CONFIDENCE INDEX DYNAMIC TIME WARPING FOR LANGUAGE-INDEPENDENT EMBEDDED SPEECH RECOGNITION

Xianglilan Zhang[⋆†] *Jiping Sun*[‡] *Zhigang Luo* [⋆] *Ming Li*[†]

[⋆]School of Computer, National University of Defense Technology; Changsha, China
[†]David R. Cheriton School of Computer Science, University of Waterloo; Waterloo, Ontario, Canada
[‡]Voice Enabling Systems Technology; Waterloo, Ontario, Canada

ABSTRACT

Language-independent embedded speech recognition is a necessary and important application. Considering personal privacy, collection difficulty of all the reference words, and limited storage space of mobile devices, language-independent (LI) embedded speech recognition should be classified into lightweight speaker-dependent (SD) cases. Dynamic time warping (DTW) is the state-of-the-art algorithm for small foot-print SD automatic speech recognition. To decrease the high computational complexity of DTW, and to avoid constraints-induced coarse approximation and inaccuracy problems, we introduce a novel confidence index dynamic time warping (CIDTW) approach. CIDTW defines a new cost function, called the confidence index cost function (CICF), to measure the similarity between merged speech training and testing data, while follows the same DTW process. With extensive experiments on three representative SD datasets, CIDTW achieves better accuracy and overall six times faster speeds compared with DTW.

Index Terms— language-independent and lightweight speaker-dependent speech recognition, confidence index DTW, confidence index cost function

1. INTRODUCTION

With the increase in connections across countries, the contact information in mobile devices could include names of people from many different countries. Therefore, language-independent (LI) embedded speech recognition (SR) is critical. Most of the modern embedded SR applications of mobile devices are speaker-independent (SI). SI applications are based on Hidden Markov Model (HMM) [1], the accuracy of which is governed by the amount of training data. Therefore, SI applications have to use a large SR server to store the training data. When doing SR, all of the information in personal mobile devices has to be uploaded to the remote SR server, which has an inherent risk of loss of personal information. Additionally, lack of training data in non-English languages is the most important reason that these applications can not achieve a good accuracy when doing non-English speech recognition. Due to personal privacy consideration, and excessive time, storage and cost factors associated with the collection of multi-language training data, we classify the LI application as speaker-dependent (SD) application.

Considering that storage space of mobile devices and personal information are limited, our goal is to develop lightweight SD approach by using only one sample for each word as training data. This approach can also be applied to other similar real-time applications such as menu-driven recognition, and voice control on vehicles and robotics. While HMM needs sophisticated implementation of large-scale software and lots of training data [2], DTW aims at small-scale embedded systems (i.e., cell phones, mobile applications) with its simplicity in hardware implementation [3]. Thus, we choose DTW for this work.

However, the time complexity of DTW is a limitation for large databases [4, 5] and real-time applications. The basic idea of DTW speed up is to shorten the lengths of either or both of the two processed speech signals. Many variations have been proposed for accelerating DTW computing process [6]. Be it lower bounding measure [7], global constraint region usage [8], multi-scale DTW [9], or any other combination of the first two methods [10], they are all based on constraint algorithms in iterative fashion [11]. These algorithms tend to have coarse approximation [13] and inaccuracy [9] problems.

To avoid the above shortcomings of constraint algorithms, we hereby propose a novel confidence index dynamic time warping (CIDTW) method. We define a new cost function, called confidence index cost function (CICF), to measure the similarity between merged speech training data and testing data, and use the general DTW process to find similarities

Xianglilan Zhang is currently a visiting Ph.D. student at David R. Cheriton School of Computer Science of University of Waterloo (from October 2010 to October 2013).

Ming Li is the corresponding author.

This work has been supported by Chinese Scholarship Council, partially supported by IDRC Research Chair in Information Technology, NSERC Discovery Grant OGP0046506, the Canada Research Chair program, a CFI Infrastructure grant, an NSERC Collaborative grant, Ontarios Premiers Discovery Award, and the Killam Prize.

between them. Unlike most of the current DTW variations, CIDTW only change the cost function in DTW process and does not have any constraints. On the other hand, its simplicity guarantees flexibility, efficiency, and ease of implementation. Therefore, CIDTW is very suitable for real-time applications with limited storage space and small vocabulary.

Three datasets, Chinese names, Chinese and English names, and Chinese and English names along with Chinese address terms, have been tested by using DTW and CIDTW. The results show that CIDTW achieves better accuracy and overall six times faster speeds compared with DTW.

2. DYNAMIC TIME WARPING ALGORITHM

Consider two input speech signals, L with length m and S with length n, vary in time. The distance of point i in L and point j in S is given by formula 1 :

$$DTW[i,j] = Cost[i,j] + min \begin{cases} DTW[i-1,j] \\ DTW[i,j-1] \\ DTW[i-1,j-1] \end{cases} \quad (1)$$

where $Cost[i,j]$ is the Euclidean distance between point i and point j.

The $DTW[m,n]$ represents the similarity of L and S. The smaller this value is, the closer the two speech signals are.

3. PROPOSED CONFIDENCE INDEX DYNAMIC TIME WARPING METHOD

Our CIDTW method can process spectrograms or Mel Frequency Cepstral Coefficients (MFCC) acoustic features of audio files. In this paper, we use MFCC as input to the CIDTW method. In the remainder of this paper, we will specify the input speech file format as MFCC. To use CIDTW method for speech recognition, we record each word for only one time as training data. This method consists of three steps:

1. Merge adjacent and similar time frames of training and testing MFCC.

2. Calculate the CICF between merged training and testing MFCC.

3. Apply the CICF to general DTW process.

The following subsections give a detailed description of each step.

3.1. Time frames merging of MFCC

The first step of CIDTW method is to merge the adjacent similar time frames into one new MFCC. The rationale behind the time frames merging is that some adjacent time frames, particularly those from the same phoneme, are very similar [2]. Thus, the length of any MFCC could be decreased by merging adjacent and similar time frames. Such merging can significantly shorten the alignment time of two MFCCs.

In our method, we merge frames by replacing them with their mean vector. As shown in algorithm 1, the adjacent time frames chosen to be merged together depend on their Euclidean distances, $d(F_i, F_{i+1})$. In this paper, i represents the i^{th} time frame of a certain MFCC. Assuming that an MFCC has n time frames, the total number of Euclidean distances of adjacent time frames is $n-1$.

Algorithm 1 Time Frames Merging

Require: Time frames F_1, F_2, ..., F_n; Distance function $d(i,j)$; Merge ratio β;

1: Compute distance value between each adjacent frames $d(F_i, F_{i+1}), 1 \le i \le n-1$
2: Set merge threshold q as $\lceil (n-1) * \beta \rceil$
3: Find the q^{th} smallest distance value, $d_{(q)}$
4: Merge consecutive time frames between i to $i+k$ that satisfy $d(F_j, F_{j+1}) \le d_{(q)}$,$j \in [i,k]$;
$d(F_{i-1}, F_i) > d_{(q)}$;
$d(F_k, F_{k+1}) > d_{(q)}$
5: **return** the merged time frames of an MFCC

3.2. Confidence index cost function of merged training and testing MFCC

We call the training MFCC after merging time frames as **model template**, the merged time frame in training or testing MFCC as **node**, the average distance between all aligned pairs of nodes in all model templates as ***D.all***, and the average distance between node i of one model template and its 'closest' aligned nodes in all model templates as ***D.i***. We do DTW alignment between all model templates and get these statistics – $D.all$ and $D.i$.

Consider that if a node aligns with certain nodes more closely, the closest scores will affect it much more than using the average of all alignments, thus a much lower weight should be assigned to it. By using such a partially average distance between all the 'closest' nodes, $D.i$, the alignment system becomes more discriminative. Suppose that there are m model templates, and node i in model template j has t aligned nodes, where $\alpha\%$ aligned nodes are the 'closest' nodes to node i. Then the $D.i$ of node i is defined by formula 2:

$$D.i = (\sum_{k=1}^{P} AscDist[i, i_k])/P \quad (2)$$

where

$$P = t * \alpha\%$$

AscDist is the ascending order Euclidean distance list of node i and its 'closest' aligned node i_k.

Suppose that a node is represented by a n-vector and a model template has N nodes. The definition of confidence index cost function (CICF) between node i in a model template and node j in a testing MFCC is represented by formula 3:

$$CICF[i,j] = \frac{c.i}{\sum_{l=1}^{N} c.l} * \sqrt{\sum_{k=1}^{n}(i_k - j_k)^2} \qquad (3)$$

where

$$c.i = \begin{cases} 1.0;\ if\ D.i \geq\ D.all \\ D.i/D.all;\ else \end{cases}$$

Here, $c.i$ represents the confidence index (CI) of node i in model templates. If one node's average distance from its aligned nodes is shorter than the global average, that node is very similar to nodes in other model templates. Then it is given lower confidence as model element. Therefore, its CI is much smaller than 1. On the other hand, if a node's average distance from its aligned nodes is longer than the global average, that node is quite different from nodes in other templates. Then it is given higher confidence as model element. Hence its CI is approaching 1. By applying the CI factor of node in model templates to cost function, CICF could capture the most important acoustic features of a speech signal.

3.3. CICF applied to DTW process

For testing the similarity between node i in model template and node j in merged testing MFCC, we replace the cost function in general DTW with CICF, that is:

$$CIDTW[i,j] = CICF[i,j] + min \begin{cases} CIDTW[i-1,j] \\ CIDTW[i,j-1] \\ CIDTW[i-1,j-1] \end{cases} \qquad (4)$$

Overall, the merging of training and testing MFCC time frames largely increases the training-testing alignment speed, and the CICF guarantees accurate speech recognition.

4. RESULTS

4.1. Data Preparation

We use the Audacity software to manually record a total of 45 different names over 1.9 hours in a quiet environment. The recording settings are 8k Hz, mono channel, 16 bits PCM. Each name is repeated 10 times. At first, 10 Chinese names are recorded. In order to test whether our method is compatible with multiple languages, we introduce some English names in the next 20 names. Since our goal is to enhance name recognition accuracy, especially for Chinese words, we introduce 15 different Chinese terms to address 'father', 'mother', 'son', 'daughter', 'grandparents', etc.[1] We use these 45 names to perform three experiments. The first experiment is to test the first 10 Chinese names (dataset1), the second one is to test dataset2 consisting of the first 10 Chinese names and the next 20 Chinese and English names, the last one is to test dataset3 including dataset2 and Chinese address terms.

Referring to Chapter 3 of HTK manual [14], the HCopy function in HTK is used to convert the audio files of .wav format into .mfc files. In HTK, the frame period is 25msec, the fast Fourier transform (FFT) uses a Hamming window, the signal has first order pre-emphasis applied to it by using a coefficient of 0.97, the filterbank has 26 channels, and the output is 13 MFCC coefficients. Since the input of our CIDTW method is text format files, the HList function in HTK is used to convert these binary .mfc files into text format.

4.2. CIDTW VS. DTW

We use traditional DTW and our CIDTW to test the datasets 1, 2, and 3. Cross validation approach is applied to compare the DTW method with CIDTW method. Since each name is recorded ten times, one audio file of a certain name is randomly picked out as speech training data, the other nine files are testing data. Therefore, ten cross validation experiments have been performed on each dataset. Every cross validation experiment has unique speech training data, which is completely different than the training data in other nine experiments. The number of speech training data vs. testing data in each dataset is: 10 vs. 90, 30 vs. 270, and 45 vs. 405 respectively.

We set the merge ratio in section 3.1 as the golden ratio (1:1.618) since this is preferred in previous works of salient data selection [15, 16]. According to section 3.2, the CIDTW could choose the percentage ($\alpha\%$) of the number of aligned nodes as closest ones for a node in model templates. We first analyze the impact of different percentage settings on the average accuracy of ten cross validation groups in each dataset. As shown in Figure 1, for all of the datasets, the CIDTW method achieves the best recognition result when choosing a 10% or 20% closest aligned node number for each node. Due to page limitation, we will only show the CIDTW results of choosing 20% closest aligned node number. For 10% and 15%, our CIDTW also outperforms DTW.

The results of DTW and CIDTW in cross validation experiments are listed in Table 1. The accuracy of the highlighted groups of CIDTW and DTW reaches 100%. The number of '100% accuracy' groups in CIDTW is more than DTW.

As shown in Table 2, the average accuracy of CIDTW is better than DTW, and the average speed of CIDTW is sig-

[1]The name list and .wav format source audio files could be found at $https://www.dropbox.com/sh/8b3hhf0x0ao9bh6/1Huqkn_wrM/NameList\%26SourceAudioFiles$.

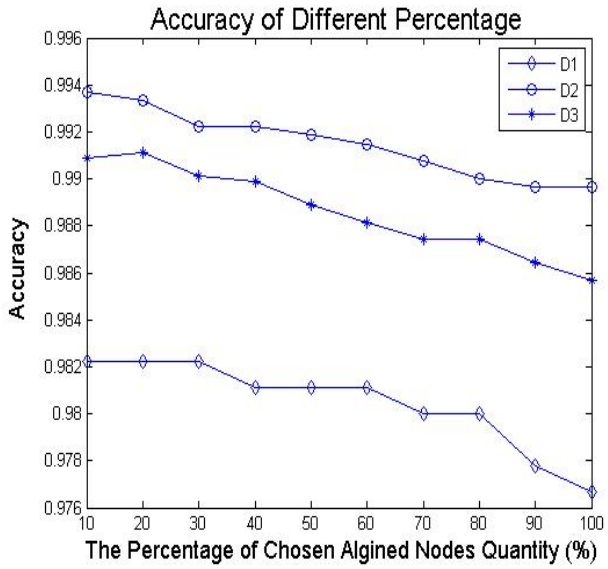

Fig. 1: Accuracy of choosing different number of aligned nodes. Here, the accuracy is the average of the whole ten cross validation groups. The diamond-line represents dataset 1, circle-line represents dataset2, and star-line represents dataset3.

Table 1: Accuracy (%) of DTW and CIDTW for three datasets. 'D' is for dataset, and 'G' is for cross validation group.

	\	G1	G2	G3	G4	G5
D1	DTW	98.89	93.33	94.44	91.11	96.67
	CIDTW	94.44	100	95.56	98.89	96.67
D2	DTW	99.63	97.78	98.15	97.04	98.89
	CIDTW	98.52	100	97.41	99.63	98.89
D3	DTW	98.52	98.52	98.52	97.53	98.27
	CIDTW	98.27	100	97.53	99.01	99.26
	\	G6	G7	G8	G9	G10
D1	DTW	96.67	96.67	91.11	95.56	100
	CIDTW	100	100	98.89	97.78	100
D2	DTW	98.89	98.89	97.04	98.52	100
	CIDTW	100	100	100	98.89	100
D3	DTW	99.26	99.01	97.53	99.01	100
	CIDTW	99.51	99.51	99.51	99.01	99.51

nificantly faster than the DTW, that is, CIDTW is 6.15 times faster than DTW for Dataset1, 6.11 times faster for Dataset2, and 5.74 times faster for Dataset3 .

Table 2: Overall Recognition Accuracy and CPU Time.

Dataset	Algorithm	Accuracy (%)	CPU Time (s)
1	DTW	95.44	378.18
	CIDTW	98.22	61.51
2	DTW	98.48	3117.92
	CIDTW	99.33	509.90
3	DTW	98.62	5681.62
	CIDTW	99.11	989.43

5. CONCLUSIONS AND FUTURE WORK

In this paper, we introduce a novel CIDTW approach to provide efficient lightweight SD-SR service for real-time applications with small vocabulary and limited storage space, such as offline voice dialing on mobile devices, menu-driven recognition, and voice control on vehicles and robotics. Unlike most of the current DTW variations, CIDTW follows the general DTW process while only changing original cost function into CICF, hence no constraints need to be specified. By testing on three representative datasets, CIDTW demonstrates better accuracy and faster speed compared with DTW. Its simplicity and light computational complexity makes it very suitable for those small-footprint and real-time applications mentioned above.

We hope to develop simpler and more efficient methods; we are in the process of improving our CIDTW algorithm and make it available on mobile devices, i.e. cell phones. Specifically, our upcoming work is to recognize contact names in continuous speech.

6. REFERENCES

[1] N. Y. Talking, "Powerful New Language Tools Leverage AI", *IEEE Intelligent Systems*, vol. 27, no. 2, pp. 2-7, March-April 2012.

[2] J. Sun, Y. Sun, K. Abida and F. Karray, "A novel template matching approach to speaker-independent arabic spoken digit recognition", *AIS 2012*, vol. 7326, pp. 192-199, 2012.

[3] S. V. Chapaneri, "Spoken digits recognition using weighted MFCC and improved features for dynamic time warping", *Int. Journal of Computer Application*, vol. 40, no. 3, pp. 6-12, 2012.

[4] D. J. Berndt and J. Clifford, "Using dyanmic time warping to find patterns in time series", *AAAI Workshop on*

Knowledge Discovery in Databases, 31st July - 1st Aug. 1994, pp. 359-370.

[5] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, E. Keogh, "Searching and mining trillions of time series subsequences under dynamic time warping", *KDD'12*, Aug. 2012, pp. 262-270.

[6] M. M*ü*ller, "Information retrieval for music and motion", *Springer-Verlag Berlin Heidelberg*, 2007.

[7] S. W. Kim, S. Park and W.W. Chu, "An index-based approach for similarity search supporting time warping in large sequence databases", *Data Engineering, 2001 Proc. 17 th Int. Conf. on* April 2001, pp. 607-614.

[8] Y. Zhu and D. Shasha, "Warping indexes with envelope transforms for query by humming", *SIGMOD 2003*, June 2003, pp. 181-192.

[9] M. M*ü*ller, H. Mattes and F. Kurth, "An efficient multiscale approach to audio synchronization", *Proc. ISMIR*, Oct. 2006, pp. 192-197,

[10] Y. Sakurai, M. Yoshikawa and C. Faloutsos, "FTW: fast similarity search under the time warping distance", *PODS 2005*, June 2005, pp.326-337.

[11] P. Papapetrou, V. Athitsos, M. Potamias, G. Kollios and D. Gunopulos, "Embedding-based subsequence matching in time-series databases", *ACM Trans. on Database Systems*, vol. 36, no. 3, pp. 17:1-17:39, 2011.

[12] F. O. Karray and C. D. Silva, "Soft computing and intelligent systems design: theory, tools and applications", *Pearson Education Limited*, 2004.

[13] E. Keogh, "exact indexing of dynamic time warping", *Proc. of the 28th VLDB Conf.* Aug. 2002, pp. 406-417.

[14] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, P. Woodland, "The HTK book (for HTK version 3.4)", *Cambridge University Engineering Department*, 2006.

[15] M. Livio, "The golden ratio: the story of phi, the world's most astonishing number", *Broadway Books*, 2002.

[16] A. Lu, R. Maciejewski, D.S. Ebert, "Volume composition using eye tracking data", *In Proceedings of EuruVis '06*, 2006, pp. 655-662.