"Walking a dog with shortest leash"

**Theorem:** Assuming SETH, no $O(n^{2-\delta})$ algorithm for these problems.

**History:**
- Bringmann ’14 for Frechet dist. (discrete & continuous)
- Backurs, Indyk ’15 for edit dist.
- Alhoobar, Backurs, Vassilevska ’15 for LCS
- Bringmann, Kleinemann ’15

**Reduce O(V) → Discrete Frechet (Bringmann’14)**

Suppose disc Frechet could be solved in $O(n^{2-\delta})$ time.

Given vectors $a_1, \ldots, a_n, b_1, \ldots, b_n \in \{0,1\}^d$,

define alphabet $\Sigma = \{0, 1, \#, \$, \', \%, @\}$

Define strings $A = \$ f(a_1) \$ f(a_2) \ldots \$ f(a_n)$

$B = @'g(b_1) \% g(b_2) \% \ldots \% g(b_n)'$

Where $f(a) = a(1) \& a(2) \& \ldots \& a(d)$

$g(b) = b(1) \& b(2) \& \ldots \& b(d)$

**Claim:** disc Frechet dist $< r \iff \exists$ orth pair.

**Pf:** ($\iff$) Suppose $a_i \cdot b_j = 0$.

$A = \$ f(a_1) \$

$B = @'g(b_1) \% b_j '$
then Fréchet dist \( < r \).

(\( \Rightarrow \)) Suppose Fréchet dist \( < r \).

\[
A = \begin{array}{c}
\$ f(ai) \#
\end{array}
\]

\[
B = \begin{array}{c}
@ \$ \#
\end{array}
\]

\[
g(bj) \#
\]

\[
\Rightarrow a \cdot bj = 0
\]

\[
\#
\]

Ranks:
- same pf implies hardness of \( c \)-approx for some const \( c \)
- edit dist/ LCS/ DTW "similar" but more complicated
- Abbeel et al. '16: holds for much weaker version of \( \text{SET-H} \)
  (for circuit-SAT with sublinear depth)

---

**Cond. Lower Bds via 3SUM**

3SUM Problem: Given set \( S \) of \( n \) numbers, decide \( \exists a, b, c \in S \) s.t. \( a + b + c = 0 \)

(3-Set version: given \( A, B, C \),
\[
\exists a \in A, b \in B, c \in C \text{ s.t. } a + b + c = 0
\]
(all vars. are equiv)

Conjecture: no \( O(n^{2.5}) \)-time alg\( \text{m} \) for 3SUM
(\( \ldots \) for ints)
**Conjecture**

No $O(n^{2.3})$-time algm for 3SUM

(for reals, or for ints)

(strongest: for ints in $[n^2]$).

**History:** Gajentaan-Overmars '93 in Comp geometry
(preydates sETH, #PSP Conj, etc.)

(problems reducible from 3SUM
called 3SUM-hard)

---

**Exs of 3SUM-Hard Problems in Comp. Geometry**

**3-Collinear-Pts.**

*affine, degeneracy testing*

Given set $P$ of $n$ pts in 2D,
decide $\exists$ 3 pts of $P$ lying on a common line

**3SUM $\rightarrow$ 3-Collinear-Pts:**

Given sets $A$, $B$, $C$ of $n$ numbers,
define $P$:

$$(x_1, y_1) \quad \forall a \in B$$

$$(y_2, y_2) \quad \forall c \in C$$
\[ (a, 0), (b, 1), \left(\frac{c}{2}, \frac{c}{2}\right) \text{ collinear} \]
\[ \iff a + b = c \]

3SUM \rightarrow 3\text{-Collinear-Pts}:

Given \( S \), a set of \( n \) numbers,
define \( P \):

\[ y = x^3 \]

\[ (a, a^3), (b, b^3), (c, c^3) \text{ collinear} \]
\[ \iff a + b + c = 0. \]

(extends to \( d \) dims: \( x \rightarrow (x, x^2, x^3, \ldots, x^d, x^{d+1}) \))

Weird moment curve by Jeff E.

3-Concurrent-Lines: Given \( n \) lines in 2D,
decide \( \exists 3 \) lines that intersect at common pt

(3-Concurrent-Lines \( \iff 3\text{-Collinear-Pts} \))
(3-Sets)\
3SUM \rightarrow 3-\text{Concurrent-Lines}:

\begin{align*}
  y & = 6 \\
  a \in A \\
  x & = a \\
  x + y & = c \\
  a \in c \in C.
\end{align*}

\text{Coverage:} \quad \text{Given} \ n \ \text{objects in 2D,} \\
\text{decide whether union covers a region} \\
\text{e.g. rectangle}

3SUM \rightarrow \text{Coverage}
Motion Planning: Given n objects in 2D space, decide 3 ways to move robot from one position to another.