit}} + \underbrace{\frac{1}{2\sigma_n^2} \text{tr}(T)}_{\text{trace term}}, \quad (6)$$

where

$$G_{\text{FITC}} = \text{diag}[K_{\mathbf{ff}} - Q_{\mathbf{ff}}] + \sigma_n^2 I \quad G_{\text{VFE}} = \sigma_n^2 I \quad (7)$$

$$T_{\text{FITC}} = 0 \quad T_{\text{VFE}} = K_{\mathbf{ff}} - Q_{\mathbf{ff}}. \quad (8)$$

¹Matthews et al. [14] show that this procedure approximates the posterior over the entire process $f(\cdot)$ correctly.

The common objective function has three terms, of which the data fit and complexity penalty have direct analogues to the full GP. The **data fit** term penalises the data lying outside the covariance ellipse $Q_{\mathbf{ff}} + G$. The **complexity penalty** is the integral of the data fit term over all possible observations \mathbf{y} . It characterises the *volume* of possible datasets that are compatible with the data fit term. This can be seen as the mechanism of *Occam's razor* [15], by penalising the methods for being able to predict too many datasets. The **trace term** in VFE ensures that the objective function is a true lower bound to the marginal likelihood of the full GP. Without this term, VFE is identical to the earlier DTC approximation [6] which can grossly over-estimate the marginal likelihood. The trace term penalises the sum of the conditional variances at the training inputs, conditioned on the inducing inputs [16]. Intuitively, it ensures that VFE not only models this specific dataset \mathbf{y} well, but also approximates the covariance structure of the full GP $K_{\mathbf{ff}}$.

3. Comparative behaviour

As our main test case we use the one dimensional dataset² considered in [7, 9] with 200 input-output pairs. Of course, sparse methods are not necessary for this toy problem, but all of the issues we raise are illustrated nicely in this one dimensional task which can easily be plotted.

3.1. FITC can severely underestimate the noise variance, VFE overestimates it

In the full GP with Gaussian likelihood we assume a homoscedastic (input *independent*) noise model with noise variance parameter σ_n^2 . It fully characterises the uncertainty left after completely learning the latent function. In this section we show how FITC can also use the diagonal term $\text{diag}(K_{\mathbf{ff}} - Q_{\mathbf{ff}})$ in G_{FITC} as heteroscedastic (input dependent) noise [7] to account for these differences, thus, **invalidating the above interpretation of the noise variance parameter**. In fact, the FITC objective function encourages **underestimation of the noise variance**, whereas the VFE bound **encourages overestimation**. The latter is in line with previously reported biases of variational methods [17].

Fig. 1 shows the configuration most preferred by the FITC objective for a subset of 100 data points of the Snelson dataset, found by an exhaustive search for a minimum over hyperparameters, inducing inputs *and* number of inducing points. The noise is shrunk to practically zero, despite the mean prediction not going through every data point. Note how the learned mean still behaves well and how the training data lie well within the predictive variance. Only when considering predictive probabilities will this behaviour cause diminished performance. **VFE, on the other hand, is able to approximate the posterior predictive distribution almost exactly.**

For both approximations, the complexity penalty decreases with decreased noise variance, by reducing the volume of datasets that can be explained. However, for a full GP and VFE this is accompanied by a data fit penalty for data points lying far away from the predictive mean. FITC, on the other hand, has an additional mechanism to avoid this penalty: its diagonal correction term $\text{diag}(K_{\mathbf{ff}} - Q_{\mathbf{ff}})$. This term can be seen as an input dependent or **heteroscedastic noise** term (**discussed as a modelling advantage** by Snelson and Ghahramani [7]), which **is zero exactly at an inducing input**, and which grows to the prior variance away from an inducing input. By placing the inducing inputs near training data that happen to lie near the mean, the heteroscedastic noise term is locally shrunk, resulting in a reduced complexity penalty. Data points both far from the mean and far from inducing inputs do not incur a data fit penalty, as the heteroscedastic noise term

²Obtained from <http://www.gatsby.ucl.ac.uk/~snelson/>

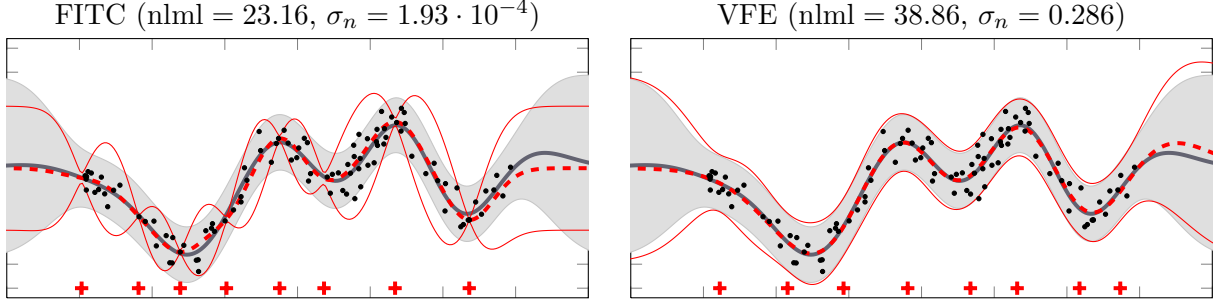


Figure 1: Behaviour of FITC and VFE on subset of 100 data points of the Snelson dataset for 8 inducing inputs (red crosses indicate inducing inputs; red lines indicate mean and 2σ) compared to the prediction of the full GP in grey. Optimised values for the full GP: $\text{nlml} = 34.15$, $\sigma_n = 0.274$

has increased around these points. This mechanism removes the need for the homoscedastic noise to explain deviations from the mean, such that σ_n^2 can be turned down to reduce the complexity penalty further.

This explains the extreme pinching (severely reduced noise variance) observed in Fig. 1. In examples with more densely packed data, there may not be any places where a near-zero noise point can be placed without incurring a huge data-fit penalty. However, inducing inputs will be placed in places where the data happens to randomly cluster around the mean, which still results in a decreased noise estimate, albeit less extreme.

Remark 1 FITC has alternative mechanisms to explain deviations from the learned function than the likelihood noise and will underestimate σ_n^2 as a consequence. In extreme cases, σ_n^2 can incorrectly be estimated to be almost zero.

As a consequence of this additional mechanism, σ_n^2 can no longer be interpreted in the same way as for VFE or the full GP. σ_n^2 is often interpreted as the amount of uncertainty in the dataset which can not be explained. Based on this interpretation, a low σ_n^2 is often used as an indication that the dataset is being fitted well. Active learning applications rely on a similar interpretation to differentiate between inherent noise, and uncertainty in the latent GP which can be reduced. FITC's different interpretation of σ_n^2 will cause efforts like these to fail.

VFE, on the other hand, is biased towards over-estimating the noise variance, because of both the data fit and the trace term. $Q_{\mathbf{ff}} + \sigma_n^2 I$ has $N - M$ eigenvectors with an eigenvalue of σ_n^2 , since the rank of $Q_{\mathbf{ff}}$ is M . Any component of \mathbf{y} in these directions will result in a larger data fit penalty than for $K_{\mathbf{ff}}$, which can only be reduced by increasing σ_n^2 . The trace term can also be reduced by increasing σ_n^2 .

Remark 2 The VFE objective tends to over-estimate the noise variance compared to the full GP.

3.2. VFE improves with additional inducing inputs, FITC may ignore them

Here we investigate the behaviour of each method when more inducing inputs are added. For both methods, adding an extra inducing input gives it an extra basis function to model the data with. We discuss how and why VFE always improves, while FITC may deteriorate.

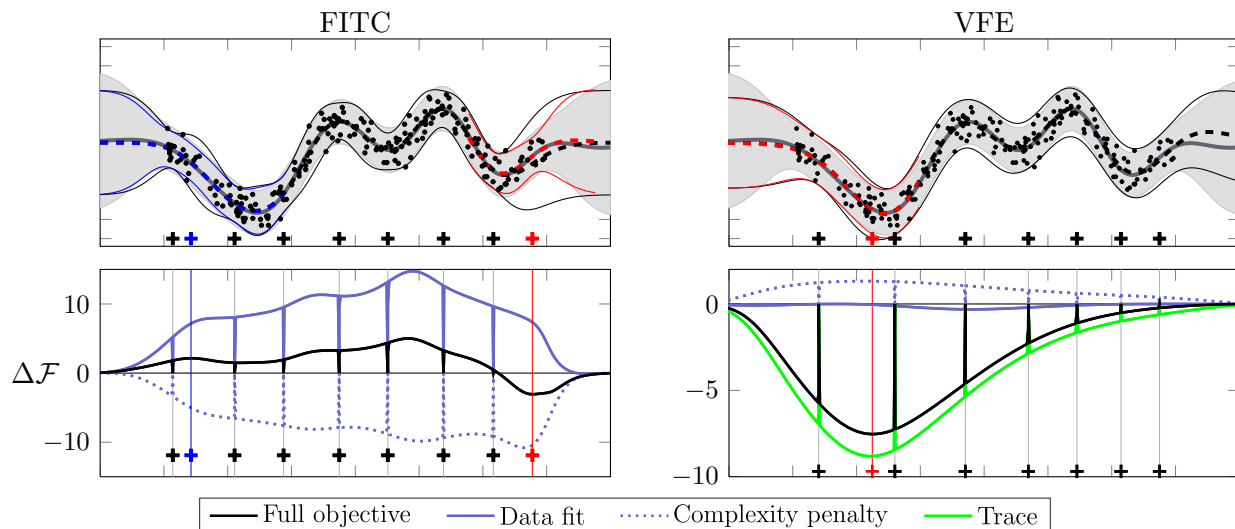


Figure 2: *Top*: Fits for FITC and VFE on 200 data points of the Snelson dataset for 7 optimised inducing inputs (black). *Bottom*: Change in objective function from adding an inducing input anywhere along the x -axis. The overall change is decomposed into the change in the individual terms (see legend). Two particular additional inducing inputs and their effect on the predictive distribution shown in red and blue.

Fig. 2 shows an example of how the objective function changes when an inducing input is added anywhere in the input domain. While the change in objective function looks reasonably smooth overall, there are pronounced spikes for both, FITC and VFE. These return the objective to the value without the additional inducing input and occur at the locations of existing inducing inputs. We discuss the general change first before explaining the spikes.

Mathematically, adding an inducing input corresponds to a rank 1 update of $Q_{\mathbf{ff}}$, and can be shown to always improve VFE’s bound³, see Supplement for a proof. VFE’s complexity penalty increases due to an extra non-zero eigenvalue in $Q_{\mathbf{ff}}$, but gains in data fit and trace.

Remark 3 *VFE’s posterior and marginal likelihood approximation become more accurate (or remain unchanged) regardless of where a new inducing input is placed.*

For FITC, the objective can change either way. Regardless of the change in objective, the heteroscedastic noise is decreased at all points (see Supplement for proof). For a squared exponential kernel, the decrease is strongest around the newly placed inducing input. This decrease has two effects. One, it reduces the complexity penalty since the diagonal component of $Q_{\mathbf{ff}} + G$ is reduced and replaced by a more strongly correlated $Q_{\mathbf{ff}}$. Two, it worsens the data fit term as the heteroscedastic term is required to fit the data when the homoscedastic noise is underestimated. Fig. 2 shows reduced error bars with several data points now outside of the 95% prediction bars. Also shown is a case where an additional inducing input improves the objective, where the extra correlations outweigh the reduced heteroscedastic noise.

Both VFE and FITC exhibit pathological behaviour (spikes) when inducing inputs are clumped, that is, when they are placed exactly on top of each other. In this case, the objective function has the same value as when all duplicate inducing inputs were removed (see Supplement for a proof).

³de G. Matthews [18] independently proved this result by considering the KL divergence between processes. Titsias [9] proved this result for the special case when the new inducing input is selected from the training data.

In other words, for all practical purposes, a model with duplicate inducing inputs reduces to a model with fewer, individually placed inducing inputs.

Theoretically, these pathologies only occur at single points, such that no gradients towards or away from them could exist and they would never be encountered. In practise, however, these peaks are widened by a finite *jitter* that is added to $K_{\mathbf{u}\mathbf{u}}$ to ensure it remains well conditioned enough to be invertible. This finite width provides the gradients that allow an optimiser to detect these configurations.

As VFE always improves with additional inducing inputs, these configurations must correspond to maxima of the optimisation surface and clumping of inducing inputs does not occur for VFE. For FITC, configurations with clumped inducing inputs can and often do correspond to minima of the optimisation surface. By placing them on top of each other, FITC can avoid the penalty of adding an extra inducing input and can gain the bonus from the heteroscedastic noise. Clumping, thus, constitutes a mechanism that allows FITC to effectively remove inducing inputs at no cost. In practise we find that convergence towards these minima can be slow.

We illustrate this behaviour in Fig. 3 for 15 randomly initialised inducing inputs. FITC places some of them exactly on top of each other, whereas VFE spreads them out and recovers the full GP well.

Remark 4 In FITC, having a good approximation $Q_{\mathbf{ff}}$ to $K_{\mathbf{ff}}$ needs to be traded off with the gains coming from the heteroscedastic noise. *FITC does not always favour a more accurate approximation to the GP.*

Remark 5 FITC avoids losing the gains of the heteroscedastic noise by placing inducing inputs on top of each other, effectively removing them.

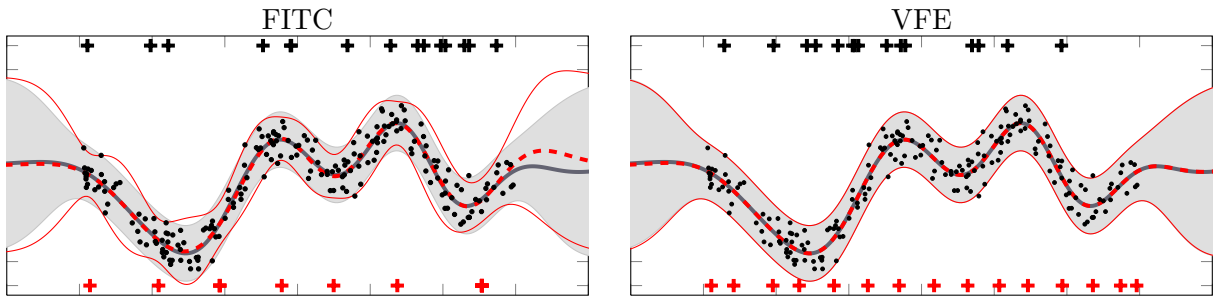


Figure 3: Fits for 15 inducing inputs for FITC and VFE (initial as black crosses, optimised red crosses). Even following joint optimisation of inducing inputs and hyperparameters, FITC avoids the penalty of added inducing inputs by clumping some of them on top of each other (shown as a single red cross). VFE spreads out the inducing inputs to get closer to the true full GP posterior.

3.3. FITC does not recover the true posterior, VFE does

In the previous section we showed that FITC has trouble using additional resources to model the data, and that the optimiser moves away from a more accurate approximation to the full GP. Here we show that this behaviour is inherent to the FITC objective function.

Both VFE and FITC *can* recover the true posterior by placing an inducing input on every training input [9, 12]. For VFE, this must be a *global* minimum, since the KL gap to the true marginal likelihood is zero. For FITC, however, this solution is merely a saddle point. The derivative of the inducing inputs is zero for this configuration, but the objective function can still be improved. As with the clumping behaviour, adding jitter subtly makes this behaviour more obvious by perturbing the gradients. In Table 1 we show this behaviour on a subset of 100 data points of the Snelson dataset. VFE is at a minimum and does not move the inducing inputs, while FITC improves its objective and moves the inducing inputs considerably.

Method	NLML init	NLML final	rms distance \mathbf{Z}
Full GP	—	33.8923	—
VFE	33.8923	33.8923	0
FITC	33.8923	28.3869	10.0830

Table 1: Results of optimising VFE and FITC after initialising at the solution that gives the correct posterior and marginal likelihood. We observe that FITC moves to a significantly different solution.

Remark 6 *FITC generally does not recover the full GP, even when it has enough resources.*

3.4. FITC relies on local optima

So far, we have observed some cases where FITC fails to produce results in line with the full GP, and characterised why. However, in practice, FITC has performed well, and pathological behaviour is not always observed. In this section we discuss the optimiser dynamics and show that they help FITC behave reasonably.

To demonstrate this behaviour, we consider a 4d toy dataset: 1024 training and 1024 test samples drawn from a 4d Gaussian Process with isotropic squared exponential covariance function ($l = 1.5, s_f = 1$) and true noise variance $\sigma_n^2 = 0.01$. We fit both FITC and VFE to this dataset with the number of inducing inputs ranging from 16 to 1024, and compare them to the full GP in Fig. 4.

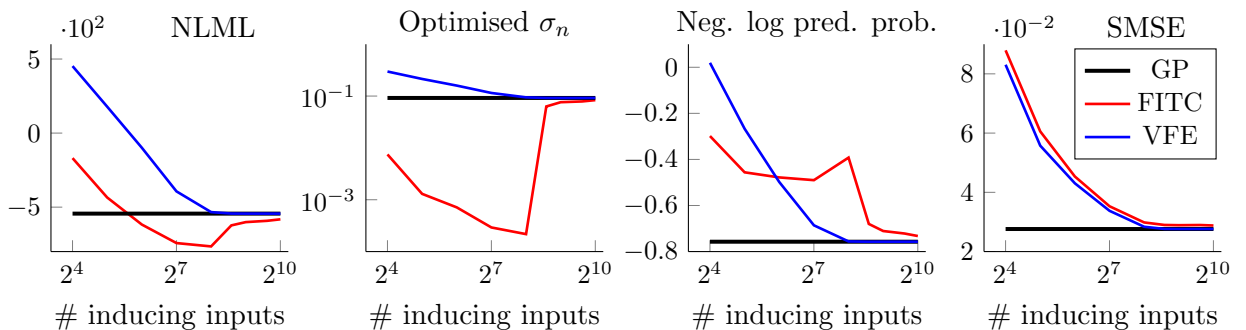


Figure 4: Optimisation behaviour of VFE and FITC for varying number of inducing inputs compared to the full GP. We show the objective function (negative log marginal likelihood), the optimised noise σ_n , the negative log predictive probability and standardised mean squared error as defined in [1].

VFE monotonically approaches the values of the full GP but initially overestimates the noise variance, as discussed in Section 3.1. Conversely, we can identify three regimes for the objective function of FITC: 1) Monotonic improvement for few inducing inputs, 2) a region where FITC over-estimates the marginal likelihood, and 3) recovery towards the full GP for many inducing inputs. Predictive performance follows a similar trend, first improving, then declining while the bound is estimated to be too high, followed by a recovery. The recovery is counter to the usual intuition that over-fitting worsens when adding more parameters.

We explain the behaviour in these three regimes as follows: When the number of inducing inputs are severely limited (regime 1), FITC needs to place them such that $K_{\mathbf{ff}}$ is well approximated. This correlates most points to some degree, and ensures a reasonable data fit term. The marginal likelihood is under-estimated due to lack of a flexibility in $Q_{\mathbf{ff}}$.

As the number of inducing inputs increases (regime 2), the marginal likelihood is over-estimated and the noise drastically under-estimated. Additionally, performance in terms of log predictive probability deteriorates. This is the regime closest to FITC’s behaviour in Fig. 1. There are enough inducing inputs such that they can be placed such that a bonus can be gained from the heteroscedastic noise, without gaining a complexity penalty from losing long scale correlations.

Finally, in regime 3, FITC starts to behave more like a regular GP in terms of marginal likelihood, predictive performance and noise variance parameter σ_n . FITC’s ability to use heteroscedastic noise is reduced as the approximate covariance matrix $Q_{\mathbf{ff}}$ is closer to the true covariance matrix $K_{\mathbf{ff}}$ when many (initial) inducing input are spread over the input space.

In the previous section we showed that after adding a new inducing input, a better minimum obtained without the extra inducing input could be recovered by clumping. So it is clear that the minimum that was found with fewer active inducing inputs still exists in the optimisation surface of many inducing inputs; the optimiser just does not find it.

Remark 7 When running FITC with many inducing inputs its resemblance to the full GP solution relies on local optima, rather than the objective function changing.

3.5. VFE is hindered by local optima

So far we have seen that the VFE objective function is a true lower bound on the marginal likelihood and does not share the same pathologies as FITC. Thus, when optimising, we really are interested in finding a global optimum. The VFE objective function is not completely trivial to optimise, and often tricks, such as initialising the inducing inputs with k-means and initially fixing the hyperparameters [19, 20], are required to find a good optimum. Others have commented that VFE has the tendency to underfit [3]. Here we investigate the underfitting claim and relate it to optimisation behaviour.

As this behaviour is not observable in our 1D dataset, we illustrate it on the pumadyn32nm dataset⁴ (32 dimension, 7168 training, 1024 test), see Table 2 for the results of a representative run with random initial conditions and 40 inducing inputs.

Using a squared exponential ARD kernel with separate lengthscales for every dimension, a full GP on a subset of data identified four lengthscales as important to model the data while scaling the other 28 length scales to large values (in the table we plot the inverse lengthscales).

FITC was consistently able to identify the same four lengthscales and performed similarly compared

⁴obtained from <http://www.cs.toronto.edu/~delve/data/datasets.html>






Method	NLML/ N	σ_n	inv. lengthscales	RMSE
GP (SoD)	-0.099	0.196		0.209
FITC	-0.145	0.004		0.212
VFE	1.419	1		0.979
VFE (frozen)	0.151	0.278		0.276
VFE (init FITC)	-0.096	0.213		0.212

Table 2: Results for pumadyn32nm dataset. We show negative log marginal likelihood (NLML) divided by number of training points, the optimised noise variance σ_n^2 , the ten most dominant inverse lengthscales and the RMSE on test data. Methods are full GP on 2048 training samples, FITC, VFE, VFE with initially frozen hyperparameters, VFE initialised with the solution obtained by FITC.

to the full GP but scaled down the noise variance σ_n^2 to almost zero. VFE, on the other hand, **was unable to identify these relevant lengthscales when jointly optimising the hyperparameters** and inducing inputs, and only identified some of the them when initially freezing the hyperparameters. One might say that VFE “underfits” in this case. However, we can show that VFE still *recognises* a good solution: When we initialised VFE with the FITC solution it consistently obtained a good fit to the model with correctly identified lengthscales and a noise variance that was close to the full GP.

Remark 8 *VFE has a tendency to find under-fitting solutions. However, this is an optimisation issue. The bound correctly identifies good solutions.*

4. Conclusion

In this work, we have thoroughly investigated and characterised the differences between FITC and VFE, both in terms of their objective function and their behaviour observed during practical optimisation. We highlight several instances of undesirable behaviour in the FITC objective: over-estimation of the marginal likelihood, sometimes severe under-estimation of the noise variance parameter, wasting of modelling resources and not recovering the true posterior. The common practice of using the noise variance parameter as a diagnostic for good model fitting is unreliable. In contrast, VFE is a true bound to the marginal likelihood of the full GP and behaves predictably: It correctly identifies good solutions, always improves with extra resources and recovers the true posterior when possible. In practice however, **the pathologies of the FITC objective do not always show up**, thanks to “good” local optima and (unintentional) early stopping. While VFE’s objective recognises a good configuration, **it is often more susceptible to local optima and harder to optimise than FITC**.

However, based on the superior properties of the VFE objective function, **we recommend using VFE**, while paying attention to optimisation difficulties. These can be mitigated by careful initialisation, random restarts, other optimisation tricks and comparison to the FITC solution to guide VFE optimisation.

Acknowledgements

We would like to thank Alexander Matthews, Thang Bui, and Richard Turner for useful discussions.

References

- [1] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning series)*. The MIT Press, 2005.
- [2] Miguel Lázaro-Gredilla, Joaquin Quiñonero-Candela, Carl Edward Rasmussen, and Aníbal R Figueiras-Vidal. Sparse spectrum gaussian process regression. *The Journal of Machine Learning Research*, 11:1865–1881, 2010.
- [3] Miguel Lázaro-Aredilla and Anibal Figueiras-Vidal. Inter-domain gaussian processes for sparse inference using inducing features. In *Advances in Neural Information Processing Systems*, pages 1087–1095, 2009.
- [4] Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in Neural Information Processing Systems*, pages 1313–1320, 2009.
- [5] Zichao Yang, Alexander J. Smola, Le Song, and Andrew Gordon Wilson. A la carte - learning fast kernels. In *Artificial Intelligence and Statistics*, December 2015.
- [6] Matthias Seeger, Christopher K. I. Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse gaussian process regression. In Christopher Bishop and B. J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.
- [7] Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Neural Information Processing Systems*, volume 18, 2006.
- [8] Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.
- [9] Michalis K. Titsias. **Variational learning of inducing variables in sparse gaussian processes**. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, pages 567–574, 2009.
- [10] Alex J Smola and Peter Bartlett. Sparse greedy gaussian process regression. In *Advances in Neural Information Processing Systems 13*. Citeseer, 2001.
- [11] Lehel Csató and Manfred Opper. Sparse on-line gaussian processes. *Neural computation*, 14(3):641–668, 2002.
- [12] Edward Snelson. *Flexible and efficient Gaussian process models for machine learning*. PhD thesis, University College London, 2007.
- [13] Yuan Qi, Ahmed H. Abdel-Gawad, and Thomas P. Minka. Sparse-posterior gaussian processes for general likelihoods. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, 2010.
- [14] Alexander G. de G. Matthews, James Hensman, Richard E. Turner, and Zoubin Ghahramani. On sparse variational methods and the kullback-leibler divergence between stochastic processes. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence*

and Statistics, 2016.

- [15] Carl Edward Rasmussen and Zoubin Ghahramani. Occam’s razor. In *Advances in Neural Information Processing Systems 13*, 2001.
- [16] Michaelis K. Titsias. Variational model selection for sparse gaussian process regression. Technical report, University of Manchester, 2009.
- [17] 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.
- [18] Alexander G. de G. Matthews. PhD thesis, University of Cambridge, in submission.
- [19] James Hensman, Alexander G de G Matthews, and Zoubin Ghahramani. Scalable variational gaussian process classification. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, 2015.
- [20] James Hensman, Nicolo Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Conference on Uncertainty in Artificial Intelligence*, September 2013.

Supplemental Material

A. Proofs for additional inducing inputs

In this section we give proofs to the claims on how the objective functions for FITC and VFE change upon adding an inducing input. For this, we first restate the objective function:

$$\mathcal{F} = \frac{N}{2} \log(2\pi) + \underbrace{\frac{1}{2} \log |Q_{\mathbf{ff}} + G|}_{\text{complexity penalty}} + \underbrace{\frac{1}{2} \mathbf{y}^\top (Q_{\mathbf{ff}} + G)^{-1} \mathbf{y}}_{\text{data fit}} + \underbrace{\frac{1}{2\sigma_n^2} \text{tr}(T)}_{\text{trace term}}, \quad (\text{A.9})$$

where

$$G_{\text{FITC}} = \text{diag}[K_{\mathbf{ff}} - Q_{\mathbf{ff}}] + \sigma_n^2 I \quad G_{\text{VFE}} = \sigma_n^2 I \quad (\text{A.10})$$

$$T_{\text{FITC}} = 0 \quad T_{\text{VFE}} = K_{\mathbf{ff}} - Q_{\mathbf{ff}}. \quad (\text{A.11})$$

$K_{\mathbf{ff}}$ denotes the covariance matrix and $Q_{\mathbf{ff}} = K_{\mathbf{fu}} K_{\mathbf{uu}}^{-1} K_{\mathbf{uf}}$ the approximate covariance matrix. Note that the approximate covariance matrix $Q_{\mathbf{ff}}$ is the only quantity depending on the inducing inputs.

The main results we show in this supplement are:

1. Adding an inducing input corresponds to a rank 1 update of the approximate covariance matrix $Q_{\mathbf{ff}} = K_{\mathbf{fu}} K_{\mathbf{uu}}^{-1} K_{\mathbf{uf}}$ (see Section A.1)
2. For VFE the objective function can never get worse by adding an inducing input anywhere in the input domain (see Section A.2)
3. For FITC the heteroscedastic noise always decreases when adding an inducing input anywhere in the input domain (see Section A.3)
4. Adding an inducing input on top of an existing inducing input does not change the objective function, neither for FITC nor for VFE (see Section A.4)

A.1. Adding an inducing input anywhere

In this section we show that adding an inducing input corresponds to a rank 1 update of the $Q_{\mathbf{ff}}$ matrix for the case without jitter.

Let $K_{\mathbf{uu}}$ denote the covariance matrix of the M inducing inputs. We then add a new $M + 1$ st inducing input and denote all quantities depending on the new set of $M + 1$ inducing inputs by a superscript $+$.

The updated approximate covariance matrix $Q_{\mathbf{ff}}^+$ is then given by:

$$Q_{\mathbf{ff}}^+ = K_{\mathbf{fu}}^+ (K_{\mathbf{uu}}^+)^{-1} K_{\mathbf{uf}}^+ \quad (\text{A.12})$$

We proceed by first computing an explicit expression for the $M + 1 \times M + 1$ matrix $(K_{\mathbf{uu}}^+)^{-1}$ before computing $Q_{\mathbf{ff}}^+$. For this we employ the block matrix inversion formula⁵, see Eq. (A.16)

$$\begin{aligned} (K_{\mathbf{uu}}^+)^{-1} &= \begin{pmatrix} K_{\mathbf{uu}} & k_{\mathbf{u}} \\ k_{\mathbf{u}}^\top & k \end{pmatrix}^{-1} \\ &= \begin{pmatrix} K_{\mathbf{uu}}^{-1} + \frac{1}{c} \mathbf{a} \mathbf{a}^\top & -\frac{1}{c} \mathbf{a} \\ -\frac{1}{c} \mathbf{a}^\top & \frac{1}{c} \end{pmatrix} \quad \mathbf{a} = K_{\mathbf{uu}}^{-1} k_{\mathbf{u}} \end{aligned}$$

⁵https://en.wikipedia.org/wiki/Block_matrix#Block_matrix_inversion

where $c = k - k_{\mathbf{u}}^{\top} K_{\mathbf{uu}}^{-1} k_{\mathbf{u}}$ is the Schur complement of $K_{\mathbf{uu}}$, $k_{\mathbf{u}}^{\top} = (k(Z_1, Z_{M+1}), \dots, k(Z_M, Z_{M+1}))$ is the vector of covariances between the old inducing inputs and the added inducing input and $k = k(Z_{M+1}, Z_{M+1})$ is the covariance function evaluated at the new inducing input. Note that c needs to be non-zero for this expression to make sense.

$$K_{\mathbf{uf}}^+ = \begin{pmatrix} K_{\mathbf{uf}} \\ k_{\mathbf{f}}^{\top} \end{pmatrix}$$

where $k_{\mathbf{f}}^{\top} = (k(Z_{M+1}, x_1), \dots, k(Z_{M+1}, x_N))$ is the vector of covariances between the data points and the new inducing input.

We can now compute $Q_{\mathbf{ff}}^+$ by the product in Eq. (A.12) to see that is is indeed given by a rank 1 update of $Q_{\mathbf{ff}}$

$$\begin{aligned} Q_{\mathbf{ff}}^+ &= K_{\mathbf{fu}}^+ (K_{\mathbf{uu}}^+)^{-1} K_{\mathbf{uf}}^+ \\ &= K_{\mathbf{fu}} K_{\mathbf{uu}}^{-1} K_{\mathbf{uf}} + \frac{1}{c} (K_{\mathbf{fu}} \mathbf{a} \mathbf{a}^{\top} K_{\mathbf{uf}} + K_{\mathbf{fu}} \mathbf{a} \mathbf{a}^{\top} K_{\mathbf{uf}} \\ &\quad - K_{\mathbf{fu}} \mathbf{a} k_{\mathbf{f}}^{\top} - k_{\mathbf{f}} \mathbf{a}^{\top} K_{\mathbf{uf}} + k_{\mathbf{f}} k_{\mathbf{f}}^{\top}) \\ &= K_{\mathbf{fu}} K_{\mathbf{uu}}^{-1} K_{\mathbf{uf}} + \frac{1}{c} (K_{\mathbf{fu}} \mathbf{a} - k_{\mathbf{f}}) (K_{\mathbf{fu}} \mathbf{a} - k_{\mathbf{f}})^{\top} \\ &= K_{\mathbf{fu}} K_{\mathbf{uu}}^{-1} K_{\mathbf{uf}} + \mathbf{b} \mathbf{b}^{\top} \\ &= Q_{\mathbf{ff}} + \mathbf{b} \mathbf{b}^{\top} \\ Q_{\mathbf{ff}}^+ &= Q_{\mathbf{ff}} + \mathbf{b} \mathbf{b}^{\top} \end{aligned} \tag{A.13}$$

where we have introduced the rank 1 update vector $\mathbf{b} = \frac{1}{\sqrt{c}} (K_{\mathbf{fu}} K_{\mathbf{uu}}^{-1} k_{\mathbf{u}} - k_{\mathbf{f}})$.

Thus, in the case of no *jitter*, the update is indeed given by a rank 1 update. These results also extend to the case with finite jitter, which is then absorbed into the definition of $K_{\mathbf{uu}}$ and k , respectively.

A.2. The VFE objective function always improves when adding an additional inducing input

We now compute the change in objective function when adding the $M + 1$ st inducing input:

$$\begin{aligned} 2(\mathcal{F}^+ - \mathcal{F}) &= \log |Q_{\mathbf{ff}}^+ + \sigma_n^2 I| - \log |Q_{\mathbf{ff}} + \sigma_n^2 I| + \mathbf{y}^{\top} (Q_{\mathbf{ff}}^+ + \sigma_n^2 I)^{-1} \mathbf{y} - \mathbf{y}^{\top} (Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1} \mathbf{y} \\ &\quad + \frac{1}{\sigma_n^2} \text{tr}(K_{\mathbf{ff}} - Q_{\mathbf{ff}}^+) - \frac{1}{\sigma_n^2} \text{tr}(K_{\mathbf{ff}} - Q_{\mathbf{ff}}) \\ &= \log |Q_{\mathbf{ff}} + \mathbf{b} \mathbf{b}^{\top} + \sigma_n^2 I| - \log |Q_{\mathbf{ff}} + \sigma_n^2 I| \\ &\quad + \mathbf{y}^{\top} (Q_{\mathbf{ff}} + \mathbf{b} \mathbf{b}^{\top} + \sigma_n^2 I)^{-1} \mathbf{y} - \mathbf{y}^{\top} (Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1} \mathbf{y} - \frac{1}{\sigma_n^2} \text{tr}(\mathbf{b} \mathbf{b}^{\top}) \end{aligned}$$

To deal with the log-determinant-terms and the inverses, we employ the Matrix determinant lemma⁶ and the Sherman–Morrison formula⁷, respectively

$$\begin{aligned}
2(\mathcal{F}^+ - \mathcal{F}) &= \log(1 + \mathbf{b}^\top (Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1} \mathbf{b}) + \log |Q_{\mathbf{ff}} + \sigma_n^2 I| - \log |Q_{\mathbf{ff}} + \sigma_n^2 I| \\
&\quad + \mathbf{y}^\top (Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1} \mathbf{y} - \mathbf{y}^\top \frac{(Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1} \mathbf{b} \mathbf{b}^\top (Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1}}{1 + \mathbf{b}^\top (Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1} \mathbf{b}} \mathbf{y} \\
&\quad - \mathbf{y}^\top (Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1} \mathbf{y} + \frac{1}{\sigma_n^2} \text{tr}(\mathbf{b} \mathbf{b}^\top) \\
&= \log(1 + \mathbf{b}^\top (Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1} \mathbf{b}) - \frac{1}{\sigma_n^2} \text{tr}(\mathbf{b} \mathbf{b}^\top) \\
&\quad - \mathbf{y}^\top \frac{(Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1} \mathbf{b} \mathbf{b}^\top (Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1}}{1 + \mathbf{b}^\top (Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1} \mathbf{b}} \mathbf{y}
\end{aligned}$$

We can bound the first two terms by noting

$$\begin{aligned}
\text{tr}(\mathbf{b} \mathbf{b}^\top) &= \mathbf{b}^\top \mathbf{b} \\
\log(1 + x) &\leq x \\
\mathbf{b}^\top (Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1} \mathbf{b} &\leq \frac{1}{\sigma_n^2} \mathbf{b}^\top \mathbf{b}
\end{aligned}$$

Thus,

$$\log(1 + \mathbf{b}^\top (Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1} \mathbf{b}) - \frac{1}{\sigma_n^2} \text{tr}(\mathbf{b} \mathbf{b}^\top) \leq 0$$

and equality holds for $\mathbf{b} = 0$, as is the case when both inducing inputs lie on top of each other. It remains to show that the term including the \mathbf{y} s (including its sign) is non-positive. This can be shown quite easily:

$$\begin{aligned}
-\mathbf{y}^\top \frac{(Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1} \mathbf{b} \mathbf{b}^\top (Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1}}{1 + \mathbf{b}^\top (Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1} \mathbf{b}} \mathbf{y} &= -\frac{(\mathbf{y}^\top (Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1} \mathbf{b})^2}{1 + \mathbf{b}^\top (Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1} \mathbf{b}} \\
&\leq -(\mathbf{y}^\top (Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1} \mathbf{b})^2 \\
&\leq 0
\end{aligned}$$

where the second to last inequality holds as $(Q_{\mathbf{ff}} + \sigma_n^2 I)^{-1}$ is positive definite. Equalities hold, again, if $\mathbf{b} = 0$ which corresponds to duplication of an existing inducing input.

This concludes the proof that the VFE objective function always improves or stays the same.

The change of the objective function for FITC is less clear than for VFE. We can give no proofs about the changes in general. In experiments, we observe that the change in the data fit term can be positive or negative, whereas the complexity penalty term seems to always improve by adding an inducing input. We hypothesise that this is indeed the case but cannot give a proof for this claim.

In this section we have assumed that all matrix inverses exist and that for duplication of an inducing input we find $\mathbf{b} = 0$. However, for a duplicate inducing input, the matrix $K_{\mathbf{uu}}$ becomes singular,

⁶https://en.wikipedia.org/wiki/Matrix_determinant_lemma

⁷https://en.wikipedia.org/wiki/Sherman-Morrison_formula

such that care has to be taken when reasoning. In Section A.4 we show that these arguments can, indeed, be made rigorous and that $\mathbf{b} = 0$ for duplicate inducing inputs. Thus, when duplicating an inducing input, the VFE and FITC objective functions do not change. For VFE, this configuration corresponds to a maximum of the objective function.

A.3. The heteroscedastic noise is decreased when new inducing inputs are added

While the objective function can change either way, the heteroscedastic noise, which is given by $\text{diag}(K_{\mathbf{ff}} - Q_{\mathbf{ff}})$ always decreases or remains the same when a new inducing input is added:

$$\text{diag}(K_{\mathbf{ff}} - Q_{\mathbf{ff}}^+) = \text{diag}(K_{\mathbf{ff}} - (Q_{\mathbf{ff}} + \mathbf{b}\mathbf{b}^\top)) \quad (\text{A.14})$$

$$= \text{diag}(K_{\mathbf{ff}} - Q_{\mathbf{ff}}) - \text{diag}(\mathbf{b}\mathbf{b}^\top) \quad (\text{A.15})$$

The diagonal elements of $\mathbf{b}\mathbf{b}^\top$ are given by b_m^2 , which are always larger or equal to zero, such that the heteroscedastic noise always decreases (or stays the same).

A.4. Adding an inducing input on top of another inducing input

In this section we show that duplication of an inducing input, that is, placing an additional inducing input on top of an existing one, does not change the approximate covariance matrix: $Q_{\mathbf{ff}}^+ = Q_{\mathbf{ff}}$.

One might be tempted to use Eq. (A.13) to evaluate the update when placing an additional inducing input on top of an existing one. In that case $K_{\mathbf{uu}}^{-1}k_{\mathbf{u}} = \hat{e}_M$, where \hat{e}_M denotes the indicator vector with a one at the M th position and zeros otherwise, and $K_{\mathbf{fu}}K_{\mathbf{uu}}^{-1}k_{\mathbf{u}} = k_{\mathbf{f}}$ suggesting $\mathbf{b} = 0$. However, in this case, the Schur complement vanishes as well, $c = 0$.

In the following we show that this reasoning can be made exact by considering a finite *jitter* term ϵI added onto $K_{\mathbf{uu}}^+$ before inversion. The result can be expanded to second order in ϵ and the limit $\epsilon \rightarrow 0$ then leads to the desired result. The intuition behind the fact that Eq. (A.12) is well behaved, even if $K_{\mathbf{uu}}^+$ is singular, is, that the eigenvector of $K_{\mathbf{uu}}^+$ that corresponds to the zero eigenvalue is never excited by the matrix $K_{\mathbf{uf}}^+$ which has a duplicate row. The eigenvector only has two non-zero elements, which have the same absolute value but different signs, thus cancelling with the duplicate rows in $K_{\mathbf{uf}}^+$.

Moreover, we obtain a correction term that scales with the jitter. For reasons of numerical stability, one has to employ some form of regularisation of the (possibly) singular matrix $K_{\mathbf{uu}}$ in practise. One common way that is implemented in many toolboxes is the constant jitter ϵI introduced above. We assume that the original $K_{\mathbf{uu}}$ is non-singular for now.

Similarly to before, we again employ the block matrix inversion formula⁸ of the following form:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} + A^{-1}BF^{-1}CA^{-1} & -A^{-1}BF^{-1} \\ -F^{-1}CA^{-1} & F^{-1} \end{pmatrix} \quad (\text{A.16})$$

where $F = D - CA^{-1}B$ is the Schur complement of A .

$$(K_{\mathbf{uu}}^+ + \epsilon I_{M+1 \times M+1})^{-1} = \begin{pmatrix} K_{\mathbf{uu}} + \epsilon I_{M \times M} & k_{\mathbf{u}} \\ k_{\mathbf{u}}^\top & k + \epsilon \end{pmatrix}^{-1} \quad (\text{A.17})$$

⁸https://en.wikipedia.org/wiki/Block_matrix#Block_matrix_inversion

where $k_{\mathbf{u}}^{\top} = (k(Z_1, Z_M), k(Z_2, Z_M), \dots, k(Z_M, Z_M))$ and $k = k(Z_M, Z_M)$ similarly to before. In order to perform the inversion, we expand the inverses in Eq. (A.16) to second order in ϵ :

$$\begin{aligned}
(K_{\mathbf{uu}} + \epsilon I)^{-1} &\approx K_{\mathbf{uu}}^{-1}(1 - \epsilon K_{\mathbf{uu}}^{-1} + \epsilon^2 K_{\mathbf{uu}}^{-2} - \epsilon^3 K_{\mathbf{uu}}^{-3}) & A^{-1} \\
(K_{\mathbf{uu}} + \epsilon I)^{-1} k_{\mathbf{u}} &\approx \hat{e}_M - \epsilon k_{\mathbf{u}}^{-1} + \epsilon^2 K_{\mathbf{uu}}^{-1} k_{\mathbf{u}}^{-1} - \epsilon^3 K_{\mathbf{uu}}^{-2} k_{\mathbf{u}}^{-1} & A^{-1} B \\
k_{\mathbf{u}}^T (K_{\mathbf{uu}} + \epsilon I)^{-1} &\approx \hat{e}_M^T - \epsilon (k_{\mathbf{u}}^{-1})^T + \epsilon^2 (k_{\mathbf{u}}^{-1})^T K_{\mathbf{uu}}^{-1} - \epsilon^3 (k_{\mathbf{u}}^{-1})^T K_{\mathbf{uu}}^{-2} & CA^{-1} \\
k_{\mathbf{u}}^T (K_{\mathbf{uu}} + \epsilon)^{-1} k_{\mathbf{u}} &\approx k - \epsilon + \epsilon^2 k^{-1} - \epsilon^3 (k_{\mathbf{u}}^{-1})^T k_{\mathbf{u}}^{-1} & CA^{-1} B \\
k + \epsilon - k_{\mathbf{u}}^T (K_{\mathbf{uu}} + \epsilon)^{-1} k_{\mathbf{u}} &\approx 2\epsilon(1 - \frac{\epsilon}{2} k^{-1} + \frac{\epsilon^2}{2} (k_{\mathbf{u}}^{-1})^T k_{\mathbf{u}}^{-1}) & \text{Schur} \\
(k + \epsilon - k_{\mathbf{u}}^T (K_{\mathbf{uu}} + \epsilon)^{-1} k_{\mathbf{u}})^{-1} &\approx \frac{1}{2\epsilon} + \frac{1}{4} k^{-1} - \frac{\epsilon}{4} (k_{\mathbf{u}}^{-1})^T k_{\mathbf{u}}^{-1} + \frac{\epsilon}{8} (k^{-1})^2 & \text{Schur}^{-1}
\end{aligned}$$

where $\hat{e}_M = (0, \dots, 0, 1)^{\top}$ is the indicator vector with a one at position M and $k_{\mathbf{u}}^{-1} = (k^{-1}(Z_1, Z_M), \dots, k^{-1}(Z_M, Z_M))^{\top}$ is the M th column of $K_{\mathbf{uu}}^{-1}$. Note that the elements of $k_{\mathbf{u}}^{-1}$ are not element wise inverses but elements of an inverse matrix! Analogously, k^{-1} denotes the (M, M) element of the matrix $K_{\mathbf{uu}}^{-1}$.

$$\begin{aligned}
(K_{\mathbf{uu}} + \epsilon)^{-1} &= \left(\begin{array}{c|c} K_{\mathbf{uu}}^{-1} & \begin{smallmatrix} 0 \\ \vdots \\ 0 \end{smallmatrix} \\ \hline 0 & \dots & 0 & 0 \end{array} \right) - \epsilon \left(\begin{array}{c|c} K_{\mathbf{uu}}^{-2} & \begin{smallmatrix} 0 \\ \vdots \\ 0 \end{smallmatrix} \\ \hline 0 & \dots & 0 & 0 \end{array} \right) + \frac{\epsilon}{2} \left(\begin{array}{c|c} k_{\mathbf{u}}^{-1} (k_{\mathbf{u}}^{-1})^{\top} & \begin{smallmatrix} 0 \\ \vdots \\ 0 \end{smallmatrix} \\ \hline 0 & \dots & 0 & 0 \end{array} \right) \\
&+ \frac{1}{2} \left(\begin{array}{c|c} 0 & 0 \\ 0_{M-1 \times M-1} & \vdots \\ 0 & 0 \\ \hline 0 & \dots & 0 & \epsilon^{-1} \\ 0 & \dots & 0 & -\epsilon^{-1} \end{array} \right) \\
&+ \frac{1}{4} \left(\begin{array}{c|c} 0 & 0 \\ 0_{M-1 \times M-1} & \vdots \\ 0 & 0 \\ \hline 0 & \dots & 0 & k^{-1} \\ 0 & \dots & 0 & -k^{-1} \end{array} \right) + \frac{1}{2} \left(\begin{array}{c|c} 0_{M-1 \times M-1} & -k_{u \setminus m}^{-1} \\ \hline -(k_{u \setminus m}^{-1})^T & -2k^{-1} \end{array} \middle| \begin{array}{c} k_{\mathbf{u}}^{-1} \\ 0 \end{array} \right) \\
&+ \frac{\epsilon}{4} \left(\begin{array}{c|c} 0_{M-1 \times M-1} & 2K_{\mathbf{uu}}^{-1} k_{\mathbf{u}}^{-1} - k^{-1} k_{\mathbf{u}}^{-1} \\ \hline (2K_{\mathbf{uu}}^{-1} k_{\mathbf{u}}^{-1} - k^{-1} k_{\mathbf{u}}^{-1})^T & -(2K_{\mathbf{uu}}^{-1} k_{\mathbf{u}}^{-1} - k^{-1} k_{\mathbf{u}}^{-1}) \end{array} \middle| \begin{array}{c} 0 \\ 0 \end{array} \right) \\
&+ \frac{\epsilon}{8} \left(\begin{array}{c|c} 0 & 0 \\ 0_{M-1 \times M-1} & \vdots \\ 0 & 0 \\ \hline 0 & \dots & 0 & (k^{-1})^2 - 2(k_{\mathbf{u}}^{-1})^T k_{\mathbf{u}}^{-1} \\ 0 & \dots & 0 & -((k^{-1})^2 - 2(k_{\mathbf{u}}^{-1})^T k_{\mathbf{u}}^{-1}) \end{array} \middle| \begin{array}{c} 0 \\ \vdots \\ 0 \\ -((k^{-1})^2 - 2(k_{\mathbf{u}}^{-1})^T k_{\mathbf{u}}^{-1}) \\ (k^{-1})^2 - 2(k_{\mathbf{u}}^{-1})^T k_{\mathbf{u}}^{-1} \end{array} \right) + \mathcal{O}(\epsilon^2)
\end{aligned}$$

For the matrix $(K_{\mathbf{uu}} + \epsilon I)^{-1}$ we find:

$$(K_{\mathbf{uu}} + \epsilon)^{-1} = K_{\mathbf{uu}}^{-1} - \epsilon K_{\mathbf{uu}}^{-2} + \mathcal{O}(\epsilon^2)$$

When we now multiply out the product $K_{\text{fu}}^+(K_{\text{uu}}^+ + \epsilon I)^{-1}K_{\text{uf}}^+$, we note that K_{uf}^+ will have a duplicate row and K_{fu}^+ will have a duplicate column. Due to this, all terms that have the submatrix $0_{M-1 \times M-1}$ in their upper left hand corner cancel. This includes the term that contains the exploding (in the limit $\epsilon \rightarrow 0$) inverse jitter ϵ^{-1} , and we are left with:

$$\begin{aligned} Q_{\text{ff}}^+ &= K_{\text{fu}}^+(K_{\text{uu}}^+ + \epsilon I)^{-1}K_{\text{uf}}^+ = K_{\text{fu}}K_{\text{uu}}^{-1}K_{\text{uf}} - \epsilon K_{\text{fu}}K_{\text{uu}}^{-2}K_{\text{uf}} + \frac{\epsilon}{2}K_{\text{fu}}k_{\text{u}}^{-1}(k_{\text{u}}^{-1})^\top K_{\text{uf}} + \mathcal{O}(\epsilon^2) \\ &= Q_{\text{ff}} - \epsilon K_{\text{fu}}K_{\text{uu}}^{-2}K_{\text{uf}} + \frac{\epsilon}{2}K_{\text{fu}}k_{\text{u}}^{-1}(k_{\text{u}}^{-1})^\top K_{\text{uf}} + \mathcal{O}(\epsilon^2) \end{aligned}$$

Such that the correction to the original approximate covariance matrix is given by:

$$Q_{\text{ff}}^+ - Q_{\text{ff}} = \frac{\epsilon}{2}K_{\text{fu}}k_{\text{u}}^{-1}(k_{\text{u}}^{-1})^\top K_{\text{uf}} + \mathcal{O}(\epsilon^2) \quad (\text{A.18})$$

We can now take the limit $\epsilon \rightarrow 0$ as all the "infinities" have cancelled above. For finite jitter, the correction term is again given by a rank-1 update to first order in ϵ .