Compressed Indexes for Approximate String Matching

Ho-Leung Chan · Tak-Wah Lam · Wing-Kin Sung · Siu-Lung Tam · Swee-Seong Wong

Received: 6 September 2007 / Accepted: 2 December 2008 / Published online: 17 December 2008
© Springer Science+Business Media, LLC 2008

Abstract We revisit the problem of indexing a string \( S[1..n] \) to support finding all substrings in \( S \) that match a given pattern \( P[1..m] \) with at most \( k \) errors. Previous solutions either require an index of size exponential in \( k \) or need \( \Omega(mk) \) time for searching. Motivated by the indexing of DNA, we investigate space efficient indexes that occupy only \( O(n) \) space. For \( k = 1 \), we give an index to support matching in \( O(m + \text{occ} + \log n \log \log n) \) time. The previously best solution achieving this time complexity requires an index of \( O(n \log n) \) space. This new index can also be used to improve existing indexes for \( k \geq 2 \) errors. Among others, it can support 2-error matching in \( O(m \log n \log \log n + \text{occ}) \) time, and \( k \)-error matching, for any \( k > 2 \), in \( O(mk^{-1} \log n \log \log n + \text{occ}) \) time.

Keywords Compressed index · Approximate string matching

Result in this paper has appeared in a preliminary form in the Proceedings of the 14th Annual European Symposium (ESA 2006).

Current address of H.-L. Chan: Department of Computer Science, University of Pittsburgh, USA.
The research T.-W. Lam was supported by the Hong Kong RGC Grant HKU 7140/06E.

H.-L. Chan · T.-W. Lam · S.-L. Tam
Department of Computer Science, University of Hong Kong, Hong Kong, Hong Kong

H.-L. Chan
e-mail: hlchan@cs.hku.hk

T.-W. Lam
e-mail: twlam@cs.hku.hk

S.-L. Tam
e-mail: sltam@cs.hku.hk

W.-K. Sung (✉) · S.-S. Wong
Department of Computer Science, National University of Singapore, Singapore, Singapore
e-mail: ksung@comp.nus.edu.sg

S.-S. Wong
e-mail: wongss@comp.nus.edu.sg
1 Introduction

Given a string $S[1..n]$ over a finite alphabet $\Sigma$ and an integer $k \geq 0$, we want to build an index for $S$, such that for any subsequent query pattern $P[1..m]$, we can report efficiently all locations in $S$ that match $P$ with at most $k$ errors. The primary concern is how to achieve efficient pattern matching given limited space for indexing. We consider two kinds of errors: In the Hamming distance case, an error is a character substitution; in the edit distance case, an error can be a character substitution, insertion or deletion.

For exact string matching (i.e., $k = 0$), simple and efficient solutions have been known in the 1970s. Suffix trees [16, 23] use $O(n)$ space\(^1\) and achieve the optimal matching time, i.e. $O(m + occ)$, where $occ$ is the number occurrences of $P$ in $S$. Suffix arrays [15], also using $O(n)$ space but with a smaller constant, give an $O(m + occ + \log n)$ matching time. Recently, two compressed solutions, namely, CSA [10] and FM-index [9], have been proposed, they require only $O(n)$-bit space and they can support matching in $O(m + occ \log^\epsilon n)$ time, for any constant $\epsilon > 0$.

Approximate matching is a challenging problem even if only one error is allowed. The simplest solution is to search the suffix tree of $S$ for every 1-error modification of the query pattern, this requires $O(m^2 + occ)$ time\(^2\) [7]. The first non-trivial improvement was due to Amir et al. [1], who showed that the matching time can be improved to $O(m \log n \log \log n + occ)$ using an index occupying $O(n \log^2 n)$ space. Later Buchshaum et al. [4] further improved the matching time to $O(m \log \log n + occ)$, as well as reducing the index space to $O(n \log n)$. Huynh et al. [12] and Lam et al. [13] further compressed the index to $O(n)$ space, while achieving the time complexity reported in [1] and [4], respectively. It has been an open problem whether a time complexity linear in $m$ and $occ$ can be achieved. Recently, Cole et al. [8] resolved in the affirmative with an $O(n \log n)$-space index that supports one-error matching in $O(m + \log \log n \log n + occ)$ time. And more recently, Chan et al. [5] found that Cole et al.’s index admits a time-space tradeoff, i.e., the space can be reduced to $O(n)$ space, yet the time complexity increases to $O(m + \log^3 n \log \log n + occ)$. In this paper, we give new techniques to compress Cole et al.’s index to $O(n)$ space, while retaining the same time complexity.

To cater for $k = O(1)$ errors, one can perform a brute-force search on a one-error index (i.e., repeatedly modify the pattern at different $k - 1$ positions and search for one-error matches); the matching becomes very slow, involving a factor of $m^k$ in the time complexity. A breakthrough result has been given by Cole et al. [8], who devised a recursive solution to build an index that occupies $O(n \log^k n)$ space and takes $O(m + \log^k n \log \log n + occ)$ time to perform a $k$-error matching. Our new 1-error index is essentially a compressed version of the Cole et al.’s 1-error index and can replace it as the base case in their recursive solution. This gives an $O(n \log^{k-1} n)$-space index for $k$-error matching with the same time complexity.

---

1Unless otherwise stated, the space complexity is measured in terms of the number of words, where a word can store $O(\log n)$ bits.

2Unless otherwise stated, all matching time mentioned applies to both Hamming and edit distance.