Approximate Search Techniques for Big Data Analysis

Ph.D. student: Ambika Shrestha Chitrakar
Supervisor: Prof. Slobodan Petrovic
Second supervisor: Prof. Katrin Franke
Institution: Gjøvik University College
Date: 14th of October 2014

Search problem

- Process of matching pattern P and its occurrences in the text T
- Two types of search
  - Exact search – P should be a substring of T
  - Approximate search – allows k errors for transforming P into its occurrences in T
    - Distance functions such as Levenshtein distance (Edit distance), Hamming distance can be used to define k errors in ASA.
    - In Edit distance, k = number of character insertions, deletions, and substitutions
    - In Hamming distance, k = number of character substitutions
- When P=definitely and T=definately, k = 1
- When P=survey and T=surgery, k = 2

Why approximate search?

- Search is a fundamental problem in applications, especially in big data analysis
- Exact search does not work well when there is error (e.g., spelling mistakes)
  - In text
    - E.g., OCR processed documents, big data (large volume of data from variety of sources)
  - In pattern
    - E.g., typing mistake, user may not know the exact spelling of the search keyword

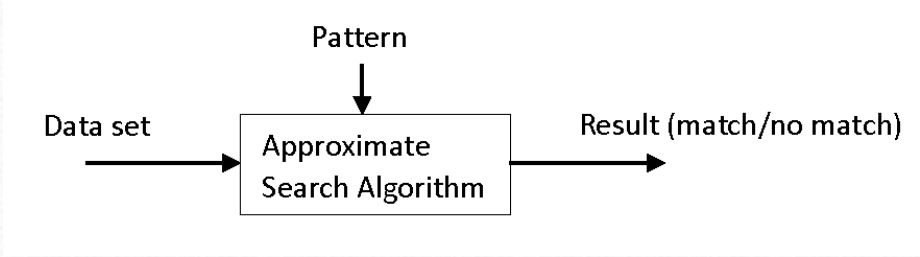
Problem description

- Two types of search methods
  - Online methods:
    - Dynamic data set
    - Indexing is possible only on pattern
  - Offline methods:
    - Static data set
    - Indexing is possible both on pattern and data set

What is the efficiency (speed) of search algorithms?

Problem Description (contd..)

- Efficiency of the algorithm can be affected by the following parameters:
  - Size of the data set
  - Size of the pattern
  - Choice of the error threshold
  - Requirement of the multi-pattern
  - Construction time and usage of space in indexing



Motivation

- Requirement on ASA to efficiently handle growing data sets
- Efficiency is desirable in any application, it is crucial when immediate decision has to be made on the basis of search results
  - Intrusion detection system (IDS)
- Efficiency of search algorithms is also important where manual analysis is done
  - Forensic Toolkit (FTK)

Research Qestions

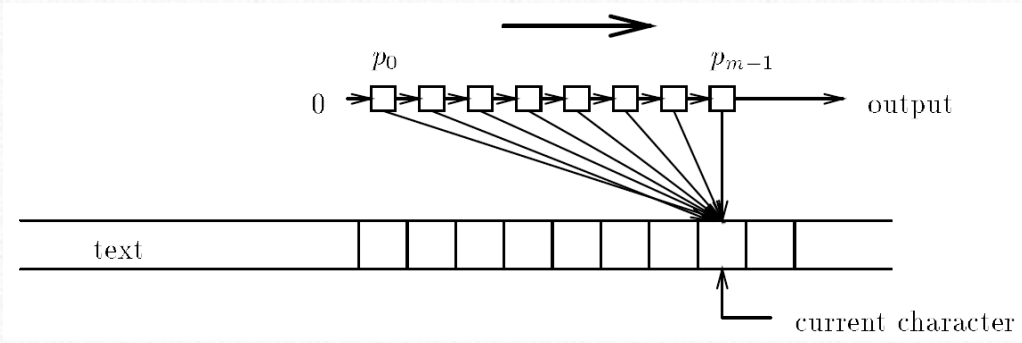
- RQ1: How can we improve the performance of existing approximate search algorithms in order to effectively use them in dynamic data sets whose size grows very fast over time?
- RQ2: What are the theoretical benefits of using modern hardware technologies (parallel implementation, multi-level logic etc.) in implementation of approximate search algorithms for big data analysis?
- RQ3: What are the benefits on applying approximate search algorithms specially tailored for big data analysis in intrusion detection, digital forensics and cryptanalysis?

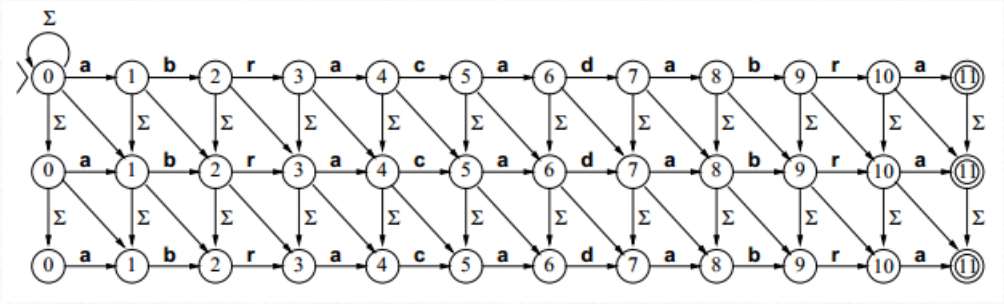
Methods for ASA

- Dynamic programming
- Automata theory
- Bit-parallelism
- Filtring

Popular methods for ASA

		s	u	r	g	e	r	y
	0	1	2	3	4	5	6	7
s	1	**0**	1	2	3	4	5	6
u	2	1	**0**	1	2	3	4	5
r	3	2	1	**0**	1	2	3	4
v	4	3	2	1	**1**	2	3	4
e	5	4	3	2	2	**1**	**2**	3
y	6	5	4	3	3	2	2	**2**





Filtration: filtering and verification

Thank you for your attention!

Any feedback or questions?