Incorporating Uncertainty into Deep Learning for Spoken Language Assessment

Andrey Malinin, Anton Ragni, Kate M. Knill, Mark J. F. Gales
University of Cambridge, Department of Engineering
Trumpington St, Cambridge CB2 1PZ, UK
{am969, ar527, kate.knill, mjfg}@eng.cam.ac.uk

Abstract

There is a growing demand for automatic assessment of spoken English proficiency. These systems need to handle large variations in input data owing to the wide range of candidate skill levels and L1s, and errors from ASR. Some candidates will be a poor match to the training data set, undermining the validity of the predicted grade. For high stakes tests it is essential for such systems not only to grade well, but also to provide a measure of their uncertainty in their predictions, enabling rejection to human graders. Previous work examined Gaussian Process (GP) graders which, though successful, do not scale well with large data sets. Deep Neural Networks (DNN) may also be used to provide uncertainty using Monte-Carlo Dropout (MCD). This paper proposes a novel method to yield uncertainty and compares it to GPs and DNNs with MCD. The proposed approach *explicitly* teaches a DNN to have low uncertainty on training data and high uncertainty on generated artificial data. On experiments conducted on data from the Business Language Testing Service (BULATS), the proposed approach is found to outperform GPs and DNNs with MCD in uncertainty-based rejection whilst achieving comparable grading performance.

1 Introduction

Systems for automatic assessment of spontaneous spoken language proficiency (Fig. 1) are becoming increasingly important to meet the demand for English second language learning. Such systems are able to provide throughput and consistency which are unachievable with human examiners. This is a challenging task. There is a large variation in the quality of spoken English across all proficiency levels. In addition, candidates of the same skill level will have different accents, voices, mispronunciations, and sentence construction errors. All of which are heavily influenced by the candidate's L1 language and compounded by ASR errors. It is therefore impossible in practice to observe all these variants in training. At test time, the predicted grade's validity will decrease the more the candidate is mismatched to the data used to train the system. For deployment of these systems to high-stakes tests the performance on all candidates needs to be consistent and highly correlated with human graders. To achieve this it is important that these systems can detect outlier speakers who need to be examined by, for example, human graders.

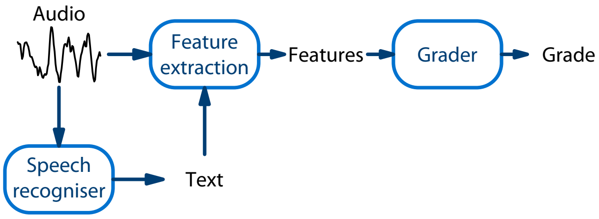


Figure 1: Automatic Assessment System

Previously, separate models were used to filter out ”non-scorable” candidates (Yoon and Xie, 2014; Zechner et al., 2009; Higgins et al., 2011; Xie et al., 2012). However, such models reject candidates based on whether they can be scored at all, rather than an automatic grader's uncertainty [1] in its predictions. It was shown by van Dalen et al. (2015) that Gaussian Process (GP) graders give

[1] Uncertainty is used in the sense of the inverse of confidence to be consistent with Gal and Ghahramani (2016) and van Dalen et al. (2015).

state-of-the-art performance for automatic assessment and yield meaningful uncertainty estimates for rejection of candidates. There are, however, computational constraints on training set sizes for GPs. In contrast, Deep Neural Networks (DNNs) are able to scale to large data sets, but lack a native measure of uncertainty. However, Gal and Ghahramani (2016) have shown that Monte-Carlo Dropout (MCD) can be used to derive an uncertainty estimate for a DNN.

Alternatively, a Deep Density Network (DDN), which is a Mixture Density Network (Bishop, 1994) with only one mixture component, may be used to yield a mean and variance corresponding to the predicted grade and the uncertainty in the prediction. Similar to GP and DNNs with MCD, a standard DDN provides an *implicit* modelling of uncertainty in its prediction. This implicit model may not be optimal for the task at hand. Hence, a novel approach to *explicitly* model uncertainty is proposed in which the DDN is trained in a multi-task fashion to model a low variance real data distribution and a high variance artificial data distribution which represents candidates with unseen characteristics.

2 Prediction Uncertainty

The principled method for dealing with uncertainty in statistical modelling is the Bayesian approach, where a conditional posterior distribution over grades, g, given inputs, $\boldsymbol{x}$, and training data $\mathcal{D} = \{\hat{g}, \hat{\boldsymbol{x}}\}$ is computed by marginalizing over all models:

$$\mathrm{p}(g|\boldsymbol{x},\mathcal{D}) = \int \mathrm{p}(g|\boldsymbol{x},\mathcal{M})\mathrm{p}(\mathcal{M}|\mathcal{D})\mathrm{d}\mathcal{M} \quad (1)$$

where $\mathrm{p}(\mathcal{M}|\mathcal{D})$ is a prior over a model given the data. Given the posterior, the predictive mean and the variance (uncertainty) can be computed using:

$$\mu_g(\boldsymbol{x}) = \int \mathrm{p}(g|\boldsymbol{x},\mathcal{D})g\mathrm{d}g \quad (2)$$

$$\sigma_g^2(\boldsymbol{x}) = \int \mathrm{p}(g|\boldsymbol{x},\mathcal{D})g^2\mathrm{d}g - \mu_g^2(\boldsymbol{x}) \quad (3)$$

2.1 Gaussian Processes

Eq. 2, 3 can be analytically solved for a class of models called Gaussian Processes (GP) (Rasmussen and Williams, 2006), a powerful non-parametric model for regression. The GP induces a conditional posterior in the form of a normal distribution over grades g given an input $\boldsymbol{x}$ and training data $\mathcal{D}$:

$$\mathrm{p}(g|\boldsymbol{x};\mathcal{D}) = \mathcal{N}(g;\mu_g(\boldsymbol{x}|\mathcal{D}),\sigma_g^2(\boldsymbol{x}|\mathcal{D})) \quad (4)$$

With mean function $\mu_g(\boldsymbol{x}|\mathcal{D})$ and variance function $\sigma_g^2(\boldsymbol{x}|\mathcal{D})$, which is a function of the similarity of an input $\boldsymbol{x}$ to the training data inputs $\hat{\boldsymbol{x}}$, where the similarity metric is defined by a covariance function $k(.,.)$. The nature of GP variance means that the model is uncertain in predictions for inputs far away from the training data, given appropriate choice of $k(.,.)$. Unfortunately, without sparsification approaches, the computational and memory requirements of GPs become prohibitively expensive for large data sets. Furthermore, GPs are known to scale poorly to higher dimensional features (Rasmussen and Williams, 2006).

2.2 Monte-Carlo Dropout

Alternatively, a grader can be constructed using Deep Neural Networks (DNNs) which have a very flexible architecture and scale well to large data sets. DNNs, however, lack a native measure of uncertainty. Uncertainty estimates for DNNs can be computed using a Monte-Carlo ensemble approximation to Eq. 2, 3:

$$\hat{\mu}_g(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x};\mathcal{M}^{(i)}) \quad (5)$$

$$\hat{\sigma}_g^2(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N}\Big(f(\boldsymbol{x};\mathcal{M}^{(i)})\Big)^2 - \hat{\mu}_g^2(\boldsymbol{x}) \quad (6)$$

where there are N DNN models in the ensemble, $\mathcal{M}^{(i)}$ is a DNN with a particular architecture and parameters sampled from $\mathrm{p}(\mathcal{M}|\mathcal{D})$ using Monte Carlo Dropout (MCD) (Srivastava et al., 2014), and $f(\boldsymbol{x};\mathcal{M}^{(i)})$ are the DNN predictions. Recent work by Gal and Ghahramani (2016) showed that MCD is equivalent to approximate variational inference in GPs, and can be used to yield meaningful uncertainty estimates for DNNs. Furthermore, Gal and Ghahramani (2016) show that different choices of DNN activation functions correspond to different GP covariance functions. MCD uncertainty assumes that for inputs further from the training data, different subnets will produce increasingly differing outputs, leading to larger variances. Unfortunately, it is difficult to know beforehand which activation functions accomplish this in practice.

3 Deep Density Networks

Instead of relying on a Monte Carlo approximation to Eq. 1, a DNN can be modified to produce a prediction of both a mean and a variance:

$$\mu_g(\boldsymbol{x}) = f_\mu(\boldsymbol{x}; \mathcal{M}) \tag{7}$$

$$\sigma_g^2(\boldsymbol{x}) = f_{\sigma^2}(\boldsymbol{x}; \mathcal{M}) \tag{8}$$

parametrising a normal distribution over grades conditioned on the input, similar to a GP. This architecture is a Deep Density Network (DDN), which is a Mixture Density Network (MDN) (Bishop, 1994) with only one mixture component. DDNs are trained by maximizing the likelihood of the training data. The variance of the DDN represents the natural spread of grades at a given input. This is an *implicit* measure of uncertainty, like GP and MCD variance, because it is learned automatically as part of the model. However, this doesn't enforce higher variance further away from training points in DDNs. It is possible to *explicitly* teach a DDN to predict a high or low variance for inputs which are unlike or similar to the training data, respectively (Fig. 2). This requires a novel training procedure. Two normal distributions are constructed: a low-variance real (training) data distribution $\mathrm{p_D}$ and a high-variance artificial data distribution $\mathrm{p_N}$, which models data outside the real training data region. The DDN needs to model both distributions in a multi-task (MT) fashion. The loss function for training the DDN with explicitly specified uncertainty is the expectation over the training data of the KL divergence between the distribution it parametrizes and both the real and artificial data distributions:

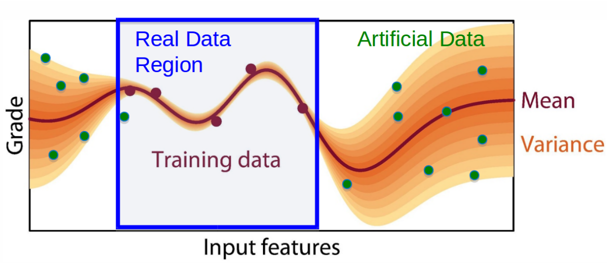


Figure 2: Desired variance characteristic

$$\begin{aligned}\mathcal{L} = \quad & \mathrm{E}_{\hat{\mathrm{x}}}[\mathrm{KL}(\mathrm{p_D}||\mathrm{p}(g|\hat{\boldsymbol{x}}; \mathcal{M})] + \\ & \alpha \cdot \mathrm{E}_{\tilde{\mathrm{x}}}[\mathrm{KL}(\mathrm{p_N}||\mathrm{p}(g|\tilde{\boldsymbol{x}}; \mathcal{M})]\end{aligned} \tag{9}$$

where α is the multi-task weight.

The DDN with explicit uncertainty is trained in a two stage fashion. First, a standard DDN $\mathcal{M}_0$ is trained, then a DDN $\mathcal{M}$ is instantiated using the parameters of $\mathcal{M}_0$ and trained in a multi-task fashion. The real data distribution $\mathrm{p_D}$ is defined by $\mathcal{M}_0$ (Eq. 7, 8). The artificial data distribution $\mathrm{p_N}$ is constructed by generating artificial inputs $\tilde{\boldsymbol{x}}$ and the associated mean and variance targets $\mu(\tilde{\boldsymbol{x}}), \sigma^2(\tilde{\boldsymbol{x}})$:

$$\mathrm{p_N} = \mathcal{N}(g; f_\mu(\tilde{\boldsymbol{x}}; \mathcal{M}_0), \sigma^2(\tilde{\boldsymbol{x}})) \tag{10}$$

The predictions of $\mathcal{M}_0$ are used as the targets for $\mu(\tilde{\boldsymbol{x}})$. The target variance $\sigma^2(\tilde{\boldsymbol{x}})$ should depend on the similarity of $\tilde{\boldsymbol{x}}$ to the training data. Here, this variance is modelled by the squared normalized Euclidean distance from the mean of $\hat{\boldsymbol{x}}$, with a diagonal covariance matrix, scaled by a hyperparameter λ. The artificial inputs $\tilde{\boldsymbol{x}}$ need to be different to, but related to the real data $\hat{\boldsymbol{x}}$. Ideally, they should represent candidates with unseen characteristics, such as L1, accent and proficiency. A simple approach to generating $\tilde{\boldsymbol{x}}$ is to use a Factor Analysis (FA) (Murphy, 2012) model trained on $\hat{\boldsymbol{x}}$. The generative model of FA is:

$$\tilde{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \gamma\boldsymbol{\Psi}),\ \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \gamma\boldsymbol{I}) \tag{11}$$

where $\boldsymbol{W}$ is the loading matrix, $\boldsymbol{\Psi}$ the diagonal residual noise variance, $\boldsymbol{\mu}$ the mean, all derived from $\hat{\boldsymbol{x}}$, and γ is used to control the distance of the generated data from the real training data region. During training the artificial inputs are sampled from the FA model.

4 Experimental Results

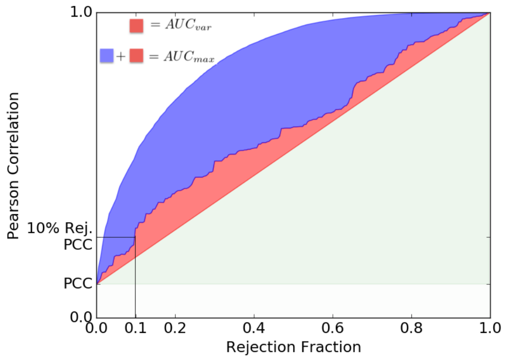
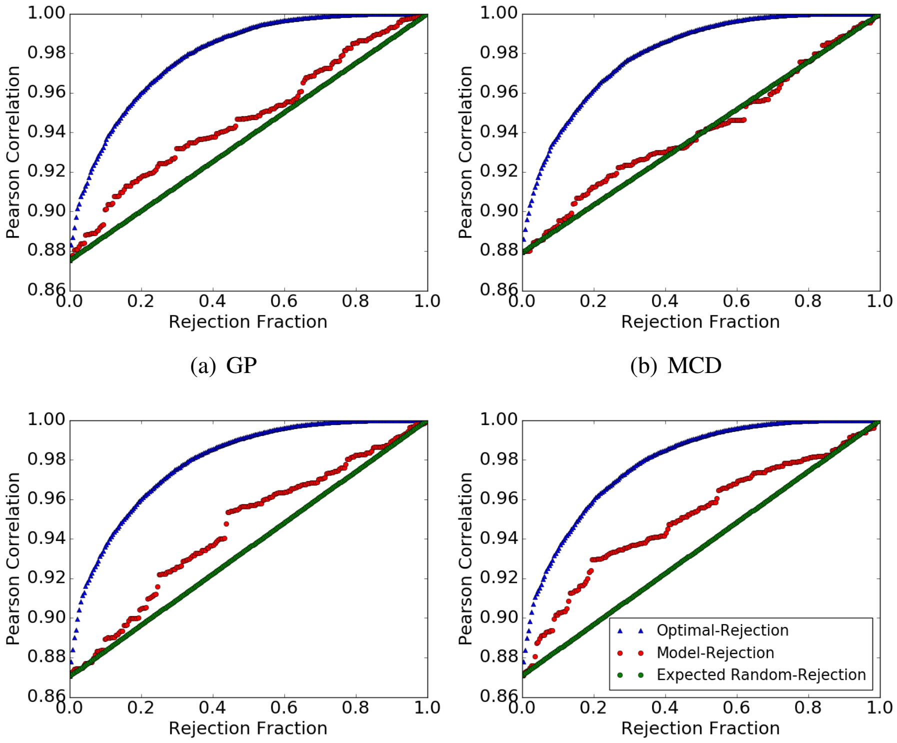


Figure 3: An example Rejection Plot

$$\mathrm{AUC_{RR}} = \frac{\mathrm{AUC_{var}}}{\mathrm{AUC_{max}}} \tag{12}$$

As previously stated, the operating scenario is to use a model's estimate of the uncertainty in



(a) GP

(b) MCD

(c) DDN

(d) DDN+MT

Figure 4: Rejection Plots for models

its prediction to reject candidates to be assessed by human graders for high-stakes tests, maximizing the increase in performance while rejecting the least number of candidates. The rejection process is illustrated using a rejection plot (Fig. 3). As the rejection fraction is increased, model predictions are replaced with human scores in some particular order, increasing overall correlation with human graders. Fig. 3 has 3 curves representing different orderings: expected random rejection, optimal rejection and model rejection. The expected random performance curve is a straight line from the base predictive performance to 1.0, representing rejection in a random order. The optimal rejection curve is constructed by rejecting predictions in order of decreasing mean square error relative to human graders. A rejection curve derived from a model should sit between the random and optimal curves. In this work, model rejection is in order of decreasing predicted variance.

The following metrics are used to assess and compare models: Pearson Correlation Coefficient (PCC) with human graders, the standard performance metric in assessment (Zechner et al., 2009; Higgins et al., 2011); 10% rejection PCC, which illustrates the predictive performance at a particular operating point, i.e. rejecting 10% of candidates; and Area under a model's rejection curve (AUC) (Fig 3). However, AUC is influenced by the base PCC of a model, making it difficult to compare the rejection performance. Thus, a metric independent of predictive performance is needed. The proposed metric, $\texttt{AUC}_{\texttt{RR}}$ (Eq. 12), is the ratio of the areas under the actual ($\texttt{AUC}_{\texttt{var}}$) and optimal ($\texttt{AUC}_{\texttt{max}}$) rejection curves relative to the random rejection curve. Ratios of 1.0 and 0.0 correspond to perfect and random rejection, respectively.

All experiments were done using 33-dimensional pronunciation, fluency and acoustic features derived from audio and ASR transcriptions of responses to questions from the BULATS exam (Chambers and Ingham, 2011). The ASR system has a WER of 32% on a development set. The training and test sets have 4300 and 224 candidates, respectively. Each candidate provided a response to 21 questions, and the features used are aggregated over all 21 questions into a single feature vector. The test data was graded by expert graders at Cambridge English. These experts have inter-grader PCCs in the range 0.95-0.97. Candidates are equally distributed across CEFR grade levels (Europe, 2001).

The input features where whitened by subtracting the mean and dividing by the standard deviation for each dimension computed on all training speakers. The Adam optimizer (Kingma and Ba, 2015), dropout (Srivastava et al., 2014) regularization with a dropout keep probability of 0.6 and an exponentially decaying learning rate are used with decay factor of 0.86 per epoch, batch size 50. All networks have 2 hidden layers with 180 rectified linear units (ReLU) in each layer. DNN and DDN models were implemented in Tensorflow (Abadi et al., 2015). Models were initialized using the Xavier Initializer (Glorot and Bengio, 2010). A validation set of 100 candidates was selected from the training data to tune the model and hyper-parameters. GPs were run using Scikit-Learn (Pedregosa et al., 2011) using a squared exponential covariance function.

Grader	PCC	10% Rej. PCC	$\mathtt{AUC}$	$\mathtt{AUC}_{\mathtt{RR}}$
GP	0.876	0.897	0.942	0.233
MCD	0.879	0.892	0.937	0.040
MCD$_{\mathtt{tanh}}$	0.865	0.886	0.938	0.226
DDN	0.871	0.887	0.941	0.230
+MT	0.871	0.902	0.947	0.364

Table 1: Grading and rejection performance

The Gaussian Process grader, GP, is a competitive baseline (Tab. 1). GP variance clearly yields uncertainty which is useful for rejection. A DNN with ReLU activation, MCD, achieves grading performance similar to the GP. However, MCD fails to yield an informative uncertainty for rejection, with performance barely above random. If the `tanh` activation function, MCD$_{\mathtt{tanh}}$, is used instead, then a DNN is able to provide a meaningful measure of uncertainty using MCD, at the cost of grading performance. It is likely that ReLU activations correspond to a GP covariance function which is not suited for rejection on this data.

The standard DDN has comparable grading performance to the GP and DNNs. $\mathtt{AUC}_{\mathtt{RR}}$ of the DDN is on par with the GP, but the 10% rejection PCC is lower, indicating that the DDN is not as effective at rejecting the worst outlier candidates. The approach proposed in this work, a DDN trained in a multi-task fashion (DDN+MT), achieves significantly higher rejection performance, resulting in the best $\mathtt{AUC}_{\mathtt{RR}}$ and 10% rejection PCC, showing its better capability to detect outlier candidates. Note, $\mathtt{AUC}$ reflects similar trends to $\mathtt{AUC}_{\mathtt{RR}}$, but not as clearly, which is demonstrated by Fig. 4. The model was found to be insensitive to the choice of hyper-parameters α and γ, but λ needed to be set to produce target noise variances $\sigma^2(\tilde{\boldsymbol{x}})$ larger than data variances $\sigma^2(\hat{\boldsymbol{x}})$.

5 Conclusions and Future Work

A novel method for explicitly training DDNs to yield uncertainty estimates is proposed. A DDN is a density estimator which is trained to model two distributions in a multi-task fashion (1) the low variance (uncertainty) true data distribution and (2) a generated high variance artificial data distribution. The model is trained by minimizing the KL divergence between the DDN and the true data distribution (1) and between the DDN and the artificial data distribution (2). The DDN should assign its prediction of low or high variance (uncertainty) if the input is similar or dissimilar to the true data respectively. The artificial data distribution is given by a factor analysis model trained on the real data. During training the artificial data is sampled from this distribution.

This method outperforms GPs and Monte-Carlo Dropout in uncertainty based rejection for automatic assessment. However, the effect of the nature of artificial data on rejection performance should be further investigated and other data generation methods, such as Variational Auto-Encoders (Kingma and Welling, 2014), and metrics to assess similarity between artificial and real training data should be examined. The proposed approach must also be assessed on other tasks and datasets.

Acknowledgments

This research was funded under the ALTA Institute, University of Cambridge as well as the Engineering and Physical Sciences Research Council. Thanks to Cambridge English, University of Cambridge, for support and access to the BULATS data.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. http://tensorflow.org/.

C. M. Bishop. 1994. Mixture density networks. *Technical Report NCRG 4288, Neural Computing Research Group, Department of Computer Science, Aston University* .

Lucy Chambers and Kate Ingham. 2011. The BULATS online speaking test. *Research Notes* 43:21–25.

Council of Europe. 2001. *Common European framework of reference for languages: Learning, teaching, assessment*. Cambridge, U.K: Press Syndicate of the University of Cambridge.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML-16)*.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*. volume 9, pages 249–256.

Derrick Higgins, Xiaoming Xi, Klaus Zechner, and David Williamson. 2011. A three-stage approach to the automated scoring of spontaneous spoken responses. *Computer Speech and Language* 25(2):282–306.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Diederik P. Kingma and Max Welling. 2014. Autoencoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*.

Kevin P. Murphy. 2012. *Machine Learning*. The MIT Press.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Carl Edward Rasmussen and Christopher K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. MIT Press.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(1):1929–1958.

Rogier C. van Dalen, Kate M. Knill, and Mark J. F. Gales. 2015. Automatically Grading Learners' English Using a Gaussian Process. In *Proceedings of the ISCA Workshop on Speech and Language Technology for Education (SLaTE)*.

Shasha Xie, Keelan Evanini, and Klaus Zechner. 2012. Exploring Content Features for Automated Speech Scoring. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Su-Youn Yoon and Shasha Xie. 2014. Similarity-Based Non-Scorable Response Detection for Automated Speech Scoring. In *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications*.

Klaus Zechner, Derrick Higgins, Xiaoming Xi, and David M. Williamson. 2009. Automatic scoring of non-native spontaneous speech in tests of spoken english. *Speech Communication* 51(10):883–895. Spoken Language Technology for Education Spoken Language.