Improved Greedy Algorithm to Look for Median Strings

Ferenc Kruzslicz

The distance of a string from a set of strings is defined by the .sum of distances to each string of the given set. A string that is closest to the set is called the median of the set. To find a median string is NP-Hard problem in general, so it is useful to develop fast algorithms that give a good approximation of the median string. These methods are significally depend on the type of distance used to measure the dissimilarity between strings. This algorithm is based on edit distance of strings, and constructing the approximate median in a letter by letter manner.

Introduction. If optical character recognition (OCR) problem is considered as a "black box" process, where images are mapped to strings, then we use a certain kind of off-line approach. In this way the efficiency of some OCR processes could be increased in OCR software and language independent manner. Suppose we have a set of strings as results of OCR processes of the same input bitmap. When the same OCR software was used to produce this set, with different paper orientation, changed resolution or simply repeated OCR processes we can eliminate the effects of pollution (fingerprints on the glass etc.) While in case of different OCR software their efficiency can be compared to each other.

To find a median string that is minimal in sum of distances form a given input set of strings, is known to be NP-hard problem. Therefore it is interesting to find fast algorithms, that give as good approximations. One of the latest algorithms is called greedy algorithm, because it builds up the approximate median string letter by letter, by always choosing the best possible continuation. In this paper an improvement of this algorithm is described.

The real advantage of the improved algorithm appears when the probability of edit operations in the garbling process is increased. In other words the improved algorithm works better if the strings in the input set are far from each other. In the example the string *recognition* was garbled with delete, insert and substitute string-edit operations. For substituting and inserting only the letters *r, e, c, g, t, i, o, n, s, p, a* were used, and each of the operations and its place was equally distributed.

For example let us consider the following test set H = { *ggroeonitin, rpcsogngapaoponc, secsgttin, gecciicn, eectgcgnitiopn, repsogniporpassn, raatnini, rnrcpnto, nirnscogtipntgo, nrectogansinageine* } where the cost of all edit operations is 1.

- The greedy algorithm gives the result *recrognitin* with summarized distances 75,
- and the improved algorithm found the string *rectognitin* with sum of distances 74,
- while the median of H is *recognition*, where the sum of distances is 73.

Conclusions. The improved approximate median algorithm is a simple refinement of the greedy algorithm. It has the same time complexity $O(k^2 n|\Sigma|)$ as the previous one (where $|\Sigma|$ is the size of the alphabet, n is the number of input strings, and k is the length of the longest one). The space complexity was a bit reduced as well, because the new algorithm runs only in $O(kn)$ space. The garbled strings are closer to each other the improvement is less significant. Therefore the new improved greedy algorithm is more suitable for searching approximate median of highly dissimilar strings.