Bounded Gaussian process with multiple outputs and ensemble combination

Jeremy H. M. Wong
Institute for Infocomm Research (I^2R)
*A*STAR*, Singapore
jeremy_wong@i2r.a-star.edu.sg

Huayun Zhang
Institute for Infocomm Research (I^2R)
*A*STAR*, Singapore
zhang_huayun@i2r.a-star.edu.sg

Nancy F. Chen
Institute for Infocomm Research (I^2R)
*A*STAR*, Singapore
nfychen@i2r.a-star.edu.sg

***Abstract*—Spoken Language Assessment (SLA) is a subjective task, where different human raters often assign differing scores for the same input. It also often has a bounded score range. Prior work of applying a Gaussian Process (GP) to SLA uses a Gaussian output, which is unbounded, and does not consider inter-rater uncertainty. This paper investigates using a bounded beta density function output for a GP in SLA and proposes to extend this bounded GP framework to utilise the multiple output samples per input in the training set. In the experiments, various types of Neural Network (NN) and GP models are trained. This paper investigates combining ensembles of these GPs and NNs. Experiments on the speechocean762 dataset show that using a beta output is better able to predict the inter-rater uncertainty than a Gaussian output. Using multiple output samples in the training set further improves the beta-output GP's inter-rater uncertainty prediction. Combination between a GP and NN yields improvements.**

***Index Terms*—Gaussian process, bounded output, uncertainty, ensemble combination, spoken language assessment**

I. Introduction

The task of Spoken Language Assessment (SLA) is to assign a score to spoken audio from a student, relating to the oral proficiency. Being a subjective task, different human raters may assign varying scores to the same audio, because of the limited coverage of the rubric and each teacher's bias. The task can be made more objective, by increasing the coverage of the rubric to make decisions in cases where the raters disagree. However, such decisions may not generalise well to the diversity of requirements of different users. Furthermore, expressing information about whether raters disagree may allow the feedback given by the system to not unfairly penalise the student user when raters would have disagreed on the score, and to instead seek clarification or human intervention. Rather than making the task more objective, this paper instead modifies the model to better fit the subjectivity of the task, thereby allowing better generalisation to diverse users and providing uncertainty output information that can be used to calibrate feedback. Subjectivity is often accounted for by providing annotations from multiple human raters per input in the dataset. It may be useful for an automatic model to also take this uncertainty into consideration. This paper investigates making a Gaussian process (GP) [1] better utilise and predict such uncertainty.

This work was supported by the A*STAR Computational Resource Centre through the use of its high performance computing facilities.

Three novelties are proposed in this paper. Work in [2] uses a GP for SLA, with a standard Gaussian output density function and while only considering a single reference output for each input in the training set. A Gaussian output has an infinite support. In SLA, the output score is often assumed to be bounded, which may not match well with this. The *first novelty* is to take the approach in [3] of a GP with a bounded output density, and apply it to a real-world SLA task. Matching the support may improve the uncertainty modelling. A GP naturally exhibits distributional uncertainty [4], with a predicted variance that increases for test inputs far from the training inputs. It also reduces the influence of model uncertainty [5], through its interpretation as a marginalisation over a distribution of functions [1]. The *second novelty* is to improve the data uncertainty [5] modelling of a bounded-output GP, by extending the formulation to take into account the inter-rater agreement, represented by the diversity of annotations from different raters for the same input. The computational cost of this approach is reduced by omitting redundant random variables. Finally, having investigated different SLA model types, using bounded and unbounded GPs, and also Neural Networks (NN), the *third novelty* in this paper is to investigate combining ensembles [6] of these different model types to further reduce the influence of model uncertainty. This builds upon prior works that often only combine ensembles with different NN parameters [7] or NN topologies [8].

II. Related work

A GP with a beta density output can be implemented using Laplace's approximation [3]. This is similar to using Laplace's approximation to implement a GP for classification [9]. This paper investigates applying such a beta-output GP to SLA. This paper also proposes to extend the beta-output GP to consider having multiple output samples in the training set. This differs from a multi-output GP [10], [11], because here, all outputs arise from the same task. Considering multiple output samples per input aims to incorporate some data uncertainty [5] into a GP, which instead naturally models distributional uncertainty [12] through an output standard deviation that increases for inputs that are further away from the training set. Data uncertainty can also be captured by NNs, by interpreting the NN outputs as parameters of a distribution [13] and

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00062

explicitly training these toward the distribution represented by the multiple raters [14], [15].

III. Gaussian process

A GP computes the outputs, $\boldsymbol{y}$, from input feature vectors, $\mathbf{X}$, by first placing a jointly Gaussian prior on latent variables, $\boldsymbol{f}$,

$$p\left(\boldsymbol{f}|\mathbf{X}\right) = \mathcal{N}\left(\boldsymbol{f}; \mathbf{0}, \mathbf{K}\left(\mathbf{X}, \mathbf{X}\right)\right). \tag{1}$$

The covariance is defined by the kernel, $\mathbf{K}$, which computes a similarity between features. The squared exponential kernel,

$$k_{ij}\left(\mathbf{X}, \mathbf{X}'\right) = s^2 \exp\left[-\frac{\left(\boldsymbol{x}_i - \boldsymbol{x}'_j\right)^\top \left(\boldsymbol{x}_i - \boldsymbol{x}'_j\right)}{2l^2}\right], \tag{2}$$

is used here, where i and j are the data point indexes, l is a length hyper-parameter, and s is a scale hyper-parameter.

The marginal likelihood of the outputs is computed as

$$p\left(\boldsymbol{y}|\mathbf{X}\right) = \int p\left(\boldsymbol{y}|\boldsymbol{f}\right) p\left(\boldsymbol{f}|\mathbf{X}\right) d\boldsymbol{f}, \tag{3}$$

and the hyper-parameters may be optimised by maximising

$$\mathcal{F} = \log p\left(\boldsymbol{y}^{\text{ref}}\middle|\mathbf{X}\right), \tag{4}$$

where $\boldsymbol{y}^{\text{ref}}$ are the training set reference outputs. During inference, the predicted output, $\widehat{y}$, from a test set input, $\widehat{\boldsymbol{x}}$, can be inferred from the posterior,

$$p\left(\widehat{y}|\widehat{\boldsymbol{x}}, \boldsymbol{y}, \mathbf{X}\right) = \int p\left(\widehat{y}\middle|\widehat{f}\right) p\left(\widehat{f}\middle|\widehat{\boldsymbol{x}}, \boldsymbol{y}, \mathbf{X}\right) d\widehat{f}, \tag{5}$$

by, for example, choosing the mean. Here, $\widehat{f}$ represents the latent variable for the test set data point.

A. Gaussian density output

In a standard GP, the output density function is Gaussian,

$$p\left(\boldsymbol{y}|\boldsymbol{f}\right) = \prod_{i=1}^{N} \mathcal{N}\left(y_i; f_i, \sigma^2\right), \tag{6}$$

where N is the dataset size and σ is a hyper-parameter that can represent noise in the observed output. It is assumed in (6) that the outputs for different data points are independent of each other, when given the latent variables. Choosing the output density as a Gaussian allows $p(\boldsymbol{y}|\mathbf{X})$, $p(\widehat{f}|\widehat{\boldsymbol{x}}, \boldsymbol{y}, \mathbf{X})$, and $p(\widehat{y}|\widehat{\boldsymbol{x}}, \boldsymbol{y}, \mathbf{X})$ to also be Gaussian.

B. Beta density output

A Gaussian density function, with an infinite support, may not match well with tasks having a bounded output. A beta density,

$$\mathcal{B}\left(y; \alpha, \beta\right) = \frac{\Gamma\left(\alpha + \beta\right)}{\Gamma\left(\alpha\right)\Gamma\left(\beta\right)} y^{\alpha-1}\left(1-y\right)^{\beta-1}, \tag{7}$$

has a bounded support and may thereby better match the assumptions of the application domain. Here, Γ is the gamma function, and α and β are the parameters of the density function. This can be used as the output density of a GP, to allow the GP to also have a bounded output support,

$$p\left(\boldsymbol{y}|\boldsymbol{f}\right) = \prod_{i=1}^{N} \frac{\mathcal{B}\left(\frac{y_i - y_{\min}}{y_{\max} - y_{\min}}; \nu\Phi\left(f_i\right), \nu\left(1 - \Phi\left(f_i\right)\right)\right)}{y_{\max} - y_{\min}}, \tag{8}$$

where ν is a hyper-parameter that controls the sharpness of the density function, and $y_{\max}$ and $y_{\min}$ are the upper and lower bounds of the support respectively. The beta density mean is computed by parsing the latent variable through a function that squashes its value to be within $[0, 1]$, such as the normal cumulative density, $\Phi(f) = \int_{-\infty}^{f} \mathcal{N}(f'; 0, 1) df'$.

However, when using a beta output density, the resulting density functions that are used in hyper-parameter optimisation and inference are no longer closed within a single family. To allow for computational tractability, approximations such as Laplace [9], variational [16], and expectation propagation [17] can be used, with a comparison in [18]. The experiments in this paper use Laplace's approximation, as its cheaper computational cost allows for quicker experimentation [18]. This substitutes the joint log-posterior of $\boldsymbol{y}$ and $\boldsymbol{f}$,

$$\log p\left(\boldsymbol{y}, \boldsymbol{f}|\mathbf{X}\right) = \log p\left(\boldsymbol{y}|\boldsymbol{f}\right) + \log p\left(\boldsymbol{f}|\mathbf{X}\right), \tag{9}$$

with its second-order Taylor expansion around the $\boldsymbol{f}$ that maximises (9). This yields approximated forms for the marginal likelihood in (3) and posterior in (5) that are related to Gaussians, which can then be used to optimise the hyper-parameters and perform inference. A more detailed description of the application of Laplace's approximation and expectation propagation to a beta-output GP may be found in [3].

IV. Multiple output samples for each input

The uncertainty represented by a model's output posterior may be taxonomised into three types [4], [5], of data, model, and distributional uncertainties. Distributional uncertainty is the expectation that the outputs for test inputs far from the training inputs should be predicted with higher uncertainty. In a GP kernel, distant test inputs have a small correlation with the training data, and thus will have uncertain predictions. Data uncertainty expresses a distribution over the possible hypotheses for inputs that overlap, due to noise, limited information in the input, or subjectivity in the task. Model uncertainty is the lack of knowledge of which model best represents the data, exemplified by the different biases that influence the predictions. Combination in an ensemble, in section VI, aims to reduce the influence of model uncertainty, and obtain a purer representation of data uncertainty, by being a Monte Carlo approximation of a Bayesian NN (BNN) [19]. A GP may already be doing this implicitly, since it can be interpreted as marginalising over a distribution of functions [1]. In SLA, the collection of multiple rater scores, expressed as a reference distribution later in (14), suggests marginalising out a human analogue of model uncertainty, and thus this reference distribution may be interpreted as a reference of data uncertainty. The reference distribution can also be interpreted

as representing the fraction of raters who would have assigned each score.

Work in [2] only uses a single reference output per input in the training set. This paper instead proposes to allow the bounded GP to better utilise the reference data uncertainty in the training data. It is hoped that by leveraging upon the multiple rater scores in the training set, the bounded GP would be better able to predict the fraction of raters for each score on the test set, expressing a more accurate data uncertainty. This expression of uncertainty may be useful to calibrate the feedback that is given to a user. For example, if a student user is predicted to have a low score, but it is also predicted that multiple raters would have disagreed on that score, then it may not be appropriate to penalise the student, but to instead seek clarification or human intervention, so that better learning outcomes can be achieved.

The bounded GP formulation thus far does not consider multiple output annotations per input, but instead assumes there being only a single output reference per input. In SLA, a single reference score can be computed by combining the scores from the multiple raters as a majority vote [20], mean [21], or median [21]. However, this omits information about the inter-rater uncertainty. This paper proposes that the hyper-parameters of the GP can be optimised while considering the multiple rater scores in the training set, by maximising the joint marginal log-likelihood of all output samples,

$$\mathcal{F}_{\text{joint}} = \log p\left(\mathbf{Y}^{\text{ref}} \middle| \mathbf{X}\right), \tag{10}$$

instead of only the combined scalar reference in (4). Here, $\mathbf{Y}^{\text{ref}}$ represents the collection of the annotated scores from all of the multiple raters, for all of the training data points. A naive extension to allow for multiple output samples per input is to repeat the input for each output sample. However, this will increase the dimension of the kernel matrix proportionally to the number of repetitions. The computational cost involved in matrix multiplication scales quadratically with the matrix dimension, and that of matrix inversion scales cubically. Thus, this repetition approach may not be computationally feasible.

Repeating the inputs results in a kernel that expresses perfect correlation between the latent variables associated with these repetitions. Perfectly correlated random variables are redundant, and there is no need to compute them explicitly [22]. This paper proposes that for a GP with a beta output density, multiple output samples per input in the training set can be modelled by replacing the single sample output density of (8) with the joint likelihood of observing all output samples, without expressing the redundant latent variables,

$$p\left(\mathbf{Y}|\boldsymbol{f}\right) = \prod_{i=1}^{N} \prod_{r=1}^{R_i} \frac{\mathcal{B}\left(\frac{y_{ir}-y_{\min}}{y_{\max}-y_{\min}}; \nu\Phi\left(f_i\right), \nu\left(1-\Phi\left(f_i\right)\right)\right)}{y_{\max}-y_{\min}}, \tag{11}$$

where r indexes the rater and R_i is the number of raters for the ith data point. Omission of the repetitions retains the original kernel size, thus preserving the computational cost. It is assumed that the output samples are conditionally independent of each other, when given the latent variables. This output density can be substituted into the Taylor approximation of (9) for both training and inference. During inference, (11) is used for the training data, but the posterior still uses the single sample output density of (8) for $p(\widehat{y}|\widehat{f})$ in (5). Using multiple output samples per input is instead applied to a GP with a Gaussian output density in [22], [23], where unlike here, the density functions can be expressed analytically.

V. Spoken language assessment setup

In SLA, each audio input of sequential features is assigned an oral proficiency score within a bounded range. Prior works have used NNs, which can either be designed for regression with a sigmoid output layer that computes a continuous scalar score [24], or classification with a softmax output over the possible integer score classes [25]. Bidirectional Long Short-Term Memory (BLSTM) [26] or self-attention [27] together with pooling can accommodate different sequence lengths between the input and output [28], [29]. It may not be trivial to use a GP for sequential inputs. Work in [2] extracts hand-crafted sentence-level features to predict sentence-level scores. The GP kernel can also be designed to operate on sequences [30]. In this paper, a NN SLA model was first used to extract sentence-level bottleneck features, which were then used as inputs to a GP [31], to compute sentence-level scores. The NN was not jointly fine-tuned together with the GP, to avoid the risk of overfitting [32].

The input features to the NN comprised a concatenation of goodness of pronunciation [33], log phone posterior [34], log posterior ratio [34], tempo [24], phonetic embedding [24], and pitch [35] features, forming a sequence with one feature vector per phone in the sentence. The NN model comprised a BLSTM layer with 32 nodes per direction, a pooling layer that computed an equally weighted average over all phones in the sentence, a linear layer to map to the output dimension, and an output sigmoid or softmax layer. The continuous sigmoid output was scaled to the bounds of the score range. Sentence-level bottleneck features for the GP were extracted after the pooling layer. Dropout [36], with a 60% omission probability, was used before the BLSTM and linear layers. The NNs were trained toward a combined reference score, computed as a mean between the multiple rater annotations. Note that this differs from [21], which instead computes the combined sentence-level reference as a median. A model with a sigmoid output, named $\text{NN}_{\text{scalar}}$, was trained toward this combined reference using a Mean Squared Error (MSE) criterion. A model with a softmax output, named $\text{NN}_{\text{categorical}}$, was trained toward the combined reference using a cross-entropy criterion.

VI. Ensemble combination

Different model types may be biased in varied ways, yielding different predictive behaviours and presenting an uncertainty of which to use. A combination may reduce the influence of this model uncertainty and leverage upon this diversity. This paper proposes to combine NN and GP models. An ensemble of GP and NN models is also used in [37], as a teacher for knowledge distillation. The GP posterior is first

discretised by cumulating the likelihoods of the continuous output scores that would have been rounded to each discrete integer score,

$$P\left(\mathring{\widehat{y}}=c\middle|\widehat{\boldsymbol{x}},\boldsymbol{y},\mathbf{X}\right)=\frac{\int_{c-0.5}^{c+0.5}p\left(\widehat{y}|\widehat{\boldsymbol{x}},\boldsymbol{y},\mathbf{X}\right)d\widehat{y}}{\sum\limits_{c'}\int_{c'-0.5}^{c'+0.5}p\left(\widehat{y}'|\widehat{\boldsymbol{x}},\boldsymbol{y},\mathbf{X}\right)d\widehat{y}'},\quad(12)$$

where $\mathring{\widehat{y}}$ represents the discrete random variable of the output, that can take possible integer values c. The discrete posteriors of multiple models are then combined as a mixture model,

$$\overline{P}\left(\mathring{\widehat{y}}=c\middle|\widehat{\boldsymbol{x}}\right)=\sum_{m=1}^{M}\lambda_m P\left(\mathring{\widehat{y}}=c\middle|\widehat{\boldsymbol{x}};m\right),\quad(13)$$

where m enumerates the models in the ensemble, M is the ensemble size, and λ_m are combination weights satisfying $\sum_m\lambda_m=1$ and $\lambda_m\geq 0$. This Monte Carlo approximation of a BNN may reduce model uncertainty, yielding a purer data uncertainty. The combined hypothesis can be inferred from the combined posterior, by choosing the mean, median, or mode.

The discretisation of the GP posterior is aligned with the standard practice for the speechocean762 dataset, of first rounding the hypothesis and reference scores to the closest integers, before computing the evaluation metrics [21]. Furthermore, the discretisation facilitates combination between continuous output GPs and categorical NNs. It may also be possible to perform combination as a mixture of continuous densities. However, this may not be straightforward, because of the need to then choose a single hypothesis as either the mean, median, or mode of the combined density.

VII. Experiments

Experiments used the speechocean762 dataset [21]. This comprises 2500 sentences and 125 disjoint speakers, in each of the training and test sets. The sentences are read in English by native Mandarin speakers. Only the sentence-level pronunciation accuracy scores were used. These scores were provided by 5 human raters per sentence, and range between 0 to 10. Following [21], a time delay NN hybrid speech recognition model [38] was first trained on the Librispeech [39] 960 hours data, following the standard Kaldi [40] recipe, up till the cross-entropy stage. This was used to force align the speechocean762 training and test audio toward the transcriptions. This forced alignment was then used to compute SLA features, described in section V. The NN models, also described in section V, were trained using the speechocean762 training set, with 10% of the sentences held out for validation. Sentence-level bottleneck features were extracted from a NN model, and principle component analysis whitening was used to better abide by the tied covariance assumption of the kernel in (2). These transformed bottleneck features were used as inputs to the GP. The GP hyper-parameters were optimised using gradient descent. The bottleneck features from $\text{NN}_{\text{scalar}}$ were used, as initial experiments suggested better GP performance than when using features from $\text{NN}_{\text{categorical}}$.

TABLE I
Performance of NN and GP SLA models

Model	PCC↑	MSE↓	KL↓
$\text{NN}_{\text{scalar}}$	0.711	1.232	-
$\text{NN}_{\text{categorical}}$	0.701	1.208	1.26
GP_{gauss}	0.710	1.149	3.10
GP_{beta}	0.714	1.133	2.36

During inference, both the hypothesised and combined reference scores were rounded to the closest integers before computing the evaluation measures, following [21]. The models were evaluated against the combined reference by measuring the Pearson's Correlation Coefficient (PCC) and the MSE. A discrete Kullback-Leibler (KL) divergence was used to assess the model's ability to match the uncertainty of the multiple raters. A continuous KL divergence was avoided, because of the difficulties of normalisation and interpretation. The continuous GP posterior was first discretised using (12). Then, the discrete KL divergence was computed as

$$\text{KL}=\sum_{i=1}^{\widehat{N}}\sum_{c}\sum_{r=1}^{R_i}\frac{\delta\left(c,\widehat{y}_{ir}^{\text{ref}}\right)}{\widehat{N}R_i}\log\frac{\frac{1}{R_i}\sum\limits_{r'=1}^{R_i}\delta\left(c,\widehat{y}_{ir'}^{\text{ref}}\right)}{P\left(\mathring{\widehat{y}}=c\middle|\widehat{\boldsymbol{x}}_i,\boldsymbol{y},\mathbf{X}\right)},\quad(14)$$

where $\widehat{N}$ is the test set size, $\widehat{y}_{ir}^{\text{ref}}$ is the reference score from the rth rater for the ith test sample, and the reference distribution is a mixture of Kronecker delta functions, δ. Statistical significance for MSE and KL divergence was measured using a two-tailed paired t-test. Significance for PCC was measured using the $Z_1^\star$ approach in [41], by first computing an approximately normally distributed transformation [42] from the two PCCs being compared, then computing the two-tailed normal cumulative density of this transformed quantity.

The first experiment investigates using a GP with a beta output density for SLA, referred to as GP_{beta}, compared against NNs and a Gaussian-output GP, referred to as GP_{gauss}. The results in table I suggest that for PCC and MSE, the GPs perform comparably against the NNs. Using a beta output may yield improvements over a Gaussian output, but not significantly, with $\rho_{\text{PCC}}=0.246$ and $\rho_{\text{MSE}}=0.260$. The beta output performs at least as well as a Gaussian output, despite the approximations in its implementation. PCC and MSE evaluate how well the scalar hypothesis matches with a scalar combined reference score. It may also be useful to evaluate how well the posterior of the model matches with the distribution of scores from the multiple raters, using the KL divergence. The results suggest that GP_{beta} is significantly better than GP_{gauss} at matching the distribution of scores from the multiple raters, with $\rho_{\text{KL}}<0.001$. The KL divergence was not computed for $\text{NN}_{\text{scalar}}$, because this model does not compute a probabilistic output.

The next experiment assesses the proposed extension of allowing a beta-output GP to utilise multiple output samples for each input in the training set. The results in table II suggest that using separate training set output scores from the multiple raters may not yield PCC or MSE gains. However, the KL di-

TABLE II
USING MULTIPLE RATER SCORES IN A BETA-OUTPUT GP

Training outputs	PCC	MSE	KL
single mean	0.714	1.133	2.36
multiple	0.711	1.282	0.90

TABLE III
ENSEMBLE COMBINATION BETWEEN GPS AND NNS

Combination	PCC	MSE	KL
$\text{NN}_{\text{scalar}}$ + $\text{NN}_{\text{categorical}}$	0.727	1.126	1.62
GP_{gauss} + GP_{beta}	0.712	1.137	2.37
GP_{beta} + $\text{NN}_{\text{categorical}}$	0.728	1.068	1.29
GP_{gauss} + GP_{beta} + $\text{NN}_{\text{scalar}}$ + $\text{NN}_{\text{categorical}}$	0.731	1.096	1.49

vergence results suggest that when using the multiple training set output scores, the test set posterior computed by the GP is able to more closely match the distribution of reference scores from the multiple raters, with $\rho_{\text{KL}} < 0.001$. This suggests that although a GP is designed to primarily capture distributional uncertainty [12], incorporating information about the training set data uncertainty into the bounded GP does allow it to also better predict the test set data uncertainty.

Having already trained various model types, the final experiment investigates combinations between ensembles of these NN and GP models. Four models were used, namely $\text{NN}_{\text{scalar}}$, $\text{NN}_{\text{categorical}}$, GP_{gauss}, and GP_{beta}. The GPs did not use multiple output samples in the training set. The combined posterior was computed using (13) with equal weights, and the combined hypothesis was inferred as the mean. The results in table III, compared to the single models in table I, show that combining $\text{NN}_{\text{scalar}}$ and $\text{NN}_{\text{categorical}}$ yields PCC and MSE improvements, while combining GP_{gauss} and GP_{beta} may not. This suggests that GP_{gauss} and GP_{beta} may predict similar hypotheses. Combining GP_{beta} with $\text{NN}_{\text{categorical}}$ yields improvements, especially for MSE. This suggests that NN and GP models may behave differently from each other. The diversity between the predictions of two models can be assessed using the inter-model PCC, which between $\text{NN}_{\text{scalar}}$ and $\text{NN}_{\text{categorical}}$ is 0.832, GP_{gauss} and GP_{beta} is 0.975, and GP_{beta} and $\text{NN}_{\text{categorical}}$ is 0.848. A smaller value indicates a wider prediction diversity. These results support the observed trends from combination, showing that the GP_{gauss} and GP_{beta} hypotheses are similar, while GP_{beta} and $\text{NN}_{\text{categorical}}$ are more diverse. Adding GP_{gauss} and $\text{NN}_{\text{scalar}}$ into the combination of GP_{beta} and $\text{NN}_{\text{categorical}}$ does not yield consistent further gains. The combinations do not improve the KL divergence, indicating the difficulty of trying to compute a purer data uncertainty by marginalising out the model uncertainty in an ensemble.

VIII. CONCLUSION

This paper has applied a bounded-output GP to SLA, proposed to extend the bounded-output GP framework to accommodate having multiple output samples in the training set, and investigated ensemble combination between GP and NN models. These aim to predict a more accurate data uncertainty, which may better inform the type of feedback that should be given to a user. A bounded GP is able to better predict the test set data uncertainty than an unbounded GP. Using multiple output samples in the training set for a bounded GP further improves this prediction. A combination of GPs and NNs is able to leverage upon their diverse behaviours to yield PCC and MSE improvements.

REFERENCES

[1] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.

[2] R. C. van Dalen, K. M. Knill, and M. J. F. Gales, "Automatically grading learners' English using a Gaussian process," in *SLaTE*, Leipzig, Germany, Sep 2015, pp. 7–12.

[3] B. S. Jensen, J. B. Nielsen, and J. Larsen, "Bounded Gaussian process regression," in *MLSP*, Southampton, UK, Sep 2013.

[4] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *NeurIPS*, Montréal, Canada, Dec 2018, pp. 7047–7058.

[5] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *NIPS*, Long Beach, USA, Dec 2017, pp. 5574–5584.

[6] J. G. Fiscus, "A post-processing system to yield reduced word error rates: recognizer output voting error reduction (ROVER)," in *ASRU*, Santa Barbara, USA, Dec 1997, pp. 347–354.

[7] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, Oct 1990.

[8] L. Deng and J. C. Platt, "Ensemble deep learning for speech recognition," in *Interspeech*, Singapore, Sep 2014, pp. 1915–1919.

[9] C. K. I. Williams and D. Barber, "Bayesian classification with Gaussian processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1342–1351, Dec 1998.

[10] K. Yu, V. Tresp, and A. Schwaighofer, "Learning Gaussian processes from multiple tasks," in *ICML*, Bonn, Germany, Aug 2005, pp. 1012–1019.

[11] E. V. Bonilla, K. M. A. Chai, and C. K. I. Williams, "Multi-task Gaussian process prediction," in *NIPS*, Vancouver, Canada, Dec 2007, pp. 153–160.

[12] B. Xu, R. Kuplicki, S. Sen, and M. P. Paulus, "The pitfalls of using Gaussian process regression for normative modeling," *PLoS ONE*, vol. 16, no. 9, Sep 2021.

[13] C. M. Bishop, "Mixture density networks," Aston University, Tech. Rep., Feb 1994.

[14] N. R. Prabhu, G. Carbajal, N. Lehmann-Willenbrock, and T. Gerkmann, "End-to-end label uncertainty modeling for speech-based arousal recognition using Bayesian neural networks," in *Interspeech*, Incheon, South Korea, Sep 2022, pp. 151–155.

[15] J. H. M. Wong, H. Zhang, and N. F. Chen, "Modelling inter-rater uncertainty in spoken language assessment," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2886–2898, Jul 2023.

[16] M. Opper and C. Archambeau, "The variational Gaussian approximation revisited," *Neural Computation*, vol. 21, no. 3, pp. 786–792, Mar 2009.

[17] T. P. Minka, "Expectation propagation for approximate Bayesian inference," in *UAI*, Seattle, USA, Aug 2001, pp. 362–369.

[18] H. Nickisch and C. E. Rasmussen, "Approximations for binary Gaussian process classification," *Journal of Machine Learning Research*, vol. 9, no. 67, pp. 2035–2078, Oct 2008.

[19] D. J. C. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, no. 3, pp. 415–447, May 1992.

[20] B. Lin and L. Wang, "A noise robust method for word-level pronunciation assessment," in *Interspeech*, Brno, Czechia, Aug 2021, pp. 781–785.

[21] J. Zhang, Z. Zhang, Y. Wang, Z. Yan, Q. Song, Y. Huang, K. Li, D. Povey, and Y. Wang, "speechocean762: an open-source non-native English speech corpus for pronunciation assessment," in *Interspeech*, Brno, Czechia, Aug 2021, pp. 3710–3714.

[22] J. H. M. Wong, H. Zhang, and N. F. Chen, "Multiple output samples per input in a single-output Gaussian process," in *Symposium for Celebrating 40 Years of Bayesian Learning in Speech and Language Processing and Beyond @ ASRU*, Taipei, Taiwan, Dec 2023.

[23] ——, "Variational Gaussian process data uncertainty," in *ASRU*, Taipei, Taiwan, Dec 2023.

[24] H. Zhang, K. Shi, and N. F. Chen, "Multilingual speech evaluation: case studies on English, Malay and Tamil," in *Interspeech*, Brno, Czechia, Aug 2021, pp. 4443–4447.
[25] R. Duan and N. F. Chen, "Unsupervised feature adaptation using adversarial multi-task training for automatic evaluation of children's speech," in *Interspeech*, Shanghai, China, Oct 2020, pp. 3037–3041.
[26] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks," in *IJCNN*, Montreal, Canada, Jul 2005, pp. 2047–2052.
[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, Long Beach, USA, Dec 2017, pp. 5998–6008.
[28] J. H. M. Wong, H. Zhang, and N. F. Chen, "Variations of multi-task learning for spoken language assessment," in *Interspeech*, Incheon, South Korea, Sep 2022, pp. 4456–4460.
[29] Y. Gong, Z. Chen, I.-H. Chu, P. Chang, and J. Glass, "Transformer-based multi-aspect multi-granularity non-native English speaker pronunciation assessment," in *ICASSP*, Singapore, May 2022, pp. 7262–7266.
[30] H. Lodhi, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels," in *NIPS*, Denver, USA, Nov 2000, pp. 563–569.
[31] R. Salakhutdinov and G. E. Hinton, "Using deep belief nets to learn covariance kernels for Gaussian processes," in *NIPS*, Vancouver, Canada, Dec 2007, pp. 1249–1256.
[32] S. W. Ober, C. E. Rasmussen, and M. van der Wilk, "The promises and pitfalls of deep kernel learning," in *UAI*, Jul 2021, pp. 1206–1216.
[33] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech Communication*, vol. 30, no. 2-3, pp. 95–108, Feb 2000.
[34] W. Hu, Y. Qian, F. K. Soong, and Y. Wang, "Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers," *Speech Communication*, vol. 67, pp. 154–166, Mar 2015.
[35] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *ICASSP*, Florence, Italy, May 2014, pp. 2494–2498.
[36] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, Jun 2014.
[37] J. H. M. Wong, H. Zhang, and N. F. Chen, "Distilling knowledge from Gaussian process teacher to neural network student," in *Interspeech*, Dublin, Ireland, Aug 2023, pp. 426–430.
[38] A. Waibel, T. Hanazawa, G. E. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339, Mar 1989.
[39] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *ICASSP*, Brisbane, Australia, Apr 2015, pp. 5206–5210.
[40] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *ASRU*, Hawaii, USA, Dec 2011.
[41] J. H. Steiger, "Tests for comparing elements of a correlation matrix," *Psychological Bulletin*, vol. 87, no. 2, pp. 245–251, 1980.
[42] O. J. Dunn and V. Clark, "Correlation coefficients measured on the same individuals," *Journal of the American Statistical Association*, vol. 64, no. 325, pp. 366–377, Mar 1969.