Effects of Frequency-Based Inter-frame Dependencies on Automatic Speech Recognition

Ludovic Trottier, Brahim Chaib-draa, and Philippe Giguère

Department of Computer Science and Software Engineering,
Université Laval, Québec (QC), G1V 0A6, Canada
trottier@damas.ift.ulaval.ca
{chaib,philippe.giguere}@ift.ulaval.ca
http://www.damas.ift.ulaval.ca

Abstract. The hidden Markov model (HMM) is a state-of-the-art model for automatic speech recognition. However, even though it already showed good results on past experiments, it is known that the state conditional independence that arises from HMM does not hold for speech recognition. One way to partly alleviate this problem is by concatenating each observation with their adjacent neighbors. In this article, we look at a novel way to perform this concatenation by taking into account the frequency of the features. This approach was evaluated on spoken connected digits data and the results show an absolute increase in classification of 4.63% on average for the best model.

Keywords: acoustic model, inter-frame dependencies, speech recognition and automatic digit recognition

1 Introduction

Automatic speech recognition (ASR) is the transcription of spoken words into text. An ASR model thus takes as input an audio signal and classifies it into a sequence of words. The most common model that is used to perform ASR is the hidden Markov model (HMM) with a Gaussian mixture model as observation density (GMM-HMM). Even though GMM-HMM already showed promising results on various tasks, it is well known that the model has strong assumptions. One of them is the state conditional independence: knowing the latent state at current time makes the current observation independent of all other observations. It has been shown that this hypothesis does not hold when a HMM is used for speech recognition [1].

There are multiple ways in which one can modify the HMM in order to reduce the detrimental effects of state conditional independence on classification performance. Models such as trajectory HMM [3], switching linear dynamical systems [4] and inter-frame HMM [5] were proposed to overcome the assumptions of standard HMM. A recent method (HMM with a deep neural network observation density) used a rectangular window to concatenate each observation with their preceding and succeeding MFCC frames [6]. It has been evaluated

that the window helped the model achieve outstanding results by reducing the effects of the state conditional independence [2]. In this paper, we investigate how the use of a window can help to increase the performance of GMM-HMM by looking at a triangular window that models inter-frame dependencies based on frequency. Signal processing theory suggests that detecting changes is longer for lower frequency signals, and faster for higher frequency signals. Thus, the frequency appears to be a good metric of the speed of variation of information. Therefore, if we create a triangular window that uses information on a longer time-range for low frequencies and a shorter one for high frequencies, we expect to improve the classification of a frame.

The remainder of this paper is structured as follows. Section 2 elaborates on the temporal window and the chosen HMM. In Section 3, the experimental framework and the results of the experiments are described, and we conclude this work in Section 4.

2 Temporal Window

It is normally required that features be extracted from utterances before any classification. Generally, time-based waveform are not as robust as frequency-based features for ASR. For this reason, the most common feature extraction method used for ASR is the Mel frequency cepstral coefficients (MFCC). It uses various transformations, such as the Fourier transform, in order to output a series of frequency-based vectors (frames). As a matter of convention, the coefficients in a frame are ordered from low to high frequency.

In this paper, we propose a novel way of combining feature frames (MFCC) extracted at different times that takes into account the velocity of change at different frequencies. This type of approach that we dubbed triangular window is motivated by signal processing theories which show that the rate at which information changes in signals is proportional to frequency. This is why our window collects frames farther apart for low frequency coefficients and closer for high ones, as depicted in Fig. 1.

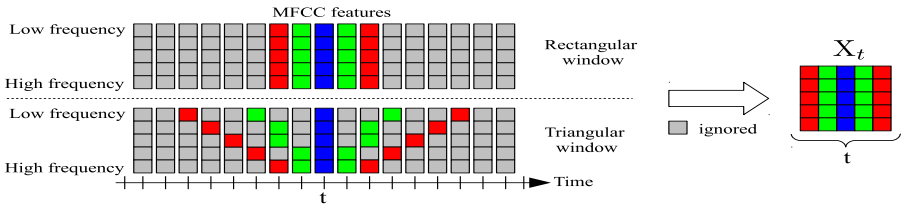

Fig. 1. Example of a rectangular and triangular window of size 5. A given column, on the leftmost figure, represents a frame at a particular time t and the rows are coefficients (MFCCs). Our triangular window approach performs coefficients sub-sampling on each row based on the frequency. The classical rectangular window approach is simply the concatenation of nearby frames together. We denote as X_t the resulting matrix after concatenation.

The difference between rectangular and triangular windows is seen in Fig. 1. The former concatenate coefficients from adjacent frames, whereas the latter concatenate coefficients from other frames based on their frequency. The coefficients at lower frequencies are taken further apart to avoid redundant information. Those at higher frequencies can be made closer because the frames quickly become independent. Importantly, coefficients should not be duplicated in order to avoid redundant information.

In models such as GMM-HMM, where each mixture component is a Gaussian, the resulting matrix X_t can be vectorized to get back a vector $\mathbf{x}_t$. However, $\mathbf{x}_t$ will be high-dimensional if the size of X_t is large. Thus, the evaluation of the observation density would be expensive to compute since it requires the inversion of a high-dimensional covariance matrix Σ. In addition, Σ is more likely to be non-invertible and the learning process could also over-fit it.

For these reasons, the observation density was set to the matrix equivalent of GMM: the matrix normal mixture model (MNMM). In this case, the density function of each component in a MNMM is given by the matrix normal density function:

$$p(X|M,U,V) = \frac{\exp\left(-\frac{1}{2}\operatorname{Tr}\left[V^{-1}(X-M)^\top U^{-1}(X-M)\right]\right)}{(2\pi)^{\frac{np}{2}}|V|^{\frac{n}{2}}|U|^{\frac{p}{2}}}\ , \tag{1}$$

where X and M are $n \times p$ dimensional matrices, U is $n \times n$ and V is $p \times p$. Both models are equivalent if $\Sigma = \text{kron}(V,U)$, the Kronecker product between V and U, and if $\text{vec}(M) = \mu$, the mean of the Gaussian. Most of the time, these models achieve similar results even though MNMM is approximating Σ. However, notice that the $np \times np$ matrix Σ is divided into two smaller matrices U and V which avoids the problems cited earlier.

The Expectation-Maximization (EM) for learning MNMM, as proposed in [7], can be used in our approach to learn MNMM-HMM. To achieve that, we simply replace the posterior probability that observation j belongs to component i ($\tau_{i,j}$) by the posterior probability that frame t belongs to component k of latent state j ($\gamma_t(j,k)$). $\tau_{i,j}$ can be found using component weights whereas $\gamma_t(j,k)$ can be computed using the well-known Forward-Backward recursion. Consequently, the equation for learning the mean of the mixture component k in state j is:

$$\hat{M}_{j,k} = \sum_{t=1}^{T} \gamma_t(j,k) X_t \Bigg/ \sum_{t=1}^{T} \gamma_t(j,k)\ . \tag{2}$$

The equations for the among-row $U_{j,k}$ and among-column $V_{j,k}$ covariance can also be written using the γ smoothing value:

$$\hat{U}_{j,k} = \sum_{t=1}^{T} \gamma_t(j,k) \left(X_t - \hat{M}_{j,k}\right) V_{j,k}^{-1} \left(X_t - \hat{M}_{j,k}\right)^\top \Bigg/ p \sum_{t=1}^{T} \gamma_t(j,k)\ , \tag{3}$$

$$\hat{V}_{j,k} = \sum_{t=1}^{T} \gamma_t(j,k) \left(X_t - \hat{M}_{j,k}\right)^\top \hat{U}_{j,k}^{-1} \left(X_t - \hat{M}_{j,k}\right) \Bigg/ n \sum_{t=1}^{T} \gamma_t(j,k)\ . \tag{4}$$

The update equations for the mixture weights and the state transitions stay the same. Having now in hands a suitable model that can be used efficiently with the new features X_t, we have elaborated a number of experiments to test the validity of our triangular window.

3 Connected Digits Recognition

The database we used for our experiments is Aurora 2 [8] which contains 11 spoken digits (*zero* to *nine* with *oh*). The digits are connected, thus they can be spoken in any order and in any amount (up to 7) with possible pauses between them. The utterances are noisy and the signal-to-noise ratio (SNR) varies from -5 dB, 0 dB, ..., 20 dB. Different kinds of noise have corrupted the signals such as train, airport, car, restaurant, etc. The training set contains 16,880 utterances, test set A and B have 28,028 and test C has 14,014. On average, an utterance last approximately 2 seconds.

We defined 25 different triangular windows by hands and tested their accuracy with MNMM-HMM. We compared our results with the state-of-the-art GMM-HMM. The latent structure of both HMMs was fixed to 7 states, left-to-right transition only, with additional non-emitting entry and exit. We chose 7 states as the maximum number of states that prevents frames deprivation (more states than frames in HMM). The number of mixture components per state for the emission model was fixed to 4. We used standard 39 dimensional MFCC + Delta + DoubleDelta for GMM-HMM and 13 dimensional MFCC for MNMM-HMM. Both models were trained using EM and the number of iterations it ran was fixed to 10 (we saw a convergence of the log-likelihood after 10 iterations).

Phoneme-based training and testing was performed and 20 different phonemes (including silence) were used to represent the 11 digits. The phonemes were chosen according to The CMU Pronouncing Dictionary[1]. The algorithms were initialized using a standard uniform phoneme segmentation and a silence phoneme was concatenated to the beginning and the end of each utterance. The language model was a uniform bi-gram.

Fig. 2 shows the word accuracy of MNMM-HMM with the best triangular window compared to GMM-HMM. It used coefficients at time $[t-\boldsymbol{\alpha}_i, t-\boldsymbol{\delta}_i, t, t+\boldsymbol{\delta}_i, t+\boldsymbol{\alpha}_i]$ where $\boldsymbol{\delta} = [1,1,2,2,\ldots,6,6,7]$ and $\boldsymbol{\alpha} = [2,3,4,5,\ldots,12,13,14]$ (i iterates from high to low frequency). In addition, the word accuracy computes the number of insertions, deletions and substitutions that are necessary to transform the recognized utterance into the reference. We also added the classification results from the original paper that presented the Aurora 2 database as an additional comparison for our method [8]. We refer to it as HTK since they used the Hidden Markov Model Toolkit to perform the recognition [9]. We can see from the results in Fig. 2 that the triangular window increases the performance of the recognition of about 4.63% on average compared to GMM-HMM.

We notice that the model did not outperform HTK. Indeed, HTK uses a more complex training algorithm that includes, among others, short pause and silence

[1] `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`

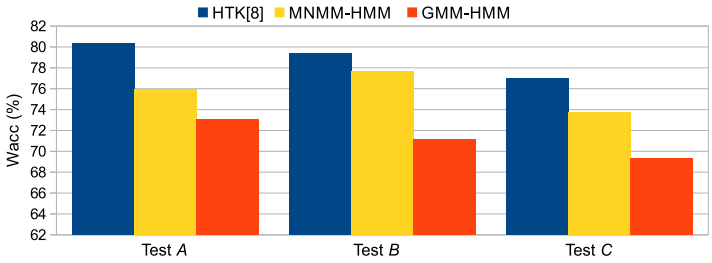

Fig. 2. Word accuracy on Aurora 2 database. For MNMM-HMM, multiple triangular windows were defined and the best one is shown (as explained in the text).

inference, cost function on word transitions and pruning. Instead, a standard EM algorithm was applied on both GMM-HMM and MNMM-HMM. In our preliminary work, we wanted to focus on whether the triangular window could help, or not, the classification. Moreover, the performance increase achieved using a window does not depend on these add-ons. Therefore, the triangular window should still increase the classification results once all the add-ons are implemented.

Moreover, Fig. 3 shows the performance of MNMM-HMM with the best triangular window (depicted earlier) along with rectangular windows of different sizes. We can see that the triangular window helped achieve better accuracy. Also, there is a slight degradation of performance as we increase the size of the rectangular window. This is due to the fact that the coefficients concatenated on both sides are either non informative, redundant or related to other phonemes. This shows that using a triangular window is better than its rectangular counterpart.

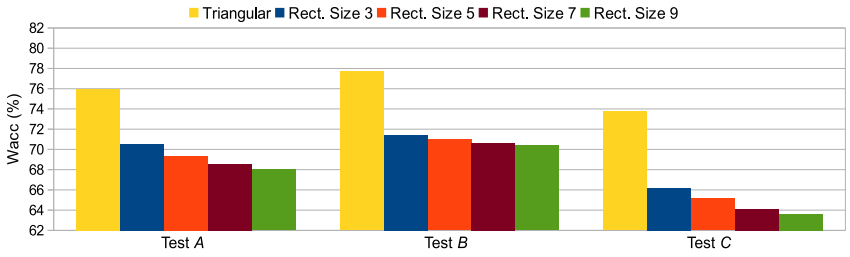

Fig. 3. Comparison for MNMM-HMM of rectangular and triangular windows. The sizes refer to the horizontal size of X_t using a rectangular window. The triangular window is the one that achieved the best performance reported in Fig. 2.

4 Conclusion

In this work, the novel concept of triangular window is investigated. The goal of the triangular window is to take into account the fact that changes in lower-frequency signals cannot be detected rapidly. Therefore, by modifying the concatenation strategy of features depending on their frequency, the model managed to incorporate more useful information while avoiding adding confusing information. In order to avoid over-fitting and matrix inversion problems, we selected MNMM which is the matrix equivalent of GMM.

The performance was evaluated on the spoken connected digits database Aurora 2. The reference models were the GMM-HMM along with HTK. Even though MNMM-HMM did not outperformed HTK, due to some training procedures that were not implemented, it increased the classification results of GMM-HMM by about 4.63% on average. Finally, the triangular window was compared to the rectangular window that inspired this work. It was observed that MNMM-HMM had better results when incorporating frequency dependencies to the concatenation of adjacent frames.

We plan to apply our approach to other existing models such as HTK and deep belief network HMM. For DBN-HMM, the triangular window could reduce the number of neurons on the input level to allow deeper networks to be trained.

References

1. Rabiner, L., Juang, B.H.: Fundamentals of Speech Recognition. Prentice Hall, United States edition (1993)
2. Pan, J., Liu, C., Wang, Z., Hu, Y., Jiang, H.: Investigation of deep neural networks (DNN) for large vocabulary continuous speech recognition: Why DNN surpasses GMMS in acoustic modeling. In: ISCSLP, pp. 301–305. IEEE (2012)
3. Zen, H., Tokuda, K., Kitamura, T.: Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences. Computer Speech & Language, 21.1, 153–173 (2007)
4. Mesot, B, Barber, D.: Switching linear dynamical systems for noise robust speech recognition. Audio, Speech, and Language Processing, IEEE Transactions on 15.6, 1850–1858 (2007)
5. Hanna, P., Ming, J., Smith, F.J.: Inter-frame dependence arising from preceding and succeeding frames - Application to speech recognition. Speech Communication, 28.4, 301–312 (1999)
6. Hinton, G., al.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. Signal Processing Magazine, IEEE, 29.6, 82–97 (2012)
7. Viroli, C.: Finite mixtures of matrix normal distributions for classifying three-way data. Statistics and Computing, 21.4, 511–522 (2011)
8. Pearce, D., Hirsch, H., Ericsson Eurolab Deutschland Gmbh: The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition Systems under Noisy Conditions. In: ISCA ITRW ASR2000, pp. 29–32. (2000)
9. Young, S. J., Evermann, G., Gales, M. J. F., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P. C.: The HTK Book, version 3.4. Cambridge University Engineering Department, Cambridge, UK (2006)