Lomonosov Moscow State University
Faculty of Computer Science

Review of matherials on

Gaussian Processes for Machine Learning

Pavel Izmailov

Moscow, 2016

1 Theory

In this section an introduction to Gaussian process theory is provided.

1.1 Gaussian Process

Consider the following definition

Definition 1. *A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

A Gaussian process is completely specified by it's mean function and covariance function. These functions are defined as follows

Definition 2. *Let $f(x)$ be a real-valued Gaussian process. Then the functions*

$$m(x) = \mathbb{E}[f(x)],$$

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))],$$

are the mean function and the covariance function of the process f respectively.

We will write the Gaussian process as $f(x) \sim \mathcal{GP}(m(x), k(x, x'))$.

1.2 GP-regression

Consider the following task. We have a dataset $\{(x_i, f_i)|i = 1, \ldots, n\}$, generated from a Gaussian process $f \sim \mathcal{GP}(m(x), k(x, x'))$, let $x \in \mathbb{R}^d$. We will denote the matrix comprised of points $x_1, \ldots, x_n$ by $X \in \mathbb{R}^{n \times d}$ and the vector of corresponding values $f_1, ..., f_n$ by $f \in \mathbb{R}^n$. We want to predict the values $f_* \in \mathbb{R}^m$ of this random process at a set of other m points $X_* \in \mathbb{R}^{m \times d}$. The joint distribution of f and f_* is given by

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right),$$

where $K(X, X) \in \mathbb{R}^{n \times n}$, $K(X, X_*) = K(X^*, X)^T \in \mathbb{R}^{n \times m}$, $K(X^*, X^*) \in \mathbb{R}^{m \times m}$ are the matrices comprised of pairwise values of the covariance function k for the given sets.

The conditional distributin

$$f_*|X_*, X, f \sim \mathcal{N}(\hat{m}, \hat{K}),$$

where

$$\mathbb{E}[f_*|f] = \hat{m} = K(X_*, X)K(X, X)^{-1}f,$$

$$\text{cov}(f_*|f) = \hat{K} = K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*).$$

Thus, predicting the values of the Gaussian process at a new data point requires solving a linear system with a matrix of size $n \times n$ and thus scales as $O(n^3)$.

1.2.1 Noisy case

Consider the following model. We now have a dataset $\{(x_i, y_i)|i = 1, \dots n\}$, where $y_i = f(x_i) + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma_n)$. This means that we only have access to the noisy observations and not the true values of the process at data points. With the notation and logics similar to the one we used it the previous section we can find the conditional distribution for the values f_* of the process at new points X_* in this case:

$$f_*|y \sim \mathcal{N}(\hat{m}, \hat{K}),$$

$$\mathbb{E}[f_*|y] = \hat{m} = K(X_*, X)^{-1}(K(X, X) + \sigma_n^2 I)^{-1} y,$$
$$\operatorname{cov}(f_*|y) = \hat{K} = K(X_*, X_*) - K(X_*, X)(K(X, X) + \sigma_n^2 I)^{-1} K(X, X_*).$$

1.3 GP-classification

To be wrritten.

1.4 Kernel functions

To be wrritten.

1.5 Hyper-parameter estimation

Bayesian paradigm provides a way of estimating the kernel hyper-parameters of the GP-model through th maximizization of the marginal likelihood of the model. Marginal likelihood is given by

$$p(y|X) = \int p(y|f, X)p(f|X)df,$$

which is the likelihood, marginalized over the hidden values f of the underlying process.

1.6 Theoretical perspectives

To be wrritten.

2 Review of existing methods

It follows from the discussion above, that full Gaussian process regression scales as $O(n^3)$ and thus cannot be applied to big datasets. In this section we will review several approximate methods, that make Gaussian processes practical.

2.1 Methods, based on inducing inputs

Most of the existing methods are based on introducing a set of m function points that are called inducing inputs. Using these inputs one can make approximate predictions of the values of the hidden process at test points with a complexity of $O(nm^3)$ instead of $O(n^3)$.

Consider the following situation. We have a dataset of n examples x_i with corresponding values y_i. We will denote the matrix of pairwise values of the covariance function by K_{nn}. Now we introduce a set of m inducing inputs. We will denote the corresponding covariance matrix by K_{mm} and the matrices of covariances between the inducing points and training points by K_{nm} and K_{mn}. We will denote the vectors, comprised of noisy and true function values y_i and f_i at training points by $\boldsymbol{y}$ and $\boldsymbol{f}$ respectively. We will also introduce a distribution $q(\boldsymbol{u})$ over the hidden function values $\boldsymbol{u}$ at the inducing inputs.

It's easy to see, that

$$p(y|f) = \mathcal{N}(y|f, \sigma_n I),$$
$$p(f|u) = \mathcal{N}(f|K_{nm}K_{mm}^{-1}u, \tilde{K}),$$
$$p(u) = \mathcal{N}(u|0, K_m m),$$

where $\tilde{K} = K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}$.

2.1.1 Variational learning of inducing points

The method discussed here was introduced in [1]. To be written.

2.1.2 Stochastic variational inference

The method discussed here was proposed in [2]. The method doesn't provide a way to choose the positions of inducing points. It provides a way to find the predictive distribution and optimize hyper-parameters for large datasets.

For using stochastic variational inference, we have to provide a lower bound for the marginal likelihood, that factorizes over the training examples. To obtain such an ELBO (evidence lower bound) two ancillary lower bounds are found.

By applying the Jensen inequality we obtain

$$\log p(y|u) = \log\left(\int p(y|f)p(f|u)du\right) \geq \int \log(p(y|f))p(f|u)du = L_1.$$

As $p(y|f)$ factorizes over examples we obtain

$$\exp(L_1) = \prod_{i=1}^{n} \mathcal{N}(y_i|\mu_i, \sigma_n) \exp\left(-\frac{1}{2\sigma_n}\tilde{K}_{ii}\right).$$

Note that

$$\log p(y|u) - L_1 = \mathrm{KL}\left(p(f|u) \;||\; p(f|u, y)\right).$$

Using the lower bound L_1 we obtain a lower bound for the marginal likelihood

$$\log p(y) = \log\left(\int p(y|u)p(u)du\right) \geq \log\left(\int \exp(L_1)p(u)du\right) = L_2.$$

With some algebraic manipulations we obtain the following expression for L_2

$$L_2 = \log \mathcal{N}(y|0, K_{nm}K_{mm}^{-1}K_{mn} + \sigma_n I) - \frac{1}{2\sigma_n}\mathrm{tr}(\tilde{K}).$$

This is exactly the expression for the lower bound, used in the method, described in the section 2.1.1 for the optimal approximating distribution $q(u) = \mathcal{N}(u|\Lambda, \hat{u})$, where

$$\Lambda = \frac{1}{\sigma_n}K_{mm}^{-1}K_{mn}K_{nm}K_{mm}^{-1} + K_{mm}^{-1},$$

$$\hat{u} = \frac{1}{\sigma_n}\Lambda^{-1}K_{mm}^{-1}K_{mn}y.$$

In the method, described in section 2.1.1, this lower bound is being maximized over the kernel hyper-parameters and the optimal distribution $q(u)$ is used for making predictions at unseen points x as follows

$$\mathbb{E}f(x) = K_{xm}K_{mm}^{-1}\hat{u},$$

$$\mathrm{cov}(f(x), f(x')) = k(x, x') - K_{xm}K_{mm}^{-1}K_{mx'} + K_{xm}K_{mm}^{-1}\Lambda^{-1}K_{mm}^{-1}K_{mx'}.$$

Unfortunately, evaluating Λ takes $O(nm^2)$ operations and thus this method cannot be applied to big datasets. To overcome this limitation, we will use stochastic optimization to find the approximate optimal distribution $q(u)$ and to optimize for hyper-parameters.

Let the variational distribution q be normal with mean μ and covariance matrix Σ. The final ELBO is derived as follows

$$\log p(y) \geq \int \left(L_1 + \log p(u) - \log q(u)\right) q(u)du = L_3.$$

This lower bound factorizes over the examples

$$L_3 = \sum_{i=1}^{n}\left(\log \mathcal{N}(y_i|k_i^T K_{mm}^{-1}\mu, \sigma_n) - \frac{1}{2\sigma_n}\tilde{K}_{ii} - \frac{1}{2}\mathrm{tr}(\frac{1}{\sigma_n}\Sigma K_{mm}^{-1}k_i k_i^T K_{mm}^{-1})\right) - \mathrm{KL}\left(q(u) \;||\; p(u)\right) =$$

$$= \sum_{i=1}^{n} \left(\log \mathcal{N}(y_i | k_i^T K_{mm}^{-1} \mu, \sigma_n) - \frac{1}{2\sigma_n} \tilde{K}_{ii} - \frac{1}{2} \mathrm{tr}(\Sigma \Lambda_i) \right) -$$

$$-\frac{1}{2} \left(\log \frac{|K_{mm}|}{|\Sigma|} - m + \mathrm{tr}(K_{mm}^{-1} \Sigma) + \mu^T K_{mm}^{-1} \mu \right),$$

where $\Lambda_i = \frac{1}{\sigma_n} K_{mm}^{-1} k_i k_i^T K_{mm}^{-1}$, and k_i is the i-th column of the matrix K_{mn}.

In stochastic variational inference natural gradients are used to maximize the ELBO. The canonical parameters for the normal distribution $q(u)$ are

$$\eta_1 = \Sigma^{-1} \mu, \quad \eta_2 = -\frac{1}{2} \Sigma^{-1}.$$

The expectation parameters are

$$\beta_1 = \mu, \quad \beta_2 = \mu \mu^T + \Sigma.$$

In the exponential family the natural gradients are equal to the gradients with respect to expectation parameters. To find these gradients we first reparametrise the ELBO

$$L_3(\beta_1, \beta_2) = \sum_{i=1}^{n} \left(\log \mathcal{N}(y_i | k_i^T K_{mm}^{-1} \beta_1, \sigma_n) - \frac{1}{2\sigma_n} \tilde{K}_{ii} - \frac{1}{2} \mathrm{tr}((\beta_2 - \beta_1 \beta_1^T) \Lambda_i) \right) -$$

$$-\frac{1}{2} \left(\log |K_{mm}| - \log |\beta_2 - \beta_1 \beta_1^T| - m + \mathrm{tr}(K_{mm}^{-1} (\beta_2 - \beta_1 \beta_1^T)) + \beta_1^T K_{mm}^{-1} \beta_1 \right).$$

Differentiating with respect to expectation parameters we obtain

$$\frac{\partial L_3}{\partial \beta_1} = -\frac{1}{\sigma_n} \sum_{i=1}^{n} \left(K_{mm}^{-1} k_i y_i \right) + \Sigma^{-1} \mu, \tag{1}$$

$$\frac{\partial L_3}{\partial \beta_2} = \frac{1}{2} \left(-\sum_{i=1}^{n} (\Lambda_i) + \Sigma^{-1} - K_{mm}^{-1} \right). \tag{2}$$

The natural gradient descent updates of these parameters are

$$\eta_{1(t+1)} = \Sigma_{(t+1)}^{-1} \mu_{(t+1)} = \Sigma_{(t)}^{-1} \mu_{(t)} + \ell \left(\frac{1}{\sigma_n} K_{mm}^{-1} K_{mn} y - \Sigma_{(t)}^{-1} \mu_{(t)} \right),$$

$$\eta_{2(t+1)} = -\frac{1}{2} \Sigma_{(t+1)}^{-1} = -\frac{1}{2} \Sigma_{(t)}^{-1} + \ell \left(-\frac{1}{2} \Lambda + \frac{1}{2} \Sigma_{(t+1)}^{-1} \right),$$

where ℓ is the step length. It's easy to see, that if $\ell = 1$ the method converges to the optimal distribution $q(u)$ in one iteration. Unfortunately, we can not directly compute the updates described above, because the computational complexity of computing the matrix Λ is $O(nm^2)$. We will use approximations to the natural gradients, obtained by considering the data points individually or in batches. The formulas for these approximations can be obtained from equalities 1, 2.

Finally, we need to find the derivatives of the ELBO with respect to kernel hyper-parameters θ apart from σ_n

$$\frac{\partial L_3}{\partial \theta} = \sum_{i=1}^{n} \left[\frac{1}{\sigma_n}(y_i - k_i^T K_{mm}\mu) \left(\frac{\partial k_i^T}{\partial \theta} K_{mm}^{-1} - k_i^T K_{mm}^{-1} \frac{\partial K_{mm}}{\partial \theta} K_{mm}^{-1} \right) \mu + \right.$$

$$+ \frac{1}{2\sigma_n} \left(-\frac{\partial K_{nn}}{\partial \theta} + \frac{\partial K_{nm}}{\partial \theta} K_{mm}^{-1} K_{mn} + K_{nm} K_{mm}^{-1} \frac{\partial K_{mm}}{\partial \theta} K_{mm}^{-1} K_{mn} + K_{nm} K_{mm}^{-1} \frac{\partial K_{mn}}{\partial \theta} \right)_{ii} +$$

$$\left. + \frac{1}{\sigma_n} \mathrm{tr} \left(\Sigma \left(K_{mm}^{-1} \frac{\partial K_{mm}}{\partial \theta} K_{mm}^{-1} k_i k_i^T K_{mm}^{-1} - K_{mm}^{-1} \frac{\partial k_i}{\partial \theta} k_i^T K_{mm}^{-1} \right) \right) \right]$$

$$- \frac{1}{2} \mathrm{tr} \left(K_{mm}^{-1} \frac{\partial K_{mm}}{\partial \theta} \right) + \frac{1}{2} \mathrm{tr} \left(\Sigma K_{mm}^{-1} \frac{\partial K_{mm}}{\partial \theta} K_{mm}^{-1} \right) + \frac{1}{2} \mu^T K_{mm}^{-1} \frac{\partial K_{mm}}{\partial \theta} K_{mm}^{-1} \mu,$$

and for σ_n we have the same formula plus the following correction

$$\sum_{i=1}^{n} \left(-\frac{1}{2\sigma_n} + \frac{1}{2\sigma_n^2}(k_i^T K_{mm}^{-1}\mu - y_i)^2 + \frac{1}{2\sigma_n^2}\tilde{K}_{ii} + \frac{\mathrm{tr}(\Sigma \Lambda_i)}{2\sigma_n} \right).$$

Now, we can optimize the kernel hyper-parameters and the noise variance alongside the variational parameters.

We can also maximize the L_3 with procedures, other than stochastic gradient descent. However, in most of the effective optimization methods we can't use natural gradients, because they are not necesserily a descending direction. Thus, we have to use the usual gradients. However, there is a problem with this approach as well. The steps in the direction of the antigradient does not guarantee that the updated covariance Σ is positive definite.

To solve this problems, we use Choletsky decomposition L_Σ of Σ and optimize L_3 with respect to it.

$$L_3(L_\Sigma, \mu) = \sum_{i=1}^{n} \left(\log \mathcal{N}(y_i | k_i^T K_{mm}^{-1}\mu, \sigma_n) - \frac{1}{2\sigma_n}\tilde{K}_{ii} - \frac{1}{2}\mathrm{tr}(L_\Sigma L_\Sigma^T \Lambda_i) \right) -$$

$$- \frac{1}{2} \left(\log \frac{|K_{mm}|}{|L_\Sigma L_\Sigma^T|} - m + \mathrm{tr}(K_{mm}^{-1} L_\Sigma L_\Sigma^T) + \mu^T K_{mm}^{-1} \mu \right) =$$

$$= \sum_{i=1}^{n} \left(\log \mathcal{N}(y_i | k_i^T K_{mm}^{-1}\mu, \sigma_n) - \frac{1}{2\sigma_n}\tilde{K}_{ii} - \frac{1}{2}\mathrm{tr}(L_\Sigma^T \Lambda_i L_\Sigma) \right) -$$

$$- \frac{1}{2} \left(\log |K_{mm}| - 2\sum_{j=1}^{m} \log (L_\Sigma)_{jj} - m + \mathrm{tr}(L_\Sigma^T K_{mm}^{-1} L_\Sigma) + \mu^T K_{mm}^{-1} \mu \right)$$

The gradients with respect to μ and L_σ are given by

$$\frac{\partial L_3}{\partial \mu} = \sum_{i=1}^{n} \left(\Lambda_i \mu - \frac{y_i}{\sigma_n} K_{mm}^{-1} k_i \right) + K_{mm}^{-1}\mu,$$

$$\frac{\partial L_3}{\partial L_\Sigma} = -\sum_{i=1}^{n} \Lambda_i \Sigma_L + \begin{pmatrix} \frac{1}{(L_\Sigma)_{11}} & 0 & \dots & 0 \\ 0 & \frac{1}{(L_\Sigma)_{22}} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \frac{1}{(L_\Sigma)_{mm}} \end{pmatrix} - K_{mm}^{-1} L_\Sigma.$$

Literature

[1] Titsias M. K. (2009). Variational Learning of Inducing Variables in Sparse Gaussian Processes. In: *International Conference on Artificial Intelligence and Statistics*, pp. 567–574.

[2] Hensman J., Fusi N., Lawrence D. (2013). Gaussian Processes for Big Data. In: *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*.