# Scalable Variational Gaussian Process Classification

**James Hensman**
University of Sheffield

**Alex Matthews**
University of Cambridge

**Zoubin Ghahramani**
University of Cambridge

## Abstract

Gaussian process classification is a popular method with a number of appealing properties. We show how to scale the model within a variational inducing point framework, outperforming the state of the art on benchmark datasets. Importantly, the variational formulation can be exploited to allow classification in problems with millions of data points, as we demonstrate in experiments.

## 1 Introduction

Gaussian processes (GPs) provide priors over functions that can be used for many machine learning tasks. In the regression setting, when the likelihood is Gaussian, inference can be performed in closed-form using linear algebra. When the likelihood is non-Gaussian, such as in GP classification, the posterior and marginal likelihood must be approximated. Kuss and Rasmussen [2005] and Nickisch and Rasmussen [2008] provide excellent comparisons of several approximate inference methods for GP classification. More recently, Opper and Archambeau [2009] and Khan et al. [2012] considered algorithmic improvements to variational approximations in non-conjugate GP models.

The computational cost of inference in GPs is $\mathcal{O}(N^3)$ in general, where $N$ is the number of data. In the regression setting, there has been much interest in low-rank or *sparse* approaches to reduce this computational complexity: Quiñonero-Candela and Rasmussen [2005] provides a review. Many of these approximation schemes rely on the use of a series of *inducing points*, which can be difficult to select. Titsias [2009] suggests a variational approach (see section 2) which provides an objective function for optimizing these points. This variational idea was extended by Hensman et al. [2013], who showed how the variational objective could be reformulated with additional parameters to enable stochastic optimization, which allows GPs to be fitted to millions of data.

Despite much interest in approximate GP classification *and* sparse approximations, there has been little overlap of the two. The approximate inference schemes which deal with the non-conjugacy of the likelihood generally scale with $\mathcal{O}(N^3)$, as they require factorization of the covariance matrix. Two approaches which have addressed these issues simultaneously are the IVM [Lawrence et al., 2003] and Generalized FITC [Naish-Guzman and Holden, 2007] which both have shortcomings as we shall discuss.

It is tempting to think that sparse GP classification is simply a case of combining a low-rank approximation to the covariance with one's preferred non-conjugate approximation, but as we shall show this does not necessarily lead to an effective method: it can be very difficult to place the inducing input points, and scalability of the method is usually restricted by the complexity of matrix-matrix multiplications.

For these reasons, there is a strong case for a non-conjugate sparse GP scheme which provides a variational bound on the marginal likelihood, combining the scalability of the stochastic optimization approach with the ability to optimize the positions of the inducing inputs. Furthermore, a variational approach would allow for integration of the approximation within other GP models such as GP regression networks [Wilson et al., 2012], latent variable models [Lawrence, 2004] and deep GPs [Damianou and Lawrence, 2013], as the variational objective could be optimized as part of such a model without fear of overfitting.

The rest of the paper is arranged as follows. In section 2, we briefly cover some background material and existing work. In section 3, we show how variational approximations to the covariance matrix can be used post-hoc to provide variational bounds with non-Gaussian likelihoods. These approaches are not entirely satisfactory, and so in section 4 we provide a variational approach which does not first approximate the covariance matrix, but delivers a variational bound

directly. In section 5 we compare our proposals with the state of the art, as well as demonstrating empirically that our preferred method is applicable to very large datasets through stochastic variational optimization. Section 6 concludes.

## 2 Background

**Gaussian Process Classification** Gaussian process priors provide rich nonparametric models of functions. To perform classification with this prior, the process is 'squashed' through a sigmoidal inverse-link function, and a Bernoulli likelihood conditions the data on the transformed function values. See Rasmussen and Williams [2006] for a review.

We denote the binary class observations as $\mathbf{y} = \{y_n\}_{n=1}^N$, and then collect the input data into a design matrix $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$. We evaluate the covariance function at all pairs of input vectors to build the covariance matrix $\mathbf{K}_{nn}$ in the usual way, and arrive at a prior for the values of the GP function at the input points: $p(\mathbf{f}) = \mathcal{N}(\mathbf{f} \,|\, \mathbf{0}, \mathbf{K}_{nn})$.

We denote the probit inverse link function as $\phi(x) = \int_{-\infty}^{x} \mathcal{N}(a \,|\, 0,\, 1)\mathrm{d}a$ and the Bernoulli distribution $\mathcal{B}(y_n \,|\, \phi(f_n)) = \phi(f_n)^{y_n}(1-\phi(f_n))^{1-y_n}$. The joint distribution of data and latent variables becomes

$$p(\mathbf{y}, \mathbf{f}) = \prod_{n=1}^{N} \mathcal{B}(y_n \,|\, \phi(f_n))\, \mathcal{N}(\mathbf{f} \,|\, \mathbf{0}, \mathbf{K}_{nn}) \ . \quad (1)$$

The main object of interest is the posterior over function values $p(\mathbf{f} \,|\, \mathbf{y})$, which must be approximated. We also require an approximation to the marginal likelihood $p(\mathbf{y})$ in order to optimize (or marginalize) parameters of the covariance function. An assortment of approximation schemes have been proposed (see Nickisch and Rasmussen [2008] for a comparison), but they all require $\mathcal{O}(N^3)$ computation.

**Sparse Gaussian Processes for Regression** The computational complexity of any Gaussian process method scales with $\mathcal{O}(N^3)$ because of the need to invert the covariance matrix $\mathbf{K}$. To reduce the computational complexity, many approximation schemes have been proposed, though most focus on regression tasks, see Quiñonero-Candela and Rasmussen [2005] for a review. Here we focus on inducing point methods [Snelson and Ghahramani, 2005], where the latent variables are augmented with additional input-output pairs $\mathbf{Z}, \mathbf{u}$, known as 'inducing inputs' and 'inducing variables'.

The random variables $\mathbf{u}$ are points on the function in exactly the same way as $\mathbf{f}$, and so the joint distribution can be written

$$p(\mathbf{f}, \mathbf{u}) = \mathcal{N}\left( \begin{bmatrix} \mathbf{f} \\ \mathbf{u} \end{bmatrix} \Big| \mathbf{0}, \begin{bmatrix} \mathbf{K}_{nn} & \mathbf{K}_{nm} \\ \mathbf{K}_{nm}^\top & \mathbf{K}_{mm} \end{bmatrix} \right) \quad (2)$$

where $\mathbf{K}_{mm}$ is formed by evaluating the covariance function at all pairs of inducing inputs points $\mathbf{z}_m, \mathbf{z}_{m'}$, and $\mathbf{K}_{nm}$ is formed by evaluating the covariance function across the data input points and inducing inputs points similarly. Using the properties of a multivariate normal distribution, the joint can be re-written as

$$\begin{aligned} p(\mathbf{f}, \mathbf{u}) &= p(\mathbf{f} \,|\, \mathbf{u})p(\mathbf{u}) \quad (3) \\ &= \mathcal{N}(\mathbf{f} \,|\, \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{u}, \mathbf{K}_{nn} - \mathbf{Q}_{nn})\mathcal{N}(\mathbf{u} \,|\, \mathbf{0},\, \mathbf{K}_{mm}) \end{aligned}$$

with $\mathbf{Q}_{nn} = \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{nm}^\top$. The joint distribution now takes the form

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}) = p(\mathbf{y} \,|\, \mathbf{f})p(\mathbf{f} \,|\, \mathbf{u})p(\mathbf{u}) \ . \quad (4)$$

To obtain computationally efficient inference, integration over $\mathbf{f}$ is approximated. To obtain the popular FITC method (in the case of Gaussian likelihood), a factorization is enforced: $p(\mathbf{y} \,|\, \mathbf{u}) \approx \prod_n p(y_n \,|\, \mathbf{u})$. To get a variational approximation, the following inequality is used

$$\log p(\mathbf{y} \,|\, \mathbf{u}) \geq \mathbb{E}_{p(\mathbf{f} \,|\, \mathbf{u})}\left[\log p(\mathbf{y} \,|\, \mathbf{f})\right] \triangleq \log \tilde{p}(\mathbf{y} \,|\, \mathbf{u}) \ . \quad (5)$$

Substituting this bound on the conditional into the standard expression $p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{u}$ gives a tractable bound on the marginal likelihood for the Gaussian case [Titsias, 2009]:

$$\begin{aligned} \log p(\mathbf{y}) \geq &\log \mathcal{N}(\mathbf{y} \,|\, \mathbf{0}, \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{nm}^\top + \sigma^2\mathbf{I}) \\ &- \frac{1}{2\sigma^2}\mathrm{tr}(\mathbf{K}_{nn} - \mathbf{Q}_{nn}) \ , \end{aligned} \quad (6)$$

where $\sigma^2$ is the variance of the Gaussian likelihood term. This bound on the marginal likelihood can then be used as an objective function in optimizing the covariance function parameters as well as the inducing input points $\mathbf{Z}$. The bound becomes tight when the inducing points are the data points: $\mathbf{Z} = \mathbf{X}$ so $\mathbf{K}_{nm} = \mathbf{K}_{mm} = \mathbf{K}_{nn}$ and (6) becomes equal the true marginal likelihood $\log \mathcal{N}(\mathbf{y} \,|\, \mathbf{0},\, \mathbf{K}_{nn} + \sigma^2\mathbf{I})$.

Computing this bound (6) and its derivatives costs $\mathcal{O}(NM^2)$. A more computationally scalable bound can be achieved by introducing additional variational parameters [Hensman et al., 2013]. Noting that (6) implies an approximate posterior $\tilde{p}(\mathbf{u} \,|\, \mathbf{y})$, we introduce a variational distribution $q(\mathbf{u}) = \mathcal{N}(\mathbf{u} \,|\, \mathbf{m}, \mathbf{S})$ to approximate this distribution, and applying a standard variational bound, obtain:

$$\begin{aligned} \log p(\mathbf{y}) \geq &\log \mathcal{N}(\mathbf{y} \,|\, \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{m},\, \sigma^2\mathbf{I}) \\ &- \frac{1}{2\sigma^2}\mathrm{tr}(\mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{S}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn}) \quad (7) \\ &- \frac{1}{2\sigma^2}\mathrm{tr}(\mathbf{K}_{nn} - \mathbf{Q}_{nn}) - \mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})] \ . \end{aligned}$$

This bound has a unique optimum in terms of the variational parameters $\mathbf{m}, \mathbf{S}$, at which point it is tight to the original sparse GP bound (6). The advantage of the representation in (7) is that it can be optimized in a stochastic [Hensman et al., 2013] or distributed [Dai et al., 2014, Gal et al., 2015] fashion.

Of course for the Bernoulli likelihood, the required integrals for (6) and (7) are not tractable, but we will build on them both in subsequent sections to build sparse GP classifiers.

**Related work** The informative vector machine (IVM) [Lawrence et al., 2003] is the first work to approach sparse GP classification to our knowledge. The idea is to combine assumed density filtering with a selection heuristic to pick points from the data $\mathbf{X}$ to act as inducing points $\mathbf{Z}$: the inducing variables $\mathbf{u}$ are then a subset of the latent function variables $\mathbf{f}$.

The IVM offers superior performance to support vector machines [Lawrence et al., 2003], along with a probabilistic interpretation. However we might expect better performance by relaxing the condition that the inducing points be a sub-set of the data, as is the case for regression [Quiñonero-Candela and Rasmussen, 2005].

Subsequent work on sparse GP classification [Naish-Guzman and Holden, 2007] removed the restriction of selecting $\mathbf{Z}$ to be a subset of the data $\mathbf{X}$, and ostensibly improved over the assumed density filtering scheme by using expectation propagation (EP) for inference.

Naish-Guzman and Holden [2007] noted that when using the FITC approximation for a Gaussian likelihood, the equivalent prior (see also Quiñonero-Candela and Rasmussen [2005]) is

$$p(\mathbf{f}) \approx \mathcal{N}(\mathbf{f} \,|\, \mathbf{0},\, \mathbf{Q}_{nn} + \mathrm{diag}(\mathbf{K}_{nn} - \mathbf{Q}_{nn})) \ . \qquad (8)$$

The Generalized FITC method combines this approximate prior with a Bernoulli likelihood and uses EP to approximate the posterior. The form of the prior means that the linear algebra within the EP updates is simplified, and a round of updates costs $\mathcal{O}(NM^2)$. EP is nested inside an optimization loop, where the covariance hyper-parameters and inducing inputs are optimized against the EP approximation to the marginal likelihood. Computing the marginal likelihood approximation and gradients costs $\mathcal{O}(NM^2)$.

The generalized FITC method works well in practise, often finding solutions which are as good as or better than the IVM, with fewer inducing inputs points [Naish-Guzman and Holden, 2007].

**Discussion** Despite performing significantly better than the IVM, GFITC does not satisfy our requirements for a scalable GP classifier. There is no clear way to distribute computation or use stochastic optimization in GFITC: we find that it is limited to a few thousand data. Further, the positions of the inducing input points $\mathbf{Z}$ can be optimized against the approximation to the marginal likelihood, but there is no guarantee that this will provide a good solution: indeed, our experiments in section 5 show that this can lead to strange pathologies.

To obtain the ability to place inducing inputs as Titsias [2009] and to scale as Hensman et al. [2013], we desire a bound on the marginal likelihood against which to optimize $\mathbf{Z}$. In the next section, we attempt to build such a variational approximation in the same fashion as FITC, by first using existing variational methods to approximate the covariance, and then using further variational approximate methods to deal with non-conjugacy.

## 3 Two stage approaches

A straightforward approach to building sparse GP classifiers is to separate the low-rank approximation to the covariance from the non-Gaussian likelihood. In other words, simply treat the approximation to the covariance matrix as the prior, and then select an approximate inference algorithm (e.g. from one of those compared by Nickisch and Rasmussen [2008]), and then proceed with approximate inference, exploiting the form of the approximate covariance where possible for computation saving.

This is how the generalized FITC approximation was derived. However, as described above we aim to construct variational approximations.

It's possible to construct a variational approach in this mould by using the fact that the probit likelihood can be written as a convolution of a unit Gaussian and a step function. Without modifying our original model, we can introduce a set of additional latent variables $\mathbf{g}$ which relate to the original latent variables $\mathbf{f}$ through a unit variance isotropic Gaussian:

$$p(\mathbf{y}, \mathbf{f}, \mathbf{g}) = \prod_{n=1}^{N} \mathcal{B}(y_n | \theta(g_n))\, \mathcal{N}(\mathbf{g} | \mathbf{f}, \mathbf{I})\, \mathcal{N}(\mathbf{f} | \mathbf{0}, \mathbf{K}) \quad (9)$$

where $\theta$ is a step-function inverse-link. The original model (1) is recovered by marginalization of the additional latent vector: $\int \prod_{n=1}^{N} \mathcal{B}(y_n \,|\, \theta(g_n))\, \mathcal{N}(\mathbf{g} \,|\, \mathbf{f}, \mathbf{I}) \mathrm{d}\mathbf{g} = \prod_{n=1}^{N} \mathcal{B}(y_n \,|\, \phi(f_n))$.

We can now proceed by using a variational sparse GP bound on $\mathbf{g}$, followed by a further variational approximation to deal with the non-Gaussian likelihood.

### 3.1 Sparse mean field approach

Encouraged by the success of a (non-sparse) factorizing approximation made by [Hensman et al., 2014], we couple a variational bound on $p(\mathbf{g})$ with a mean-field approximation. Substituting the variational bound for a Gaussian sparse GP (6) with our augmented model (9) (where $\mathbf{g}$ in (9) replaces $\mathbf{y}$ in (6)), we arrive at a bound on the joint distribution:

$$p(\mathbf{y},\,\mathbf{g}) \geq \prod_{n=1}^{N} \mathcal{B}(y_n \,|\, \theta(g_n))\, \mathcal{N}(\mathbf{g} \,|\, \mathbf{0}, \mathbf{Q}_{nn} + \mathbf{I}) \\ \exp\{-\tfrac{1}{2}\mathrm{tr}(\mathbf{K}_{nn} - \mathbf{Q}_{nn})\} \ . \tag{10}$$

Assuming a factorizing distribution $q(\mathbf{g}) = \prod_n q(g_n)$, we obtain a lower bound on the marginal likelihood in the usual variational way, and the optimal form of the approximating distribution is a truncated Gaussian

$$q^\star(g_n) = \mathcal{B}(y_n \,|\, \theta(g_n))\mathcal{N}(g_n \,|\, a_n, \tilde{\sigma}_n^2)/\gamma_n \ , \tag{11}$$

where $\tilde{\sigma}_n^{-2}$ is given by the $n^{\text{th}}$ diagonal element of $[\mathbf{Q}_{nn} + \mathbf{I}]^{-1}$, $\gamma_n$ are the required normalizers and $a_n$ are free variational parameters. Some cancellation in the bound leads to a tractable expression:

$$\begin{aligned} \log p(\mathbf{y}) \geq & \sum_{n=1}^{N} \log \gamma_n - \tfrac{1}{2}\log|\mathbf{Q}_{nn} + \mathbf{I}| \\ & - \tfrac{1}{2}\mathrm{tr}([\mathbf{Q}_{nn} + \mathbf{I}]^{-1}\mathbb{E}_{q(\mathbf{g})}[\mathbf{g}\mathbf{g}^\top]) \\ & + \tfrac{1}{2}\sum_{n=1}^{N}\Big\{\log\tilde{\sigma}_n^2 + \frac{\mathbb{E}_{q(g_n)}[(a_n - g_n)^2]}{\tilde{\sigma}_n^2}\Big\} \\ & - \tfrac{1}{2}\mathrm{tr}(\mathbf{K}_{nn} - \mathbf{Q}_{nn}). \end{aligned} \tag{12}$$

All the components of this bound can be computed in maximum $\mathcal{O}(NM^2)$ time. The expectations under the factorizing variational distribution (and their gradients) are available in closed form.

Approximate inference can now proceed by optimizing this bound with respect to the variational parameters $\mathbf{a} = \{a_n\}_{n=1}^N$, alongside the hyper-parameters of the covariance function and the inducing points $\mathbf{Z}$. Empirically, we find it useful to optimize the variational parameters on an 'inner loop', which costs $\mathcal{O}(NM)$ per iteration. Computing the relevant gradients outside this loop costs $\mathcal{O}(NM^2)$.

**Predictions** To make a prediction for a new latent function value $g_\star$ at a test input point $\mathbf{x}_\star$, we would like to marginalize across the variational distribution:

$$p(g_\star \,|\, \mathbf{y}) \approx \int p(g_\star \,|\, \mathbf{g}) q(\mathbf{g}) \mathrm{d}\mathbf{g} \ . \tag{13}$$

Since this is in general intractable, we approximate it by Monte Carlo. Since the approximate posterior $q$ is a factorized series of truncated normal distributions, it is straight-forward to sample from. Prediction of a single test point costs $\mathcal{O}(N)$.

Alternatively, we can employ a Gaussian approximation to the posterior as suggested by Nickisch and Rasmussen [2008]. In our sparse formulation, this results in $p(\mathbf{g} \,|\, \mathbf{y}) \approx \mathcal{N}(\mathbf{g} \,|\, \mathbf{\Sigma}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn}\mathbb{E}_{q(\mathbf{g})}[\mathbf{g}], \mathbf{\Sigma})$, with $\mathbf{\Sigma} = \mathbf{K}_{mm} - \mathbf{K}_{mn}[\mathbf{Q}_{nn} + \mathbf{I}]^{-1}\mathbf{K}_{nm}$. Substituting in to (13) results in a computational cost of $\mathcal{O}(M^2)$ to predict a single test point.

### 3.2 A more scalable method?

The mean-field method in section 3.1 is somewhat unsatisfactory since the number of variational parameters scales with $N$. The variational parameters are also dependent on each other, and so application of stochastic optimization or distributed computing is difficult. For the Gaussian likelihood case, Hensman et al. [2013] proposes a bound on $\log p(\mathbf{y})$ (7) which can be optimized effectively with $\mathcal{O}(M^2)$ parameters. Perhaps it is possible to obtain a scalable algorithm by substituting this bound for $p(\mathbf{g})$ as in the above?

Substituting (7) into (9) (again replacing $\mathbf{y}$ with $\mathbf{g}$) results in a tractable integral to obtain a bound on the marginal likelihood, as Hensman et al. [2013] points out. The result is

$$\begin{aligned} \log p(\mathbf{y}) &= \log \int p(\mathbf{y} \,|\, \mathbf{g}) p(\mathbf{g}) \mathrm{d}\mathbf{g} \\ &\geq \prod_n \mathcal{B}(y_n \,|\, \phi(\mathbf{k}_n^\top \mathbf{K}_{mm}^{-1}\mathbf{m})) \\ &\quad - \tfrac{1}{2}\mathrm{tr}(\mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{S}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn}) \\ &\quad - \tfrac{1}{2}\mathrm{tr}(\mathbf{K}_{nn} - \mathbf{Q}_{nn}) - \mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})] \ , \end{aligned}$$

where $\mathbf{k}_n^\top$ is the $n^{\text{th}}$ row of $\mathbf{K}_{nm}$. This bound can be optimized in a stochastic or distributed way [Tolvanen, 2014], but has been found to be less effective than might be expected (Owen Thomas, personal communication). To understand why, consider the case where the inducing points are set to the data points $\mathbf{Z} = \mathbf{X}$, so that $\mathbf{u} = \mathbf{f}$. The bound reduces to

$$\begin{aligned} \log p(\mathbf{y}) \geq & \prod_n \mathcal{B}(y_n \,|\, \phi(\mathbf{m}_n)) \\ & - \tfrac{1}{2}\mathrm{tr}(\mathbf{S}) - \mathrm{KL}[q(\mathbf{f})||p(\mathbf{f})] \ . \end{aligned} \tag{14}$$

and the optimum occurs where $\mathbf{S}^{-1} = \mathbf{K}_{nn}^{-1} + \mathbf{I}$, and $\mathbf{m}$ is the maximum *a posteriori* (MAP) point.

This approximation is reminiscent of the Laplace approximation, which also places the mean of the posterior at the MAP point, and approximates the covariance with $\mathbf{S}^{-1} = \mathbf{K}_{nn}^{-1} + \mathbf{W}$, where $\mathbf{W}$ is the Hessian of the log likelihood evaluated at the MAP point. The Laplace approximation is known to be relatively

ineffective for classification [Nickisch and Rasmussen, 2008], so it is no surprise that this variational approximation should be ineffective. From here we abandon this bound: the next section sees the construction of a variational bound which is scalable and (empirically) effective.

## 4 A single variational bound

Here we obtain a bound on the marginal likelihood without introducing the additional latent variables as above, and without making factorizing assumptions. We first return to the bound (5) on the conditional used to construct the variational bounds for the Gaussian case:

$$\log p(\mathbf{y} \,|\, \mathbf{u}) \geq \mathbb{E}_{p(\mathbf{f} \,|\, \mathbf{u})} \left[\log p(\mathbf{y} \,|\, \mathbf{f})\right] \tag{15}$$

which is in general intractable for the non-conjugate case. We nevertheless persist, recalling the standard variational equation

$$\log p(\mathbf{y}) \geq \mathbb{E}_{q(\mathbf{u})} \left[\log p(\mathbf{y} \,|\, \mathbf{u})\right] - \mathrm{KL}\left[q(\mathbf{u}) || p(\mathbf{u})\right] \ . \tag{16}$$

Substituting (15) into (16) results in a (further) bound on the marginal likelihood:

$$\begin{aligned}\log p(\mathbf{y}) &\geq \mathbb{E}_{q(\mathbf{u})} \left[\log p(\mathbf{y} \,|\, \mathbf{u})\right] - \mathrm{KL}\left[q(\mathbf{u}) || p(\mathbf{u})\right] \\ &\geq \mathbb{E}_{q(\mathbf{u})} \left[\mathbb{E}_{p(\mathbf{f}|\mathbf{u})} \left[\log p(\mathbf{y}|\mathbf{f})\right]\right] - \mathrm{KL}\left[q(\mathbf{u}) || p(\mathbf{u})\right] \\ &= \mathbb{E}_{q(\mathbf{f})} \left[\log p(\mathbf{y} \,|\, \mathbf{f})\right] - \mathrm{KL}\left[q(\mathbf{u}) || p(\mathbf{u})\right] \end{aligned} \tag{17}$$

where we have defined:

$$q(\mathbf{f}) := \int p(\mathbf{f} \,|\, \mathbf{u}) q(\mathbf{u}) \mathrm{d}\mathbf{u} \ . \tag{18}$$

Consider the case where $q(\mathbf{u}) = \mathcal{N}(\mathbf{u} \,|\, \mathbf{m}, \mathbf{S})$. This gives the following functional form for $q(\mathbf{f})$:

$$q(\mathbf{f}) = \mathcal{N}(\mathbf{f} \,|\, \mathbf{A}\mathbf{m}, \mathbf{K}_{nn} + \mathbf{A}(\mathbf{S} - \mathbf{K}_{mm})\mathbf{A}^\top) \tag{19}$$

with $\mathbf{A} = \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}$.

Since in the classification case the likelihood factors as

$$p(\mathbf{y} \,|\, \mathbf{f}) = \prod_{i=1}^{N} p(y_i \,|\, f_i) \ , \tag{20}$$

we only require the marginals of $q(\mathbf{f})$ in order to compute the expectations in (17). We are left with some one-dimensional integrals of the log-likelihood, which can be computed by e.g. Gauss-Hermite quadrature:

$$\log p(\mathbf{y}) \geq \sum_{n=1}^{N} \mathbb{E}_{q(f_n)} \left[\log p(y_n \,|\, f_n)\right] - \mathrm{KL}\left[q(\mathbf{u}) || p(\mathbf{u})\right] \ . \tag{21}$$

Our algorithm then consists of maximizing the parameters of $q(\mathbf{u})$ with respect to this bound on the marginal likelihood using gradient based optimization. To maintain positive-definiteness of $\mathbf{S}$, we represent it using a lower triangular form $\mathbf{S} = \mathbf{L}\mathbf{L}^\top$, which allows us to perform unconstrained optimization.

**Computations and Scalability** Computing the KL divergence of the bound (21) requires $\mathcal{O}(M^3)$ computations. Since we expect the number of required inducing points $M$ to be much smaller than the number of data $N$, most of the work will be in computing the expected likelihood terms. To compute the derivatives of these, we use the Gaussian identities made familiar to us by Opper and Archambeau [2009]:

$$\begin{aligned}\frac{\partial}{\partial \mu} \mathbb{E}_{\mathcal{N}(x|\mu,\sigma^2)}\big[f(x)\big] &= \mathbb{E}_{\mathcal{N}(x|\mu,\sigma^2)}\big[\frac{\partial}{\partial x} f(x)\big] \\ \frac{\partial}{\partial \sigma^2} \mathbb{E}_{\mathcal{N}(x|\mu,\sigma^2)}\big[f(x)\big] &= \tfrac{1}{2}\mathbb{E}_{\mathcal{N}(x|\mu,\sigma^2)}\big[\frac{\partial^2}{\partial x^2} f(x)\big] \ . \end{aligned} \tag{22}$$

We can make use of these by substituting $f$ for $\log p(y_n \,|\, f_n)$ and $\mu, \sigma^2$ for the marginals of $q(\mathbf{f})$ in (21). These derivatives also have to be computed by quadrature methods, after which derivatives with respect to $\mathbf{m}, \mathbf{L}, \mathbf{Z}$ and any covariance function parameters requires the application of straight-forward algebra.

We also have the option to optimize the objective in a distributed fashion due to the ease of parallelizing the simple sum over $N$, or in a stochastic fashion by selecting mini-batches of the data at random as we shall show in the following.

**Predictions** Our approximate posterior is given as $q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f} \,|\, \mathbf{u})q(\mathbf{u})$. To make predictions at a set of test points $\mathbf{X}_\star$ for the new latent function values $\mathbf{f}_\star$, we substitute our approximate posterior into the standard probabilistic rule:

$$p(\mathbf{f}_\star \,|\, \mathbf{y}) = \int p(\mathbf{f}_\star \,|\, \mathbf{f}, \mathbf{u}) p(\mathbf{f}, \mathbf{u} \,|\, \mathbf{y}) \mathrm{d}\mathbf{f}\mathrm{d}\mathbf{u} \tag{23}$$

$$\approx \int p(\mathbf{f}_\star \,|\, \mathbf{f}, \mathbf{u}) p(\mathbf{f} \,|\, \mathbf{u}) q(\mathbf{u}) \mathrm{d}\mathbf{f}\mathrm{d}\mathbf{u} \tag{24}$$

$$= \int p(\mathbf{f}_\star \,|\, \mathbf{u}) q(\mathbf{u}) \mathrm{d}\mathbf{u} \tag{25}$$

where the last line occurs due to the consistency rules of the GP. The integral is tractable similarly to (19), and we can compute the mean and variance of a test-latent $f_\star$ in $\mathcal{O}(M^2)$, from which the distribution of the test label $y_\star$ is easily computed.

**Limiting cases** To relate this method to existing work, consider two limiting cases of this bound. First, when the inducing variables are equal to the data points $\mathbf{Z} = \mathbf{X}$, and second, where the likelihood is replaced with Gaussian noise.

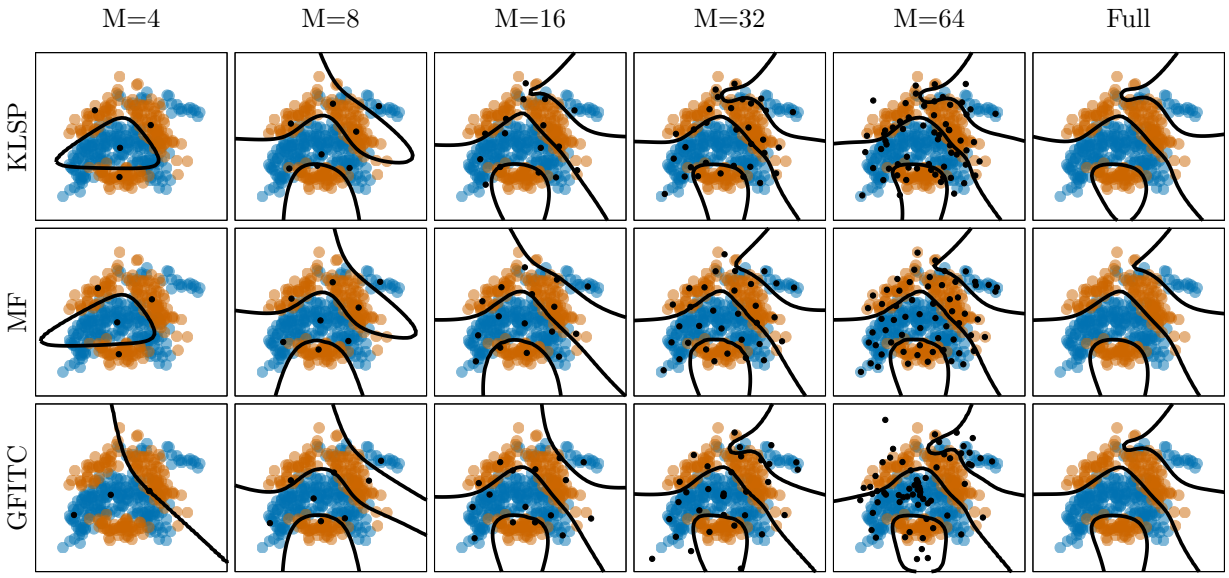


Figure 1: The effect of increasing the number of inducing points for the *banana* dataset. Rows represent the KL method, the mean field method and Generalized FITC, whilst columns show increasing numbers of inducing points. In each pane, the colored points represent training data, the inducing inputs are black dots and the decision boundaries are black lines. The rightmost column shows the result of the equivalent non-sparse methods

When $\mathbf{Z} = \mathbf{X}$, the approximate posterior in equation (18) reduces to $q(\mathbf{f}) = \mathcal{N}(\mathbf{f} \,|\, \mathbf{m}, \mathbf{S})$. In this special case the number of parameters required to represent the covariance can be reduced to $2N$ [Opper and Archambeau, 2009], and we have recovered the full-Gaussian approximation (the 'KL' method described by Nickisch and Rasmussen [2008]).

If the likelihood were Gaussian, the expectations in equation (21) would be computable in closed form, and after a little re-arranging the result is as [Hensman et al., 2013], equation (7). In this case the bound has a unique solution for $\mathbf{m}$ and $\mathbf{S}$, which recovers the variational bound of Titsias [2009], equation (6).

In the case where $\mathbf{Z} = \mathbf{X}$ and the likelihood is Gaussian, exact inference is recovered.

## 5 Experiments

We have proposed two variational approximations for GP classification. The first in section 3 comprises a mean-field approximation after making a variational approximation to the prior over an augmented latent vector. The second in section 4 proposes to minimize the KL divergence using a Gaussian approximation at a set of inducing points. We henceforth refer to these as the MF (mean-field) and KL methods respectively.

**Increasing the number of inducing points** To compare the methods with the state-of-the-art Generalized FITC method, we first turn to the two-dimensional Banana dataset. For all three methods, we initialized the inducing points using k-means clustering. For the generalized FITC method we used the implementation provided by Rasmussen and Nickisch [2010]. For all the methods we used the L-BFGS-B optimizer [Zhu et al., 1997].

With the expectation that increasing the number of inducing points should improve all three methods, we applied 4 to 64 inducing points, as shown in Figure 1. The KL method pulls the inducing points positions toward the decision boundary, and provides a near-optimal solution with 16 inducing points. The MF method is less able to adapt the inducing input positions, but provides good solutions at the same number of inducing points. The Generalized FITC method appears to pull inducing points towards the decision boundary, but is unable to make good use of 64 inducing points, moving some to the decision boundary and some toward the origin.

**Numerical comparison** We compared the performance of the classifiers for a number of commonly used classification data sets. We took ten folds of the data and report the median hold out negative log probability and 2-$\sigma$ confidence intervals. For comparison we used the EP FITC implementation from the GPML toolbox [Rasmussen and Nickisch, 2010] which is generally considered to be amongst the best implementations of this algorithm. For the KL and sparse KL methods we found that the optimization behaviour was improved by freezing the kernel hyper parameters

Table 1: comparison of the performance of our proposed methods and Generalized FITC on benchmark datasets.

| Dataset | KL | EP | EPFitc M=8 | EPFitc M=3.0% | KLSp M=8 | KLSp M=3.0% | MFSp M=8 | MFSp M=3.0% |
|---|---|---|---|---|---|---|---|---|
| thyroid | $.11 \pm .05$ | $.11 \pm .05$ | $.15 \pm .04$ | $.13 \pm .04$ | $.13 \pm .06$ | $.09 \pm .05$ | $.22 \pm .16$ | $.15 \pm .14$ |
| heart | $.46 \pm .14$ | $.43 \pm .11$ | $.42 \pm .08$ | $.44 \pm .08$ | $.42 \pm .11$ | $.47 \pm .18$ | $.41 \pm .15$ | $.43 \pm .19$ |
| twonorm | $.17 \pm .43$ | $.08 \pm .01$ | Large | $.08 \pm .01$ | $.08 \pm .01$ | $.09 \pm .02$ | Large | Large |
| ringnorm | $.21 \pm .22$ | $.18 \pm .02$ | $.34 \pm .01$ | $.20 \pm .01$ | $.41 \pm .09$ | $.15 \pm .03$ | $.46 \pm .00$ | Large |
| german | $.51 \pm .09$ | $.48 \pm .06$ | $.49 \pm .04$ | $.50 \pm .04$ | $.49 \pm .05$ | $.51 \pm .08$ | $.52 \pm .06$ | $.51 \pm .09$ |
| waveform | $.23 \pm .09$ | $.23 \pm .02$ | $.22 \pm .01$ | $.22 \pm .01$ | $.23 \pm .01$ | $.25 \pm .04$ | $.28 \pm .01$ | Large |
| cancer | $.56 \pm .09$ | $.55 \pm .08$ | $.58 \pm .06$ | $.56 \pm .07$ | $.56 \pm .07$ | $.58 \pm .13$ | $.58 \pm .11$ | $.59 \pm .11$ |
| flare solar | $.59 \pm .02$ | $.60 \pm .02$ | $.57 \pm .02$ | $.57 \pm .02$ | $.59 \pm .02$ | $.58 \pm .02$ | $.64 \pm .05$ | $.60 \pm .04$ |
| diabetes | $.48 \pm .04$ | $.48 \pm .03$ | $.50 \pm .02$ | $.51 \pm .02$ | $.47 \pm .02$ | $.51 \pm .04$ | Large | $.50 \pm .04$ |

at the beginning of optimization and then unfreezing them once a reasonable set of variational parameters has been attained. Table 1 shows the result of the experiments. In the case where a classifier gives a extremely confident wrong prediction for one or more test points one can obtain a numerically high negative log probability. These cases are denoted as 'large'.

The table shows that mean field based methods often give over confident predictions. The results show similar performance between the sparse KL and FITC methods. The confidence intervals of the two methods either overlap or sparse KL is better. As we shall see, Sparse KL runs faster in the timed experiments that follow and can be run on very large datasets using stochastic gradient descent.

**Time-performance trade-off** Since all the algorithms used perform optimization there is a trade-off for amount of time spent on optimization against the classification performance achieved. The whole trade-off curve can be used to characterize the efficiency of a given algorithm.

Naish-Guzman and Holden [2007] consider the *image* dataset amongst their benchmarks. Of the datasets they consider it is one of the most demanding in terms of the number of inducing points that are required. We ran timed experiments on this dataset recording the wall clock time after each function call and computing the hold out negative log probability of the associated parameters at each step.

We profiled the MFSp and KLSp algorithms for a variety of inducing point numbers. Although GPML is implemented in MATLAB rather than Python both have access to fast numerical linear algebra libraries. Optimization for the sparse KL and FITC methods was carried out using the LBFGS-B Zhu et al. [1997] algorithm. The mean field optimization surface is challenging numerically and we found that performance was improved by using the scaled conjugate gradients algorithm. We found no significant qualitative difference between the wall clock time and CPU times.

Figure 2 shows the results. The *efficient frontier* of the comparison is defined by the algorithm that for any given time achieves the lowest negative log probability. In this case the efficient frontier is occupied by the sparse KL method. Each of the algorithms is showing better performance as the number of inducing points increases. The challenging optimization behaviour of the sparse mean-field algorithm can be seen by the unpredictable changes in hold out probability as the optimization proceeds. The supplementary material shows that in terms of classification error rate this algorithm is much better behaved, suggesting that MFSp finds it difficult to produce well calibrated predictions.

The supplementary material contains plots of the hold out classification accuracy for the *image* dataset. It also contains similar plots for the *banana* dataset.

**Stochastic optimization** To demonstrate that the proposed KL method can be optimized effectively in a stochastic fashion, we first turn to the MNIST data set. Selecting the odd digits and even digits to make a binary problem, results in a training set with 60000 points. We used ADADELTA method [Zeiler, 2012] from the *climin* toolbox [Osendorfer et al., 2014]. Selecting a step-rate of 0.1, a mini-batch size of 10 with 200 inducing points resulted in a hold-out accuracy of 97.8%. The mean negative-log-probability of the testing set was 0.069. It is encouraging that we are able to fit highly nonlinear, high-dimensional decision boundaries with good accuracy, on a dataset size that is out of the range of existing methods.

Stochastic optimization of our bound allows fitting of GP classifiers to datasets that are larger than previously possible. We downloaded the flight arrival and departure times for every commercial flight in the USA

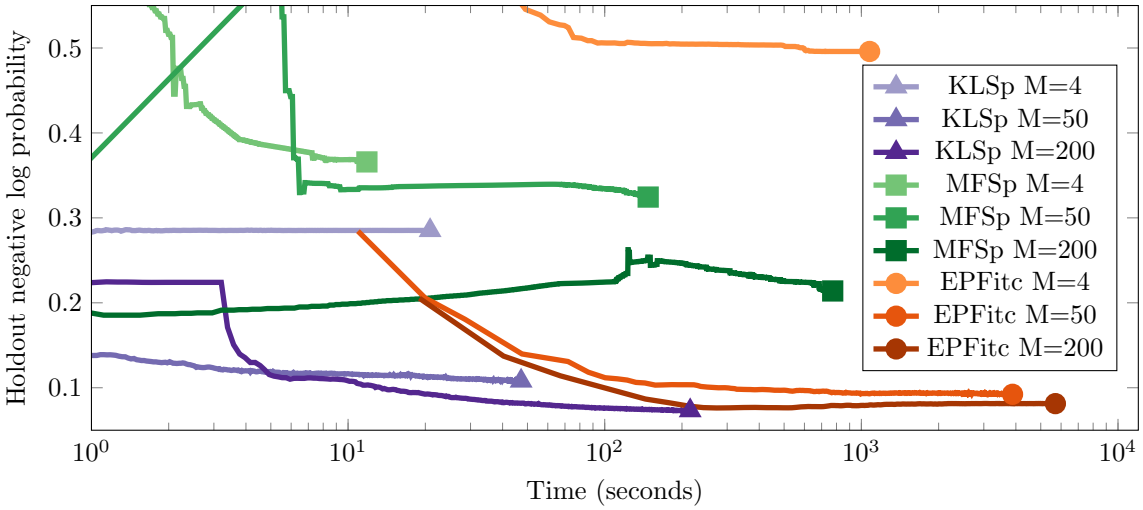


Figure 2: Temporal performance of the different methods on the *image* dataset.

from January 2008 to April 2008. [1] This dataset contains information about 5.9 million flights, including the delay in reaching the destination. We build a classifier which was to predict whether any flight was to subject to delay.

As a benchmark, we first fitted a linear model using scikits-learn [Pedregosa et al., 2011] which in turns uses LIBLINEAR [Fan et al., 2008]. On our randomly selected hold-out set of 100000 points, this achieved a surprisingly low error rate of 37%, the negative-log probability of the held-out data was 0.642.

We built a Gaussian process kernel using the sum of a Matern-$\frac{3}{2}$ and a linear kernel. For each, we introduced parameters which allowed for the scaling of the inputs (sometimes called Automatic Relevance Determination, ARD). Using a similar optimization scheme to the MNIST data above, our method was able to exceed the performance of a linear model in a few minutes, as shown in Figure 3. The kernel parameters at convergence suggested that the problem is highly nonlinear: the relative variance of the linear kernel was negligible. The optimized lengthscales for the Matern part of the covariance suggested that the most useful features were the time of day and time of year.

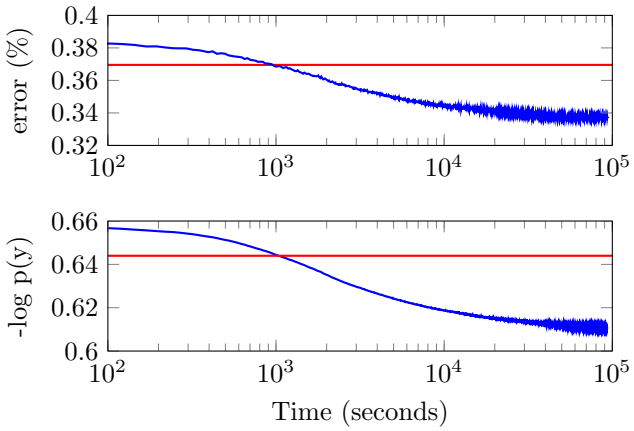


Figure 3: Performance of the airline delay dataset. The red horizontal line depicts performance of a linear classifier, whilst the blue line shows performance of the stochastically optimized KLsparse method.

## 6 Discussion

We have presented two novel variational bounds for performing sparse GP classification. The first, like the existing GFITC method, makes an approximation to the covariance matrix before introducing the non-conjugate likelihood. These approaches are somewhat unsatisfactory since in performing approximate inference, we necessarily introduce additional parameters (variational means and variances, or the parameters of EP factors), which naturally scale linearly with $N$.

Our proposed KLSP bound outperforms the state-of-the art GFITC method on benchmark datasets, and is capable of being optimized in a stochastic fashion as we have shown, making GP classification applicable to big data for the first time.

In future work, we note that this work opens the door for several other GP models: if the likelihood factorizes in $N$, then our method is applicable through Gauss-Hermite quadrature of the log likelihood. We also note that it is possible to relax the restriction of $q(\mathbf{u})$ to a Gaussian form, and mixture model approximations follow straightforwardly, allowing scalable extensions of Nguyen and Bonilla [2014].

[1] Hensman et al. use this data to illustrate regression, here we simply classify whether the delay $\leq$ zero.

## References


C. Cortes and N. D. Lawrence, editors. *Advances in Neural Information Processing Systems*, Cambridge, MA, 2014. MIT Press.

Z. Dai, A. Damianou, J. Hensman, and N. Lawrence. Gaussian process models with parallelization and gpu acceleration. *arXiv*, 2014.

A. Damianou and N. Lawrence. Deep gaussian processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pages 207–215, 2013.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

Y. Gal, M. van der Wilk, and C. E. Rasmussen. Distributed variational inference in sparse Gaussian process regression and latent variable models. In Cortes and Lawrence [2014].

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In A. Nicholson and P. Smyth, editors, *Uncertainty in Artificial Intelligence*, volume 29. AUAI Press, 2013.

J. Hensman, M. Zwießele, and N. D. Lawrence. Tilted variational bayes. In S. Kaski and J. Corander, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33, pages 356–364. JMLR W&CP, 2014.

E. Khan, S. Mohamed, and K. P. Murphy. Fast bayesian inference for non-conjugate gaussian process regression. In P. Bartlett, editor, *Advances in Neural Information Processing Systems*, pages 3140–3148, Cambridge, MA, 2012. MIT Press.

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

N. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. In *Proceedings of the 16th Annual Conference on Neural Information Processing Systems*, number EPFL-CONF-161319, pages 609–616, 2003.

N. D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. *Advances in neural information processing systems*, 16: 329–336, 2004.

A. Naish-Guzman and S. Holden. The generalized fitc approximation. In *Advances in Neural Information Processing Systems*, pages 1057–1064, 2007.

T. Nguyen and E. Bonilla. Automated variational inference for Gaussian process models. In Cortes and Lawrence [2014].

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

M. Opper and C. Archambeau. The variational Gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009.

C. Osendorfer, J. Bayer, S. Diot-Girard, T. Rueckstiess, and S. Urban. Climin. `http://github.com/BRML/climin`, 2014. Accessed: 2014-10-19.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

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

C. E. Rasmussen and H. Nickisch. Gaussian processes for machine learning (gpml) toolbox. *The Journal of Machine Learning Research*, 11:3011–3015, 2010.

C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. MIT Press, Cambridge, MA, 2006.

E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, pages 1257–1264, 2005.

M. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In D. van Dyk and M. Welling, editors, *Proceedings of the Twelfth International Workshop on Artificial Intelligence and Statistics*, volume 5, pages 567–574, Clearwater Beach, FL, 16-18 April 2009. JMLR W&CP 5.

V. Tolvanen. Gaussian processes with monotonicity constraints for big data. Master's thesis, Aalto Univeristy, June 2014.

A. G. Wilson, D. A. Knowles, and Z. Ghahramani. Gaussian process regression networks. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML)*, Edinburgh, June 2012. Omnipress.

M. D. Zeiler. Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4): 550–560, 1997.

Supplementary Material for :

# Scalable Variational Gaussian Process Classification

James Hensman, Alex Matthews and Zoubin Ghahramani

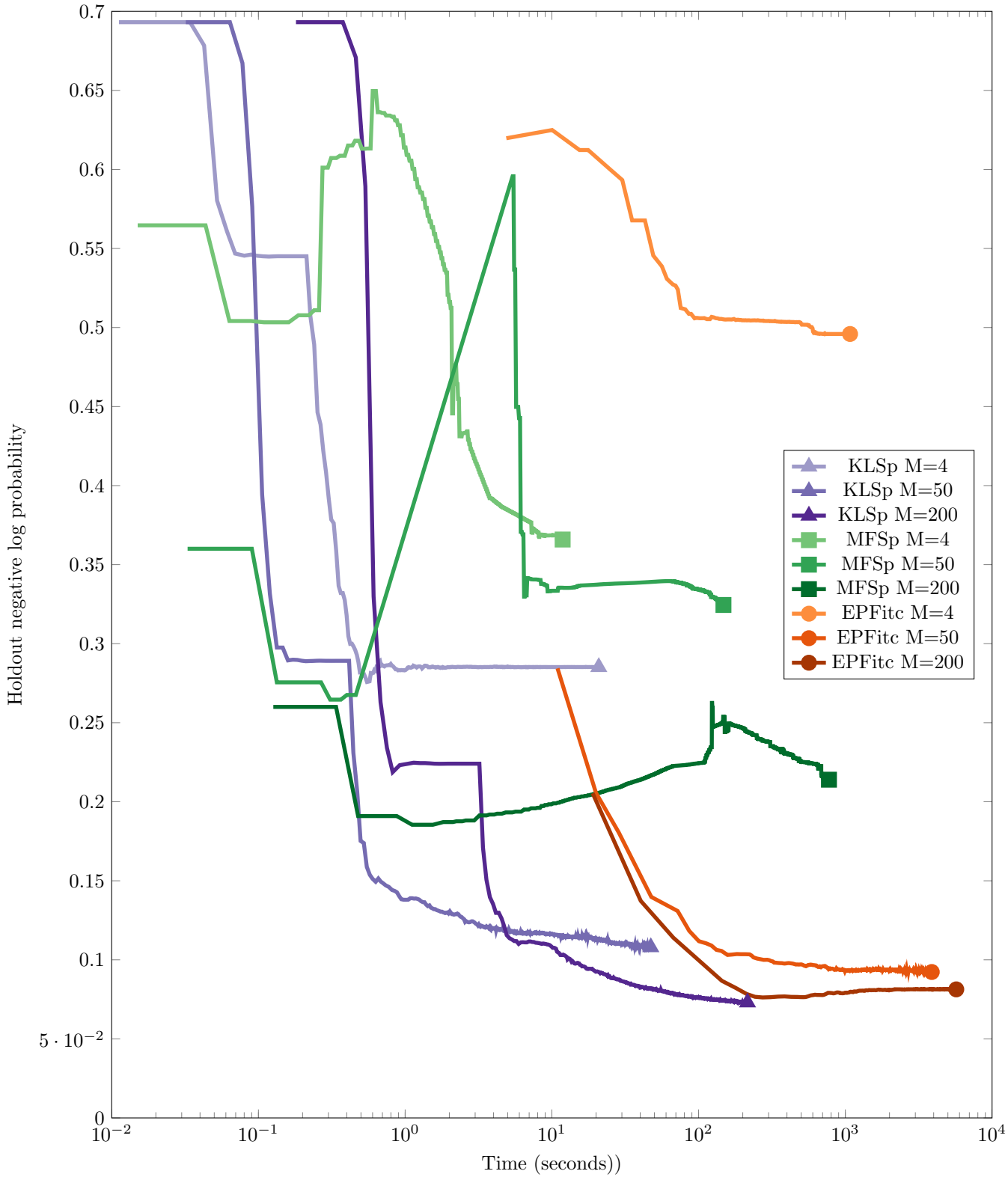


Figure S.1 : Hold out predictive densities of the different methods on the *image* dataset.

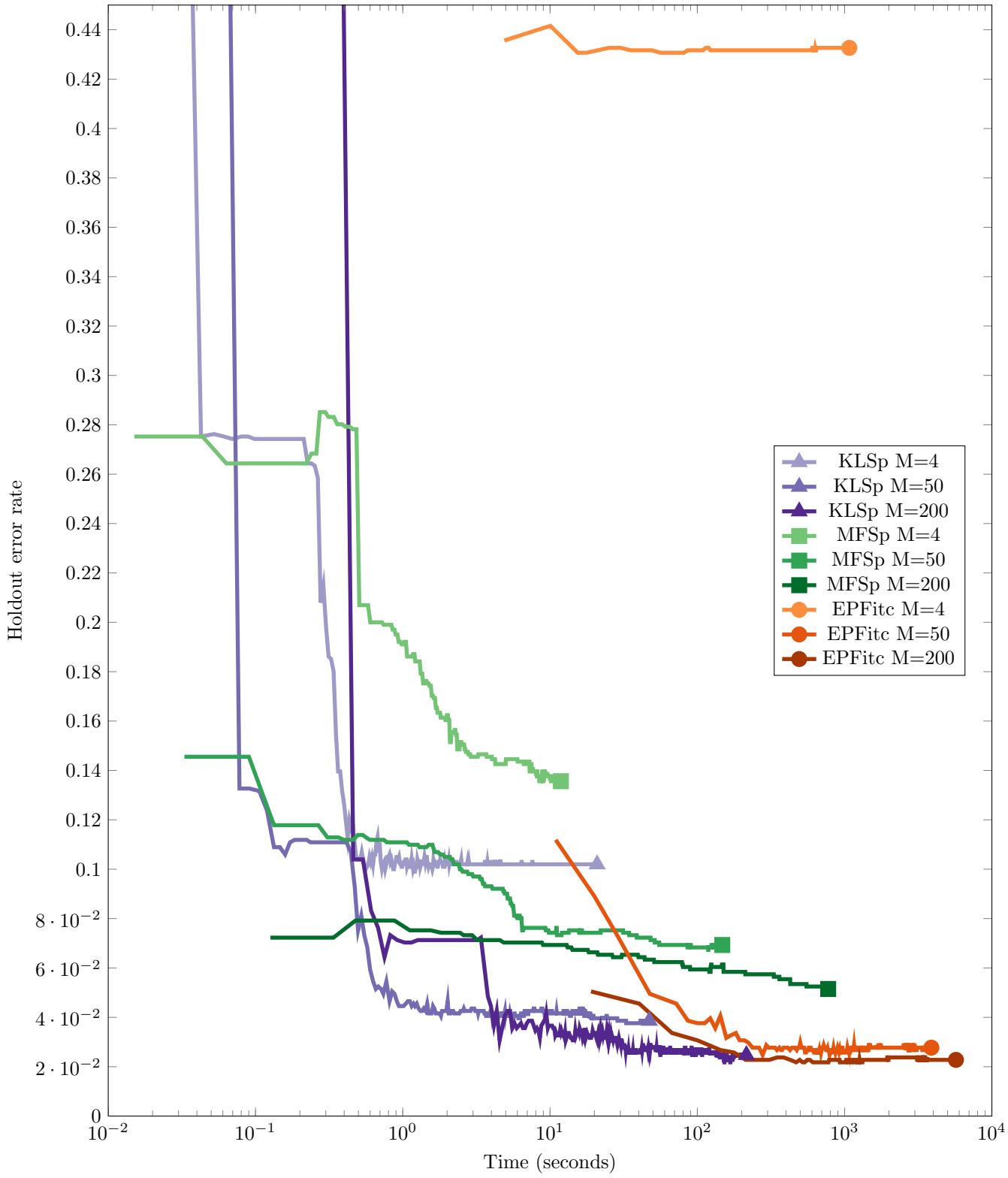


Figure S.2 : Hold out errors of the different methods on the *image* dataset.

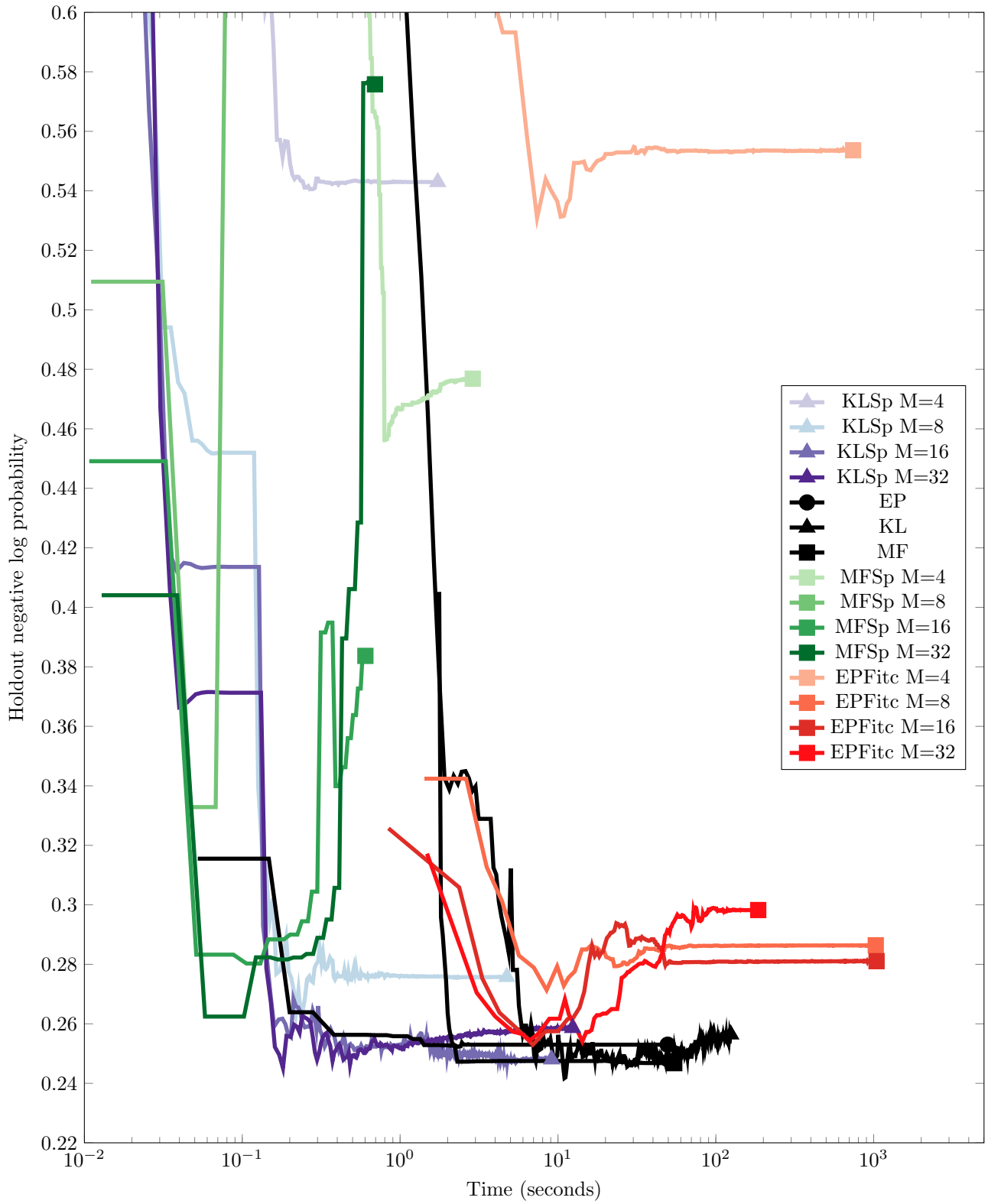


Figure S.3 : Hold out predictive densities of the different methods on the *banana* dataset.

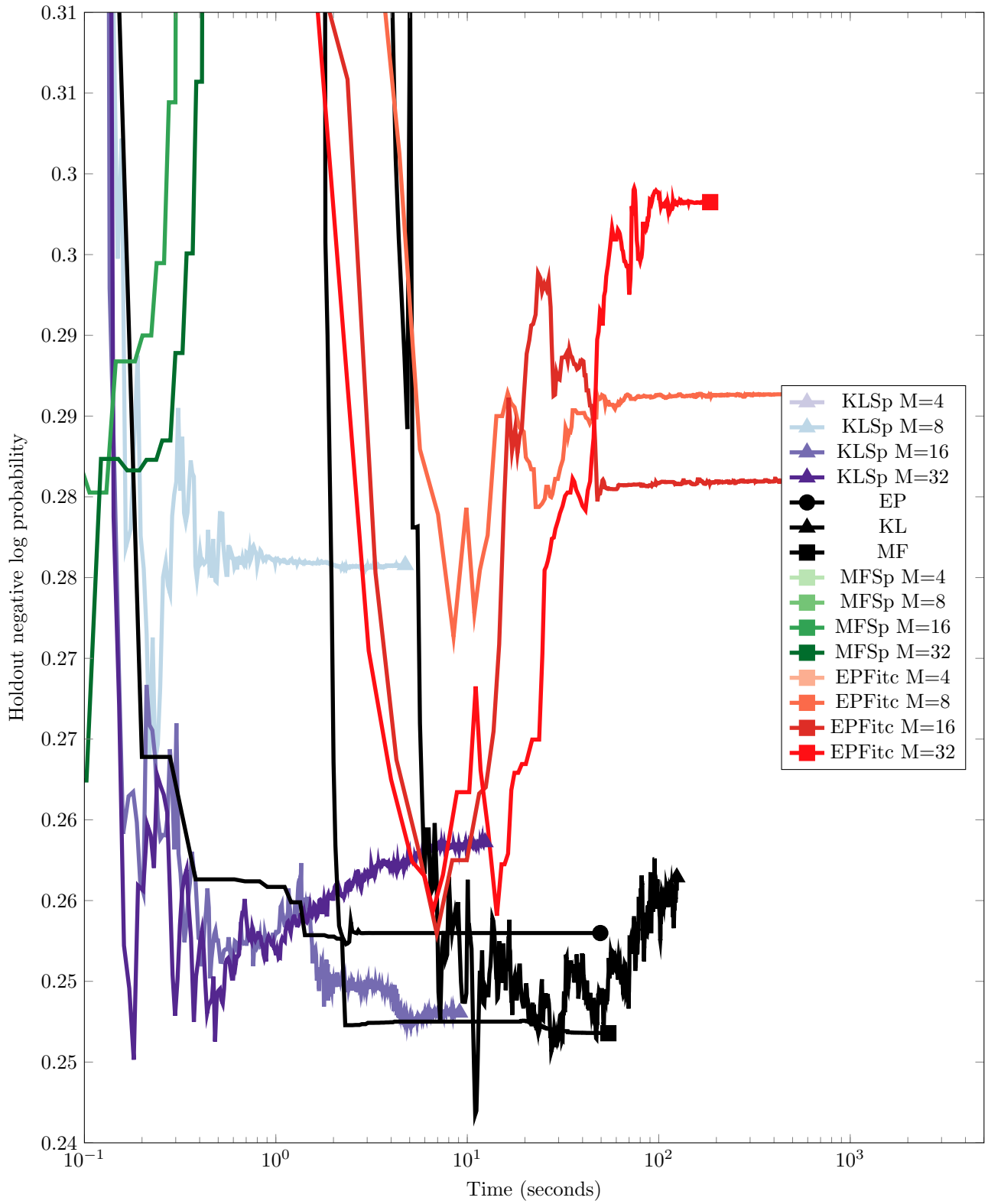


Figure S.4 : Hold out predictive densities of the different methods on the *banana* dataset.

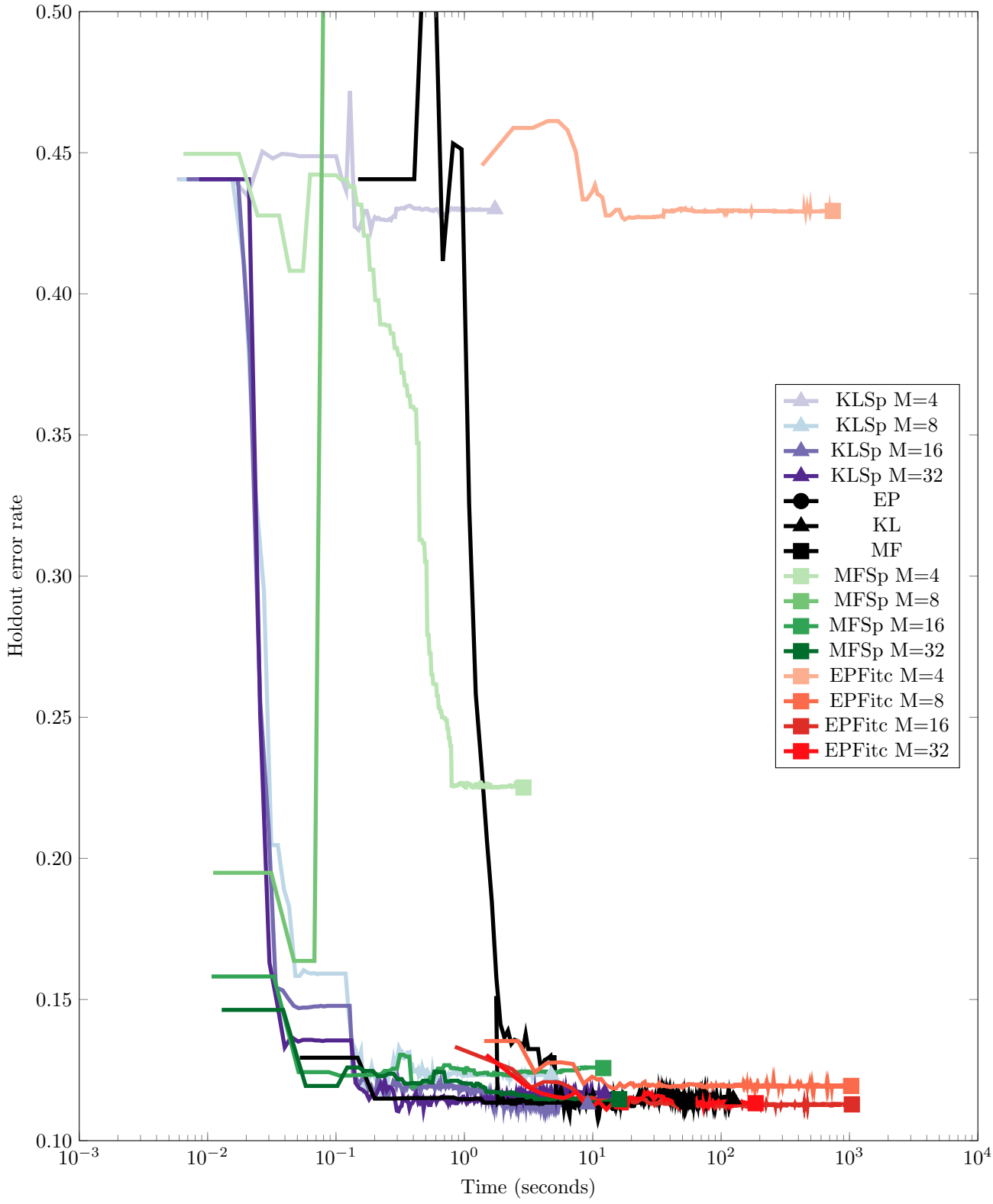


Figure S.5 : Hold out errors of the different methods on the *banana* dataset.

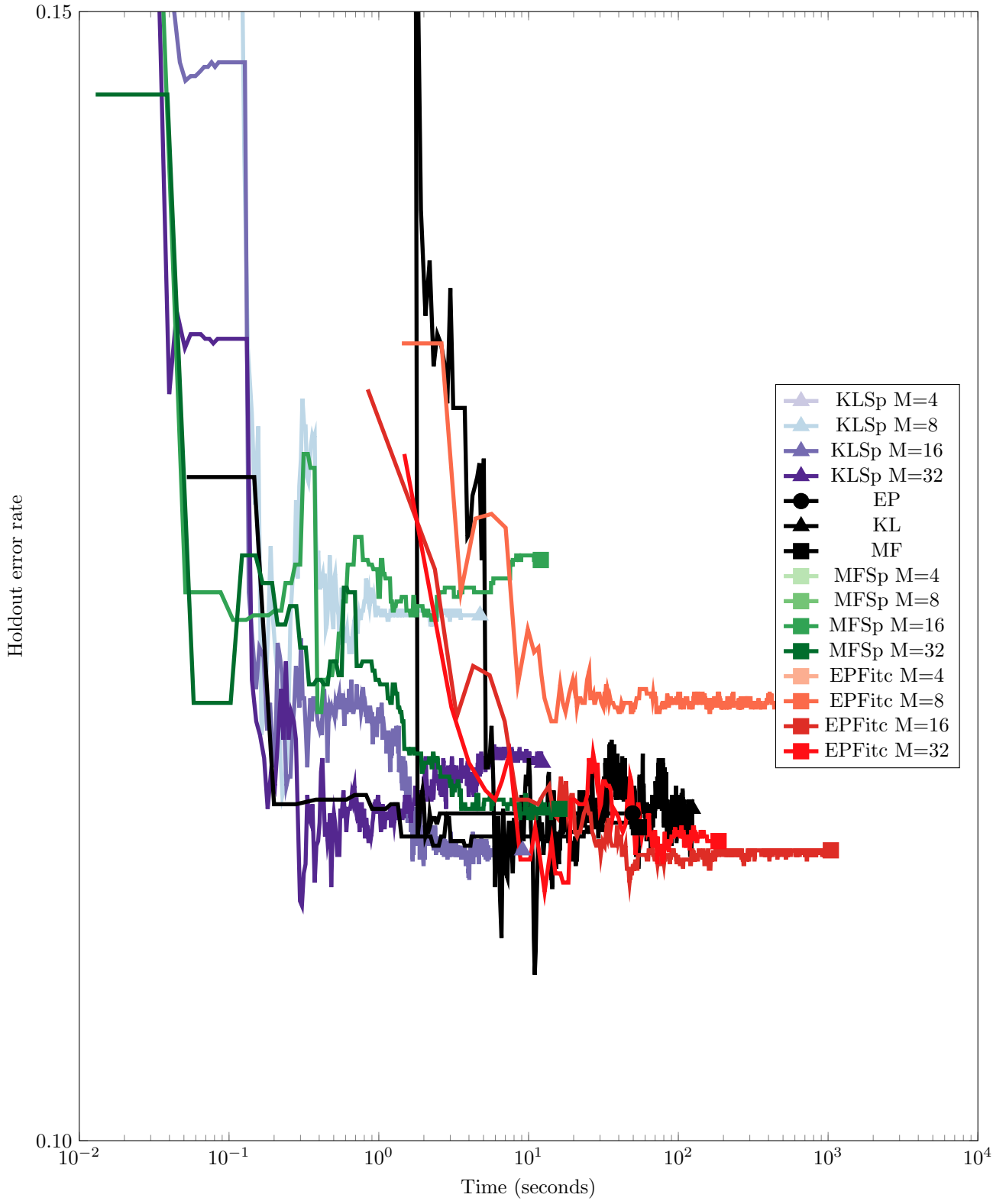


Figure S.6 : Hold out errors of the different methods on the *banana* dataset.