POLYPHASE SPEECH RECOGNITION

Hui Lin, Jeff Bilmes

{hlin,bilmes}@ee.washington.edu
Department of Electrical Engineering
University of Washington, Seattle
Seattle, WA 98195-2500

ABSTRACT

We propose a model for speech recognition that consists of multiple semi-synchronized recognizers operating on a polyphase decomposition of standard speech features. Specifically, we consider multiple out-of-phase downsampled speech features as separate streams which are modeled separately at the lowest level, and are then integrated at the higher level (words) during first-pass decoding. Our model lessens the severity of the oversampling problem in many speech recognition systems – i.e., that speech modulation energy is most important below 25Hz but a 100Hz frame rate gives a modulation bandwidth of 50Hz. Our polyphase approach moreover captures wider and more diverse dynamics within the speech signal. Our integrative network is high-level, namely it couples together and decodes word strings from different recognizers simultaneously and asynchronously. We provide preliminary results on the 10-word vocabulary version of the SVitchboard (small-vocabulary switchboard) task and show that our polyphase recognition system significantly outperforms an optimized baseline (HMM) approach.

Index Terms— polyphase speech recognition, dynamic Bayesian network

1. INTRODUCTION

Perceptual experiments [1, 2] have shown that the modulation frequency band between 1 and 16Hz is where the most important information lies for speech intelligibility. This fact is further confirmed by automatic speech recognition (ASR) experiments [3] which shows that the low modulation frequency bands (0-1Hz) and high modulation frequency bands (16-50Hz) are harmful (or useless) for ASR. The most widely used cepstral-based features in state-of-art ASR system, however, are typically sampled at a rate of 100Hz, giving a 50Hz bandwidth for representing modulation energy, something that is overkill. In other words, most speech features oversample in the modulation domain.

Acoustic frame oversampling can have several deleterious effects. First, the effect of the acoustics can dominate the rest of the model (pronunciation and length scores, language model scores, etc.). This can be corrected to some extent by language and acoustic model scaling factors to counter balance the dominance of the acoustics. The extent to which these factors can fully correct for this imbalance is not fully known, however — an alternative approach, as presented here, would lessen the imbalance in the first place. Second, by representing modulation energy in the (apparently less informative) bands between 25Hz and 50Hz, there is a danger that the model may become sensitive to aspects of the signal that do not have a strong importance for the underlying words contained in an utterance – a perhaps better approach would be to concentrate the models representational power on those most informative modulation bands. Third, oversampling might lead to speech recognition systems that are more computationally expensive than necessary.

To address this problem, we introduce a novel polyphase speech recognition model. The model operates on multiple streams of speech features, each of which derived from a downsampling of phase-shifted original speech features. These multiple streams are then modeled separately at the lowest level, and are not forced to remain fully synchronized with each other. A higher level combination strategy is used to integrate the word hypotheses of the various polyphase recognizers together to arrive at a final hypotheses. We moreover perform this representation entirely within one statistical system, so that a single first-pass decoding procedure is used to produce final word hypotheses.

Our approach is indeed a form of classifier combination for speech recognition. Indeed, many ASR approaches have been proposed in the past for the combination of multiple recognizer outputs, where each base recognizer extracts some unique characteristic of the speech signal. These aspects of speech might be represented at the acoustic level (multiple observation streams), the hidden level (multiple hidden Markov chains), or both. One popular combination method is the multi-stream approach, where the speech signal has been divided into multiple, possibly semi-independent, streams of partially coupled information. For instance, in the multi-band approach [4], the speech signal is divided spectrally, and where each speech stream represents a different spectral sub-band on which an independent recognizer is applied. The different recognizers are the combined at a later stage. In [5], heterogeneous acoustic measurements are proposed to increase the amount of acoustic-phonetic information extracted from the speech signal, and phone classifiers utilizing these heterogeneous measurements are combined through hierarchical and committee-based techniques. At the hidden layer, articulatory-based approaches to speech recognition are becoming more popular [6], where the speech signal is represented by multiple semi-synchronous streams of articulatory gestures. Early multi-stream work also includes that of HMM decomposition [7], where both speech and noise are consider a separate stream. Dynamic Bayesian networks (DBNs) have also been used for multi-stream [8], including audio-visual speech recognition [9, 10, 11]. Even HTK has the ability to represent multiple synchronous acoustic streams.

Our approach is novel, in that our streams are in fact derived from a polyphase decomposition of some original feature stream. For example, one instance of our approach divides the even numbered and odd numbered frames into separate streams. Alone, this method might not be useful, but we use the framework of DBNs [12, 13] to represent a partially-synchronous integrative procedure

This work was supported by an ONR MURI grant, No. N000140510388.

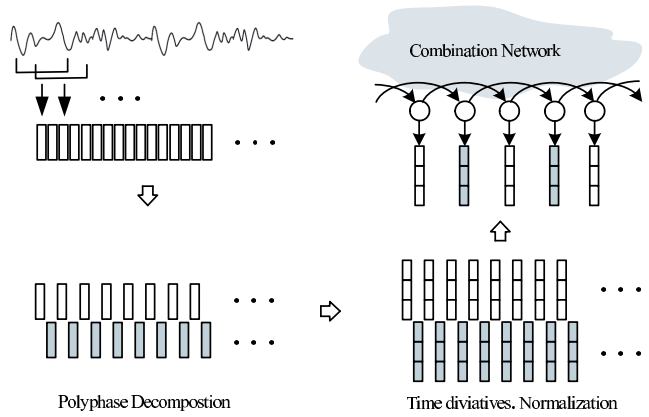

Fig. 1. Diagram of polyphase speech recognition.

between these polyphase streams. In particular, the streams are combined at the word level, but the word hypotheses are allowed to desynchronize from one another to a certain extent. Moreover, our decoder jointly decodes from all streams simultaneously, rather than having a first pass for each individual downsampled recognizer. The benefits of such joint decoding is that it has the ability to discover desynchronized hypotheses that look good in tandem. Moreover, our approach allows us to simultaneously train each sub-stream recognizer with respect to each other.

2. POLYPHASE FEATURES

2.1. Polyphase Decompositions

Polyphase decompositions [14] are fundamental to many applications in multirate digital signal processing [15]. The basic concept is briefly reviewed here. Let $h(n)$ be a discrete sequence. We can write its z-transform as follows:

$$\begin{aligned} H(z) &= \sum_{n=-\infty}^{\infty} h(2n)z^{-2n} + \sum_{n=-\infty}^{\infty} h(2n+1)z^{-2n-1} \\ &= E_0(z^2) + z^{-1}E_1(z^2) \end{aligned} \tag{1}$$

where $E_0(z) = \sum_{n=-\infty}^{\infty} h(2n)z^{-n}$ and $E_1(z) = \sum_{n=-\infty}^{\infty} h(2n+1)z^{-n}$.

This representation is called the two-component polyphase decomposition of $H(z)$. $E_0(z), E_1(z)$ are called polyphase components, which are the z-transforms of the even and odd numbered sampled sequences $e_0(n) = h(2n)$ and $e_1(n) = h(2n+1)$, respectively. Similarly, it is possible to represent $H(z)$ in M-component polyphase form:

$$H(z) = \sum_{k=0}^{M} z^{-k} E_k(z^M) \tag{2}$$

The sequence $h(n)$ is divided into M sub-sequences $e_k(n) = h(nM + k), k = 0, ..., M - 1$, and each of them is merely a M-fold decimated version of $h(n + k)$. One benefit of a polyphase decomposition is that the computation is reduced when dealing with sub-sequences, while computation/memory can be shared among polyphase components.

In our polyphase ASR framework, we apply the concept of polyphase decomposition onto the speech feature stream to obtain so called *polyphase features*. Specifically, let $x(1:T) = (x(1), x(2), \ldots, x(T))$ be a sequence of speech feature vectors. The M-component polyphase decomposition of this sequence results in M separate sequences $x_m(t) = x(m+tM)$, for $m = 1, \ldots, M$, each of approximate length $T' = T/M$.

2.2. Generation of Polyphase Features

The generation of polyphase features is straightforward, but is illustrated in Fig. 1 for the case of $M = 2$. For example, a basic cepstral-based feature sequence $x(n)$ is first extracted from speech using a common frame rate (e.g., 100Hz). As mentioned in Section 1, such a sample rate oversamples in the modulation domain. To overcome this problem, $x(n)$ is decomposed into two sub-sequences with even and odd numbered frames of $x(n)$, say $x_e(n) = x(2n)$ and $x_o(n) = x(2n + 1)$. Both $x_e(n)$ and $x_o(n)$ are sequences sampled at the rate of 50Hz. With a bandwidth of 25Hz, $x_e(n)$ and $x_o(n)$ do not contain high modulation frequency energy (above 25Hz), which as mentioned above is less useful for speech intelligibility. Speech features are augmented with delta and double-deltas. We apply the delta-computation only *after* the downsampling has occurred, so that each stream has its own unique delta sequence (derivatives are applied to $x_o(n)$ and $x_e(n)$ separately). When utilizing the same absolute delta window size (as measured in number of frames), these new deltas therefore integrate information over a larger time span compared to the derivatives of the original feature sequence $x(n)$. This might itself have benefit, as it has been shown that long-time features can be quite useful [16]. We refer to these extended features as *polyphase features*, and to each sub-feature sequence as a polyphase component.

Of course, the generation of polyphase features are not limited to the $M = 2$ case. For higher values of M, however, we would start with a higher initial sampling rate to avoid representing too low a modulation bandwidth range. For instance, we can decimate a 150 Hz feature sequence by a factor $M = 3$ to obtained another group of polyphase features with 3 components each at 50Hz.

3. STATISTICAL POLYPHASE MODELING

One way to deal with the multi-stream features produced above would be to have M separate recognizers, each separately trained and separately decoded, and then whose hypotheses would be combined using a standard method, say using ROVER [17]. We propose an alternative "polyphase" statistical model that essentially consists of M separate recognizers, one for each polyphase component, and then an integration network that allows for the asynchronous integration of multiple word hypotheses. This model therefore allows a joint-decision to be made regarding the best word hypothesis. It allows each polyphase component recognizer to be trained jointly as well. We utilize dynamic Bayesian networks (DBN) [12, 18, 13] to represent, encode, and implement this model — DBNs provide a flexible and powerful representational framework with which one may describe an enormous family of models, polyphase speech recognition included.

Before describing our DBN, we first provide a simple overview of our model when $M = 2$. We have a feature stream $x_{1:T}$, a high-level "integrative" set of hidden variables $h_{1:T}$, and two lower-level polyphase component hidden variables $h^e_{1:T}$ and $h^o_{1:T}$. Let $r_e = 2{:}2{:}T$ and $r_o = 1{:}2{:}T$ denote the even and odd frame indices

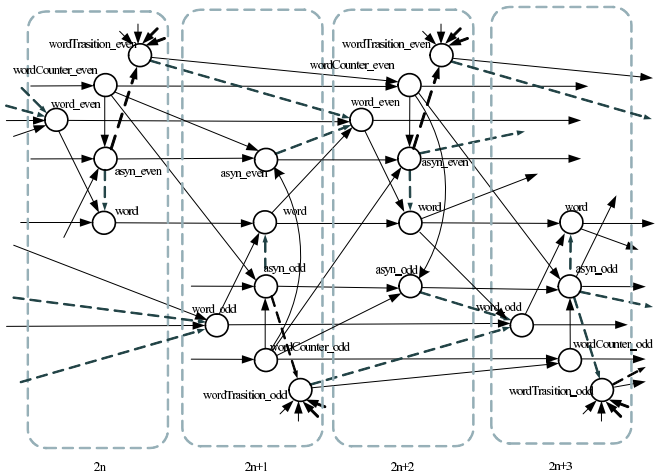

Fig. 2. A fragment of four frames of our DBN showing the asynchronous integration between two $M = 2$ polyphase components. The graph expresses word-level asynchrony, and allows the discovery of the best joint-hypotheses between the two separate polyphase component recognizers. In this figure, all nodes correspond to discrete random variables, arrows point from parents to children, and dashed arrows indicate switching parents. Other parts of the DBN (e.g., the polyphase features) are not shown for simplicity, but the parts missing are similar to two out-of-phase copies of what is described in detail in [13]

respectively. We have that:

$$
\begin{aligned}
p(x_{1:T}) &= \sum_{h_{1:T}} p(x_{1:T}|h_{1:T})p(h_{1:T}) \\
&= \sum_{h_{1:T}} p(x_{2:2:T}, x_{1:2:T}|h_{1:T})p(h_{1:T}) \\
&= \sum_{h_{1:T}} p(h_{1:T}) \prod_{j\in\{e,o\}} p(x_{r_j}|h_{1:T}) \\
&= \sum_{h_{1:T}} p(h_{1:T}) \prod_{j\in\{e,o\}} \sum_{h^j_{r_j}} p(x_{r_j}|h^j_{r_j}, h_{1:T})p(h^j_{r_j}|h_{1:T}).
\end{aligned}
$$

We see that the polyphase component features are assumed independent given the high-level integrative combination model $h_{1:T}$ and that each individual polyphase component model may have its own asynchronous evolution. Of course, the nature and extent of the asynchrony, and the details of the component models are not given here, which is where we resort to a DBN description.

3.1. Joint Decoding with Word-level Asynchrony

In this work, word-level asynchrony means that the eventual decoded word strings of all recognizers are consistent, but that the start/end time points of the words within each string may differ to a degree To achieve word-level asynchrony during decoding, a high-level combination network is expressed using DBNs. Unlike previously proposed DBNs for representing asynchrony between multiple features [6], our network not only constrains the degree of asynchrony between strings but also makes sure that there is consistency between them.

For simplicity, we describe our combination network for the case of an $M = 2$-component polyphase model, but it is easily generated into $M > 2$ multiple components, as our results in a later section show. Figure 2 shows a fragment of our DBN (the full DBN is not shown here for simplicity and space limitations). The general mechanism to achieve word-level asynchrony is as follows. At the beginning of the utterance, all streams are synchronized — in other words, the values of the word variables (`word_even` and `word_odd` in the graph) are the same at the first frame [1]. As time progresses, suppose `word_even` transfers to another word "right", while the best hypothesis for `word_odd` is still "silence". Since asynchrony is allowed in our model, `word_odd` is not forced to transfer to the word "right" immediately; meanwhile there are variables (`asyn_even` and `asyn_odd`) that keep track of the number of frames of asynchrony. When this number reaches a maximum threshold (defined as MAX_ASYN), `word_odd` is forced to be the value "right". If a word transition is triggered for `word_odd` during the allowed period of asynchrony, `word_odd` will also be forced to transfer to the value "right". This ensures the consistency of the word sequences.

Several additional variables, along with their dependencies, are described next. Variable `word`: `word_even` exists only in even frames and `word_odd` exists only in odd frames. The `word` variable always copies the value of the existing word variable (`word_even` or `word_odd`) at the current frame as long as their asynchrony variable is zero; otherwise, it will copy the value of itself from the previous frame. This variable is redundant for the 2-component case since the word variable of one component can always copy the value from the other one who is ahead of time. It is necessary, however, when combining more than two recognizers and is used to keep a record of the current decoded word value. Variables `asyn_even` and `asyn_odd`: The degree of asynchrony is calculated by comparing the indexes of words from all components (the indexes are presented by the counter variable, `wordCounter_even` and `wordCounter_odd`). This asynchrony variable has three parents. For `asyn_even`, its parents are `wordCounter_even(0)`, `wordCounter_odd(-1)`, and `asyn_even(-1)`, where 0 indicates current frame and -1 indicates previous frame. If `wordCounter_even(0)` equals the maximum value of all counters, `asyn_even(0)` is set to zero; otherwise, 1 will be added to its previous value. Variables `word_even` and `word_odd`: The major change is in the dependencies, so the logic behind the interaction between these variables is described most simply in Algorithm 1

Algorithm 1

if `wordTransition_even(-2)`$== 0$ **then**
  if `asyn_even(-1)`$<$ MAX_ASYN **then**
    No transition: `word_even(0)`$=$`word_even(-2)`
  else
    Force transition: `word_even(0)`$=$`word(-1)`
  end if
else
  if `asyn_even(-1)`$== 0$ **then**
    An usual transition using bigram
  else
    Force transition: `word_even(0)`$=$`word(-1)`
  end if
end if

4. EXPERIMENTS

Our experiments were performed on SVitchboard 1, a set of small-vocabulary tasks from Switchboard 1 [19]. In particular, we use the

[1] For the even stream, the first frame is frame 0, and for the odd stream, the first frame is frame 1.

10-word vocabulary task. The "ABC" sets are used for training, "D" set for development and "E" for the final testing. All of our models were implemented using GMTK [13].

The baseline features were generated by framing the waveforms with a 25ms length window and 10ms shift. For each frame, 12 perceptual linear prediction (PLP) coefficients plus log-energy were extracted along with their first and second derivatives, giving a feature vector of 39 dimensions. Speaker normalization was then applied to the feature vector, with the statistics of each speaker (mean and variance) estimated from Switchboard 1. A monophone HMM system was trained with word alignments. The baseline result is illustrated in Table 1, and it already significantly outperforms previously published baselines [19, 20] due to the newly generated (and differently normalized) features.

To obtain 50Hz polyphase features, two approaches were used. The first is by decimating the 100Hz 13-dim base PLP features, which were generated with 25ms window lengths and 10ms shifts. The other method is to apply a 3-component polyphase decomposition of the 150Hz 13-dim base PLP features generated with 25ms window lengths and 6.666ms shifts. These extracted features were then expanded with their first and second derivatives, forming 39-dim polyphase features. Speaker normalization was also applied to all the polyphase features by using statistics estimated from all of Switchboard 1 data. The results using these polyphase features are shown in Table 1. All polyphase component systems outperform the baseline consistently.

The 2-component decomposed polyphase features were combined for joint decoding using the graph described in Sec 3.1. Polyphase 3-component decomposed features were also combined for joint decoding. MAX_ASYN was set to 5, which means approximately up to 100ms asynchrony between word boundaries was allowed. Results (in Table 1) show that polyphase combination further improves the performance, and the joint decoding also outperforms ROVER, implemented using the NIST ROVER program which combined the outputs of the individual polyphase component recognizers.

Table 1. Word Error Rate on SVitchboard 10-word vocabulary Task.

		Dev.	Test
Baseline		16.5	16.7
2-comp. decomposition 100Hz $\downarrow$ 50Hz	even	15.3	16.3
	odd	14.6	15.9
	ROVER	15.0	16.2
	joint	13.9	**15.0**
3-comp. decomposition 150Hz $\downarrow$ 50Hz	3-1	14.8	16.4
	3-2	15.1	16.1
	3-3	14.4	15.6
	ROVER	14.0	15.6
	joint	**13.6**	15.4

5. DISCUSSION AND FUTURE WORK

We have introduced a new polyphase representation for speech recognition where a polyphase decomposition is applied to standard speech features, and a novel semi-synchronous integrative speech recognition model, expressed as a DBN, allows each polyphase component feature set to evolve separately from each other, but where word-hypotheses are jointly decoded. Results show significant improvements over an optimized SVitchboard-10 task. Future work will investigate further variations of our polyphase decomposition, will use improved modulation features [21], and will evaluate on larger speech corpora.

6. REFERENCES

[1] R. Drullman, J.M. Festen, and R. Plomp, "Effect of temporal envelope smearing on speech perception," *Journal of Acousitc Society America*, pp. 1053–1064, 1995.

[2] R. Drullman, J.M. Festen, and R. Plomp, "Effect of reducing slow temporal modulations on speech perception," *Journal of Acousitc Society America*, pp. 2670–2680, 1995.

[3] Noboru Kanedera, Takayuki Arai, Hynek Hermansky, and Misha Pavel, "On the relative importance of various components of the modulation spectrum for automatic speech recognition," *Speech Communication*, vol. 28, no. 1, pp. 43–55, 1999.

[4] N. Mirghafori, "A multi-band approach to automatic speech recognition," *Ph.D. thesis, UC Berkeley*, 1998.

[5] A. Halberstadt and J. Glass, "Heterogeneous measurements and multiple classifiers for speech recognition," *Ph.D. thesis, MIT*, Novembeer 1998.

[6] Karen Livescu and James Glass, "Feature-based pronunciation modeling with trainable asynchrony probabilities," in *Proc. ICSLP*, Jeju, South Korea, October 2004.

[7] A.P. Varga and R.K. Moore, "Hidden Markov model decomposition of speech and noise," in *Proc. ICASSP*, Alburquerque, April 1990, pp. 845–848.

[8] Y. Zhang, Q. Diao, S. Huang, W. Hu, C. Bartels, and J. Bilmes, "DBN based multi-stream models for speech," in *Proc. ICASSP*, Hong Kong, China, April 2003.

[9] S. Dupont and J. Luettin, "Audio-visual speech modelling for continuous speech recognition," *IEEE Transactions on Multimedia*, 2000.

[10] G. Gravier, S. Axelrod, G. Potamianos, and C. Neti, "Maximum entropy and MCE based HMM stream weight estimation for aud io-visual ASR," in *Proc. ICASSP*, 2002.

[11] J. N. Gowdy, A. Subramanya, C. Bartels, and J. Bilmes, "DBN-based multi-stream models for audio-visual speech recogni tion," *Proc. ICASSP*, May 2004.

[12] T. Dean and K. Kanazawa, "Probabilistic temporal reasoning," *AAAI*, pp. 524–528, 1988.

[13] Jeff Bilmes and Chris Bartels, "Graphical model architectures for speech recognition," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 89–100, September 2005.

[14] M. Bellanger, G. Bonnerot, and M. Coudreuse, "Digital filtering by polyphase network: Application to sample-rate alteration and filter banks," *IEEE Transaction on Acoustics,Speech, and Signal Processing*, vol. ASSP-24, no. 2, April 1976.

[15] P.P Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks, and applications: a tutorial," *Proceedings of IEEE*, vol. 78, pp. 56–93, Jan. 1990.

[16] T.H. Applebaum and B.A. Hanson, "Regression features for recognition of speech in quite and in noise," in *Proc. ICASSP*, Toronto, Canada 2001.

[17] J. Fiscus, "A post-processing system to yield reduced word error rates: Recogniser output voting error reduction (ROVER)," in *Proceedings 1997 IEEE Workshop on Automatic Speech Recognition and Understanding*, Santa Barbara, CA, 1997, pp. 347–352.

[18] K. Murphy, "Dynamic Bayesian Networks," 2003, from book: Probabilistic Graphical Models, M. Jordan (to appear).

[19] Simon King, Chris Bartels, and Jeff Bilmes, "Svitchboard 1: Small vocabulary tasks from switchboard 1," in *Proc. Interspeech*, Lisbon, Portugal, Sep. 2005.

[20] Karen Livescu et al, "Articulatory feature-based methods for acoustic and audio-visual speech recognition: Summary from the 2006 JHU summer workshop," in *Proc. ICASSP*, Honolulu, Hawaii, April 2007.

[21] S. Schimmel and L. Atlas, "Coherent envelope detection for modulation filtering of speech," Philadelphia, PA, March 2005.