Sparse-posterior Gaussian Processes for general likelihoods

Abstract

Gaussian processes (GPs) provide a probabilistic nonparametric representation of functions in regression, classification, and other problems. Unfortunately, exact learning with GPs is intractable for large datasets. A variety of approximate GP methods have been proposed that essentially map the large dataset into a small set of basis points. Among them, two state-of-the-art methods are sparse pseudo-input Gaussian process (SPGP) (Snelson & Ghahramani, 2006) and variable-sigma GP (VSGP) Walder et al. (2008), which generalizes SPGP and allows each basis point to have its own length scale. However, VSGP was only derived for regression. In this paper, we propose a new sparse GP framework that uses expectation propagation to directly approximate general GP likelihoods using a sparse and smooth basis. It includes both SPGP and VSGP for regression as special cases. Plus as an EP algorithm, it inherits the ability to process data online. As a particular choice of approximating family, we blur each basis point with a Gaussian distribution that has a *full covariance matrix* representing the data distribution around that basis point; as a result, we can summarize local data manifold information with a small set of basis points. Our experiments demonstrate that this framework outperforms previous GP classification methods on benchmark datasets in terms of minimizing divergence to the non-sparse GP solution as well as lower misclassification rate.

1. Introduction

Gaussian processes (GP) are powerful nonparametric Bayesian approach to modelling unknown functions. As such, they can be directly used for classification and regression (Rasmussen & Williams, 2006), or embedded into a larger model such as factor analysis (Teh et al., 2005), relational learning (Chu et al., 2006), or reinforcement learning (Deisenroth et al., 2009). Unfortunately, the cost of GPs can be prohibitive for large datasets. Even for the regression case where the GP prediction formula is analytic, training the exact GP model with N points demands an $O(N^3)$ cost for inverting the covariance matrix and predicting a new output requires $O(N^2)$ cost in addition to storing all of the training points.

Ideally, we would like a compact representation, much smaller than the number of training points, of the posterior distribution for the unknown function. This compact representation could be used to summarize training data for Bayesian learning, or it could be passed around as a message in order to do inference in a larger probabilistic model. One successful approach is to map the training data into a small set of basis points, then compute the exact posterior that results from those points. The basis points could literally be a subset of the training instances (Csató, 2002; Lawrence et al., 2002) or they could be artificial "pseudo-inputs" that represent the training set (Snelson & Ghahramani, 2006). A general framework proposed by Quiñonero-Candela & Rasmussen (2005) shows that many sparse GP regression algorithms, including the "pseudo-inputs" approach, can viewed as exact inference methods with an approximate, sparse GP *prior*.

Furthermore, the GP that is used to for inference on the basis points need not match the original GP, so that a more compact representation can be used without degenerating the approximation quality. In particular, as shown by Walder et al. Walder et al. (2008), the GP applied to the basis points should have a longer length scale than the original (since the data is now sparser). Their "variable sigma Gaussian process" (VSGP) algorithm (Walder et al., 2008) allows a different length scale for each basis point. Lázaro-Gredilla & Figueiras-Vidal (2009) generalized this idea to allow an arbitrary set of extra parameters (such as a frequency-scale) for each basis point. However, this extension as well as VSGP is limited to linear regression.

In this paper, we provide a new framework, Sparse And Smooth Posterior Approximation (SASPA), for learning sparse GP models with arbitrary likelihoods. Unlike Quiñonero-Candela & Rasmussen (2005)'s work, this new framework is constructed from a different perspective, the

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

posterior-approximation perspective: we directly approximates the posterior distributions of exact full GPs by a sparse, *blurred* GP using expectation propagation (Minka, 2001b). The SASPA framework gives new insight to previous sparse GP models, for example, VSGP, generalize over them to handle flexible likelihoods, and provides convenient practical tools for designing new GP learning methods.

In summary, the main contributions of this paper include the following:

- We present a new framework, SASPA, for sparse Gaussian process learning. In this framework, VSGP and other sparse GP algorithms can be obtained by employing a particular choice of approximating family.
- In a richer approximating family, we blur each basis point with another distribution. In particular, we use a Gaussian distribution that has a full covariance matrix representing the data distribution around the basis point. Therefore, the SASPA model can effectively summarizes *local data manifold* information with a small set of basis points.
- We describe how to apply the SASPA framework to GP models with regression and classification likelihoods in section 3.2.
- Finally, in section 6, we demonstrate the improved approximation quality of SASPA over previous sparse GP methods on both synthetic data and standard UCI benchmark data.

2. Gaussian process models

We denote N independent and identically distributed samples as $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}_N$, where $\mathbf{x}_i$ is a d dimensional input and y_i is a scalar output. We assume there is a latent function f that we are modeling and the noisy realization of latent function f at $\mathbf{x}_i$ is y_i.

A Gaussian process places a prior distribution over the latent function f. Its projection at the samples $\{\mathbf{x}_i\}$ defines a joint Gaussian distribution:

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{m}^0, K)$$

where $m_i^0 = m^0(\mathbf{x}_i)$ is the mean function and $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ is the covariance function, which encodes the prior notation of smoothness. Normally the mean function is simply set to be zero and we follow this tradition in this paper. A typical kernel covariance function is the squared exponential, also know as Radial Basis Function (RBF),

$$k(\mathbf{x}, \mathbf{x}') = \exp\big(-\frac{||\mathbf{x}' - \mathbf{x}||^2}{2\eta^2}\big), \tag{1}$$

where η is a hyperparameter.

For regression, we use a Gaussian likelihood function

$$p(y_i|f) = \mathcal{N}(y_i|f(\mathbf{x}_i), v_y) \tag{2}$$

where v_y is the observation noise. For classification, the data likelihood has the form

$$p(y_i|f) = (1-\epsilon)\sigma(f(\mathbf{x}_i)y_i) + \epsilon\sigma(-f(\mathbf{x}_i)y_i) \tag{3}$$

where ϵ models the labeling error and $\sigma(\cdot)$ is a nonlinear function, ie., a cumulative Gaussian distribution or a step function, so that $\sigma(f(x_i)y_i) = 1$ if $f(x_i y_i) \geq 0$ and $\sigma(f(x_i)y_i) = 0$ otherwise.

Given the Gaussian process prior over f and the data likelihood, the posterior process is

$$p(f|\mathcal{D}, \mathbf{t}) \propto GP(f|0, K) \prod_{i=1}^{N} p(y_i|f) \tag{4}$$

Since the Gaussian process is grounded on the N examples, they are called the basis points.

For the regression problem, the posterior process has an analytical form. But to make a prediction on a new sample, we need to invert a N by N matrix. If the training set is big, this matrix inversion will be too costly. For classification or other nonlinear problems, the computational cost is even higher since we do not have a analytical solution to the posterior process and the complexity of the process grows with the number of training samples.

3. Sparse-posterior GP: a SASPA perspective

In this section we present the SASAP framework, started with the approximating family.

3.1. Approximating family

To save the computational and memory cost, we approximate the exact Gaussian process posterior (4) by a sparse posterior process parameterized by $(\mathbf{u}, \mathbf{b}, \mathbf{\Lambda})$:

$$q(f) \propto GP(f|0, K)\mathcal{N}(\mathbf{u}|\mathbf{g}_B(f), \mathbf{\Lambda}^{-1}) \tag{5}$$

where $\mathbf{u} = (u_1, \ldots, u_M)$, the subscript B denote the basis set $(b_1, \ldots, b_M)$, and

$$\mathbf{g}_B(f) = \big[\int f(\mathbf{x})\phi(\mathbf{x}|b_1)\mathrm{d}\mathbf{x}, \ldots, \int f(\mathbf{x})\phi(\mathbf{x}|b_M)\mathrm{d}\mathbf{x}\big]^{\mathrm{T}}$$

In this sparse representation, we can change the number of basis points, M, to regulate its model complexity. Normally, we set $M \ll N$. In general, the basis point b_k is blurred by a convolving function $\phi(\mathbf{x}|b_k)$, which can be a

Gaussian distribution, representing how the data are distributed *locally* around b_k. The parameter u_k represents a virtual regression target for b_k, and λ_k is its precision.

The approximate posterior process $q(f)$ is a Gaussian process, so it has a mean and covariance function. To determine these, first consider the following functional:

$$Z(m^0) = \int GP(f|m^0, K)\mathcal{N}(\mathbf{u}|\mathbf{g}_B(f), \mathbf{\Lambda}^{-1})\mathrm{d}f$$

Now $\mathbf{g}_B(f)$ follows a joint Gaussian distribution with the mean $\tilde{\mathbf{m}}_B^0$ and $\tilde{\mathbf{V}}_B^0$:

$$\begin{aligned}\tilde{m}_{Bj}^0 &= \int\int f(\mathbf{x})\phi(\mathbf{x}|b_j)GP(f|m^0, K)\mathrm{d}\mathbf{x}\mathrm{d}f \\ &= \int m^0(\mathbf{x})\phi(\mathbf{x}|b_j)\mathrm{d}\mathbf{x} \qquad (6)\\ \tilde{V}_{Bij}^0 &= \int\int \phi(\mathbf{x}|b_i)K(\mathbf{x},\mathbf{x}')\phi(\mathbf{x}'|b_j)\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{x}' \qquad (7)\end{aligned}$$

Therefore, we can compute $Z(m^0)$ as follows:

$$\begin{aligned}Z(m^0) &= \int \mathcal{N}(\mathbf{u}|\mathbf{g}_B, \mathbf{\Lambda}^{-1})\mathcal{N}(\mathbf{g}_B|\tilde{\mathbf{m}}_B^0, \tilde{\mathbf{V}}_B^0)\mathrm{d}f \\ &= \mathcal{N}(\mathbf{u}|\tilde{\mathbf{m}}_B^0, \beta^{-1}) \qquad (8)\end{aligned}$$

where $\beta = (\tilde{\mathbf{V}}_B^0 + \mathbf{\Lambda}^{-1})^{-1}$. Define $\alpha = \beta\mathbf{u}$. Then the mean and the covariance functions of the sparse process are characterized by α and β. In particular, we derive the following theorem to describe the relationship between them.

Theorem 1 *The posterior process $q(f)$ defined in (5) has the mean function $m(\mathbf{x})$ and covariance function $V(\mathbf{x},\mathbf{x}')$:*

$$\begin{aligned}m(\mathbf{x}) &= \tilde{K}(\mathbf{x}, B)\alpha \qquad (9)\\ V(\mathbf{x},\mathbf{x}') &= K(\mathbf{x},\mathbf{x}') - \tilde{K}(\mathbf{x}, B)\beta\tilde{K}(B,\mathbf{x}') \qquad (10)\end{aligned}$$

where $\tilde{K}(\mathbf{x}, B) = [\tilde{K}(\mathbf{x}, b_1), \ldots, \tilde{K}(\mathbf{x}, b_M)]$, $\tilde{K}(\mathbf{x}, b_j) = \int K(\mathbf{x},\mathbf{x}')\phi(\mathbf{x}'|b_j)\mathrm{d}\mathbf{x}'$, and $\tilde{K}(B,\mathbf{x}) = (\tilde{K}(\mathbf{x}, B))^{\mathrm{T}}$.

When using the RBF covariance function (1) and setting $\phi(\mathbf{x}|b_i) = \mathcal{N}(\mathbf{x}|a_i, c_i)$ where $b_i = (a_i, c_i)$, we have

$$\begin{aligned}\tilde{K}(x, B) &= (2\pi\eta^2)^{M/2}\cdot \\ &\cdot [\mathcal{N}(\mathbf{x}|a_1, c_1 + \eta^2\mathbf{I}), \ldots, \mathcal{N}(\mathbf{x}|a_M, c_M + \eta^2\mathbf{I})]. \qquad (11)\end{aligned}$$

Proof: First, consider the minimization of the KL divergence between the posterior process $q(f) \propto GP(f|m^0, K)\mathcal{N}(\mathbf{u}|\mathbf{g}_B(f), \mathbf{\Lambda}^{-1})$ for the sparse process and a process $\bar{q}(f)$ in the exponential family. Since $q(f)$ also belongs to the exponential family, this minimization will achieve the optimal solution, i.e., $\bar{q}(f) = q(f)$.

Now to obtain the mean function $m(\mathbf{x})$ and the covariance function $V(\mathbf{x},\mathbf{x}')$ for $q(f)$, we can solve $\bar{q}(f)$ by the KL minimization. This leads to the following moment matching equations:

$$\begin{aligned}m(\mathbf{x}) =& m^0(\mathbf{x}) + \int K(\mathbf{x},\mathbf{a}')\frac{\mathrm{d}\log Z}{\mathrm{d}m^0(\mathbf{a}')}\mathrm{d}\mathbf{a}' \qquad (12)\\ V(\mathbf{x},\mathbf{x}') =& K(\mathbf{x},\mathbf{x}') + \qquad (13)\\ & \int\int K(\mathbf{x},\mathbf{a})\frac{\mathrm{d}^2\log Z}{\mathrm{d}m^0(\mathbf{a})\mathrm{d}m^0(\mathbf{a}')}K(\mathbf{a}',\mathbf{x}')\mathrm{d}\mathbf{a}\mathrm{d}\mathbf{a}'\end{aligned}$$

Based on (8), it is easy to obtain

$$\begin{aligned}\frac{\mathrm{d}\log Z}{\mathrm{d}m^0(\mathbf{a})} &= \frac{\mathrm{d}}{\mathrm{d}\tilde{\mathbf{m}}_B}\big(-\frac{1}{2}(\mathbf{u}-\tilde{\mathbf{m}}_B)^{\mathrm{T}}\beta(\mathbf{u}-\tilde{\mathbf{m}}_B)\big)\frac{\mathrm{d}\tilde{\mathbf{m}}_B}{\mathrm{d}m^0(\mathbf{a})} \\ &= [\phi(\mathbf{x}|b_1), \ldots, \phi(\mathbf{x}|b_M)]\beta(\mathbf{u}-\tilde{\mathbf{m}}_B) \qquad (14)\end{aligned}$$

Combining (12) and (14) gives

$$\begin{aligned}m(\mathbf{x}) =& m^0(\mathbf{x}) + \tilde{K}(\mathbf{x}, B)\beta(\mathbf{u}-\boldsymbol{\rho}) \\ =& m^0(\mathbf{x}) + \tilde{K}(\mathbf{x}, B)(\alpha - \beta\tilde{\mathbf{m}}_B^0) \qquad (15)\end{aligned}$$

where $\tilde{K}(\mathbf{x}, B)$ is defined in (11). Setting the prior mean function $m^0(\mathbf{x}) = 0$, we have $\tilde{\mathbf{m}}_B^0 = 0$. As a result, equation (9) holds.

From (14) it follows that

$$\begin{aligned}&\frac{\mathrm{d}^2\log Z}{\mathrm{d}m^0(\mathbf{x})\mathrm{d}m^0(\mathbf{x}')} \\ &= -[\phi(\mathbf{x}|b_1), \ldots, \phi(\mathbf{x}|b_M)]\beta\frac{\mathrm{d}\boldsymbol{\rho}}{\mathrm{d}m^0(\mathbf{x}')} \\ &= -[\phi(\mathbf{x}|b_1), \ldots, \phi(\mathbf{x}|b_M)]\beta\cdot \\ &\quad\cdot[\phi(\mathbf{x}'|b_1), \ldots, \phi(\mathbf{x}'|b_M)]^{\mathrm{T}} \qquad (16)\end{aligned}$$

Based on the above equation and (13), we have

$$V(\mathbf{x},\mathbf{x}') = K(\mathbf{x},\mathbf{x}') - \tilde{K}(\mathbf{x}, B)\beta\tilde{K}(B,\mathbf{x}')$$

Thus (10) holds. □

Based on Theorem 1, we have the following corollary:

Corollary 2 *The projection of the blurred Gaussian posterior process $q(f)$ onto B is a Gaussian distribution with the following mean and covariance:*

$$\begin{aligned}\tilde{\mathbf{m}}_B &= \hat{\mathbf{K}}\alpha \qquad (17)\\ \tilde{\mathbf{V}}_B &= \hat{\mathbf{K}} - \hat{\mathbf{K}}\beta\hat{\mathbf{K}}^{\mathrm{T}} \qquad (18)\end{aligned}$$

where $\hat{K}_{ij} = \int\int \phi(\mathbf{x}'|b_i)K(\mathbf{x},\mathbf{x}')\phi(\mathbf{x}'|b_j)\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{x}'$.

Assuming $\mathbf{b}$ is given, the remaining question is how to estimate $(\mathbf{u}, \mathbf{\Lambda})$ – or equivalently (α, β) – for the sparse posterior process $q(f)$, such that it well approximates the exact posterior process $p(f|\mathcal{D}, \mathbf{t})$.

3.2. Inference by expectation propagation

We apply expectation propagation to fit $q(f)$. EP has three steps, message deletion, data projection, and message updates, iteratively applied to each training point. In the message deletion step, we compute the partial belief $q^{\setminus i}(f; m^{\setminus i}, v^{\setminus i})$ by removing a message $\tilde{t}_i$ (from the *i-th* point) from the approximate posterior $q^{\text{old}}(f|m, v)$. In the data projection step, we minimize the KL divergence between $\tilde{p}(f) \propto p(t_i; f)q(f; m^{\setminus i}, v^{\setminus i})$ and the new approximate posterior $q^{\text{old}}(f|m, v)$, such that the information from each data point is incorporated into the model. Finally, the message $\tilde{t}_i$ is updated based on the new and old posteriors.

Based on (5), the sparse GP is an exponential family with features $(\mathbf{g}_B(f), \mathbf{g}_B(f)\mathbf{g}_B(f)^{\mathrm{T}})$. As a result, we can determine the sparse GP that minimizes $KL(\tilde{p}(f)|q(f))$ by matching the moments on $\mathbf{g}_B(f)$.

Similar to (12) and (13), the moment matching equations are

$$\tilde{\mathbf{m}}_B = \tilde{\mathbf{m}}_B^{\setminus i} + \tilde{V}^{\setminus i}(B, \mathbf{x}_i)\frac{\mathrm{d}\log Z}{\mathrm{d}m^{\setminus i}(\mathbf{x}_i)} \tag{19}$$

$$\tilde{\mathbf{V}}_B = \tilde{\mathbf{V}}_B^{\setminus i} + \tilde{V}^{\setminus i}(B, \mathbf{x}_i)\frac{\mathrm{d}^2\log Z}{(\mathrm{d}m^{\setminus i}(\mathbf{x}_i))^2}\tilde{V}^{\setminus i}(\mathbf{x}_i, B) \tag{20}$$

where $\tilde{V}^{\setminus i}(B, \mathbf{x})_j = \int \phi(\mathbf{x}'|b_j)V^{\setminus i}(\mathbf{x}', \mathbf{x})\mathrm{d}\mathbf{x}'$.

Combining (17) and (19) gives

$$\begin{aligned} \mathbf{p}_i &= \hat{\mathbf{K}}^{-1}\tilde{K}(B, x_i) \\ \mathbf{h} &= \hat{\mathbf{K}}^{-1}\tilde{V}^{\setminus i}(B, \mathbf{x}_i) = \mathbf{p}_i - \beta^{\setminus i}\tilde{K}(B, x_i) \\ \alpha &= \alpha^{\setminus i} + \mathbf{h}\frac{\mathrm{d}\log Z}{\mathrm{d}m^{\setminus i}(\mathbf{x}_i)} \end{aligned} \tag{21}$$

where we use (10) to obtain the last equation in the second line.

Inserting (18) to (20), we get

$$\beta = \beta^{\setminus i} - \mathbf{h}\mathbf{h}^{\mathrm{T}}\frac{\mathrm{d}^2\log Z}{(\mathrm{d}m^{\setminus i}(\mathbf{x}_i))^2} \tag{22}$$

These equations define the projection update. This update can be equivalently interpreted as multiplying $q^{\setminus i}(f)$ by an approximate factor $\tilde{t}_i(f)$ defined as:

$$\tilde{t}_i(f) = \mathcal{N}(\sum_j p_{ij}\int f(\mathbf{x})\phi(\mathbf{x}|b_j)\mathrm{d}\mathbf{x}|g_i, \tau_i^{-1}) \tag{23}$$

$$\tau_i^{-1} = (-\nabla_m^2 \log Z)^{-1} - \tilde{K}(\mathbf{x}_i, B)\mathbf{h} \tag{24}$$

$$g_i = m^{\setminus i}(\mathbf{x}_i) + (-\nabla_m^2 \log Z)^{-1}\nabla_m \log Z \tag{25}$$

The approximation factor $\tilde{t}_i(f)$ can be viewed as a message from the *i-th* data point to the sparse GP. To check the validity of this update, we compute

$$\begin{aligned} \tilde{Z} &= \int \tilde{t}_i(f)q^{\setminus i}(f)df \propto \mathcal{N}(u_i|\mathbf{p}_i^{\mathrm{T}}\tilde{\mathbf{m}}_B^{\setminus i}, \tau_i^{-1} + \mathbf{p}_i^{\mathrm{T}}\tilde{\mathbf{V}}_B^{\setminus i}\mathbf{p}_i) \\ &= \mathcal{N}(u_i|m^{\setminus i}(\mathbf{x}_i), \tau_i^{-1} + \tilde{K}(\mathbf{x}_i, B)\mathbf{h}) \end{aligned} \tag{26}$$

which has the same derivatives as the original $Z = \int t_i(f)q^{\setminus i}(f)df$. Therefore, the multiplication $\tilde{t}_i(f)q^{\setminus i}(f)$ leads to the same $q(f|m, v)$. In other words, we have $\tilde{t}_i(f) \propto q(f)/q^{\setminus i}(f)$.

To delete this message, we multiple its reciprocal with the current $q(f)$. Using the same trick as before, we can solve the multiplication using the following moment matching equations:

$$\mathbf{h}^{\setminus i} = \mathbf{p}_i - \beta\tilde{K}(B, \mathbf{x}_i) \tag{27}$$

$$\begin{aligned} \tilde{Z}_d &= \int \frac{1}{\tilde{t}_i(f)}q^{\setminus i}(f)\mathrm{d}f \\ &\propto \mathcal{N}(u_i|\mathbf{p}_i^{\mathrm{T}}\tilde{\mathbf{m}}_B^{\setminus i}, -\tau_i^{-1} + \tilde{K}(\mathbf{x}_i, B)\mathbf{h}^{\setminus i}) \end{aligned}$$

$$\frac{\mathrm{d}^2\log\tilde{Z}_d}{(\mathrm{d}m(\mathbf{x}_i))^2} = -(-\tau_i^{-1} + \tilde{K}(\mathbf{x}_i, B)\mathbf{h}^{\setminus i})^{-1} \tag{28}$$

$$\frac{\mathrm{d}\log\tilde{Z}_d}{\mathrm{d}m(\mathbf{x}_i)} = (-\frac{\mathrm{d}^2\log Z}{(\mathrm{d}m(\mathbf{x}_i))^2})(g_i - \tilde{K}(\mathbf{x}_i, B)\alpha) \tag{29}$$

$$\alpha^{\setminus i} = \alpha + \mathbf{h}^{\setminus i}\frac{\mathrm{d}\log\tilde{Z}_d}{\mathrm{d}m(\mathbf{x}_i)} \tag{30}$$

$$\beta^{\setminus i} = \beta - \mathbf{h}^{\setminus i}\frac{\mathrm{d}^2\log\tilde{Z}_d}{(\mathrm{d}m(\mathbf{x}_i))^2}(\mathbf{h}^{\setminus i})^{\mathrm{T}} \tag{31}$$

The EP inference for SASPA is summarized in Algorithm 1.

Algorithm 1 SASPA

```
1. Initialize $q(f)$, $g_i$, and $\tau_i$ all to be 0.
2. Loop until the change over all $g_i$, and $\tau_i$ is smaller
   than a threshold
     Loop over all training data point $\mathbf{x}_i$
       Deletion. Compute $\alpha^{\setminus i}$ and $\beta^{\setminus i}$ for $q^{\setminus i}(f)$
           via (30) and (31).
       Projection. Compute $\alpha$ and $\beta$ for the posterior $\mathbf{q}(f)$
           via (21) and (22).
       Inclusion. Update $g_i$, and $\tau_i$ for the message $\tilde{t}_i$
           via (24) and (25).
```

3.3. Regression

Given the linear regression likelihood (2), the quantities in the projection step (21)(22) are

$$\frac{\mathrm{d}\log Z}{\mathrm{d}m(\mathbf{x}_i)} = \frac{y_i - m^{\setminus i}(x_i)}{v_y + v^{\setminus i}(x_i, x_i)} \tag{32}$$

$$\frac{\mathrm{d}^2\log Z}{(\mathrm{d}m(\mathbf{x}_i))^2} = \frac{-1}{v_y + v^{\setminus i}(x_i, x_i)} \tag{33}$$

3.4. Classification

Given the classification likelihood (3) where $\sigma(\cdot)$ is the step function, the quantities in the projection step (21)(22) are

$$z = \frac{m^{\setminus i}(x_i)y_i}{\sqrt{v^{\setminus i}(x_i, x_i)}} \tag{34}$$

$$Z = \epsilon + (1 - 2\epsilon)\psi(z) \tag{35}$$

$$\frac{\mathrm{d}\log Z}{\mathrm{d}m(\mathbf{x}_i)} = \gamma y_i \tag{36}$$

$$\frac{\mathrm{d}^2\log Z}{(\mathrm{d}m(\mathbf{x}_i))^2} = -\frac{\gamma(m^{\setminus i}(x_i)y_i + v^{\setminus i}(x_i, x_i)\gamma)}{v^{\setminus i}(x_i, x_i)} \tag{37}$$

where $\gamma = \frac{(1-2\epsilon)\mathcal{N}(z|0,1)}{Z\sqrt{v^{\setminus i}(\mathbf{x}_i)}}$ and $\psi(\cdot)$ is the standard Gaussian cumulative distribution function.

4. Related work

One of the simplest and fastest approaches to reducing the cost of GPs is to train on a subset of the data. For example, the IVM trains on an intelligently chosen subset. Alternatively, we can train on all points, but approximate the contribution of each point. Quiñonero-Candela & Rasmussen (2005) compared several such approximations for regression problems and showed that they can be interpreted as exact inference on an approximate model. This perspective allowed them to show that the FITC approximation was an improvement over DTC. Walder et al. (2008) and Lázaro-Gredilla & Figueiras-Vidal (2009) later extended FITC to allow basis-dependent length-scales or frequency-scales, and showed that these extensions can also be viewed as exact inference on an approximate model.

However, while the perspective of Quiñonero-Candela & Rasmussen (2005) is useful for comparing sparse approximations, it has serious limitations as a framework for designing algorithms. First, if we treat the inducing points as model parameters and train them to maximize likelihood, then the approximation may overfit and diverge from the original GP (Titsias, 2009). Second, because the framework relies on exact inference, it only applies to regression problems with linear-Gaussian likelihoods. For classification problems, the inference stage must also be approximate, leaving us with two separate stages of approximation (see e.g. Naish-Guzman & Holden (2007)). Thirdly, this stagewise design is an obstacle to online learning, where we want to interleave the choice of inducing points with the acquisition of new data.

Our work is inspired by the work of Csató & Opper (2000), who showed that online GP classification using EP could be made sparse, using an approximation equivalent to FITC. However, the FITC approximation was introduced as a subroutine, not presented as a part of EP itself. (Naish-Guzman & Holden (2007) presented an equivalent batch algorithm, in which FITC is applied to the GP prior, followed by EP to approximate the likelihood terms.) If we use linear Gaussian likelihoods and the delta function as the blurring function, SASPA reduces to FITC for regression. Similarly, with linear Gaussian likelihoods and sphere Gaussians as the blurring functions, SASPA reduces to VSGP.

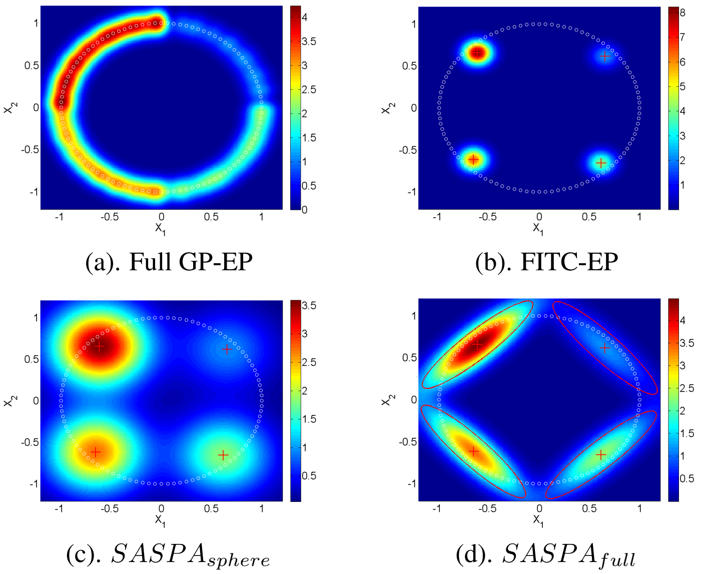
Csató (2002) later gave a batch EP algorithm, where a different sparse GP approximation (DTC) was used as a subroutine within EP. The software distributed by Csato can run either online or batch and has an option to use either FITC or DTC. In this paper, we clarify Csato and Opper's work by showing that the FITC approximation can in fact be viewed as part of the overall EP approximation. Furthermore we extend it to include basis-dependent length-scales as in Walder et al. (2008) and Lázaro-Gredilla & Figueiras-Vidal (2009).

5. model selection

The SASPA framework we have presented works for any choice of inducing points, and does not specifies how they should be chosen. Therefore we can use any of the online or batch selection methods available in the literature. Because our goal is to approximate the posterior distribution from the full GP, the optimal strategy would be to minimize the KL divergence between the two, as done by Titsias (2009). However, we want to minimize the divergence $KL(p\,||\,q)$ not $KL(q\,||\,p)$ as done by Titsias (2009). This remains an open problem. For the current paper, we are focusing on the EP component, so we are content to use a simple-minded batch approach for basis selection that nevertheless works quite well: clustering the training data. The advantage of this algorithm is that it provides centers for the inducing points as well as covariance information that can be used to choose length-scales.

6. Experiments

We evaluate SASPA on both synthetic and real world data



(a). Full GP-EP

(b). FITC-EP

(c). $SASPA_{sphere}$

(d). $SASPA_{full}$

Figure 1. Illustration on simple circle data. The heatmaps represent the values of the posterior means of different methods. Red ellipses and crosses are the mean and the standard deviation of local covariances for SASPA. The white dots are the training data points. $SASPA_{full}$ uses the full local covariance matrix in (d), significantly improving the approximation quality along the circle.

and compare its predictive performance with alternative sparse GP methods. We use the RBF kernels for all the experiments. To all the sparse GP models, we use Kmeans to choose basis centers and define the covariance matrix c_i based on the data in each cluster.

We first examine the performance of the new method on a toy regression problem. Since we can control the generative process of the synthetic data, it is easier for us to gain insight into how the method performs. For regression, we sample 100 data points along a circle with some additive Gaussian noise. The output in different quadrant has different values plus certain additive Gaussian noise.

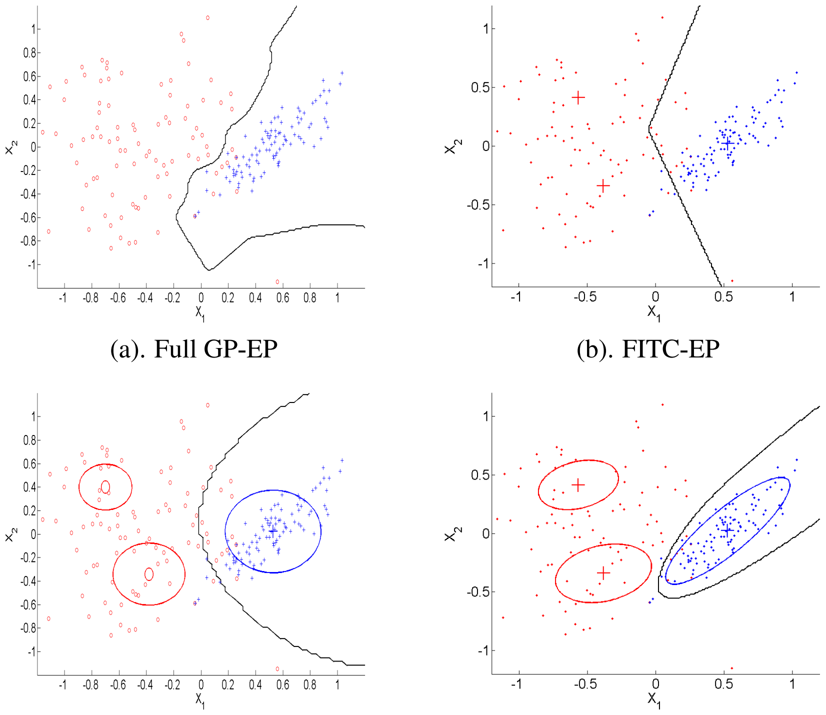
The mean of the exact posterior distribution of the GP is shown in figure 1a, with approximations in the other panels. Each approximation used the same four basis points, chosen via K-means clustering. Note that when the local covariance matrices become sphere matrices, our method reduces to the multiscale method of (Walder et al., 2008). As shown in the figure, the use of the full local covariance matrix improves the approximation quality.

The next test is a synthetic classification task. Each class is sampled from a multivariate Gaussian distribution. Figure 2 shows an example dataset, with the decision boundaries and the basis points used by each algorithm. Full-GP-EP uses all the training data samples as the basis points in an EP approximation, as described in (Minka, 2001a). FITC-EP uses FITC approximation in an EP framework (Naish-Guzman & Holden, 2007; Csató & Opper, 2000), implemented as a special case of SASPA with no blurring. Finally we have SASPA with sphere and full local covariance matrices for the blurring function $\phi(\mathbf{x}|b_i)$. Quantitative results are shown in figure 4, where we repeatedly sample 200 points for training and 2000 for testing. SOGP (Csató, 2002) corresponds to the DTC approximation applied to the same basis points as the other algorithms. The basis points are chosen by K-means in each case; further improvement is possible by optimizing these. The KL divergence to Full-GP-EP was computed as $\sum_i \mathrm{KL}(p(y_i|x_i) \mid\mid q(y_i|x_i))$ where x_i is a test point and $p(y_i|x_i)$ is the predictive distribution from Full-GP-EP and $q(y_i|x_i)$ is the predictive distribution from the sparse algorithm.

Finally we tested our methods on two standard UCI datasets, Ionosphere and Heart. The results are summarized in figures 4 and 5. Here we include the Informative Relevance Machine (IVM) (Lawrence et al., 2002), which applies one iteration of EP to a subset of the data. We see that basis points chosen by K-means are competitive with the ones chosen by IVM.

7. Conclusions

In this paper, we present a new perspective to understand sparse GP methods using the expectation propagation framework and develop new inference methods based on this perspective. Empirical results demonstrate improved approximation quality and prediction accuracy with the new extensions.



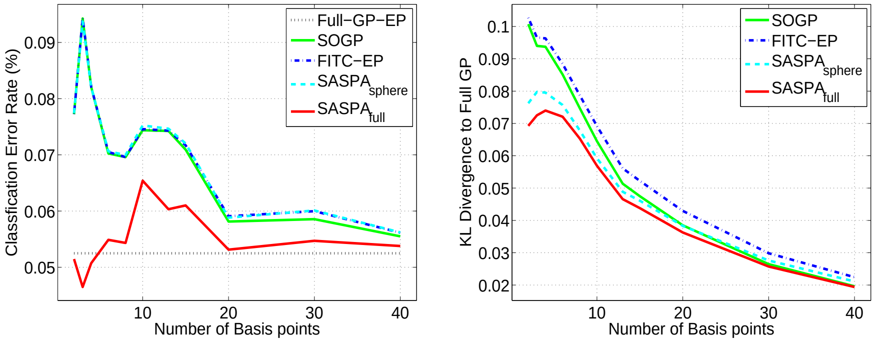
(a). Full GP-EP

(b). FITC-EP

(c). $SASPA_{sphere}$

(d). $SASPA_{full}$

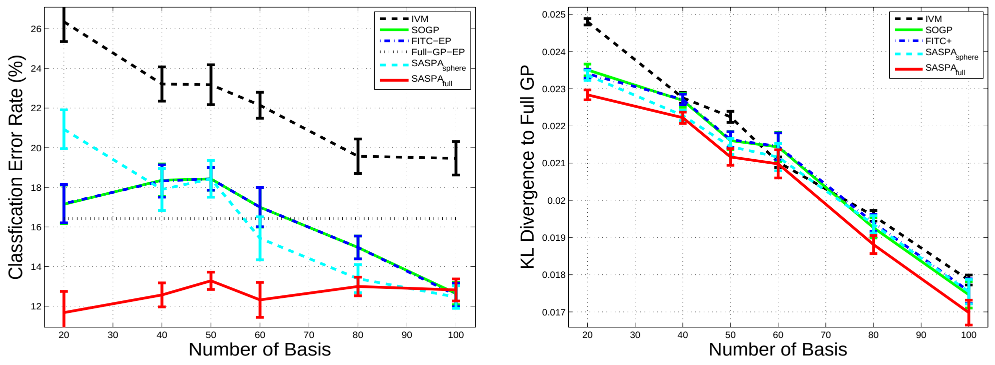
Figure 2. Classification on synthetic data. The blue and red ellipses show the standard deviation of local covariances for SASPA. The black curve is the decision boundary. With only three basis points, the true, complex decision boundary in (a) is well approximated by an ellipse by our method (d).



(a). Classification error

(b). Approximation error

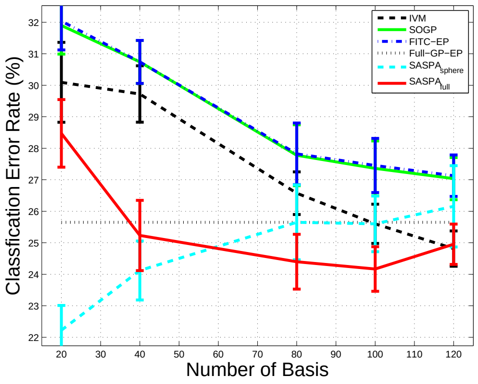
Figure 3. Effect of different approximation families in SASPA. The results are averaged over 20 random datasets. Note that all sparse GP algorithms used the same basis point locations. Thus we are emphasizing how well each algorithm makes use of its basis points.



(a). Classification error

(b). Approximation error

Figure 4. Classification on UCI dataset Ionosphere. The results are averaged over 20 random splits of the dataset. Note that all sparse GP algorithms, except IVM, used the same basis point locations.



(a). Results on Heart dataset

Figure 5. Classification on UCI benchmark dataset Heart. Note that all sparse GP algorithms, except IVM, used the same basis point locations. Thus we are emphasizing how well each algorithm makes use of its basis points.

References

Chu, W., Sindhwani, S., Ghahramani, Z., and Keerthi, S. S. Relational learning with Gaussian processes. In *Advances in Neural Information Processing Systems 18*, 2006.

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

Csató, Lehel and Opper, Manfred. Sparse representation for Gaussian process models. In *Advances in Neural Information Processing Systems 13*, pp. 444–450. MIT Press, 2000.

Deisenroth, M. P., Rasmussen, C. E., and Peters, J. Gaussian process dynamic programming. *Neurocomputing*, 72(7–9):1508–1524, 2009.

Lawrence, Neil, Seeger, Matthias, and Herbrich, Ralf. Fast sparse Gaussian process methods: The informative vector machine. In *Advances in Neural Information Processing Systems 15*, pp. 609–616. MIT Press, 2002.

Lázaro-Gredilla, Miguel and Figueiras-Vidal, Aníbal. Inter-domain Gaussian processes for sparse inference using inducing features. In *Advances in Neural Information Processing Systems 21*, 2009.

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

Minka, Thomas P. Expectation propagation for approximate Bayesian inference. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pp. 362–369, 2001b.

Naish-Guzman, Andrew and Holden, Sean. The generalized FITC approximation. In *Advances in Neural Information Processing Systems 19*, 2007.

Quiñonero-Candela, J. and Rasmussen, C. E. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1935–1959, 12 2005.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. The MIT Press, December 2006. ISBN 026218253X.

Snelson, Edward and Ghahramani, Zoubin. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18*, pp. 1257–1264. MIT press, 2006.

Teh, Y. W., Seeger, M., and Jordan, M. I. Semiparametric latent factor models. In *The 8th Conference on Artificial Intelligence and Statistics (AISTATS)*, 2005.

Titsias, Michalis K. Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, 2009.

Walder, Christian, Kim, Kwang In, and Schölkopf, Bernhard. Sparse multiscale Gaussian process regression. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pp. 1112–1119, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: http://doi.acm.org/10.1145/1390156.1390296.