Sequential Learning in GPs with Memory and Bayesian Leverage Score

Abstract

Limited access to previous data is challenging when using Gaussian process (GP) models for sequential learning. This results in inaccuracies in posterior, hyperparameter learning, and inducing variables. The recently proposed 'dual' sparse GP model enables inference of variational parameters in such a setup. In this paper, using the dual GP, we tackle the problem arising due to a lack of access to previous data for estimating hyperparameters of a sparse Gaussian process. We propose utilizing the concept of 'memory'. To pick representative memory, we develop the 'Bayesian leverage score' built on the ridge leverage score. We experiment and perform an ablation study with a sequential learning data set, *split MNIST*, to showcase the usefulness of the proposed method.

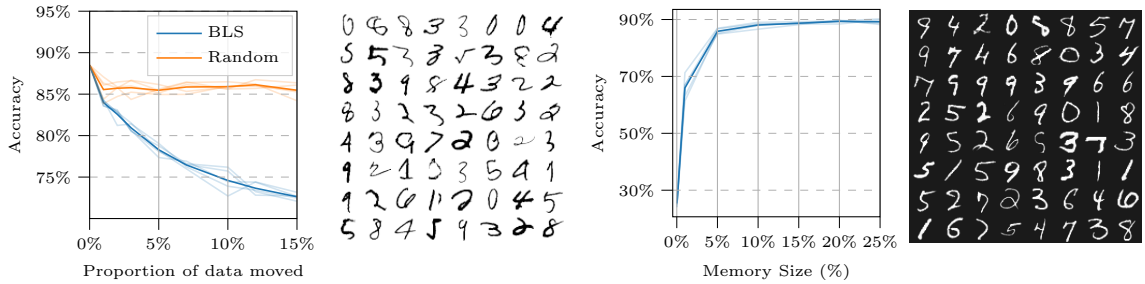
Keywords: Sequential learning, sparse Gaussian process, online learning

1. Introduction

Uncertainty quantification is essential in sequential decision making problems where uncertainty is essential for exploration, *i.e.* model based reinforcement learning or Bayesian optimization. Gaussian process (GP) models are a popular modelling technique for such problems. However, performing exact online inference in GP models requires access to all the data, which is infeasible when data grows. Sparse GP methods are typically used for reducing complexity, but also they assume full access to the training data. These methods are infeasible in the sequential learning setting where access to past data is limited and hyperparameters cannot be learned with access to full data. There have been various techniques proposed for these problems. Csató and Opper (2002) use expectation propagation for posterior inference and a projection method to obtain a sparse representation, but cannot estimate hyperparameters. Bui et al. (2017) extend to hyperparameters and generalize to α-divergence, but perform slow gradient based optimization for non-conjugate likelihoods.

In this paper, we show that the recently proposed 'dual' SVGP method of Adam et al. (2021) is a powerful backbone for the sequential setting. To tackle the inaccuracy arising due to forgetting previous tasks, we propose to add 'memory' of past data helping with learning. Furthermore, we show that the dual parameters naturally define a new metric called Bayesian leverage score (BLS), a generalisation of ridge leverage score, that can be used for picking memory. The proposed use of memory is in a similar spirit to recent approaches for continual deep learning (Nguyen et al., 2018; Titsias et al., 2020; Pan et al., 2020), but here the memory selection is done via an extension of the traditional leverage score (Alaoui and Mahoney, 2015).

© 2022 .



(a) Accuracy vs. representation (b) Points with highest BLS (c) Accuracy vs. memory size (d) Points from memory

Figure 1: Ablation study on *split MNIST*. (a) Evolution of test accuracy of the model conditioned on the training set as we move data from training set to test set on the basis of BLS vs. randomly. (b) Digits from train-set with highest BLS. (c) Evolution of test accuracy as the memory size is increased. (d) Points from memory.

2. Methods

Gaussian process models define a prior over functions $f \sim \mathcal{GP}(0, \kappa)$ that is characterized by a covariance kernel function $\kappa_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}')$, where $\boldsymbol{\theta}$ are hyper parameters associated with the kernel. Given a set of data $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, we define a likelihood $p(y_i \,|\, \mathbf{f}_i)$ where $\mathbf{f}_i = \mathbf{f}(\mathbf{x}_i)$. One of the prominent methods to reduce complexity and deal with non-Gaussian likelihoods is the sparse variational GP (SVGP) where the function is defined on a set of m inducing variables $\mathbf{Z}$, where $m \ll n$ and the approximate posterior is defined as $q(\mathbf{m}, \mathbf{V})$. The induced posterior of the function is: $q_{\mathbf{u}}(\mathbf{f}) = \mathrm{N}(\mathbf{f}; \mathbf{A}\mathbf{m}^*, \mathbf{K}_{\mathbf{xx}} - \mathbf{A}\mathbf{K}_{\mathbf{zz}}^{-1}\mathbf{A}^\top + \mathbf{A}\mathbf{V}^*\mathbf{A}^\top)$, where $\mathbf{K}_{\mathbf{xx}}$ is an $n \times n$ matrix with $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ as the ij^{th} entry, $\mathbf{A} = \mathbf{K}_{\mathbf{xz}}\mathbf{K}_{\mathbf{zz}}^{-1}$. $\mathbf{K}_{\mathbf{xz}}$ and $\mathbf{K}_{\mathbf{zz}}$ are defined similarly to $\mathbf{K}_{\mathbf{xx}}$. Adam et al. (2021) showed that the optimal variational parameters $(\mathbf{m}^*, \mathbf{V}^*)$ are:

$$\mathbf{m}^* \equiv \mathbf{V}^*\boldsymbol{\lambda}^* \quad \text{and} \quad \mathbf{V}^* \equiv [\mathbf{K}_{\mathbf{zz}}^{-1} + \boldsymbol{\Lambda}^*]^{-1}, \tag{1}$$

where the dual parameters $(\boldsymbol{\lambda}^*, \boldsymbol{\Lambda}^*)$ are obtained using natural gradient updates which leads to faster and better convergence (Chang et al., 2020; Khan and Lin, 2017). In a sequential learning setup, the model does not have access to all the data, $\mathcal{D} = \{\mathcal{D}_{\text{old}}, \mathcal{D}_{\text{new}}\}$. The dual framework in Adam et al. (2021) can be adapted to sequential learning of $\boldsymbol{\lambda}^*$ and $\boldsymbol{\Lambda}^*$ by performing variational updates just on $\mathcal{D}_{\text{new}}$, leading to $\boldsymbol{\lambda}^*_{\text{new}}$ and $\boldsymbol{\Lambda}^*_{\text{new}}$. Given the dual parameters are natural parameters of the approximate likelihood terms, we can sum $\boldsymbol{\lambda}^*_{\text{new}} + \boldsymbol{\lambda}^*_{\text{old}}$ to obtain an estimate of $\boldsymbol{\lambda}^*$ and same for $\boldsymbol{\Lambda}^*$. In practice the updates work well and is the variational approach to Csató and Opper (2002). In this paper, we are interested in learning hyperparameters $\boldsymbol{\theta}$ of kernel and likelihood under the dual framework. We do this by employing memory, this is contrary to existing methods which involves extra KL regularization terms Bui et al. (2017).

Dual-SVGP with Memory: The idea is to reconstruct the ELBO as if we had access to all the data but now only have $\mathcal{M} \subset \mathcal{D}_{\text{old}}$. Fortunately, the dual SVGP elbo is already in a separated form. Meaning that the posterior and hyperparameters are separated, leading to a better bound for hyperparameter learning Adam et al. (2021). Our sequential updates

of $\boldsymbol{\lambda}^*$ and $\boldsymbol{\Lambda}^*$ also means we have an approximate KL calculation based on the pseudo data conversion of the dual parameters highlighted in Eq. (2). We do however not have access to the variational expectation term for the $\mathcal{D}_{\text{old}}$, we therefore use the memory $\mathcal{M}$ to approximate this term as follows:

$$\sum_{i\in\mathcal{D}_{\text{new}}} \mathbb{E}_{q_{\mathbf{u}}(f_i)}[\log p(y_i \mid f_i)] + S\sum_{i\in\mathcal{M}} \mathbb{E}_{q_{\mathbf{u}}(f_i)}[\log p(y_i \mid f_i)] + \underbrace{\log \mathcal{Z} - \mathbb{E}_{q(\mathbf{u})}[\log \mathrm{N}(\tilde{\mathbf{y}} \mid \mathbf{u}, \tilde{\boldsymbol{\Sigma}})]}_{\mathbb{D}_{KL}[q(\mathbf{u}) \,\|\, p_{\boldsymbol{\theta}}(\mathbf{u})]}, \tag{2}$$

where $\tilde{\mathbf{y}} = \mathbf{K}_{\mathbf{zz}}\boldsymbol{\Lambda}^{-1}\boldsymbol{\lambda}$, $\tilde{\boldsymbol{\Sigma}} = \mathbf{K}_{\mathbf{zz}}\boldsymbol{\Lambda}^{-1}\mathbf{K}_{\mathbf{zz}}$, $\log \mathcal{Z} = -\frac{1}{2}\log|\tilde{\boldsymbol{\Sigma}} + \mathbf{K}_{\mathbf{zz}}| - \frac{1}{2}\tilde{\mathbf{y}}^\top\left[\tilde{\boldsymbol{\Sigma}} + \mathbf{K}_{\mathbf{zz}}\right]^{-1}\tilde{\mathbf{y}} + \mathrm{c}$, and $S = \frac{n_{\text{old}}}{n_{\mathcal{M}}}$, n_{old} is the number of data points seen in previous batches, and $n_{\mathcal{M}}$ is the number of data points stored in memory. We use this as an objective for hyperparameter learning. However, in the non-stationary continual learning problems, we are also interested in optimizing inducing variable $\mathbf{Z}$. Gradient based methods fail for this and therefore we resort to *pivoted* Cholesky (Fine and Scheinberg, 2001) technique shown to do well in the online setting in Maddox et al. (2021) and offline setting in Burt et al. (2020).

Bayesian Leverage Score (BLS): The memory point selection is therefore key to our method being able to learn appropriate hyper parameters. Our belief here is that memory should be selected such that it is representative of the data distribution. We build on the ridge leverage score (RLS, Alaoui and Mahoney, 2015), which is used for selecting an effective subset for 'sketching' in kernel ridge regression. We extend it to the Bayesian, non-conjugate case, and use this to define a distribution over input points from which we sample the memory points.

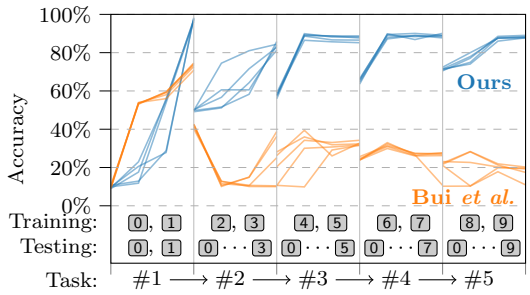
The ridge leverage score is defined as the diagonal of $\mathbf{K}_{\mathbf{xx}}(\mathbf{K}_{\mathbf{xx}} + \lambda\mathbf{I})^{-1}$, where λ is the ridge parameter of kernel ridge regression (Alaoui and Mahoney, 2015). By representing the dual parameters as $(\alpha_i \coloneqq \mathbb{E}_{p(f_i \mid \mathbf{y}_n)}[\nabla_{f_i} \log p(y_i \mid f_i)]$, $\beta_i \coloneqq \mathbb{E}_{p(f_i \mid \mathbf{y}_n)}[\nabla^2_{f_i} \log p(y_i \mid f_i)])$, we can write the posterior of an approximate GP regression model as:

$$q_*(\mathbf{f} \mid \mathbf{m}^*_{\mathbf{f}}, \mathbf{V}^*_{\mathbf{f}}) \propto \mathrm{N}(\mathbf{f} \mid \mathbf{0}, \mathbf{K}_{\mathbf{xx}}) \prod_{i=1}^n \mathrm{e}^{-\frac{1}{2}\beta_i^*(\tilde{y}_i - f_i)^2}, \tag{3}$$

where $\tilde{y}_i \coloneqq \alpha_i^*/\beta_i^* + m_{f,i}$ and $m_{f,i}$ is the prediction mean at x_i. An interpretation for the role of β_i^* as a local curvature: it is the noise precision for the Gaussian approximation of the ith likelihood. Following this interpretation, we define a leverage score for our VI setting with respect to the *approximate* GP regression model Eq. (3). We refer to this as the *Bayesian* leverage score (BLS):

$$h_i^{\text{bls}} \coloneqq \left[\mathbf{K}_{\mathbf{xx}}(\mathbf{K}_{\mathbf{xx}} + \mathrm{diag}(\boldsymbol{\beta}_*^{-1}))^{-1}\right]_{ii} = \beta_i^* v_{f,i,i}^*. \tag{4}$$

This score measures the sensitivity of the predictions with respect to each data example. Its interpretation is similar to that of RLS: high-leverage inputs are those that are difficult to predict (high predictive variance $v_{*,i}$) but carry a large amount of signal (low noise variance $1/\beta_{*,i}$). For Gaussian likelihoods, BLS reduces to RLS, but it extends the applicability to non-Gaussian likelihoods. Intuitively, the BLS score indicates how difficult the example was for the model, similar to the RLS score; we illustrate this for MNIST digits in Fig. 1(a). In Eq. (4) we detail the full GP calculation which is infeasible for large data sets, however we can swap the β_i^* and $v_{f,i,i}^*$ for their sparse GP counterparts greatly reducing the computational complexity and making the BLS feasible for large datasets.



Task	0, 1	2, 3	4, 5	6, 7	8, 9
#1	.95(.020)				
#2	.74(.039)	.95(.005)			
#3	.83(.052)	.85(.013)	.96(.003)		
#4	.78(.014)	.86(.012)	.92(.008)	.96(.003)	
#5	.94(.007)	.84(.009)	.82(.022)	.92(.015)	.87(.004)

Figure 2: (a) Progression of overall accuracy on *split MNIST*. Training starts with 0 vs. 1 and each task introduces new digits while testing on all classes thus far. The overall accuracy drops when introducing a new task, but recovers and does not suffer from forgetting. (b) Test accuracy on *split MNIST* over all tasks thus far.

3. Experiments

Split MNIST (Zenke et al., 2017) is a continual learning data set and a variant of MNIST where training data comes in five batches of two digits each. Performance is measured by multi-class classification accuracy on all digits seen thus far. The model at each step has access only to the current batch of the classification task and thus should learn incrementally on different tasks without forgetting about the previous ones. Previous work (Bui et al., 2017)fails (Fig. 2) in such a setup as it forgets about the previous tasks (Bui et al. (2017). Our proposed model is able to retain information about previous tasks with only a marginal drop in accuracy when the new task is introduced (*cf.* Fig. 2).

We show that BLS is a valid metric as it determines examples of difficulty. We perform the same continual learning experiment on *split MNIST*, but this time we select data points from our training set and move them to the test set. Points are chosen either randomly or based on the BLS score. We then retrain the model on the reduced training set and test on the increased testing set. Randomly selecting has a small negative effect on performance. However, using the BLS score is detrimental, showing the importance of the examples for the model that are moved to the test set. Fig. 1(*b*) shows digits with the highest BLS score. Another ablation study showcases the size of memory needed and how it affects the model accuracy. We train our sequential model with different memory sizes using BLS and report test accuracy (Fig. 1(*c*)). As expected, the accuracy increases with memory size, but remarkably a memory size of just 5% achieves optimal performance. Fig. 1(*d*) shows samples from memory after the model has observed all the tasks.

4. Conclusion

The lack of access to previous data for a sequential sparse GP model is problematic for hyperparameter learning. In this paper, we solve this problem by introducing the concept of memory. This approach is novel and different from previous work, which attempts to replace missing data with additional regularization terms in the ELBO (Bui et al., 2017). We further derive a novel Bayesian leverage score, for selecting memory, and show its usefulness in a sequential learning data set, split MNIST.