Sequence Alignment Algorithms for Run-Length-Encoded Strings

Guan-Shieng Huang[1] Jia-Jie Liu[2] Yue-Li Wang[3]

[1]National Chi Nan University, Taiwan
shieng@ncnu.edu.tw

[2]Shih Hsin University, Taiwan
jjliu@cc.shu.edu.tw

[3]National Chi Nan University, Taiwan
yuelwang@ncnu.edu.tw

June 27–29, 2008

國立暨南國際大學

Motivation

- Could string processing be done on compressed strings directly?
- Every one knows that data compression can save storage space; the tradeoff is to take more processing time.
- However, in some situations, both time and space can be saved through data compression.

Why is it possible to save both time and space through data compression?

- The size of the input data is reduced after compression.
- In complexity theory, time complexity and space complexity are measured with respect to the input size.
- A faster algorithm is possible on smaller input.

國立暨南國際大學

Run-Length Compression

Let x and y be two strings over a constant-sized alphabet.
The size of x is m, being compressed into m' runs.
The size of y is n, being compressed into n' runs.
(E.g., $x = aaabbccc \implies (a,3)(b,2)(c,3)$)

国立暨南国際大學

What We Have Done

We focused on string processing on run-length-encoded strings. We improved algorithms for solving the following problems:

1. the string edit distance problem;
2. the pairwise global alignment problem;
3. the pairwise local alignment problem;
4. the approximate matching problem

under a unified framework.

Assumption

- *The linear-gap model with arbitrary scoring matrix*
- *The size of the alphabet is constant*

國立暨南國際大學

Problems Description

1. the string edit distance problem
   - Input: two strings x, y and a substitution matrix δ that measures the cost for each edit operation (i.e. insertion, deletion, and substitution) performed on x
   - Output: the minimum sum of costs that can transform x into y
2. the pairwise global alignment problem
   - Input: two strings x, y and a scoring matrix δ that measures the aligned score of any two characters from the alphabet
   - Output: inset appropriate spaces (or gaps) into x and y, to make them equal-length, such that the aligned scored is maximized
3. the pairwise local alignment problem: find substrings x' of x and y' of y such that the alignment score of x' and y' is maximized
4. the approximate matching problem:
   - Input: a text string T, a pattern string P, and a number K
   - Output: locate all end-positions of substrings from T such that the edit distances of each candidate against P is at most K

國立暨南國際大學

Our Contribution

1. Edit distance problem, global alignment problem: $\mathrm{O}(\min\{m'n, mn'\})$ time
   - $\mathrm{O}(m'n + mn')$ time (Mäkinen & Navarro & Ukkonen, 2003) (Crochemore & Landau & Ziv-Ukelson, 2003)
   - $\mathrm{O}(\min\{m'n, mn'\})$ time for the edit distance problem with unit cost (Liu & Huang & Wang & Lee, 2007)
2. Local alignment problem: $\mathrm{O}(\min\{m'n, mn'\})$ time
   - $\mathrm{O}(m'n + mn')$ time only for LZW compression (Crochemore & Landau & Ziv-Ukelson, 2003)
3. Approximate matching: $\mathrm{O}(n'm)$
   - $\mathrm{O}(n'mm')$ time under some restriction (Mäkinen & Navarro & Ukkonen, 2003)

國立暨南國際大學

- Mäkinen, V., Navarro, G., Ukkonen, E.: Approximate matching of run-length compressed strings. Algorithmica (2003)
- Crochemore, M., Landau, G.M., Ziv-Ukelson, M.: A subquadratic sequence alignment algorithm for unrestricted scoring matrices. SIAM Journal on Computing (2003)
- Liu, J.J., Huang, G.S., Wang, Y.L., Lee, R.C.T.: Edit distance for a run-length-encoded string and an uncompressed string. Information Processing Letters (2007)
- Liu, J.J., Wang, Y.L., Lee, R.C.T.: Finding a longest common subsequence between a run-length-encoded string and an uncompressed string. Journal of Complexity (2008)

國立暨南國際大學

Related Work

- Wagner & Fischer (1974), Levenshtein (1966):
  Defined the string-to-string correction problem.
- Longest-common-subsequence problem on run-length-encoded strings
  - Bunke & Csirik (1995): $\mathrm{O}(m'n + mn')$ time
  - Apostolico & Landau & S. Skiena (1999): $\mathrm{O}(m'n' \lg(m'n'))$ time
  - Mitchell (1997): $\mathrm{O}((m' + n' + d) \lg(m' + n' + d))$ where d is the number of matches of runs
- Extensions
  - Arbell & Landau & Mitchell (2002): $\mathrm{O}(m'n + mn')$ time for the edit distance problem with unit cost
  - Mäkinen & Navarro & Ukkonen (2003): $\mathrm{O}(m'n + mn')$ time for the general edit distance problem
  - Crochemore & Landau & Ziv-Ukelson (2003): $\mathrm{O}(m'n + mn')$ time for the alignment problem
- Liu & Huang & Wang & Lee (2007): $\mathrm{O}(\min\{m'n, mn'\})$ time for the edit distance problem with unit cost

國立暨南國際大學

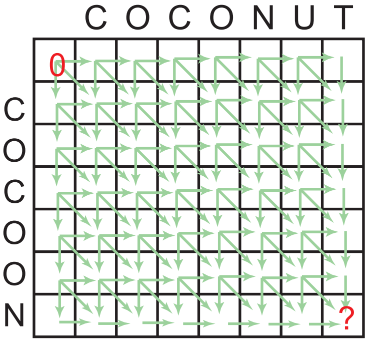
The String Edit Distance Problem

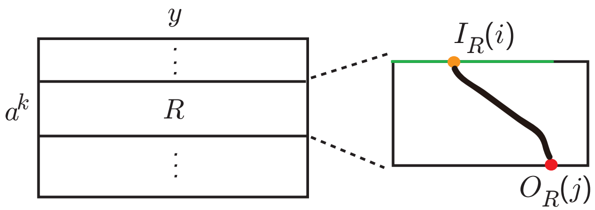
- Input: two run-length-compressed strings x and y over a constant-sized alphabet Σ.
- A substitution matrix $\delta : (\Sigma \cup \{-\}) \times (\Sigma \cup \{-\}) \longrightarrow \mathbb{R}$ is given to measure the cost of each character insertion, deletion, and substitution.
- Output: the minimum cost of edit operations that can transform x into y.
- Its time complexity is $\mathrm{O}(\min\{m'n, mn'\})$.

國立暨南國際大學

Basic idea

- The edit distance problem can be reduced to the shortest path problem on edit graphs.
- The goal is to find a shortest path from $(0, 0)$ to (m, n).





Hirschberg in 1975 observed that

$$O_R(j) = \min_{1 \le i \le j} \{ I_R(i) + DIST(i,j) \} \quad \text{for } 1 \le j \le n$$

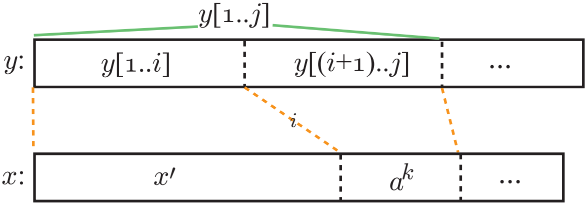
where $DIST(i,j)$ is the cost of the optimal (i.e. shortest) path starting from $I_R(i)$ and ending at $O_R(j)$ where $1 \le i \le j \le n$.

$$O_R(j) = \min_{1 \le i \le j} \{I_R(i) + DIST(i,j)\} \quad \text{for } 1 \le j \le n$$

can be instantiated by

$$E(x'a^k, y[1..j]) = \min_{0 \le i \le j} \left\{ E(x', y[1..i]) + E(a^k, y[(i+1)..j]) \right\} \ .$$

- $O_R(j) = E(x'a^k, y[1..j])$
  $=$ the edit distance of $x'a^k$ and $y[1..j]$.
- $DIST(i,j) = E(a^k, y[(i+1)..j])$
  $=$ the edit distance of a^k and $y[(i+1)..j]$.



Observations

$$O_R(j) = \min_{1 \le i \le j} \{I_R(i) + DIST(i,j)\} \quad \text{for } 1 \le j \le n$$

$$E(x'a^k, y[1..j]) = \min_{0 \le i \le j} \left\{ E(x', y[1..i]) + E(a^k, y[(i+1)..j]) \right\}$$

1. $DIST(i,j)$ can be evaluated in $\mathrm{O}(1)$ time for each i and j.
2. Let $i^*(j)$ be the parameter that minimizes the recurrence for a specific j. Then all $i^*(j)$ for $1 \le j \le n$ can be computed in $\mathrm{O}(n)$ time.

國立暨南國際大學

Observation I

How to evaluate $DIST(i,j) = E(a^k, y[(i+1)..j])$ for each i and j in $\mathrm{O}(1)$ time?

- $E(aaaaa, abcaa) =$?
- $E(aaaaa, abca) =$?
- $E(aaaaa, abcaaa) =$?

After preprocessing on string y, $E(a^k, y[(i+1)..j])$ can be answered in $\mathrm{O}(1)$ time.

Lemma

Let the length of z be $|z|$ and the number of occurrences of a in z be $\sigma_a(z)$. Then

- $0 \leq s \leq 2d$:

$$E(a^k, z) = d \max\{|z|, k\} - (d - s) \min\{|z|, k\} - s \min\{\sigma_a(z), k\}$$

- $s \geq 2d \geq 0$:

$$E(a^k, z) = d(|z| + k) - 2d \min\{\sigma_a(z), k\}$$

where s is the cost for a substitution and d is the cost for an indel.

The general case for any substitution matrix, even with negative weights, can be handled similarly.

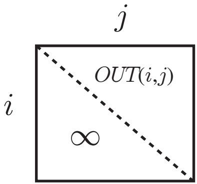
國立暨南國際大學

Observation II

Find all $i^*(j)$ for $1 \leq j \leq n$ in $\mathrm{O}(n)$ time.

$$O_R(j) = \min_{1 \leq i \leq j} \{I_R(i) + DIST(i,j)\} \quad \text{for } 1 \leq j \leq n$$

Let $OUT(i,j) = I_R(i) + DIST(i,j)$. Then the matrix $OUT(i,j)$ is a Monge matrix.



The Monge Property

Definition

An $m \times n$ matrix $M = (c_{i,j})_{m \times n}$ is called **Monge** iff

$$c_{i,j} + c_{i',j'} \leq c_{i,j'} + c_{i',j}$$

for all $1 \leq i \leq i' \leq m$ and $1 \leq j \leq j' \leq n$.

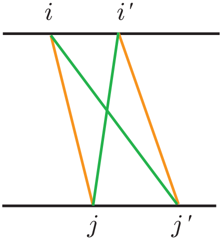
Named after Gaspard Monge (1746–1818) by A. J. Hoffman in 1961.

$$\begin{array}{cc}
 & \begin{array}{ccccc} & j & & j' & \end{array} \\
\begin{array}{c} \\ i \\ \\ i' \\ \\ \end{array} & \begin{pmatrix}
\ddots & \vdots & & \vdots & \\
\cdots & \diamond & \cdots & \circ & \cdots \\
 & \vdots & \ddots & \vdots & \\
\cdots & \circ & \cdots & \diamond & \cdots \\
 & & \vdots & & \ddots
\end{pmatrix}
\end{array}$$

國立暨南國際大學

A Geometric Interpretation of the Monge Property

This property is a consequence of the triangle inequality.



$$
\begin{array}{cc}
 & \begin{array}{ccccc} & j & & j' & \end{array} \\
\begin{array}{c} \\ i \\ \\ i' \\ \\ \end{array} &
\begin{pmatrix}
\ddots & \vdots & & \vdots & \\
\cdots & \diamond & \cdots & \circ & \cdots \\
 & \vdots & \ddots & \vdots & \\
\cdots & \circ & \cdots & \diamond & \cdots \\
 & & \vdots & & \ddots
\end{pmatrix}
\end{array}
$$

$$d(i,j) + d(i',j') \leq d(i,j') + d(i',j)$$

Lemma (Aggarwal and Park, 1987)

All of the row minima and column minima in an $m \times n$ *Monge matrix can be determined in time* $\mathrm{O}(m+n)$*, provided that each entry in the matrix can be accessed in time* $\mathrm{O}(1)$*.*

Remarks

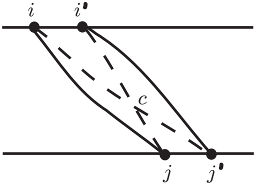
1. *When there are many minima in a row or column, we can simply choose the first one.*
2. *All of the row and column maxima can also be found in the same time bound.*

國立暨南國際大學

$$
\begin{aligned}
O_R(j) &= \min_{1\le i\le j}\{I_R(i)+DIST(i,j)\} \quad \text{for } 1\le j\le n\\
OUT(i,j) &= I_R(i)+DIST(i,j)\ .
\end{aligned}
$$

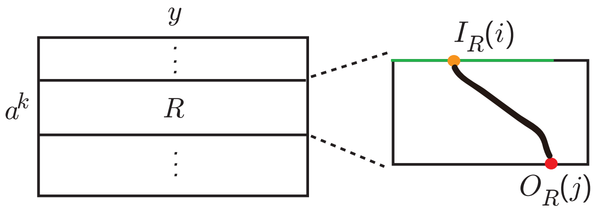
Lemma (Aggarwal and Park, 1988)

The matrices $DIST$ and OUT are Monge.



Lemma

All values on the bottom of a strip can be evaluated in $\mathrm{O}(n)$ *time.*

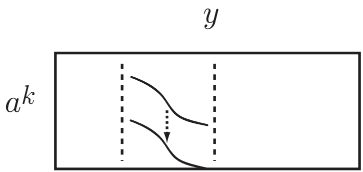


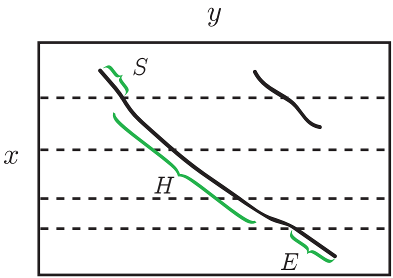
$$O_R(j) = \min_{1 \le i \le j} \{I_R(i) + DIST(i,j)\} \quad \text{for } 1 \le j \le n$$

Theorem

The edit distance problem on run-length-encoded strings can be solved in $\mathrm{O}(\min\{m'n, mn'\})$ *time.*

Local Alignment Algorithm





Question?

References I

A. Apostolico, M. J. Atallah, L. L. Larmore, and S. Mcfaddin.
Efficient parallel algorithms for string editing and related problems.
SIAM Journal on Computing, 19(5):968–988, 1990.

A. Aggarwal, M. M. Klawe, S. Moran, P. Shor, and R. Wilher.
Geometric applications of a matrix-searching algorithm.
Algorithmica, 2(1):195–208, 1987.

O. Arbell, G. M. Landau, and J. S. B. Mitchell.
Edit distance of run-length encoded strings.
Information Processing Letters, 83(6):307–314, 2002.

A. Apostolico, G. M. Landau, and S. Skiena.
Matching for run-length encoded strings.
Journal of Complexity, 15(1):4–16, 1999.

國立暨南國際大學

References II

A. Aggarwal and J. Park.
Notes on searching in multidimensional monotone arrays.
In *Proceedings of the 29th IEEE Symposium on Foundations of Computer Science (FOCS 1988)*, pages 497–512.

H. Bunke and J. Csirik.
An improved algorithm for computing the edit distance of run-length coded strings.
Information Processing Letters, 54(2):93–96, 1995.

G. Benson.
A space efficient algorithm for finding the best nonoverlapping alignment score.
Theoretical Computer Science, 145(1–2):357–369, 1995.

國立暨南國際大學

References III

W. W. Bein, M. J. Golin, L. L. Larmore, and Y. Zhang.
The Knuth-Yao quadrangle-inequality speedup is a consequence of total-monotonicity.
In *Proceedings of the 7th annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2006)*, pages 31–40.

R. E. Burkard, B. Klinz, and R. Rudolf.
Perspectives of monge properties in optimization.
Discrete Applied Mathematics, 70(2):95–161, 1996.

R. E. Burkard.
Monge properties, discrete convexity and applications.
European Journal of Operational Research, 176(1):1–14, 2007.

國立暨南國際大學

References IV

M. Crochemore, G. M. Landau, and M. Ziv-Ukelson.
A subquadratic sequence alignment algorithm for unrestricted scoring matrices.
SIAM Journal on Computing, 32(6):1654–1673, 2003.

D. Gusfield.
Algorithms on Strings, Trees, and Sequences.
Cambridge University Press, 1997.

D. S. Hirschberg.
A linear space algorithm for computing maximal common subsequences.
Communications of the ACM, 18(6):341–343, 1975.

國立暨南國際大學

References V

J. W. Kim, A. Amir, G. M. Landau, and K. Park.
Computing similarity of run-length encoded strings with affine gap penalty.
In *Proceedings of 12th String Processing and Information Retrieval (SPIRE 2005)*, volume 3772 of *Lecture Notes in Computer Science*, pages 429–435. Springer-Verlag, 2005.

S. K. Kannan and E. W. Myers.
An algorithm for locating nonoverlapping regions of maximum alignment score.
SIAM Journal on Computing, 25(3):648–662, 1996.

C. Ledergerber and C. Dessimoz.
Alignments with non-overlapping moves, inversions and tandem duplications in $O(n^4)$ time.
Journal of Combinatorial Optimization, 2007.
(to appear).

References VI

V. I. Levenshtein.
Binary codes capable of correcting, deletions, insertions and reversals.
Soviet Physics Doklady, 10:707–710, 1966.

J. J. Liu, G. S. Huang, Y. L. Wang, and R. C. T. Lee.
Edit distance for a run-length-encoded string and an uncompressed string.
Information Processing Letters, 105(1):12–16, 2007.

J. J. Liu, Y. L. Wang, and R. C. T. Lee.
Finding a longest common subsequence between a run-length-encoded string and an uncompressed string.
Journal of Complexity, 24(2):173–184, 2008.

G. M. Landau and M. Ziv-Ukelson.
On the common substring alignment problem.
Journal of Algorithms, 41(2):338–359, 2001.

國立暨南國際大學

References VII

J. Mitchell.
A geometric shortest path problem, with application to computing a longest common subsequence in run-length encoded strings.
Technical report, SUNY Stony Brook, 1997.

V. Mäkinen, G. Navarro, and E. Ukkonen.
Approximate matching of run-length compressed strings.
Algorithmica, 35(4):347–369, 2003.

J. P. Schmidt.
All highest scoring paths in weighted grid graphs and their application to finding all approximate repeats in strings.
SIAM Journal on Computing, 27(4):972–992, 1998.

R. A. Wagner and M. J. Fischer.
The string-to-string correction problem.
Journal of the ACM, 21(1):168–173, 1974.

國立暨南國際大學