Technical Report

TR-FSU-INF-CV-2013-01

An Efficient Approximation for Gaussian Process Regression

Paul Bodesheim[1], Alexander Freytag[1], Erik Rodner[1,2], and Joachim Denzler[1]

[1]Computer Vision Group, Friedrich Schiller University Jena, Germany
http://www.inf-cv.uni-jena.de
[2]UC Berkeley EECS, International Computer Science Institute, United States

Abstract. Gaussian processes are a powerful tool for regression problems. Beside computing regression curves using predictive mean values, the uncertainty of the estimations can be computed in terms of predictive variances. However, the complexity of learning and testing the model is often too large for practical use. We present an efficient approximation of the Gaussian process regression framework leading to reduced complexities of both runtime and memory. The idea is to approximate Gaussian process predictive mean and variance using a special diagonal matrix instead of the full kernel matrix. We show that this simple diagonal matrix approximation of the Gaussian process predictive variance is a true upper bound for the exact variance. Experimental results are presented for a standard regression task.

1 Introduction

The goal of regression is to model the dependencies between data points and output values. A non-parametric approach is Gaussian process regression [5], where the resulting model can be fully described using the already observed data. Given a fixed set of training samples $\mathbf{X}$ with corresponding output values $\mathbf{y}$, the assumption of Gaussian process regression is that latent functions f are drawn from a Gaussian process prior with mean function $\mu(\cdot)$ and covariance function $\kappa(\cdot,\cdot)$. These latent functions map each input $\mathbf{x} \in \mathbf{X}$ to its output value $y(\mathbf{x}) = \mathrm{f}(\mathbf{x}) + \varepsilon$ while considering induced noise ε for regularization. A typical noise model is the Gaussian noise model with zero mean and noise variance σ_{n}^2, *i.e.*, $\varepsilon \sim \mathcal{N}(0, \sigma_{\mathsf{n}}^2)$.

With Gaussian process regression, prediction can be done in a Bayesian manner by marginalizing over latent functions f. Due to the Gaussian noise model, we obtain a Gaussian distribution for the output value y^* of a sample $\mathbf{x}^*$ given the training data: $y^* \mid \mathbf{X}, \mathbf{y}, \mathbf{x}^* \sim \mathcal{N}(\mu_*, \sigma_*^2)$, whose moments can be computed in closed form (assuming a zero mean Gaussian process prior):

$$\mu_* = \mathbf{k}_*^{\mathsf{T}} \left(\mathbf{K} + \sigma_{\mathsf{n}}^2 \mathbf{I}\right)^{-1} \mathbf{y} \quad \text{and} \tag{1}$$

$$\sigma_*^2 = k_{**} - \mathbf{k}_*^{\mathsf{T}} \left(\mathbf{K} + \sigma_{\mathsf{n}}^2 \mathbf{I}\right)^{-1} \mathbf{k}_* + \sigma_{\mathsf{n}}^2 \ . \tag{2}$$

Here, $\mathbf{K} = \kappa(\mathbf{X}, \mathbf{X})$ is the covariance (kernel) matrix of the training data, $\mathbf{k}_* = \kappa(\mathbf{X}, \mathbf{x}^*)$ are similarities between the test sample and the training set, and $k_{**} = \kappa(\mathbf{x}^*, \mathbf{x}^*)$ is the self-similarity of $\mathbf{x}^*$. A more detailed description of Gaussian process regression can be found in [5].

An obvious drawback of Gaussian process regression is the complexity of learning and computing predictive variances, which is cubic and quadratic in the number of training samples, respectively. This is mainly caused by the products with the inverse matrix in (1) and (2). Moreover, the demanded memory is quadratic in the number of training samples as well, since the whole kernel matrix or the representation of its inverse has to be stored in memory. To overcome these limitations, we present an efficient approximation of Gaussian process regression in this article leading to time and memory efficient methods for learning and computing predictive variances.

The remainder of this paper is organized as follows. Related work on approximations of Gaussian process techniques is summarized in Sect. 2. Afterwards, we present the approximation of predictive variances in Sect. 3.1 and verify that it is a true upper bound of the exact computation (Sect. 3.2). How to transfer our approximation idea to computations of the predictive mean is carried out in Sect. 3.3. Results in a standard regression task are shown in Sect. 4. Conclusions can be found in the last section.

2 Related Work

A lot of work has been investigated in developing sparse approximations of Gaussian process regression and an overview of various techniques can be found in [4]. The authors introduce a unifying scheme based on latent inducing variables and the following sparse approximation methods can be derived within this scheme:

- Subset of Regressors (SoR),
- Deterministic Training Conditional (DTC),
- Fully Independent Training Conditional (FITC), and
- Partially Independent Training Conditional (PITC).

The Subset of Data (SoD) method is also mentioned in [4] but does not fall inside their general scheme. This is the simplest possible but less sophisticated sparse approximation. For comparisons, we apply the FITC approximation originally proposed by [7] as a baseline method representing the state-of-the-art. An extensive overview of approximating Gaussian processes in binary classification tasks with noise models more complex than the Gaussian noise model is given in [3]. The authors focus on

- Laplace Approximation (LA),
- Expectation Propagation (EP),
- Variational Bounds (VB),
- Factorial Variational Method (FV),
- KL-Divergence minimization, and

– sampling methods using Markov Chain Monte Carlo (MCMC) techniques.

In the following section, we present a simple diagonal approximation of the kernel matrix to speed up Gaussian process regression.

3 Diagonal Kernel Matrix Approximations

In this section, we introduce the main idea of our approximation for Gaussian process regression. After presenting how to compute the approximation in Sect. 3.1, we verify in Sect. 3.2 that our variance approximation is a true upper bound of the exact predictive variance and derive bounds for the approximation error. Finally, we show how to transfer our idea to approximate the Gaussian process predictive mean in Sect. 3.3.

3.1 Approximating the Gaussian Process Predictive Variance

The key idea of our approximation is speeding up Gaussian process predictive variance computations by using a diagonal matrix instead of the full kernel matrix to allow for efficient matrix inversion. This approximation has a quadratic runtime during learning and a linear runtime in the test step with only linear memory demand for both steps. Usually, the memory demand is quadratic, since at least once the whole kernel matrix has to be kept in memory. Note that we have to store all training samples with a linear memory demand in both cases, if the dimension of the features is fixed. Instead of calculating (2), we compute:

$$\tilde{\sigma}_*^2 = k_{**} - \mathbf{k}_*^\mathsf{T} \tilde{\mathbf{D}}^{-1} \mathbf{k}_* + \sigma_\mathsf{n}^2 \tag{3}$$

with $\tilde{\mathbf{D}}$ being a diagonal matrix whose elements are given by $(\tilde{\mathbf{D}})_{jj} = \mathbf{1}^\mathsf{T}(\mathbf{K} + \sigma_\mathsf{n}^2\mathbf{I})_{\cdot j} = \sum_{i=1}^{N}(\mathbf{K} + \sigma_\mathsf{n}^2\mathbf{I})_{ij}$. The number of training samples is denoted with N. During learning, we need time $\mathcal{O}(N^2)$ to compute the values of the matrix $\tilde{\mathbf{D}}$ whereas learning with the exact Gaussian process regression model takes time $\mathcal{O}(N^3)$. For a test sample we need $\mathcal{O}(N)$ time to compute the variance approximation instead of $\mathcal{O}(N^2)$ to compute the exact variance. Furthermore, our variance approximation needs memory only linear in the number of training samples during both learning and testing, since each element of the diagonal matrix $\tilde{\mathbf{D}}$ can be calculated one after another. In the following section, we show that our variance approximation is an upper bound for the exact variance and also present bounds for the approximation error.

3.2 Upper and Lower Bounds of the Variance Approximation

A trivial bound for our variance approximation can be derived by noting that $\tilde{\mathbf{D}}$ as well as $\tilde{\mathbf{D}}^{-1}$ are diagonal matrices with positive entries and are therefore positive definite. This fact directly leads to $\tilde{\sigma}_*^2 \leq k_{**} + \sigma_\mathsf{n}^2 \doteq \text{ub}$. Tighter bounds can be obtained by considering the eigenvalues of $\tilde{\mathbf{D}}$, which are directly the elements on the main diagonal.

Proposition 1 (Bounds for the variance approximation). *The proposed variance approximation* $\tilde{\sigma}_*^2 = k_{**} - \mathbf{k}_*^\mathsf{T}\tilde{\mathbf{D}}^{-1}\mathbf{k}_* + \sigma_\mathsf{n}^2$ *is bounded as follows:*

$$k_{**} + \sigma_\mathsf{n}^2 - (\lambda_{min}(\tilde{\mathbf{D}}))^{-1}\|\mathbf{k}_*\|^2 \;\leq\; \tilde{\sigma}_*^2 \;\leq\; k_{**} + \sigma_\mathsf{n}^2 - (\lambda_{max}(\tilde{\mathbf{D}}))^{-1}\|\mathbf{k}_*\|^2 \qquad (4)$$

with $\lambda_{max}(\tilde{\mathbf{D}}) = \max_{1\leq j\leq N}\sum_{i=1}^{N}(\tilde{\mathbf{K}})_{ij}$ *and* $\lambda_{min}(\tilde{\mathbf{D}}) = \min_{1\leq j\leq N}\sum_{i=1}^{N}(\tilde{\mathbf{K}})_{ij}$ *denoting the maximum and minimum eigenvalue of the diagonal matrix* $\tilde{\mathbf{D}}$*. We refer to the lower and upper bound in* (4) *as* lb^* *and* ub^**, respectively.*

Proof. We can prove this proposition by showing that:

$$(\lambda_{\mathsf{max}}(\tilde{\mathbf{D}}))^{-1}\|\mathbf{k}_*\|^2 \leq \mathbf{k}_*^\mathsf{T}\tilde{\mathbf{D}}^{-1}\mathbf{k}_* \leq (\lambda_{\mathsf{min}}(\tilde{\mathbf{D}}))^{-1}\|\mathbf{k}_*\|^2 \qquad (5)$$

holds for arbitrary $\mathbf{k}_*$. This directly follows from:

$$\lambda_{\mathsf{min}}(\tilde{\mathbf{D}}^{-1})\|\mathbf{k}_*\|^2 \leq \mathbf{k}_*^\mathsf{T}\tilde{\mathbf{D}}^{-1}\mathbf{k}_* \leq \lambda_{\mathsf{max}}(\tilde{\mathbf{D}}^{-1})\|\mathbf{k}_*\|^2 \;\;, \qquad (6)$$

where $\lambda_{\mathsf{min}}(\tilde{\mathbf{D}}^{-1}) = (\lambda_{\mathsf{max}}(\tilde{\mathbf{D}}))^{-1}$ and $\lambda_{\mathsf{max}}(\tilde{\mathbf{D}}^{-1}) = (\lambda_{\mathsf{min}}(\tilde{\mathbf{D}}))^{-1}$ denote the minimum and maximum eigenvalue of the diagonal matrix $\tilde{\mathbf{D}}^{-1}$. □

To verify that our approximation is a true upper bound for the exact variance, we need the following lemma stating that $\tilde{\mathbf{D}} - \tilde{\mathbf{K}}$ is positive semidefinite.

Lemma 1 ($\tilde{\mathbf{D}} - \tilde{\mathbf{K}}$ is positive semidefinite). *If* $\tilde{\mathbf{K}}$ *is the regularized kernel matrix obtained with a nonnegative kernel and* $\tilde{\mathbf{D}}$ *is a diagonal matrix with* $(\tilde{\mathbf{D}})_{jj} = \sum_{i=1}^{N}(\tilde{\mathbf{K}})_{ij}$*, the matrix* $\tilde{\mathbf{D}} - \tilde{\mathbf{K}}$ *is positive semidefinite.*

Proof. The proof is given in [2, Proposition 1, page 23], where it is shown that the graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{K}$ is positive semidefinite. □

Now we can show that our approximation is a true upper bound for the exact variance or, vice versa, the exact variance is a lower bound for our variance approximation.

Theorem 1 (Upper bound for the Gaussian process predictive variance). *Our variance approximation* $\tilde{\sigma}_*^2 = k_{**} - \mathbf{k}_*^\mathsf{T}\tilde{\mathbf{D}}^{-1}\mathbf{k}_* + \sigma_\mathsf{n}^2$ *is a true upper bound for the exact Gaussian process predictive variance* $\sigma_*^2 = k_{**} - \mathbf{k}_*^\mathsf{T}\tilde{\mathbf{K}}^{-1}\mathbf{k}_* + \sigma_\mathsf{n}^2$ *for a positive definite and nonnegative kernel. Furthermore, the error* $\tilde{\sigma}_*^2 - \sigma_*^2$ *made by the approximation is* $\mathcal{O}(\|\mathbf{k}_*\|^2)$*.*

Proof. We have to verify that $\tilde{\sigma}_*^2 \geq \sigma_*^2$. Equivalently, we show that $\tilde{\sigma}_*^2 - \sigma_*^2 \geq 0$. This inequality reduces to $\mathbf{k}_*^\mathsf{T}\tilde{\mathbf{K}}^{-1}\mathbf{k}_* - \mathbf{k}_*^\mathsf{T}\tilde{\mathbf{D}}^{-1}\mathbf{k}_* = \mathbf{k}_*^\mathsf{T}(\tilde{\mathbf{K}}^{-1} - \tilde{\mathbf{D}}^{-1})\mathbf{k}_* \geq 0$ and finally, we have to show that the matrix $\tilde{\mathbf{M}} = \tilde{\mathbf{K}}^{-1} - \tilde{\mathbf{D}}^{-1}$ is positive semidefinite if the kernel is positive definite and nonnegative. Since $\tilde{\mathbf{D}}^{-1}$ and thus $\tilde{\mathbf{D}}$ is positive semidefinite as well as $\tilde{\mathbf{D}} - \tilde{\mathbf{K}}$ (Lemma 1), we can apply Theorem 18.3.4 of [1, p. 433-434] to prove that $\tilde{\mathbf{M}} = \tilde{\mathbf{K}}^{-1} - \tilde{\mathbf{D}}^{-1}$ is positive semidefinite. Using the same arguments as in Proposition 1 leads to the following bounds:

$$\lambda_{\mathsf{min}}(\tilde{\mathbf{M}})\|\mathbf{k}_*\|^2 \;\leq\; \mathbf{k}_*^\mathsf{T}\tilde{\mathbf{M}}\mathbf{k}_* \;\leq\; \lambda_{\mathsf{max}}(\tilde{\mathbf{M}})\|\mathbf{k}_*\|^2 \;\;. \qquad (7)$$

As a direct result, the approximation error is in $\mathcal{O}(\|\mathbf{k}_*\|^2)$. □

The theorem shows that the approximation error depends on $\|\mathbf{k}_*\|^2$, which can be seen as a modified Parzen estimate with quadratic kernel terms. For samples similar to the training data we get a high approximation error whereas for outliers the error is small.

3.3 Approximating the Gaussian Process Predictive Mean

Since our proposed variance approximation leads to substituting the full matrix in (2) by a diagonal matrix, we can do the same substitution in the calculation of the Gaussian process predictive mean. We obtain an approximate predictive mean $\tilde{\mu}_*$ defined as:

$$\tilde{\mu}_* = \mathbf{k}_*^\mathsf{T} \tilde{\mathbf{D}}^{-1} \mathbf{1} \ . \tag{8}$$

The asymptotic runtime for computing the mean value of a test sample remains $\mathcal{O}(N)$ and thus linear in the number of training samples, but the complexity for learning the model can be reduced from $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2)$.

4 Experiments in Standard Regression

We compare our approximation with the exact calculation and the FITC approximation of [7] in a standard regression task. For a standard regression task on 1D inputs, the results are shown in Fig. 1. We varied the scale of the Gaussian kernel and fixed the noise variance of the Gaussian noise model: $\sigma_\mathsf{n}^2 = 0.1$. Each fourth point is used as an inducing input in the FITC approximation.

Compared to the exact calculation and the FITC approximation, we observe a smoother predictive mean curve of our approximation at each scale. The developing of predictive variances in the exact case and for our approximation is similar and we observe that our approximate variance is an upper bound for the exact one. Due to the sparsity of the FITC method, its predictive variances strongly vary in local areas, especially for a small σ. In contrast, our approximation takes each training input into account leading to smooth results while reducing the computational complexity compared to exact calculations.

5 Conclusions

We have shown that simple diagonal approximations of kernel matrices allow for efficient Gaussian process regression with impressive results in standard regression tasks. Using our approximation, the runtime complexity as well as the memory demand during learning and testing can be reduced by one order with respect to the number of training samples. Additionally, computing the approximate predictive variance is possible in linear time only and we have shown that this approximation is a true upper bound for the exact variance.

We assume that our approximation can be used in various applications, since it is simple in spirit, easy to implement, and does not limit the flexibility of the model, while saving time and memory. This is especially helpful in large-scale scenarios similar to [6], where Gaussian process regression can be successfully used for classification in visual recognition tasks.

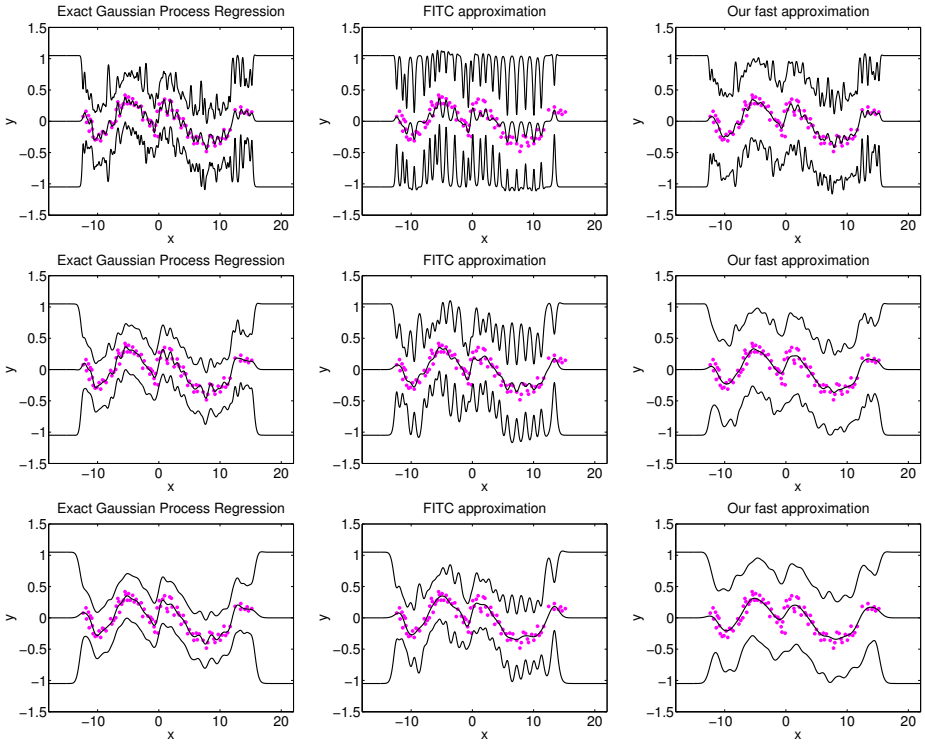

Fig. 1. Predictive distributions (mean and standard deviation) for Gaussian process regression with 1D inputs. Rows correspond to different scales of the Gaussian kernel: (top) $\sigma = 0.25$, (middle) $\sigma = 0.5$, (bottom) $\sigma = 0.75$.

References

1. Harville, D.A.: Matrix Algebra From a Statistician's Perspective. Springer (1997)
2. Luxburg, U.v.: Statistical Learning with Similarity and Dissimilarity Functions. Ph.D. thesis, Technical University of Berlin, Germany (2004)
3. Nickisch, H., Rasmussen, C.E.: Approximations for binary gaussian process classfication. JMLR 9, 2035–2078 (2008)
4. Quinonero-Candela, J., Rasmussen, C.E.: A unifying view of sparse approximate gaussian process regression. JMLR 6, 1939–1959 (2005)
5. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press (2006)
6. Rodner, E., Freytag, A., Bodesheim, P., Denzler, J.: Large-scale gaussian process classification with flexible adaptive histogram kernels. In: ECCV. pp. 85–98 (2012)
7. Snelson, E., Ghahramani, Z.: Sparse gaussian processes using pseudo-inputs. In: NIPS. pp. 1257–1264 (2005)