Character Distance Text Sampling in the case of Small Alphabets

Francesco Pio Marino

Department of Mathematics and Computer Science
University of Catania, Catania, Italy - marfra99x@gmail.com

Extended Abstract

String Matching consists in finding all occurrences of a given pattern x, of length m, in a large text y, both over an alphabet Σ. Applications require two kinds of solutions: *online* and *offline* string matching. Solutions based on the first approach assume that the text is not preprocessed and thus they need to scan the text *online*[8], when searching. Solutions based on the second approach tries to drastically speed up searching by preprocessing the text and building a data structure to make searching faster, such kind of problem is known as *indexed searching* [6, 9, 10].
A more suitable solution to the problem is *sampled string matching*, introduced in 1991 by Vishkin [1], which consists in the construction of a succint sampled version of the text called partial-index and in the application of any online string matching algorithm, like Boyer-Moore-Horspool algorithm [5], directly on the sampled sequence.Apart the theoretical result of Vishkin, a more practical solution has been recently introduced by *Claude et al.* [2], based on alphabet reduction. They are able to speed up the searching up to 5 times using less than 14% of the text size. They also consider indexing the sampled text building a suffix array using the sampled version of the text, and get a sampled suffix array. This approach is similar to the sparse suffix array [4] as both index a subset of the suffixes, but the different sampling properties induce rather different algorithms and performance characteristics.
Recently a new solution has been introduced by *Faro et al.* [3] using a new approach for the creation of the partial-index this method is called *Characters Distance Sampling*, it turns out that this solution shows a sub-linear behaviour in practice and speeds up online searching by a factor of up to 9, using limited additional space whose amount goes from 11% to 2.8% of the text size, with a gain up to 50% if compared with previous solutions. However has been demonstrated by experimental results that these sampling string matching approach is not applicable to small alphabet [2]. The same authors has recently introduced a new indexed approach based on their different text sampling method [7]. The main idea behind their new text sampling approach is to sample the distances between consecutive occurrences of a given set of *pivot characters* and then to create a suffix array of the sampled text.

In this paper we take into account the problem of using sampled matching solutions also for small alphabets obtaining a new efficient method which turns out to be much more feasible for searching in biological data like genome or protein sequences. The main problem of sampling string matching using small alphabets is related with the equidistribution of the characters. Specifically, in the case of sampling string matching methods which elect a pivot character to build a data structure, the additional space is based on the number of occurrences of the pivot character that has to be stored. In the average case such number is $\frac{m}{\Sigma}$, showing an inversely proportional trend between the size of the alphabet and the average number of characters to store, this result is confirmed by experimental results. Our new proposed solution is based on the concept of condensed alphabet, using q-grams in order to enlarge the size of the alphabet and speed up the whole process of pre-processing and searching.

This approach allows to reduce the space consumption of the resulting sampled text. This kind of solution has been applied in both cases of online and offline version of the algorithms based on the character distance sampling approach.

For this new method we elect a pivot character using the new condensed alphabet and we redefine the distance function used in the previous algorithms as the *q-characters distance function* which describes the distance of two consecutive occurrence of a specific q-gram in the text. In the offline version of the algorithm the suffix array has been used to create an indexed of the text. The pre-processing of this solution is divided into three different phases: first of all we enlarge the size of the alphabet using the condensed method; later we use the characters distance sampling to build the partial-index; finally we create the suffix array using the method exposed by Faro and Marino. [7].

From our experimental results it turns out that our solution leads to reduce the space consumption up to 80% and to speed up the searching up to 95% when compared against standard string matching algorithms on biological data.

References

1. U. Vishkin, Deterministic sampling— - A new technique for fast pattern matching. In Proc. of the ACM Symposium on Theory of Computing (STOC), pp.170-180 (1990)
2. F. Claude, G. Navarro, H. Peltola, L. Salmela, J. Tarhio, String matching with alphabet sampling, Journal of Discrete Algorithms, vol. 11, pp. 37–50 (2012)
3. S. Faro, F.P. Marino, A. Pavone. Efficient Online String Matching Based on Characters Distance Text Sampling. arXiv:1908.05930, 2018.
4. J. Karkkainen, E. Ukkonen, Sparse suffix trees, in: Proc. 2nd Annual International Conference on Computing and Combinatorics (COCOON), LNCS 1090, pp. 219–230 (1996)
5. R. N. Horspool, Practical fast searching in strings, Software: Practice & Experience 10 (6), pp. 501–506 (1980)
6. P. Ferragina, G. Manzini. Indexing compressed text. Journal of the ACM, 52 (4), pp. 552–581, 2005
7. S. Faro and F.P. Marino. Reducing Time and Space in Indexed String Matching by Characters Distance Text Sampling In *Proc. of Stringology*, pages 148–159, 2020.
8. S Faro, T Lecroq, The Exact Online String Matching Problem: a Review of the Most Recent Results, ACM Computing Surveys (CSUR) vol. 45 (2), pp. 13 (2013)
9. A. Apostolico, The myriad virtues of suffix trees, in: A. Apostolico, Z. Galil (Eds.), Combinatorial Algorithms on Words, Vol. 12 of NATO Advanced Science Institutes, Series F, Springer-Verlag, pp. 85–96 (1985)
10. U. Manber, G. Myers, Suffix arrays: A new method for online string searches, SIAM J. Comput. 22 (5), pp. 935–948 (1993)