Scalable Gaussian Process for Extreme Classification

Dhaka, Akash Kumar; Andersen, Michael Riis; Moreno, Pablo Garcia; Vehtari, Aki

Published in:
Proceedings of 2020 IEEE 30th International Workshop on Machine Learning for Signal Processing

Link to article, DOI:
10.1109/MLSP49062.2020.9231675

Publication date:
2020

Document Version
Peer reviewed version

SCALABLE GAUSSIAN PROCESS FOR EXTREME CLASSIFICATION

Akash Kumar Dhaka[1]*, Michael Riis Andersen*[2]*, Pablo Garcia Moreno*[3]*, Aki Vehtari*[1]

[1]Aalto University, Dept. of Computer Science, [2] DTU Compute, Technical University of Denmark
[3]Amazon.com

ABSTRACT

We address the limitations of Gaussian processes for multiclass classification in the setting where both the number of classes and the number of observations is very large. We propose a scalable approximate inference framework by combining the inducing points method with variational approximations of the likelihood that have been recently proposed in the literature. This leads to a tractable lower bound on the marginal likelihood that decomposes into a sum over both data points and class labels, and hence, is amenable to doubly stochastic optimization. To overcome memory issues when dealing with large datasets, we resort to amortized inference, which coupled with subsampling over classes reduces the computational and the memory footprint without a significant loss in performance. We demonstrate empirically that the proposed algorithm leads to superior performance in terms of test accuracy, and improved detection of tail labels.

Index Terms— Gaussian process classification, variational inference, augmented model.

1. INTRODUCTION

Multiclass classification refers to the supervised learning problem where each instance is labelled with a value chosen from a discrete set with cardinality $K > 2$. The goal of multiclass classification is to learn a mapping from an input space to the set of labels based on a set of input-output pairs $(\mathbf{x}_n, y_n)$, where $\mathbf{x}_n \in \mathbb{R}^D$ and $y_n \in \{1, 2, ..., K\}$. Extreme classification (EC) [1, 2] deals with the complexity introduced when the number of classes K is extremely large so that evaluation of the likelihood becomes prohibitively expensive using standard inference techniques. For example, consider the softmax function which maps K function values to a probability vector,

$$p(y = c|\mathbf{f}) = \frac{\exp(f_c)}{\sum_{i=1}^{K} \exp(f_i)}, \tag{1}$$

where $\mathbf{f} = [f_1, \ldots, f_K]$ is a vector of scores for each class for a given observation. Evaluating Eq. (1) and its gradients scales linearly with K. For very large data sets, this motivates the search for sub-linear, efficient, and accurate approximations.

Besides the computational challenges, the statistical challenges include 1) the average number of observations per class, N/K is small, 2) sparse data for a subset of classes, and 3) class imbalance in general. Bayesian methods in the setting where K is large, have received less attention than standard multi-class classification. Recently, Bayesian inference algorithms for extreme classification have been proposed for linear models [3, 4, 5].

While linear models have been shown to scale to very big data sets, non-linear models such as Gaussian processes (GPs) [6] can provide better performance by modeling non-linearities and covariate interactions. In the context of multi-class classification, imposing GP priors on each score function, f_i for $i = 1, \ldots, K$, allows modelling complex and non-linear dependencies in a probabilistic framework. Naive computations for GPs scale cubically with number of data points N, and for K-class GP classification the computation scales as $\mathcal{O}(KN^3)$. This makes it computationally non-trivial to apply GPs to scenarios where K is large.

There has been extensive work on how to reduce the computational cost arising due to large N, including sparse GPs using the inducing points framework [7, 8]. This reduces the computational cost per GP to $\mathcal{O}(\mathcal{B}M^2 + M^3)$, where M is the number of inducing points and $\mathcal{B}$ is the mini-batch size.

We propose a scalable GP framework for extreme classification by combining sparse GPs with recently proposed variational approximations of the likelihood terms. In particular, we study two different approximations: the One-vs-Each (OVE) approximation [4] and the *augment and reduce* (AR) approximation [5]. This allows us to approximate the likelihood and gradient for each observation using a small subset of the $(K - 1)$ negative classes such that the resulting cost will be independent of K. While AR offers better empirical performance than OVE, it introduces a set of local variational parameters for each observation. Since the number of variational parameters scales with N, the memory footprint can be prohibitively large for large datasets. We resolve this issue using amortized inference, where a neural network (NN) learns a mapping from the input space to the variational parameters. The NN is learnt jointly with the hyperparameters of the GP. We show that this solution does not degrade the performance of the AR approximation, but it keeps the memory footprint

constant with respect to N. In addition, the optimisation problem is simplified as we tie up local parameters. Overall, this variational approximation performs better than previous GP approaches in literature on 4 out of 5 datasets in terms of accuracy and coverage. Finally, we share insights into how these likelihoods are related to each other.

1.1. Related Work

Relevant work on multiclass classification include [9, 10]. [10] use expectation propagation (EP) and an OVE-style bound that uses the probit function instead of the logistic function. EP is a fixed point algorithm which is hard to scale when the number of outcomes is large (in contrast to SVI). It does not offer a bound on the marginal likelihood, and it can suffer from convergence issues. Earlier variational approximations using augmented variables [11, 12] lack scalability. Sampling the latent function as done in [13] is not scalable for large K.

2. BACKGROUND

2.1. Gaussian processes for classification

GPs provide a principled way of imposing prior distributions over function spaces. We consider the problem where we have D-dimensional input vectors $\mathbf{x}_n \in \mathbb{R}^D$ associated with target class labels $y_n \in \{1, \dots, K\}$ for $n = 1, \dots, N$. We model the latent score function for each class $f_i \sim \mathcal{GP}(0, k)$ using a GP prior with covariance function $k(\cdot, \cdot, \boldsymbol{\theta})$. Given the set of input vectors $\mathbf{x}_n$, the joint prior distribution on the latent variables is given as

$$p(\mathbf{F}) = \prod_{i=1}^{K} p(\mathbf{f}^i), \quad p(\mathbf{f}^i) = \mathcal{N}(\mathbf{f}^i|\mathbf{0}, \mathbf{K}_{ff}),$$

where $\mathbf{f}^i = [f_i(\mathbf{x}_1), \dots, f_i(\mathbf{x}_N)]$ and $[\mathbf{K}_{ff}]_{nm} = k(\mathbf{x}_n, \mathbf{x}_m, \boldsymbol{\theta})$. We will use $\mathbf{f}_n = [f_1(\mathbf{x}_n), \dots, f_K(\mathbf{x}_n)]$ to denote the values of the latent functions for the n'th data point $\mathbf{x}_n$. We apply a link function $g : \mathcal{I}^k \mapsto \mathbb{R}^k$ that maps the probabilities of a categorical distribution that live in a K dimensional simplex $\mathbf{p}_n \in \mathcal{I}^K$ to the $\mathbf{f}_n \in \mathbb{R}^K$. The generative process is then

$$y_n \sim \mathrm{Cat}(\mathbf{p}_n), \quad \mathbf{p}_n = g^{-1}(\mathbf{f}_n).$$

2.2. Inducing points and Stochastic Variational Inference

The coupled training points can be made conditionally independent given a set of inducing points $\mathbf{z}$ living in the same space as $\mathbf{x}$ [7, 14]. We augment the model with inducing output variables for each class, $\mathbf{u}^i = [f^i(\mathbf{z}_1), \dots, f^i(\mathbf{z}_M)]$, i.e. the latent functions evaluated at the inducing points $\mathbf{z}$. The joint model for $(\mathbf{y}, \mathbf{f}, \mathbf{u})$ is then

$$\mathbf{u}^i \sim \mathcal{N}(\mathbf{0}, \mathbf{K_{uu}}), \tag{2}$$

$$\mathbf{f}^i|\mathbf{u}^i \sim \mathcal{N}(\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{u}^i, \mathbf{K}_{ff} - \mathbf{Q}_{ff}), \tag{3}$$

$$y_n \sim \mathrm{Cat}(g^{-1}(\mathbf{f}_n)), \tag{4}$$

where $\mathbf{Q}_{ff} = \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}$. The matrices $[\mathbf{K}_{uu}]_{ij} = k(\mathbf{z}_i, \mathbf{z}_j; \boldsymbol{\theta})$ and $[\mathbf{K}_{fu}]_{ij} = k(\mathbf{x}_i, \mathbf{z}_j; \boldsymbol{\theta})$ are the covariance matrix between inducing points and the cross-covariance matrix between the training points and inducing points, respectively.

Based on this generative model, [14] proposed to approximate the posterior distribution $p(\mathbf{F}, \mathbf{U}|\mathbf{X}, \mathbf{y})$ by $\prod_i q(\mathbf{f}^i, \mathbf{u}^i) = \prod_i p(\mathbf{f}^i|\mathbf{u}^i)q(\mathbf{u}^i)$, where $q(\mathbf{u}^i)$ is a variational multivariate Gaussian distribution $q(\mathbf{u}^i) = \mathcal{N}(\mathbf{u}^i|\mathbf{m}^i, \mathbf{S}^i)$. The variational parameters $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}^i\}_{i=1}^K$, where $\boldsymbol{\lambda}^i = \{\mathbf{m}^i, \mathbf{S}^i\}$, and the kernel parameters θ are estimated by maximizing the evidence lower bound (ELBO)

$$\begin{aligned}\mathrm{ELBO}(\boldsymbol{\lambda}, \boldsymbol{\theta}, \boldsymbol{\eta}) = &\sum_{i=1}^{K} -\mathrm{KL}(q(\mathbf{u}^i)||p(\mathbf{u}^i)) + \\ &\sum_{n=1}^{N} \mathrm{E}_{q(\mathbf{f_n}|\boldsymbol{\lambda})} \log p(y_n|\mathbf{f_n}).\end{aligned} \tag{5}$$

Since the approximate posterior distribution $\prod_i q(\mathbf{f}^i|\boldsymbol{\lambda}) = \prod_i \int p(\mathbf{f}^i|\mathbf{u}^i)q(\mathbf{u}^i)d\mathbf{u}^i$ is a multivariate Gaussian, the marginals $q(\boldsymbol{f_n}|\boldsymbol{\lambda})$ are analytically available

$$q(\boldsymbol{f_n}) = \prod_i \mathcal{N}(f_n^i|m_n^i, (\sigma_n^i)^2), \tag{6}$$

$$m_n^i = \mathbf{k}_{nu}\mathbf{K}_{uu}^{-1}\mathbf{m}^i, \tag{7}$$

$$(\sigma_n^i)^2 = k_{nn} + \mathbf{k}_{nu}\mathbf{K}_{uu}^{-1}(\mathbf{S}^i - \mathbf{K}_{uu})\mathbf{K}_{uu}^{-1}\mathbf{k}_{un}. \tag{8}$$

The key idea is that conditioned on the inducing points, the training points become decoupled and the bound can be maximized using stochastic optimization. The ELBO contains two terms: the first is the sum of KL divergences between the prior distribution and $q(\mathbf{u}^i)$ for each class, which can be computed analytically. The second term is the sum of expectations of log likelihoods with respect to the vector of latent score function values $\mathbf{f_n} = [f^1, \dots, f^K]$ at datapoint $\mathbf{x}_n$.

3. APPROXIMATE OBSERVATION MODELS

The second term of Eq. (5) involves a set of intractable expectations. In the binary classification unidimensional expectations can be approximated using quadrature methods [13]. In the multiclass scenario the link function $g(\cdot)$ couples all the latent variables $\mathbf{f}_n$, and for large K the high-dimensional integrals are not feasible with quadrature methods.

In this work, we consider two different approximations of the likelihood, where the high-dimensional integrals are replaced with a product of $(K-1)$ uni-dimensional integrals, each constituting a function operating on the pairwise differences $f_n^{ci} = f_n^c - f_n^i$ between the latent function values belonging to the target class c and one of the remaining classes i. As a result, we get approximations of Eq. (5) that also

decomposes as a sum over classes

$$\text{ELBO}(\boldsymbol{\lambda}, \boldsymbol{\theta}, \boldsymbol{\eta}) \approx \sum_{i=1}^{K} \Big[-\text{KL}(q(\mathbf{u}^i)||p(\mathbf{u}^i)) + \sum_{n=1}^{N} \text{E}_{q(f_n^{ci})} \log p(y_n|f_n^{ci}) \Big]. \tag{9}$$

Since $q(f_n^{ci})$ are univariate Gaussians, the expectations in Eq. (9) can be efficiently approximated by quadrature. In addition, this decomposition is amenable to stochastic optimization, making it possible to process only a random subset of the negative classes $\mathcal{S}_n \subseteq \{1, \dots, K\} \backslash c$, where c is the target class as in Eq. (10). This enables sparse updates

$$\text{ELBO}(\boldsymbol{\lambda}, \boldsymbol{\theta}, \boldsymbol{\eta}) \approx \sum_{n=1}^{N} \frac{K-1}{|\mathcal{S}_n|} \sum_{i \in \mathcal{S}_n} \Big(-\frac{1}{N} \text{KL} (q(\mathbf{u}^i)||p(\mathbf{u}^i)) + \text{E}_{q(f_n^{ci})} \log p(y_n|f_n^{ci}) \Big) \tag{10}$$

with constant computational complexity $\mathcal{O}(1)$ wrt. K. We choose $|\mathcal{S}_n| \ll K$, so that at each optimisation step, we make fewer updates to parameters reducing number of operations and memory footprint.

Next we describe the two different approximations for the likelihood: the One-vs-Each (OVE) approximation, and the Augment and Reduce (AR) approximation.

3.1. One-vs-Each (OVE)

The OVE approximation is done by replacing the exact probability by a lower bound based on pairwise probabilities corresponding to the event $y_n = c$ conditioned on the event that y_n takes one of the two labels $y_n \in \{c, k\}$ [4]. The joint log-likelihood function for the OVE approximation for the n'th observation is given by (see [4] for more details)

$$\log P(y_n = c|\mathbf{f}_n) = \log \frac{1}{1 + \sum_{i \neq c} e^{f_i - f_c}} \geq \log \prod_{i \neq c} \frac{1}{1 + e^{f_i - f_c}} = \sum_{i \neq c} \log \sigma(f_n^{ci}),$$

where the inequality follows from the fact that $(1 + \sum_i p_i) \leq \prod_i (1 + p_i)$ for $0 \leq p_i \leq 1$. Combining this bound with simple random sampling of the negative classes and substituting it into Eq. (10) yields the following approximate lower bound

$$\mathcal{L}_{\text{ove-sgd}} = \sum_{n=1}^{N} \frac{K-1}{|\mathcal{S}_n|} \sum_{i \in \mathcal{S}_n} \Big[-\frac{1}{N} \text{KL}(q(\mathbf{u}^i)||p(\mathbf{u}^i)) + \text{E}_{q(f_n^{ci})} \log \sigma(f_n^{ci}) \Big]. \tag{11, 12}$$

The stochastic OVE bound is an an unbiased estimate of the full OVE bound, but it is biased with respect to the original objective in Eq. (1) [3].

3.2. Augment and Reduce (AR)

Ruiz et al. [5] introduced a family of variational bounds for categorical likelihoods under the name of *augment and reduce* (A&R). The likelihood $p(y_n = c|\mathbf{f}_n)$ is augmented with a set of auxiliary variables $\boldsymbol{\epsilon}_n = [\epsilon_n^1, \dots, \epsilon_n^K]$ such that

$$p(y_n = c|\mathbf{f}_n) = \int_{-\infty}^{\infty} \phi(\epsilon_n^c) \prod_{i \neq c} \Phi(f_n^c - f_n^i + \epsilon_n^i) d\epsilon_n^i, \tag{13}$$

where $\phi(\cdot), \Phi(\cdot)$ are the PDF and CDF of the auxiliary variables, respectively. The integral is intractable in general, but can be approximated with the following variational bound with respect to a variational distribution $q(\epsilon_n)$

$$\log p(y_n|\mathbf{f}_n) \geq \text{E}_{q(\epsilon_n)} \Big[\log \frac{p(\epsilon_n)}{q(\epsilon_n)} + \frac{(K-1)}{|\mathcal{S}_n|} \sum_{i \in \mathcal{S}_n} \log \Phi(\epsilon_n + f_n^c - f_n^i) \Big]. \tag{14}$$

Thus, having a tractable CDF is a requirement for this approximation. The choices of the distributions for $\phi(\epsilon_n)$ and $q(\epsilon_n)$ determine the form of the likelihood. In this paper, we explore the following two specific choices: the logit and the softmax bounds [5].

3.2.1. AR Logit Bound

Choosing $\phi(\epsilon_n)$ to be the standard logistic distribution leads to the so called AR-logit bound on Eq. (13)

$$\log p(y_n|\mathbf{f}_n) \geq \text{E}_{q(\epsilon)} \Big[\log \frac{\sigma(\epsilon)\sigma(-\epsilon)}{q(\epsilon)} + \frac{(K-1)}{|\mathcal{S}_n|} \sum_{i \in \mathcal{S}_n} \log \sigma(\epsilon + f_n^c - f_n^i) \Big]. \tag{15}$$

While the second term in the bound is intractable, we can use the reparameterization trick to approximate the expectation. Substituting this bound into Eq. (10) yields a lower bound that decomposes over classes. We will refer to this lower bound as $\mathcal{L}_{\text{arlogit}}$. The essence of the AR bound is that the K GPs, which are independent a priori, become coupled by the auxiliary variable for each data point. Assuming a Dirac delta distribution for ϵ centered at zero, the AR-logit bound collapses to the OVE bound plus a constant in Eq. (11). This generalises the OVE bound.

3.2.2. AR Softmax Bound

The equivalent AR bound for the softmax can be derived by substituting a standard Gumbel distribution for $\phi(\epsilon_n)$ in Eq. (14). By also choosing a Gumbel for the variational distribution $q(\epsilon_n)$, the general form of the bound given in Eq. (13)

simplifies to Eq. (16), since the expectation has an analytical solution

$$\log p(y_n|\mathbf{f}_n) \geq 1 - \log(\alpha) - \frac{1}{\alpha}\Big(1+ \frac{(K-1)}{|\mathcal{S}_n|}\sum_{k\in\mathcal{S}_n}\exp(f_n^k - f_n^c)\Big). \quad (16)$$

Optimizing the variational parameter $\alpha \in [1, \infty)$ will provide a tighter bound to the softmax likelihood Eq. (1) than the OVE and OVE-SGD bounds. Unlike in the previous bounds, the expectation of Eq. (16) with respect to the marginals $q(f_n^{ci})$ given in Eq. (6) can be computed in closed form

$$\mathrm{E}_{q(\boldsymbol{f_n})}\left[\log p(y_n|\mathbf{f}_n)\right] \geq 1 - \log(\alpha_n) - \frac{1}{\alpha_n}\Big(1 + \frac{(K-1)}{|\mathcal{S}_n|}\sum_{i\in\mathcal{S}_n}\exp(-m_n^{ci} + \frac{(\sigma_n^{ci})^2}{2})\Big), \quad (17)$$

where $m_n^{ci} = \mathrm{E}_{q(f_n^{ci})}\left[f_n^{ci}\right]$ and $\sigma_n^{ci} = \mathrm{Var}_{q(f_n^{ci})}\left[f_n^{ci}\right]$. Thus, this method does not require one-dimensional quadratures like in the ARlogit and OVE bound described above, hence removing the bias introduced by them [15].

3.3. OPTIMIZATION AND AMORTIZED INFERENCE

We optimize all the bounds introduced in section 3 with respect to both the variational parameters $\boldsymbol{\lambda}$ and the kernel parameters $\boldsymbol{\theta}$ using the ADAM optimizer with mini-batching. The OVE aproximation Eq. (11) is parameter free, but both AR approximations (Eq. (15) and (16)) introduce additional parameters in the ELBO due to the presence of the local variational distributions. This increases the dimensionality of the optimization problem, increasing the chance that the optimizer will get trapped in a local minima or a saddle point. To solve this problem, [5] proposes a nested loop approach in which they update the local variational parameters of a batch in a local/inner loop, re-estimate the ELBO quantity for this batch and then update the kernel parameters and $q(U;\lambda)$ parameters. The approach still needs to store the $\mathcal{O}(N)$ variational parameters. We refer to this scheme as the Inner-Loop-method (IL).

In contrast, we propose an amortized scheme (AMO) that reduces the memory footprint by embedding the constraint that similar data points which lie close to each other in the input space are likely to have similar auxiliary variables, and by extension similar variational parameters. We model ϵ_n as

$$\epsilon_n \sim q(\epsilon_n|\eta_n), \quad \eta_n = u(\boldsymbol{x}_n; \tilde{\boldsymbol{\lambda}}),$$

where η is the augmented variable parameterised by μ, β in the ARLOGIT bound and α in the ARSOFT bound. The map u can be any non-linear map from the input space to the variational parameters. In this work, we use a neural network with two hidden layers. The strength of the similarity constraint is controlled by the complexity and size of the network. Since the parameters are tied through by sharing of network weights, the optimisation problem is simplified.

4. EXPERIMENTS AND RESULTS

We evaluate the different methods empirically based on several benchmark datasets. For all datasets, we standardize by subtracting mean and dividing by standard deviations. BibTeX [1], Mediamill, Delicious [16] are all multilabel datasets which means that each datapoint may have more than one label assigned to it. We pick the first label for each datapoint as done in [5, 4]. This lowers the final number of classes for the last three datasets as given in Table 1. The mean values of $q(\mathbf{U})$ for each class are initialized randomly from $\mathcal{N}(0.1, 0.5)$ and the covariance matrix was initialised as an identity matrix.

4.1. Performance Metrics

We quantify the performance of the proposed methods with the classification accuracy and the *coverage*, motivated by the extreme learning community [17]. When the distribution of class labels are severely imbalanced, the classification performance for the infrequent classes will not be clearly reflected in the accuracy metric. It is given as the percentage of classes in test-set which have a non-zero number of true positives,

$$\text{Coverage} = K^{\text{TP}}/K^* \quad (18)$$

where K^* represents the number of classes in the test set and K^{TP} is the number of classes with at least one true positive.

4.2. Baselines methods

Since most extreme classification methods, such as DISMEC [17] and PPD-Sparse [18], are based on linear models, we include linear models for both the OVE and AR-soft likelihood as baselines. We also compare our methods against two multi-class GP methods from the literature: the Robust-Max (GP-RM) likelihood [19], which was introduced for making models more robust to outliers, and Villa/Hernandez-Lobato likelihood (GP-HL), which can be derived in two ways by either taking the limit of noise parameter to zero in GP-RM, or by replacing the sigmoid function with a Gaussian CDF in the $\mathcal{L}_{\text{ove}}$ approximation. The computations are carried out using the GPFlow implementation [10].

4.3. Results

Table 1 compares the performance of the baseline methods with the proposed methods. The proposed GP methods perform better than linear models for all datasets except for the Delicious and Mediamill dataset, where the performance is similar to linear model. The ARSOFT approximation performs better than the rest on the first three datasets.

The experiments show that the AR methods generally perform better than both the non-stochastic and stochastic OVE methods when the number of negative class samples is fixed. The difference is more pronounced when K is large and $|\mathcal{S}|$ is

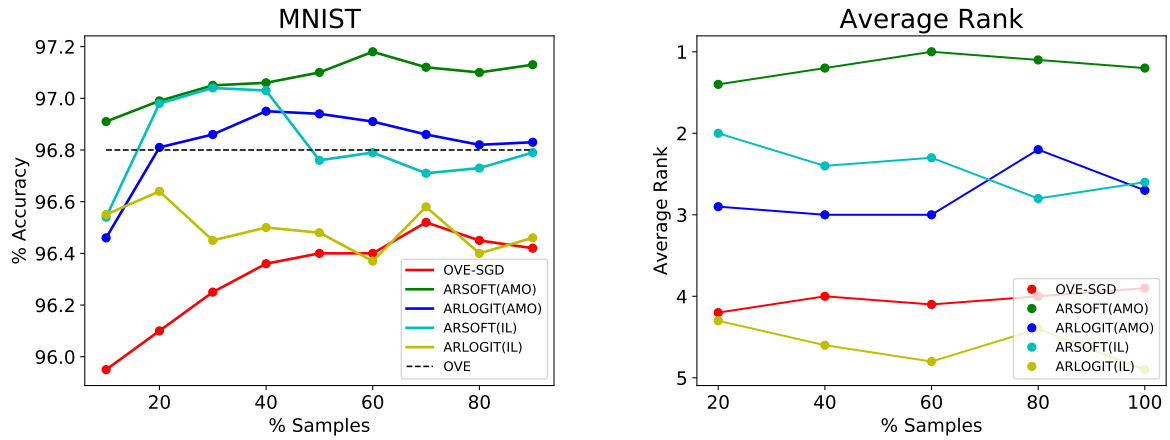

Fig. 1: The plot on the left shows the test set classification accuracy (higher is better) for the MNIST dataset as a function the sample size for the negative classes. The optimisation scheme is mentioned in parantheses. The plot on the right is a ranking plot from 1 to 5 (1 being best, 5 being lowest) for the different likelihood approximations and optimisation schemes for all datasets considered.

Name	N	K	Linear		GP-RM, GP-HL	OVE	$\lvert\mathcal{S}\rvert$	OVE-SGD	ARSOFT AMO	ARLOGIT AMO
			OVE	ARSOFT						
MNIST	60000	10	91.9	92.4	95.4, 95.8	96.8	1	95.9	**96.9**	96.1
Fashion	60000	10	84.0	84.2	84.8, 86.3	87.8	2	86.5	**87.4**	86.6
BibTeX	4880	147	35.2	36.1	23.3, 34.2	35.9	30	36.4	**39.4**	36.8
Mediamill	30993	50	31.5	31.3	37.8, **38.9**	35.5	20	35.9	36.0	35.3
Delicious	12920	355	17.7	**18.3**	15.9, 17.5	16.4	30	16.0	16.4	16.2

Table 1: The third column gives accuracies obtained by a linear model combined with OVE and the best AR likelihood Ruiz2018. RM and HL refer to GP model with Robust max likelihood and Hernandez-Lobato likelihood, respectively. $|\mathcal{S}|$ is the subsample size. The baseline for GP was obtained using GPFlow, while for the linear models we used code provided by [5].

Name	Linear-OVE		Linear-ARSOFT		GP-RM		GP-HL		OVE-SGD		ARSOFT(AMO)		ARLOGIT(AMO)	
	A	C	A	C	A	C	A	C	A	C	A	C	A	C
M	31.5	7.0	31.3	7.2	37.8	12.5	**38.9**	20.9	35.9	35.0	36.0	**42.0**	35.3	22.0
M-10D	29.6	4.1	29.7	4.1	**34.8**	12.5	30.1	5.0	32.2	12.5	33.6	16.2	32.9	**16.5**
M-1000N	29.7	8.8	29.5	7.4	**32.3**	8.3	26.0	15.5	29.8	18.7	31.0	**24.0**	29.9	11.0
M-WMF	20.1	7.3	20.7	7.3	26.0	16.7	23.2	14.0	23.5	17.0	**26.4**	**35.5**	24.9	21.5

Table 2: Performance of models on Mediamill with different slices. M is the original Mediamill dataset, M-10D is reduced to $D = 10$ dimensions, M-1000N only contains $N = 1000$ observations, and in M-WMF the most frequent classes have been removed. A and C denote Accuracy and Coverage, respectively.

relatively small. This is consistent with the behavior observed by Ruiz et al. [5].

The performance of the amortized AR methods is better or similar to their non-amortized counterparts (see Figure 1), while having the advantage of a lower memory footprint. The Inner-Loop method (IL) does not perform as well for bigger data sets like BibTex.

The left panel in Figure 1 shows the classification accuracy for all methods on MNIST dataset when the percentage of negative class samples is varied from 10% to 90%. As expected, the general tendency is that classification accuracy increases when the percentage of negative samples is increased. The right panel shows the average rank for each method across all datasets. It is seen that the amortized AR method with the softmax likelihood is uniformly superior for all sample percentages. From here onwards, we only show results for

amortized inference since they were mostly superior or similar to the inner loop inference, and more robust. An explanation could be that the optimisation in the local step can be challenging, quite sensitive to variational parameter update schedule and can get stuck in local minima, when the number of classes is high.

Table 1 shows that for the full Mediamill dataset, the proposed methods perform slightly worse than the baseline GP-RM and GP-HL methods. To further analyze this, we tested the methods on several different slices of the original Mediamill dataset. In particular, we manipulated the dimensionality D, the number of observations N, and the class imbalance by removing the most frequent classes. This resulted in the following three new datasets: M-10D, M-1000N, M-WMF, respectively, shown in Table 2. Both the baseline and proposed GP have better accuracy and coverage than the linear models for all variations of Mediamill. The accuracy for all baseline methods drop substantially when the most frequent classes are removed from the training set. The proposed methods seem to have disadvantage in case of high-class imbalance, but the relative performance gets better when the class imbalance is reduced. The two proposed methods have better coverage than the baseline methods for all variations of the Mediamill dataset. The ARSOFT method produced significantly better coverage in three out four variations of the Mediamill dataset, while producing comparable performance to the ARLOGIT method for the M-10D variant.

For all the data sets used in the experiments, a sample size of about 20-30% worked well and was sufficient for optimisation to be stable. The performance then saturated for higher sample sizes.

5. CONCLUSION

We proposed a scalable framework for extreme classification using Gaussian processes. The core idea is to combine the approximate likelihood method called Augment and Reduce with an amortized variational inference scheme. We applied the proposed methods to several benchmark datasets and demonstrated that the proposed method is capable of performing on par or even better compared to state-of-the-art methods for GP multi-class classification.

6. REFERENCES

[1] Y Prabhu and M Verma, "FastXML:Fast, accuraste and stable tree-classifier for extreme multi-label learning," in *KDD*, 2014.

[2] K Bhatia, H Jain, P Kar, M Varma, and P Jain, "Sparse local embeddings for extreme multi-label classification," in *NIPS*, 2015.

[3] F Fagan and G Iyengar, "Unbiased scalable softmax optimization," *arXiv preprint arXiv:1803.08577*, 2018.

[4] MK Titsias, "One-vs-each approximation to softmax for scalable estimation of probabilities," in *NIPS*, 2016.

[5] FJR Ruiz, MK Titsias, AB Dieng, and DM Blei, "Augment and reduce: Stochastic inference for large categorical distributions," in *ICML 18*, 2018.

[6] CE Rasmussen and CKI Williams, *Gaussian Processes for Machine Learning*, MIT Press, 1 2006.

[7] MK Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in *AISTATS 12*, 2009.

[8] K Krauth, E V Bonilla, K Cutajar, and M Filippone, "AutoGP: Exploring the capabilities and limitations of Gaussian process models," in *UAI'17*, 2017.

[9] J Riihimäki, P Jylänki, and A Vehtari, "Nested expectation propagation for Gaussian process classification," *JMLR*, vol. 14, pp. 75–109, 2013.

[10] C Villacampa-Calvo and D Hernández-Lobato, "Scalable multi-class Gaussian process classification using expectation propagation," in *ICML'17*, 2017.

[11] M Girolami and S Rogers, "Variational Bayesian multinomial probit regression with Gaussian process priors," in *Neural Computation 18*, pp. 790–1817. 2006.

[12] JH Albert and S Chib, "Bayesian analysis of binary and polychotomous response data," *JASA*, vol. 88, pp. 669–679, 1993.

[13] J Hensman, A Matthews, and Z Ghahramani, "Scalable Variational Gaussian Process Classification," in *AISTATS 15*, 2015, vol. 38 of *PMLR*, pp. 351–360.

[14] J Hensman, N Fusi, and N Lawrence, "Gaussian processes for big data," in *UAI 2013*.

[15] AD Saul, *Gaussian Process Based Approaches for Survival Analysis*, Ph.D. thesis, 2018.

[16] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Effective and efficient multilabel classification in domains with large number of labels,," in *ECML/PKDD 2008 Workshop on Mining Multidimensional Data*, 2008.

[17] R Babbar and B Schölkopf, "Dismec: Distributed sparse machines for extreme multi-label classification," in *WSDM'17*, 2017, pp. 721–729.

[18] IEH Yen, X Huang, W Dai, P Ravikumar, I Dhillon, and E Xing, "Ppdsparse: A parallel primal-dual sparse method for extreme classification," in *SIGKDD'17*, 2017.

[19] D Hernández-Lobato, J Miguel Hernández-Lobato, and P Dupont, "Robust multi-class Gaussian process classification," in *NeurIPS*, 2011, pp. 280–288.