Dynamic Programming Algorithms in Speech Recognition

Titus Felix FURTUNĂ
Academy of Economic Studies, Bucharest
titus@ase.ro

In a system of speech recognition containing words, the recognition requires the comparison between the entry signal of the word and the various words of the dictionary. The problem can be solved efficiently by a dynamic comparison algorithm whose goal is to put in optimal correspondence the temporal scales of the two words. An algorithm of this type is Dynamic Time Warping. This paper presents two alternatives for implementation of the algorithm designed for recognition of the isolated words.
***Keywords**: dynamic programming, speech recognition, word detection.*

Introduction

Studies in speaking recognition field, as well as studies in other fields, follow two trends: fundamental research whose goal is to devise and test new methods, algorithms and concepts in a non-commercial manner and applied research whose goal is to improve existing methods, following specific criteria.
This article deals with isolated words recognition within applied research trend.
The fundamental research aims at medium and especially long term benefits, while applied research aims at quick performances improvements of existing methods or extending their use in domains where they have less been used so far.
Improving performances in voice recognition can be done taking into account the following criteria:

- dimension of recognizable vocabulary;
- spontaneous ness degree of speaking to be recognized
- dependence/independence on the speaker;
- time to put in motion the system
- system accommodating time at new speakers;
- decision and recognition time;
- recognition rate (expressed by word or by sentence).

Today's vocal recognition systems are based on the general principles of forms' recognition [3][7]. The methods and algorithms that have been used so far can be divided into four large classes:

- Discriminant Analysis Methods based on Bayesian discrimination;
- Hidden Markov Models;
- Dynamic Programming –Dynamic Time Warping algorithm (DTW) [8];
- Neuronal Networks.

This article presents an example/alternative of dynamic programming DTW algorithm implementation in speech recognition.

1. Dynamic Time Warping Algorithm (DTW)

Dynamic Time Warping algorithm (DTW) [Sakoe , H. & S. Chiba-8] is an algorithm that calculates an optimal warping path between two time series. The algorithm calculates both warping path values between the two series and the distance between them.
Suppose we have two numerical sequences (a_1,a_2, ..., a_n) and (b_1, b_2, ..., b_m). As we can see, the length of the two sequences can be different. The algorithm starts with local distances calculation between the elements of the two sequences using different types of distances. The most frequent used method for distance calculation is the absolute distance between the values of the two elements (Euclidian distance). That results in a matrix of distances having *n* lines and *m* columns of general term:

$$d_{ij} = \left| a_i - b_j \right|,\ i = \overline{1,n},\ j = \overline{1,m}.$$

Starting with local distances matrix, then the minimal distance matrix between sequences is determined using a dynamic programming algorithm and the following optimization criterion:

$$a_{ij} = d_{ij} + \min(a_{i-1,j-1}, a_{i-1,j}, a_{i,j-1}),$$

where a_{ij} is the minimal distance between the subsequences $(a_1, a_2, ..., a_i)$ and $(b_1, b_2, ..., b_j)$.
A warping path is a path through minimal distance matrix from a_{11} element to a_{nm} element consisting of those a_{ij} elements that have formed the a_{nm} distance.
The global warp cost of the two sequences is defined as shown below:

$$GC = \frac{1}{p}\sum_{i=1}^{p} w_i ,$$

where w_i are those elements that belong to warping path, and p is the number of them.
The calculations made for two short sequences are shown in figure 1 including the highlight of the warping path.

	-2	**10**	**-10**	**15**	**-13**	**20**	**-5**	**14**	**2**
3	5	12	25	37	53	70	78	89	90
-13	16	28	15	43	37	70	78	105	104
14	32	20	39	16	43	43	62	62	74
-7	37	37	23	38	22	49	45	66	71
9	48	38	42	29	44	33	47	50	57
-2	48	50	46	46	40	55	36	52	**54**

Fig.1. The warping path

There are three conditions imposed on DTW algorithm that ensure them a quick convergence:

1. monotony – the path never returns, that means that both indices i and j used for crossing through sequences never decrease.
2. continuity – the path advances gradually, step by step; indices i and j increase by maximum 1 unit on a step.
3. boundary –the path starts in left-down corner and ends in right-up corner.

An example of warping path implementation using Java programming language is shown below:
Because optimal principle in dynamic programming is applied using "backward" technique, identifying the warp path uses a certain type of dynamic structure called "stack". Like any dynamic programming algorithm, the DTW one has a polynomial complexity.
When sequences have a very large number of elements, at least two inconveniences show up:

```
public static void dtw(double a[],double b[],double dw[][], Stack<Double> w){
        // a,b  - the sequences, dw   - the minimal distances matrix
        // w    - the warping path
        int n=a.length,m=b.length;
        double d[][]=new double[n][m]; // the euclidian distances matrix
        for(int i=0;i<n;i++)
            for(int j=0;j<m;j++)d[i][j]=Math.abs(a[i]-b[j]);
        // determinate of minimal distance
        dw[0][0]=d[0][0];
        for(int i=1;i<n;i++)dw[i][0]=d[i][0]+dw[i-1][0];
        for(int j=1;j<m;j++)dw[0][j]=d[0][j]+dw[0][j-1];
        for(int i=1;i<n;i++)
            for(int j=1;j<m;j++)
                if(dw[i-1][j-1]<=dw[i-1][j])
                    if(dw[i-1][j-1]<=dw[i][j-1])dw[i][j]=d[i][j]+dw[i-1][j-1];
                    else dw[i][j]=d[i][j]+dw[i][j-1];
                else
                    if(dw[i-1][j]<=dw[i][j-1])dw[i][j]=d[i][j]+dw[i-1][j];
                    else dw[i][j]=d[i][j]+dw[i][j-1];
        int i=n-1,j=m-1;
        double element=dw[i][j];
        // determinate of warping path
        w.push(new Double(dw[i][j]));
        do{
            if(i>0&&j>0)
                if(dw[i-1][j-1]<=dw[i-1][j])
                    if(dw[i-1][j-1]<=dw[i][j-1]){i--;j--;} else j--;
                else
                    if(dw[i-1][j]<=dw[i][j-1])i--; else j--;
            else if(i==0)j--; else i--;
            w.push(new Double(dw[i][j]));
       }
        while(i!=0||j!=0);
    }
```

- memorizing large matrices of numbers;
- performing large numbers of distances calculations.

There is an improvement in standard DTW algorithm that sorts out the two problems named above: FastDTW (Fast Dynamic Time Warping) [Stan Salvador, Philip Chan - 6]. The proposed solution consists of dividing distances matrices into 2,4,8.16, etc matrices of smaller dimensions through a repeatedly process of splitting in two the input sequences. This way, the distance calculations are performed only on these smaller matrices and the warp path is then put together by merging the warp paths calculated for smaller matrices. From algorithmic point of view, the proposed solution is based on "Divide et Impera" method.

2. Using DTW Algorithm in Speech Recognition

Vocal Signal Analysis. Sound travels through the environment as a longitudinal wave with a speed that depends on the environment density. The easiest way to represent sounds is a sinusoidal graphic. The graphic presents variation of air pressure depending on time

The shape of the sound wave depends on three factors: *amplitude*, *frequency* and *phase*.

The amplitude is the displacement of the sinusoidal graph above and below temporal axis ($y = 0$) and it corresponds to the energy the sound wave is loaded with. Amplitude measurement can be performed using pressure units (decibels DB), which measure the amplitude following a logarithmic function as regards a standard sound. Measuring amplitude using decibels is important in practice because it is a direct representation of how the sound volume is perceived by people.

The frequency is the number of cycles the sinusoid makes every second. A cycle consists of an oscillation starting with the medium line, then it reaches the maximum, then it reaches the minimum and then back to medium line. The frequency is measured in cycles per second or Hertz (Hz). The reverse of frequency is called the period. It is the time needed for the sound wave to complete a cycle.

The last factor is the phase. It measures the position from the beginning of the sinusoidal curve. The phase cannot be perceived by human senses, but humans can detect the relative phase changes between the two signals. In fact, this is the way human sensorial system perceives a sound location, counting on different phases perceived by the ears.

In order to take apart a wave sound into sinusoidal curves we use Fourier's theorem. That says that any periodical complex wave can be taken apart into sinusoidal curves having different frequencies, amplitudes and phases. This process is called Fourier analysis and it results in a set of amplitudes, phases and frequencies for each sinusoidal wave component. Adding these sinusoidal curves together we get the original sound wave. A point of frequency or phase together with the amplitude is called a spectrum. Any periodical signal shows a recursive time model which corresponds to the first vibration rate of the signal called fundamental frequency. This can be measured from speech sound verifying signal oscillation period around 0 axis. A spectrum shows the frequency of a short sequence in a sound, and if we want to analyze its evolution varying with time, we need a way to show it. This is called a spectrogram. A spectrogram is a diagram in two dimensions (frequency and time) in which the color of the points (dark-strong, light-weak) determines the amplitude intensity. This has a major role in voice recognition, and a human expert could reveal many details only by looking at a sound spectrogram.

Word Detection. Today's detection techniques can accurately identify the starting and ending point of a spoken word within an audio stream, based on processing signals varying with time. They evaluate the energy and average magnitude in a short time unit, and also calculate the average zero-crossing rate.

Establishing the starting and ending point is a simple problem if the audio recording is performed in ideal conditions. In this case the ratio signal-noise is large because it's easy to

determine the location within the stream that contains a valid signal by analyzing the samples. In real conditions things are not so simple, the background noise has a significant intensity and can disturb the isolation process of the word within the stream.

The best detection algorithm for isolated words recognition is Rabiner-Lamel Algorithm. If we consider a signal-window $\{s_1, s_2, ..., s_n\}$ where n is the number of samples of the window and s_i , $i = 1,n$, is the numerical expression of the samples, the energy associated with the signal-window is:

$$E(n) = \frac{1}{n}\sum_{i=1}^{n} s_i{}^2.$$

The average zero-crossing rate is:

$$ZCR(n) = \sum_{i=1}^{n-1} sign(s_i) \cdot sign(s_{i+1}),$$

where $sign(s_i) = \begin{cases} 1 & \text{if } s_i > 0 \\ 0 & \text{if } s_i < 0 \end{cases}$.

The method uses three numerical levels: two for energy (superior, inferior) and one for the average zero-crossing rate. The point starting from which the energy overrides the superior level and the rate of positive and negative values doesn't override the established level is considered the starting point of a voice area (non silence). Searching for the first point of this kind is performed by crossing the window from start to end and it determines the first voice area within the window. The reverse crossing, from end to start, allows identification of the ending point of the last voice area. Identification of the inside silence areas can be done by crossing the window between these two points. The start of a silence area is the point from which the energy decreases below the value of the inferior level. Notice the removing of the silence area before and after pronouncing a word over the microphone in the figure 2 as shown below:

Words Identification Using DTW Algorithm

Words identification can be performed by straight comparison of the numeric forms of the signals or by signals spectrogram comparison. The comparison process in both cases must compensate for both the different length of the sequences and non-linear nature of the sound. The DTW Algorithm succeeds in sorting out these problems by finding the warp path corresponding to the optimal distances between two series of different lengths.

There are some particularities when the algorithm is applied to the two cases:

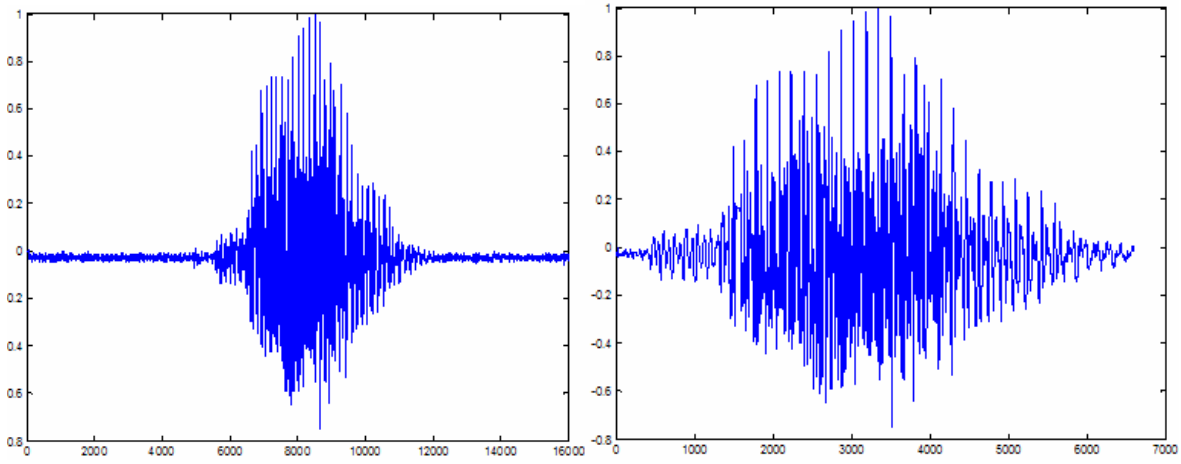

Fig.2. Vocal signal for the word "nouă"

1. Straight comparison of the numeric forms or the signals. In this case, for each numeric sequence, a new sequence is created, sequence whose dimensions are much smaller. The algorithm deals with these sequences. The numeric sequence can have some thousand numeric values, while a subsequence can have some hundred. Decreasing the number of numeric values can be performed by removing those ones between the extreme

points. This process of reducing the length of the numeric sequence must not alter its form. Apparently, the process leads to a decrease in recognizing precision. However, taking into account an increase in speed, the precision is, in fact, increased by enlarging the number of words in the dictionary.

2. Signals spectrogram representations and applying the DTW algorithm for comparison of two spectrograms. The method consists of splitting the numeric signal in a number of "windows" (intervals) which will overlap. For each window, real number intervals, (sound frequencies) the Quick Fourier's transform… will be calculated and it will be stored in a matrix: the sound spectrogram. The parameters will be the same for all calculation operations of the: the window length, Fourier's transform length, the overlap length for two successive windows. The Fourier's transform is symmetrical related to the center and the complex numbers from the second half are the conjugated complex number of the symmetrical numbers from the first half. Due to this fact, only values from the first half can be retain, so that the spectrogram will be a complex numbers matrix, its number of lines equaling half of Fourier's transform length and its number of columns depending on the sound length. The DTW will be applied on a real number matrix resulted from conjugating the spectrogram values, matrix called energies matrix.

Conclusions

DTW Algorithm is very useful for isolated words recognition in a limited dictionary. For a fluent speech recognition, Hidden Markov Chains are used. Using dynamic programming ensures a polynomial complexity to the algorithm: $O(n^2v)$, where n is sequences' lengths and v is the number of words in the dictionary.

There are some weaknesses of the DTW. First, the $O(n^2v)$ complexity may not be satisfactory for a larger dictionary which could ensure an increase in the success rate of the recognition process. Secondly, it is difficult to evaluate two elements from two different sequences, taking into account that there are many channels having distinct features.

However, DTW remains an easy-to-implement algorithm, open to improvements, very appropriate for applications that need simple words recognition: telephones, car computers, security systems, etc.

References

[1] Benoit Legrand, C.S. Chang, S.H. Ong, Soek-Ying Neo, Nallasivam Palanisamy, *Chromosome classification using dynamic time warping*, ScienceDirect Pattern Recognition Letters 29 (2008) 215–222

[2] Cory Myers, Lawrence R. Rabiner, Aaron E. Rosenberg, *Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition*, Ieee Transactions On Acoustics, Speech, And Signal Processing, Vol. Assp-28, No. 6, December 1980

[3] F. Jelinek. "Continuous Speech Recognition by Statisical Methods." IEEE Proceedings 64:4(1976): 532-556

[4] Rabiner, L. R., A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proc. of IEEE, Feb. 1989

[5] Rabiner, L. R., Schafer, R.W., Digital Processing of Speech Signals, *Prentice Hall,* 1978.

[6] Stan Salvador, Chan, *FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space*, **IEEE** Transactions on Biomedical. Engineering, vol. 43, no. 4

[7] Young, S., A Review of Large-Vocabulary Continuous Speech Recognition, IEEE Signal Processing Magazine, pp. 45-57, Sep. 1996

[8] Sakoe, H. & S. Chiba. (1978) Dynamic programming algorithm optimization for spoken word recognition. IEEE, Trans. Acoustics, Speech, and Signal Proc., Vol. ASSP-26.

[9] Furtună, F., Dârdală, M., *Using Discriminant Analisys in Speech Recognition*, The Proceedings Of The Fourth National Conference Humman Computer Interaction Rochi 2007, Universitatea Ovidius Constanţa, 2007, MatrixRom, Bucharest, 2007

[10] *** * *,** Speech Separation by Humans and Machines, *Kluwer Academic Publishers*, 2005