Scalable Large Margin Gaussian Process Classification

Martin Wistuba ✉ and Ambrish Rawat

IBM Research
martin.wistuba@ibm.com, ambrish.rawat@ie.ibm.com

Abstract. We introduce a new Large Margin Gaussian Process (LMGP) model by formulating a pseudo-likelihood for a generalised multi-class hinge loss. We derive a highly scalable training objective for the proposed model using variational-inference and inducing point approximation. Additionally, we consider the joint learning of LMGP-DNN which combines the proposed model with traditional Deep Learning methods to enable learning for unstructured data. We demonstrate the effectiveness of the Large Margin GP with respect to both training time and accuracy in an extensive classification experiment consisting of 68 structured and two unstructured data sets. Finally, we highlight the key capability and usefulness of our model in yielding prediction uncertainty for classification by demonstrating its effectiveness in the tasks of large-scale active learning and detection of adversarial images.

1 Introduction

This work brings together the effectiveness of large margin classifiers with the non-parametric expressiveness and principled handling of uncertainty offered by Gaussian processes (GPs). Gaussian processes are highly expressive Bayesian non-parametric models which have proven to be effective for prediction modelling. One key aspect of Bayesian models which is often overlooked by traditional approaches is the representation and propagation of uncertainty. In general, decision makers are not solely interested in predictions but also in the confidence about the predictions. An action might only be taken in the case when the model in consideration is certain about its prediction. This is crucial for critical applications like medical diagnosis, security, and autonomous cars. Bayesian formalism provides a principled way to obtain these uncertainties. Bayesian methods handle all kinds of uncertainties in a model, be it in inference of parameters or for obtaining the predictions. These methods are known to be effective for online classification [18], active learning [29], global optimization of expensive black-box functions [11], automated machine learning [32], and as recently noted, even in machine learning security [30].

Classical Gaussian process classification models [36] are generalised versions of linear logistic regression. These classifiers directly use a function modelled as a Gaussian process with a logit link or probit function [27] for obtaining the desired probabilities. Alternatively, margin classifiers like Support Vector

Machines (SVMs) employ hinge loss for learning decision functions. Gaussian process classifiers often perform similar to non-linear SVMs [16] but are preferred by some practitioners due to added advantages like uncertainty representation and automatic hyperparameter determination. Therefore, it is natural to look for a probabilistic generalisation of the hinge loss that can benefit from the numerous advantages of Bayesian modelling.

The contributions in this work are threefold. We derive a pseudo-likelihood for a general multi-class hinge loss and propose a large margin Gaussian process (LMPG). We provide a scalable learning scheme based on variational inference [2, 34, 10] to train this model. Additionally, we propose a hybrid model which combines deep learning components such as convolutional layers with the LMGP which we refer to as LMGP-DNN. This allows to jointly learn the feature extractors as well as the classifier design such that it can be applied both on structured and unstructured data. We compare the proposed LMGP on 68 structured data sets to a state-of-the-art binary Bayesian SVM with the one-vs-rest approach and the scalable variational Gaussian process [10]. On average, LMGP provides better prediction performance and needs up to an order of magnitude less training time in comparison to the competitor methods. The proposed LMGP-DNN is compared on the image classification data sets MNIST [17] and CIFAR-10 [15] to a standard (non-Bayesian) neural network. We show that we achieve similar performance, however, require increased training time. Finally, we demonstrate the effectiveness of uncertainties in experiments on active learning and adversarial detection.

2 Related Work

Motivated by a probabilistic formulation of the generalised multi-class hinge loss, this work derives and develops a scalable training paradigm for large margin Gaussian process based classification. In the vast related literature this is an advancement on two fronts - first, a novel approach to Gaussian process based classification and second, Bayesian formulation of margin classifiers, like SVMs. We position our work with respect to both these directions of research. With reference to Gaussian process based classifiers, our work closely relates to scalable variational Gaussian processes (SVGP) [10]. Infamous for the cubic dependency of learning schemes with respect to number of data samples has, in the past, limited the applicability of Gaussian process based models. However recent developments in sparse-approximation schemes [34, 31] have enabled learning of GP-based models for large scale datasets. The two works, SVGP and LMGP differ in their choice of objective functions. While SVGP utilises a variational approximation of the cross-entropy between predicted probabilities and the target probabilities for learning, LMGP seeks to maximise the margin between GP predictions. In both works, this is achieved with the use of variational inference along with inducing point approximation which scales learning to large data sets.

The probabilistic formulation of Support Vector Machines has a long standing history. However, most work has been limited to the binary-classification case with extensions to multi-class being enabled with the one-vs-rest scheme. [33] interprets SVM training as learning a maximum a posteriori solution of a model with Gaussian process priors. In addition, works like [28] have investigated extensions that benefit SVMs with certain key aspects of Bayesian formalism like model selection. For the task of binary classification, [24] make a key observation and reformulate the hinge loss in the linear SVM training objective to a location-scale mixture of Gaussians. They derive a pseudo-likelihood by introducing local latent variables for each data point and subsequently marginalize them out for predictions. A multi-class extension to this linear model has been considered in [23] with learning enabled by an expectation-maximisation based algorithm. A non-linear version of this setup is considered by [9] where the linear decision function is modeled as a Gaussian process. They approximate the resulting joint posterior using Markov chain Monte Carlo (MCMC) or expectation conditional maximization (ECM). Furthermore, they scale the inference using the fully independent training conditional approximation (FITC) [31]. The basic assumption behind FITC is that the function values are conditionally independent given the set of inducing points. Then, training the Gaussian process is no longer cubically dependent on the number of training instances. Moreover, the number of inducing points can be freely chosen. [20] extend the work of [24] by applying a mean field variational approach to it. Most recently, [35] propose an alternate variational objective and use coordinate ascent to maximize it. They demonstrate improved performance over a classical SVM, competitor Bayesian approaches, and Gaussian process-based classifiers. In the scope of this work, we contrast performance with the one-vs-rest extension of [35] and call it Bayesian SVM.

3 Large Margin Gaussian Process

This section details the proposed Large Margin Gaussian process (LMGP). We begin with a discussion of the probabilistic formulation of the hinge loss for the binary case and follow it by establishing a Bayesian interpretation of the generalised non-linear multi-class case [5]. We then establish the complete model formulation of LMGP and detail a variational-inference based scheme for scalable learning. We conclude with a description of LMGP-DNN model that extends the applicability of LMGP to image data.

3.1 Probabilistic Hinge Loss

For a binary classification task, a model trained with hinge loss seeks to learn a decision boundary with maximum margin, i.e. the separation between the decision boundary and the instances of the two classes. We represent the labeled data for a binary classification task with N observations and M-dimensional representation as $D = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$, where $\mathbf{x}_n \in \mathbb{R}^M$ and $y_n \in \{-1, 1\}$ represent predictors and labels, respectively. Training such a model, as in the case of the classical SVM, involves learning a decision function $f : \mathbb{R}^M \rightarrow \mathbb{R}$ that minimizes the regularized hinge loss,

$$\mathcal{L}\left(D, f, \gamma\right) = \sum_{n=1}^{N} \max\left\{1 - y_n f\left(\mathbf{x}_n\right), 0\right\} + \gamma R\left(f\right) \quad . \tag{1}$$

The regularizer R punishes the choice of more complex functions for f, and γ is a hyperparameter that controls the impact of this regularization. A linear SVM uses a linear decision function $f(\mathbf{x}_n) = \boldsymbol{\theta}^T \mathbf{x}_n$. Non-linear decision functions are traditionally obtained by applying the kernel trick.

For the linear case, [24] show that minimizing Equation (1) is equivalent to estimating the mode of a pseudo-posterior (maximum a posteriori estimate)

$$p\left(f|D\right) \propto \exp\left(-\mathcal{L}\left(D, f, \gamma\right)\right) \propto \prod_{n=1}^{N} L\left(y_n|\mathbf{x}_n, f\right) p\left(f\right), \tag{2}$$

derived for a particular choice of pseudo-likelihood factors L, defined by location-scale mixtures of Gaussians. This is achieved by introducing local latent variables λ_n such that for each instance,

$$L\left(y_n|\mathbf{x}_n, f\right) = \int_0^\infty \frac{1}{\sqrt{2\pi\lambda_n}} \exp\left(-\frac{1}{2}\frac{\left(1 + \lambda_n - y_n f\left(\mathbf{x}_n\right)\right)^2}{\lambda_n}\right) \mathrm{d}\lambda_n \quad . \tag{3}$$

In their formulations, [24] and [9] consider γ as a model parameter and accordingly develop inference schemes. Similar to [35], we treat γ as a hyperparameter and drop it from the expressions of prior and posterior for notational convenience. [9] extend this framework to enable learning of a non-linear decision function f. Both [9] and [35] consider models where $f(x)$ is sampled from a zero-mean Gaussian process i.e. $\mathbf{f} \sim \mathcal{N}(0, K_{NN})$, where $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]$ is a vector of function evaluations and K_{NN} is the covariance function evaluated at data points.

3.2 Generalised Multi-Class Hinge Loss

Modeling a multi-class task with SVM is typically achieved by decomposing the task into multiple independent binary classification tasks. Although simple and powerful, this framework cannot capture correlations between the different classes since the modeled binary tasks are independent. As an alternate approach, numerous extensions based on generalised notion of margins have been proposed in the literature [6]. One can view these different multi-class SVM loss functions as a combination of margin functions for the different classes, a large margin loss for binary problems, and an aggregation operator, combining the various target margin violations into a single loss value. We consider the popular formulation of [5] which corresponds to combining relative margins with the max-over-others operator. A multi-class classification task involves N observations with integral labels $Y = \{1, \ldots, C\}$. A classifier for this task can be modeled as a combination of a decision function $f : \mathbb{R}^M \rightarrow \mathbb{R}^C$ and a decision rule to compute the class labels,

$$\hat{y}\left(\mathbf{x}_n\right) = \underset{t \in Y}{\arg\max}\, f_t\left(\mathbf{x}_n\right) \ . \tag{4}$$

[5] propose to minimize the following objective function for learning the decision function f:

$$\mathcal{L}\left(D, f, \gamma\right) = \sum_{n=1}^{N} \max\left\{1 + \max_{t \neq y_n, t \in Y} f_t\left(\mathbf{x}_n\right) - f_{y_n}\left(\mathbf{x}_n\right),\ 0\right\} + \gamma R\left(f\right) \ \ , \tag{5}$$

where again γ is a hyperparameter controlling the impact of the regularizer R.

With the prior associated to $\gamma R\left(f\right)$, maximizing the log of Equation (2) corresponds to minimizing Equation (5) with respect to the parameters of f. This correspondence requires the following equation to hold true for the data-dependent factors of the pseudo-likelihood,

$$\prod_{n=1}^{N} L\left(y_n \mid \mathbf{x}_n, f\right) = \exp\left(-2 \sum_{n=1}^{N} \max\left\{1 + \max_{t \neq y_n, t \in Y} f_t\left(\mathbf{x}_n\right) - f_{y_n}\left(\mathbf{x}_n\right),\ 0\right\}\right) . \tag{6}$$

Analogously to [24], we show that $L\left(y_n \mid \mathbf{x}_n, f\right)$ admits a location-scale mixture of Gaussians by introducing local latent variables $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_n]$. This requires the lemma established by [1].

Lemma 1. *For any $a, b > 0$,*

$$\int_0^\infty \frac{a}{\sqrt{2\pi\lambda}} e^{-\frac{1}{2}\left(a^2\lambda + b^2\lambda^{-1}\right)} \mathrm{d}\lambda = e^{-|ab|} \ \ . \tag{7}$$

Now, we prove following theorem.

Theorem 1. *The pseudo-likelihood contribution from an observation y_n can be expressed as*

$$L\left(y_n \mid \mathbf{x}_n, f\right) = \int_0^\infty \frac{1}{\sqrt{2\pi\lambda_n}} e^{\frac{-1}{2\lambda_n}\left(1 + \lambda_n + \max_{t \neq y_n, t \in Y} f_t(\mathbf{x}_n) - f_{y_n}(\mathbf{x}_n)\right)^2} \mathrm{d}\lambda_n \tag{8}$$

Proof. Applying Lemma 1 while substituting $a = 1$ and $b = 1 + \max_{t \neq y_n, t \in Y} f_t\left(\mathbf{x}_n\right) - f_{y_n}\left(\mathbf{x}_n\right)$, multiplying through by e^{-b}, and using the identity $\max\left\{b, 0\right\} = \frac{1}{2}\left(|b| + b\right)$, we get,

$$\int_0^\infty \frac{1}{\sqrt{2\pi\lambda_n}} \exp\left(-\frac{1}{2}\frac{\left(b + \lambda_n\right)^2}{\lambda_n}\right) \mathrm{d}\lambda_n = e^{-2\max\{b, 0\}} \ \ . \tag{9}$$

3.3 Scalable Variational Inference for LMGP

We complete the model formulation by assuming that $f_j(\mathbf{x})$ is drawn from a Gaussian process for each class, j, i.e. $\mathbf{f}_j \sim \mathcal{N}(0, K_{NN})$ and $\boldsymbol{\lambda} \sim \mathbb{1}_{[0,\infty)}(\boldsymbol{\lambda})$.

Inference in our model amounts to learning the joint posterior $p(\mathbf{f}, \boldsymbol{\lambda}|D)$, where $\mathbf{f} = [\mathbf{f}_1, \dots, \mathbf{f}_C]$. However, computing the exact posterior is intractable. We use variational inference (VI) combined with an inducing point approximation for jointly learning the C GPs corresponding to each class. In VI, the exact posterior over the set of model parameters $\boldsymbol{\theta}$ is approximated by a variational distribution q. The parameters of q are updated with the aim to reduce the dissimilarity between the exact and approximate posteriors, as measured by the Kullback-Leibler divergence. This is equivalent to maximizing the evidence lower bound (ELBO) [12] with respect to parameters of q, where

$$\text{ELBO} = \mathbb{E}_{q(\boldsymbol{\theta})}\left[\log p\left(\mathbf{y}|\boldsymbol{\theta}\right)\right] - \text{KL}\left[q\left(\boldsymbol{\theta}\right)||p\left(\boldsymbol{\theta}\right)\right] \ . \tag{10}$$

Using this as objective function, we could potentially infer the posterior $q(\mathbf{f}, \boldsymbol{\lambda})$. However, inference and prediction using this full model involves inverting an $N \times N$ matrix. An operation of complexity $O(N^3)$ is impractical. Therefore, we employ the sparse approximation proposed by [10]. We augment the model with $P \ll N$ inducing points which are shared across all GPs. Similar to [10], we consider a GP prior for the inducing points, $p(\mathbf{u}_j) = \mathcal{N}\left(0, K_{PP}\right)$ and consider the marginal

$$q(\mathbf{f}_j) = \int p(\mathbf{f}_j|\mathbf{u}_j)q(\mathbf{u}_j)\mathrm{d}\mathbf{u}_j \tag{11}$$

with

$$p(\mathbf{f}_j|\mathbf{u}_j) = \mathcal{N}\left(\kappa\mathbf{u}, \tilde{K}\right) \ . \tag{12}$$

The approximate posterior $q(\mathbf{u}, \boldsymbol{\lambda})$ factorizes as

$$\prod_{j\in Y} q(\mathbf{u}_j) \prod_{n=1}^{N} q(\lambda_n) \tag{13}$$

with

$$q(\lambda_n) = \mathcal{GIG}(1/2, 1, \alpha_n),\ q(\mathbf{u}_j) = \mathcal{N}(\boldsymbol{\mu}_j, \Sigma_j) \ . \tag{14}$$

Here, $\kappa = K_{NP}K_{PP}^{-1}$, $\tilde{K} = K_{NN} - K_{NP}\kappa^T$ and $\mathcal{GIG}$ is the generalized inverse Gaussian. K_{PP} is the kernel matrix resulting from evaluating the kernel function between all inducing points. Analogously, we denote the cross-covariance between data points and inducing points, or between all data points by K_{NP} or K_{NN}, respectively. The choice of variational approximations is inspired from the exact conditional posterior computed by [9]. Using Jensen's inequality, we derive the

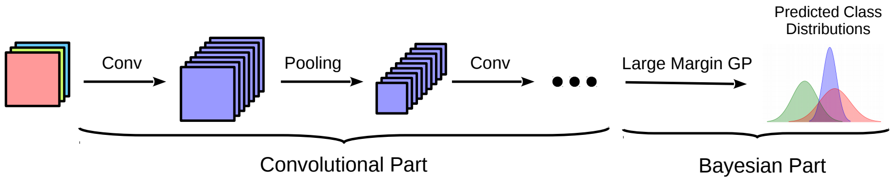


Fig. 1: LMGP-DNN for image classification.

final training objective,

$$\begin{aligned}
&\mathbb{E}_{q(\mathbf{u},\boldsymbol{\lambda})}\left[\log p\left(\mathbf{y}|\mathbf{u},\boldsymbol{\lambda}\right)\right]-\mathrm{KL}\left[q\left(\mathbf{u},\boldsymbol{\lambda}\right)||p\left(\mathbf{u},\boldsymbol{\lambda}\right)\right] && (15)\\
\geq&\mathbb{E}_{q(\mathbf{u},\boldsymbol{\lambda})}\left[\mathbb{E}_{p(\mathbf{f}|\mathbf{u})}\left[\log p\left(\mathbf{y},\boldsymbol{\lambda}|\mathbf{f}\right)\right]\right]+\mathbb{E}_{q(\mathbf{u})}[\log p\left(\mathbf{u}\right)]-\mathbb{E}_{q(\mathbf{u},\boldsymbol{\lambda})}[\log q(\mathbf{u},\boldsymbol{\lambda})] && (16)\\
=&\sum_{n=1}^{N}\left(-\frac{1}{2\sqrt{\alpha_n}}\left(2\tilde{K}_{n,n}+\left(1+\boldsymbol{\kappa}_n\left(\boldsymbol{\mu}_{t_n}-\boldsymbol{\mu}_{y_n}\right)\right)^2+\boldsymbol{\kappa}_n\varSigma_{t_n}\boldsymbol{\kappa}_n^{\intercal}+\boldsymbol{\kappa}_n\varSigma_{y_n}\boldsymbol{\kappa}_n^{\intercal}-\alpha_n\right)\right.\\
&\qquad\left.-\boldsymbol{\kappa}_n\left(\boldsymbol{\mu}_{t_n}-\boldsymbol{\mu}_{y_n}\right)-\frac{1}{4}\log\alpha_n-\log\left(B_{\frac{1}{2}}\left(\sqrt{\alpha_n}\right)\right)\right)\\
&\qquad-\frac{1}{2}\sum_{j\in Y}\left(-\log|\varSigma_j|+\mathrm{trace}\left(K_{PP}^{-1}\varSigma_j\right)+\boldsymbol{\mu}_j^{\intercal}K_{PP}^{-1}\boldsymbol{\mu}_j\right)=\mathcal{O} && (17)
\end{aligned}$$

where $B_{\frac{1}{2}}$ is the modified Bessel function [13], and $t_n = \arg\max_{t\in Y, t\neq y_n} f_t\left(\mathbf{x}_n\right)$. $\mathcal{O}$ is maximized using gradient-based optimization methods. We provide a detailed derivation of the variational objective and its gradients in the appendix.

3.4 LMGP-DNN

Deep Neural Networks (DNNs) are well known for their end-to-end learning capabilities for numerous tasks that involve unstructured data. Their effectiveness is often attributed to their capacity to learn hierarchical representation of data. In Section 3.3 we show that our proposed LMGP can be learned with gradient-based optimization schemes. This enables us to combine it with various deep learning components such as convolutional layers and extend its applicability to unstructured data as shown in Figure 1. The parameters of the LMGP-DNN model which includes convolution and the variational parameters are jointly learned by means of backpropagation. The ability to jointly learn features with the one-vs-rest Bayesian SVMs has been previously explored in [26] and [25]. LMGP-DNN explores the same for the multi-class case.

4 Experimental Evaluation

In this section we conduct an extensive study of the LMGP model and analyze its classification performance on structured and unstructured data. Additionally, we analyze the quality of its uncertainty prediction in a large-scale active learning experiment and for the challenging problem of adversarial image detection.

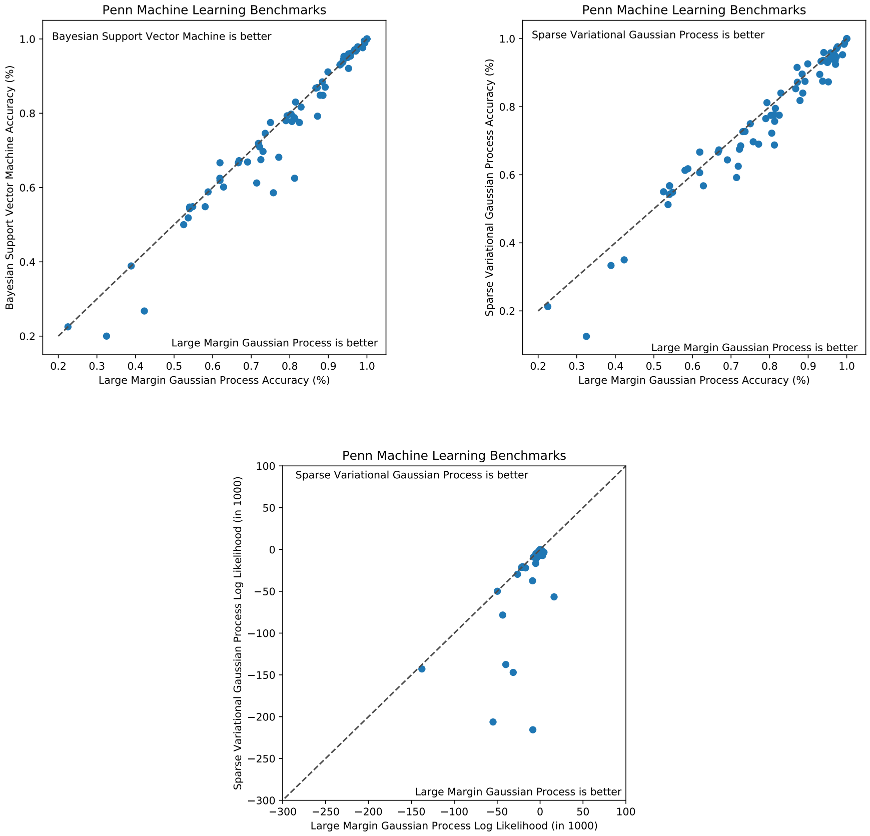


Fig. 2: Pairwise comparison of the LMGP versus the Bayesian SVM and SVGP. On average, LMGP provides better results.

Table 1: Mean average rank across 68 data sets. The smaller, the better. Our proposed LMGP is on average the most accurate prediction model.

Bayesian SVM	LMGP	SVGP
1.96	**1.68**	2.33

4.1 Classification

Our classification experiment is investigating two different types of data. In the first part, we investigate the classification performance of the multi-class Bayesian SVM on structured data against Bayesian state-of-the-art models. In the second part, we compare the hybrid Bayesian SVM model against standard convolutional neural networks for the task of image classification.

4.2 Structured Data Classification

We evaluate the proposed LMGP with respect to classification accuracy on the Penn Machine Learning Benchmarks [22]. From this benchmark, we select all multi-class classification data sets consisting of at least 128 instances. This subset consists of 68 data sets with up to roughly one million instances. We compare the classification accuracy of our proposed LMGP with the the scalable variational Gaussian process (SVGP) [10] and the most recently proposed binary Bayesian support vector machine (Bayesian SVM) [35] (one-vs-rest setup). We use the implementation available in GPflow [21] for SVGP and implement the one-vs-rest Bayesian SVM and LMGP as additional classifiers in GPflow by extending its classifier interface. The shared back end of all three implementations allows a fair training time comparison. For this experiment, all models are trained using 64 inducing points. Gradient-based optimization is performed using Adam [14] with an initial learning rate of $5 \cdot 10^{-4}$ for 1000 epochs.

Figure 2 contrasts the LMGP with SVGP and one-vs-rest Bayesian SVM. The proposed LMGP clearly outperforms the other two models for most data sets. While this is more pronounced against SVGP, the Bayesian SVM and LMGP models exhibit similar performance. This claim is supported by the comparison of mean ranks (Table 1). The rank per data set is computed by ranking the methods for each data set according to classification accuracy. The most accurate prediction model is assigned rank 1, second best rank 2 and so on. In case of ties, an average rank is used, e.g. if the models exhibit classification accuracies of 1.0, 1.0, and 0.8, they are assigned ranks of 1.5, 1.5, and 3, respectively.

One primary motivation for proposing LMGP is scalability. Classification using the one-vs-rest Bayesian SVM requires training an independent model per class which increases the training time by a factor equal to the number of classes. Contrastingly, SVGP and LMGP enable multi-class classification with a single model. This results in significant benefits in training time. As evident in Figure 3, the LMGP requires the least training time.

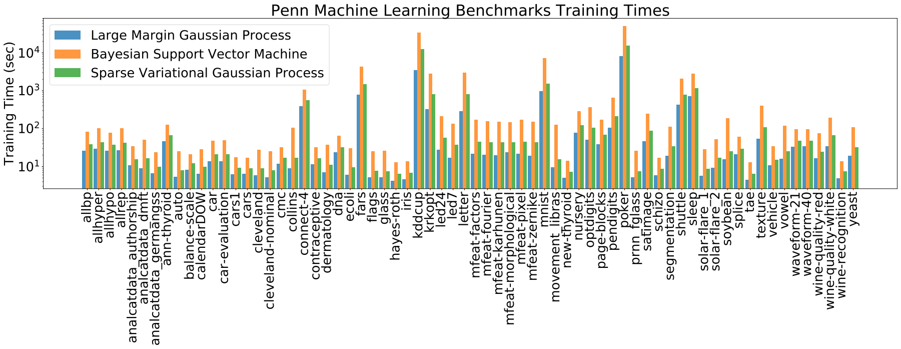


Fig. 3: Our proposed LMGP clearly needs less time than its competitors.

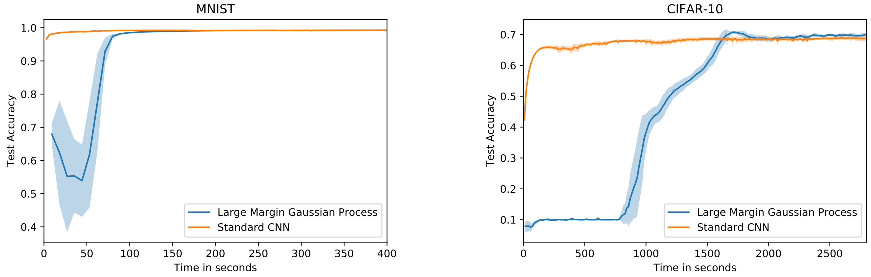


Fig. 4: The jointly learned model of a convolutional network and an LMGP performs as good as a standard network. The price of gaining a Bayesian neural network is a longer training time.

In conclusion, LMGP is the most efficient model without compromising on prediction accuracy. In fact, on average it has a higher accuracy.

4.3 Image Classification with LMGP-DNN

In Section 3.4 we describe how deep learning can be used to learn a feature representation jointly with an LMGP. Image data serves as a typical example for unstructured data. We compare the LMGP-DNN to a standard convolutional neural network (CNN) with a softmax layer for classification. We evaluate these models on two popular image classification benchmarks, MNIST [17] and CIFAR-10 [15].

We observe same performance of the LMGP-DNN as a standard CNN with softmax layer. The two different neural networks share the first set of layers, for MNIST: `conv(32,5,5)-conv(64,3,3)-max_pool-fc(1024)-fc(100)`, and for CIFAR-10: `conv(128,3,3)-conv(128,3,3)-max_pool-conv(128,3,3) -max_pool-fc(256)-fc(100)`. As in our previous experiment, we use Adam to perform the optimization.

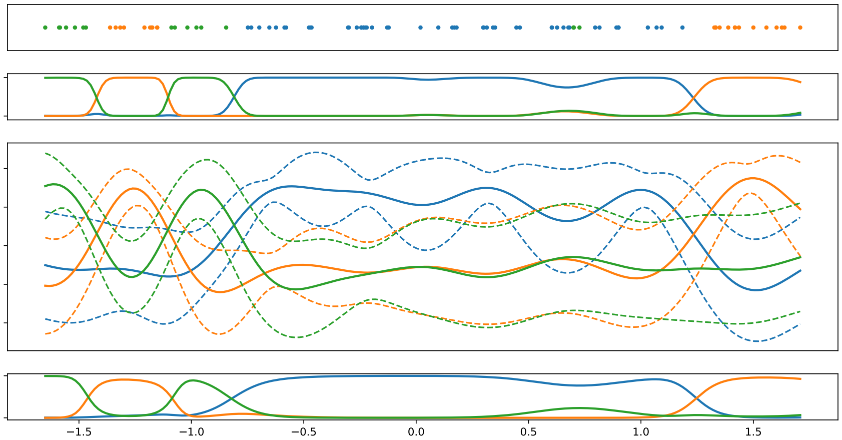


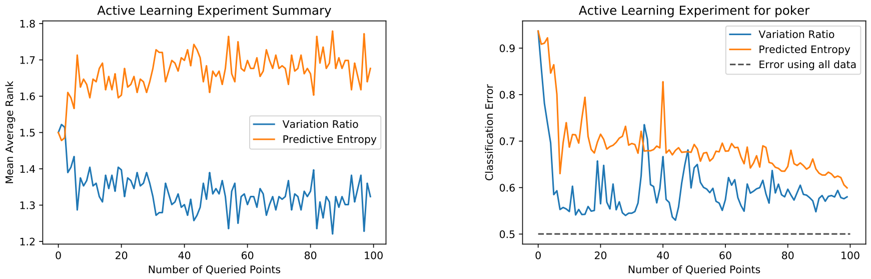
Fig. 5: From top to bottom: 1. Data points belonging to three classes, 2. Prediction probabilities from LMGP 3. Predictions from the three Gaussian processes of the LMGP model along with their uncertainties, and 4. SVM probability predictions

Figure 4 shows that the LMGP-DNN achieves the same test accuracy as the standard CNN. The additional training effort of a LMGP-DNN model pays off in achieving probabilistic predictions with uncertainty estimates. While the variational objective and the likelihood exhibits the expected behavior during the training, we note an odd behavior during the initial epochs. We suspect that this is due to initialization of parameters which could result in the KL-term of the variational objective dominating the expected log-likelihood.

4.4 Uncertainty Analysis

Most statistical modelling approaches are concerned with minimizing a specific loss-metric, e.g. classification error. However, practitioners have additional concerns, like interpretability and certainty of the predictions. Bayesian methods provide a distribution over predictions rather than just point-estimates, which is a significant advantage in practice as it allows for development of informed decision-making systems. Figure 5 shows that LMGP exhibits a key artefact of GPs where uncertainty in the predicted scores of GPs is higher (3rd row) in the regions with few datapoints. This aspect of our model is central to its utility in the tasks of active learning and adversarial detection and is often overlooked by classical models like SVMs (4th row in Figure 5). We want to emphasise that there are scenarios where uncertainty as obtained from Bayesian models is beneficial and that the prediction error by itself only plays a tangential role.

Active Learning Active learning is concerned with scenarios where the process of labeling data is expensive. In such scenarios, a query policy is adopted to label



(a) Average rank across 68 data sets.

(b) Representative results for the largest data set.

Fig. 6: The Bayesian query policy (variation ratio) decreases the error of the model faster and clearly outperforms the policy based on point-estimates only. For both figures, the smaller the better.

samples from a large pool of unlabeled instances with the aim to improve model performance. We contrast between two policies to highlight the merits of using prediction uncertainty obtained from the LMGP model. While the first policy utilizes both mean and variance of the predictive distribution of the LMGP, the second policy relies only on the mean. For this experiment we use the same data sets as specified in Section 4.2.

We use the variation ratio (VR) as the basis of a Bayesian query policy. It is defined by

$$\text{Variation Ratio} = 1 - F/S \ , \tag{18}$$

where F is the frequency of the mode and S the number of samples. The VR is the relative number of times the majority class is not predicted. Its minimum zero is reached when all Monte Carlo samples agree on the same class. The instance with highest VR is queried. We compare this to a policy which queries the instance with maximum entropy of class probabilities. These are computed using softmax over the mean predictions,

$$\mathbb{H}\left(f\left(\mathbf{x}_n\right)\right) = -\sum_{t \in Y} f_t\left(\mathbf{x}_n\right) \log\left(f_t\left(\mathbf{x}_n\right)\right) \ . \tag{19}$$

For a fair comparison, we use the same LMGP for both policies. Initially, one instance per class, selected uniformly at random, is labeled. Then, one hundred further instances are queried according to the two policies. As only few training examples are available, we modify the training setup by reducing the number of inducing points to four.

We report the mean average rank across 68 data sets for the two different query policies in Figure 6a. Since both policies start with the same set of labeled instances, the performance is very similar at the beginning. However, with increasing number of queried data points, the Bayesian policy quickly outperforms

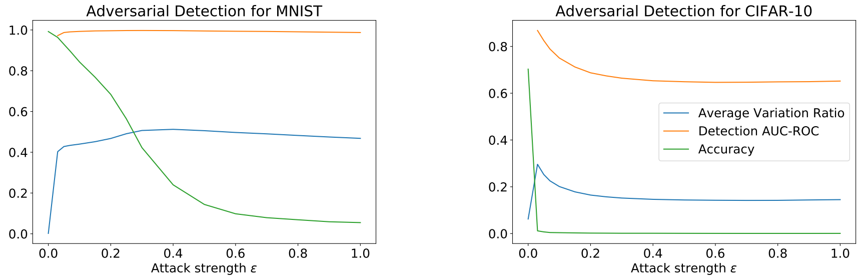


Fig. 7: The accuracy on adversarial images decreases with increasing attack strength. A significant increase of the average variation ratio indicates that it is a good feature to detect adversarial images.

the other policy. Of the 68 data sets, the *poker* data set, with more than one million instances, is the largest and consequently the most challenging. Within the first queries, we observe a large decrease in classification error as shown in Figure 6b. We note the same trend of mean ranks across the two policies. The small number of labeled instances is obviously insufficient to reach the test error of a model trained on all data points as shown by the dashed line.

Similarly, one could employ LMGP-DNN for active learning of unstructured data [7].

Adversarial Image Detection With the rise of Deep Learning, its security and reliability is a major concern. A recent development in this direction is the discovery of adversarial images [8]. These correspond to images obtained by adding small imperceptible adversarial noise resulting in high confidence misclassification. While various successful attacks exist, most defense and detection methods do not work [4]. However, [4] acknowledge that the uncertainty obtained from Bayesian machine learning models is the most promising research direction. Several studies show that Bayesian models behave differently for adversarial examples compared to the original data [3, 19, 30].We take a step further and use the variation ratio (VR) determined by the LMGP, as defined in Equation (18), for building a detection model for adversarial images.

We attack the LMGP-DNN described in Section 4.3 with the popular Fast Gradient Sign Method (FGSM) [8]. We generate one adversarial image per image in the test set. We present the results for detection and classification under attack in Figure 7. LMGP-DNN is not robust to FGSM since its accuracy drops with increasing attack strength ϵ. However, the attack does not remain unperceived. The VR rapidly increases and enables the detection of adversarial images. The ranking of original and adversarial examples with respect to VR yields an ROC-AUC of almost 1 for MNIST. This means that the VR computed for any original example is almost always smaller than the one computed for any adversarial example.

CIFAR-10 exhibits different results under the same setup. Here, the detection is poor and it significantly worsens with increasing attack strength. Potentially, this is an artifact of the poor classification model for CIFAR-10. In contrast to the MNIST classifier, this model is under-confident on original examples. Thus, a weaker attack succeeds in reducing the test accuracy to 1.16%. We believe a better network architecture combined with techniques such as data augmentation will lead to an improved performance in terms of test accuracy and subsequently better detection. Nevertheless, the detection performance of our model is still better than a random detector, even for the strongest attack.

5 Conclusions

We devise a pseudo-likelihood for the generalised multi-class hinge loss leading to the large margin Gaussian process model. Additionally, we derive a variational training objective for the proposed model and develop a scalable inference algorithm to optimize it. We establish the efficacy of the model on multi-class classification tasks with extensive experimentation on structured data and contrast its accuracy to two state-of-the-art competitor methods. We provide empirical evidence that our proposed method is on average better and up to an order of magnitude faster to train. Furthermore, we extend our formulation to a LMGP-DNN and report comparable accuracy to standard models for image classification tasks. Finally, we investigate the key advantage of Bayesian modeling in our approach by demonstrating the use of prediction uncertainty in solving the challenging tasks of active learning and adversarial image detection. The uncertainty-based policy outperforms its competitor in the active learning scenario. Similarly, the uncertainty-enabled adversarial detection shows promising results for image data sets with near-perfect performance on MNIST.

References

1. Andrews, D.F., Mallows, C.L.: Scale mixtures of normal distributions. Journal of the Royal Statistical Society. Series B (Methodological) **36**(1), 99–102 (1974), http://www.jstor.org/stable/2984774
2. Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: A review for statisticians. CoRR **abs/1601.00670** (2016), http://arxiv.org/abs/1601.00670
3. Bradshaw, J., de G. Matthews, A.G., Ghahramani, Z.: Adversarial examples, uncertainty, and transfer testing robustness in gaussian process hybrid deep networks (2017)
4. Carlini, N., Wagner, D.: Adversarial examples are not easily detected: Bypassing ten detection methods (2017)
5. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. Journal of Machine Learning Research **2**, 265–292 (2001), http://www.jmlr.org/papers/v2/crammer01a.html
6. Dogan, Ü., Glasmachers, T., Igel, C.: A unified view on multi-class support vector classification. Journal of Machine Learning Research **17**, 45:1–45:32 (2016), http://jmlr.org/papers/v17/11-229.html

7. Gal, Y., Islam, R., Ghahramani, Z.: Deep bayesian active learning with image data. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. pp. 1183–1192 (2017), http://proceedings.mlr.press/v70/gal17a.html
8. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples (2014)
9. Henao, R., Yuan, X., Carin, L.: Bayesian nonlinear support vector machines and discriminative factor modeling. In: Advances in Neural Information Processing Systems. pp. 1754–1762 (2014)
10. Hensman, J., de G. Matthews, A.G., Ghahramani, Z.: Scalable variational gaussian process classification. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San Diego, California, USA, May 9-12, 2015 (2015), http://jmlr.org/proceedings/papers/v38/hensman15.html
11. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. J. Global Optimization **13**(4), 455–492 (1998). https://doi.org/10.1023/A:1008306431147, https://doi.org/10.1023/A:1008306431147
12. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Machine Learning **37**(2), 183–233 (1999). https://doi.org/10.1023/A:1007665907178, https://doi.org/10.1023/A:1007665907178
13. Jørgensen, B.: Statistical properties of the generalized inverse Gaussian distribution. No. 9 in Lecture notes in statistics, Springer, New York, NY [u.a.] (1982)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014), http://arxiv.org/abs/1412.6980
15. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep. (2009)
16. Kuss, M., Rasmussen, C.E.: Assessing approximate inference for binary gaussian process classification. Journal of Machine Learning Research **6**, 1679–1704 (2005), http://www.jmlr.org/papers/v6/kuss05a.html
17. LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010), http://yann.lecun.com/exdb/mnist/
18. Li, L., Chu, W., Langford, J., Schapire, R.E.: A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010. pp. 661–670 (2010). https://doi.org/10.1145/1772690.1772758, http://doi.acm.org/10.1145/1772690.1772758
19. Li, Y., Gal, Y.: Dropout inference in bayesian neural networks with alpha-divergences. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. pp. 2052–2061 (2017), http://proceedings.mlr.press/v70/li17a.html
20. Luts, J., Ormerod, J.T.: Mean field variational bayesian inference for support vector machine classification. Computational Statistics & Data Analysis **73**, 163–176 (2014). https://doi.org/10.1016/j.csda.2013.10.030, https://doi.org/10.1016/j.csda.2013.10.030
21. Matthews, A.G.d.G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., Hensman, J.: GPflow: A Gaussian process library using TensorFlow. Journal of Machine Learning Research **18**(40), 1–6 (apr 2017), http://jmlr.org/papers/v18/16-537.html

22. Olson, R.S., La Cava, W., Orzechowski, P., Urbanowicz, R.J., Moore, J.H.: Pmlb: a large benchmark suite for machine learning evaluation and comparison. BioData Mining **10**(1), 36 (Dec 2017). https://doi.org/10.1186/s13040-017-0154-4, https://doi.org/10.1186/s13040-017-0154-4
23. Perkins, H., Xu, M., Zhu, J., Zhang, B.: Fast parallel svm using data augmentation. arXiv preprint arXiv:1512.07716 (2015)
24. Polson, N.G., Scott, S.L., et al.: Data augmentation for support vector machines. Bayesian Analysis **6**(1), 1–23 (2011)
25. Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A., Carin, L.: Variational autoencoder for deep learning of images, labels and captions. In: Advances in neural information processing systems. pp. 2352–2360 (2016)
26. Pu, Y., Yuan, W., Stevens, A., Li, C., Carin, L.: A deep generative deconvolutional image model. In: Artificial Intelligence and Statistics. pp. 741–750 (2016)
27. Rasmussen, C.E., Williams, C.K.I.: Gaussian processes for machine learning. Adaptive computation and machine learning, MIT Press (2006), http://www.worldcat.org/oclc/61285753
28. Seeger, M.: Bayesian model selection for support vector machines, gaussian processes and other kernel classifiers. In: Advances in neural information processing systems. pp. 603–609 (2000)
29. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2009)
30. Smith, L., Gal, Y.: Understanding measures of uncertainty for adversarial example detection. arXiv preprint arXiv:1803.08533 (2018)
31. Snelson, E., Ghahramani, Z.: Sparse gaussian processes using pseudo-inputs. In: Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]. pp. 1257–1264 (2005), http://papers.nips.cc/paper/2857-sparse-gaussian-processes-using-pseudo-inputs
32. Snoek, J., Larochelle, H., Adams, R.P.: Practical bayesian optimization of machine learning algorithms. In: Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States. pp. 2960–2968 (2012), http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms
33. Sollich, P.: Probabilistic methods for support vector machines. In: Advances in neural information processing systems. pp. 349–355 (2000)
34. Titsias, M.K.: Variational learning of inducing variables in sparse gaussian processes. In: Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS 2009, Clearwater Beach, Florida, USA, April 16-18, 2009. pp. 567–574 (2009), http://www.jmlr.org/proceedings/papers/v5/titsias09a.html
35. Wenzel, F., Galy-Fajou, T., Deutsch, M., Kloft, M.: Bayesian nonlinear support vector machines for big data. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 307–322. Springer (2017)
36. Williams, C.K.I., Barber, D.: Bayesian classification with gaussian processes. IEEE Trans. Pattern Anal. Mach. Intell. **20**(12), 1342–1351 (1998). https://doi.org/10.1109/34.735807, https://doi.org/10.1109/34.735807