FORWARD/BACKWARD ALGORITHMS FOR JOINT MULTI PATTERN SPEECH RECOGNITION

Nishanth Ulhas Nair and T.V. Sreenivas

Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore - 560012, India.
emails: catchnishanth@gmail.com, tvsree@ece.iisc.ernet.in

ABSTRACT

We are addressing the problem of jointly using multiple noisy speech patterns for automatic speech recognition (ASR), given that they come from the same class. If the user utters a word K times, the ASR system should try to use the information content in all the K utterances of the word simultaneously and improve its speech recognition accuracy compared to that of the single pattern speech recognition. In this paper we propose two types of Multi-pattern Joint Likelihood Forward Backward (MJLFB) algorithm to address this problem. We show an analysis of the differences between the two types of MJLFB algorithms. We also propose a new criteria (which does not require any threshold) to calculate joint probability of the feature vectors emitted given the Hidden Markov Model (HMM) state. The new formulation is tested in the context of speaker independent isolated word recognition (IWR). When 10 percent of speech is affected by burst noise at -5 dB SNR (local), it is shown that joint recognition using only two noisy speech utterances reduces the noisy speech recognition error rate by 52.40 percent, when compared to the single pattern recognition using the Forward Algorithm (FA).

1. INTRODUCTION

Robust speech recognition mainly addresses the issues arising out of pattern recognition under mismatch between training and test conditions. It also has to address issues due to out of vocabulary (OOV) words, unexpected environmental sounds, along with limited training data. There are many applications, wherein the speech recognition task per se may be limited, such as limited vocabulary isolated word recognition (IWR) but the performance is severely affected because of high levels of ambient noise (example: street noise, machine floor noise, speech babble noise, etc.) or bursty channel noise (example: HF radio, packet loss in IP networks, etc.) or even foreign accent pronunciation affecting certain parts of a word. Keeping such applications in mind, we have proposed a new approach of harnessing better IWR performance using multiple repetitions of a word while testing [1]. Using clean speech IWR models, but test utterances corrupted with high levels of burst noise, we could show that the new solution can provide nearly 50% reduction in the word error rates. In that paper we presented a multi pattern joint decoding algorithm using a combination of multi-pattern DTW (dynamic time warping) as well as HMM Viterbi path search using a variety of multiple pattern emission likelihood models. This new algorithm determines a more robust likelihood score and maximum likelihood (ML) state transition path through the HMM compared to the independent decoding of the noisy patterns.

In this paper, we address the next issue of evaluating the total probability (or likelihood) of the multiple patterns through all possible paths of the HMM, instead of only the ML path. We develop the forward/backward algorithm for joint evaluation of multi pattern likelihood. With this algorithm also, we find that there is significant advantage to multi-pattern joint evaluation, when compared to independent single pattern evaluation.

In the literature, there are few similar approaches to robust ASR: in [2] they attempt to recognize multiple utterances spoken simultaneously by different talkers, where each utterance is an interference to the other; whereas in our approach multiple patterns aid each other for better recognition. There are other simultaneous decoding algorithms, such as in [3] [4], for finding the second best, third best likelihood scores in sentence recognition. In [5], a 3D HMM search space and a Viterbi-like decoding algorithm was proposed for Utterance Verification. The two axes in the trellis belonged to HMM states and one axis belongs to the observation sequence. In our paper we have multiple observation sequences jointly analyzed using one HMM model for improving speech recognition performance.

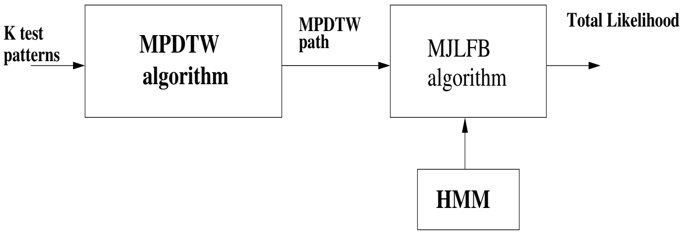

Figure 1: *Joint recognition of K patterns*

2. JOINT MULTI PATTERN LIKELIHOOD

Our goal is to determine the probability likelihood of the multiple patterns, recognizing that they belong to the same model λ and constrain the temporal evolution of the HMM states, across the multiple patterns, unlike independent evaluation of the individual path.

Consider a T length observation vector sequence (also called pattern or utterance), $O_{1:T} = (O_1, O_2, \ldots, O_T)$. The forward algorithm (FA) or backward algorithm [6] can be used to calculate the total probability of the sequence $O_{1:T}$ given the HMM λ: i.e., $P(O_{1:T}/\lambda)$. In the problem that we are addressing, we have K number of observation vector speech sequences $O^1_{1:T_1}$, $O^2_{1:T_2}$, …, $O^K_{1:T_K}$, of lengths T_1, T_2 , …, T_K, respectively, where $O^i_{1:T_i} = (O^i_{1:1}, O^i_{2:2}, \ldots, O^i_{T_i:T_i})$ is the observation vector sequence of the i^{th} pattern and $O^i_{t_i:t_i}$ is the feature vector of the i^{th} pattern at time frame t_i. Let each of these K observation sequences belong to the same pattern class (spoken word); they are different utterances of the same word spoken by the same speaker. Some of these utterances could be affected by random burst noise or may be wrongly pronounced. We want to jointly recognize them in an “intelligent” way, so as to improve speech recognition performance. This is realized by identifying which feature vectors are noisy or unreliable, and then use this information for improving their joint likelihood. The joint likelihood problem is proposed to be solved using a two stage algorithm in which the first stage aligns the multiple patterns for least distortion using a multi pattern dynamic time warping (MPDTW) algorithm and then in the second stage the HMM evolution of observation vectors of the multiple patterns is tracked along the optimum MPDTW path (see Figure 1).

The Dynamic Time Warping (DTW) algorithm has been formulated to obtain the least distortion alignment path between two patterns. We have extended this algorithm to multiple patterns such as the K patterns above. Figure 2 shows an example of such an alignment for K=3. This least distortion path is called the MPDTW

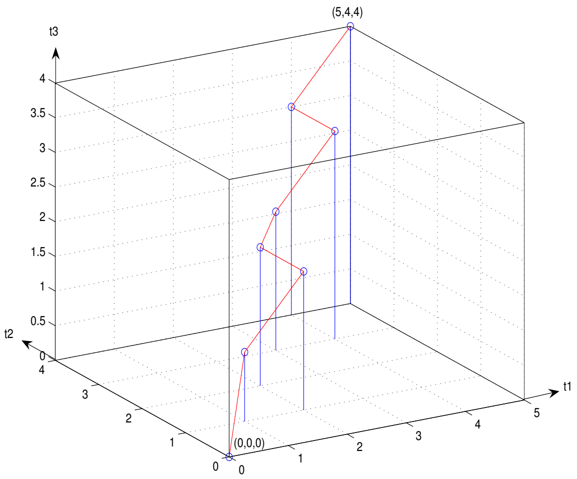

Figure 2: *Example MPDTW path for 3 utterances; K = 3*

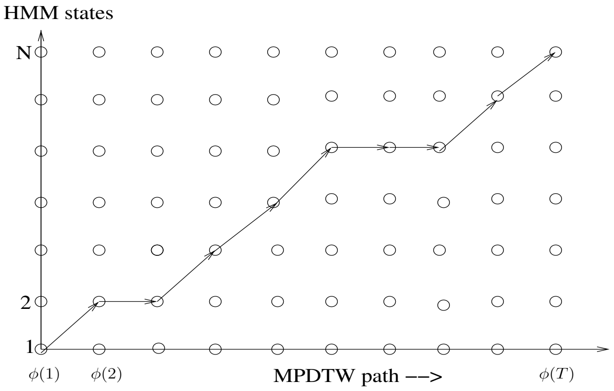

Figure 3: Example state sequence along MPDTW path for Left-Right HMM

path. Let ϕ be the optimum MPDTW path for K patterns and $\phi(t) = (t_1, \ldots, t_K)$ where $(t_1, \ldots, t_K)$ are the coordinates of the optimum path in the K dimensional rectangular grid of the optimum path. The optimum MPDTW path length is $T \geq max(T_1, T_2, \ldots, T_K)$ and $\phi(t)$, $1 \leq t \leq T$ is the sequence of path coordinates. The terminal points have to be $\phi(1) = (1, \ldots, 1)$ and $\phi(T) = (T_1, \ldots, T_K)$. $\phi = (\phi(1), \phi(2), \ldots, \phi(T))$. The algorithm to determine ϕ is presented in [1]. Endpoint errors are not a problem as the MPDTW algorithm can be easily extended to relax end point constraints.

Since our goal is to determine the joint likelihood of all the K patterns given a HMM, we consider the MPDTW path $\phi(t)$, $1 \leq t \leq T$ as a 1-D evolution of the multiple patterns. Thus, the joint likelihood can be determined by forming a trellis of HMM states and $\phi(t)$. We can now develop the forward/backward algorithm between HMM states and $\phi(t)$.

These algorithms are used to calculate the total probability in an "intelligent" manner such that we are making use of the "best" information possible and avoiding the noisy or unreliable information among multiple patterns.

2.1 Forward/Backward Algorithm (MJLFB)

We refer to this algorithm as "multi-pattern joint likelihood forward/backward" (MJLFB) algorithm. Figure 3 shows the typical HMM trellis and a typical HMM transition path is indicated. Although the example corresponds to a left-right HMM, the development below is valid for any general ergodic HMM with a full transition matrix. Following the terminology of a standard HMM [6], we define the forward variable $\alpha_{\phi(t)}(j)$ and the backward variable $\beta_{\phi(t)}(j)$ to represent the partial joint-likelihood along the path $\phi(t)$; i.e.,

$$\alpha_{\phi(t)}(j) = P(O^1_{1:t_1}, O^2_{1:t_2}, \ldots, O^K_{1:t_K}, q_{\phi(t)} = j/\lambda) \qquad (1)$$

$$\beta_{\phi(t-1)}(j) = P(O^1_{t_1:T_1}, \ldots, O^K_{t_K:T_K}/q_{\phi(t-1)} = j, \lambda) \qquad (2)$$

such that $\sum_{j=1}^{N} \alpha_{\phi(t)}(j).\beta_{\phi(t)}(j) = P(O^1_{1:T_1}, O^2_{1:T_2}, \ldots, O^K_{1:T_K}/\lambda)$ $\forall$ t. $q_{\phi(t)}$ is state at $\phi(t)$ and λ is the HMM model. Thus we can utilize either $\alpha_{\phi(t)}(j)$ or $\beta_{\phi(t)}(j)$ alone to determine the total joint-likelihood by using appropriate initial conditions. The initial conditions for recursive evaluation of $\alpha_{\phi(t)}$ or $\beta_{\phi(t)}$ is given by:

$\alpha_{\phi(1)}(j) = \Pi_j.P(O^1_{1:1}, O^2_{1:1}, \ldots, O^K_{1:1}/q_{\phi(1)} = j, \lambda); 1 \leq j \leq N.$

$\beta_{\phi(T)}(j) = 1; 1 \leq j \leq N$, where N is the total number of HMM states, Π_j is the state initial probability at state j.

Let us define the MPDTW path transition vector, $\Delta\bar{t} = \phi(t) - \phi(t-1) = (\Delta t^1_1, \Delta t^2_2, \ldots, \Delta t^K_K)$. Depending on the local constraints chosen in MPDTW, $\Delta\bar{t}$ can be a K dimensional vector of only 0's and 1's; example: $\Delta\bar{t} = [0, 1, 1, 0, \ldots, 1]$. $\Delta\bar{t}$ will comprise of at least one non-zero value and a maximum of K non-zero value. Let set $S_{\bar{t}} = \{O^i_{t_i:t_i} | \Delta t^i_i \neq 0, i = 1, 2, \ldots, K\}$. Let $\{O_{\phi(t)}\} = (O^m_{t_m:t_m}, \ldots, O^n_{t_n:t_n})$ such that $(O^m_{t_m:t_m}, \ldots, O^n_{t_n:t_n})$ are all the feature vectors in the set $S_{\bar{t}}$. $\{O_{\phi(t)}\}$ is a subset of the vectors $(O^1_{t_1:t_1}, O^2_{t_2:t_2}, \ldots, O^K_{t_K:t_K})$, retaining only those $O^k_{t_k:t_k}$ whose Δt^k_k is non-zero. The set $S_{\bar{t}}$ and $\{O_{\phi(t)}\}$ can have a minimum of one feature vector and a maximum of K feature vectors. Let $\bar{t^*} = \{t_i | \Delta t^i_i \neq 0, i = 1, 2, \ldots, K\}$. The recursion for the evaluation of $\alpha_{\phi(t)}(j)$ along all the grid points of the trellis (shown in Figure 3) is given by:

$$\alpha_{\phi(t)}(j) = \sum_{i=1}^{N} [\, \alpha_{\phi(t-1)}(i).a_{ij}].b_j(\{O_{\phi(t)}\}) \qquad (3)$$

$t = 2, 3, \ldots, T,\ j = 1, 2, \ldots, N.$ a_{ij} is the state transition probability from state i to state j (as in standard HMM), $b_j(\{O_{\phi(t)}\})$ is the joint likelihood of $\{O_{\phi(t)}\}$ being emitted by state j. It is the same as joint likelihood of all the vectors $(O^m_{t_m:t_m}, \ldots, O^n_{t_n:t_n})$ emitted by state j, where $(O^m_{t_m:t_m}, \ldots, O^n_{t_n:t_n})$ consist of all the feature vectors in the set $S_{\bar{t}}$. Here the state j can emit a variable number of vectors ranging from 1 to K, corresponding to the number of non-zero values in the $\Delta\bar{t}$ vector. Because of this, when $\alpha_{\phi(t)}$ reaches $\alpha_{\phi(T)}$ each state j would have emitted the exact number of multi-pattern feature vectors = $(T_1 + T_2 + \ldots + T_K)$. One way to calculate the joint likelihood is $b_j(\{O_{\phi(t)}\}) = \prod_{O^i_{t_i:t_i} \in S_{\bar{t}}} b_j(O^i_{t_i:t_i})$.

At termination the total joint likelihood is computed as:

$$P(\{O^{1:K}\}/\lambda) = P(O^1_{1:T_1}, O^2_{1:T_2}, \ldots, O^K_{1:T_K}/\lambda) = \sum_{i=1}^{N} \alpha_{\phi(T)}(i) \qquad (4)$$

$P(\{O^{1:K}\}/\lambda)$ is the total likelihood of K utterances.

We can write similar recursive equations for the backward variable:

$$\beta_{\phi(t-1)}(i) = \sum_{j=1}^{N} a_{ij}.\beta_{\phi(t)}(j).b_j(\{O_{\phi(t)}\}) \qquad (5)$$

$t = T, \ldots, 2$, $i = 1, 2, \ldots, N$.

$$P(\{O^{1:K}\}/\lambda) = \sum_{i=1}^{N} \Pi_i.b_i(\{O_{\phi(1)}\}).\beta_{\phi(1)}(i) \qquad (6)$$

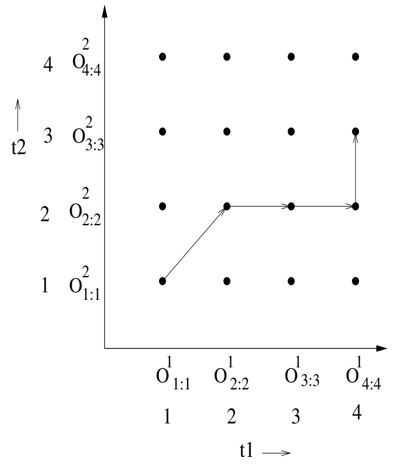

Figure 4: *Example MPDTW path for K=2*

Figure 4 provides an example of this algorithm. We show K=2 example with an initial portion of the MPDTW path $\phi(t)$. The t1 time axis is for utterance $O^1_{1:T_1}$ and t2 time axis is for utterance $O^2_{1:T_2}$. We fit a layer of HMM states (of a class) on this path (Figure 3). Now we traverse along this MPDTW path. Consider a point on the HMM trellis, say state j and $\phi(t=2) = (t1=2, t2=2)$. For this point $\Delta\bar{t} = (1,1)$ and hence we emit both $O^1_{2:2}$ and $O^2_{2:2}$. For the next point, i.e., $\phi(t=3) = (t1=3, t2=2)$, $\Delta\bar{t} = (1,0)$ and hence we emit only one vector, i.e., $O^1_{3:3}$.

2.2 MJLFB-Approximate

We also consider a variant of the algorithm in section 2.1, wherein instead of emitting a variable number of vectors at each HMM state transitions, we emit all the K vectors corresponding to the K pattern joint likelihood. So we are basically creating a new virtual utterance which is some kind of a combination of all the K utterances (with repetitions of feature vectors possible) and which lies on the MPDTW path. We are doing recognition on this virtual utterance. The total number of feature vectors emitted at the end of the MPDTW path by each state will be greater than or equal to $T_1 + T_2 + \ldots + T_K$. Although this algorithm is an approximate solution, it has some advantages over the exact solution of MJLFB algorithm. The recursive solution to calculate the forward variable $\alpha_{\phi(t)}(j)$ is as follows:

$$\alpha_{\phi(t)}(j) = \sum_{i=1}^{N} [\,\alpha_{\phi(t-1)}(i).a_{ij}].b_j(O^1_{t_1:t_1}, \ldots, O^K_{t_K:t_K}) \qquad (7)$$

$t = 2,3,\ldots,T,\ j = 1,2,\ldots,N.$ $b_j(O^1_{t_1:t_1}, \ldots, O^K_{t_K:t_K})$ is the joint likelihood of the observations $O^1_{t_1:t_1}, \ldots, O^K_{t_K:t_K}$ generated by state j. Similarly a recursive equation can be given for calculating the backward variable also. One way to calculate the joint likelihood (approximate) is $b_j(\{O_{\phi(t)}\}) = b_j(O^1_{t_1:t_1}).b_j(O^2_{t_2:t_2}) \ldots b_j(O^K_{t_K:t_K})$.

2.3 Feature Vector Weighting

In missing feature theory approaches [7], [8] one attempts to determine which cells of a spectrogram-like time frequency display of speech information are unreliable (or missing) because of degradation due to noise or to other types of interference. The cells that are determined to be unreliable or missing are either ignored in subsequent processing and statistical analysis, or they are filled in by optimal estimation of their putative values [8].

Like in the missing feature theory approach, it is important to identify which portions of speech are unreliable. We can give a lesser or zero weighting to the unreliable (noisy) feature vectors and a higher weighting to the reliable ones. In our earlier paper [1], we have considered various criteria for weighting the feature vectors, to achieve robustness to burst noise. In particular Criteria-3 of [1] was found most promising and hence used in the present experiments also. A hard threshold is required for this criteria. In Criteria-3, the joint likelihood $b_j(\{O_{\phi(t)}\})$ for equations (3) and (5) is calculated as follows:

$$b_j(\{O_{\phi(t)}\}) = \begin{cases} \left[\prod_{O^i_{t_i:t_i} \in S_{\bar{t}}} b_j(O^i_{t_i:t_i})\right]^{\frac{1}{r}} & \text{if } d(\bar{t^*}) < \gamma \\ \max_{O^i_{t_i:t_i} \in S_{\bar{t}}} b_j(O^i_{t_i:t_i}) & \text{if } d(\bar{t^*}) \geq \gamma \end{cases} \qquad (8)$$

where γ is a threshold, r is the cardinality of the set $S_{\bar{t}}$. The multi-vector distance measure is defined as, $d(\bar{t^*}) = \sum_{O^i_{t_i:t_i} \in S_{\bar{t}}} d(O^i_{t_i:t_i}, C_{\bar{t^*}})$, where $C_{\bar{t^*}}$ is the centroid of the all vectors in $S_{\bar{t}}$ and $d(O^i_{t_i:t_i}, C_{\bar{t^*}})$ is the Euclidean distance between $O^i_{t_i:t_i}$ and $C_{\bar{t^*}}$. In equation (8), if we choose $\gamma = \infty$, then $b_j(\{O_{\phi(t)}\})$ is always equal to $\prod_{O^i_{t_i:t_i} \in S_{\bar{t}}} b_j(O^i_{t_i:t_i})$ (product operation), and when $\gamma < 0$, then it is always equal to $\max_{O^i_{t_i:t_i} \in S_{\bar{t}}} \{b_j(O^i_{t_i:t_i})\}$ (max operation). Similarly we can get a equation for MJLFB-Approximate algorithm.

In addition, we have considered a new weighting criteria without the need for the threshold γ used in Criteria-3 of [1]. We refer to this as Criteria-5. This is because in Criteria-3, the optimum value of threshold γ might be difficult to find in a natural noisy environment, where the noise will have varying power. Also the threshold used is a hard threshold and is non adaptive. For MJLFB-Approximate algorithm, we define $b_j(O^1_{t_1:t_1}, \ldots, O^K_{t_K:t_K})$ as follows:

$$b_j(O^1_{t_1:t_1}, \ldots, O^K_{t_K:t_K}) = \prod_{i=1}^{K} b_j(O^i_{t_i:t_i})^{\{b_j(O^i_{t_i:t_i})/\sum_{i=1}^{K}(b_j(O^i_{t_i:t_i}))\}} \qquad (9)$$

Equation (9) basically finds a weighted geometric mean of the $b_j(O^i_{t_i:t_i})$'s values, where each $b_j(O^i_{t_i:t_i})$ is raised to the power of the percentage of its contribution. When the values of $b_j(O^i_{t_i:t_i})$'s are close to each other, then Criteria-5 becomes close to the product operation of Criteria-3 and if the various values $b_j(O^i_{t_i:t_i})$'s are very different, then Criteria-5 becomes close to max operation of Criteria-3. Criteria-5 behaves somewhat similar to Criteria-3 (when Criteria-3 is set with optimum threshold). The advantage of using Criteria-5 is that we don't need to set any threshold.

For MJLFB algorithm, a similar equation is given as follows:

$$b_j(\{O_{\phi(t)}\}) = \prod_{O^i_{t_i:t_i} \in S_{\bar{t}}} b_j(O^i_{t_i:t_i})^{\{b_j(O^i_{t_i:t_i})/\sum_{O^i_{t_i:t_i} \in S_{\bar{t}}}(b_j(O^i_{t_i:t_i}))\}} \qquad (10)$$

3. ANALYSIS OF MJLFB AND MJLFB-APPROXIMATE

Now let us analyze which of these two algorithms - MJLFB algorithm and MJLFB-Approximate algorithm - is better and under what conditions. In the example shown in Figure 4, let us assume that the vector $O^2_{2:2}$ is clean and the vectors $O^1_{3:3}$ and $O^1_{4:4}$ are noisy or badly articulated. Let us use Criteria-3 to calculate joint probability and let us choose a low value for the threshold, so that the max operation dominates. Using MJLFB-Approximate algorithm, since $O^2_{2:2}$ is re-emitted (by state j) at time instants (3,2) and (4,2), the max operation will be most likely used as the multi vector distortion measure will most likely be above the threshold. So only the clean $O^2_{2:2}$ vector will be used to calculate joint probability. However in the case of MJLFB algorithm, since $O^2_{2:2}$ is emitted only once at time instant (2,2) and not emitted at time instants (3,2) and (4,2), only the noisy vectors $O^1_{3:3}$ and $O^1_{4:4}$, contribute to the calculate of joint probability. This affects $P(\{O^{1:K}\}/\lambda)$. So in this case the recognition accuracy is likely to decrease if we use MJLFB algorithm when compared to MJLFB-Approximate algorithm.

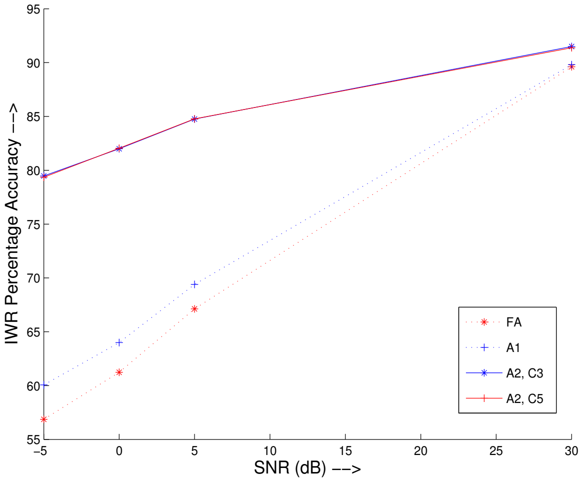

Figure 5: *Percentage accuracies for experiments FA, A1, A2 for different levels of burst noises. FA - Forward Algorithm, A1 - best of two utterances using FA, A2 - MPDTW algorithm + MJLFB-Approximate algorithm. Results for A2 using Criteria-3 (C3) are shown at threshold* $\gamma = 0.5$. *C5 stands for Criteria-5.*

Now let us consider an other case when we are using Criteria-3 and the value of the threshold is very high, so that the product operation dominates. Let vector $O^2_{2:2}$ be noisy or badly articulated and vectors $O^1_{3:3}$ and $O^1_{4:4}$ are clean. Since the product operation will mostly be used, using MJLFB-Approximate algorithm, the noisy vector $O^2_{2:2}$ will affect the calculation of the joint probability at time instants (3,2) and (4,2) as it is re-emitted. Now using MJLFB algorithm, as vector $O^2_{2:2}$ is not re-emitted, only the clean vectors $O^1_{3:3}$ and $O^1_{4:4}$ contribute to the calculation of joint probability. So MJLFB algorithm is expected to give better speech recognition accuracy than MJLFB-Approximate algorithm.

For the case of clean, well articulated speech, MJLFB algorithm is expected to perform better than MJLFB-Approximate algorithm as it is the "exact" way of joint recognition. This is true when we use Criteria-3 at lower values of threshold. At higher (sub optimal) values of threshold, MJLFB-Approximate algorithm could be better.

If Criteria-5 is used, we expect that using MJLFB algorithm would give better recognition accuracy than MJLFB-Approximate algorithm for well articulated clean speech and worse values for speech with burst noise or speech with bad articulation. This is because Criteria-5 behaves similar to MJLFB-Approximate algorithm when the threshold of MJLFB-Approximate algorithm is optimum.

Finally we conclude that if we look at the best performances of MJLFB-Approximate algorithm and MJLFB algorithm, MJLFB-Approximate algorithm is better than MJLFB algorithm for noisy speech (burst noise), and MJLFB algorithm is better than MJLFB-Approximate algorithm for clean speech.

4. EXPERIMENTS

Based on the formulations of sections 2 and 3 we conducted experiments - A1, A2, A3 for speaker independent IWR along with the base line system of standard FA for a single pattern, for the cases of both clean and noisy speech. Since the normal FA uses one utterance (pattern) to make a recognition decision and the proposed algorithms use K utterances to make a decision, the comparison of results may not be fair. For a fairer comparison we formulated the experiment A1, which also uses K utterances using the standard FA and the best likelihood of the K utterances is chosen. So we compare the new algorithms (experiment A2 and A3) with this experiment A1 also.

The experiment A1 is as described. Given $O^1_{1:T_1}$, $O^2_{1:T_2}$, ..., $O^K_{1:T_K}$ as the individual patterns, we can obtain the joint likelihood score as $\theta_j = \max_{1 \le i \le K} P(O^i_{1:T_i}/\lambda_j)$, where λ_j are the clean word models and the FA is used to calculate $P(O^i_{1:T_i}/\lambda_j)$. We select the word as $j^* = \arg\max_j \theta_j$. We have restricted to two patterns. For each word of a test speaker, A1 is done for utterance 1 and utterance 2, utterance 2 and 3, utterance 3 and 1. Experiment A2 is the MPDTW algorithm followed by MJLFB-Approximate algorithm proposed in this paper. Experiment A3 is the MPDTW algorithm followed by MJLFB algorithm proposed in this paper. In all joint recognition experiments (A2 and A3), we have restricted to two pattern joint recognition and compared the performance with respect to single pattern recognition (FA and A1). Thus, for each word of a test speaker, utterance 1 is jointly recognized with utterance 2, utterance 2 with 3, utterance 3 with 1. Please note that in the noisy case (burst noise), all the three utterances are noisy. As the number of test utterances $K = 2$, for the new experiments we chose the Local Continuity Constraints for MPDTW as (1,0) or (0,1) or (1,1) and the slope weighting function is equal to 1.

We used IISc-BPL database which contains 75 word vocabulary for 36 female and 34 male adult speakers, with three repetitions for each word by the same speaker, digitized at 8kHz sampling rate. The vocabulary consists of a good number of phonetically confusing words used in Voice Dialer application. Left to Right HMMs are trained for clean speech using the Segmental K Means (SKM) algorithm. 25 male and 25 female speakers are used for training, with three repetitions of each word by each speaker. We tested the algorithm for 20 unseen speakers (11 female and 9 male) in both clean and noisy cases. Test words are three utterances for each word by each speaker, at each Signal to Noise Ratio (SNR). In the noisy case, burst noise was added to 10% of the frames of each word at -5 dB, 0 dB, 5 dB SNRs (local) to all the three utterances. (The remaining 90% of the frames are clean; the range of -5dB to +5dB indicates severe to mild degradation of the 10% frames.) The burst noise can occur randomly anywhere in the spoken word with uniform probability distribution. MFCC, Δ MFCC, and Δ^2 MFCC is used without the energy components (total 36 dimension vector). Cepstral Mean Subtraction was done. Variable number of states are used for each word model; i.e. using the average duration of the training patterns, for each second of speech, 8 HMM states were assigned, with 3 Gaussian mixtures per state.

We experimented for various values of the threshold γ. Hence we experimented with a range of values for γ and found that there is indeed an optimum value. For the noisy patterns with burst noises at -5 dB SNR, $\gamma = 0.5$ is found to be optimum. It is also clear that $\gamma < 0$ provides closer to optimum performance than $\gamma = \infty$, indicating that the max operation is more robust than the product operation.

The results for clean and noisy speech is given in Table 1. In the table, ASRA (Cl) stands for ASR accuracy for clean speech. In the tables, for experiment A2, in the ASRA column, the ASR accuracy and the symbol C3 or C5 is written. C3 and C5 stands for experiment carried out using Criteria-3 and Criteria-5 respectively. In the table -5dB ASRA stands for ASR Accuracy for noisy speech which has burst noise of 10% at SNR -5dB. It can be seen that the baseline performance of FA for clean speech is close to 90%. For example, for noisy case at -5 dB SNR burst noise it decreases to $\approx 57\%$. Interestingly, the experiment A1 provides a mild improvement of 0.2% and 3.2% for clean and noisy speech (at -5dB SNR burst noise) respectively, over the FA benchmark. This shows that use of multiple patterns is indeed beneficial, but just maximization of likelihoods is weak. The proposed new algorithms for joint recognition provides dramatic improvement for the noisy case, w.r.t. the FA performance. For example at -5 dB SNR burst noise the proposed algorithms (experiment A2 and A3) using Criteria-3 at threshold $\gamma = 0.5$, gave an improvement of 22.60% speech recognition accuracy (using MJLFB-Approximate algorithm) and 20.31% speech recognition accuracy (using MJLFB algorithm) compared to FA performance. We also see that as the SNR improves, the gap in the speech recognition accuracy between performance of Criteria-3

Table 1: *Comparison of ASR percentage accuracy (ASRA) for clean and noisy speech for FA, A1, A2, and A3. FA - Forward Algorithm, Experiment A1 - best of two utterances using FA, Experiment A2 - MPDTW algorithm + MJLFB-Approximate algorithm, Experiment A3 - MPDTW algorithm + MJLFB algorithm. C3 - using Criteria-3 for A2 or A3, C5 - using Criteria-5 for A2 or A3*

Algorithm	ASRA(Cl)	ASRA(Cl)	5 dB ASRA	5 dB ASRA	0 dB ASRA	0 dB ASRA	-5 dB ASRA	-5 dB ASRA
FA	89.61%	89.61%	67.13%	67.13%	61.24%	61.24%	56.87%	56.87%
A1	89.81%	89.81%	69.40%	69.40%	64.00%	64.00%	60.07%	60.07%
C3, $\gamma = \infty$	91.87%, A2	91.80%, A3	72.42%, A2	72.22%, A3	66.07%, A2	66.11%, A3	61.31%, A2	61.53%, A3
C3, $\gamma = 2$	91.87%, A2	91.78%, A3	80.58%, A2	80.11%, A3	77.02%, A2	76.02%, A3	73.91%, A2	73.18%, A3
C3, $\gamma = 1$	91.80%, A2	92.13%, A3	83.80%, A2	82.24%, A3	80.82%, A2	78.76%, A3	78.09%, A2	76.09%, A3
C3, $\gamma = 0.75$	91.73%, A2	92.18%, A3	84.51%, A2	83.05%, A3	81.69%, A2	79.76%, A3	79.13%, A2	76.98%, A3
C3, $\gamma = 0.5$	91.49%, A2	92.02%, A3	84.76%, A2	83.44%, A3	82.00%, A2	79.84%, A3	79.47%, A2	77.18%, A3
C3, $\gamma = 0.25$	91.49%, A2	91.98%, A3	84.73%, A2	83.38%, A3	82.00%, A2	79.91%, A3	79.44%, A2	77.09%, A3
C3, $\gamma < 0$	91.49%, A2	91.98%, A3	84.73%, A2	83.38%, A3	82.00%, A2	79.91%, A3	79.44%, A2	77.09%, A3
C5	91.38%, A2	92.02%, A3	84.78%, A2	83.38%, A3	82.05%, A2	79.96%, A3	79.35%, A2	77.11%, A3

at threshold $\gamma = \infty$ and $\gamma < 0$ reduces. In fact as SNR approaches to that of clean speech, $\gamma = \infty$ is better than $\gamma < 0$. We see that for clean speech, the speech recognition accuracy improved by 2.26% using MJLFB-Approximate algorithm and 2.57% using MJLFB algorithm over that of FA. Hence you see MJLFB algorithm is better than MJLFB-Approximate algorithm for clean speech. From our experiments, we know that MJLFB-Approximate algorithm gives similar recognition results compared to the Constrained Multi Pattern Viterbi algorithm proposed in [1].

A graph showing variation of IWR percentage accuracy versus the burst noise at some SNR is shown in Figure 5. In this figure the experiment A2 using Criteria-3 is plotted at threshold $\gamma = 0.5$, where Criteria-3 works very well for speech with burst noise. We see that using Criteria-5 gives us optimum or near optimum values. The performance of Criteria-5 is equal to to the optimum performance of Criteria-3, as we predicted in section 3. And the advantage of Criteria-5 is that we don't need any threshold. We also see that as per our analysis in section 3 and the results shown in Table 1, using Criteria-3 for MJLFB algorithm (experiment A3) for clean speech at lower thresholds gives better recognition results than using it for MJLFB-Approximate algorithm (experiment A2). At higher thresholds MJLFB-Approximate algorithm is better. For noisy speech (speech with burst noise) it is better to use MJLFB-Approximate algorithm than MJLFB algorithm.

The proposed approaches will be slower than the baseline algorithms because of the MPDTW algorithm. As K increases, the time taken by the MPDTW algorithm to run will increase exponentially with K. But this can be reduced by choosing appropriate global path constraints for the MPDTW algorithm.

We also experimented on speech affected by speech babble noise. Babble noise from NOISEX 92 was added to the entire speech utterances at 5 dB and 10 dB SNR. We found that speech affected by babble noise at 5 dB SNR, the FA gave 44.18% speech recognition accuracy. Using the MJLFB algorithm (experiment A3 for $K = 2$ utterances) using Criteria-5, the accuracy increased to 47.80%. At 10 dB SNR noisy speech (babble noise), the FA gave an accuracy of 59.64%, while the MJLFB algorithm (using Criteria-5) gave an accuracy of 64.36%. We see that the improvement in the recognition accuracy in the babble noise case is not as much as in the burst noise case. Since the burst noise affects only a small region of the utterances, our proposed methods do better in successfully neglecting the noisy portions of the utterances and using the clean portion of the other utterances.

5. CONCLUSIONS

In this paper we addressed the problem of jointly using multiple utterances of speech to improve speech recognition performance (over that of a single utterance), especially in the presence of burst noise. We proposed two types of MJLFB algorithms to address this problem. We got a very significant improvement in ASR accuracy for both clean and noisy speech (burst noise with unknown characteristics). We also got some improvement in ASR accuracy when speech was affected by babble noise. We proposed a new criteria (which does not require any threshold) to calculate the joint likelihood of feature vectors emitted from some given state and HMM model. We also analyzed the two versions of the MJLFB algorithms under various conditions. From our experimental results, we come to the conclusion that using multiple speech utterances is indeed important and it greatly helps in improving speech recognition performance.

REFERENCES

[1] N.U. Nair and T.V. Sreenivas, "Joint Decoding of Multiple Speech Patterns For Robust Speech Recognition," *IEEE Workshop on Automatic Speech Recognition and Understanding.*, pp. 93-98, 9-13 Dec. 2007.

[2] A.N. Deoras, A. Hasegawa-Johnson, "A factorial HMM approach to simultaneous recognition of isolated digits spoken by multiple talkers on one audio channel," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 1, pp. I- 861-4, May 2004.

[3] R. Schwartz, Y.-L Chow, "The N-best algorithms: an efficient and exact procedure for finding the N most likely sentence hypotheses," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 1, pp. 81-84, Apr. 1990.

[4] J Wu, V Gupta, "Application of simultaneous decoding algorithms to automatic transcription of known and unknown words," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 2, pp. 589-592, Mar. 1999.

[5] E. Lleida, R.C. Rose, "Utterance verification in continuous speech recognition: decoding and training procedures", *IEEE Trans. on Speech and Audio Process.*, vol. 8, issue: 2, pp: 126-139, Mar 2000.

[6] L.R. Rabiner and B-H. Juang, *Fundamentals of Speech Recognition.*, Pearson Education Inc, pp. 240-262, 1993.

[7] M.P. Cooke, P.G. Green, and M.D. Crawford, "Handling missing data in speech recognition," *Proc. Int. Conf. Spoken Lang. Process.*, pp. 1555-58, 1994.

[8] B. Raj and R.M. Stern, "Missing-feature approaches in speech recognition," *IEEE Signal Process. Magazine.*, vol. 2, pp. 101-116, Sep 2005.