Stochastic Variational Inference for Gaussian Process Latent Variable Models using Back Constraints

Thang D. Bui **Richard E. Turner**
tdb40@cam.ac.uk ret26@cam.ac.uk
Computational and Biological Learning Lab, Department of Engineering
University of Cambridge, Trumpington Street, Cambridge, CB2 1PZ, UK

Abstract

Gaussian process latent variable models (GPLVMs) are a probabilistic approach to modelling data that employs Gaussian process mapping from latent variables to observations. This paper revisits a recently proposed variational inference technique for GPLVMs and methodologically analyses the optimality and different parameterisations of the variational approximation. We investigate a structured variational distribution, that maintains information about the dependencies between hidden dimensions, and propose a mini-batch based stochastic training procedure, enabling more scalable training algorithm. This is achieved by using variational recognition models (also known as back constraints) to parameterise the variational approximation. We demonstrate the validity of our approach on a set of unsupervised learning tasks for texture images and handwritten digits.

1 Introduction

Gaussian Processes (GPs) are flexible nonparametric distributions over continuous, nonlinear functions that can be used in both supervised and unsupervised settings [1]. When the input variables of a GP are themselves treated as stochastic, the resulting model is a Gaussian Process Latent Variable Model (GPLVM). Exact Bayesian inference and learning in GPLVMs, like many other generative models, are however analytically intractable. Crucially, analytically tractable approximate Bayesian schemes based on variational methods have been developed for GPLVMs [2]. However, even when such approximate solution exists, three challenging problems remain. First, the training algorithm proposed for the GPLVMs in [2] needs to inspect all datapoints for each training step, leading to an expensive procedure that does not scale. In addition, to obtain the latent representation of new datapoints at test time, a lengthy optimisation is required. Second, this work uses a simple factorial variational approximations that cause over-pruning of latent dimensions [3]. Third, this model class intrinsically preserves the dissimilarity between observations and does not preserve the so-called local distance between datapoints as discussed in [4]. A good unsupervised representation learning algorithm arguably needs to balance between these two preservations.

In this paper, we revisit the variational free-energy approximation proposed in [2] and seek a solution that is more robust to dimension pruning, can scale for large datasets and can learn to preserve local structures. We achieve these targets by, 1. choosing a structured variational approximation that helps the model to couple the hidden dimensions more effectively, 2. proposing a minibatch based stochastic variational training procedure that can scale to large datasets, 3. using back constraints (aka recognition models) to both preserve class neighbourhood structures in the latent space and significantly accelerate test time. We first review GPs and GPLVMs in section 2, followed by a discussion on the variational approximation and how to perform stochastic training using minibatches and incorporate back constraints in section 3, and to finish, in section 4, demonstrate our approach in two unsupervised learning tasks.

2 Review of Gaussian processes and Gaussian process latent variable models

This section provides a concise introduction to GPLVMs [1, 5]. Suppose we have a training set comprising N D-dimensional real valued observations $\mathbf{Y} \equiv \{\mathbf{y}_n\}_{n=1}^{N}$. The corresponding Q-dimensional input vectors $\mathbf{X} \equiv \{\mathbf{x}_n\}_{n=1}^{N}$ are assumed to be latent and stochastic. The probabilistic model describing the data can be written as follows,

$$p(\mathbf{X}) = \prod_n \mathcal{N}(\mathbf{x}_n; \mathbf{0}, \mathrm{I}_Q), \text{ if unsupervised} \tag{1}$$

$$p(f_{1:D}) = \prod_d \mathcal{GP}(f_d; \mathbf{0}, \mathbf{K}_d) \tag{2}$$

$$p(\mathbf{Y}|f_{1:D}, \mathbf{X}, \sigma_y^2) = \prod_{n,d} \mathcal{N}(y_{n,d}; f_d(\mathbf{x}_n), \sigma_y^2) \tag{3}$$

The model assumes that an (uncountably) infinite dimensional vector f_d is first drawn from a GP, each observation $y_{n,d}$ is then formed by taking a value from this vector, indexed at latent input $\mathbf{x}_n$ and corrupted by independent Gaussian noise. The GP is assumed to have a zero mean and a covariance matrix $\mathbf{K}_d$ that has an infinite number of rows and columns; this covariance matrix is fully specified by a covariance function or kernel which depends upon a small number of hyperparameters.

3 Variational inference and learning

3.1 Variational formulation

Exact Bayesian learning and inference in GPLVMs involves finding a marginal likelihood of the hyperparameters that can be optimised, and the posterior distributions over all unknown quantities which include functions $\{f_d\}_{d=1}^{D}$ and latent variables $\mathbf{X}$. Unfortunately, both the marginal likelihood and the posteriors are not analytically tractable, and thus we turn to a deterministic approximation technique called variational free-energy approximation. We first summarise the approach proposed in [2] and its practical limitations. The intractability in marginalisation is sidestepped by introducing a variational distribution $q(f_{1:D}, \mathbf{x}_{1:N})$ that approximates the true posterior, to yield the variational free energy which is a lower bound to the log marginal likelihood:

$$\mathcal{F}(q(.)) = \int_{f_{1:D},\mathbf{X}} q(f_{1:D}, \mathbf{X}) \log \frac{p(\mathbf{y}_{1:N}, f_{1:D}, \mathbf{X})|\theta}{q(f_{1:D}, \mathbf{X})} \leq \log p(\mathbf{y}_{1:N}|\theta). \tag{4}$$

We wish to optimise this quantity wrt q and θ, however it is only analytically tractable when the variational distribution is judiciously picked; here we use one that factorises between the latent variables and the latent functions,

$$q(f, \mathbf{X}) = q(f_{1:D})q(\mathbf{X}) = \left[q(u_{1:D}) \prod_d p(f_d \neq u_d | u_d) \right] q(\mathbf{X}), \tag{5}$$

where u_d is a small subset of the infinite dimensional vector f_d: $u_d = f_d(\mathbf{Z})$, and $f_d \neq u_d$ are the remaining elements. The indices $\mathbf{Z}$ of u_d are variational parameters that can be tuned so that the variational approximation is *closest* to the true posterior [6]. Critically, the prior over f_d can be exactly rewritten as $p(f_d) = p(u_d)p(f_d \neq u_d|u_d)$, leading to the cancellation of the difficult term $p(f_d \neq u_d|u_d)$ in the energy, and as a result,

$$\mathcal{F}(q(.)) = -\mathrm{KL}(q(\mathbf{X})||p(\mathbf{X})) - \mathrm{KL}(q(u_{1:D})||p(u_{1:D})) + \sum_{n,d} \left\langle \log p(y_{n,d}|f_d, \mathbf{x}_n, \sigma_y^2) \right\rangle_{q(.)} \tag{6}$$

Optimal variational distribution for $\mathbf{U}$ As the prior over $\mathbf{U}$ comprises mutually independent Gaussian distributions across layers and dimensions, and the expectations of the log likelihood in eq. (6) follow the same factorisation, the optimal form for $q(\mathbf{U})$ is also factorised, i.e. $q(u_{1:D}) = \prod_d q(u_d)$. Crucially, we can find this optimal form analytically in which each $q(u_d)$ is a multivariate Gaussian distribution whose natural parameters are,

$$\eta_{1,d} = \frac{1}{\sigma_y^2} \mathbf{K}_{zz}^{-1} \sum_{n=1}^{N} \left\langle \mathbf{K}_{z,\mathbf{x}_n} \right\rangle_{q(\mathbf{x}_n)} y_{n,d}, \quad \eta_2 = -\frac{1}{2}(\mathbf{K}_{zz}^{-1} + \frac{1}{\sigma_y^2} \mathbf{K}_{zz}^{-1} \sum_{n=1}^{N} \left\langle \mathbf{K}_{z,\mathbf{x}_n} \mathbf{K}_{z,\mathbf{x}_n}^{\intercal} \right\rangle_{q(\mathbf{x}_n)} \mathbf{K}_{zz}^{-1}) \tag{7}$$

Both optimal parameters above involve summing expectations of covariance matrices wrt $q(\mathbf{X})$, over N training datapoints, resulting in an expensive computation step when the number of training points is large. Furthermore, these expectations are only analytically tractable when the kernels and $q(\mathbf{X})$ take some special forms, for example, exponentiated quadratic kernels and a Gaussian distribution over $\mathbf{X}$ respectively.

Variational distribution for $\mathbf{X}$ By differentiating the lower bound wrt $q(\mathbf{X})$ and setting it to zero, we can derive the optimal form of $q(\mathbf{X})$ in a similar fashion as $q(\mathbf{U})$, which gives a distribution factorised across datapoints: $q(\mathbf{X}) = \prod_n q(\mathbf{x}_n)$ where each $q(\mathbf{x}_n)$ is non-Gaussian. To this end, we approximate it by a Gaussian density. In particular, the approach proposed in [2] posits a mean field approximation that assumes the latent dimensions are factorised, $q(\mathbf{x}_n) = \prod_Q q(x_{n,q})$. This approximation in practice leads to dimension over-pruning, that is it assigns zero inverse lengthscales to most of the latent dimensions, making them insignificant. In this paper, we use a structured approximation that does not assume this factorisation; this can be done by parameterising the full covariance matrix of $q(\mathbf{x}_n)$ by its Cholesky factor, $\Sigma_n = \mathrm{R}_n^\intercal \mathrm{R}_n$, as opposed to a diagonal covariance matrix in the mean field case.

3.2 Stochastic optimisation

The inducing point based variational approximation for GPLVMs presented above introduces two sets of variables parameterising $q(\mathbf{U})$ and $q(\mathbf{X})$, which can be loosely categorised as global and local variables respectively. Furthermore, the optimal natural parameters of the variational distribution over the global parameters in eq. (7) are formed from moments wrt the variational distributions over local variables. As such, minibatch based stochastic variational inference (SVI) [7] can be applied. The key idea of the SVI framework is to apply coordinate ascent updates to local and global parameters using minibatches of data. At each iteration, the optimiser first looks at a minibatch $\{\mathbf{y}_n\}_1^{N_b}$ and updates $\{q(\mathbf{x}_n)\}_1^{N_b}$ by obtimising the free energy while keeping $q(\mathbf{U})$ fixed. Next, it takes a small noisy step towards the optimal global parameters of $q(\mathbf{U})$, using eq. (7) and $\{q(\mathbf{x}_n)\}_1^{N_b}$. Unfortunately, this optimisation scheme for GPLVMs is impractical for datasets of even modest size as it can take a long time to converge. Specifically, each iteration only updates the local parameters for datapoints in a minibatch and ignores the optimal $q(\mathbf{x})$ previously found for other datapoints. We sidestep this problem by sharing variational parameters between datapoints using variational recognition models or back constraints.

3.3 Using back-constraints to parameterise $q(\mathbf{X})$

Having described the stochastic procedure above and its impracticalities, we borrow ideas from [4,8, 9] and parameterise the mean and covariance of the variational distribution over $\mathbf{X}$ using recognition models or back constraints, i.e. $q(\mathbf{x}_n) = \mathcal{N}(\mathbf{x}_n; \mu_{\phi_1}(y_n), \mathrm{R}_{\phi_2}(y_n)^\intercal \mathrm{R}_{\phi_2}(y_n))$. Specifically, the mean and the Cholesky factor of the covariance can take any parametric form with the observations as inputs, here we use neural networks and ϕ_1 and ϕ_2 are the network weights. The use of back constraints was first suggested for GP models in [4] to obtain a better latent representations that preserve local structures. That is since the distribution over latent variables are parameterised by an inverse mapping from the observations, datapoints that are *close* or *similar* will have a *similar* latent representation. More importantly, the use of recognition models facilitates the sharing of variational parameters between datapoints, enabling efficient stochastic optimisation. In particular, updating the back-constraint parameters ϕ_1 and ϕ_2 using datapoints in a minibatch also changes the latent representations of other datapoints, as opposed to independent updates due to the lack of parameter sharing in section 3.2. Obtaining the latent variables at test time for new observations is very fast as computing μ_{ϕ_1} and R_{ϕ_2} can be done in $\mathcal{O}(1)$ time. The lower bound and its gradients wrt all parameters are still analytically tractable, for example the gradients wrt ϕ_1 and ϕ_2, $\frac{\partial \mathcal{F}(q)}{\partial \phi_1} = \frac{\partial \mathcal{F}(q)}{\partial \mu} \frac{\partial \mu}{\partial \phi_1}$ and $\frac{\partial \mathcal{F}(q)}{\partial \phi_2} = \frac{\partial \mathcal{F}(q)}{\partial \mathrm{R}} \frac{\partial \mathrm{R}}{\partial \phi_2}$ as the derivatives of the neural network outputs wrt the weights can be found using the back-propagation algorithm. To update the back constraint parameters, model hyperparameters and inducing point locations, we use stochastic gradient descent with the RMSProp heuristic [10].

4 Experiments

4.1 Modelling texture images

We create a dataset of 150 16x16 patches of textures from plaid and knit images in [11] and learn a latent representation using a GPLVM. We use 10 latent dimensions and the batch training objective for this experiment. For testing, we pick 150 other patches and set 150 pixels in each image to be missing and asked the models to fill in those missing pixels. The experiments were repeated using two parameterisations of the covariance matrix of $q(\mathbf{x})$, Cholesky and diagonal, corresponding to the structured and mean field approximations respectively. We report the median root mean squared errors (RMSE), negative log loss (NLL) and their standard errors for random and block missing pixels in table 1. The results show that the Cholesky parameterisation is performing significantly better as it can capture the correlations between the latent dimensions and as a result, gives better predictive uncertainty. The learnt hyperparameters also show evidence that the diagonal parameterisation over-prunes latent dimensions by assigning them very long lengthscales [3].

Method	Random missing		Block missing	
	median NLL	median RMSE	median NLL	median RMSE
GPLVM – chol	$\mathbf{2.95 \pm 0.08}$	$\mathbf{0.49 \pm 0.00}$	$\mathbf{2.74 \pm 0.08}$	$\mathbf{0.52 \pm 0.00}$
GPLVM – diag	3.18 ± 0.11	0.50 ± 0.00	2.97 ± 0.08	0.54 ± 0.00

Table 1: Imputation results on texture images for two network architectures and different parameterisations of the covariance of $q(\mathbf{X})$.

4.2 Modelling handwritten digit images

We test the proposed stochastic training procedure on an unsupervised learning of handwritten digit images, in order to determine how well the model would scale of a large dataset. We took 18,000 images of three classes 0, 3 and 7 from the training set of the MNIST dataset, and trained two models that have two and five hidden dimensions respectively. Previous applications of a GPLVM to this dataset only used 1000 examples of one class, due to the expensive training time in batch mode. We visualise the latent representation learnt by the architecture with two hidden dimensions in fig. 1 [left] and some samples generated from the architecture with five hidden dimensions in fig. 1 [right].

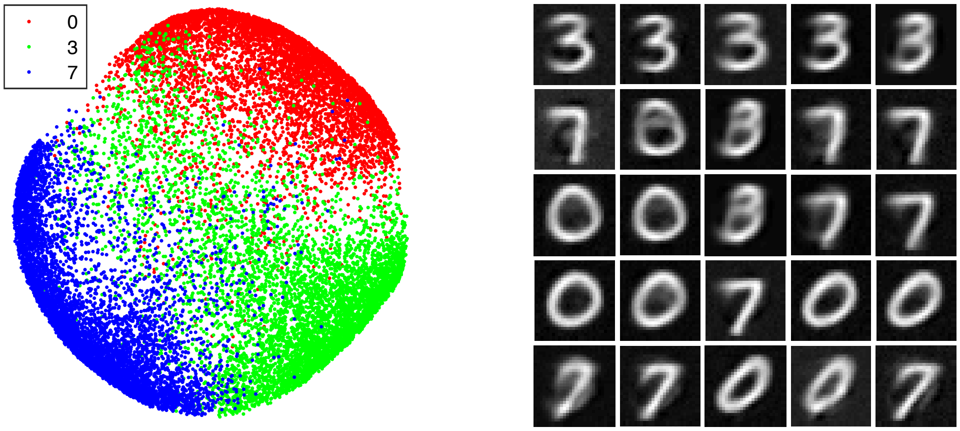

Figure 1: Left: Latent representations of three classes found using a GPLVM with 2 latent dimensions. Right: Samples generated from a similar model that has 5 latent dimensions .

5 Summary and future work

We have revisited a variational free energy approach for GPLVMs and introduced a minibatch based stochastic training algorithm that can scale well with number of datapoints. We demonstrated the validity of the proposed method on two datasets of texture and handwritten digit images. Our work is an extension of [12] to GP models with latent variables, and a complementary perspective to a distributed training procedure introduced in [13], making further progress in scaling up inference and learning in GP models.

Ongoing work includes extending our approach to generative models using deep Gaussian processes and Gaussian process state space models, quantifying the quality of the learnt generative models with and without recognition models, and comparing our approach to more recent neural network based generative models.

Acknowledgements

TB thanks Google for funding his European Doctoral Fellowship. RET thanks EPSRC grants EP/G050821/1 and EP/L000776/1.

References

[1] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[2] M. K. Titsias and N. D. Lawrence, "Bayesian Gaussian process latent variable model," in *13th International Conference on Artificial Intelligence and Statistics*, pp. 844–851, 2010.

[3] R. E. Turner and M. Sahani, "Two problems with variational expectation maximisation for time-series models," in *Bayesian Time series models* (D. Barber, T. Cemgil, and S. Chiappa, eds.), ch. 5, pp. 109–130, Cambridge University Press, 2011.

[4] N. D. Lawrence and J. Quiñonero-Candela, "Local distance preservation in the GP-LVM through back constraints," in *23rd International Conference on Machine Learning*, pp. 513–520, ACM, 2006.

[5] N. D. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," in *Advances in Neural Information Processing Systems 16*, pp. 329–336, MIT Press, 2004.

[6] M. K. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in *12th International Conference on Artificial Intelligence and Statistics*, pp. 567–574, 2009.

[7] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.

[8] P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel, "The Helmholtz machine," *Neural computation*, vol. 7, no. 5, pp. 889–904, 1995.

[9] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *31th International Conference on Machine Learning*, pp. 1278–1286, 2014.

[10] G. Hinton, "Rmsprop: Divide the gradient by a running average of its recent magnitude,," 2012.

[11] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1265–1278, Aug 2005.

[12] J. Hensman, N. Fusi, and N. D. Lawrence, "Gaussian processes for big data," in *29th Conference on Uncertainty in Artificial Intelligence*, pp. 282–290, 2013.

[13] Y. Gal, M. van der Wilk, and C. Rasmussen, "Distributed variational inference in sparse Gaussian process regression and latent variable models," in *Advances in Neural Information Processing Systems 27*, pp. 3257–3265, 2014.