PERFORMANCE MONITOR TECHNIQUES FOR NOISE ROBUST SPEECH RECOGNITION

Doctoral Thesis Proposal

Sri Harish Mallidi
Department of Electrical and Computer Engineering
Johns Hopkins University
mallidi@jhu.edu

Hynek Hermansky (Advisor)
Sanjeev Khudanpur
Gerard Meyer
{hynek,sanjeev,gglmeyer}@jhu.edu

Contents

ABSTRACT

Performance of automatic speech recognition (ASR) systems degrade rapidly when there is a mismatch between test and training acoustic conditions. Knowledge about the performance of an ASR system for an unknown test utterance can be useful in improving the robustness of ASR system. In this work we propose two techniques to estimate the performance of ASR system.

The first technique, namely, M-delta measure is an extension of the previously proposed M-measure. The M-measure, predicts confidence in the output of a probability estimator by measuring divergences of probability estimates spaced at specific time intervals. The M-measure is improved by explicitly taking into account the probability that distant frames have different phoneme labels, providing a more accurate indicator of the estimator's ability to distinguish between phonemes. The proposed techniques for confidence estimation are evaluated in a multistream-based adaptation paradigm [2].

A new autoencoder based technique for estimating the performance of ASR systems is also proposed in this work. For a well trained autoencoder, reconstruction error of a vector sampled from the training distribution will be small compared to a vector sampled from a different distribution. Statistics computed from the reconstruction error are used as estimates of the accuracy of a given test utterance. The proposed technique is used to predict performance of deep neural network based large vocabulary continuous speech recognition system. In terms of correlation with word error rate, the proposed technique performs better than M-measure.

1. INTRODUCTION

The task of an automatic speech recognition (ASR) system is to estimate the message content of speech signal. The dominant paradigm in literature to perform this task is a stochastic framework. The basic principle is as follows: Acoustic and language models are built using training data. The trained models are then used to find the most likely word sequence of a given test signal. The key equation is

$$\hat{W} = \arg\max_{W} P(X|W)\ P(W) \tag{1}$$

where X represents sequence of acoustic features of the test signal, $P(X|W)$ is the likelihood of X using model corresponding to word sequence W. The term $P(W)$ is the probability of the word sequence W. The emergence of sophisticated modeling techniques led ASR algorithms to perform really well on controlled settings [1]. The fundamental assumption in the controlled settings, is test data is sampled from the distribution of the training data. In effect, the algorithm is assuming the type of distortions it encounters in testing phase are present in the training data. Performance of ASR algorithms can degrade rapidly if this assumption is violated. This happens quite often in real speech, as signals may be corrupted by sources that were not seen during training phase.

1.1. Objective of the work

Human speech recognition is quite robust to unexpected variations. The presence of parallel processing streams and ability to monitor the confidence of decisions are crucial to the robustness of human speech recognition [2]. Aim of the present work is to investigate confidence monitoring techniques for automatic speech recognition systems to make them more robust to unexpected noises. Performance on a test data is correlated with confidence of decisions made by the ASR system. We propose measures to estimate the performance of the classifier. Goal of the performance estimation block is to predict the accuracy of a test utterance, without the knowledge of its labels.

1.2. Previous approaches:

Several techniques were proposed in the literature to estimate the performance of ASR system. Comparison of class conditional likelihoods of highest-probabilty estimate to several next lower ones is used in [6]. Entropy of phoneme posterior probability vector, computed at each frame is used as the estimate of the performance in [7, 8]. A related technique, based on the autocorrelation of the phoneme posterior probability vector was studied in [9, 10]. A technique which uses average dissimilarities of probability vectors spaced in several time spans was proposed in [11]. A Gaussian mixture model (GMM) based performance estimation technique was proposed in [12]. In [12], transformed posterior probability vectors are modeled using single-state GMM. For a test utterance, likelihood of GMM is used as an estimator of accuracy.

1.3. Main contribution

In previous section, we described the importance of performance monitor block in ASR systems and provided brief description of existing performance monitor approaches in literature. In this work, we propose two new performance monitoring techniques.

A. The first technique is an extension to previously proposed M measure. This technique is referred to as M-delta measure. The M-delta measure takes into account the probability that distant frames have different phoneme labels, providing a more accurate indicator of the estimator's ability to distinguish between phonemes. This technique is evaluated in a multi stream based adaptation paradigm [2].

B. The second technique is based on modeling the training data using an Autoencoder. Autoencoders have been proposed as an alternative to GMMs for modeling the distribution of the data [13]. There are several advantages of using an autoencoder instead of GMMs: Autoencoders relax Gaussian assumption on the input feature space. They can also efficiently model high dimensional features, allowing modeling of long term dependencies in the input feature space. In the present work, we propose to use an autoencoder for the task of performance estimation technique.

2. IMPROVING ROBUSTNESS OF ASR SYSTEM USING M-DELTA MEASURE

An extension of the M-measure, which is denoted "M-delta measure," computes the probability in each time span of two frames being an instance of the same phoneme. At test time, it estimates the M-measures for same vs. different phonemes by solving a redundant set of linear equations.

The original M-measure assumes that the distance between probability estimates in several time-spans should be large for known data (mainly for clean speech). However, this is not always accurate. If two posteriors are of the same phoneme class, the distance between them should be small, irrespective of time intervals. This means that the original M-measure ignores the effect of posterior pairs that are separated by large time intervals but belong to the same phoneme class. It accumulates symmetric KL divergence between posteriors without considering this kind of phoneme dependency.

We therefore introduce the idea of within-class and across-class M-measures, $\mathcal{M}^{\mathrm{wc}}$ and $\mathcal{M}^{\mathrm{ac}}$, to represent the accumulated KL-divergence computed from data pair of the same phoneme class and that from data pair of different classes, respectively. The new M-delta measure is defined using those within- and across-class M-measures as

$$\mathcal{M}delta = \mathcal{M}^{\mathrm{ac}} - \mathcal{M}^{\mathrm{wc}}. \tag{2}$$

Specifically, it is assumed that the M-measure can be decomposed into

$$\mathcal{M}(\Delta t) = p^{\mathrm{wc}}(\Delta t) \cdot \mathcal{M}^{\mathrm{wc}} + p^{\mathrm{ac}}(\Delta t) \cdot \mathcal{M}^{\mathrm{ac}} + \epsilon_{\Delta t}, \tag{3}$$

where $\mathcal{M}(\Delta t)$ denotes the original M-measure, which is obtained for each utterance; $p^{\mathrm{wc}}(\Delta t)$ and $p^{\mathrm{ac}}(\Delta t)$ denote the prior probability of a pair of frames separated by Δt being instances of the same and different phonemes, respectively; and $\mathcal{M}^{\mathrm{wc}}$ and $\mathcal{M}^{\mathrm{ac}}$ denote the within-class and across-class M-measures being estimated for each utterance. The probabilities of frames belonging to the same vs. different phoneme classes at each time interval are obtained using the exact transcriptions of the training data.

The error term $\epsilon_{\Delta t}$ is included because Eq. (3) is an approximate representation of the M-measure. Although the prior probabilities computed from training data are reliably estimated, these probabilities vary across test utterances, because the variety of phonemes in a test utterance is limited. To minimize the overall error of within-class and across-class M-measures, the redefined

M-measure described in Eq. (3) can be written redundantly with several values of Δt. Assume that $\mathbf{y}$, $\mathbf{A}$, $\mathbf{x}$, and ϵ are given as follows:

$$\mathbf{y} = \left[\begin{array}{ccc} \mathcal{M}(\Delta t_1) & \cdots & \mathcal{M}(\Delta t_N) \end{array} \right]^{\mathrm{T}} \in \mathbb{R}^N \tag{4}$$

$$\mathbf{A} = \left[\begin{array}{cc} p^{\mathrm{wc}}(\Delta t_1) & p^{\mathrm{ac}}(\Delta t_1) \\ \cdots & \cdots \\ p^{\mathrm{wc}}(\Delta t_N) & p^{\mathrm{ac}}(\Delta t_N) \end{array} \right] \in \mathbb{R}^{N \times 2} \tag{5}$$

$$\mathbf{x} = \left[\begin{array}{cc} \mathcal{M}^{\mathrm{wc}} & \mathcal{M}^{\mathrm{ac}} \end{array} \right]^{\mathrm{T}} \in \mathbb{R}^2 \tag{6}$$

$$\epsilon = \left[\begin{array}{ccc} \epsilon_{t_1} & \cdots & \epsilon_{t_N} \end{array} \right]^{\mathrm{T}} \in \mathbb{R}^N \tag{7}$$

Then, Eq. (3) can be written as

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \epsilon. \tag{8}$$

In this case, the within-class and across-class M-measures can be estimated as a least square solution as

$$\mathbf{x} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{y}. \tag{9}$$

The experiments below used values of $(\Delta t_1, \Delta t_2, \cdots, \Delta t_N)$ = (1, 2, 3, 4, 5, 10, 15, 20, $\cdots$, 75, 80) and $N = 20$, which were determined from preliminary experiments.

2.1. Multistream ASR System

In order to evaluate the effective of proposed M-delta measure, it is used in multistream-based adaptation paradigm used was introduced in [20]. The full frequency of the speech signal is divided into five band-limited streams, each of which covers about three barks along auditory frequency. Then, the processing streams are formed for all non-empty combinations of five band-limited streams, yielding 31 processing streams. The most reliable processing stream was selected using performance monitors and the posterior probabilities from the ANN for that stream were used for obtaining final recognition results. This adaptation paradigm can yield advantages in band-limited noise corruption by utilizing a stream that does not contain the corrupted band.

In each band-limited stream, temporal modulation information was extracted from 250 ms temporal envelopes using frequency domain linear prediction (FDLP) analysis [17]. An ANN-based probability estimator was trained for each band-limited stream with inputs as the corresponding FDLP features and triphone states as targets. The ANNs have four hidden layers of 1024 units, input layer of 576 nodes, and 1951 output units. This band-limited ANNs were used to yield 39-dimensional phoneme posterior probabilities. In the latter stage, ANN-based probability estimators were developed for 31 processing streams. The features were obtained by stacking the phoneme posterior probabilities from the band-limited ANNs.

In the present experiments, these measures were computed based on single sentence to predict accuracy for that sentence.

2.2. Speech Materials

All models were trained on 3696 clean speech utterances from TIMIT training data set and the evaluation was conducted using 400 speech utterances from the TIMIT development set under

Table 1. *Types and SNRs of noise used.*

item	noise type	SNR
clean		
sub15	subway	15
bab15	babble	15
fac10	factory	10
res10	restaurant	10
exh5	exhibition hall	5
str5	street	5
car5	car	5
exh0_b2	exhibition hall (band 2 corrupted)	0
exh0_b4	exhibition hall (band 4 corrupted)	0

several types of noise. The types and SNRs of noise are listed in Table 1. Note that in principle, the multistream-based adaptation paradigm enables an ASR system to be more robust against stream-specific noise, such as the exh0_b2 and exh0_b4 noises listed in Table 1.

2.3. Experimental Results

Figure 1 shows the correlation between the confidence measure and actual phoneme accuracy for several types of noise. This figure shows that the M-delta measure yielded significant improvement over the existing measures, such as the negative entropy and original M-measure, in the narrow-band noise conditions, i.e., exh0_b2 and exh0_b4, while it gave similar performance to the original M-measure and did not yield an advantage over the entropy under the broad-band noise corruptions.

3. ASR ERROR RATE PREDICTION USING AUTOENCODER

In this section, we propose another performance monitoring technique. The technique is based on Autoencoder. Autoencoder has been proposed as an alternative to GMM for modeling distribution of the data [13]. Autoencoder is a multi-layered feed forward neural network, trained to reconstruct the input at the output [13]. Architecture of autoencoders used in present work consist of 3 hidden nonlinear layers, and linear input and output layers, as shown in Fig. 2. The first and third hidden layers consist of 512 neurons, and the second hidden layer is a compression layer consisting of 24 neurons. Neurons corresponding to hidden layers have tanh nonlinearity.

In order to avoid the trivial solution of an identity mapping, the second hidden layer contains fewer nodes than the input and the output layer. During the training process, the parameters of the network are adjusted to minimize the average squared error cost between input feature vector $\mathbf{x}$ and output vector $\hat{\mathbf{x}}$ as shown in (1). The parameters of the network ($\mathcal{W}$) are learned using mini-batch stochastic gradient descent algorithm.

$$\min_{\mathcal{W}} \mathbf{E}||\mathbf{x} - \hat{\mathbf{x}}||_2 \tag{10}$$

Since the network is trained to minimize the reconstruction error, average reconstruction error of a vector, sampled from distribution of the training data will be small compared to a vector drawn

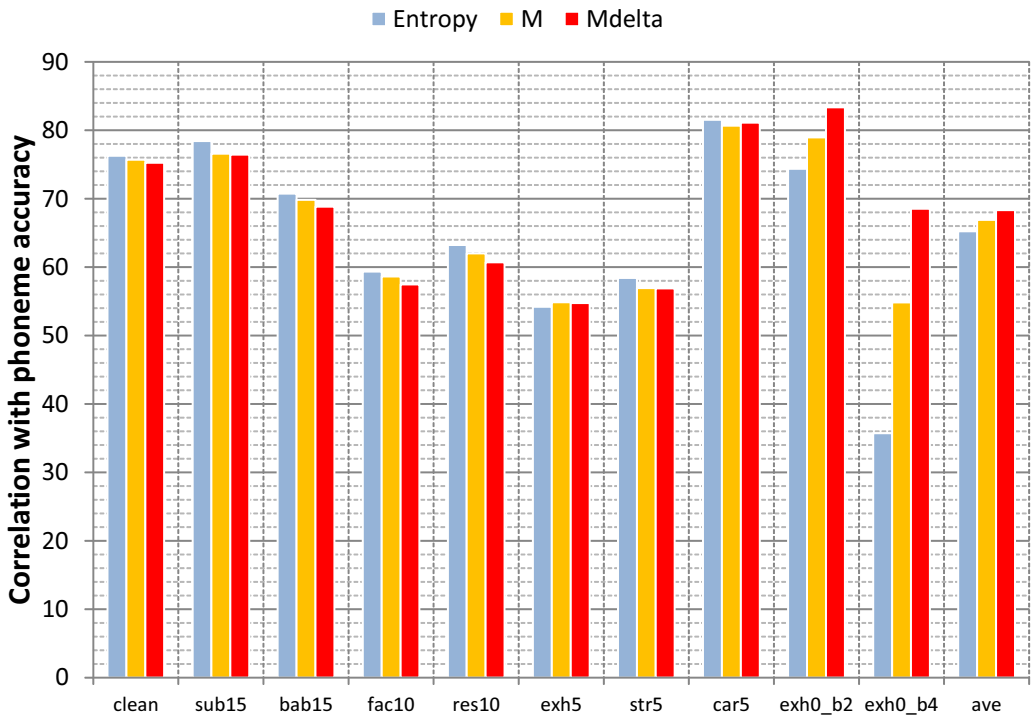

Fig. 1. Correlations with phoneme accuracy in multistream-based adaptation for several types of noise. Bars for "ave" express correlations averaged over ten conditions.

from a different distribution. This property is illustrated in figure 3, which shows distribution of l_2 norm of error vectors, computed from training data, data similar to training data, and data that deviates from the training data. Figure 3 illustrates that reconstruction error is a good indicator of the mismatch between training data and test data. Statistics derived from reconstruction error of autoencoder are used for predicting the accuracy of DNN classifier.

3.1. Speech material

In this work, we evaluate the proposed performance estimators in a large vocabulary continuous speech recognition (LVCSR) task. This section provides details of the speech database and recognition system used for the experiments.

We used Aurora4 [14] speech recognition task, which provides the required data mis-match between train and test conditions. Aurora4 database is based on the DARPA Wall Street Journal (WSJ0) corpus which consist of clean recordings of read speech. The training set consists of 14 hours of clean speech, from 83 speakers. The test set contains simultaneous recording in 14 different noise and channel mis-match conditions. There are six different noise types ("Street", "Babble", "Train", "Car", "Restaurant", "Airport") with varying signal to noise ratio levels of 5-15 dB and different microphone types. Each test condition contains 330 recordings with a total of 40 minutes of speech.

The LVCSR system used for multi-stream experiments is a hidden Markov model-deep neural network (HMM-DNN) system. The system is implemented using Kaldi speech recognition toolkit [15]. The alignments used to train DNNs are generated using a hidden Markov model-Gaussian

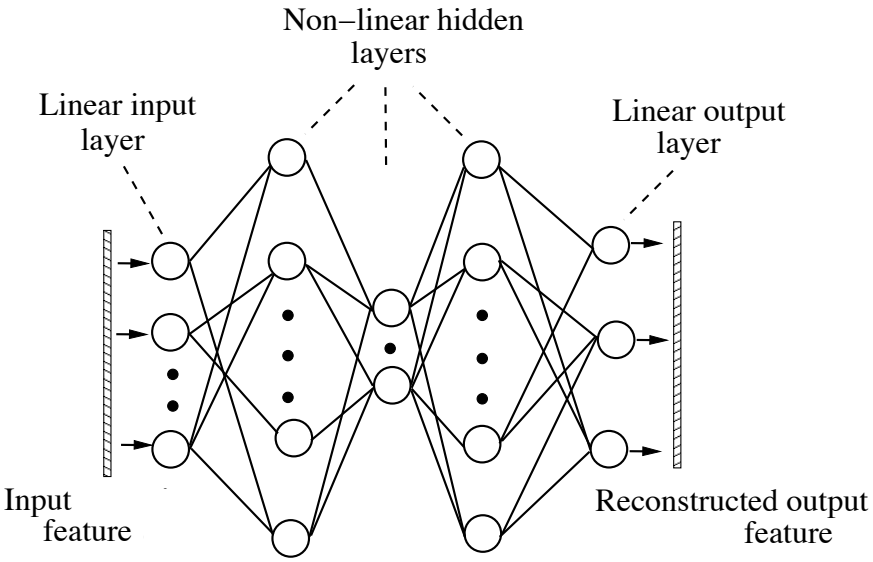

Fig. 2. A five layered autoencoder, with 3 non-linear hidden and 2 linear visible layers. Architecture of autoencoder used in this work is $\{Y \times 512 \times 24 \times 512 \times Y\}$, where Y corresponds to input feature dimension.

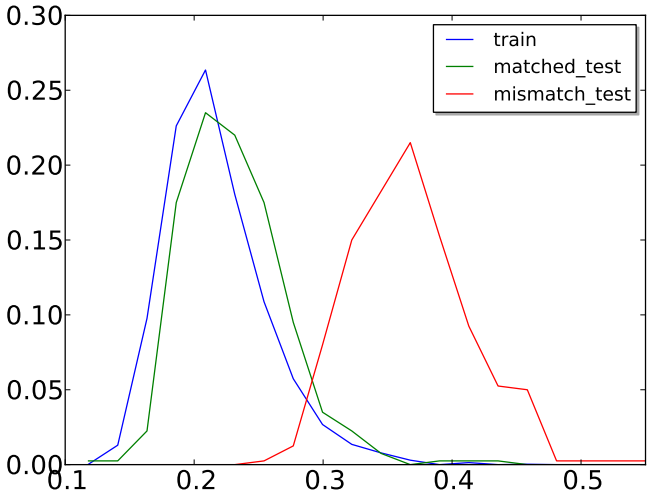

Fig. 3. Illustration of property of autoencoder useful to distinguish matched data and mismatched data. Figure contains 3 distributions of l_2 norm of error vectors, computed from training data, matched test data and mismatched test data.

mixture model (HMM-GMM) system trained using Mel frequency cepstral coefficients (MFCC). The HMM-GMM system is then used to generate context dependent triphone state level alignments, for each frame of the acoustic data. The state-level alignments are then used to train a DNN using auditory filter-bank features [16], using a context window of 20 frames. The DNN consists of 4 hidden layers. Each hidden layer consist of 1024 sigmoid neurons. It was initialized by a layer wise restricted Boltzmann machine pre-training [19]. Word error rate values of each utterance in the Aurora4 test set are computed by hybrid decoding of the DNN. The measures proposed in this work are for estimation of the per utterance word error rates.

3.2. Autoencoder Experiments

In this section, we present experiments performed to obtain suitable measures for predicting performance a classifier.

3.2.1. Input feature representation

Similar to DNNs, autoencoders can also be trained on different types of feature representations. In order to identify the best feature representation for the task of performance prediction, we experimented with the 3 feature representations used in speech recognition.

(a) Auditory filter-bank features: Auditory filter-bank features are extracted from speech signal by time-frequency analysis using 128 highly-asymmetric and overlapping constant-Q bandpass filters. This is followed by a lateral inhibitory network. The final stage involves envelope extraction, lowpass filtering of envelope and down sampling of spectrum resulting in 32 dimensional auditory spectrogram. More details about auditory filter-bank features can be found in [16].

Bottleneck features: Bottleneck features is more uncorrelated compared to filter-bank features [18]. Even though filter-bank features are more suitable for training neural networks, the squared error measure obtained from autoencoder is more suitable for uncorrelated features. Auditory filter-bank features are used to train a 6 layer bottleneck neural network with triphone state alignments. The bottleneck layer consist of 32 linear neurons.

Log-Posterior features: Since most of the previously proposed performance estimation techniques operate on the phoneme posterior probabilities, we experimented with Log-Posterior feature representation. Log-Posterior features are computed as follows: Phoneme posterior probabilities are computed by merging the tied-state posteriors corresponding to the phoneme. The phoneme posterior probability vectors are then transformed using logarithm, resulting Log-Posterior features.

3.2.2. Measures based on reconstruction error

In this section, we present details of the measures proposed for performance prediction. The proposed measures are based on the reconstruction error of the autoencoder. The reconstruction error at frame t is denoted as e_t.

Average l_2 norm ($\overline{||\mathbf{e}||}_2$):
Average l_2 norm is defined as frame-average of l_2 norm of reconstruction error vectors. For a given test utterance, auditory filter-bank energies are extracted. Log-Posterior features are computed by forward passing the auditory filter-bank energies through the DNN. Reconstruction error vectors

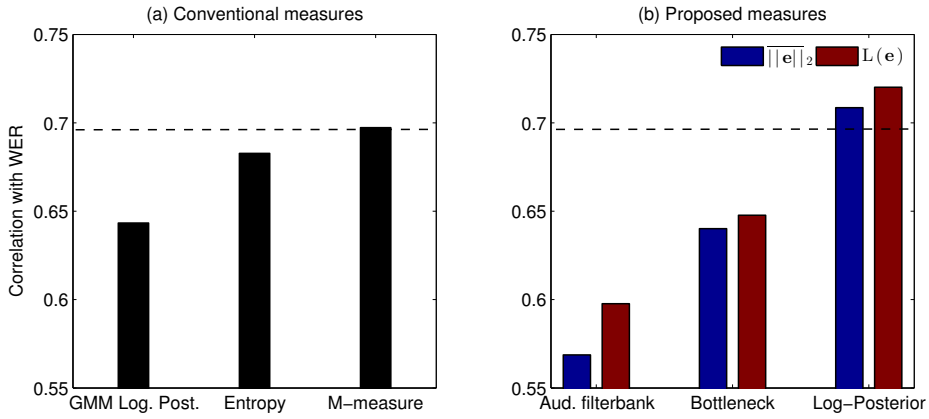

Fig. 4. Comparison of baseline performance monitoring methods with proposed performance monitoring methods. (a) shows correlation of the baseline measures. (b) shows the correlation of proposed measures. The 3 clusters in (b) correspond to measures computed from autoencoders trained using the 3 different feature representations. Dashed line $(--)$ show the correlation value of the best baseline measure.

are computed for each frame of the test utterance and average l_2 norm of error vector is computed.If test utterance is similar to training data, average l_2 norm of error vectors is expected to be low.

Log-likelihood of error vectors ($\mathrm{L(e)}$)**:**
In this measure, we model the reconstruction error vectors using a multivariate Gaussian distribution, with diagonal covariance matrix, and use the log-likelihood as a measure to predict the performance. The error vectors of the training data are used to estimate parameters of the reference model, $\mu_{\mathbf{ref}}$ and $\sum_{\mathbf{ref}}$. Log-likelihood of error vectors corresponding to a test utterance is computed as follows:

$$\mathrm{L}(\mathbf{e}) = \frac{1}{T}\sum_{t=1}^{T} \log\mathcal{N}(e_t|\mu_{\mathbf{ref}},\sum_{\mathbf{ref}}) \tag{11}$$

where T denotes the number of frames in the test utterance. Low log-likelihood for a test case indicates that the test utterance is not similar to the training set, which indicates a higher WER.

3.3. Results

The performance estimation techniques described in this work are used to predict the word error rate. The baseline measures used for comparing effectiveness of proposed measures are Gaussian modeling of Log-Posterior features [12], average negative entropy of phoneme posteriors [7], and M-measure, which is computed by accumulating divergences between phoneme probability estimates spaced in several time-spans [11].

The measures are computed for each utterance in Aurora4 test set. Aurora4 test set consist of 4620 utterance from 14 different acoustic conditions, comprising a total of 9.3 hours of speech. Pearson correlation was computed for these 4620 utterances.

Figure 4 (a) shows the correlation values of baseline measures. It is evident from the figure that M-measure has the best correlation value (0.6973), compared to baseline measures. This observation is similar to the one in previous study [11]. Figure 4 (b) shows the correlation values

of the proposed measures. It can be seen that the autoencoder trained on Log-Posterior features perform significantly better than M-measure, as the auto encoder's correlation value is 0.7201. Also, the type of input representation for the autoencoder is crucial, as the correlation ranges from 0.5977 using auditory filter-bank features to 0.7201 using Log-Posterior features. It is also evident from the figure, that compared to GMM modeling of Log-Posteriors (GMM Log-Post. in Fig. 4), modeling the Log-Posteriors using autoencoder is giving significantly better results (correlation is increasing from 0.64 to 0.70).

4. CONCLUSIONS AND FUTURE DIRECTIONS

In this work we have proposed two new performance monitoring techniques. The first measure, M-delta is an improved version of the previous proposed M-measure. The within-class and across-class M-measures were introduced to consider phoneme class information that was ignored in the original M-measure and obtained by solving redundant set of equations. This extension (M-delta measure) yielded significant gains from the original M-measure, especially under the narrow-band noise. By reducing the influence of phoneme contexts on the confidence measures, using broad phoneme class probabilities instead of standard phoneme probabilities in the performance predictor yielded further improvement.

The second proposed technique is based on Autoencoder. Reconstruction error of a well trained Autoencoder was shown to correlate well with word error rate. Autoencoder trained using Log-Posterior features is shown to have more correlation with word error rate, compared to other feature representations. Two measures based on autoencoder error vectors were proposed. Both the measures are shown to correlate well with WER. The correlation values of the proposed measures are better than previously proposed baseline measures.

4.1. Future directiosn

The criterion used for Autoencoder training used in the present work is stochastic gradient descent algorithm. In recent studies [22], layer-wise pretraining is shown to improve the generalization capability of the network. These methods can be employed to further improve the performance estimation capability of Autoencoder. Phoneme class information can be included into this measure, by using mixture of Autoencoders can approach [23]. We also propose to Autoencoder based measures in multi-stream speech recognition.

In this work, we described various performance monitoring techniques. Combination of these techniques is a natural direction to pursue. Currently, the performance monitor methods use statistic, which is specific to each method. For example, in the case of entropy based methods, inverse entropy or negative entropy to predict the performance. In the case of M-measure, height of M curve is used to predict the performance. These methods can be transformed into more probabilistic frame work, where a model is built on prediction measure computed for each sentence (or frame). This probabilistic modeling of measures has the advantage that combination multiple measure can be achieved using hypothesis testing framework. In [24], a combination method for hypothesis testing problem is proposed. The combination method is based p-values of individual test statistics. We propose to use this approach to fuse multiple performance monitor methods. Effective fusion of streams

5. REFERENCES

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A.Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," in IEEE Signal Processing Magazine, 29, November 2012.

[2] H. Hermansky, "Multistream recognition of speech: dealing with unknown unknowns," Proc. IEEE, vol. 101, no. 5, pp. 1076-1088, 2013.

[3] P. Duchnowski, "A new structure for automatic speech recognition, Ph.D. dissertation, Dept. Electr. Comput. Eng., Massachusetts Inst. Technol., Cambridge, MA, 1992.

[4] J. B. Allen, Personal Communication, private communication, 1993, DoD Summer Workshop at Rutgers University.

[5] H. Bourlard, S. Dupont, H. Hermansky, and N. Morgan, "Towards subband-based speech recognition," in Proc. EUSIPCO, 1996, pp. 15791582.

[6] Tibrewala, Sangita, and Hynek Hermansky. "Sub-band based recognition of noisy speech." Acoustics, Speech, and Signal Processing, IEEE International Conference on. Vol. 2. IEEE Computer Society, 1997.

[7] Misra, Hemant, Shajith Ikbal, Herv Bourlard, and Hynek Hermansky. "Spectral entropy based feature for robust ASR." In Acoustics, Speech, and Signal Processing,2004. Proceedings.(ICASSP'04). IEEE International Conference on, vol.1, pp. I-193. IEEE, 2004.

[8] Okawa, Shigeki, Enrico Bocchieri, and Alexandros Potamianos. "Multi-band speech recognition in noisy environments." Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on. Vol. 2. IEEE, 1998.

[9] Mesgarani, Nima, Samuel Thomas, and Hynek Hermansky. "Adaptive Stream Fusion in Multistream Recognition of Speech." INTERSPEECH. 2011.

[10] Variani, Ehsan, and Hynek Hermansky. "Estimating Classifier Performance in Unknown Noise." INTERSPEECH. 2012.

[11] Hermansky, Hynek, Ehsan Variani, and Vijayaditya Peddinti. "Mean temporal distance: Predicting ASR error from temporal properties of speech signal".Acoustics, Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013.

[12] Tetsuji Ogawa, Feipeng Li, Hynek Hermansky "Stream selection and integration in multistream ASR using GMM-based performance monitoring". INTERSPEECH 2013

[13] B. Yegnarayana and S. Kishore, "AANN: an alternative to GMM for pattern recognition", Neural Networks, pp.459469, 2002.

[14] Parihar, N. and Picone, J., "Aurora working group: DSR front end LVCSR Evaluation, Technical Report, 2002.

[15] D Povey, A Ghoshal, G Boulianne, L Burget, O Glembek, N Goel, M Hannemann, P Motlcek, Y Qian, P Schwarz, J Silovsky , G Stemmer, and K Vesely, "The Kaldi speech recognition toolkit, in Proc. IEEE ASRU, December 2011.

[16] Sridhar Krishna Nemala, "Robust speech processing by humans and machines: the role of spectro-temporal modulations", PhD Thesis, 2012.

[17] S. Ganapathy, S. Thomas, H. Hermansky, "Temporal envelope compensation for robust phoneme recognition using modulation spectrum," J. Acoust. Soc. Amer., vol 128, 2010.

[18] K Vesely, M Karafit, F Grezl, "Convolutive bottleneck network features for LVCSR", IEEE-ASRU, 2011.

[19] Hinton, G. E. and Salakhutdinov, R. R. "Reducing the dimensionality of data with neural networks", Science, Vol. 313. no. 5786, pp. 504 - 507, 28 July 2006.

[20] F. Li "Subband hybrid feature fro multi-stream speech recognition", in Proc. ICASSP, 2014.

[21] F. Li, H. Mallidi, H. Hermansky, "Phone recognition in critical bands using sub-band temporal modulations," in Proc Interspeech, 2012.

[22] F. Seide, G. Li, X. Chen, and D. Yu, Feature engineering in context-dependent deep neural networks for conversational speech transcription, in IEEE ASRU, 2011.

[23] G. S. V. S. Sivaram, S. Thomas, H. Hermansky "Mixture of Auto-Associative Neural Networks for Speaker Verification," in Proc. Interspeech, 2011.

[24] W. R van Zwet, J. Osterhoff, "On the combination of independent test statistics," Ann. Math. Stat. 1959