Uncertainty estimation in automatic pronunciation assessment with pseudo samples based on deep kernel learning

Binghuai Lin*, and Liyuan Wang*
* Smart Platform Product Department, Tencent Technology Co., Ltd, China
E-mail: {binghuailin, sumerlywang}@tencent.com

***Abstract*—A reliable and accurate automatic pronunciation assessment is of great help for second language (L2) learners. Commonly, a Gaussian process (GP)-based model is utilized to output pronunciation scores and their uncertainty. However, the application of GPs is limited by computational intractability when data are sufficiently numerous. In this paper, we propose an improved GP-based model for uncertainty prediction in automatic pronunciation assessment utilizing a small set of pseudo samples derived from the full training set. To further exploit more expressive information from data, we optimize the network based on deep kernel learning with deep features derived from an automatic speech recognition (ASR) system. Experimental results based on one spoken test show its superiority compared with the GP-based baselines with the full training set and standard kernels. With a small number of pseudo samples, which is only 25% compared to the full training set, we can match the full GP performance.**

Keywords: automatic pronunciation assessment, uncertainty estimation, pseudo samples, deep kernel learning,

I. Introduction

There is a growing demand for learning English as a second language, and reliable and accurate assessment for language learning is necessary for L2 learners. With the development of Computer-Assisted Pronunciation Training (CAPT) technology, automatic pronunciation assessment has become possible. Commonly, automatic speech pronunciation has been designed to evaluate learners' reading aloud proficiency in restricted testing tasks such as Read-by-Words and Read-by-Sentences [1].

Traditional features such as Goodness of pronunciation (GOP) are commonly used for automatic pronunciation assessment. GOP is defined as the posterior probability of the uttered phoneme given the corresponding acoustic segment calculated by an automatic speech recognition (ASR) system [2]. Designing these handcrafted features is always cumbersome and requires human knowledge, which may lead to insufficient representations of the speech. With advances in deep learning (DL), representation learning, which learns an intermediate representation of the input signal automatically, has been proven more useful and less dependent on human knowledge [3]. It has been very promising for different speech-based systems to learn more generalized features based on deep neural networks (DNN) compared to traditional handcrafted ones.

A reliable automatic pronunciation assessment for high-stake tests is still of a big challenge. There are significant variations in the speech or spoken responses. The speech signal conveys not only the linguistic information but also a lot of information about the speaker, including gender, age, and regional origin [4]. Moreover, variations in the quality of the recordings, such as background noise and volume levels, also make the assessment process difficult [5]. Thus, uncertainty estimation is necessary for reliable automatic pronunciation assessment. There are two sources of uncertainty conceptually [6][7]. The first one is epistemic uncertainty, which is also known as knowledge uncertainty. It represents uncertainty in model predictions due to lack of understanding or knowledge of the model. It can be reduced by providing more knowledge to the model. The second one is aleatoric uncertainty or data uncertainty. It results from genuine stochasticity in the data. The noise of data results in high entropy in the prediction. There is significant knowledge uncertainty in automatic pronunciation assessment. Due to variations in pronunciation, automatic pronunciation assessment may suffer from the insufficiency of the training data. For speakers unseen by the automated system, the grade predictions may be poor [5].

Gaussian process (GP) was explored to provide a measure of the uncertainty in automatic grading, where the mean value is used for scoring, and the variance is treated as the uncertainty value of the score [5]. The non-parametric nature of GP has lead to $O(N^3)$ training cost time and $O(N^2)$ testing cost time. It becomes impractical when the amount of training data increases [7][8]. Many recent studies have attempted to make sparse approximations to the full GP to bring the time cost down. Some studies normally selected a subset from training samples based on some information criterion. The next point was chosen for inclusion into the active set to maximize the differential entropy score [9]. A fast forward selection criterion was utilized to select points from the training samples [10]. These methods lack a reliable way of learning kernel hyper-parameters, as the active set selection interferes with their learning procedure [11]. Some work made joint optimization to find active set point locations and hyper-parameters. Pseudo samples were treated as parameters and learned by maximum likelihood (ML), which were not constrained to be a subset

*These authors contributed equally to this work.

of the data. Then, the covariance function of GP is parameterized based on these pseudo samples [11]. A variational learning method defined the inducing inputs to be variational parameters and selected them by minimizing the Kullback-Leibler (KL) divergence between a variational GP and the true posterior GP [12][13]. In this paper, we conduct pseudo sample learning to improve the GP-based method. Different from previous work's implicit pseudo sample learning by ML, which may be trapped in local optima, we explicitly model the relationship between pseudo samples and raw samples based on a neural network. As a non-parametric method, the information capacity of the GP-based model may decrease with the decreasing amount of available data [14]. Deep kernel learning was developed to combine the structural properties of deep architectures with the non-parametric flexibility of kernel methods [14]. Inspired by this, we jointly learn the pseudo samples and expressive kernels based on deep kernel learning.

In this paper, we utilize a GP-based model for uncertainty prediction in automatic pronunciation assessment. A small subset of pseudo samples of number M are derived from the full training samples of number N based on a neural network transformation, where $M << N$. This leads to a sparse GP-based model, which has $O(M^2N)$ training cost time and $O(M^2)$ testing cost time. To exploit more information from the pseudo and raw samples, we utilize deep kernel learning to jointly learn the pseudo samples and expressive kernels based on deep features derived from an automatic speech recognition (ASR) system. Experimental results based on the spoken test show the superiority to the GP-based baselines with full samples and the traditional kernel functions. In section 2, we will introduce the proposed method. The experiments are conducted, and the results are shown and discussed in section 3. We will draw the conclusion and future suggestions in section 4.

II. Proposed method

A. Gaussian process

A GP-based model is nonparametric, which learns a mapping from an infinite number of inputs to corresponding output values [15]. It specifies a prior and derives a posterior distribution over functions $f(x)$. The distribution is obtained over a set of function values $f = \{f(x_1), \cdots, f(x_N)\}$ at a finite set of training points $X = \{x_1, \cdots, x_N\}$. The joint Gaussian distribution for $f(X)$ is defined as Eq. (1), where $\mathrm{m}(X)$ is the mean and $\mathrm{K}(X, X)$ is the covariance matrix as functions of X. The observed outputs $Y = \{y_1, \cdots, y_N\}$ are assumed to be Gaussian-distributed around the real function values $f(X)$ with additive observation noise $\mathcal{N}(0, \sigma^2)$ as defined in Eq. (2).

$$f \sim \mathcal{N}(\mathrm{m}(X), \mathrm{K}(X, X)) \tag{1}$$

$$Y \sim \mathcal{N}(\mathrm{m}(X), \mathrm{K}(X, X) + \sigma^2 I) \tag{2}$$

To make prediction for a new value of x^* given the training data $\mathcal{D} = \{Y, X\}$, a joint distribution over both the training and test targets is defined in Eq. (3).

$$\begin{bmatrix} Y \\ y^* \end{bmatrix} \sim \mathcal{N}(0, \begin{bmatrix} \mathrm{K}(X, X) + \sigma^2 I & \mathrm{k}(x^*, X) \\ \mathrm{k}(x^*, X)^T & \mathrm{k}(x^*, x^*) \end{bmatrix}) \tag{3}$$

As Y and y^* are jointly Gaussian distributed, the conditional distribution is also Guaassian as in Eq. (4), where $\hat{\mu}$ and $\hat{\sigma^2}$ are defined in Eq. (5) and Eq. (6), respectively.

$$p(y^*|x^*, \mathcal{D}) \sim \mathcal{N}(y^*; \hat{\mu}, \hat{\sigma^2}) \tag{4}$$

$$\hat{\mu} = \mathrm{k}(x^*, X)^T (\mathrm{K}(X, X) + \sigma^2 I)^{-1} Y \tag{5}$$

$$\hat{\sigma^2} = \mathrm{k}(x^*, x^*) - \mathrm{k}(x^*, X)^T (\mathrm{K}(X, X) + \sigma^2 I)^{-1} \mathrm{k}(x^*, X) \tag{6}$$

The commonly used kernel function is the radial basis covariance function (RBF) defined in Eq. (7), which is also used in this paper. The RBF-based kernel function is parameterized by two parameters, namely l and σ_y^2. σ_y^2 is the output variance, and l is the length scale.

$$\mathrm{k}(x, x^*) = \sigma_y^2 \exp(-\frac{(x - x^*)^2}{2l^2}) \tag{7}$$

The GP-based model takes $O(N^3)$ training cost time for the computation of the inversion of the covariance matrix. It takes $O(N^2)$ testing cost time once the inversion is done.

B. Improved GP with pseudo samples based on deep kernel learning

As the GP-based model is limited by computational intractability when the training data increases, we modify GP based on pseudo samples of number M derived from all the training samples of number N, where $M << N$. The pseudo samples are explicitly transformed from the training samples by a deep neural network. The training and testing cost times can be reduced dramatically even with the additional neural network. To exploit expressive representations of data, we jointly optimize the kernel hyper-parameters and the weights of the network based on deep kernel learning utilizing deep features derived from an ASR acoustic model. To predict scores and uncertainty of testing samples, we use pseudo samples as the input to the GP-based model.

1) Deep feature representations: For deep kernel learning, we utilize deep feature representations as pronunciation features x. The deep features of an utterance are obtained from an ASR system. First, a DNN-HMM-based system for ASR is trained. The DNN-based acoustic model takes acoustic features based on fbank as input and outputs the posterior probability of the senone given observation with acoustic frames [16]. It consists of multi-layers of hidden units between the inputs and outputs. The representations from the last layer are converted to the frame-level senone predictions based on a fully connected (FC) layer. It is optimized by cross-entropy loss between the predictions and frame-level labels aligned using a GMM-HMM system. We use the hidden outputs from the DNN's last layer and the alignment time of each phoneme in an utterance from the ASR system. We obtain the final representation of the utterance by averaging the phoneme

representations, which are calculated by averaging the hidden outputs h_{ij} of corresponding frames, as defined in Eq. (8):

$$x_{utt} = \frac{1}{q}\sum_{i=1}^{q}(\frac{1}{m_p}\sum_{j=1}^{m_p}h_{ij}) \tag{8}$$

where m_p indicates the number of frames for the corresponding phoneme p and q is the total numbers of phonemes in the utterance.

2) Network structure for pseudo samples and deep kernel learning: We propose a network structure to explicitly model the relationship between pseudo samples and the raw training data as shown in Figure 1.

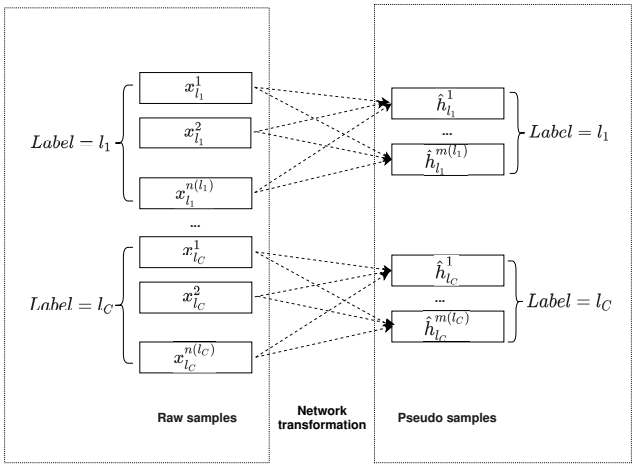

Fig. 1. Learning of hidden representations $\hat{h}$ of pseudo samples $\hat{x}$

It consists of three parts: input of raw samples x, multi-layer transformation, and hidden representations $\hat{h}$ of pseudo samples $\hat{x}$. Note that we learn the hidden representations $\hat{h}$ directly for the deep kernel instead of the original $\hat{x}$, which is not necessary for the deep kernel calculation. We denote the total number of label classes as C, where the ith label class is defined as l_i, and the number of samples for the corresponding label is $n(l_i)$. Deep features of the jth sample under label l_i is denoted as $x_{l_i}^j$. The deep features of each label class are then mapped by multiple hidden layers to calculate the hidden representations $\hat{h}_{l_i}^o$ for the oth pseudo samples $\hat{x}_{l_i}^o$. The total number of pseudo samples of label l_i is $m(l_i)$, where $m(l_i) << n(l_i)$.

For deep kernel learning, we also need to transform the deep features of raw training samples x_l to hidden representations $h(x_l)$ using a multi-layer neural network.

3) Network optimization and prediction: Based on hidden representations $\hat{H}$ of pseudo samples $\hat{X}$, we obtain the likelihood defined as Eq. (9), which is similar to Eq. (4), where x is transformed to $h(x)$ based on multiple hidden layers. To make the pseudo samples to be distributed similarly to the real data as previous work did [11], we place a Gaussian prior to pseudo samples defined in Eq. (10). We obtain the marginal likelihood defined in Eq. (11) by combining Eq. (9) and Eq. (10). We maximize the log marginal likelihood with respect to weights of the network and deep kernel hyper-parameters in Eq. (12), where θ are the parameters of the network. The log marginal likelihood can be divided into two terms shown in Eq. (13). The whole network can be optimized by a multitask learning (MTL) method combining two losses as shown in Figure 2.

$$p(y|x,\hat{H},Y) = \mathcal{N}(y|\mathrm{k}(h(x),\hat{H})^T(\mathrm{K}(\hat{H},\hat{H})+\sigma^2 I)^{-1}Y, \mathrm{k}(h(x),h(x)) - \mathrm{k}(h(x),\hat{H})^T(\mathrm{K}(\hat{H},\hat{H})+\sigma^2 I)^{-1}\mathrm{k}(h(x),\hat{H})) \tag{9}$$

$$p(Y|\hat{H}) = \mathcal{N}(Y|0,\mathrm{K}(\hat{H},\hat{H})+\sigma^2 I)) \tag{10}$$

$$p(y|x,\hat{H},\theta) = p(y|x,\hat{H},Y) * p(Y|\hat{H}) \tag{11}$$

$$\theta,\sigma_y,l = \underset{\theta,\sigma_y,l}{\operatorname{argmax}} \sum_{i=0}^{n}\log(p(y_i|x_i,\hat{H},\theta)) \tag{12}$$

$$\begin{aligned}\theta,\sigma_y,l &= \underset{\theta,\sigma_y,l}{\operatorname{argmax}}(\sum_{i=0}^{n}\log(p(y_i|x_i,\hat{H},Y)) + \sum_{i=0}^{n}\log(p(y_i|\hat{H}))) \\ &= \underset{\theta,\sigma_y,l}{\operatorname{argmax}}\left(\mathrm{Likelihood}_1 + \mathrm{Likelihood}_2\right)\end{aligned} \tag{13}$$

Prediction is made by considering a new input point x^*

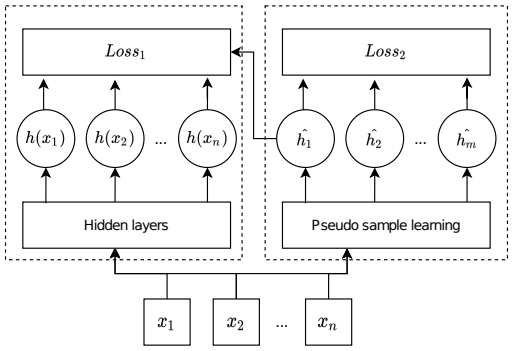

Fig. 2. Network optimization of improved GP

conditioned on the pseudo samples and hyperparameters. The new sample's distribution is predicted as Eq. (9).

III. Experiments

A. Corpus

First, we train the ASR model based on a mixed corpus of native and non-native data. The native data comes from the 960-hour native LibriSpeech corpus [17], and the non-native data is 1000-hour recordings of Chinese teenagers. The speech scoring data is obtained from restricted reading-aloud tasks of one English oral test of the Chinese National Higher Education Entrance Examination. The number of data for reading-aloud is 3,000 recorded by 3,000 Chinese ESL candidates. The scoring ranges are 0-5 with a 1 grade interval. The lowest score represents completely wrong pronunciation and the highest score represents native-like pronunciation. Two experts rated the data, and the final results are obtained by averaging these two scores, resulting in half grades in some scenarios. The averaged inter-rater correlations, calculated by

Pearson correlation coefficient (PCC) between scores of one rater and the other, is 0.76. The data is divided into 60% for training and 40% for testing. The distribution of data is shown in Figure 3.

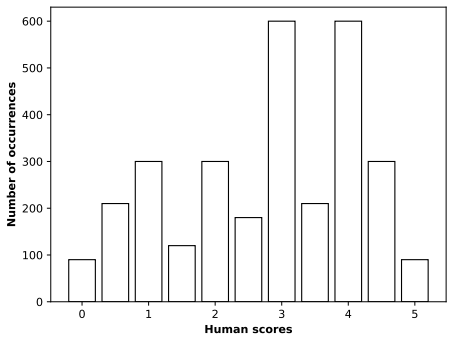

Fig. 3. Distribution of scoring

B. Experimental setup

The DNN-HMM-based ASR model was trained to achieve a word error rate (WER) of 15% for non-native data. The DNN-based acoustic model is composed of a time delay neural network with layer number of 11 [18]. We utilize the deep features from the acoustic model with the dimensionality of 256 as input for uncertainty estimation. The deep features are computed at the frame level with 30ms of each frame. These features are transformed by one hidden layer with parameters of $256 * 32$ for deep kernel learning. Parameters for ith label class in pseudo sample learning are $256 * 32$ and $n(l_i) * m(l_i)$, where $m(l_i)$ are predefined according to $n(l_i)$. In this paper, $n(l_i)$ is number of the training samples with label l_i, and $m(l_i)$ is the number of pseudo samples with label l_i. We will experiment with different numbers of pseudo samples in the following experiments. The parameters of the hidden layer are determined by tuning the model. Parameters for the RBF-based kernel, namely l and σ_y^2, are randomly initialized. We use Adam optimizer [19] with a learning rate $1e$-3 to train the network.

C. Comparative study

First, we compare the model performance with the typical GP and the previous work called sparse GP using pseudo inputs (SPGP) [11] based on full samples and their subsets. The implementation of SPGP is based on an open-source Python package [20], and the kernel for SPGP is RBF-based kernel, which is the same as ours. The performance of the scoring model is evaluated by calculating PCC between prediction scores and expert labeling, and the ratios of low prediction errors, which are defined as being inside 0.5 (i.e., less than or equal to half a grade out, denoted as $\% \leq 0.5$) or inside 1.0 (i.e., less than or equal to a full grade out, denoted as $\% \leq 1.0$) of the expert grade. Second, we evaluate the rejection performance by the Area Under the Curve (AUC) based scheme, which has been widely applied in uncertainty estimation for automatic speech assessment [7][8]. The AUC rejection ratio is defined as Eq. (14), where $\mathrm{AUC}_{\mathrm{rnd}}$ is the area under the random rejection curve, $\mathrm{AUC}_{\mathrm{opt}}$ is the optimal rejection curve, and $\mathrm{AUC}_{\mathrm{mod}}$ is the model rejection curve. Finally, we will do some ablation studies to show the effectiveness of our proposed method.

$$\mathrm{AUC}_{\mathrm{RR}} = \frac{\mathrm{AUC}_{\mathrm{mod}} - \mathrm{AUC}_{\mathrm{rnd}}}{\mathrm{AUC}_{\mathrm{opt}} - \mathrm{AUC}_{\mathrm{rnd}}} \tag{14}$$

1) Prediction performance of the scoring model: First, we compare the prediction performance with the typical GP and SPGP with full samples. The comparison results are shown in Table I. From the results, we can see the proposed method achieves results comparable to the typical GP and SPGP baselines with full samples. Then, we compare results based on different numbers of subset samples shown in Figure 4. The subset samples of different percents of training samples are learned or selected by the proposed method, the SPGP-based method, and a random method, respectively. The dash horizontal line in the figure indicates the performance of GP with full training samples. The random selection is made by picking an active set randomly from training data many times. From the figure, we can see both the SPGP-based method and the proposed method outperform the random method significantly. The SPGP-based method performs inferior to the proposed method when the percent of samples is less than 25%. When learning pseudo samples with the number of 25% of the training samples, the proposed method achieves comparable performance compared to GP with full samples.

TABLE I
PREDICTION PERFORMANCE COMPARISON IN READING ALOUD

Model	GP	SPGP (Full)	Our (Full)
PCC	64.3	63.9	64.1
$\% \leq 0.5$	89.3	89.7	90.2
$\% \leq 1$	98.3	99.1	99.3

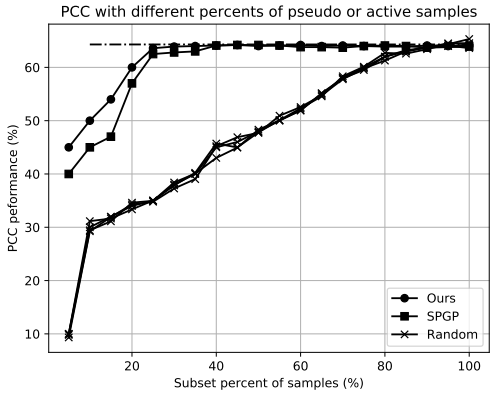

Fig. 4. PCC with different percents of pseudo or active samples

2) Rejection performance of the scoring model: Uncertainty estimation is used to reject predictions of high uncertainty. We rank all the candidates by measuring uncertainty and replacing the most uncertain predictions with human labels. We compare performance among the random replacement, the uncertainty-based replacement, and the optimal replacement. The optimal rejection is conducted by rejecting the decreasing absolute error values between predictions and human labels. Three $\mathrm{AUC_{RR}}$ values are computed based on PCC, $\% \leq 0.5$ and $\% \leq 1$. The standard deviation values derived from GP are used as the uncertainty measures of the predictions. The rejection performance is shown in Table II, where the SPGP-based method and our method are based on pseudo samples with 25% of the number of training samples.

TABLE II
REJECTION PERFORMANCE ($\mathrm{AUC_{RR}}$) COMPARISON

Model	GP	SPGP (25%)	Ours (25%)
PCC	40.3	35.7	37.1
$\% \leq 0.5$	43.7	39.7	38.7
$\% \leq 1$	47.1	41.1	44.5

From the results, we can see that the rejection performance degrades a little compared with the GP-based method with full samples. By analyzing the uncertainty score distributions of full GP and our method, we observe that the GP-based method with full samples has a larger mean and standard deviation than the proposed method, indicating underestimation of uncertainty with pseudo samples.

3) Ablation studies: To show the effectiveness of the proposed method, we conduct two experiments: (1) different features, including manual features and deep features (DP); (2) network optimization with deep kernel learning (DKL) and without deep kernel learning. The manual features of each utterance are average GOP scores [2] of each phoneme, with the dimensionality of 39 based on the Carnegie Mellon University (CMU) Pronouncing Dictionary [21]. We replace deep features with manual features for manual feature experimental setting. We remove hidden layers, which are used to transform training samples, for the optimization without deep kernel learning. We compare the rejection performance among these settings based on pseudo samples with 25% of the number of training samples. The rejection results are shown in Table III. From the results, we can see the rejection performance degrades greatly when using manual features, indicating deep features contains more sufficient pronunciation information than the manually designed features. Deep kernel learning can further facilitate representation of the data by pseudo sample learning.

TABLE III
REJECTION PERFORMANCE ($\mathrm{AUC_{RR}}$) OF DKL AND DP

	Ours	w/o DKL	w/o DP
PCC	37.1	32.1	29.3
$\% \leq 0.5$	38.7	34.3	35.1
$\% \leq 1$	44.5	35.3	30.2

IV. CONCLUSION

In this paper, we propose an improved GP-based method for automatic pronunciation assessment based on pseudo samples and deep kernel learning. We explicitly learn the pseudo samples from the training data based on neural network transformation. The hyperparameters for GP, and parameters for pseudo sample learning and deep kernel learning are jointly optimized. Experimental results based on the spoken test demonstrate the proposed method can achieve comparable scoring and rejection performance with the typical GP-based and SPGP-based methods. Also, with fewer samples the proposed method performs better than the SPGP-based method. In the future, we will extend the proposed method to assess less restricted spoken tasks.

REFERENCES

[1] J. Cheng, Y. Z. Dantilio, X. Chen, and J. Bernstein, "Automatic assessment of the speech of young english learners," in *Proceedings of the ninth workshop on innovative use of NLP for building educational applications*, 2014, pp. 12–21.

[2] 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, 2000.

[3] S. Latif, R. Rana, S. Khalifa, R. Jurdak, J. Qadir, and B. W. Schuller, "Deep representation learning in speech processing: Challenges, recent advances, and future trends," *arXiv preprint arXiv:2001.00378*, 2020.

[4] M. Benzeghiba, R. De Mori, O. Deroo, S. Dupont, T. Erbes, D. Jouvet, L. Fissore, P. Laface, A. Mertins, C. Ris *et al.*, "Automatic speech recognition and speech variability: A review," *Speech communication*, vol. 49, no. 10-11, pp. 763–786, 2007.

[5] R. C. van Dalen, K. M. Knill, and M. J. Gales, "Automatically grading learners' english using a Gaussian process." in *SLaTE*, 2015, pp. 7–12.

[6] A. Malinin, "Uncertainty estimation in deep learning with application to spoken language assessment," Ph.D. dissertation, University of Cambridge, 2019.

[7] X. Wu, K. M. Knill, M. J. Gales, and A. Malinin, "Ensemble approaches for uncertainty in spoken language assessment," *Proc. Interspeech 2020*, pp. 3860–3864, 2020.

[8] A. Malinin, A. Ragni, K. Knill, and M. Gales, "Incorporating uncertainty into deep learning for spoken language assessment," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017, pp. 45–50.

[9] N. Lawrence, M. Seeger, and R. Herbrich, "Fast sparse Gaussian process methods: The informative vector machine," in *Proceedings of the 16th annual conference on neural information processing systems*, no. CONF, 2003, pp. 609–616.

[10] M. Seeger, C. Williams, and N. Lawrence, "Fast forward selection to speed up sparse Gaussian process regression," Tech. Rep., 2003.

[11] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," *Advances in neural information processing systems*, vol. 18, pp. 1257–1264, 2005.

[12] T. D. Bui, J. Yan, and R. E. Turner, "A unifying framework for Gaussian process pseudo-point approximations using power expectation propagation," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 3649–3720, 2017.

[13] M. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in *Artificial intelligence and statistics*. PMLR, 2009, pp. 567–574.

[14] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, "Deep kernel learning," in *Artificial intelligence and statistics*. PMLR, 2016, pp. 370–378.

[15] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*. Springer, 2003, pp. 63–71.

[16] A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal processing magazine*, 2012.

[17] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: an ASR corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.
[18] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[20] GPy, "GPy: A gaussian process framework in python," http://github.com/SheffieldML/GPy, since 2012.
[21] R. Weide, "The Carnegie Mellon pronouncing dictionary [cmudict. 0.6]," *Pittsburgh, PA: Carnegie Mellon University*, 2005.