Stochastic Variational Inference for Sparse Gaussian Processes

Alexandre Piche
260478404

April 22, 2016

Coming from the stochastic process literature, Gaussian processes (GPs) (Rasmussen, 2006) have taken an important place in machine learning in the recent years for inference on functions. They have the advantages of Bayesian non-parametric methods, such as being able to grow in complexity with the amount of data, an intuitive way of handling uncertainty, and the possibility of performing exact inference. However, it is known to be difficult to use exact inference for GPs on large datasets. Specifically due to the matrix inversion, exact inference in GPs require computation of complexity $O(n^3)$. Fortunately, methods such as variational inference (Wainwright and Jordan, 2008), an approximate inference algorithm, can be used for GPs. We will investigate Hensman et al. (2013)'s implementation of stochastic variational inference algorithms for GPs, in "GPy" package, which is based on Hoffman et al. (2013). More specifically, we will analyze the method in terms of speed and accuracy performance compared to exact inference algorithm.

1 Introduction and Motivation

Gaussian processes (GPs) are one of the most important Bayesian non-parametric methods. They are simple and flexible techniques able to model complex functions, with the main disadvantage of being computationally expensive to optimize. Specifically, learning the parameters requires matrix inversion, which has complexity $O(n^3)$, making it difficult to fit a GP for large datasets.

First we will introduce GPs, a model that builds on the kernel methods seen in class, and how they can be more efficient using inducing inputs, a method known as sparse Gaussian processes. We will then adapt sparse GPs to be optimized by stochastic variational inference (SVI), an inference techniques that builds on variational inference (VI) seen in class.

2 Gaussian Processes

GPs are distributions over functions, it is thus natural to use them as a prior over an unknown function f, written as $f(x) \sim \mathcal{GP}(m(x), k(x, x'))$. A GP is completely defined by its mean: $m(x) = E[f(x)]$, and its covariance matrix: $k(x, x') = E[(f(x) - m(x))(f(x') - m(x'))]$, where k is usually a kernel function (Rasmussen, 2006).

We can learn the GP parameters: (σ^2, θ), by maximizing the probability of the observations y:

$$\begin{aligned} p(y) &= \int_f p(y|f)p(f)df \\ &= N(y|f, \sigma_n^2 I + K_{nn}) \\ p(y|f) &= \mathcal{N}(f, \sigma_n^2 I) \\ p(f) &= \mathcal{N}(0, K_{nn}) \end{aligned}$$

where $p(f)$ is our prior over the unknown function f, $p(y|f)$ is the probability of observing y given a function f, and θ are the kernel parameters. The GP posterior is given by:

$$f|y \sim \mathcal{GP}(K(\sigma^2 I + K)^{-1} y, K - K(\sigma^2 I + K)^{-1} K^T)$$

Computing it requires solving $(K + \sigma^2 I)v = y$, this operation involves matrix inversion, and thus has complexity $O(n^3)$. It is impracticable for dataset with more than a few thousands examples and more than 3 covariates Gelman et al. (2014).

2.1 Covariance Functions and Kernels

The covariance matrix K, has entry $K_{p,q} = k(x_p, x_q)$ where k is a kernel function. k controls the function smoothing and the deviation from the mean.

2.1.1 Gaussian Kernel

The most common kernel function used in practice is the Gaussian kernel (also called squared exponential) (Gelman, Carlin, Stern, and Rubin, 2014):

$$k(x, x') = \tau^2 \exp\left(-\frac{(x - x')}{2l^2} \right)$$

Notice that the covariance between two outputs is written as a function of their inputs. This kernel corresponds to a Bayesian linear regression with infinite number of basis function (Rasmussen, 2006).

2.1.2 Automatic Relevance Determination Kernel

Some features might be less important than others, they might even be irrelevant. In practice, automatic relevance determination (ARD) is used to select which features should be more important, the kernel is given by:

$$k(x, x') = \tau^2 \exp\left(-\sum_{d=1}^{D} \frac{(x_{(d)} - x'_{(d)})}{2l_d^2} \right)$$

Where D is the number of features, note that as $l_d^2 \to \infty$, the feature d becomes irrelevant. Note that it is equivalent to multiply d Gaussian kernels. ARD are able to learn any continuous function given enough data, under some conditions (Duvenaud, 2014).

2.1.3 Combining Kernels

Adding of multiplying two positive definite kernels results in a third positive definite kernel. Specifically, we can add GPs such that (Duvenaud, 2014):

$$\begin{aligned} f_a &\sim \mathcal{GP}(\mu_a, K_a) \\ f_b &\sim \mathcal{GP}(\mu_b, K_b) \\ f_a + f_b &\sim \mathcal{GP}(\mu_a + \mu_b, K_a + K_b) \end{aligned}$$

2.2 Properties of Gaussian Processes

For n prespecified points $x_1, \ldots, x_n$, the finite marginal distribution of a GP is defined as:

$$f(x_1), \ldots, f(x_n) \sim N(m(x_1), \ldots, m(x_n), K)$$

Note that the Gaussian distribution implies that by marginalizing $f(x_2), \ldots, f(x_n)$, $f(x_1) \sim N(m(x_1), \tau^2)$, thus the examination of the whole set of variables does not change the distribution of a subset of variables (Rasmussen, 2006).

2.3 Predictions

It is reasonable to assume that y are noisy observations from an unknown function f. We can denote y as $y = f(x) + \epsilon$, where $\epsilon \sim N(0, \sigma)$. The joint distribution of y and the unknown function evaluated at x_* is thus given by:

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right)$$

The covariance matrix K is evaluated at all pairs of training and testing points. We can condition the joint Normal on the observations to obtain:

$$\begin{aligned} f_*|X, y, X_* &\sim \mathcal{N}(\bar{f}_*, cov(f_*)), \text{ where} \\ \bar{f}_* &= K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} y, \\ cov(f_*) &= K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*) \end{aligned}$$

We can see $\bar{f}_*$ as a linear combination of the observations y, specifically it could have been written as $\bar{f}_* = \sum_{i=1}^{n} \alpha_i k(x_i, x_*)$, where $\alpha = (K + \sigma_n^2 I)^{-1} y$. The covariance is simply the prior variance $K(X_*, X_*)$ minus the information given by the observations.

2.4 Sparse Gaussian Processes

When dealing with large datasets, we can approximate the GP by using m inducing variables, where m is user specified and can be a lot smaller than n. The inducing inputs can be a subset of our sample or they can be pseudo-inputs lying in the same space as our sample (Titsias, 2009).

Specifically, we augment the prior $p(f)$ as $p(f|u)p(u)$ where u is the function values at the inducing variables $\{z\}_{i=1}^{m}$ such that:

$$\begin{aligned} p(y|f) &= N(y|f, \sigma_n^2 I) \\ p(f|u) &= N(f|K_{nm} K_{mm}^{-1} u, \tilde{K}) \\ p(u) &= N(u|0, K_{mm}) \end{aligned}$$

where K_{mm} is the covariance matrix between all pairs of inducing points z, and K_{nm} between all inducing points and the training points. The posterior covariance is given by $\tilde{K} = K_{nn} - K_{nm} K_{mm}^{-1} K_{mn}$ (Hensman, Fusi, and Lawrence, 2013). Learning the parameters of a GP has now complexity $O(nm^2)$ (Titsias, 2009).

3 Stochastic Variational Inference

In class, we explored variational inference (VI) as a technique to approximate the conditional distribution of latent variables z given observed variables x using optimization. Specifically how to approximate:

$$\begin{aligned} p(z|x) &= \frac{p(z, x)}{p(x)} \\ &= \frac{p(z, x)}{\int_z p(z, x)} \end{aligned}$$

where $\int_z p(z, x)$ is intractable, by minimizing:

$$q^*(z) \quad = \quad \arg\min_{q(z)\in\mathcal{Q}} KL(q(z)||p(z|x))$$

where $\mathcal{Q}$ is a family of distribution.

VI requires to go through the whole dataset at each iteration which is expensive for large datasets. Hoffman et al. (2013) proposed a stochastic version of VI: stochastic variational inference (SVI). SVI uses noisy but unbias gradient of batch data to approximate the conditional distribution more efficiently. For the stochastic optimization to converge to a local optimum, Robbins and Monro (1951) showed that the step size must satisfy:

$$\sum_t \epsilon_t = \infty \quad ; \quad \sum_t \epsilon_t^2 < \infty$$

SVI maximizes the evidence lower bound (ELBO) given by:

$$\mathcal{L}(\lambda) \quad = \quad E_q[\log p(\beta|x,z)] - E_q[\log q(\beta)] + c$$

where x are observations, z local hidden variables, and β are global hidden variables. Note the role of the global variables β.

4 Stochastic Variational Inference for Gaussian Processes

As noted above, SVI requires global variables, but GPs don't have any. We will use the variables u as global variables, and the variational distribution $q(u)$ to lower bound the quantity $p(y|X)$.

$$\begin{aligned} p(y|X) &= \int_u \int_f p(y|f)p(f|u)p(u)df du \\ &= \int_u p(y|u)p(u)du \end{aligned}$$

4.1 Global Variables

We first need to bound the conditional probability $p(y|u)$ using Jensen's inequality:

$$\begin{aligned} \log p(y|u) &= \log\langle p(y|f)\rangle_{p(f|u)} \\ &\geq \langle \log p(y|f)\rangle_{p(f|u)} = \mathcal{L}_1 \end{aligned}$$

Bringing the log into the expectation decrease the complexity from $O(n^3)$ to $O(m^3)$ (Hensman, Fusi, and Lawrence, 2013). Since $p(y|f) = \prod_{i=1}^{n} p(y_i|f_i)$, we can write:

$$\exp(\mathcal{L}_1) \quad = \quad \prod_{i=1}^{n} \mathcal{N}(y_i|\mu_i, \beta^{-1}) \exp(-\frac{1}{2}\beta \tilde{k}_{i,i})$$

where $\mu = K_{nm}K_{mm}^{-1}u$, and $\tilde{k}_{i,i}$ is the i-th diagonal element of $\tilde{K} = K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}$. The difference between the bound $\mathcal{L}_1$ and the original likelihood can be written as: $KL(p(f|u)||p(f|u,y))$. This is approximating the distribution of f given the data with the distribution of f with only the variables u. At $m = n$, the KL divergence is 0 and $\exp(\mathcal{L}_1) = p(y|f)$, since $u = y$. When $m < n$, we can maximize the bound with respect to the inducing variables $\{z_i\}_{i=1}^{m}$.

By marginalizing the inducing inputs we recover Titsias (2009) bound:

$$\begin{aligned} \log p(y|X) &= \log \int p(y|u)p(u)du \\ &\geq \log \int \exp(\mathcal{L}_1)p(u)du = \mathcal{L}_2 \end{aligned}$$

$$\mathcal{L}_2 = \log \mathcal{N}(y|0, K_{nm}K_{mm}^{-1}K_{mn} + \sigma^2 I) - \frac{1}{2\sigma^2}tr(\tilde{K})$$

It implicitly approximate the variational distribution $q(u)$ with precision:

$$\Lambda \quad = \quad \beta K_{mm}^{-1}K_{mn}K_{nm}K_{mm}^{-1} + K_{mm}^{-1}$$

and mean:

$$\hat{u} \quad = \quad \beta \Lambda^{-1} K_{mm}^{-1} K_{mn} y$$

However, to apply SVI we need to keep the representation of u explicit. We can now use the variational distribution $q(u)$ to lower bound $p(y|X)$:

$$\begin{aligned} \log p(y|X) &\geq \mathcal{L}_2 \\ &\geq \langle \mathcal{L}_1 + \log p(u) - \log q(u) \rangle_{q(u)} = \mathcal{L}_3 \end{aligned}$$

where $q(u) = N(u|m, S)$. $\mathcal{L}_2 = \mathcal{L}_3$ at the unique maximum $S = \Lambda^{-1}$ and $m = \hat{u}$. We can write the bound as:

$$\mathcal{L}_3 \quad = \quad \sum_{i=1}^{n} \{\log N(y_i|k_i^T K_{mm}^{-1} m, \beta^{-1}) - \frac{1}{2}\beta \tilde{k}_{ii} - \frac{1}{2}tr(S\Lambda_i)\} - KL(q(u)||p(u))$$

$\mathcal{L}_3$ is now in the right form to apply SVI to it, since it has global variables u and is a sum of n inputs-outputs.

4.2 Natural Gradients

The bound's gradients are given by:

$$\begin{aligned} \frac{\partial \mathcal{L}_3}{\partial m} &= \beta K_{mm}^{-1} K_{mn} y - \Lambda m, \\ \frac{\partial \mathcal{L}_3}{\partial S} &= \frac{1}{2} S^{-1} - \frac{1}{2} \Lambda \end{aligned} \tag{1}$$

For SVI we need to use the natural gradient, which is the regular gradient scaled by the inverse Fisher information: $\tilde{g}(\theta) = G(\theta)^{-1} \frac{\partial \mathcal{L}}{\partial \theta}$. We first need the canonical parameters: $\theta_1 = S^{-1} m$ and $\theta_2 = -\frac{1}{2} S^{-1}$, and the expectation parameters: $\eta_1 = m$ and $\eta_2 = m m^T + S$. Since the normal is a distribution belonging to the exponential family, the natural gradient is simplified as: $\tilde{g}(\theta) = \frac{\partial \mathcal{L}_3}{\partial \eta}$. The updates can thus be written as (Hensman, Fusi, and Lawrence, 2013):

$$\begin{aligned} \theta_{2(t+1)} &= \theta_{2(t)} + \epsilon \left(-\frac{1}{2} \Lambda + \frac{1}{2} S_t^{-1} \right) \\ \theta_{1(t+1)} &= \theta_{1(t)} + \epsilon \left(\beta K_{mm}^{-1} K_{mn} y - S_t^{-1} m_{(t)} \right) \end{aligned}$$

5 Experiments

The computation were carried using the GPy python package (The GPy authors, 2015), and scikit-learn (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot, and Duchesnay, 2011) for different utilities, the experiments code can also be find in the ipython notebook attached with the report. The inputs were normalized, and the output centered at 0 to ease optimization and maximum likelihood estimation when applicable. We used k means centers to choose our inducing inputs, the require computation time are explicitly added in the results tables.

5.1 Unified Parkinson's Disease Rating Scale

Unified Parkinsons Disease Rating Scale (UPDRS) is used to track the progression of the Parkinson's disease. It is a costly and time consuming technique that requires trained medical staff. Tsanas et al. (2010) used a dataset where simple, self-administered and non-invasive speech tests were conducted, and used it to estimate the UPDRS. The dataset can be downloaded at:

http://archive.ics.uci.edu/ml/datasets/Parkinsons+Telemonitoring.

The dataset have the following 16 features:

- 5 measures of variation in fundamental frequency

- 6 measures of variation in amplitude
- 2 measures of ratio of noise to tonal components in the voice
- A nonlinear dynamical complexity measure
- Signal fractal scaling exponent
- A nonlinear measure of fundamental frequency variation

We will perform two regressions task: (1) predict the Motor-UPDRS and (2) predict the Total-UPDRS. For the two tasks we will select 750 inducing inputs by using k-means. We used the ARD kernel and a white noise kernel, in the manner that Tsanas et al. (2010) used the lasso as a feature selection techniques. SVI optimization used ADADELTA with batchsize of 500, step size of 0.1, momentum of 0.9, and 1250 iterations. We used a training set of 5283 and testing set of 592 observations.

The accuracy is reported with the mean absolute error (MAE), as in Tsanas et al. (2010), defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|U_i - \hat{U}_i|$$

5.1.1 Motor-UPDRS

The motor-UPDRS captures motors impairment, and tasks such as speech and facial expression. It ranges from 0-108, where 108 is severe motor impairment.

GPs Comparative Results			
Methods	Log Likelihood	MAE	Computation time
CART	-	5.8	-
GP $N = 5375$	-17773.56	4.815	2144.985
SGP $M = 750$	-17900.76	4.961	1829.935 + 12.021
SVGP $M = 750$	-18230.44	5.102	1094.150 + 12.021

Tsanas et al. (2010) achieve a testing MAE of 5.8 with the CART model. It is easy to see that GPs are performing a lot better than the CART methods. The exact GP using the whole dataset is obviously performing the best, but the sparse GP and the SVI sparse GP are doing fairly well in terms of prediction and of log likelihood.

5.1.2 Total-UPDRS

The total-UPDRS is the sum of (1) Mentation, Behavior and Mood; (2) Activities of daily living; (3) Motor. The total-UPDRS ranges from 0-176, where 176 is total disability (Tsanas, Little, McSharry, and Ramig, 2010).

GPs Comparative Results			
Methods	Log Likelihood	MAE	Computation time
CART	-	7.5	-
GP $N = 5375$	-19146.89	6.05	2696.75
SGP $M = 750$	-19265.57	6.24	1492.23 + 12.021
SVGP $M = 750$	-21084.22	6.319	958.09 + 12.021

Tsanas et al. (2010) achieved a testing MAE of 7.5 with the CART model. Again, GPs have better predictive performances than the CART method. The likelihood of the SVI sparse GP is quite far from the other 2 GPs, but it is rather close in term of MAE. It is possible that it was due to optimization difficulties.

5.2 SARCOS

We will now compare the variational sparse GP to other GP on the well known SARCOS anthropomorphic robot arm dataset (http://www.gaussianprocess.org/gpml/data/). The regression task involves predicting a joint torque using 7 joint positions, 7 joint velocities and 7 joint accelerations covariates Rasmussen (2006). The training set consists of 44484 observations and the testing of 4449 observations. Given the size of the dataset, Rasmussen (2006) only used 4096 observations to train their GP. For the SVI optimization we used ADADELTA with batchsize of 1000, Gaussian, and white noise kernels, step size of 0.1 and momentum of 0.9 and 15000 iterations.

Rasmussen (2006) used the standardized mean squared error (SMSE) to evaluate the performance of their algorithms since the MSE is too sensitive to the scale of the target values.

$$SMSE = \frac{1}{N}\sum_{i=1}^{N}\frac{(U_i - \hat{U}_i)^2}{Var(U)}$$

GPs Comparative Results			
Methods	Log Likelihood	SMSE	Computation time
GP $N = 4096$	-	0.0197	6996.87
SGP $M = 1250$	-105688.10	0.0104	37220.39 + 477.77
SVGP $M = 1250$	-111860.03	0.0154	24406.92 + 477.77

Rasmussen (2006) reached a SMSE of 0.011 with a GP $N = 4096$, but I was not able to repplicate the error rate with the GPy library. In every cases, it is better to use the exact sparse GP with all the data points than a GP with a subset of regressors. SVI for sparse GP is better than our GP with a subset of regressors, but not the Rasmussen (2006) results. In terms of log-likelihood and SMSE it gets fairly close to exact sparse GP.

6 Conclusion

We introduced the concepts of Gaussian processes, sparse Gaussian processes, and of stochastic variational inference. We then reviewed Hensman et al. (2013) modification of sparse GPs so we can use SVI. Finally, we investigated on 2 large datasets how SVI can make sparse Gaussian process faster while keeping a reasonable accuracy. Specifically, we compare SVI sparse GP to exact GP and exact sparse GP on the UPDRS tasks, and we compare SVI sparse GP to exact GP on a subset of regressors and exact sparse GP on the SARCOS task. In conclusion, SVI sparse GP are way faster than exact GP and exact sparse GP, but they showed the same difficulties as other approximate inference methods requiring optimization: tweaking learning rates.

References

Blei, D. M., A. Kucukelbir, and J. D. McAuliffe (2016). Variational inference: A review for statisticians. *arXiv preprint arXiv:1601.00670*.

Duvenaud, D. (2014). *Automatic Model Construction with Gaussian Processes*. Ph. D. thesis, Computational and Biological Learning Laboratory, University of Cambridge.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin (2014). *Bayesian data analysis*, Volume 2. Taylor & Francis.

Hensman, J., N. Fusi, and N. D. Lawrence (2013). Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*.

Hoffman, M. D., D. M. Blei, C. Wang, and J. Paisley (2013). Stochastic variational inference. *The Journal of Machine Learning Research 14*(1), 1303–1347.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830.

Rasmussen, C. E. (2006). Gaussian processes for machine learning.

Robbins, H. and S. Monro (1951). A stochastic approximation method. *The annals of mathematical statistics*, 400–407.

The GPy authors (2012–2015). GPy: A gaussian process framework in python. http://github.com/SheffieldML/GPy.

Titsias, M. K. (2009). Variational learning of inducing variables in sparse gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 567–574.

Tsanas, A., M. A. Little, P. E. McSharry, and L. O. Ramig (2010). Accurate telemonitoring of parkinson's disease progression by noninvasive speech tests. *Biomedical Engineering, IEEE Transactions on 57*(4), 884–893.

Wainwright, M. J. and M. I. Jordan (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning 1*(1-2), 1–305.