PA3 Starpoint - CSE 100

Gengshan Yang

2016/5/14

a. You may have noticed that the first step to creating an Aho-Corasick Automaton is pretty much the same as creating a Multiway Trie, and that the structure of an Aho-Corasick Automaton in general looks very much like a Multiway Trie. What specific feature of the Aho-Corasick Automaton differentiates its structure from a Multiway Trie, and how does this special feature help us?

For each Aho-Corasick node, besides the pointers to all its children, it has another pointer called "fail-pointer". If we don't have this feature, we have to move the pointer to the string one step a time. In the original case, the time complexity is $O(D \cdot B)$, where D is the depth of the trie and B is the length of a database string.

We use the "failpointer" to avoid rewinding the string pointer by transiting the graph pointer to the the most recent "correct" state. The "correct" state indicates a sub-string of a pattern, which shares the suffix with the database string. Therefore we can continue comparing the database string with the last part of a query (the beginning of a query overlaps the database), withing rewinding the cursor. The time complexity is O(B+Z), and Z is the number of occurrence of the patterns.

In addition, to achieve this good property of Aho-Corasick Automaton, we have to adjust the "failpointer" after building the multi-way search tree. This process is similar to the calculation of "next" array for KMP algorithm. However, Aho-Corasick Automaton is more powerful than KMP in searching for multiple patterns in a string.

b. How does your FindAllQueries solution scale as a function of n? Include a plot of time vs. n

As shown in Fig 1, The running time scales linearly with the database size.

c. Give me a real-world example of where I might want to use an Aho-Corasick Automaton (outside of the RNA-Seq example given above in the Motivation section)

This case is from a research project on activity recognition I've worked on. Generally, we can use the Aho-Corasick to search for patterns in a time sequence after feature quantization.

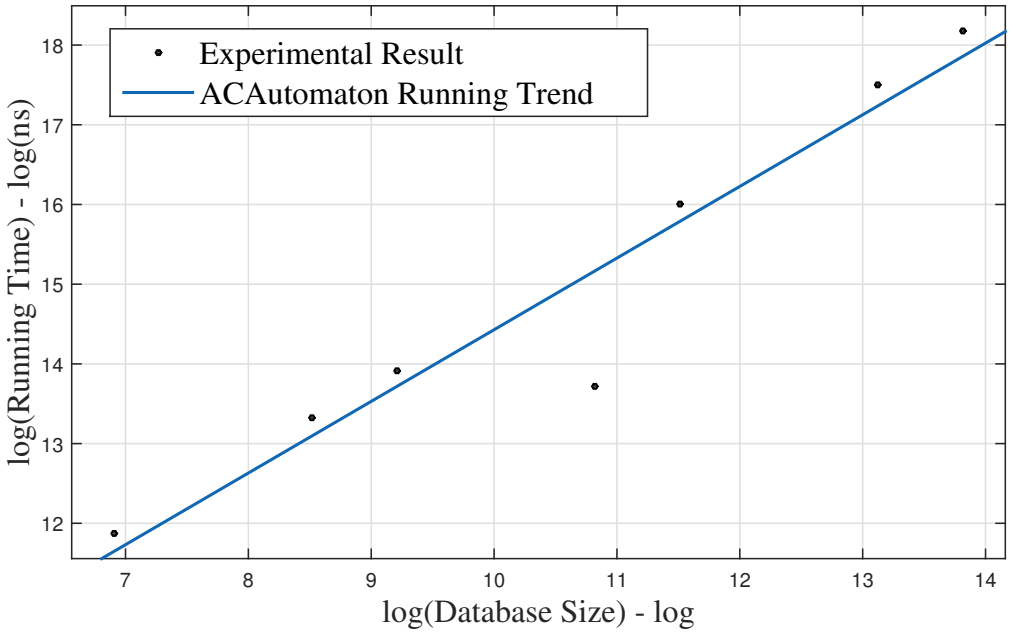

Figure 1: Running Time - Data Size plot

As for monitoring people's daily behavior, we put some inertial sensors(accelerometer and gyroscope) on people's wrists. We can first crop the recorded time sequences and extract feature vectors for each time window. Then, We can learn the patterns of specific activities, for example, waving and playing basketball can be represented by sequences of feature vectors.

Then we can quantify the patterns and sequences (corresponding to the database string), and apply the Aho-Corasick Automaton to search these pattern in the time sequences. After that, we can tell in which time stamp the people performed which activity.