Dimension as Parameter:

Today: consider algorithms that have dimension as parameter.

So both XP (\(n^k\)-type behavior) and FPT \(f(k)n^c\).

Abstract notion of dimension:

Apart from Euclidean \(L_p\) spaces, Banach spaces,

Can also define doubling dimension:

\[
\lambda = \text{smallest value such that } \forall S \subseteq X \text{ diameter } \geq D, \text{ } \exists \text{ a set of diameter } D/2 \text{ whose union covers } S.
\]

Fact: doubling dimension \((M) = \log_2 \lambda\).

Fact: \(\text{dim}(\mathbb{R}^d, L_p) = \Theta(d)\).

But now can talk about dimension of graph metric as well.

Useful fact: "doubling metrics have small nets."<br>

**def:** a \(\delta\)-net of a metric \(M\) is a subset \(N \subseteq X\) such that:

1. \(\forall x, y \in N, \ d(x, y) < \delta\) (it is a \(\delta\)-packing)
2. \(\forall z \in X, \exists y \in N \text{ s.t. } d(x, y) \leq \delta\) (it is a \(\delta\)-covering).

Can find \(\delta\)-nets greedily. (did that in the AKS algorithm last lecture.)

Fact: Suppose \(S \subseteq X\) has diameter \(D\).

And \(N\) is a \(\delta\)-net of \(S\). (actually \(N\) is a \(\delta\)-packing if \(S < \text{ suffice}\))

Then \(|N| \leq \left(\frac{D}{\delta}\right)^{\text{dim}(X)}\).

**Pf:** \(S\) can be covered by \(1 + \delta\) balls of diameter \(D/2\), \(\delta/4\), \(\delta/8\), \(\delta/16\), \(\delta/32\), \(\delta/64\), \(\delta/128\), \(\delta/256\), \(\delta/512\), \(\delta/1024\).

But each such set of diameter \(\delta/2\) contains only \(1 + \delta\) in \(N\). So \(|N| \leq \left(\frac{D}{\delta}\right)^{\text{dim}(X)}\).
Many algorithms that work for all $L_p$ spaces extend naturally to doubling spaces.

But others—like the k-means algo—are much more rooted in Euclidean space.

Today's plan:
1. Arora's TSP for Euclidean space
2. Spanning for Euclidean space

$\Rightarrow$ FPT algo for TSP in Euclidean space

These extend to doubling spaces too—ill give the high level ideas on how they translate.

**TSP in $\mathbb{R}^d$:**

**Arora '98, Mitchell '96:** TSP $(1+\varepsilon)$ approx. in time $n^{O(\varepsilon)}$ for $\mathbb{R}^2$

$2^{\log n} \text{ poly}(1/\varepsilon)$ for $\mathbb{R}^d$.

**Arora '96:** $n^{(\log n)/2}$ for $\mathbb{R}^d$.

**Rao-Smith '99:** $O(n \log n + n^{2(\log n/\varepsilon)})$ for $\mathbb{R}^d$.

Extends to general framework for many other geometric apx.

**Thm:** $2^{(\log n/\varepsilon)}$ also for TSP, with cost $(1+\varepsilon)$. [Randomized]

**Idea:** Show that exists $(1+\varepsilon)$ solution to TSP with nice properties.

Use DP to find such a TSP solution.

**Step 0:** Assume that all points are
1. at distance $\geq 1$ and
2. distance $\leq L = n^{1/\varepsilon}$ along each coordinate.

By scaling, make max dist = $L$, along some coordinate direction.

And round each point's coordinates to integers.

$\Rightarrow$ each pt moves $\leq n^{1/2}$

So OPT tour changes by at most $n^{1/2} L$. = nd

$\Rightarrow$ OPT $\geq L \cdot nd = L n^{1/2}$.

New tour $\leq (1+\varepsilon) \cdot nd$ OPT. ☺️
OK, so intcale words, and bended in L-sized box.

**Step 1:**

- Do a hierarchical path only (quad hex) on this box. Randomly.

- Fix a $2L \times 2L \times \ldots \times 2L$ box, put our $L^d$ box randomly, uniformly.

  Choose position for the anchor in the left bottom part of this bigier box.

  \[ \text{[anchor randomly placed, uniformly]} \]

From perspective of the $L^d$ box, we've picked a uniformly shifted grid system.

So have broken the space into quads.

**Step 2:**

- Put down "portals" on the boundaries.

- At distance $\delta$ from each other.

  (ie. pride a $\delta$-net on the boundary).
Step 3: So far we've not done anything that changes OPT.

Finally: if tour (OPT) crosses boundary, move to closest portal.

Since $δ$-net, total movement $\leq 2δ$ per crossing

$$\Rightarrow [\text{New tour cost}] \leq \text{OPT} + \frac{2\delta}{\text{crossings}}.$$  

Want this to be small

$$\leq \frac{3}{2} \text{OPT}. 
\text{(Lemma 1)}.$$  

Step 4: Suppose good event happens, 3 tours of length $(1+\varepsilon)\text{OPT}.$

Let's find it wavy DP.

[because it is portal respecting, this is doable].

Why? Fact: Can imagine that tour crosses each portal at most twice,
at most once in each direction.

... but may have

...
So for each cell:

"signature" = how many times does a tour enter/exit each portal

\[ \text{#signature} = 2^{(2 \times \text{#portals})} \]

Want cheapest collection of paths that satisfy signature and cover all the points within the cell.

Given the table for the smaller cells (2^d of them).

Guess the signatures on inside boundaries and enumerate, make sure we get connectivity.

Time = \(2^{O(\text{#portals} \times \text{any two levels})} \cdot \text{poly}(n)\).

Natural basin: More portals gives smaller error in Step 3.

But larger runtime in Step 4 (DP).

Is there a sweet spot?

Lets choose params: \(s^n_i = \left(\frac{\text{side length}}{\text{diam of cell at level } i}\right) \left(\frac{\varepsilon}{d \log d}\right)\).

\[ \Rightarrow (L/2)^d \]

How many portals? All of them within set of diam \( \leq d. \) (side length) \( \Rightarrow \)

And min distance \( \geq (\text{side length}) \times \left(\frac{\varepsilon}{d \log d}\right) \cdot \)

\[ \Rightarrow \text{Nets lemma: portals } \leq \left[ \frac{d}{(d \log d)} \right]^d \Rightarrow \left(\frac{d^2 \log d}{\varepsilon}\right)^d \]
For our purposes, imagine \( d \ll \log (\frac{\%}{e}) = L \).

\[ \Rightarrow \text{# portals} \leq \left( \frac{\log n}{2} \right)^{O(d)} \Rightarrow \text{runtime} = 2^{\left( \frac{\log n}{2} \right)^{O(d)}}. \]

---

**OK:** back to proving Lemma 1:

**Lemma 1:** \( E[\text{increase in length due to defours}] \leq \left( \frac{e}{2} \right) \cdot \text{OPT}. \)

**Proof:** For any edge of the original tour.

Imagine breaking it into pieces of length \( \frac{1}{2} \), say.

Now for a mini-edge.

Can lie in at most one cell boundary.

And we get a defour if it crosses cell boundary at level \( i \) of \( d_i \).

\[ \text{Pr}[\text{a defour}] \leq \frac{\text{len}(a,b) \cdot d}{(2L)^i} \leq \frac{\text{len}(a,b) \cdot d}{O(\text{side length} \cdot \text{level} \cdot i \cdot \text{cell})} \]

If so, defour is \( 2d_i = 2L \cdot (\text{side length}) \cdot \frac{e}{(d \log L)} \)

\[ \Rightarrow E[\text{defour at level } i] = \text{len}(a,b) \cdot d_i \cdot 2L \cdot \frac{e}{O(d \log L)} = \text{len}(a,b) \cdot \frac{e}{(\log L)} \]

\[ = \text{len}(a,b) \cdot \frac{e}{(\log L)} \]

\[ \Rightarrow E[\text{total defour}] = \sum_{i} \text{len}(a,b) \cdot \frac{e}{(\log L)} = O(e) \cdot \frac{\log L}{(\log L)} = O(e) \cdot \frac{\log L}{L} \]

Summing over all edges, \( O(e) \cdot \text{OPT}. \) 😊
More careful accounting can give \( \frac{O(\log d)}{\epsilon} \). and hence \( \frac{O(\theta)}{\epsilon} \) for \( d=2 \).

**Lemma (Patrascu):** Not only does \( \text{OPT} \) tour cross the cell boundary \( \leq 2 \) times at each portal, it also crosses cell boundary a total of \( O(\theta \epsilon) \) times.

Hence can reduce the DP time from \( 2^\frac{1}{\epsilon} \) to \( \left( \frac{\log n}{\epsilon} \right)^d \) times.

\[
\frac{n!}{\frac{\log n}{\epsilon}} \leq (\theta \epsilon)^{\frac{d}{\epsilon}}.
\]

[Arora 98]

**Next idea:** Esparza

Relating to bereavement

Note that the 1st level portal rules break the point set into "smaller" pieces, which are then broken by 2nd level separators, etc.

Using this idea, it is possible to show that if we make a graph on the \( n \) points plus the portals, the bereavement of the graph is only \( O(\frac{\theta}{\epsilon}) \text{ rad} \).

And then we can directly appeal to results for TSP on both TW graphs!
Spanners: graph $G$ is a spanner for metric $M=\langle V, d \rangle$ if $G=(V,E)$ satisifies:
\[ d_G(x,y) \leq (1+\epsilon) d(x,y) \quad \forall x,y \in V. \]

Clearly: if $G$ is complete (dense), this is borne.
And we need $\Omega(n^2)$ edges if $M$ is uniform, say

Thus: Any metric into doubling dimension $\delta$
has a $(1+\epsilon)$ spanner $G$ s.t. $\#E(G) \leq n(\delta,\epsilon)$ is linearized.

Today: slightly weaker bound:
\[ \#E = n(\frac{\epsilon}{2}) \cdot \log (\frac{\text{max distance}}{\text{min distance}}) \quad (\text{say } \epsilon \leq \frac{1}{2}) \]

Construction:
Say min distance = 1, max distance = $L$.

Start with $V_0=V$.
Let $V_i$ be a $2^i$-net of $V_{i-1}$. \[ i=1,2,... \quad \text{(wrt distances $d$).} \]
For each $x \in V_i$, add edges to all $y \in V_i$ s.t.
\[ d(x,y) \leq (\frac{100}{L}) \cdot 2^i. \]

Facts (1) $V_0 \subset V_2 \subset ... \subset V_{\log_2 L}$ such has a finite node.
(2) Min distance $d(x,y)$ for $x,y \in V_i \geq 2^i$.

So each node in $V_i$ adds only
\[ (\frac{100}{L} \cdot 2^i)^{O(d)} = (\frac{1}{L})^{O(d)} \]
\[ \Rightarrow \text{total } \# \text{ edges} \leq \sum_i |V_i| \cdot (\frac{\epsilon}{2})^{O(d)} \leq O(n \log L). \]

(3) "Stretch is small." take $x,y$, with distance $\in \left[ \frac{\delta \epsilon}{3}, \frac{2 \epsilon}{3} \right]$ say.

For each $x \in V_i$, let $\phi(x) \in V_{i+1}$ be the closest net point to $x$ in $V_{i+1}$. 

\text{Check details...}
So follow \( \phi \) maps from \( x \) to level \( i \) \( \xrightarrow{\phi_i} \overline{y} \)

\[ x \xrightarrow{\phi_1} x_1 \xrightarrow{\phi_2} x_2 \ldots \xrightarrow{\phi_i} \overline{y} = \overline{x} \]

\[ d(x, \overline{x}) \leq 1 + 2 + 4 + \ldots + 2^i \leq 2^{i+1} \]

\[ \Rightarrow d(\overline{x}, \overline{y}) \leq d(x, \overline{x}) + d(x, y) + d(y, \overline{y}) \]

\[ \leq 2^{i+1} + 2^{i+1} + 2^{i+1} \leq 2^{i+1} (2 + \varepsilon^i) \]

\[ \leq 2^{i+1} (\varepsilon + \varepsilon^i) \]

\[ \Rightarrow \overline{x}, \overline{y} \text{ have a direct edge between them in } V_i \]

\[ \Rightarrow \text{ this path } x \xrightarrow{\phi_i} \overline{x} \xrightarrow{\phi_i} \overline{y} \xrightarrow{\phi_i} y \text{ in a path in } G^\varepsilon \]

\[ \leq d(x, \overline{x}) + d(x, \overline{y}) + d(\overline{y}, y) \]

\[ \leq 2d(x, y) + 2(d(x, \overline{x}) + d(y, \overline{y})) \]

\[ \leq 2\overline{d}(x, y) + 2 \cdot 2^{i+1} \leq (1 + O(\varepsilon)) d(x, y). \]

\[ \square \]

**Rao Smith:** Build good spanner on the points so that

1. \((+\varepsilon)\) stretch, so that TSP tour in spanner is still good.

2. Total weight of spanner edges = \( O(\varepsilon) \cdot \text{OPT} \).

New property, we're not shown this.

Now: Do partition on this spanner, so that good tour leaves each cell only at \( O(\varepsilon)^d \) locations (where spanner crosses cells).

\[ \Rightarrow \text{DP also takes time } O(\varepsilon n \log n) + O(n \cdot \exp(\varepsilon^d)). \]
Other results for Euclidean TSP

Approximate

- [Trevian] a doubly exponential dependence required on \( \log n \) in \( d \).
- Can actually get \( 2^{O(dn^{1/2})} \) n time. [Bartal Gottlieb]
- \( \text{Unian} \)

Exact:

[de Berg et al. FOCS 19]

\( 2^{O(n^{1/2})} \) algo, and \( n \) \( 2^{o(n^{1/2})} \) algo unless ETH fails.

See also [Manu & Sidrapopoulos SODA 19] and before all this:

[Karop's] algorithm for random point sets in the plane
inspired Arora's TSP.