Volume 3, Issue 3, March 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Study of Different Algorithms for Pattern Matching

Mr. Rahul B. Diwate*
M.E. Scholar
Department of C.S.E
G.H.R.C.M., Amravati, India

Prof. Satish J. Alaspurkar
Assistant Professor
Department of C.S.E
G.H.R.C.M., Amravati, India

Abstract: ***In Real-time world problems need fast algorithm with minimum error. Now days there are many applications are use for searching results on web. There are many algorithms which are used for the searching the results. Pattern matching method is one of them. In web application people deals with the different types of data, for example text searching, image searching, audio searching and Video searching. Every search engine uses different search algorithms for handling different types of data. Full search algorithm increases the pattern matching process. This paper will discuss about complexity, efficiency and techniques used by the algorithms relates with different. This paper proposes an analysis and comparison of different algorithms for full search equivalent pattern matching like complexity, efficiency and techniques. Which algorithm is best for which type of data?***

Keywords: Pattern matching, Complexity, Efficiency, Techniques

1. Introduction

In web search engine every searching operation is done online. Now a day's different search engine are in the market like Google, yahoo etc. The performance of any search engine is depends on its searching capabilities. Searching a list for a particular item is a common task. In real applications, the list items often are records and the list is implemented as an array of objects. In search engine is deals with the different type of data (text, Image, Audio, Video).for handling such type of data there are two types of searching methods are used. Linear and Binary searching method.

Linear search: finds an item in an unsorted sequence .For search algorithms, the main steps are the comparisons of list values with the target value. Counting these for data models representing the **best case**, the **worst case**, and the **average case** produces the following table. For each case, the number of steps is expressed in terms of n, the number of items in the list.

Model	Number of Comparisons **(for n = 100000)**	Comparisons as a function of n
Best Case **(fewest comparisons)**	1 (target is first item)	1
Worst Case **(most comparisons)**	100000 (target is last item)	N
Average Case **(average number of comparisons)**	50000 (target is middle item)	n/2

Table 1: Comparisons of Linear Search

Binary search algorithm: locates an item in a sorted sequence. Sometimes string searching algorithms are called as string matching algorithms, are an important class of string algorithms that try to find a place where one or several strings (also called patterns) are found within a larger string or text.

Model	Number of Comparisons **(for n = 100000)**	Comparisons as a function of n
Best Case **(fewest comparisons)**	1 (target is in middle item)	1
Worst Case **(most comparisons)**	16 (target is not in array)	log2n

Table 2: Comparisons of Binary Search

2. Literature Review

The use of web application is increase day by day. The web new web applications are in use for searching data on the internet. String algorithms play an important role for this. Different people are working on software and hardware levels to make pattern searching faster. By applying various algorithms in various applications the approximate best algorithm for different applications is determined [1]. The recommended algorithms give the reduced complexity and also reduced computation time. The algorithms assigned to various applications may not be best optimal algorithm but better than the general algorithms. Rather than applying each algorithm to every application one application is explained with particular optimal algorithm [2]. To support a different type of data, different algorithms are used. Each algorithm plays a different role to searching a data. Pattern matching is used in different application. Web search engine is one of them. In search engine it deals with different format of data like Text, Image, Audio, Video files. They used many algorithms.

3. Need of Pattern Matching?

Pattern matching is the process of checking a perceived sequence of string for the presence of the constituents of some pattern. In contrast to pattern recognition, the match usually has to be exact. The patterns generally have the form sequences of pattern matching include outputting the locations of a pattern within a string sequence, to output some component of the matched pattern, and to substitute the matching pattern with some other string sequence (i.e., search and replace). Pattern matching concept is used in many applications Following figure shows the different applications.

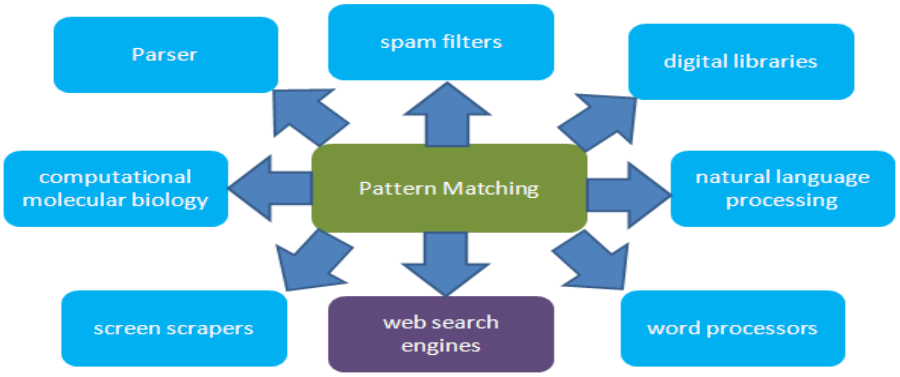

Figure 1: Applications of Pattern Matching

In pattern matching I focused on the web search engine amongst others application. Now a day's almost every people use the web application to get the desire results. But it is not necessary that peoples will only searching for text every time. They may want different type of data like audio, image and video. To handle such kind of data we need better method for searching. Pattern matching will help to find right and appropriate result. There are many algorithms used to find patterns matching.

4. Notation Used By Algorithm

Following notation are used in algorithms

Notation	Used for
n	the length of the text
m	the length of the pattern (string)
c	the size of the alphabet *

Table 3: Notations used for algorithms

Notations used in various algorithms are as given below,

1. Omega Ω Notation:-

The big O notation gives only upper bound on the function. If we are interested in lower bound values of the function then we have Ω notation. Let f(n) and g(n) be function whose domain as a subset of the positive integer, if there exists a positive constant c for all $n>n_0$ where n_0 is threshold integer such that,

f (n)>=c* g(n), Then we write it as

f (n)=Ω(g(n)), Read as "f of n equals omega of g of n"

2. Big-O Notation:-

Let f(n) and g(n) be functions whose domain is subset of the positive integers, if there is exists a positive constant c for all $n>=n_0$,Where n_0 is threshold integer such that,

f(n)<= c*g(n) , then we write it as

f(n)<= O(g(n)), and read as "f of n equals big O of g of n"

3. Theta Θ Notation:-

In some cases the time for an algorithm f(n) will be as,
f(n)=Ω(g(n)) and f(n)=O(g(n))
i.e. f and g have same order of magnitude and it is expressed
f(n) =Θ(g(n))
f(n)=Θ(g(n)) if and only if there exits positive constant c1,c2 and n_0 such that for all n>= n_0, c1*g(n)<=f(n)<=c2*g(n)

Relations of Big O, Omega Ω and Theta Θ

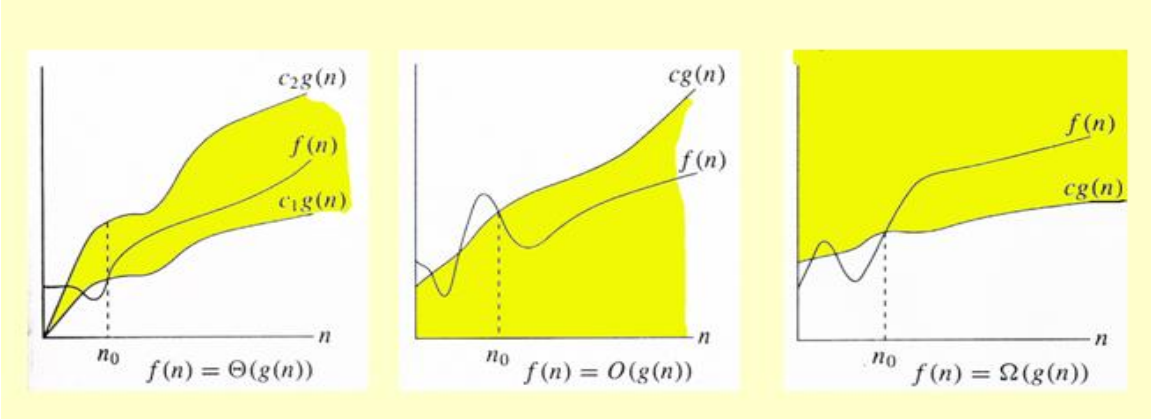

Figure 2: Relations of Big O, Omega Ω and Theta Θ

5. Algorithm Used For Matching

I. Naive string search algorithm

The "naive" approach is easy to understand and implement but it can be too slow in some cases. If the length of the text is n and the length of the pattern m, in the worst case it may take as much as (n * m) iterations to complete the task.

II. Rabin Karp String Search Algorithm

It is a string searching algorithm that uses hashing to find any one of a set of pattern strings in a text. For text of length *n* and *p* patterns of combined length *m*, its average and best case running time is O (*n*+*m*) in space O (*p*), but its worst-case time is O (*nm*).

III. Knuth–Morris–Pratt algorithm

KMP string searching algorithm searches for occurrences of a "word" W within a main "text string" T by employing the observation that when a mismatch occurs, the word itself contains sufficient information to determine where the next match could begin, thus bypassing re-examination of previously matched characters.

IV. Boyer–Moore string search algorithm

It is a particularly efficient string searching algorithm, The algorithm preprocesses the target string (key) that is being searched for, but not the string being searched in. Generally the algorithm gets faster as the key being searched for becomes longer. Its efficiency derives from the fact that with each unsuccessful attempt to find a match between the search string and the text it is searching.

6. ALGORITHMS TECHNIQUES

Every algorithm uses some special techniques to find pattern matching. Following table shows the different techniques used by different algorithms.

Algorithm	Techniques
Naive string search algorithm	Each character of the pattern is compared to a substring of the text which is the length of the pattern, until there is a mismatch or a match.
Rabin–Karp string search algorithm	Hashing
Knuth–Morris–Pratt algorithm	Two indices l and r into text string t
Boyer–Moore string search algorithm	Use both good suffix shift and bad character shift

Table 4: Different Techniques used by Different Algorithms

7. Worst Case Complexity

The worst-case complexity (denoted in asymptotic notation) measures the resources (e.g. running time, memory) an algorithm requires in the worst-case. It gives an upper bound on the resources required by the algorithm. In the case of running time, the worst-case time-complexity indicates the longest running time performed by an algorithm given *any* input of size ***n***, and thus this guarantees that the algorithm finishes on time. Moreover, the order of growth of the worst-case complexity is used to compare the efficiency of two algorithms [4].

Algorithm	Preprocessing time		Matching time/Searching Phase/Running Time
	Time complexity	Space complexity	
Naive string search algorithm	0 (none)	O(1)	O((n-m+1) m)
Rabin–Karp string search algorithm	O(m)	O(m)	Average O(n+m), worst O((n-m+1) m)
Knuth–Morris–Pratt algorithm	O(m)	O(m)	O(m+n)
Boyer–Moore string search algorithm	O(m + |Σ|)	O(m + |Σ|)	Ω(n/m), O(n)

Table 5: Comparisons of Worst case Complexity of different algorithms

8. Algorithm For Image, Audio And Video

JPEG image Compression

JPEG image compression works in part by rounding off nonessential bits of information.There is a corresponding trade-off between information lost and the size reduction. A number of popular compression formats exploit these perceptual differences, including those used in music files, images, and video. Lossy image compression can be used in digital cameras, to increase storage capacities with minimal degradation of picture quality. Similarly, DVDs use the lossy MPEG-2 Video codec for video compression [8].

Audio data Compression

Audio data compression, as distinguished from dynamic range compression, has the potential to reduce the transmission bandwidth and storage requirements of audio data. Audio compression algorithms are implemented in software as audio codec. Lossy audio compression algorithms provide higher compression at the cost of fidelity and are used in numerous audio applications. In both lossy and lossless compression, information redundancy is reduced, using methods such as coding, pattern recognition, and linear prediction to reduce the amount of information used to represent the uncompressed data [9].

Video data Compression

Video compression uses modern coding techniques to reduce redundancy in video data. Most video compression algorithms and codec's combine spatial image compression and temporal motion compensation. Video compression is a practical implementation of source coding in information theory. In practice, most video codec's also use audio compression techniques in parallel to compress the separate, but combined data streams as one package. [10].The majority of video compression algorithms use lossy compression. Uncompressed video requires a very high data rate. Highly compressed video may present visible or distracting artifacts.

Algorithm	Image	Audio	Video
Lossy data compression algorithms	YES	NO	NO
Lossy audio compression	NO	YES	NO
Lossy video compression algorithms	NO	NO	YES
FastDTW	YES	YES	YES

Table 6: Different Algorithm for Image, Audio, Video

9. DYNAMIC TIME COMPLEXITY (Fastdtw Algorithm):

The dynamic time warping (FastDTW) algorithm is able to find the optimal alignment between two time series. It is used to find an appropriate matching for Image, Audio and Video. It is often used to determine time series similarity, classification, and to find corresponding regions between two time series. FastDTW has a quadratic time and space

complexity that limits its use to only small time series data sets. FastDTW aligns two time series by warping the time dimension [5].

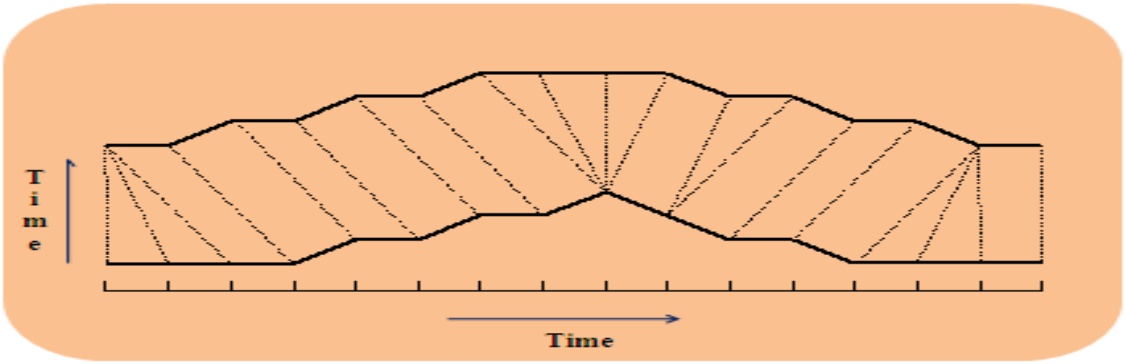

Figure 3: FastDTW wrapping between two time series

The need for methods to speed up the quadratic time and space complexity of FastDTW is creates dynamic time warping. The methods used make FastDTW faster fall into three categories:

1) *Constraints* – Limit the number of cells that are evaluated in the cost matrix.
2) *Data Abstraction* – Perform FastDTW on a reduced representation of the data.
3) *Indexing* – Use lower bounding functions to reduce the number of times FastDTW must be run during time series classification or clustering.

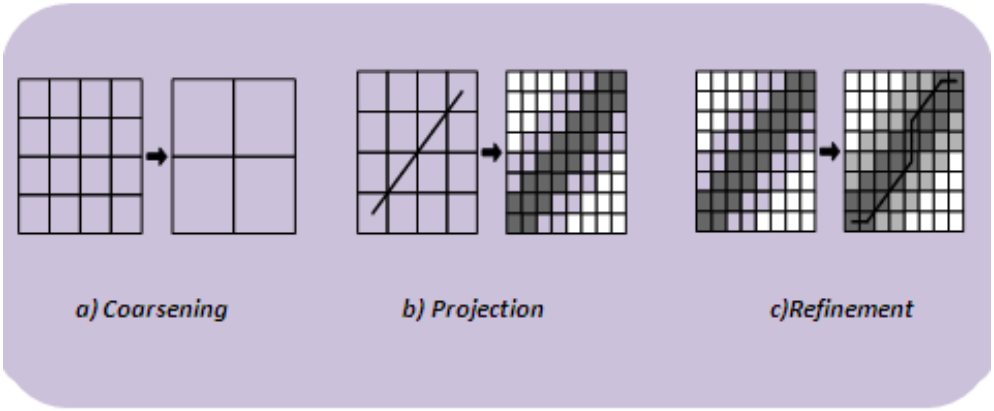

Figure 4: Diagram of three Categories

10. COMPLEXITY OF FastDTW ALGORITHM [6]

FastDTW is an approximate Dynamic Time Warping (FastDTW) algorithm that provides optimal or near-optimal alignments with an O(N) time and memory complexity, in contrast to the O(N^2) requirement for the standard FastDTW algorithm. FastDTW uses a multilevel approach that recursively projects a solution from a coarser resolution and refines the projected solution.

Time complexity –O (N)

Space Complexity – O (N)

11. Conclusion

In today's all work is performed on internet, searching is main operation performed by users. For searching a pattern matching is best techniques for searching. Each algorithms having its own characteristics. The Boyer Morris and Knuth–Morris–Pratt algorithm are more effective for searching. We focused on complexity of each algorithm, Knuth–Morris–Pratt algorithm having less time complexity and Boyer Morris algorithms having less preprocessing time complexity. Fast DTW algorithm is best for all Image, Audio and Video pattern processing. FastDTW has a linear time and space complexity. The time performance of exact string pattern matching can be greatly improved if an efficient algorithm is used.

References

1] Georgy Gimel'farb, "String matching Algorithms", COMPSCI 369 Computational Science

2] Nimisha Singla, Deepak Garg , "String Matching Algorithms and their Applicability in various Applications", International Journal of Soft Computing and Engineering (IJSCE), Volume-I, Issue-6, January 2012

3] *P.Jayaprabha and Rm. Somasundaram ,"Content Based Image Retrieval Methods Using Graphical Image Retrieval Algorithm (GIRA)"., COMPUTER SCIENCE AND APPLICATION, VOL. 1, NO. 1, JANUARY, 2012*

4] *1997,http://www.igm.univmlv.fr/~lecroq/string/index.html C. Charras and T. Lecroq: Exact String Matching Algorithms. Univ. de Rouen,*

5] *Srikanthan, Sharanyan ,"Implementing the dynamic time warping algorithm in multithreaded environments for real time and unsupervised pattern discovery"., Computer and Communication Technology (ICCCT), 2011 2nd International Conference on 394 – 398, 15-17 Sept. 2011,*

6] *Stan Salvador & Philip Chan,"FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space".KDD Workshop on Mining Temporal and Sequential Data, pp. 70-80, 2004. (http://cs.fit.edu/~pkc/papers/tdm04.pdf)*

7] *Chu, S., E. Keogh, D. Hart & Michael Pazzani. Iterative, Deepening Dynamic Time Warping for Time Series. In Proc. of the Second SIAM Intl. Conf. on Data Mining. Arlington, Virginia, 2002.*

8] *Arcangel, Cory. "On Compression". Retrieved 6 March 2013.*

9] *Jaiswal, R.C. Audio-Video Engineering. Pune, Maharashtra: Nirali Prakashan. p. 3.41. (2009). ISBN 9788190639675.*

10] *"Video Coding". Center for Signal and Information Processing Research. Georgia Institute of Technology. Retrieved 6 March 2013.*

