Robust Speech Recognition Against Packet Loss

Manhung Siu,Yu-Chung Chan

Hong Kong University of Science and Technology
{eemsiu,arthur}@ee.ust.hk

Abstract

Recognizing speech transmitted over mobile or computer networks poses new challenges such as packet loss in transmission. Viterbi algorithm, the most common speech recognition approach, searches for the most likely state sequence that explains all observation. However, because it implicitly sums the log observation probabilities, the resulting solution is sensitive to outlier frames. In this paper, we propose a robust approach that searches the state sequence that best explains x percent of the observation and is insensitive to the corruption of a limited number of frames. We evaluated the proposed algorithm on the TI-digits task. With 10% of the data loss, the proposed algorithm achieves improvement of WER 71.6% for isolated digit recognition and 32.2% for connected digit recognition.

1. Introduction

With the advancement in telephony technologies, telephone call can now be carried through a vast variety of networks such as mobile networks and the internet. Recognizing speech transmitted through these networks poses new challenges. One of them is the problem of packet loss. Further complicating the problem is the fact that very often, multiple networks are involved in transmitting phone calls making it hard to identify the lost packets at the receiving end. For example, a caller uses his mobile phone to connect to a voice information service provider, such as a stock quotation server, that is connected to a traditional phone line. This phone call will be routed first to the base-station and then, connected to the traditional phone system. If the mobile phone connection between the user and the base-station incurs losses, the missing speech will be handled by the base-station, for example, by replacing it with comforting noise and the "fixed" speech is transmitted to the voice portal. The exact time of the loss information is no longer available to the voice portal server.

In this paper, we introduce a robust approach that aims at handling corrupted frames of speech. In traditional speech recognition, we search for the state sequence that maximizes the joint observation and state sequence probability $p(O, Q)$ which implicitly sums the observations log likelihoods. Analogous to the problem of estimating mean by averaging, a single corrupted frame (outlying data point) can have significant effect on the resulting score of a path. Consider a case in isolated digit recognition that a user say "one" and one of the vowel's frame are corrupted due to packet loss. If the corrupted frame matches with the vowel sound poorly, then, this single corrupted frame can prevent the recognizer from getting the correct answer.

Instead of computing the score of a path by summing all the corresponding log observation probabilities, we propose computing the score by ignoring the contribution of worst scoring $100 - x$ percent of speech frames. This is similar to the use of trimmed mean or the median. We proposed a modified Viterbi algorithm that can efficiently find the best path that keeps only the x percent of the frames.

The rest of the paper is organized as follows. In Section 2, we briefly review the Viterbi algorithm and describe our proposed modification. In Section 3, we describe two possible implementations of the proposed algorithm. In Section 4, we report the experimental results. Finally, we conclude our work in Section 5.

2. Proposed Algorithm

2.1. The Most Likely State Sequence

HMM based speech recognition is typically viewed as finding the most likely state sequence $\hat{Q}_1^T$ given the observation O_1^T as:

$$\hat{Q}_1^T = \arg\max_{Q_1^T} \log p(Q_1^T | O_1^T) \tag{1}$$

By using the Viterbi algorithm, the above maximization can be done efficiently by computing the partial path score $\delta_t(i)$ that explains the observation O_1^t and that $q_t = j$. A recursive relation can be derived given as:

$$\delta_t(j) = \max_i [\delta_{t-1}(i)a_{ij}]\, b_j(o_t) \quad 1 \leq t < T,$$

where $b_j(o_t)$ is the observation probability for observing o_t at state j.

2.2. The Robust State Sequence

From Equation 1, the maximization can be written as the sum of observation likelihoods and transition probabilities

$$\hat{Q}_1^T = \arg\max_{Q_1^T} \underbrace{\sum_t \log p(o_t|q_t)}_{\text{obs. likelihood for } Q_1^T} + \underbrace{\sum_t \log p(q_t|q_{t-1})}_{\text{transition prob}}$$

To find a state sequence that is more robust against corruption, we denote the log likelihood of a path Q_1^T by its best $T-K$ frames, denoted by $\phi_K(\ O_1^T|Q_1^T)$. $\phi_K(\ O_1^T|Q_1^T)$ can be written as a function of the rank-ordered observation likelihoods $o_{t_1}, \ldots, o_{t_T}$ such that

$$p(o_{t_i}|q_{t_i}) \leq p(o_{t_j}|q_{t_j}) \qquad i \leq j.$$

$\phi_K(\ O_1^T|Q_1^T)$ can be written as

$$\begin{aligned}\phi_K(\ O_1^T|Q_1^T) &= \sum_{i=K+1}^{T} \log p(o_{t_i}|q_{t_i}) \\ &= \sum_{t \notin \{t_1,\ldots,t_K\}} \log p(o_t|q_t) \qquad (2)\end{aligned}$$

This is equivalent to skipping of the worst scoring K frames from the computation of the observation likelihood of the path. Thus, ϕ_K can be rewritten as

$$\phi_K(\ O_1^T|Q_1^T) = \max_{t_1,\ldots,t_K} \sum_{t \notin \{t_1,\ldots,t_K\}} \log p(o_t|q_t),$$

where $o_{t_1}, \ldots, o_{t_K}$ are the frames that are ignored or skipped.

In recognition, we search over all possible state sequences. The best robust path, $\hat{Q}_1^T$ is given by,

$$\hat{Q}_1^T(K) = \arg\max_{Q_1^T} \left(\sum_{t=1}^{T} \log p(q_t|q_{t-1}) + \phi_K(O_1^T|Q_1^T) \right) \qquad (3)$$

Notice that in this process, we only ignore contributions of the outlying frames to the total observation likelihood computation but retain contribution of state transitions and the length of the state sequence remains T. This preserves the duration information.

A recursive relationship, similar to the one in Viterbi algorithm can be obtained. At each state, K partial scores is augmented to store scores with 1 to K frames skipped. We denote the partial path score that explains the first t observations and allows k skips as $\delta_t(j,k)$. When computing these scores, on top of finding the best previous state, we need to decide whether the current frame o_t should be skipped. Denoting the partial path score for skipping the current frame as $\delta_t^s(j,k)$.

$$\delta_t^s(j,k) = \max_i a_{ij}\delta_{t-1}(i,k-1).$$

Denoting the partial path score for not skipping the current frame o_t as $\delta_t^n(j,k)$.

$$\delta_t^n(j,k) = \max_i a_{ij}\delta_{t-1}(i,k)b_j(o_t).$$

Combining the above two equations, we have

$$\begin{aligned}&\delta_t(j,k) \\ &\quad = \max(\delta_t^s(j,k), \delta_t^n(j,k)) \\ &\quad = \max_i \left[a_{ij} \max(\delta_{t-1}(i,k-1), \delta_{t-1}(i,k)b_j(o_t))\right]\end{aligned}$$

We called this the frame-skipping Viterbi algorithm (FSVA).

2.3. Missing Feature Interpretation

According to missing feature theory [2], if we know that the t_1-th frame of speech is corrupted, it is best to ignore its contributions to the search process. That is,

$$\hat{Q}_1^T = \arg\max_{Q_1^T} \sum_{t=1}^{T} \log p(q_{t-1}|q_t) + \sum_{t \neq t_1} \log p(o_t|q_t)$$

If we know that a set of K frames,$\{t_1, t_2, ..., t_K\}$ are corrupted, then

$$\hat{Q}_1^T = \arg\max_{Q_1^T} \sum_{t=1}^{T} \log p(q_{t-1}|q_t) + \sum_{t \notin \{t_1,\ldots,t_K\}} \log p(o_t|q_t)$$

In most applications, the exact location of the corrupted frames are unknown. If we know that K frames are corrupted, we can search for the best state sequence over all possible t_j. This will give Equation 3.

3. Implementation of FSVA

In this section, we describe two implementations of FSVA: 1) state-space extension and, 2) topology extension. The first approach aims to implement Equation 4 directly. The second approach aims to implement the proposed algorithm by using existing HMM software by changing the HMM topology.

3.1. State space extension

Similar to the exact N-best algorithm proposed in [3] one can estimate the best K-frame skipped state sequence by augmenting the information stored for each state. Figure 1 shows a two state left-to-right HMM for skipping 1 frame. Each box represents the extended state in which we stores $K+1$ scores where the k-th score is the current best score given that k frames has been ignored/skipped in its history. The original Viterbi partial path score is stored as the 0-th score. During the update, in addition to comparing the N possible incoming arcs, we also need to decide whether this frame should be skipped. Under this modification, the computation will be at most $K+1$ times of the original Viterbi search with the storage $K+1$ times bigger. However, in reality, the computation is only slightly bigger than the original Viterbi because $b_j(o_t)$ needs to be computed only once per state which typically dominates the computation.

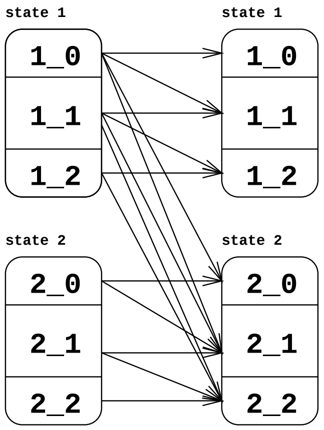

Figure 1: The search rules in FSVA.

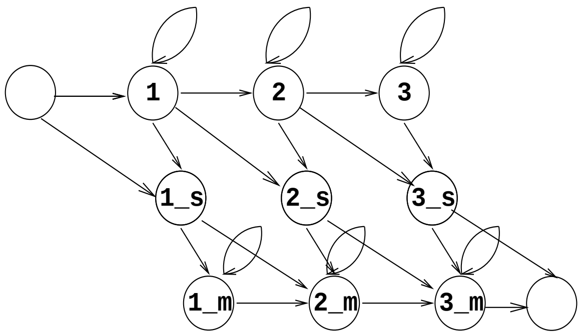

Figure 2: The topology of a frame-skipping HMM.

3.2. Topology extension

An alternative implementation of FSVA approach is to extend the topology of an HMM, using separate states for different skip statuses. The advantage of this approach is that standard Viterbi algorithm can be applied in our problem. This will also allow standard HMM softwares to be used without modifications. Suppose each word is represented by a left-to-right HMM with 3 states and no skipping arcs. To extend this HMM to skip one frame, we extend the HMM topology to include a set of *skip-states* denoted as $1_s, 2_s$, and 3_s) and a set of *skipped states* ($1_m, 2_m$, and 3_m as shown in Figure 2. The *skip-states* are visited whenever a frame is skipped. Ideally, we would like the skip-states to return an observation probability of 1 in effect no contribution to the total. Instead, it can be approximated by a uniform observation distribution that returns a constant log likelihood of c. Since all paths that we consider will include the same number of skips, the introduction of c causes a constant increase in the path likelihood but with no effect in selecting the best path. Notice that there is no self loop at the skip-states. Instead, it is forced to proceed to one of the *skipped states* $1\_m, 2\_m, 3\_m$. The skipped states denote paths that have already skipped one frame. Because of this, a skipped state can only transit to other skipped state. The skipped state's statistical properties, such as observation distribution and transition probabilities, are duplicates of the original states. We will refer this extended model as Frame Skipping HMM (FSHMM).

The FSHMM can be generalized to skip more frames. For a N state HMM, each additional frame skipped will added 2N states. Because we are adding $2N$ states, the FSHMM is not as efficient as the state-space extension approach.

The FSHMM can be extended to handle continuous speech. The extension is quite tedious and we will only outline the basic idea here. Because a sentence involves multiple words, transitions between one word unit and another is necessary. To maintain the skip information across different words, multiple word models representing the same word with different number of skips are needed. Then, the models representing fewer skips can be connected to those representing more skips. This extension, while convenient for testing using standard speech recognition software, is complex and computationally expensive to handle more than three skips.

FSHMM provides us an alternative view for obtaining the robust state sequence. This also allows us to extend the frame-skipping Viterbi algorithm. For example, the skip state can be modified to have an observation distribution that capture transmission noise or corruptions. Instead of skipping a fixed number of frames, skipping probabilities can be added on the transition between the skipped-states.

4. Experiments

In this section, we describe the experiments we performed to evaluate the FSVA. All experiments were performed using the TI-digits corpus. FSVA was evaluated under two sets of experiments; isolated digit recognition (IDR) and continuous digit recognition (CDR).

4.1. Experimental Condition and Baseline Experiments

For both sets of experiments, we used whole-word, six states left-to-right HMMs to represent the digits, 0-9 and "oh". The observation distributions are 32 component Gaussian mixtures that are obtained through EM training and mixture growing procedure as suggested by the HTK handbook [1]. Acoustic features used are the 12 mel-frequency cepstral coefficients and their first and second derivatives and the energy term.

The training and testing sets differ between the isolated digit and continuous digit experiments. For IDR, a total of 12515 digit utterances were used for training and 3576 were used for test. The length of each test utterance is about 1 seconds long. For CDR, a total of 12515 digit utterances were used for training and 12515 were used for test. Each utterance was a digit string with maximal length of 7 digits and up to 3 seconds long.

To test the effectiveness of FSVA in handling frame corruption, we simulate the frame loss phenomenon of the GSM codec. Speech frames of 25 ms are selected randomly and are replaced by Gaussian noise of the same energy level. The advantage of this simulated environment is that it allows us to control the amount of corruption.

config	clean test data	10%
IDR	99.05	88.73
CDR	97.57	79.85

Table 1: Performance of base-line experiment evaluated in different configurations in terms of WER

Table 1 shows the results of the baseline experiment [1] in which decoding was performed using the standard Viterbi algorithm. "clean" denoted the performance when no frame loss was simulated. "10%" denotes the condition that on average, 10% of the frames were dropped and replaced by comfort noise. Here, we see that the performance will be significantly degrade if clean speech data is used for training.

4.2. Evaluation of Frame Skipping Algorithm

We evaluated the proposed approach on both IDR and CDR by varying number of frames skipped from 1 to 10. The recognition results for isolated digit recognition and connected digit recognition on both the clean data and corrupted data are shown in Figures 3 respectively.

We find that FSVA are effective in recovering most of the lost performance due to corruption. The robust algorithm improved IDR's performance from 88.73% to 96.42% when skipping 10 frames, resulting 8.08% of absolute improvement . For CDR, the recognition rate improved from 79.85% to 86.45% when skipping 7 frames. When the robust algorithm is applied to clean data, slight degradations of approximately 0.2% absolute error in IDR and 0.4% in CDR.

Computationally, we observed that the FSVA run almost as fast as the traditional HMM even when skipping 10 frames.

The relative improvement of the continuous digit task is significantly lower than the isolated digit task. We suspect that this is caused by the variation of length between different continuous digit files. We are currently conducting experiments that actually skips a fixed percentage of the frames instead of a fixed number.

[1] The CDR's result is suboptimal. It can be improved by either increasing the number of mixture or increase the length of HMM. This doesn't affect our evaluation of FSVA because we mainly focus on how the algorithm improve recognition in condition that with frame loss.

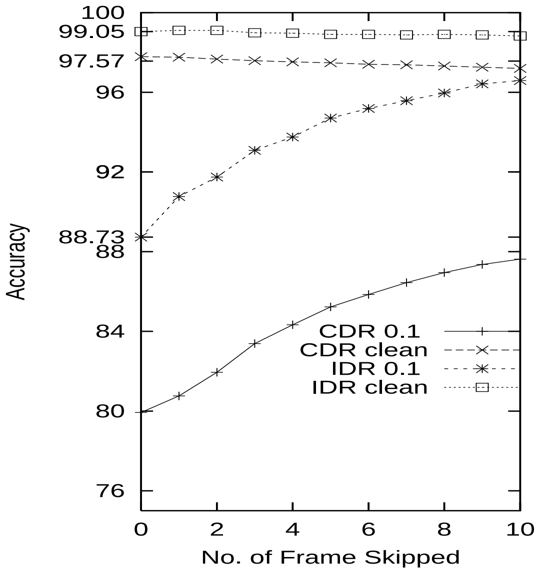

Figure 3: Performance of FSVA on TI-digits task

5. Conclusions

In this paper, we have proposed a robust approach for speech recognition that ignores contributions of the worst performing frames. This algorithm can be interpreted as a special case of missing feature approach where corrupted frames are ignored or as a robust estimation of the score for a state sequence similar to that of using a trimmed mean. We also propose a modification to the Viterbi algorithm that can perform recognition efficiently. Experiments on the TI-digits task show that FSVA improves recognition of corrupted speech utterances significantly. While the approach is motivated for handling packet loss, it may also be applicable to other types of burst noises or corruption.

6. Acknowledgment

This work is partially supported by the Hong Kong RGC under grant number DAG 98/99.EG14 and RGC6049/00E.

7. References

[1] S. Young , J. J. Odell., D. Olasen,V. Valtchev, P. C. Woodland ,*The HTK Book(for HTK Version 2.2)* 1999.

[2] R. Lippmann and B. Carlson, Robust speech recognition with time-varying filtering interruptions and noise, Proceeding of the ASR workshop, 1997.

[3] R. Schwartz and Y. L. Chow, The N-best algorithm: an efficient and exact procedure for finding the N most likely sentence hypotheses, IEEE ICASSP, pp. 81-84, Albuquerque, April 1990.