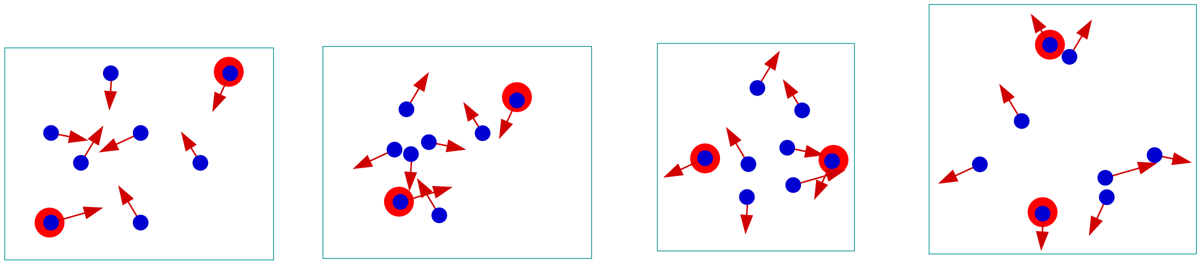
Geometric Approximation Using Coresets

Pankaj K. Agarwal

BioGeometry

Department of Computer Science
Duke University

Kinetic Geometry



S: Set of n moving points in $\mathbb{R}^2$

- $p_i = a_i + b_i t$

Maintain the diameter (width, smallest enclosing disk) of S.

☆ [A., Guibas, Hershberger, Veach]

- Diametral pair can change $\Theta(n^2)$ times
- Kinetic data structure with $\approx n^2$ events

☆ *Can we maintain the approximate diameter of S more efficiently?*

- Is there a small *coreset* $Q \subseteq S$ s.t.
  $\mathrm{diam}(Q(t)) \geq (1-\varepsilon)\,\mathrm{diam}(S(t))$?

☆ *Kinetic bounding box hierarchies?*

Shape Fitting

S: Set of n points in $\mathbb{R}^3$

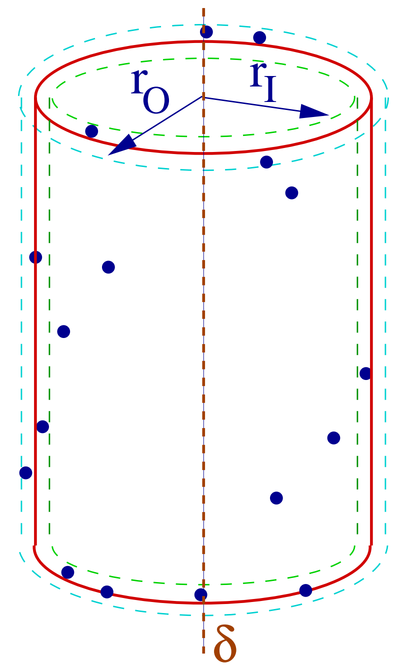
☆ *Fit a cylinder through S*

- Find a cylinder C^*

  $C^*(S) = \arg\min_{C} \max_{p \in S} d(p, C)$

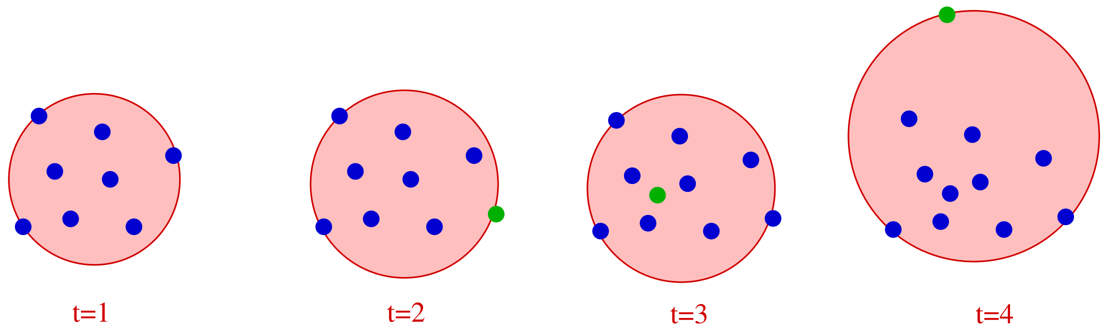
- Optimal solution: n^4 [A., Aronov, Sharir]
- $O(1)$-approximation: $\approx n^2$

☆ *Can we compute an ε-approximation of $C^*(S)$ in linear time?*



Is there is a small coreset $Q \subseteq S$ so that $C^*(Q)$ approximates $C^*(S)$?

Geometry in Streaming Model



- ☆ An incoming stream of points in $\mathbb{R}^d$
- ☆ Maintain certain geometric/statistical measures of the input stream
  - Diameter, smallest enclosing disk, k-clustering
- ☆ Use $\log^{O(1)} n$ space and processing time
- ☆ Much work done on maintaining a summary of 1D data
- ☆ Little work on multidimensional geometric data
  [A., Krishnan, Mustafa, Venkatasubramanian], [Hershberger, Suri], [Bagchi, Chaudhary, Eppstein, Goodrich]
- ☆ *How much storage and processing time (per point) needed to maintain ε-approx of smallest disk enclosing S?* Maintain a *core set*!

BioGeometry

ε-Approximation and Random Sampling

☆ $X = (S, R)$, $R \subseteq 2^S$: Set system (range space)

- δ: VC-dimension of X

☆ $A \subseteq S$ *ε-approximation* if for all $r \in R$

$$\left| \frac{|r|}{|S|} - \frac{|r \cap A|}{|A|} \right| \leq \varepsilon$$

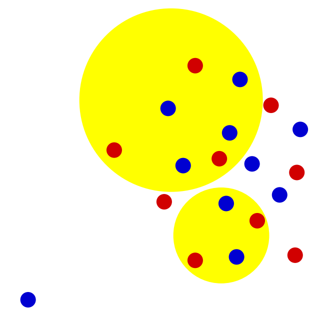
☆ A random subset $A \subset S$ of size $\frac{\delta^2}{\varepsilon^2} \log \frac{\delta}{\varepsilon}$ is an ε-approximation of S with high probability [Vapnik-Chervonenkis]

☆ Efficient deterministic algorithms for computing an ε-approximation [Matoušek, Chazelle]

ε-Approximations

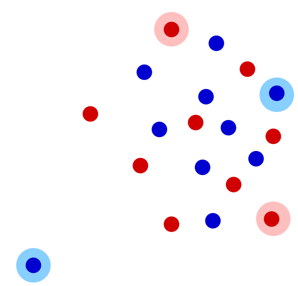
☆ An ε-approximation is a coreset of S in a *combinatorial* sense

- S: Set of points in $\mathbb{R}^2$
- $R = \{r \cap S \mid r \text{ is a disk}\}$
- A: an ε-approximation of (S, R)
- A approximates $|S \cap r|$



☆ A is not a coreset of S in a metric/geometric sense

- $\text{diam}(A)$ does not approximate $\text{diam}(S)$
- A best-fit circle for A does not approximate the best-fit circle for S



What about other sampling schemes?

Unified Framework for Coresets

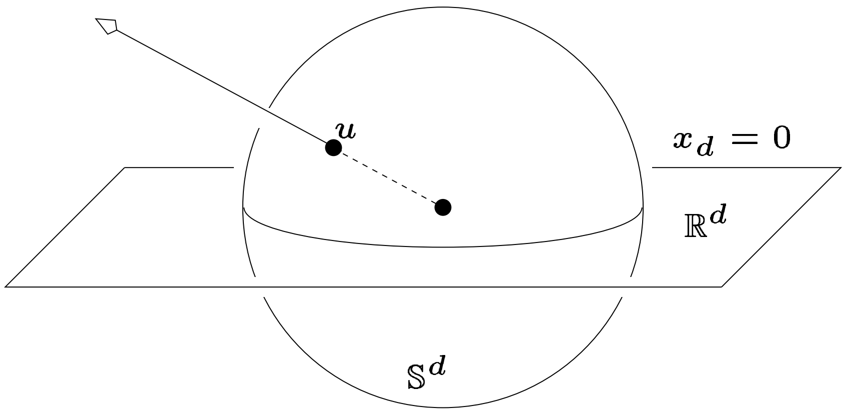
- ☆ Notion of coreset is problem specicific
- ☆ *Is there a unified framework that constructs coresets for a wide class of problems?*
  - Random subset is an ε-approximation for a large class of range spaces!
  - Easy to compute

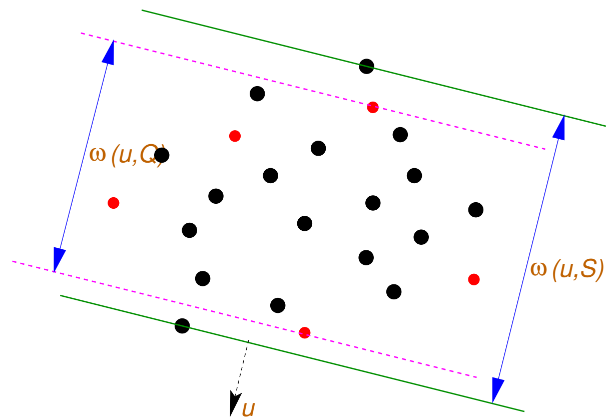
Define the notion of *ε-kernel*

- ☆ Core set for a wide class of problems

ε-Kernels

S: Set of points in $\mathbb{R}^d$





Directional width: For $u \in \mathbb{S}^{d-1}$,

$$\omega(u, S) = \max_{p \in S} \langle u, p \rangle - \min_{p \in S} \langle u, p \rangle$$

ε-kernel: $Q \subseteq S$ is an ε-kernel of S if

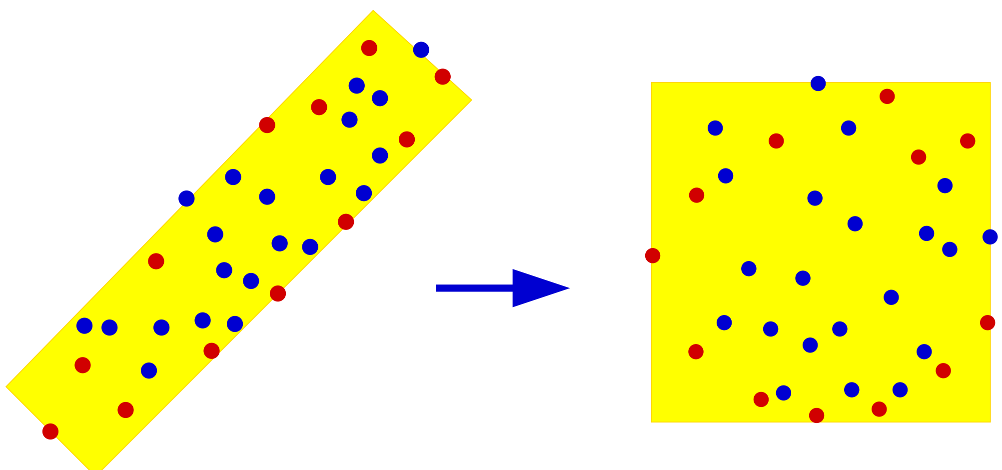
$$\omega(u, Q) \geq (1 - \varepsilon)\omega(u, S) \qquad \forall u \in \mathbb{S}^{d-1}$$

Computing ε-Kernels

Theorem A: [AHV, YAPV, Ch] $S \subseteq \mathbb{R}^d$, $\varepsilon > 0$. *We can compute an ε-kernel of S of size* $1/\varepsilon^{(d-1)/2}$ *in time* $n + 1/\varepsilon^{d-3/2}$.

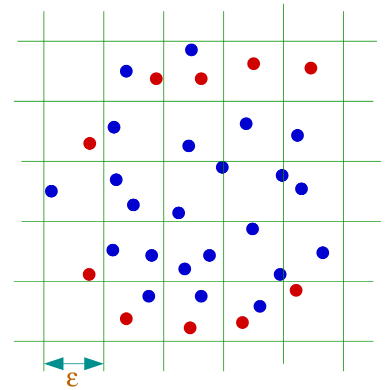
Lemma 1: $\exists$ affine transform M s.t.

- ☆ Unit hypercube $[-1, +1]^d$ is the smallest box enclosing S
- ☆ $M(S)$ is fat
- ☆ Q is an ε-kernel of $S \Leftrightarrow M(Q)$ is an ε-kernel of $M(S)$



Computing ε-Kernels

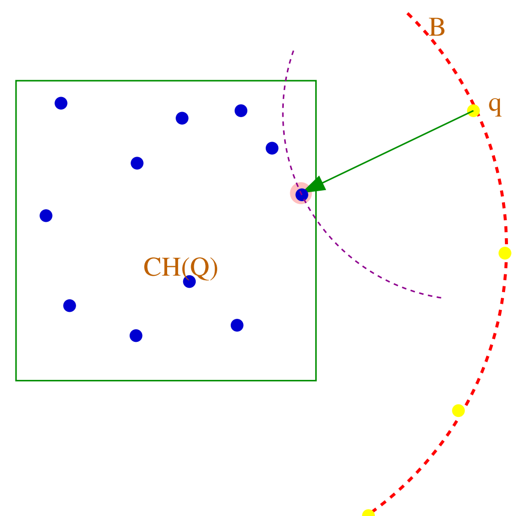
Lemma 2: S: Set of n fat points in $[-1,+1]^d$, $\varepsilon > 0$. We can compute an ε-kernel of S of size $1/\varepsilon^{(d-1)/2}$ in time $n + 1/\varepsilon^{d-3/2}$.



Sketch: Algorithm in two phases

- ☆ Compute $1/\varepsilon^{d-1}$-size approximation Q
- ☆ Draw a sphere B of radius 2 centered at origin
- ☆ Draw a grid of size $1/\varepsilon^{(d-1)/2}$ on B
- ☆ For each grid point q, select its nearest neighbor in Q

  (Suffices to compute approximate NN.)



Kernel Computation

☆ Computing ε-kernels for $\varepsilon < 0.05$ on various synthetic inputs

Input Type	*Input Size*	*Kernel Size*			
		$d = 2$	$d = 4$	$d = 6$	$d = 8$
sphere	10,000	10	994	$7,773$	$6,983$
	100,000	10	$1,824$	$22,392$	$57,276$
	$1,000,000$	10	$1,982$	$38,836$	$139,340$
cylinder	10,000	6	367	$3,834$	$6,320$
	100,000	6	859	$8,857$	$49,203$
	$1,000,000$	6	451	$12,717$	$127,385$
clustered	10,000	8	235	718	$2,502$
	100,000	12	326	$1,483$	$7,614$
	$1,000,000$	12	140	$1,554$	$13,781$

Kernel Computation: Running Time

Input Type	Input Size	$d = 2$		$d = 4$		$d = 6$	
		Prepr	*Query*	*Prepr*	*Query*	*Prepr*	*Query*
sphere	10,000	0.03	0.01	0.06	0.05	0.10	9.40
	100,000	0.54	0.01	0.90	0.50	1.38	67.22
	1,000,000	9.25	0.01	13.08	1.35	19.26	227.20
cylinder	10,000	0.03	0.01	0.06	0.03	0.10	2.46
	100,000	0.60	0.01	0.91	0.34	1.39	30.03
	1,000,000	9.93	0.01	13.09	0.31	18.94	87.29
clustered	10,000	0.03	0.01	0.06	0.01	0.10	0.08
	100,000	0.31	0.01	0.63	0.02	1.07	1.34
	1,000,000	5.41	0.01	8.76	0.02	14.75	1.08

- ☆ Prepr: Time (in secs.) in converting input into a fat set and preprocessing for nearest-neighbor (NN) queries
- ☆ Query: Time (in secs.) in answering NN queries
- ☆ Experiments run on Pentium IV 3.6GHz processor with 2GB RAM

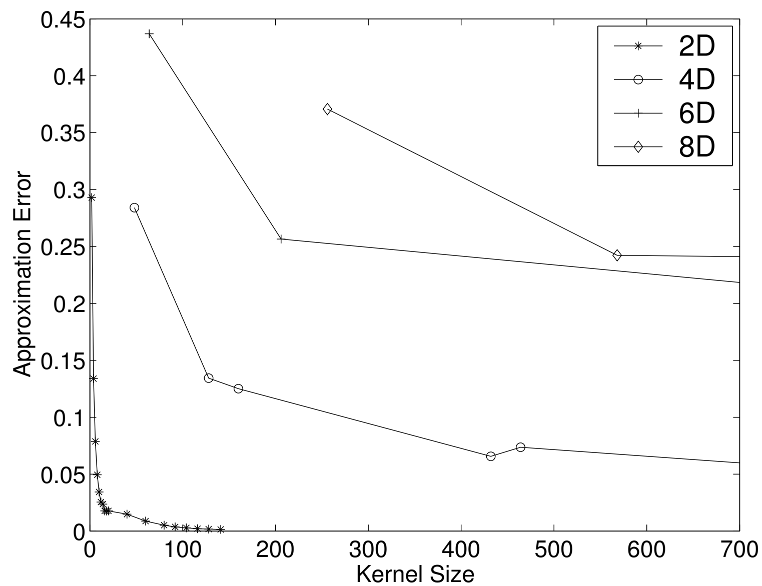
Kernel Computation

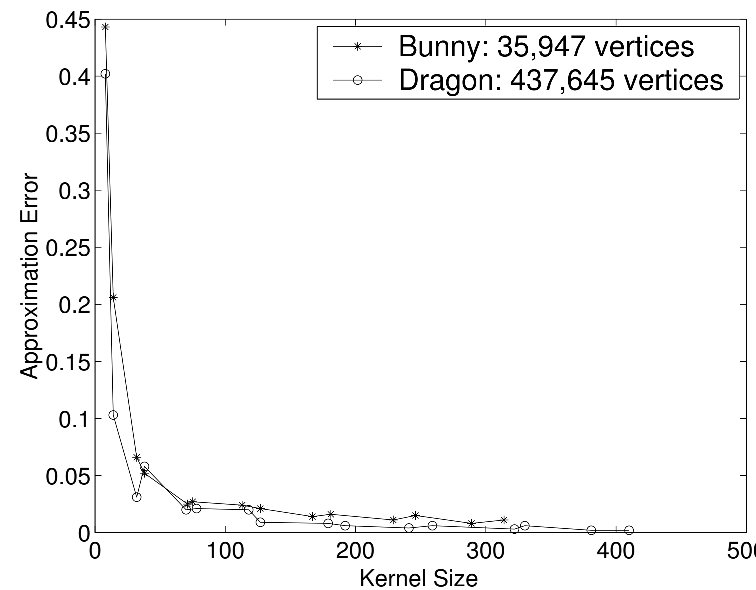
☆ Computing ε-kernels for $\varepsilon < 0.05$ on 3D models

Input Type	*Input Size*	*Running Time*		*Kernel Size*	*Diameter Error*
		Prepr	*Query*		
bunny	35,947	0.17	0.01	67	0.010
dragon	437,645	2.44	0.01	69	0.004
buddha	543,652	2.87	0.01	68	0.010

Experimental Results: Approximation Error

☆ Tradeoff between kernel size and approximation error

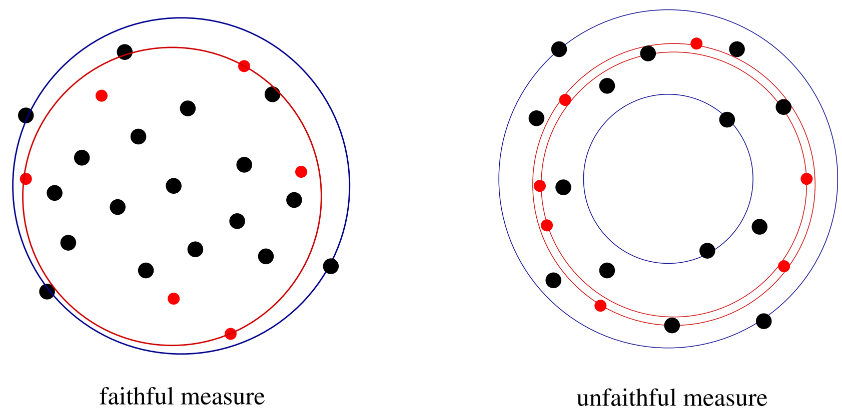




Faithful Extent Measures

$\mu(\cdot)$: Function defined over point sets in $\mathbb{R}^d$ is *faithful* if

- ☆ $\mu(S) \geq 0$ for all $S \subseteq \mathbb{R}^d$
- ☆ $\exists c > 0 \;\; (1 - c\varepsilon)\mu(S) \leq \mu(Q) \leq \mu(S)$ for any ε-kernel Q of S



Faithful measures: Diameter, width, radius of smallest enclosing ball, volume of the smallest enclosing box (simplex)

Nonfaithful measures: width of the thinnest spherical shell containing S

Computing Faithful Measures

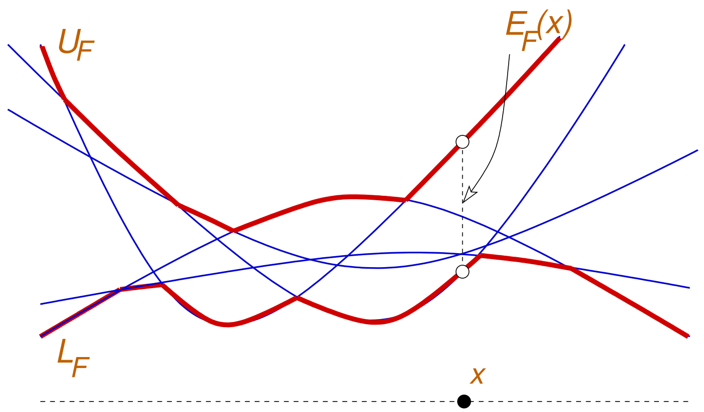
- ☆ S: Set of points, μ: A faithful measure, $\varepsilon > 0$
- ☆ Compute an (ε/c)-kernel Q of S
- ☆ Compute $\mu(Q)$ using a known algorithm
- ☆ Return $\mu(Q)$
  By definition, $\mu(Q) \geq (1-\varepsilon)\mu(S)$
- ☆ $S \subseteq \mathbb{R}^d$, $\varepsilon > 0$
  Can compute a pair $p, q \in S$ s.t. $d(p,q) \geq (1-\varepsilon)\,\mathrm{diam}(S)$
  in time $n + 1/\varepsilon^{d-3/2}$
- ☆ $S \subseteq \mathbb{R}^3$, $\varepsilon > 0$
  Can compute an ε-kernel of the smallest simplex enclosing S
  in time $n + 1/\varepsilon^{9/2}$

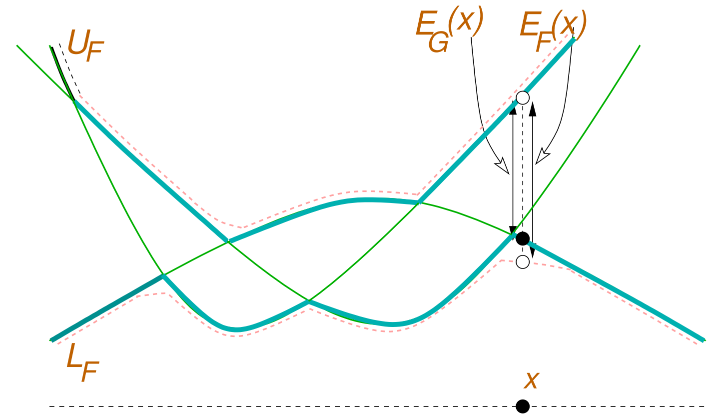
How does one handle unfaithful measures?

Extents of Functions

☆ $F = \{f_1, \ldots, f_n\}$: d-variate functions

- U_F: Upper envelope of F $U_F(x) = \max_i f_i(x)$
- L_F: Lower envelope of F $L_F(x) = \min_i f_i(x)$





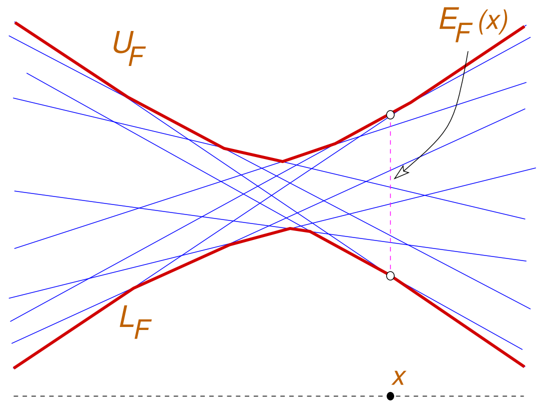
Extent of F:

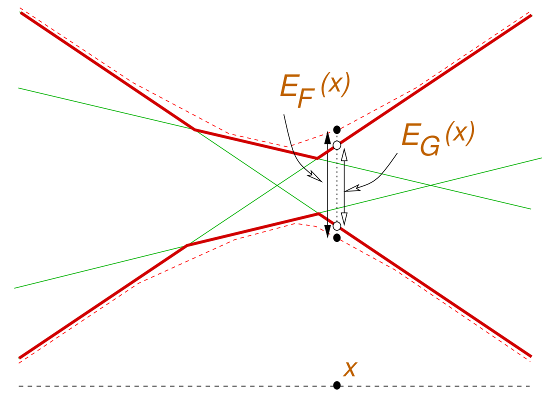
$$E_F(x) = U_F(x) - L_F(x)$$

ε-kernel: $G \subseteq F$ is an ε-kernel of F if

$$(1-\varepsilon)E_F(x) \leq E_G(x) \qquad \forall x \in \mathbb{R}^d$$

ε-Kernel of Linear Functions





☆ Many functions can be mapped to linear functions using *linearization*

☆ Upper and lower envelopes of linear functions are convex polyhedra

☆ Relationship between linear functions and points

Theoream A + Duality:

Theorem B: *H: set of n d-variate linear functions, $\varepsilon > 0$. We can compute an ε-kernel of H of size $1/\varepsilon^{d/2}$ in time $n + 1/\varepsilon^{d-1/2}$.*

ε-Kernels of Polynomials

$F = \{f_1, \ldots, f_n\}$: d-variate polynomials

Linearization [Yao-Yao, A.-Matoušek]

- ☆ Map $\varphi(x) : \mathbb{R}^d \to \mathbb{R}^k$, $\varphi(x) = (\varphi_1(x), \ldots, \varphi_k(x))$
- ☆ Each f_i maps to a k-variate linear function h_i;
  $f_(x) > 0 \Leftrightarrow h_i(\varphi(x)) > 0$
- ☆ k: Dimension of linearization

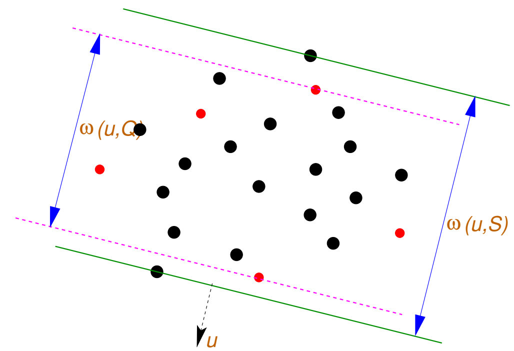
Lemma: *$K \subseteq H$ is an ε-kernel of H $\Leftrightarrow$*
$G = \{f_i \mid h_i \in K\}$ is an ε-kernel of F.

Theorem C: *F: a family of n d-variate polynomials, k: dimension of linearization, $\varepsilon > 0$. We can compute an ε-kernel of F of size $1/\varepsilon^{k/2}$ in time $n + 1/\varepsilon^{k-1/2}$.*

Application I: Kinetic Geometry

S: Set of n moving points in $\mathbb{R}^d$

- $p_i = a_i + b_i t, \qquad a_i, b_i \in \mathbb{R}^d$
- $S(t) = \{p_i(t) \mid 1 \le i \le n\}$



☆ $Q \subseteq S$ an *ε-kernel* if $\forall u \in \mathbb{S}^{d-1}, t \in \mathbb{R}$

$$(1-\varepsilon)\omega(u, S(t)) \le \omega(u, Q(t))$$

☆ $\omega(u, S(t)) = \max_{p \in S} \langle p(t), u \rangle - \min_{p \in S} \langle p(t), u \rangle$

Define $f_i(u,t) = \langle p_i(t), u \rangle$; f_i is a $\deg(2)$ polynomial

Claim: $F = \{f_1, \ldots, f_n\}, \qquad \omega(u, S(t)) = E_F(u,t)$

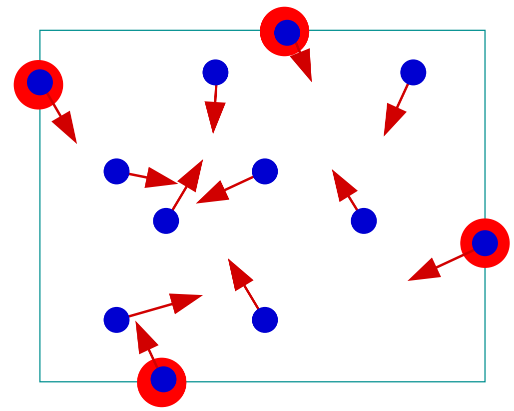
ε-kernel of $F \to \varepsilon$-kernel of S.

Apply Theorem C!

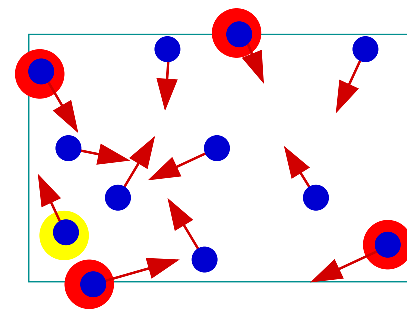
Application I: Kinetic Geometry

Corollary: *S: n linearly moving points in $\mathbb{R}^d$, $\varepsilon > 0$. An ε-kernel of size $1/\varepsilon^{d-1/2}$ can be computed in $n + 1/\varepsilon^{2d-3/2}$ time.*

Maintaining the ε-approximate bounding box of S:



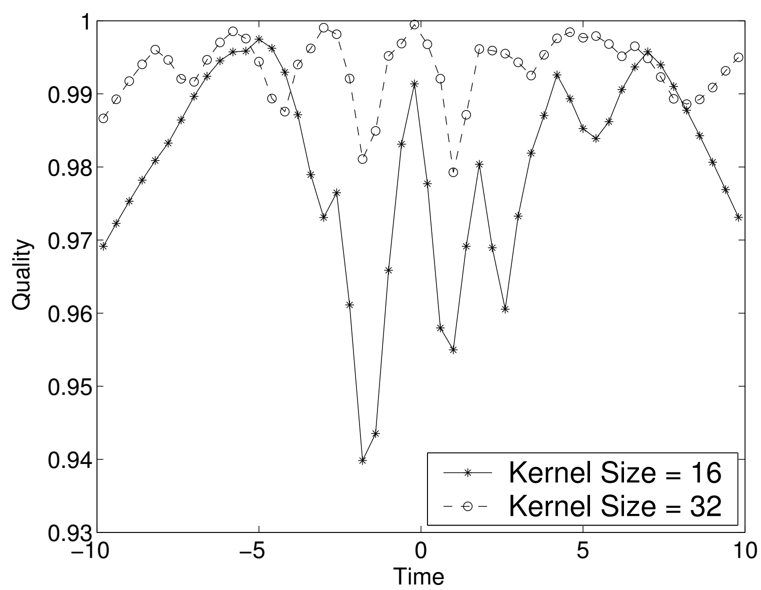
- ☆ Compute an ε-kernel Q of S
- ☆ Use a kinetic data structure to maintain the extent of S in each direction
- ☆ Bounding box is defined by the left-, right-, top-, and bottom-most points
- ☆ Events: When one of these four points change
- ☆ Approach works for maintaining width, smallest enclosing ball/rectangle/simplex, . . .



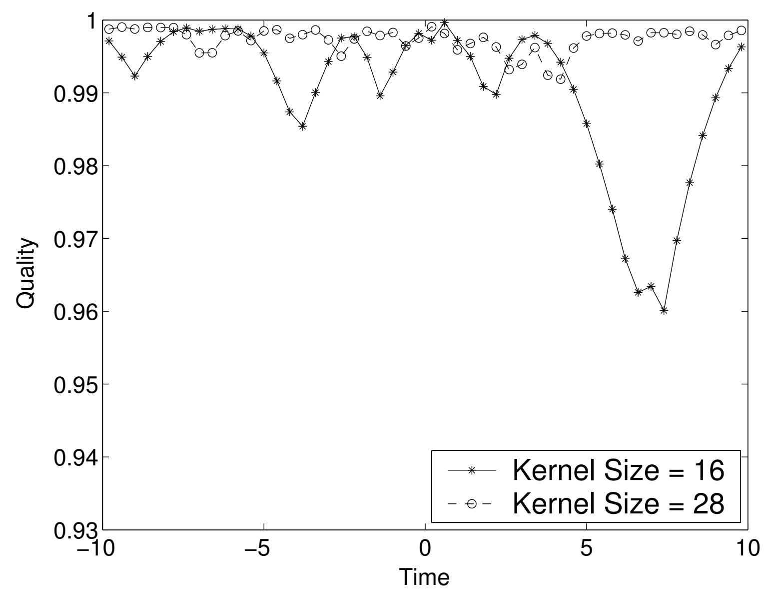
BioGeometry

Bounding Box: Quality of Kernels

- ☆ 100,000 moving points; their trajectories chosen to ensure large number of events
- ☆ Trajectories linear or quadratic
- ☆ Error < 0.04 (98% time) and 0.06 (100% time) for kernel 16.

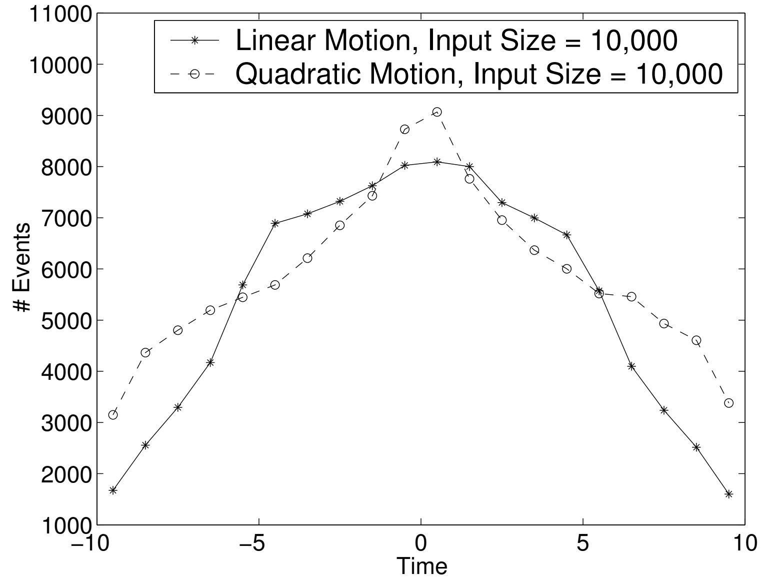


Linear Motion

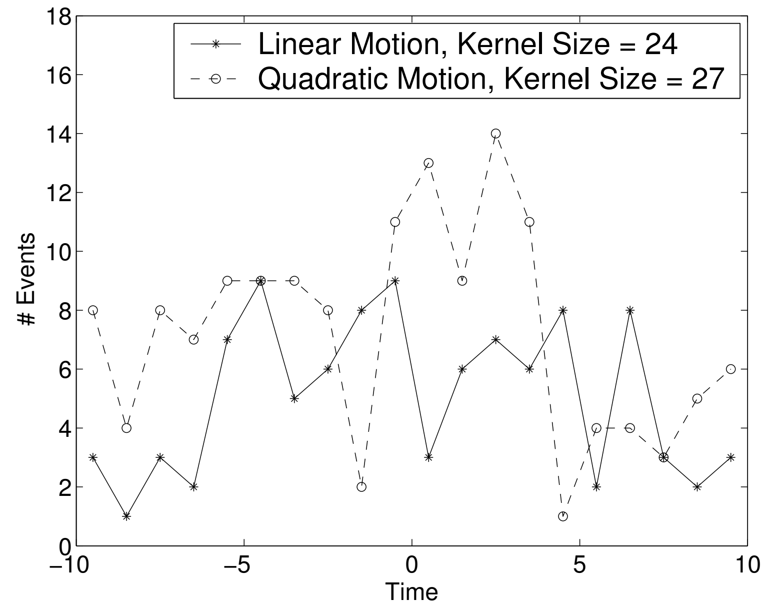


Quadratic Motion

Bounding Box: Number of Events



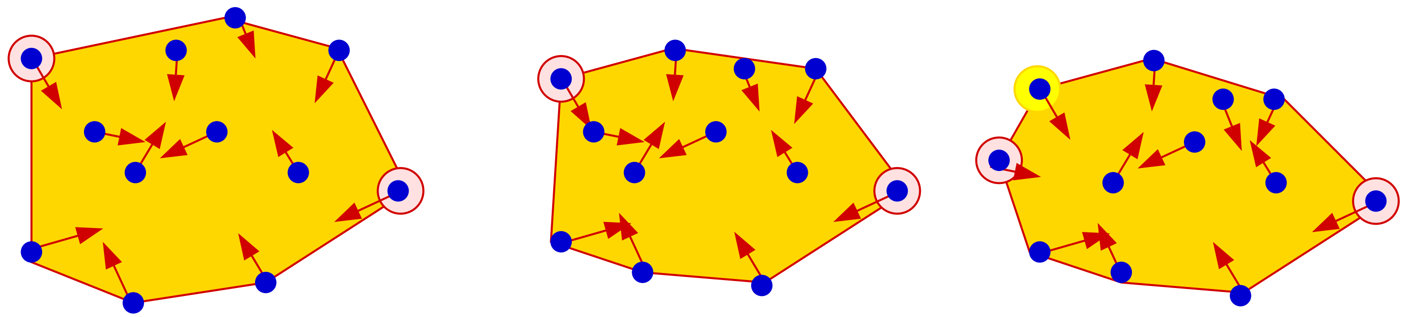
Exact algorithm

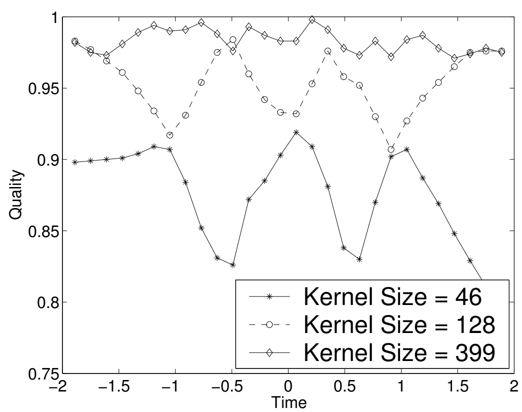


Approximation

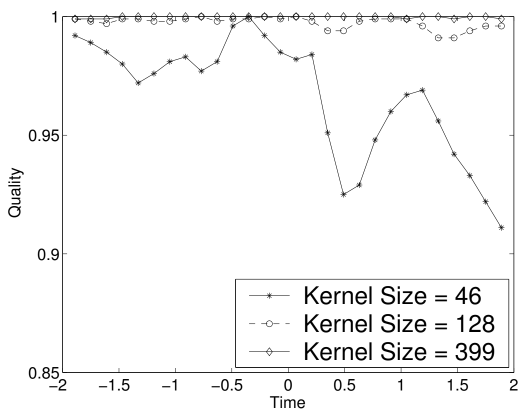
Diameter & Width: Quality of Kernel

- ☆ Maintaining the kernel of $10,000$ moving points
- ☆ Slow implementation of kinetic convex hull algorithm

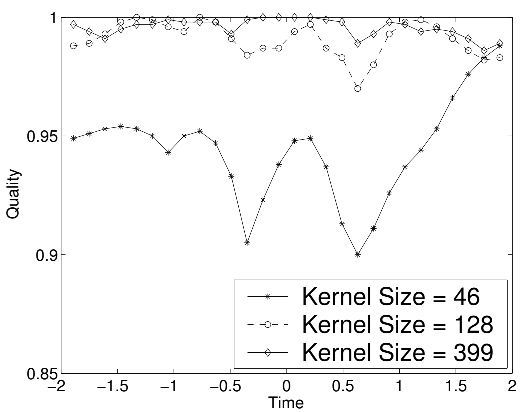




Convex hull

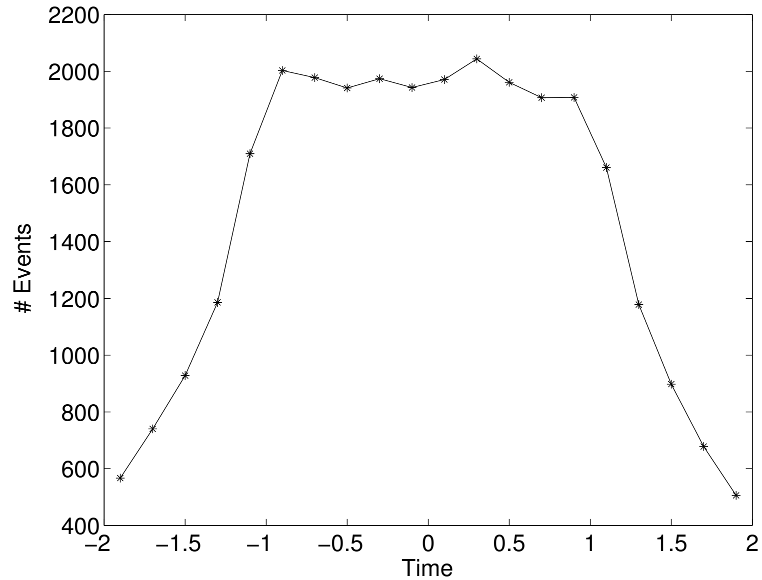


Diameter

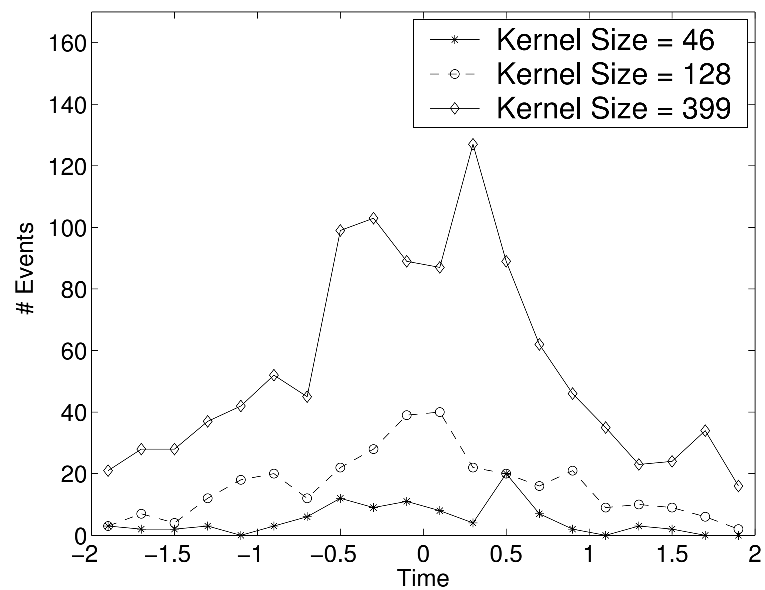


Width

Convex Hull: Number of Events



Exact algorithms



ε-kernel

ε-Kernels of Fractional Polynomials

Functions are not polynomials in many applications

- Distance between point x and circle of radius r centered at p
  $f(x) = \|x - p\| - r$

☆ $F = \{f_1, \ldots, f_n\}$: d-variate functions

☆ $f_i \equiv (h_i)^{1/r}$, h_i: d-variate polynomial, $r \geq 1 \in \mathbb{N}$

☆ $H = \{h_i \mid 1 \leq i \leq n\}$

Theorem D: *$K \subseteq H$ is an $c\varepsilon^r$-kernel of H, $c > 0$ a constant, then $\{f_i \mid h_i \in K\}$ is an ε-kernel of F.*

Corollary: *If H admits a linearization of dimension k, then we can compute an ε-kernel of F of size $1/\varepsilon^{rk/2}$ in time $n + 1/\varepsilon^{rk-1/2}$.*

Application II: Shape Fitting

S: Set of n points in $\mathbb{R}^2$

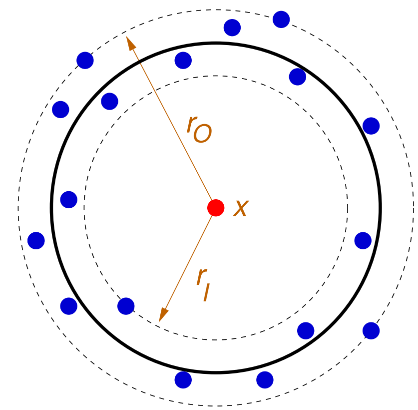
☆ *Find the best-fit circle C through S.*

(minimize the max distance between C and S.)

$\mu(x)$: Min width of annulus containing S centered at x

☆ $d(x,p)$: Distance between x and p

$\mu(x) = \max_{p \in S} d(x,p) - \min_{p \in S} d(x,p)$



$f_i(x) = d(x, p_i)$, $F = \{f_1, \ldots f_n\}$ $\mu(x) = E_F(x)$

Compute $w^* = \min_x E_F(x)$

☆ Compute an ε-kernel G of F; $|G| = 1/\varepsilon$

☆ Compute $\boxed{x^* = \arg\min_x E_G(x)}$

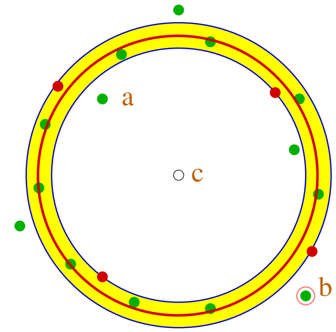
☆ Return $E_F(x^*)$; $E_F(x^*) \leq (1+\varepsilon)w^*$

☆ Time: $n + 1/\varepsilon^{O(1)}$

Shape Fitting: Incremental Algorithm

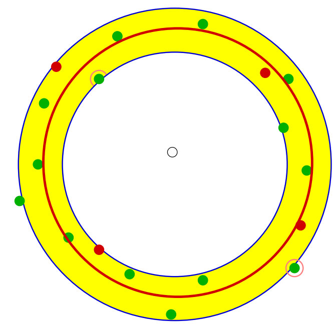
- ☆ S: Set of points in $\mathbb{R}^2$
- ☆ *Find the best-fit circle C through S*

  $\mu(X, C) = \max_{p \in X} d(p, C)$



A simple iterative algorithm

- ☆ $A \subseteq S$: Initially, $|A| = 4$
- ☆ $\mathcal{C}(A)$: Best fit circle for A
- ☆ `while` $\mu(S, \mathcal{C}(A)) > (1 + \varepsilon)\mu(A, \mathcal{C}(A))$
  - $a \in S$: Point farthest from $\mathcal{C}(A)$
  - $A = A \cup \{a\}$



Claim: *The algorithm terminates in $O(1/\varepsilon)$ steps.*

Works for other shape-fitting problems as well.

Minimum Width Annulus

Input Type	*Input Size*	*Running Time*	*Output Width*
$w = 0.05$	10^4	0.01 (2)	0.0501
	10^5	0.02 (2)	0.0500
	10^6	0.18 (2)	0.0500
$w = 0.50$	10^4	0.01 (2)	0.5014
	10^5	0.03 (2)	0.5004
	10^6	0.26 (2)	0.5001
$w = 50.0$	10^4	0.07 (9)	50.051
	10^5	0.12 (9)	50.018
	10^6	0.67 (9)	50.001

- ☆ Points chosen randomly inside an annulus of width w, inner radius 1.0
- ☆ Number of iterations is < 10

Minimum Enclosing Cylinder

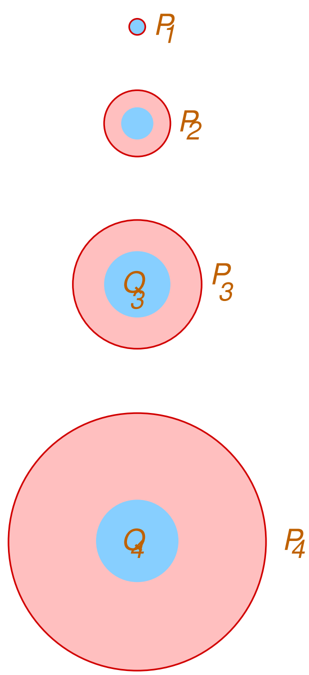
- ☆ Points chosen randomly on a cylindrical surface of radius 1, height h
- ☆ Algorithms APPR: Approximation, CORE: Kernel based INCR: Incremental

Input Type	*Input Size*	*Running Time* APPR	CORE	INCR	*Output Radius* APPR	CORE	INCR
$h = 2.0$	10,000	20.38	0.31	0.11 (6)	1.024	1.012	1.009
	100,000	226.79	1.14	0.28 (6)	1.021	1.020	1.013
	1,000,000	2640.10	12.19	1.63 (5)	1.010	1.011	1.013
$h = 20.0$	10,000	16.82	0.30	0.11 (7)	1.029	1.078	1.072
	100,000	187.21	1.06	0.29 (7)	1.086	1.056	1.021
	1,000,000	2026.54	12.47	2.37 (8)	1.066	1.039	1.068
$h = 200.0$	10,000	16.45	0.28	0.11 (7)	1.052	1.094	1.050
	100,000	186.02	1.04	0.18 (4)	1.030	1.092	1.018
	1,000,000	2067.19	11.98	2.88 (10)	1.072	1.039	1.037
bunny	35,947	68.08	0.46	0.15 (6)	0.0671	0.0672	0.0674
dragon	437,645	809.77	2.94	0.99 (7)	0.0770	0.0770	0.0770
buddha	543,652	1126.27	3.58	1.19 (7)	0.0408	0.0408	0.0408

Application III: Handling Data Stream

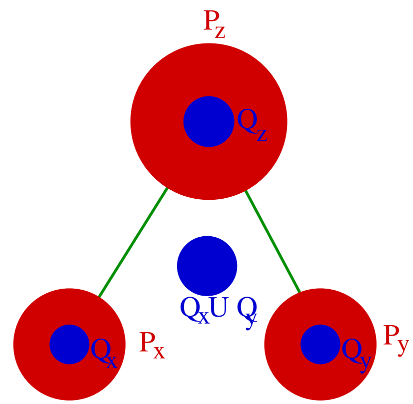
- ☆ S: Stream of points in $\mathbb{R}^2$
- ☆ *Maintain the ε-kernel using $\log^{O(1)} n$ space*
- ☆ Partition P into subsets $P_1, \ldots, P_u$
  - $|P_i| = 2^j$ for some $j \leq \log_2 n$, $j = \mathrm{rank}(P_i)$
  - P_i's are not maintained explicitly
- ☆ Maintain an $(\varepsilon/2)$-kernel Q_i of P_i
  - $|Q_i| = j/\sqrt{\varepsilon}$
  - $\bigcup_i Q_i$ is an $(\varepsilon/2)$-kernel of P.
- ☆ Maintain an $\varepsilon/3$-kernel Q of $\bigcup_i Q_i$

$$|Q| = 1/\sqrt{\varepsilon}$$



Inserting a Point

- ☆ Create a new set $P_0 = \{p\}$; $Q_0 = P_0$
- ☆ If there are two sets P_x, P_y of rank j
  - Compute an $\varepsilon/(j+1)^2$-kernel Q_z of $Q_x \cup Q_y$
  - Delete Q_x, Q_y and add Q_z;
  - $P_z = P_x \cup P_y$; $\mathrm{rank}(P_z) = j+1$
- ☆ Q_z is an $(\varepsilon/2)$-kernel of P_z



Space: $\log^2(n)/\sqrt{\varepsilon}$, Processing time: $O(1/\sqrt{\varepsilon})$ (amortized)

Improvement (Chan). Space: $1/\varepsilon$, time: $O(1)$ (amortized)

Extend to higher diemnsions.

Corollary: $(1-\varepsilon)$-approximation of width, smalelst enclosing box, ... using $1/\varepsilon^{O(1)}$ space and time in the streaming model.

Extensions

☆ Computing ε-kernels in high dimensions
[Bădoiu, Har-Peled, Indyk], [Bădoiu, Clarkson], [Har-Peled, Varadarajan], [Kumar, Mitchell, Yildirim], [Kumar, Yildirim]

- Smallest enclosing ball $\lceil 1/\varepsilon \rceil$
- Smallest enclosing ellipsoid $O(d/\varepsilon)$
- 1-median $1/\varepsilon^{O(1)}$

☆ Computing ε-kernels in presence of outliers [Har-Peled, Wang]

☆ Computing ε-kernels for k-clusters
[Har-Peled], [A., Procopiuc, Varadarajan]

- k-centers
- k-medians
- k-line-centers

Conclusions

- ☆ ε-kernels in high dimensions
  - Polynomial dependence on $d, 1/\varepsilon$
- ☆ General technique for computing core sets for clustering
- ☆ Core sets for shape fitting if we want to minimize the rms distance
  - Given S, compute a cylinder C so that the rms distance between C and S is minimum
- ☆ Core sets and range spaces with finite VC dimensions

References

Review article.

☆ A., S. Har-Peled, K. Varadarajan, Geometric approximation via coresets, *Current Trends in Combinatorial and Computational Geometry* (E. Welzl, eds.), to appear.

Technical artciles.

☆ A., S. Har-Peled, K. Varadarajan, Approximating extent measures of points, *J. ACM*, 51(2004), 606–635.

☆ A., C. M. Procopiuc, and K. Varadarajan, Approximation algorithms for k-line center, *10th Annual European Sympos. Algorithms*, 2002.

☆ T. M. Chan, Faster core-set constructions and data stream algorithms in fixed dimensions, in *Proc. 20th Annu. ACM Sympos. Comput. Geom.*, 2004, 152–159.

☆ H. Yu, A., R. Poreddy, K. Varadarajan, Practical methods for shape fitting and kinetic data structures using core sets, in *Proc. 20th Annu. Sympos. Comput. Geom.*, 2004, 263–272.