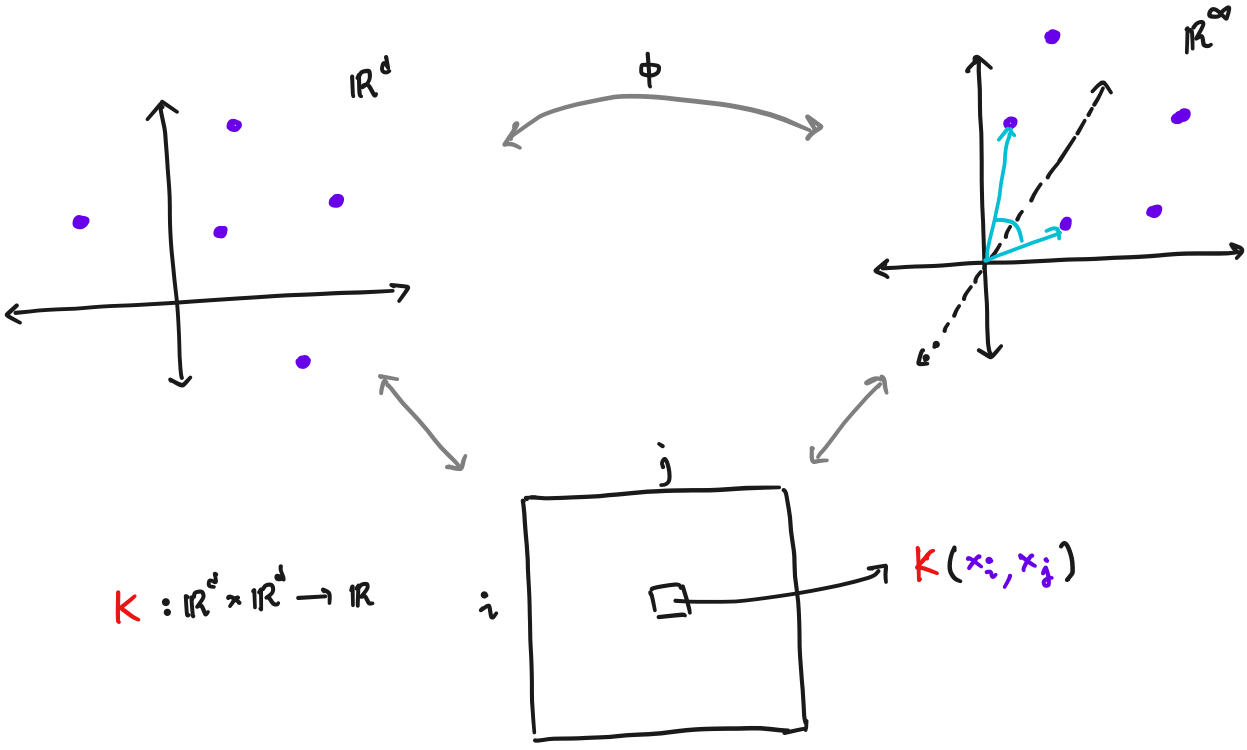
A Quasi-Monte Carlo Data Structure For Smooth Kernel Evaluations

Joint with Moses Charikar (STANFORD)
Michael Kapralov (EPFL)

RESEARCH DIRECTION: How to compute in (HIGH DIMENSIONAL) GEOMETRIC spaces.

"KERNEL TRICK":



MOTIVATING QUESTIONS:

1. Algorithmic tradeoffs? Approximation, dimensionality ... etc.
2. Are different kernels easier/harder?
3. Achieve (near)-linear time?

Problem Setup:

- Kernel function $K: \mathbb{R}^d \times \mathbb{R}^d \to [0,1]$
  (PSD, "radially decaying")
- Dataset of vectors $p_1, \dots, p_n \in \mathbb{R}^d$.
- Goal: Produce data structure s.t. on input $q \in \mathbb{R}^d$, can output

$$\frac{1}{n} \sum_{i=1}^{n} K(p_i, q)$$

(approximately + w.h.p).

Ex: Gaussian kernel: $K(p,q) = \exp(-\|p-q\|_2^2/2)$

t-Student kernel: $K(p,q) = \frac{1}{1+\|p-q\|_2^2}$.

Approx. Guarantees:

- accuracy $\varepsilon > 0$, failure probability $\delta > 0$
- Fix query $q \in \mathbb{R}^d$:

μ = true answer $\triangleq \frac{1}{n} \sum_{i=1}^{n} K(p_i, q)$

$\hat{\mu}$ = r.v output

- $\Pr\left[(1-\varepsilon)\mu \le \hat{\mu} \le (1+\varepsilon)\mu\right] \ge 1-\delta$.

Parameters

- Dataset size n in dimension d
- Accuracy ε, Failure probability δ
- True answer μ
- Query time + Space complexity.

Curse of dimensionality

[Greengard-Rokhlin '83, Alman-Chu-Schild-Song '20]

FMM: $\log^{\Theta(d)}(1/\varepsilon\mu)$ query time.

$n \log^{\Theta(d)}(1/\varepsilon\mu)$ space complexity

SETH: require $n^{1-o(1)}$ query time when $d \ge \log n$ for any $\mathrm{poly}(n)$-space complexity for "high" accuracy $\log(1/\varepsilon)$-dependence.

Theorem: [This work]

For radially decaying smooth kernels $K: \mathbb{R}^d \times \mathbb{R}^d \to [0,1]$

- Query time: $\frac{\mathrm{poly}\left(d \log\left(1/\mu\right)\right)}{\varepsilon}$

- Space complexity: $n \cdot \frac{\text{poly}\left(d \log\left(1/\mu\right)\right)}{\varepsilon}$

↳ [Charikar - Siminelakis '17, Backurs - Charikar - Indyk - Siminelakis '18, ... Phillips - Tai '20, ...]

Smoothness: $k(x,y)$ decays polynomially with $\|x-y\|_2$

ex: 2-Student kernel $\leftrightarrow$ $k(x,y) = \frac{1}{1+\|x-y\|_2^2}$

non-ex: Gaussian kernel $\leftrightarrow$ $k(x,y) = \exp\left(-\frac{\|x-y\|_2^2}{2}\right)$.

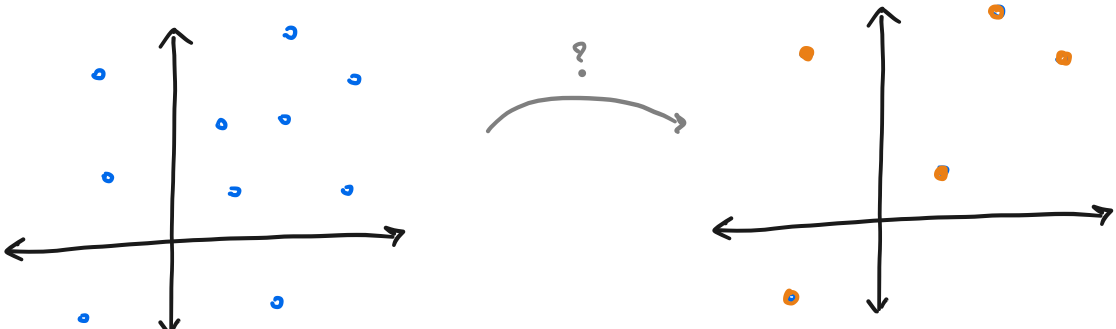
[Backurs - Charikar - Indyk - Siminelakis '18]

Smoothness is algorithmically useful primitive.

↳ $\frac{\text{poly}(\log(1/\mu))}{\varepsilon^2}$ vs. $\frac{\mu^{-0.173}}{\varepsilon^2}$

Sparsification, Coreset, Sampling ...

Which are my important data points?



$$\mu = \frac{1}{n}\sum_{i=1}^{n} k(p_i, q) \qquad \approx_{\pm \varepsilon\mu} \qquad \sum_{j=1}^{s} w_j \cdot k(p_j, q).$$

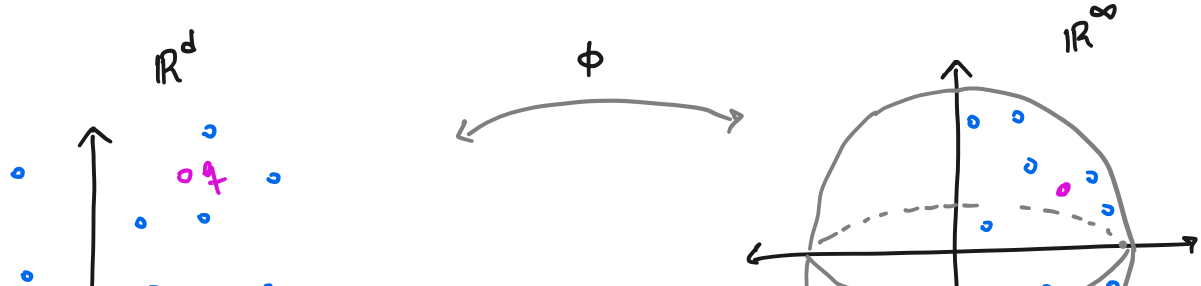
Outline

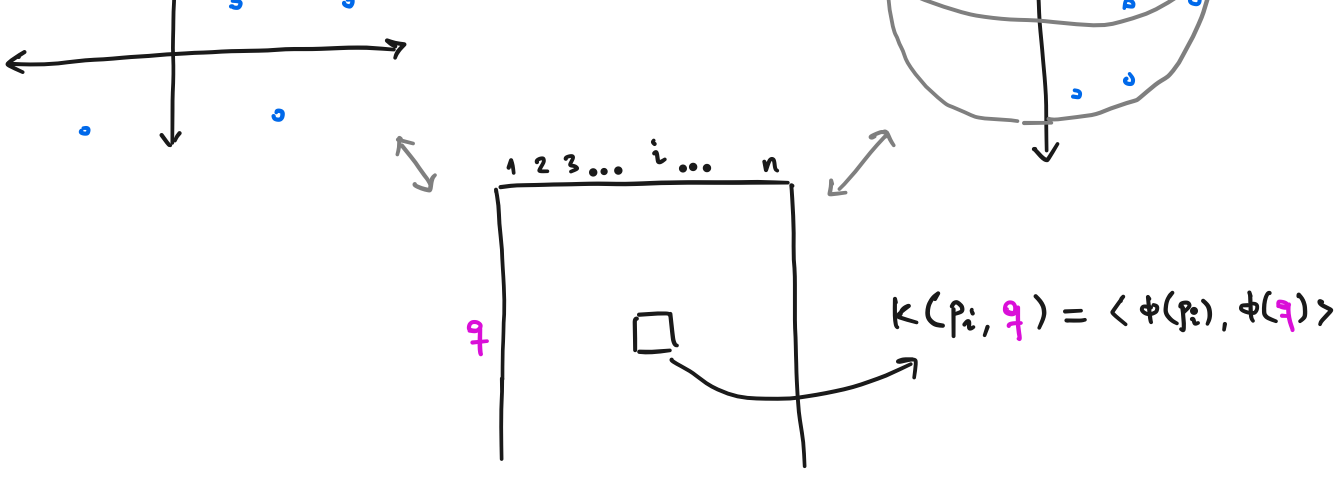
Benchmark 0: Random Sampling $d/\varepsilon^2\mu$

Benchmark 1: Discrepancy [Phillips-Tai '20] $\frac{d}{\varepsilon\mu}$

Our Approach: LSH + Discrepancy $\frac{\text{poly}(d\log(1/\mu))}{\varepsilon}$.

Discrepancy for Coresets [Phillips-Tai '20]





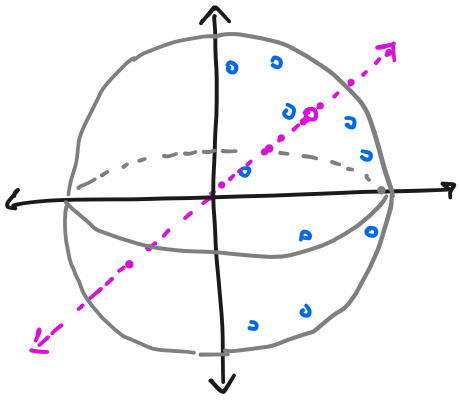
Discrepancy: $x = [\,\cdots\,] \in \{-1,1\}^n$

s.t

$$\left| \sum_{i=1}^{n} x_i \cdot K(p_i, q) \right| = \left| \sum_{x_i=1} K(p_i, q) - \sum_{x_i=-1} K(p_i, q) \right|$$

$\leq c.$ w.h.p

[Banaszczyk '98]

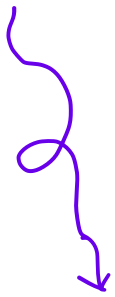


Vector Balancing Question

$$v = \sum_{i=1}^{n} x_i \phi(p_i)$$

s.t $|\langle v, \phi(q) \rangle|$ is small.

If p_i are q, becomes 1D problem



"Self-Balancing Walk" [Alweiss-Liu-Sawhney '20]

$$\text{discrepancy} \lesssim \max_{i \in [n]} \|\phi(p_i)\|_2 \, \|\phi(q)\|_2 .$$

Conceptual Question:

Kernel $K: \mathbb{R}^d \times \mathbb{R}^d \to [0,1]$ $\longleftrightarrow$ Feature map ϕ

Kernel $K: \{p_1, \dots, p_n\} \times \mathbb{R}^d \to [0,1]$ $\longrightarrow$ "Better" feature map ϕ' ?

$$\max_i \|\phi'(p_i)\|_2 \, \|\phi'(q)\|_2 \lesssim \mu.$$

What we show:

[Main Technical claim - Informal] For dataset P, smooth kernel K

$\exists$ random partition of $\mathbb{R}^d = R_1 \cup \dots \cup R_\ell$

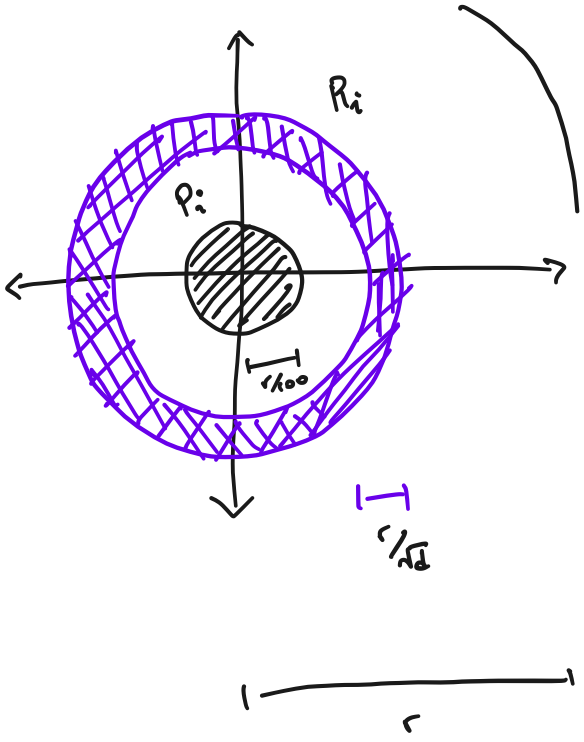
and $P = P_1 \cup \dots \cup P_\ell$

and decomposition of

s.t efficiently identifiable + stored (via decision tree in time $\text{poly}(d \log(1/\mu))$)

$k: P_i \times R_i \to [0,1]$ has feature map ϕ' s.t

$$\|\phi'(p)\|_2 \|\phi'(q)\|_2 \le \text{poly}(d \log(1/\mu)) * k(p,q).$$



Sweet Spot

1. Algorithmically decompose $\mathbb{R}^d$ into few regions.
2. Admit better feature maps.

Tradeoff: decay + separation

Open:

- Gaussian i.e "non-smooth" kernels.
- More complex kernel computations?
- Overcoming curse of dimensionality?

Thanks!