Reduction of Computation Time in Pattern Matching for Speech Recognition

Munshi Yadav
Department of Information Technology,
Guru Tegh Bahadur Institute of Technology,
New Delhi-110064.

Afshar Alam
Department of Computer Science,
Jamia Hamdard,
Delhi-110062

ABSTRACT
Speech recognition is playing a major role in today's daily life. Dynamic Time Warping (DTW) algorithm has been used in different application for the pattern matching, where the sample and stored reference data size is not equal due to time invariant or due to speed. DTW has been implemented and tested by various ways by different researchers for improving the efficiency of the algorithm. There are challenges of accuracy within reasonable time and cost of memory. Various algorithms are available for efficient computing in the sense of time and space. It has been found that generally accuracy and response time is not linear in nature. So there is tradeoff between accuracy and response time. This paper discuss a method which gives the improvement in response time as compared to exiting method in automatic speech recognition by machine in speaker dependent for isolated spoken word.

GENERAL TERMS
Pattern Recognition, Speech Recognition.

KEYWORDS
DTW algorithm, High Performance Computing.

1. INTRODUCTION
The process of dynamic time warping algorithm is important and used in many speech processing applications like isolated speech recognition for speaker dependent system, speaker verification and speaker identification. Parallel realization of DTW algorithm has been implemented in parallel by two techniques namely using linear array and circular array op processing elements(Gregory N. Stainhaour and George Carayannis, 1990)[1][11].

A problem in speech recognition is to find out the exact portion so as to match easily in the pattern matching [2][13]. It's required the start and end point detection as well as the noise of surrounding and background.

In automatic speech recognition for isolated spoken word either whole word matching will performed or Sub word matching will perform. Since the storage data will be of small size in sub word it performed very fast. (Svendsen et al., 1989)[3][10][12].

There is various way of implementation of DTW algorithm in many scientific and engineering applications. DTW algorithm is introduced by Sakoe and Chiba in 1978[4] first time. Further DTW algorithm is implemented as symmetric DTW, asymmetric DTW, Classical DTW (CDTW), Derivative DTW (DDTW) and branch and bound DTW (BnB-DTW)[5].

The paper is organized in six sections. In section (2) approximation algorithm has been explained. In section (3) branch and bound, in section (4) parallel warping, in section (5) pattern matching and in section (6) high performance computing, proposed algorithm and analysis of time complexity have been explained subsequently.

2. APPROXIMATION ALGORITHM
There are number of problems which are not easy to solve, out of which some problems are polynomial time solvable and some problems are NP complete in nature. If a problem is NP Hard then chance of solving it in polynomial time is very-very less but it does not signify that it cannot be solved. There are various methods to solve the NP complete problem but chance of getting solution in exponential time or super polynomial. We may be able to separate some part of the problem and can be solved in polynomial time. In some situation it is possible to get nearby solution which is also up to an acceptance. This nearby solution is called an approximation algorithm. It is possible to solve some NP complete problem by polynomial time using approximation algorithm, whereas some NP complete problem may be solved with fully polynomial time approximation algorithm.

Optimization problem where each possible solution has some positive cost, and we wish to find a near optimal solution. Depending on the problem an optimal solution may be defined as one with maximum possible cost or one with minimum possible cost. That is the problem may be either a maximization or minimization problem. There is a ratio defined between the cost of solution and the cost of optimum solution [5][6].

In speech recognition we try to get the solution as minimum dissimilarity for maximum elements or the maximum number of nodes in the features taken by feature extraction for a particular spoken word.

3. BRANCH AND BOUND
Branch and bound technique are useful for solving optimization, problems where global optimization is required with the help of local optimization for example Travelling Salesman Problem, 0/1 Knapsack Problem, Hamiltonian path problem etc. These problems have been solved with branch and bound technique where there is needs to define heuristic function or target function or objective function.

A point in the path of searching solution which satisfied all the constraints defined for that particular problem is call feasible solution. There are two types of bound for a given problem that is lower bound and upper bound in minimization or maximization of objective function as per requirement in optimization.

In speech recognition well known fundamental algorithm for pattern matching in asymmetric data sequences is DTW. It has been found that unnecessary calculation is required when DTW will be used for pattern matching. Here two thresholds will be defined, one for minimum value and second for maximum value. Minimum threshold will be define from the similar word matching where maximum value will be define from the unknown word matching. Here it does not expand where bad match will found. It expands in the direction where lowest local minima will be found [7].

4. PARALLEL WARPING

In this method pattern matching algorithm do not perform sequentially. Here pattern matching will take place as frame wise. Each frame in the test pattern will be compared with each frame of the reference pattern. Here number of operation will perform simultaneously [8].

In this technique the bad matches which are strongly differing will be left. From here we can get the n number of best match in the recognition. These are the concept without pruning. If the value of n is high we can say the computational cost saving will be maximum. Branch and bound with pruning is also helpful to reduce the computational time [9].

5. PATTERN MATCHING

When the number of element in both matrixes is equal then it can match two patterns by taking differences between elements to element. But when the number of elements is not equal in nature then we apply warping algorithm where number of computation in pattern matching is very large [14]. In DTW order of time complexity is $O(N^2)$ for one to one matrix. If the number of words stored in data base is w then total time taken by the algorithm is $O(N^2w)$. After using the window concept the problem size will reduce which reduce the computation time.

6. HIGH PERFORMANCE COMPUTING

For high performance computing state solution is that the machine configuration should be of super computer or multiprocessor system in a single machine for providing fast calculation and efficient result within time bound frame. Secondly design the efficient algorithm having response time better than the exiting algorithm used for the solution of a problem.

6.1 Algorithm for Proposed Solution

Algorithm HPC_Speech()

Begin

Step 1. Take input string 'w' as wave file.

Step 2. Find out the components by applying feature extraction techniques with parameters M[ ].

// where M is an array containing the feature extraction techniques.

Step 3; L= length of component of feature extracted in step 2.

Step 4; N = fun(w, L);

// where fun() is defined as function which gives compressed version of components of extracted features

Step 5; Create different threads(N)

th = Thread(N);

// where th is the number of threads.

Step 6; for k = 1 to d step by one do

{

// where d is number of isolated spoken words.

for i = 1 to th, step by one do

{

R[k][i] = Pattern_ matching_ algorithm(P[ ], i)

// where P is an array containing the pattern matching techniques and R is a two dimensional array which stores the probability of matching.

}

}

Step 7; for i=1 to th step by one do

{

for j= 1 to d step by one do

{

k=max{R[j][i]}

return (j)

//where j is base index for value k

}

S[i] = j;

// where S is an array which stores the index of maximum values for each th.

}

Step 8; Count the maximum occurrence of the index value in the S array which corresponds to a matched isolated spoken word stored in the database.

6.2 Analysis of Algorithm

Step 1; time complexity: O(1)

Step 2; time complexity: depends on the feature extraction technique used.

Step 3; time complexity: O(1)

Step 4; time complexity: depends on the compression technique used.

Step 5; time complexity: O (1)

*Step 6; time complexity: O (d*th*p)*

// where p is the time complexity of pattern matching algorithm used

*Step 7; time complexity: O (th*d*d)*

Step 8; time complexity: O (th)

$$\text{Total time complexity} = O(d*th*p) + O(th*d*d) + O(th)$$
$$= O(d*th*p) + O(th*d*d)$$
$$= O(d*th(p+d))$$
$$= O(d*th*p) \quad \text{-- (1)}$$

(for small vocabulary d<<p)

If th = N/k, where k is length of each thread

If k = 1, then th= N which makes time complexity of pattern matching algorithm i.e.

$$O(p) = O(1) \quad \text{--(2)}$$

Putting (2) in (1)

$$\text{Total time complexity} = O(d*th) \quad \text{--(3)}$$

On comparing equation (1) with (3) we infer that the time complexity can be reduced by using the concept of multithreading.

From equation (1)

*If d=N, th=1 and time complexity for pattern matching algorithm is O(N*N)*

*Therefore we get total time complexity= O(N*N*N)*

If d=N and th=N then time complexity of pattern matching algorithm = O(1)

*Total time complexity= O(N*N) --(4)*

*Time Complexity = O(d*th*p) -- (1)*

If d=N, th=N/K

*And time complexity for pattern matching algorithm is O(N*N)*

O(th)=O(N(N/K)*N*N/K*K)*

Therefore we get total time complexity= O(N*N*N)

Let the size of whole problem is N then time complexity will depend over the pattern matching algorithm applied.

If we used linear time warping algorithm for pattern matching then order of time complexity is O(N) where N is the number of samples observed in the feature extraction.

Let fraction k = N/F; Where F is an integer

Then time complexity for pattern matching will be of order O(k);

Here K<<N

With the help of dynamic time warping algorithm the order of time complexity is O(M*N) where M and N is the length of spoken word and reference word putted in the database of vocabulary.

If we use k as the size of test sample/pattern then order of time complexity will be O(K*K)

Here K*K << M*N; since K<<N

If m= n then , **$T(n) = O(n^2)$**

Here in this approach small segments will be run simultaneously and produce the result within very less amount of time. Here time complexity will be O (K) by linear time warping and O(K*K) with dynamic time warping algorithm which is very less than the actual time taken by the algorithm.

7. CONCLUSION

An experiment has been done on different utterances of different person in different environment and result has been found satisfactory. The response time has been increased in comparison to normal pattern matching algorithm.

8. FUTURE WORK

Clustering technique in database design will also help to improve the efficiency. This experiment may be performing by different techniques of feature extraction and pattern matching.

With the help of this approach stem may improve their efficiency more.

9. ACKNOWLEDGMENT

Author is thankful to Dr. Abdul Mobin (Chief Scientist in National Physical Laboratory Delhi) for his valuable suggestion to develop and implementation of algorithm

10. REFERENCES

[1] Gregory N, Stainhaour and George Carayannis, “New Parallel Impementations for DTW Algorithms”, IEEE Transactions on Acoustics , Speech, and Signal Processing. Vol. 38, No. 4, 1990, pp 705-711.

[2] Lawrence Rabiner,Biing-Hwang Juang and B.Yegnanarayana. “Fundamentals of Speech Recognition”, Dorling Kindersley (India) Pvt. Ltd. 2009. Pp 135-136.

[3] Santosh K. Gaikward, Bharti W. Gawali and Pravin Yannawar. “A Review on Speech Recognition Techniwue”, International Journal of Computer Applications (0975-8887) Volume 10-No. 3 November 2010, pp 16-24;

[4] Sakoe H. and Chiba, H.S. “Dynamic programming algorithm optimization for spoken word recognition”. IEEE Trans. Acoust. Speech Signal Process , 1978, 26,(1), pp. 43-49.

[5] S. W. Jang, Y.J. Park and G.Y. Kim. “Branch and bound dynamic time warping”, The Institution of Engineering and Technology, 2010, Vol. 46 No. 20.

[6] Thomas H. Cormen , Charkes E. Leiserson and Ronald L. Rivest, “Approximation Algorthm. Introduction to Algorithms Prentic Hall of India Private Limited 2001, ISBN81-203-1353-4 pp 1022-1023.

[7] Anany Levitin , “Introduction to The Design & Analysis of Algorithm” , Villanova University. Pearson Education (Singapur)Pvt. Ltd. 2003. Pp 376-377.

[8] Waibel Alexander, Krishanan N and Reddy Dabbala Rajagopal ,‘‘Minimizing computational cost for dynamic programming algorithms(1981)” , Carnegie Mellon University, Computer Science Department, Paper 2436.

[9] S. W. Jang, Y. J. Park and G. Y. Kim , “ Branch-and-Bound dynamic time warping ”, The Institution of Engineering and Technology 2020,vol. 46, No. 20.

[10] Peckham.J. “Automatic Speech Recognition”. New Electronics, pp 24-25

[11] H.T. Kumm and R.M. Lea Parallel “Computing Efficiency: Climbing the Learning Curve” IEEE pp 728-732

[12] Quarmby, D., “Silicon Devices in Speech Recognition” in Bristow,G.(Ed):Electronic Speech Recognition (Collins London) pp 200-215 .1986.

[13] B.A.Dautrich,L.R.Rabiner and T.B.Martin, “On The Use of Filter Bank features For Isolated Word Recognition”, Proc.IEEE Conf. ASSP.pp. 1061-1064,1983.

[14] Denes.P.and Mathews.M.V., “Spoken Digit Recognition Using Time Frequency Pattern Matching”, J.Acoust.Soc.Am, Vol.32,1960, pp1450-1455.