Window-Chained Longest Common Subsequence: Common Event Matching in Sequences

Chunyao Song, Tingjian Ge
Department of Computer Science, University of Massachusetts, Lowell

Background and Our Proposal
Processing sequence data.

Classic LCS not working, so…?
Flexible window size, minimum density per window, maximum gap and shift make sure we find meaningful match.

How to get WCCLS?
Per window LCS
Can we just derive results from LCS?

Informed Search and Exploration
When there’s gap constraint, previous method can be costly as it may involve a large number of spatial index operations when the number of matching windows is large.

So we propose WCCLS-A*
1. We store a set of window chains (C) in a priority queue Q.
2. Priority computation: f(C) = g(C)+h(C).
   a. f(C) is an upper bound of any potential window chain derived from C.
   b. g(C) is the total matching length of the windows in C.
   c. h(C) is the upper bound of the matching length of future windows to be added to C, which is computed as h(C)=LCS(x,y)-w(x,y).
3. When Q is not empty, pop C with maximum priority.
   a. If h(C)=0, then return C.
   b. If there’s no preceding window, update h(C) to 0, and put C back with new h(C) and f(C).
   c. If there’s preceding window(s), then for each preceding window, create new C with new h(C) and f(C) and put it back to Q.

Suppose LCS(8)=9, LCS(6)=12, LCS(3)=14, LCS(9)=21, LCS(5)=28.

Approximation Algorithm
- Run Per Window LCS, sample with probability \( \rho \)
- Run WCCLS, and let the result window chain be C
- for each window \( w_i \) in chain C
  - find LCS of \( w \) within original sequences, using all the matching points in \( w_i \)
- Return C

Evaluation
Data: Stock data, GPS data, synthetic data

We cannot use LCS results 4-1=3, the actual WLCS should be of length 2.
So, we propose a non trivial extension of the LCS algorithm.