Beyond Classification – Large-scale Gaussian Process Inference and Uncertainty Prediction

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

[1]Computer Vision Group, University of Jena, Germany
[2]ICSI, UC Berkeley, US
firstname.lastname@uni-jena.de

Abstract

Due to the massive (labeled) data available on the web, a tremendous interest in large-scale machine learning methods has emerged in the last years. Whereas, most of the work done in this new area of research focused on fast and efficient classification algorithms, we show in this paper how other aspects of learning can also be covered using massive datasets. The paper[1] briefly presents techniques allowing for utilizing the full posterior obtained from Gaussian process regression (predictive mean and variance) with tens of thousands of data points and without relying on sparse approximation approaches. Experiments are done for active learning and one-class classification showing the benefits in large-scale settings.

1 Introduction

Learning with kernel-based methods is one of the main techniques used in the area of visual object recognition to cope with the high complexity and difficulty of this task. Their main limitation is the high computation time for learning as well as for classifying a new test example. Apart from several (sparse) approximation methods [1] presented to reduce the number of necessary evaluations, recent work [2, 3] has shown that for histogram intersection kernels (HIK), several kernel terms can be efficiently evaluated. This allows for learning and classification with support vector machines (SVM) in sub-quadratic and constant computation time, respectively.

A recent work [4] demonstrated how to utilize the inherent properties of the HIK to speed up Gaussian process (GP) regression [5] allowing for large-scale Bayesian inference and hyperparameter optimization. In this short paper, we extend the work of [4] by presenting how to obtain estimates of the predictive GP variance for large-scale scenarios (see Figure 1 for an overview of related work). The predictive variance, which is directly available in the original GP framework, is one of the main advantages of a Bayesian method and has been used for active learning [6] as well as for one-class classification [7]. In contrast to sparse GP approaches [1], our proposed methods and approximations take every single learning example into account. In the following, we briefly describe the underlying techniques and give a short overview of the experiments performed.

2 Large-scale Gaussian Process Inference

Gaussian Process Regression and Classification For classification, we use a Gaussian process model [5]. Given training examples $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, we would like to estimate the underlying latent function f mapping inputs $\boldsymbol{x}$ to outputs y. The GP framework casts this problem into a non-parameterized Bayesian formulation by marginalizing the latent function f and assuming f to be

[1] **This workshop paper is a short version** of the following publication: Alexander Freytag and Erik Rodner and Paul Bodesheim and Joachim Denzler. *Rapid Uncertainty Computation with Gaussian Processes and Histogram Intersection Kernels.* Asian Conference on Computer Vision (ACCV 2012)

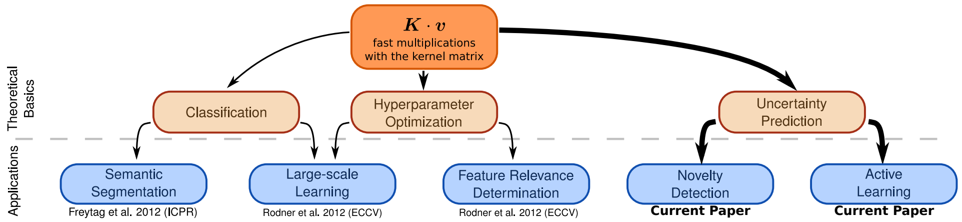

$K \cdot v$
fast multiplications with the kernel matrix
Theoretical Basics
Applications
Classification
Hyperparameter Optimization
Uncertainty Prediction
Semantic Segmentation
Freytag et al. 2012 (ICPR)
Large-scale Learning
Rodner et al. 2012 (ECCV)
Feature Relevance Determination
Rodner et al. 2012 (ECCV)
Novelty Detection
Current Paper
Active Learning
Current Paper

Figure 1: Overview of related work on large-scale Gaussian process inference using fast multiplications with kernel matrices. Our paper focuses on estimating the predictive variance.

sampled from a Gaussian process with zero mean and covariance (kernel) function K. If we assume Gaussian noise on outputs y, *i.e.*, $y = f(\boldsymbol{x}) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma_{\mathsf{n}}^2)$, we can directly obtain the predictive distribution $\mathcal{N}(\mu_*, \sigma_*^2)$ of the output y_* for a new test input $\boldsymbol{x}^*$:

$$\mu_* = \boldsymbol{k}_*^T (\mathbf{K} + \sigma_{\mathsf{n}}^2 \cdot \mathbf{I})^{-1} \boldsymbol{y} = \boldsymbol{k}_*^T \boldsymbol{\alpha} \ , \tag{1}$$

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

where $k_{**}, \boldsymbol{k}_*$, and $\mathbf{K}$ are the kernel values of the test input, between training set and test input, and of the training set itself, respectively. Furthermore, $\boldsymbol{y}$ denotes the vector of output values y_i for each training example. In this paper, we consider one-vs-all classification with the GP regression framework as done in [6, 4], *i.e.*, we have discrete outputs (labels) $y \in \{-1, 1\}$.

Histogram Intersection Kernels It has been shown that the histogram intersection kernel

$$K^{\text{hik}}(\boldsymbol{x}, \boldsymbol{x}') = \textstyle\sum_{d=1}^{D} \min(x_d, x'_d) \ , \tag{3}$$

which is often used to compare histogram feature vectors $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^D$, allows for efficient classification and learning with SVM [2, 3] due to the resulting piecewise linearity of the decision function. This property can be exploited for speeding up GP regression and hyperparameter optimization [4] as well as estimating the predictive variance as presented in the current paper.

Fast Kernel Multiplications As we have seen in Eq. (1), the predictive mean is a weighted sum of kernel values. This property is shared by SVM and many other kernel methods. The HIK allows for decomposing the weighted sum in two parts [2]:

$$\boldsymbol{k}_*^T \boldsymbol{\alpha} = \sum_{i=1}^{n} \alpha_i \sum_{d=1}^{D} \min(x_d^{(i)}, x_d^*) = \sum_{d=1}^{D} \Bigg(\sum_{\{i : x_d^{(i)} < x_d^*\}} \alpha_i x_d^{(i)} + x_d^* \sum_{\{j : x_d^{(j)} \geq x_d^*\}} \alpha_j \Bigg) \ . \tag{4}$$

The inner term is piecewise linear in x_d^* and depends on the position of x_d^* within the sorted list of all feature values $x_d^{(i)}$. Therefore, we can speed up its computation by pre-computing its first and second sum for each possible position of x_d^*. Evaluating the scores for test examples can then be done with a few evaluations of two look-up tables for each dimension. Given the vector $\boldsymbol{\alpha}$, the resulting computation time for building both tables is dominated by sorting in $\mathcal{O}(Dn \log n)$ operations. In terms of memory usage, we only have to store $\mathcal{O}(nD)$ elements in contrast to the kernel matrix of size $\mathcal{O}(n^2)$. For calculating the score of a new example, we need $\mathcal{O}(D \log n)$ operations.

Speeding up Gaussian Process Inference Similar considerations hold for multiplications of an arbitrary vector $\boldsymbol{v} \in \mathbb{R}^n$ with the kernel matrix $\mathbf{K}$, which can be done in $\mathcal{O}(nD)$. The efficient computation of products $\mathbf{K}\boldsymbol{v}$ is an essential part in our overall framework. An overview of how to benefit from the HIK properties within the GP framework is given in Figure 1. Please note that sparsity as well as a quantization of the feature space can be exploited, allowing for going down to a constant time for computing the kernel sum [2]. Furthermore, we can also indirectly perform the inversion of the kernel matrix in Eq. (1) and (2) in an efficient manner by using a linear solver [4].

3 Exploiting the Bayesian Framework

Up to now, we only considered fast computations and efficient updates of the predictive mean derived from GP regression. However, in many scenarios such as active learning or novelty detection it is important to get an estimate for the uncertainty of the prediction as well, which is related to the predictive variance. As presented in Sect. 2, the predictive variance σ_*^2 depends on three terms. While the first and third terms reflect the a-priori uncertainties $k_{**} = K(\boldsymbol{x}^*, \boldsymbol{x}^*)$ and σ_{n}^2, the second term reduces the a-priori uncertainty based on the similarities between test example $\boldsymbol{x}^*$ and training examples $\mathbf{X}$. Since this term is a quadratic instead of a linear form in $\boldsymbol{k}_*$, the previously presented techniques for fast computations of scalar products $\boldsymbol{k}_*^T \boldsymbol{\alpha}$ [4] can not be applied here. In the following, we show how to approximate the second term in an efficient manner.

RAPU – Rough Approximation of the Predictive Uncertainty Since the kernel matrix $\mathbf{K}$ is symmetric and positive definite, the same holds for its inverse. Therefore, we can bound the quadratic form in Eq. (2) using its largest eigenvalue:

$$\sigma_*^2 \leq k_{**} - ||\boldsymbol{k}_*||^2 \left(\lambda_{\max}(\mathbf{K} + \sigma_{\mathsf{n}}^2 \cdot \mathbf{I})\right)^{-1} + \sigma_{\mathsf{n}}^2 \quad . \tag{5}$$

It was already shown [4] that the computation of $\lambda_{\max}$ can be done efficiently using histogram intersection kernels and the Arnoldi iteration method. Therefore, it remains to efficiently compute $||\boldsymbol{k}_*||^2$. If we approximate $||\boldsymbol{k}_*||^2$ by a lower bound, we still obtain a valid upper bound approximation for the predictive variance as given in Eq. (5). For this purpose, we observe that especially when dealing with sparse features, the majority of mixed terms will vanish. Therefore, neglecting the mixed terms is well justifiable and we obtain a squared Parzen-like expression:

$$||\boldsymbol{k}_*||^2 = \sum_{i=1}^{n} \Big(\sum_{d=1}^{D} \min\left(x_d^*, x_d^{(i)}\right) \Big)^2 \geq \sum_{i=1}^{n} \sum_{d=1}^{D} \left(\min(x_d^*, x_d^{(i)}) \right)^2 = \sum_{i=1}^{n} \sum_{d=1}^{D} \min\left((x_d^*)^2, (x_d^{(i)})^2 \right)$$

This expression is equivalent to the one in Eq. (4) for squared features and $\alpha_i = 1$. Therefore, we can directly apply the same fast computation methods as described in [4] with squared feature values. Finally, for an unseen example $\boldsymbol{x}^*$, we can compute the squared kernel vector within $\mathcal{O}(D \log n)$ operations or in $\mathcal{O}(D)$ time when using the quantization approach (q-RAPU) presented in [4].

FAPU – Fine Approximation of the Predictive Uncertainty In the previous paragraph, we derived squared Parzen-like approximations of the GP predictive variance using a special case of quadratic form approximations (see Eq. (5)). Although these scores are efficiently computable using histogram intersection kernels, the assumption of sparse features is not always valid. In these cases, using the RAPU-method is not justifiable. Therefore, we can derive a finer approximation scheme based on the k largest eigenvalues and their eigenvectors. Due to the lack of space, we skip the exact formulas based on standard linear algebra. This method requires $\mathcal{O}(Dn + kT_1 n)$ operations in total to compute the approximation of σ_*^2, with T_1 being the maximum number of Arnoldi iterations.

PUP – Precise Uncertainty Prediction While the previously presented approaches were based on approximations, we can also exploit the properties of histogram intersection kernels to compute the exact predictive variance. To obtain the score for a single test example $\boldsymbol{x}^*$, we first compute the kernel vector $\boldsymbol{k}_*$ requiring $\mathcal{O}(Dn)$ operations. After that, we apply an iterative linear solver to compute the vector $\boldsymbol{\alpha}_* = (\mathbf{K} + \sigma_{\mathsf{n}}^2 \cdot \mathbf{I})^{-1} \boldsymbol{k}_*$. This takes $\mathcal{O}(T_2 D n)$ operations where T_2 denotes the maximum number of iterations of the linear solver. Finally, we can compute the product of $\boldsymbol{k}_*$ and $\boldsymbol{\alpha}_*$ in $\mathcal{O}(n)$ operations to obtain the second term needed to compute σ_*^2. In total, we need $\mathcal{O}(T_2 D n)$ operations to compute the exact predictive variance for an unseen example during testing. Summarizing, we are able to efficiently compute the predictive variance with selectable precision as well as selectable time to spent.

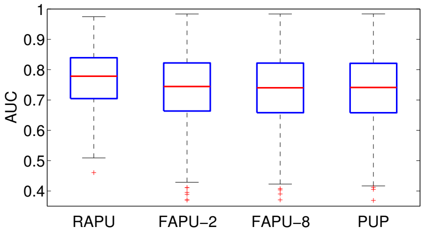
4 Experiments and Applications

We evaluate our approach on the real world ImageNet dataset [9] and the results are as follows:

1. Our approximation techniques allow for computing classification uncertainties in microseconds while requiring only a linear amount of memory.
2. Approximating uncertainties with the presented methods can be directly applied to one-class classification and active learning without any loss in recognition performance.

Experimental Setup We use the same histogram features as done in [4]. If not stated otherwise, the FAPU method with $k = 2$ eigenvectors is applied to compute σ_*^2. All experiments are conducted on a 3.4 GHz CPU using a C++ implementation without any parallelization.

Runtime Evaluation We use the ILSVRC'10 subset of ImageNet and randomly select 10 or 50 examples from each of the 1,000 classes to build up a training set. All remaining examples are used for testing. Resulting runtimes of our GP variance estimation techniques are given in the table on



Approach	10,090	50,050
q-RAPU	33.89μs	33.89μs
RAPU	62.07ms	266.97ms
FAPU, $k = 2$	105.37ms	1.15 s
FAPU, $k = 8$	105.63ms	1.47 s
PUP	22.92 s	$>$ 1min
GP-standard	1.54 s	—*

Figure 2: (Left) variance estimation for one-class classification; (Right) Runtimes needed for the computation of the predictive variance using techniques presented in Sect. 3 in comparison to the baseline GP on the ImageNet dataset. * not possible due to excessive memory demand.

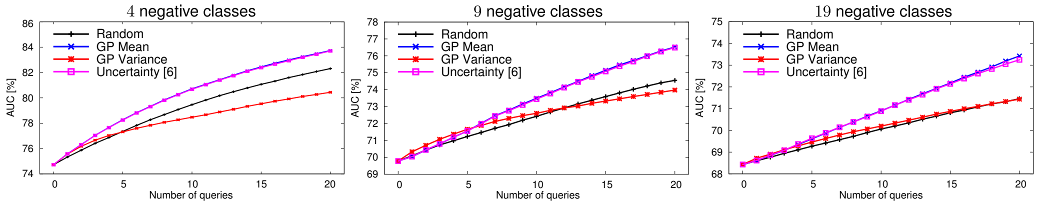


Figure 3: Active learning results on 100 binary classification tasks derived from ImageNet.

the right side of Figure 2. For rapid uncertainty prediction, the quantized RAPU methods turns out to be highly suitable with computation times in the order of microseconds. All methods except of the GP baseline are able to compute uncertainties in the large-scale learning case.

One-Class Classification As demonstrated by [7], the predictive variance can be used for one-class classification, where only data from a single class is available during training. The decision whether or not a new example belongs to the target class can be made based on its classification uncertainty. We evaluate our approximations on 1,000 OCC tasks derived from the ImageNet dataset. In each task, we train the GP classifier with 100 randomly chosen examples and averaged AUC scores are computed to measure performance. The results are given in the left plot of Figure 2 and show comparable performance for all methods, which highlights that the applied approximation techniques do not negatively influence the recognition performance.

Active Learning Our techniques can also be used to speed up the active learning methods of [6], which is validated in an experiment on the ImageNet dataset. We randomly pick a single positive class as well as four, nine, or nineteen classes serving as negative examples. Starting with two randomly chosen examples per class, we query new examples by integrating our variance estimation in the AL methods of [6]. Each task is repeated with 100 random initializations and the final results are averaged. Experimental results are given in Figure 3 showing the same behavior as observed in [6], *i.e.*, best results are obtained when combining predictive mean and variance, which highlights the suitability of our variance computation techniques.

5 Conclusions

In this short paper, we presented how to perform Gaussian process inference in large-scale scenarios, especially focusing on aspects beyond classification, such as using the whole posterior distribution available for a new test input. The work is an interesting example of how a bunch of algorithmic tricks can be used to speed up non-sparse kernel machines to a large extent.

Discussion of Future Research Directions The methods proposed in this paper are easy to implement and their underlying ideas are not restricted to the utilization for Gaussian process inference. In general, we believe that a lot of the kernel algorithms currently used can be applied to large-scale settings when taking the structure of a special type of kernel function into account. How the massive amount of data generated by crowd-sourcing can be used to build complex visual models will be one of the key research questions in the upcoming future. Learning those models requires far more than only efficient classifiers. Flexible probabilistic models are necessary, which are able to consider the inherent uncertainties of an ill-posed categorization task.

References

[1] Joaquin Quiñonero Candela and Carl E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

[2] Subhransu Maji, Alexander C. Berg, and Jitendra Malik. Classification using intersection kernel support vector machines is efficient. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*, pages 1–8, 2008.

[3] Jianxin Wu. A fast dual method for hik svm learning. In *Proceedings of the European Conference on Computer Vision (ECCV'10)*, pages 552–565, 2010.

[4] Erik Rodner, Alexander Freytag, Paul Bodesheim, and Joachim Denzler. Large-scale gaussian process classification with flexible adaptive histogram kernels. In *Proceedings of the European Conference on Computer Vision (ECCV'12)*, 2012.

[5] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. The MIT Press, 2006.

[6] Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell. Gaussian processes for object categorization. *International Journal of Computer Vision*, 88(2):169–188, 2010.

[7] Michael Kemmler, Erik Rodner, and Joachim Denzler. One-class classification with gaussian processes. In *Proceedings of the Asian Conference on Computer Vision (ACCV'10)*, volume 2, pages 489–500, 2010.

[8] Alexander Freytag, Björn Fröhlich, Erik Rodner, and Joachim Denzler. Efficient semantic segmentation with gaussian processes and histogram intersection kernels. In *Proceedings of the International Conference on Pattern Recognition (ICPR'12)*, 2012.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*, pages 248–255, 2009.