Quadratic Hardness for Sequence Problems

Arturs Backurs (MIT)

Piotr Indyk (MIT)

Plan

- Problems:
  - (Discrete) Frechet Distance
  - Edit Distance and LCS
  - Dynamic Time Warping
- Birds eye view on the upper bounds
  - Dynamic programming, quadratic time
- Recent conditional quadratic lower bounds
  - Assuming Strong Exponential Time Hypothesis, etc

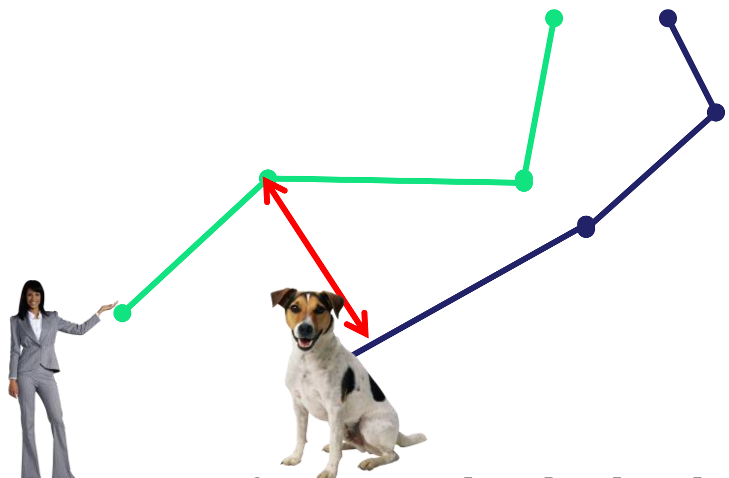
Piotr

Arturs

(Discrete) Frechet Distance

[Alt-Godau'95]

- ``Dog walking distance''
  - Smallest length leash that enables dog-walking along two routes



- Definition:
  - Let F= set of monotone functions $[0,1] \rightarrow [0,1]$
  - For two curves $P,Q: [0,1] \rightarrow R^2$:

$$D_{Fr}(P,Q) = \min_{f,g \in F} \max_{t \in [0,1]} ||P(f(t)) - Q(g(t))||$$

- Discrete version:
  - $F=\{f:[0,1] \rightarrow \{1\ldots n\}\}$,
  - $P,Q: \{1\ldots n\} \rightarrow R^2$

Frechet Distance: Algorithm

- Discrete version:
  - Let $F=\{f:[0,1] \rightarrow \{1 \ldots n\}\}$,
  - For two curves $P,Q: \{1 \ldots n\} \rightarrow R^2$:

$$D_{Fr}(P,Q) = \min_{f,g \in F} \max_{t \in [0,1]} ||P(f(t)) - Q(g(t))||$$

- Dynamic programming:
  - $A[i,j]$ = distance between $P(1) \ldots P(i)$ and $Q(1) \ldots Q(j)$
  - $A[i,j]=\max\Big[||P(i)-Q(j)||, \min(A[i-1,j-1],A[i,j-1], A[i-1,j])\Big]$
- Time: $O(n^2)$
- Can be improved to $O(n^2 \log\log n/\log n)$ [Agarwal-Avraham-Kaplan-Sharir'12] (also [Buchin-Buchin-Meulemans-Mulzer'14])
- Many algorithms for special cases and variants

Edit distance
(a.k.a. Levenshtein distance)

- Definition:
  - x,y – two sequences of symbols of length n
  - edit(x,y)=the minimum number of symbol insertions, deletions or substitutions needed to transform x into y
- Example: edit(meaning,matching)=4

meaning $\xrightarrow{\text{insert a}}$ maeaning $\xrightarrow{\text{e} \to \text{t}}$ mataning $\xrightarrow{\text{a} \to \text{c}}$ matcning $\xrightarrow{\text{n} \to \text{h}}$ matching

- Variants:
  - edit'(x,y)=the minimum number of symbol insertions or deletions needed to transform x into y
  - edit'(x,y)=2n – 2 LCS(x,y)

Computing edit distance

- A simple $O(n^2)$ time dynamic programming algorithm [Wagner-Fischer'74]
- Can be improved to $O(n^2/\log n)$ [Masek-Paterson'80]
- Better algorithms for special cases:[U83,LV85,M86, GG88,GP89,UW90,CL90,CH98,LMS98,U85,CL92,N99,CPSV00,MS00,CM02,BCF08,AK08,AKO10…]
- Approximation algorithm:$(\log n)^{O(1/\varepsilon)}$ –approx in $O(n^{1+\varepsilon})$ time [Andoni-Krauthgamer-Onak'10]

Dynamic Time Warping

- Definition:
  - x, y: two sequences of points of length n
  - $A[i,j]=||x_i-y_j||+\min(A[i-1,j],A[i-1,j-1],A[i,j-1])$
  - $DTW(x,y)=A[n,n]$
- Speech processing

- A simple $O(n^2)$ time dynamic programming algorithm

What do these problems have in common ?

- Widely used metrics
- Dynamic-programming algorithms with (essentially) quadratic running time
- We have no idea if/how we can do any better

- Plausible explanation: the problems are SETH-hard

SETH-hardness

- SETH (Strong Exponential Time Hypothesis). SAT problem cannot be solved in $2^{N(1-\Omega(1))} \cdot M^{O(1)}$ time
  - N – number of variables
  - M – number of clauses

Orthogonal Vectors Conjecture

- Orthogonal Vectors Problem. Given two sets of vectors $A,B \subseteq \{0,1\}^d$, $|A|=|B|=n$, determine whether there are $a \in A$, $b \in B$ such that $\Sigma_{i=1}^{d} a^i b^i = 0$
  - Can be solved trivially in $O(n^2 d)$ time
  - Best known algorithm runs in $n^{2-1/O(\log c(n))}$ time, where $d=c(n)\cdot\log n$ [Abboud-Williams-Yu'15]
- Conjecture: OVP cannot be solved in $n^{2-\Omega(1)}\cdot d^{O(1)}$ time

Quadratic hardness

Theorem*: No $n^{2-\Omega(1)}$ algorithm for EDIT, DTW, Frechet distances unless OVC fails [Bringmann'14; Backurs-Indyk'15; Abboud-Backurs-Williams'15; Bringmann-Kunnemann'15]

- Basic approach: reduce OVP to distance computation:
  - $A \subseteq \{0,1\}^d \rightarrow$ sequence x, $|x| \leq n \cdot d^{O(1)}$
  - $B \subseteq \{0,1\}^d \rightarrow$ sequence y, $|y| \leq n \cdot d^{O(1)}$
  - distance(x,y)=small if exists $a \in A$, $b \in B$ with $\Sigma_i a^i b^i = 0$
  - distance(x,y)=large, otherwise
  - The construction time is $n \cdot d^{O(1)}$

*See also [Abboud-Williams-Weimann'14]