

Hardness of Approximation in P

Open Problems

Based on talks of Amir Abboud, Bingkai Lin, Pasin Manurangsi and my* internal monologue

*Hi I'm Karthik C. S.

Outline

- Problems in fine-grained complexity
- Problems in Fine-grained complexity
- Problems in Fine-grained complexity

Problems in fine-grained complexity

- Does ETH imply Gap-ETH?
 - ★ Only serious attempt without trying to have Linear sized PCP: Applebaum'17
- Does SETH imply Gap-SETH?
 - ★ Maybe it is possible to rule out Karp reductions?
- Big-Gap-ETH: No $2^{o(n)}$ time algorithm for projection Label cover on constant sized alphabet for distinguishing 1 vs $\varepsilon \rightarrow 0$.
 - ★ Any non-trivial consequences in P?
 - ★ Bafna-Minzer-Vyas'25: No $2^{o(n/\log^C n)}$ time algorithm under ETH

Problems in Fine-grained complexity

Gap-ETH

- Set Cover

- ★ Rule out $\Omega_k(\log n)$ or even just $(\log n)^{0.01}$ factor in $n^{o(k)}$ or even just $f(k)\text{poly}(n)$ time

- Biclique

- ★ Rule out $\Omega(1)$ in $n^{o(k)}$ time

- Densest-k-subgraph

- ★ Rule out $o(k)$ or even just $k^{0.01}$ factor in $f(k)\text{poly}(n)$ time

- 2-CSP

- ★ Rule out $o(k)$ factor in $f(k)\text{poly}(n)$ time

- Clique

- ★ Cannot distinguish $n^{0.99}$ vs. k in $n^{o(k)}$ time or even $f(k)\text{poly}(n)$ time

Same Questions
but with
Weaker assumptions

ETH

- Clique
 - ★ Rule out $o(k)$ factor in $f(k)\text{poly}(n)$ time
 - ★ Rule out $\Omega(1)$ factor in $n^{o(k)}$ time
- 2-CSP
 - ★ Rule out $\Omega(1)$ factor in $n^{o(k)}$ time
- Biclique and Densest-k-subgraph
 - ★ Rule out $\Omega(1)$ factor in $f(k)\text{poly}(n)$ time
- Set Intersection
 - ★ Cannot distinguish $n^{1/k}$ vs. $\text{poly}(k)$ in $n^{o(k)}$ time

FPT Assumptions

- 2-CSP

- ★ Show approximating $\Omega(1)$ factor is W[1]-hard

- Clique

- ★ Show approximating $k^{0.01}$ factor is W[1]-hard

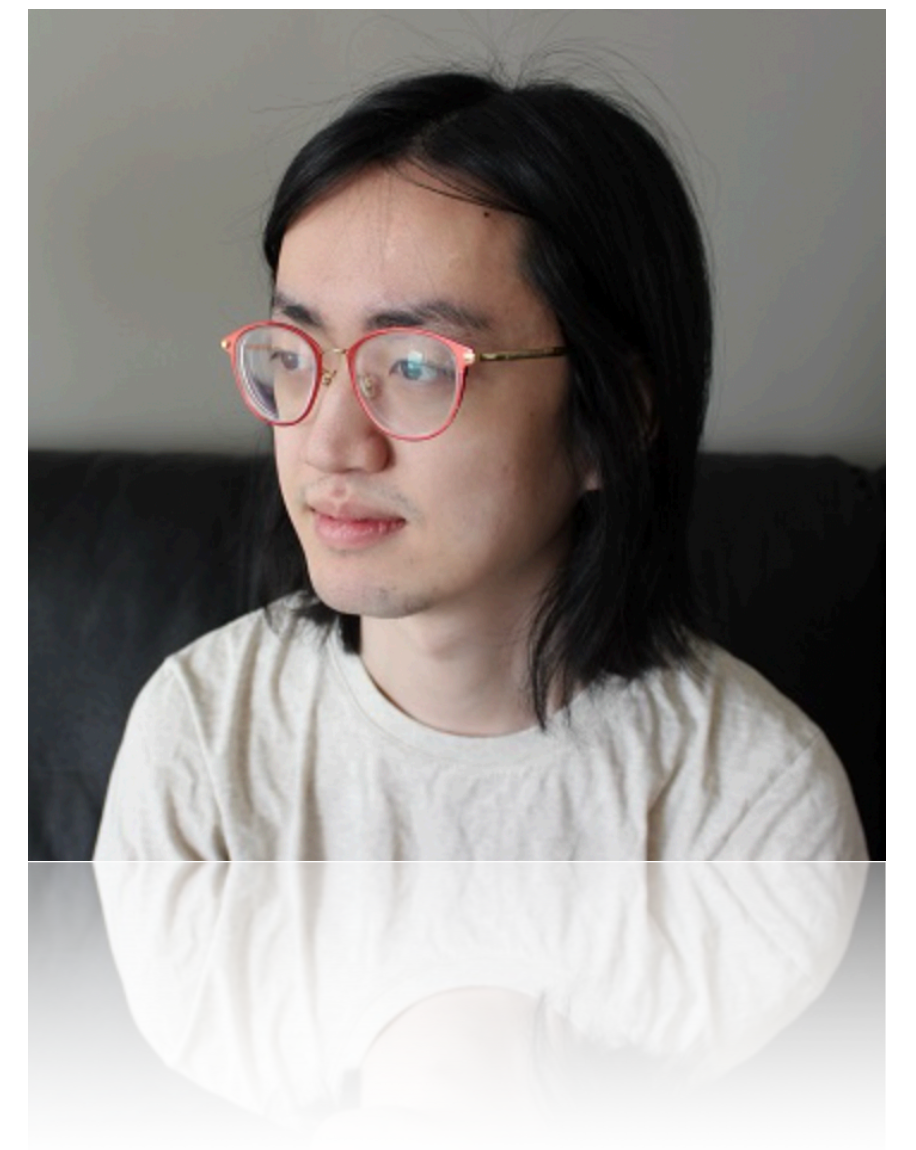
- Setcover

- ★ Show approximating $f(k)$ factor is W[2]-hard (for every f)

Problems in Fine-grained complexity

Distributed PCP Limitations

Starting from OVH on n vectors in $n^{o(1)}$ or even $(\log n)^3$ dimensions, can we prove hardness of approximation of Max Inner Product?



Distributed PCP Limitations

Hitting set conjecture: Given $A_1, \dots, A_n$ and $B_1, \dots, B_n$ subsets of $\{1, \dots, c \cdot \log n\}$, finding A_i which intersects every B_j requires $n^{2-o(1)}$ time. Does the Distributed PCP question apply to this conjecture?

Concrete target: Given $a_1, \dots, a_n, b_1, \dots, b_n \in \{0,1\}^{O(\log n)}$ decide in $n^{2-o(1)}$ time:

Yes: There is a_i such that for all b_j , we have $\|a_i - b_j\| \leq \alpha$

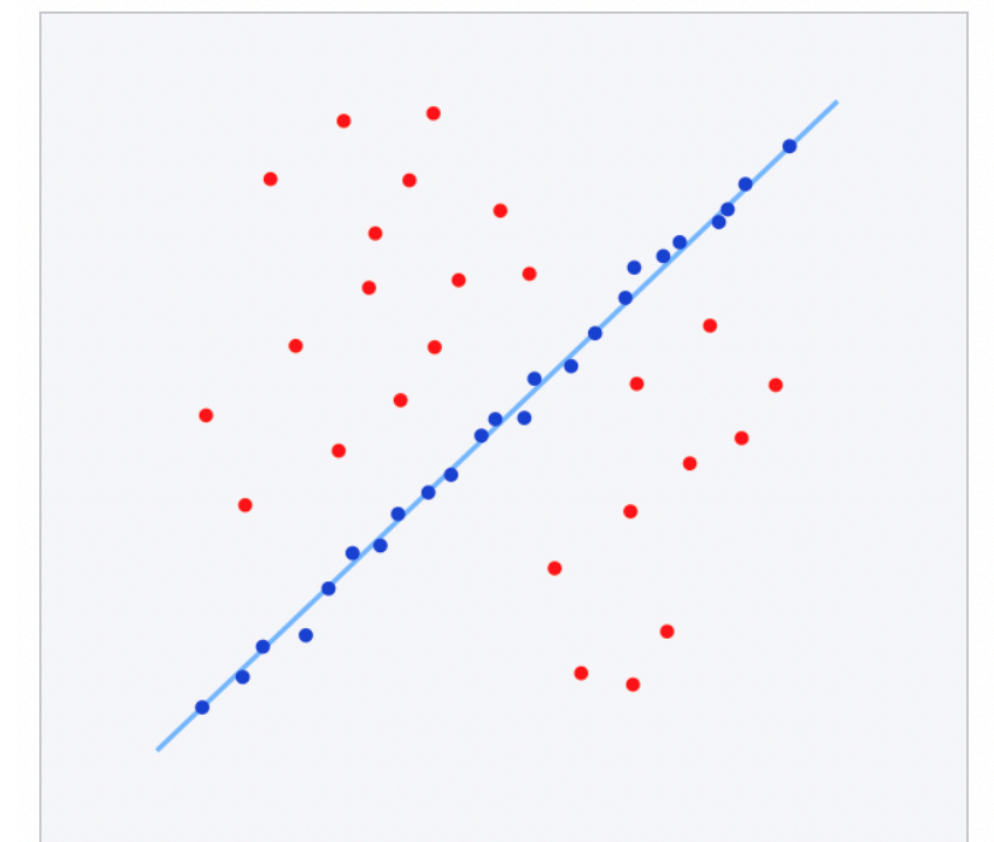
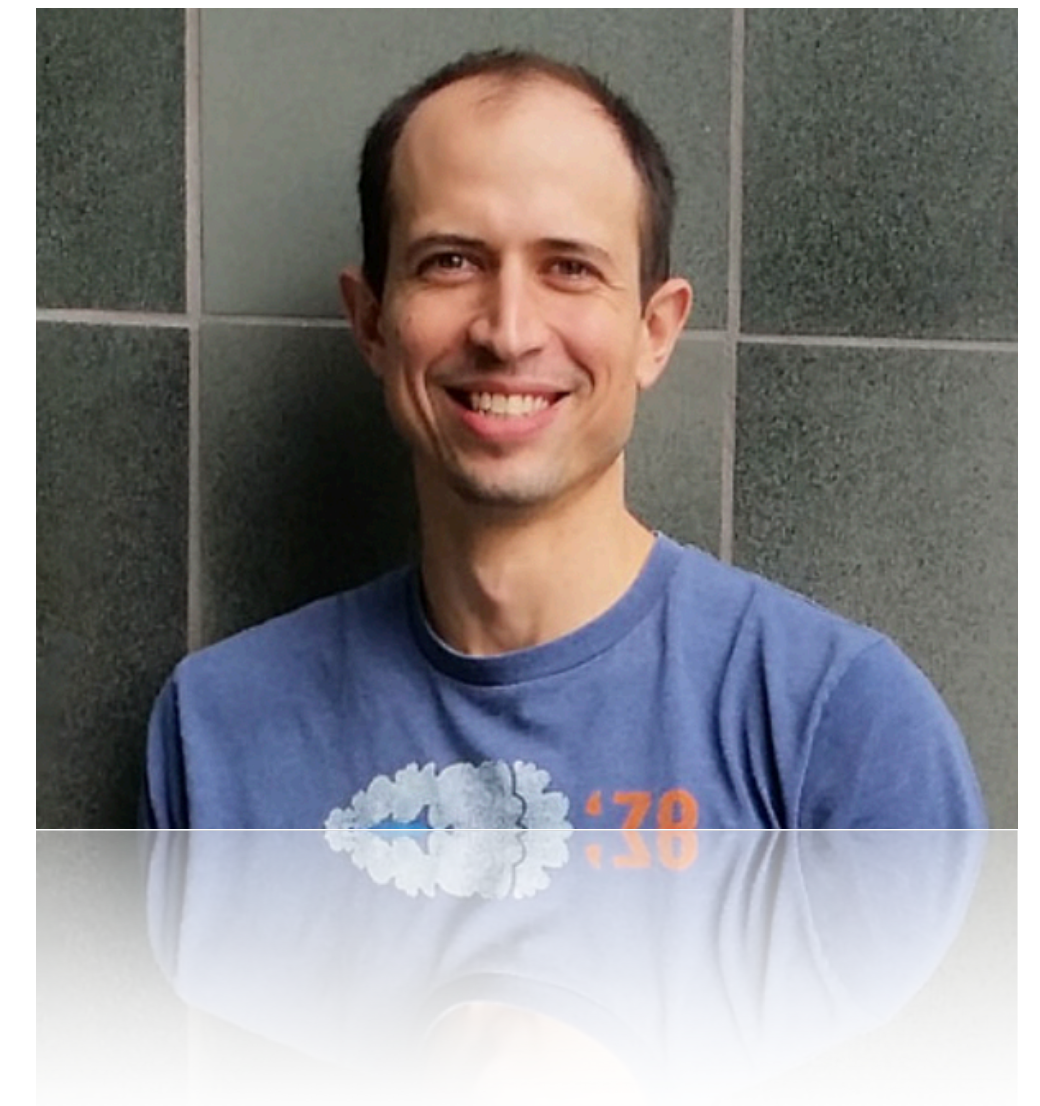
No: For every a_i there is some b_j , such that $\|a_i - b_j\| > (1 + \varepsilon) \cdot \alpha$

Closest Pair

Assuming SETH, show that for every ε there is some δ such that no algorithm running in $n^{2-\varepsilon}$ time can $1 + \delta$ approximate the closest pair of input n points in $\mathbb{R}^{O(\log n)}$.

Clique and Colinearity

- “3-clique (triangle detection) is hypothesized to require n^ω time. What about finding a triangle in a graph that contains a 4-clique? On one hand, it's hard for me to see how containing a 4-clique (or even a $\text{polylog}(n)$ -clique) would help you algorithmically. On the other hand, I have no idea how to do (very efficient) gap amplification here. Especially tricky since the graph is dense wlog.”
- Computer Vision Community: There are 100 input points on a line, find the line. QUICKLY!



Fitted line with RANSAC; outliers have no influence on the result.

Stringology

Assuming something, show that for every ε there is some δ such that no algorithm running in $n^{2-\varepsilon}$ time can $1 + \delta$ approximate the edit distance of two strings of length n .

Triangle detection in Large Even Girth

Is Triangle detection easy/easier in C_{2k} free graphs? (for $k > 2$)

Approximate OMV from OMV

OMV Conjecture: Given Boolean matrix $n \times n$ matrix M , there is no algorithm running in $n^{3-\varepsilon}$ time which takes as input $v_1, \dots, v_n \in \{0,1\}^n$, correctly computes sequentially, $Mv_1, \dots, Mv_n$ (where we take the Boolean product).

Approximate OMV Conjecture: For every ε there is some δ , given Boolean matrix $n \times n$ matrix M , there is no algorithm running in $n^{3-\varepsilon}$ time which takes as input $v_1, \dots, v_n \in \{0,1\}^n$, computes sequentially, $z_1, \dots, z_n \in \{0,1\}^n$ such that for all $i \in [n]$, we have $\|Mv_i - z_i\| \leq n^{1-\delta}$.

SETH vs. UGC

Prove that assuming the Strong Exponential Time Hypothesis, for every $\varepsilon > 0$, no algorithm running in time $n^{2-\varepsilon}$, given as input a graph on n vertices and $n^{1+o(1)}$ edges and an integer k , can distinguish between the following cases:

Completeness: There exists a vertex cover of size k .

Soundness: Every vertex cover is of size at least $1.5 \cdot k$.

Tradeoff curve for PTAS

Take any NP-hard problem with a PTAS. Suppose it's run time is $2^{1/\varepsilon} \cdot n$.
Then prove lower bound that it is tight - for all ε from $1/n$ to constant.

Candidate Problem: 1-center in Hamming metric

Don't Relax just because it's not NP-hard

Is there a well-studied NP-hard problem which is NP-hard to approximate to factor α , but can be approximated to factor $\alpha + \epsilon$, but we can show a super-linear conditional time lower bound for approximating it to a factor $\beta > \alpha$.

Illuminates hardness of easy instances in a fine-grained way.

Controversy Time

**Self Reductions
are the
Gold Standard**