MEASURING UNCERTAINTY IN DEEP REGRESSION MODELS: THE CASE OF AGE ESTIMATION FROM SPEECH

Nanxin Chen[†] *Jesús Villalba*[†] *Yishay Carmiel*[⋆] *Najim Dehak*[†]

[†] Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD 21218, USA
[⋆] IntelligentWire, Seattle, WA 98121, USA

{bobchennan, jvillal7, ndehak3}@jhu.edu, ycarmiel@intelligentwire.com

ABSTRACT

Age estimation from speech recently received a lot of attention. Approaches such as i-vectors and deep learning have been successfully applied to this task achieving great performance. However, one drawback of those methods is that they produce a hard age estimation without any kind of confidence measure about the quality of the prediction. Designing systems with the ability to provide a confidence measure about their output is extremely valuable for several applications where the cost of making bad decisions is worse than making no decision, e.g., forensics. In this paper, we propose a novel framework to jointly predict the age and its estimation uncertainty in a context of neural regression model. This model is trained using probabilistic fashion instead of using the classical minimum mean square error objective used for regression tasks. The probabilistic output corresponds to a Gaussian posterior. The proposed neural network will estimate both the posterior mean which corresponds to the predicted age and the variance which quantifies the uncertainty of the prediction. We evaluated our approach on two different datasets NIST SRE 2008 - 2010 and Switchboard.

Index Terms— uncertainty estimation, age estimation, deep learning, RNN, LSTM

1 Introduction

Given the ubiquity of mobile phones and home assistant devices allowing human-machine interaction, there is a growing interest on extracting other relevant information from speech other than the message itself. For example, the speaker identity, ethnicity, emotion and age can be used to offer tailored product and services to customers [1]. This information can help to optimally pair customer and agents in call centers scenario. Nevertheless, we can also use it in forensics setting to narrow the list of suspects in a criminal investigation.

In this paper, we focus on the age estimation problem. Approaches like neural networks [2] and i-vectors [3, 4] have been largely explored to estimate the age. Deep learning has been extremely successful in most speech related areas like speech recognition [5], speaker recognition [6, 7] and speech synthesis [8]. Training a neural network based regression model to predict the speaker's age is straightforward. Despite the promising results, these models still make errors, which can be very harmful [9]. We would like the model to provide a measure indicating the uncertainty of the prediction, i.e., whether the prediction is reliable or not. We could use this uncertainty to establish a confidence interval for the prediction.

Beyond the theoretical interest of this issue, it has a very wide range of real applications. An evident example is forensics. In court, it is not enough to provide a prediction of the suspect's age but the judge will want to know how accurate the prediction is. In this case, the age uncertainty is equivalent to the log-likelihood ratio used in speaker recognition task. Log-ratios close to zero indicate that we are not sure about the speaker identity while high log-ratios (positive or negative) indicate that we are confident about the target or non-target hypothesis. Another example is a call center where agents are specialized to deal with different age ranges. If the system is not confident about the customer age, it will send the call to a generic agent.

In Bayesian modelling there are two types of uncertainty [10]:

- Epistemic: it measures the ignorance of the data generating process.
- Aleatoric: it captures our uncertainty with respect to information which the data cannot explain. It can be further divided into two types: task-dependent or Homoscedastic uncertainty which is constant for all inputs; and data-dependent or Heteroscedastic uncertainty which depends on how noisy input is.

In previous work, Gal [11] shows the connection between dropout and Bayesian inference in deep Gaussian processes. The Bayesian interpretation of dropout enables to predict epistemic uncertainty. There are also several works using Bayesian networks to measure uncertainty [12, 13]. In speech recognition, uncertainty is used to improve noise robustness [14]. Also, in speaker verification the uncertainty about the likelihood ratio is used to decide whether the decisions are reliable or not [15, 16].

In this paper, we consider aleatoric uncertainty. Though we focus on age estimation, this framework is general enough for any regression problem. This is a representative sequential regression problem where, given a sequence of feature frames, we intend to jointly predict the speaker age and its uncertainty.

The rest of the paper is organized as follows. Section 2 presents the regression model and its extension to estimate uncertainty. Section 3 describes the experimental setup and present results on NIST and switchboard databases. Finally, Section 4 summarizes the paper and proposes further discussion.

2 Joint age and uncertainty estimation

2.1 Maximum likelihood regression

The typical loss function for regression problems consists in minimizing the sum-of-squares error,

$$\text{MSE} = \sum_{i=1}^{N} (t_i - y(\mathbf{x}_i))^2 \tag{1}$$

where t_i is the true value, $\mathbf{x}_i$ is the input feature and y is the prediction function.

We can view MSE from a probabilistic point of view. Let's assume that the posterior distribution of t_i given and observed feature $\mathbf{x}_i$ is Gaussian with mean $y(\mathbf{x}_i)$,

$$P(t_i|\mathbf{x}_i) = \mathcal{N}(t_i|y(\mathbf{x}_i), \sigma^2) \tag{2}$$

where the constant variance σ^2 measures the uncertainty of the prediction. Now, to estimate the parameters of the predictor, we maximize the log-likelihood of the training data $\{\mathbf{X}, \mathbf{T}\}$,

$$\log P(\mathbf{T}|\mathbf{X}) = -\frac{1}{2\sigma^2} \sum_{i=1}^{N} (t_i - y(\mathbf{x}_i))^2 - \frac{N}{2} \log(2\pi\sigma^2) \ . \tag{3}$$

As σ^2 is a constant which does not depend on any input $\mathbf{x}_i$, it is evident that maximizing (3) is equivalent to minimizing the MSE in (1).

We can also maximize the likelihood w.r.t. σ^2 to get the closed form expression

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (y(\mathbf{x}_i) - t_i)^2 \ . \tag{4}$$

This is a case of homoscedastic uncertainty, i.e., the uncertainty is the same no matter which signal we have in the input. This is very limited given that different signals involve different difficulties. For the case of age, we have noted that young age predictions are more accurate than those of elder people because of the lack of data for aged people. Also, noisy speech implies more uncertainty than clean speech.

We can estimate the variance on the own training set but may be better to use a held-out set for this purpose.

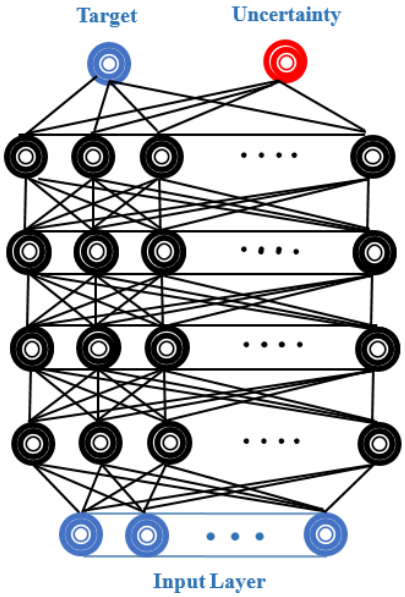

Fig. 1. Neural regressor with uncertainty estimation. The network structure can be of various types, e.g., RNN and feed-forward network.

2.2 Heteroscedastic uncertainty

We can extend the previous model to consider heteroscedastic uncertainty, i.e., having a different uncertainty estimation for each recording. To accomplish this goal, we just modified the objective function as,

$$\log P(\mathbf{T}|\mathbf{X}) = \sum_{i=1}^{N} \log \mathcal{N}(t_i|y(\mathbf{x}_i), \sigma^2(\mathbf{x}_i)) \tag{5}$$

where in this case σ^2 is a function that depends on the input $\mathbf{x}_i$. We will jointly optimize the parameters of the functions y and σ^2. In practice, we used a neural network with two different outputs, one for mean y and one for variance σ^2; as shown in Figure 1. Thus, we can optimize the objective by stochastic back-propagation methods.

This framework allows us to provide sample dependent confidence intervals. For example, as we assumed a Gaussian posterior, we can say that predicted age is $y \pm 2\sigma$ with 95% of confidence.

2.3 LSTM network

Long-short term memory networks (LSTM) [17] are a type of recurrent neural network that attains great performance in sequence modeling problems [18, 19]. Each LSTM layer consists of a structure, which contains a memory cell. This cell can accumulate information during a long period of time. It also includes 'gate' units–input, forget and output– which decide when to write, delete or propagate the information of the memory cell. LSTMs overcome the vanishing gradient problem better than basic RNNs.

Preliminary experiments comparing feed-forward networks, TDNN [20, 5] and LSTM showed better performance of the latter. Thus, we used LSTM networks for this paper experiments. However, we remark that we present a general framework that can be applied with other architectures.

2.4 Variance estimation for sequences

Since speech is sequential data, for most network architectures we will get and age prediction per frame. For LSTM networks, it is also common practice, given that they accumulate the sequence information in its memory cells, to provide one prediction per sequence. However, LSTMs still have a limited memory span, and cannot cope with very long sequences. In this case, it is common practice to produce a prediction per chunk of frames, e.g., one value every 5 seconds.

Thus, for each frame or chunk we obtain a different age and uncertainty value. The question is now how to compute the age and uncertainty for the whole utterance. Given that the age prediction is a Gaussian posterior, we can assume that sequence posterior is the average of Gaussian variables. The average of Gaussian variables is also Gaussian with mean

$$y = \frac{1}{M}\sum_{j=1}^{M} y_j \tag{6}$$

where M is number of frames or chunks.

Meanwhile, the variance of the sequence depends on the degree of correlation between predictions. If we consider, that all prediction are independent between them,

$$\sigma^2 = \frac{1}{M^2}\sum_{j=1}^{M} \sigma_j^2 \ . \tag{7}$$

However, as all the predictions intend to predict the same age, we should expect some correlation between predictions. With correlation coefficient ρ and $M = 2$, the variance is

$$\sigma^2 = \frac{1}{4}(\sigma_1^2 + \sigma_2^2 + 2\rho\sigma_1\sigma_2) \ . \tag{8}$$

In case maximum correlation, i.e. $\rho = 1$, the sequence variance is

$$\sigma^2 = \frac{1}{M^2}(\sum_{j=1}^{M} \sigma_j)^2 \ . \tag{9}$$

3 Experiments

3.1 General experimental setup

We experimented on NIST SRE 2008-10 (in-domain) and Switchboard (out-of-domain) datasets.

In previous works, performance was reported in terms of mean absolute error (MAE). However, this measure does not take into account the uncertainty, so it is not useful for our purposes. Instead, we propose to use the likelihood of evaluation data given the predicted posterior distributions (mean and uncertainty) in (5). In fact, MAE approximates the square root of the likelihood when $\sigma = 1$, so they are related. We take the homeostatic framework as baseline–equivalent to minimum square error– and show that the heteroscedastic attains better performance.

Model	Log Likelihood
Homo(Train)	-3.70
Homo(Val)	-3.72
Hetero last-frame	-4.49
Hetero frame-avg ($\rho = 1$)	**-3.45**
Hetero chunk-avg ($\rho = 1$)	-3.47
Hetero frame-avg ($\rho = 0.75$)	-5.32
Hetero chunk-avg ($\rho = 0.75$)	-3.58
Hetero frame-avg ($\rho = 0$)	-7208.27
Hetero chunk-avg ($\rho = 0$)	-13.83

Table 1. Likelihood comparison on NIST SRE data. Averaged over 15 folds.

For all experiments, the features were 20 MFCC with appended first and second derivatives plus probability of voicing (POV), pitch and delta-pitch [21]. Short-time cepstral mean variance normalization (CMVN) was used with 3 second sliding window.

The regressor consisted of two LSTM layers with 256 neurons as well as two ReLU fully connected layers with 512 and 256 neurons respectively. Finally, two linear output layers predict the age mean and log-variance. The LSTM layers used dropout with 30% drop rate [22]. Adam optimizer [23] was used with 0.002 learning rate. Similar to [24], the targets are normalized to zero mean and unit variance. We considered the case where the LSTM provides a prediction per frame, and the case where it provides a prediction every 5 seconds (chunks). Training was done feeding 5 second chunks to the network.

3.2 In-domain experiments

We experimented on NIST SRE 2008-10 which has 1597 speakers between 20 to 70 years old and 9442 telephone utterances with 8 kHz sampling rate. For consistency, we used the same experimental setup as in [4, 2]. Data was divided into 15 folds without overlapped speakers. 15 independent test are executed: each time we train on 14 folds and evaluate in the remain fold.

Table 1 reports the log-likelihood in (5) for different cases. Homo(Train) denotes the baseline system with constant variance estimated on the training set while Homo(Val) estimate the variance on held-out validation data. Hetero denotes our proposed approach. We consider several cases: last-frame which just takes the output of the last frame of the sequence; frame-avg which makes decisions at a frame level basis; and chunk-avg which splits the utterance into 5 second segments and gets predictions from the last frame of each segment. In the frame-avg and chunk-avg the sequence level variance is

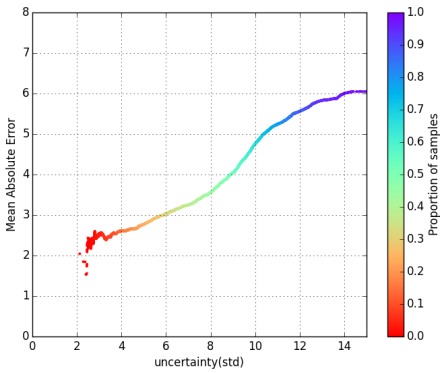

Fig. 2. How mean absolute error changed with various uncertainty range. Color indicates the proportion of samples in this range.

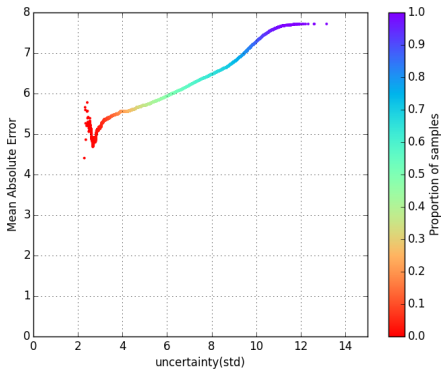

Fig. 3. How mean absolute error changed with various uncertainty range for out-of-domain data (Switchboard data).

computed using equations in Section 2.4. The heteroscedastic model outperforms the baselines (frame-avg) for both frame and chunk level predictions if we assume strong correlation between predictions. Evaluating in the last frame performs poorly because the sequences are too long for the LSTM memory span. Assuming independence across predictions also performs poorly because it leads us to be over-confident about the prediction.

Next, we consider the utility of the uncertainty for a real application where the cost of taking a bad decision is larger than not taking any decision, e.g., forensics. In this case we put a threshold τ on the uncertainty–we use the standard deviation here σ–, and reject all the decisions whose uncertainty is larger than the threshold. To evaluate how good the uncertainty is, we report the mean absolute error on the utterances that we consider reliable:

$$\text{MAE}(\tau) = \frac{1}{\sum_{\sigma_i \leq \tau} \mathbb{1}} \sum_{\sigma_i \leq \tau} |t_i - y_i| \,. \qquad (10)$$

This criteria shows the correlation between uncertainty and the real metric (MAE).

Figure 2 shows the result for the best system on Table 1. The x-axis indicates the threshold for uncertainty and y-axis represents the mean absolute error calculated by (10). The colormap shows the proportion of data whose uncertainty is equal or less than the threshold. The figure shows that our method reduces MAE from more than 6 to around 3 by keeping the 30% segments with lower uncertainty.

Model	Log Likelihood
Homo(Train)	-3.84
Homo(Val)	-4.80
Hetero frame-avg ($\rho = 1$)	**-3.75**

Table 2. Likelihood comparison on Switchboard data.

3.3 Out-of-domain experiments

An interesting topic is whether the estimated uncertainty can be generalized to another corpus. To show the generalization of our framework, we used the Switchboard data. The SWB data contains 1,962 speakers between 14 to 85 years old and 20,905 utterances of SWB Cellular and SWB 2 Phases II and III. Table 2 reports the likelihood on the Switchboard corpus. Again, our method outperforms the baseline.

Figure 3 plots MAE versus uncertainty threshold. We observe that we cannot reduce the MAE as much as in the NIST data. However, using the threshold that provides MAE=3 on NIST we reduce Switchboard MAE from 8 to around 6.

To be noted, regardless of in-domain or out-of-domain, the slope is always less than 1 which means the actual mean absolute error is less than the uncertainty that we estimated.

4 Conclusion and Future Work

This paper proposes a new framework to predict uncertainty in neural-based regression systems. Likelihood is optimized instead of original mean square error to jointly optimize the target value and uncertainty estimation. We adopted speech-based age estimation as the study case. We show how to deal with uncertainty in the case of sequences where we obtain frame level predictions and need to compute a global uncertainty for the whole sequence. We experimented on NIST SRE (in-domain) and Switchboard datasets (out-of-domain). In both cases, utterances with low uncertainty provide lower mean absolute error than utterances with high uncertainty which proofs the utility of our method.

5 References

[1] Björn Schuller, Stefan Steidl, Anton Batliner, Felix Burkhardt, Laurence Devillers, Christian MüLler, and Shrikanth Narayanan, "Paralinguistics in speech and languagestate-of-the-art and the challenge," *Computer Speech & Language*, vol. 27, no. 1, pp. 4–39, 2013.

[2] Anna Fedorova, Ondřej Glembek, Tomi Kinnunen, and Pavel Matějka, "Exploring ann back-ends for i-vector based speaker age estimation," in *Proc. InterSpeech*, 2015.

[3] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[4] Mohamad Hasan Bahari, Mitchell McLaren, David A van Leeuwen, et al., "Speaker age estimation using i-vectors," *Engineering Applications of Artificial Intelligence*, vol. 34, pp. 99–108, 2014.

[5] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi.," in *Proc. InterSpeech*, 2016, pp. 2751–2755.

[6] Yuan Liu, Yanmin Qian, Nanxin Chen, Tianfan Fu, Ya Zhang, and Kai Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.

[7] David Snyder, Pegah Ghahremani, Daniel Povey, Daniel Garcia-Romero, Yishay Carmiel, and Sanjeev Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Spoken Language Technology Workshop, 2016 IEEE*, 2016, pp. 165–170.

[8] Zhizheng Wu and Simon King, "Improving trajectory modelling for dnn-based speech synthesis by using stacked bottleneck features and minimum trajectory error training," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 2016.

[9] Jessica Guynn, "Google photos labeled black people 'gorillas'," 2015.

[10] Armen Der Kiureghian and Ove Ditlevsen, "Aleatory or epistemic? does it matter?," *Structural Safety*, vol. 31, no. 2, pp. 105–112, 2009.

[11] Yarin Gal and Zoubin Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[12] David Ha, Andrew Dai, and Quoc V Le, "Hypernetworks," *arXiv preprint arXiv:1609.09106*, 2016.

[13] David Krueger, Chin-Wei Huang, Riashat Islam, Ryan Turner, Alexandre Lacoste, and Aaron Courville, "Bayesian hypernetworks," *arXiv preprint arXiv:1710.04759*, 2017.

[14] Hank Liao and MJF Gales, "Issues with uncertainty decoding for noise robust automatic speech recognition," *Speech Communication*, vol. 50, no. 4, pp. 265–277, 2008.

[15] Jesús Villalba, Eduardo Lleida, Alfonso Ortega, and Antonio Miguel, "A New Bayesian Network to Assess the Reliability of Speaker Verification Decisions," in *Proc. InterSpeech*, 2013, pp. 3132 – 3136.

[16] Jesus Villalba, Antonio Miguel, Alfonso Ortega, and Eduardo Lleida, "Bayesian Networks to Model the Variability of Speaker Verification Scores in Adverse Environments," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2327–2340, 2016.

[17] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] Felix A Gers and Jürgen Schmidhuber, "Lstm recurrent networks learn simple context-free and context-sensitive languages," *Neural Networks, IEEE Transactions on*, vol. 12, no. 6, pp. 1333–1340, 2001.

[19] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney, "Lstm neural networks for language modeling.," in *Proc. InterSpeech*, 2012, pp. 194–197.

[20] Vijayaditya Peddinti, Vimal Manohar, Yiming Wang, Daniel Povey, and Sanjeev Khudanpur, "Far-field asr without parallel data.," in *Proc. InterSpeech*, 2016, pp. 1996–2000.

[21] Pegah Ghahremani, Bagher BabaAli, Daniel Povey, Korbinian Riedhammer, Jan Trmal, and Sanjeev Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *Proc. ICASSP*. IEEE, 2014, pp. 2494–2498.

[22] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[23] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] Seyed Omid Sadjadi, Sriram Ganapathy, and Jason W Pelecanos, "Speaker age estimation on conversational telephone speech using senone posterior based i-vectors," in *Proc. ICASSP*, 2016, pp. 5040–5044.