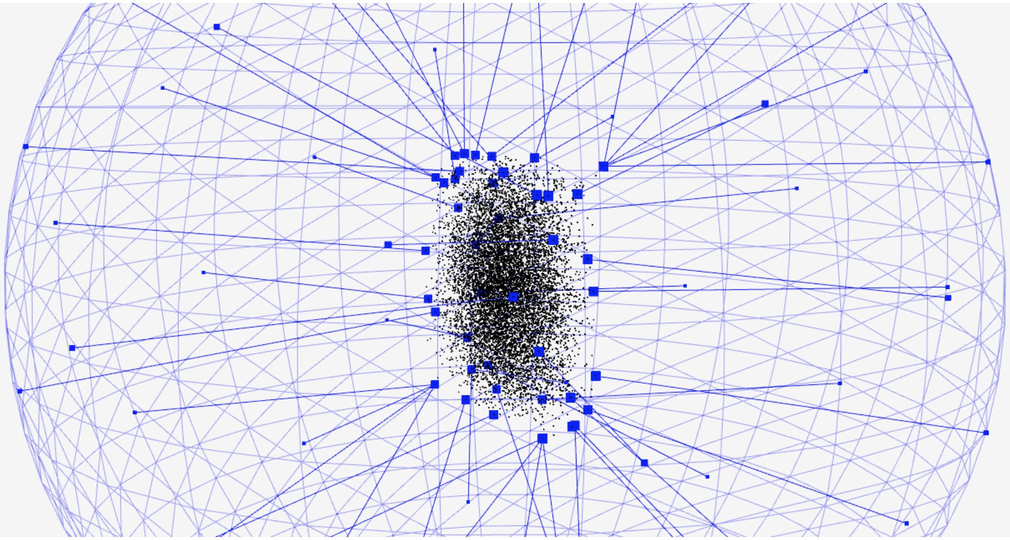
Computational Geometry: The Framework of Coresets



Project presentation - April '17

Charles Hamesse

Subjects

1. About geometric approximations
2. Envelopes and extent measures
3. Directions and directional widths
4. A simple coreset construction
5. An improved construction
6. Implementation

1. About geometric approximations

About geometric approximations: an introduction

Usual technique to develop approximation algorithms:

- extract small amounts of **relevant** information from the input data
- perform relatively-heavy computation on this extracted data.

If the input is a set of points, the question can be reduced to finding a small subset - a **coreset** - of the points, such that one can perform the desired computation on the coreset.

About geometric approximations: a state of the art

Considerable work has been done on measuring various descriptors of the extent of a set P of n points in $\mathbb{R}^d$.

- Exact algorithms for computing extent measures are generally expensive.
- It is said that the best known algorithms for computing the smallest volume bounding box or tetrahedron containing P in $\mathbb{R}^d$ require $O(n^d)$ time.

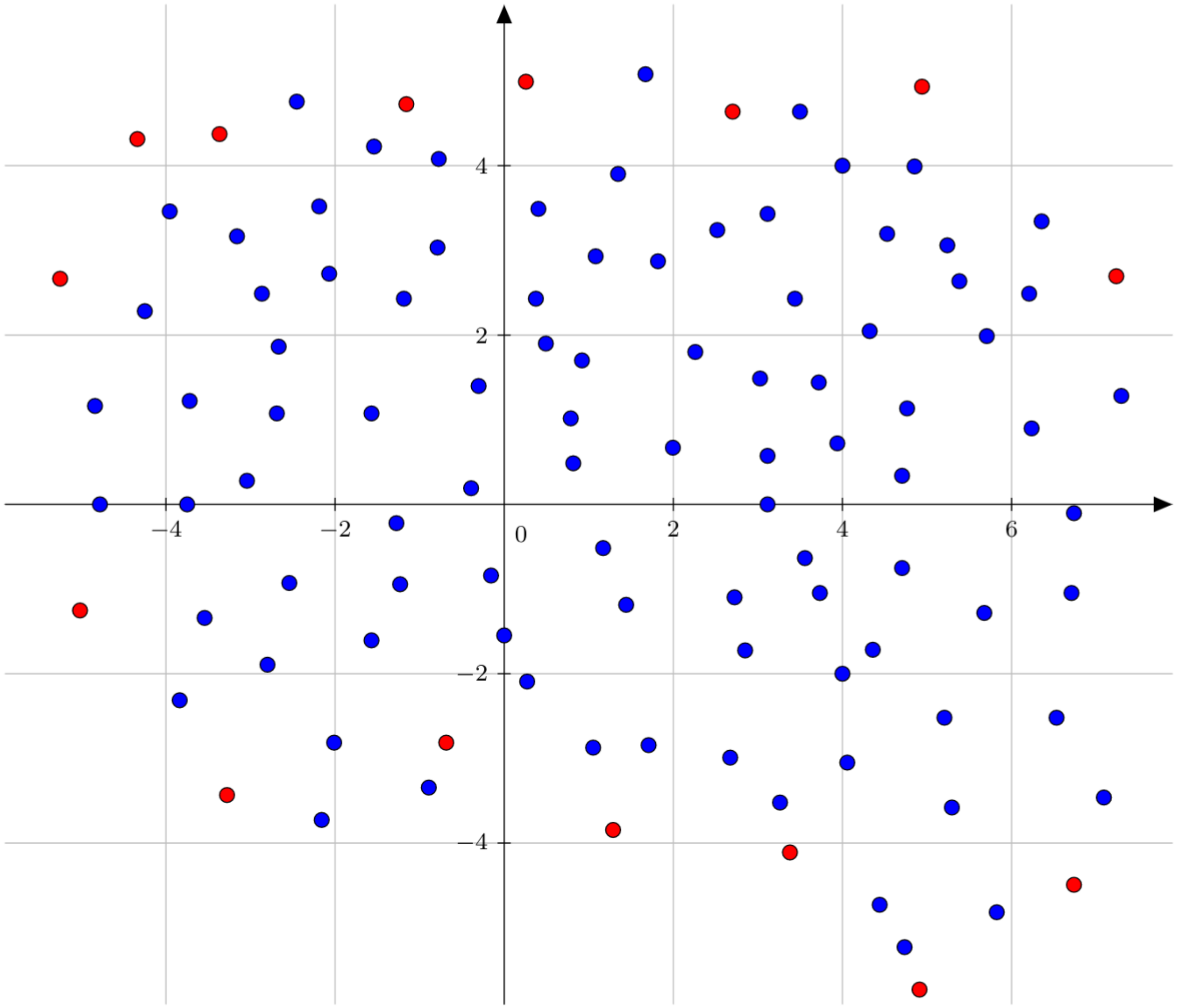
About geometric approximations: the goal

Ideally, one would like to argue that:

- for any extent measure μ
- for any paramater ε such that $0 < \varepsilon < 1$

There exists a subset $Q \in P$ with $|Q| = 1/\varepsilon^{O(1)}$ such that $\mu(Q) \geq (1 - \varepsilon)\mu(P)$

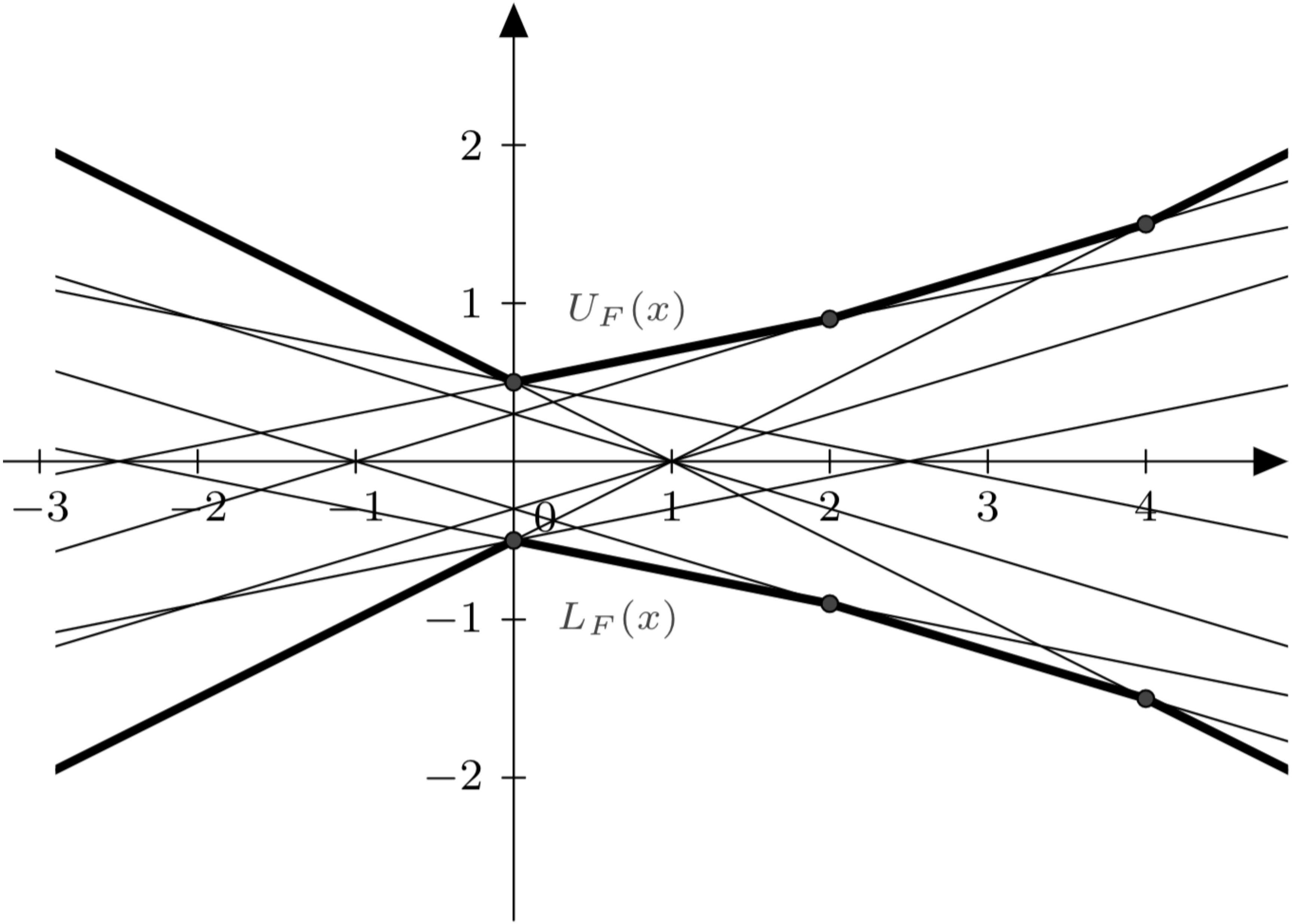
About geometric approximations: an example coreset



Both red and blue points belong to P, red points belong to P's coreset Q.

2. Envelopes and extent measures

Envelopes and extent measures: intuition



Envelopes and extent measures: formal definition of envelopes

Let $F = \{f_1, ..., f_n\}$ be a set of n d-variate functions defined over $x = (x_1, ..., x_d) \in \mathbb{R}^d$:

1. The **lower envelope** of F is the graph of the function:

$$L_F : \mathbb{R}^d \rightarrow \mathbb{R} : x \rightarrow \min_{f \in F} f(x)$$

2. The **upper envelope** of F is the graph of the function:

$$U_F : \mathbb{R}^d \rightarrow \mathbb{R} : x \rightarrow \max_{f \in F} f(x)$$

Envelopes and extent measures: formal definition of extents

The **extent** is defined as:

$$J_F : \mathbb{R}^d \rightarrow \mathbb{R} : x \rightarrow U_F(x) - L_F(x)$$

Now let's introduce a parameter $\varepsilon > 0$ and let Δ be a subset of $\mathbb{R}^d$. The ε-approximation of the extent of F within Δ is the subset G such that, for each $x \in \Delta$:

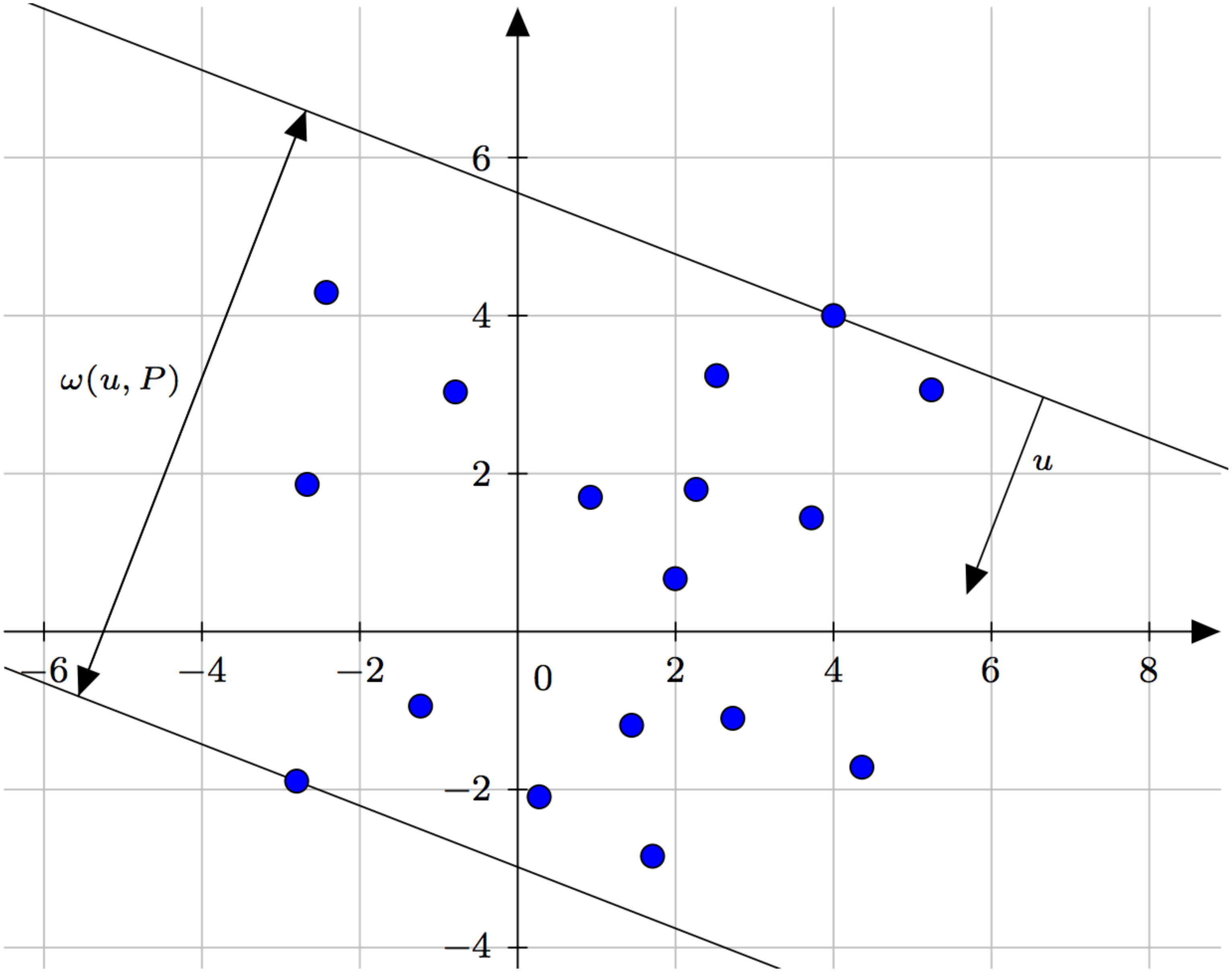
$$(1 - \varepsilon) J_F(x) \leq J_G(x)$$

Observation: $J_G(x) \leq J_F(x)$, as $G \subseteq F$.

If $\Delta = \mathbb{R}^d$, this is simply an ε-approximation of the extent of F.

3. Directions and directional widths

Directions and directional widths: intuition



Directions and directional widths: directions and the n-sphere

Observation: Directions can be expressed as unit vectors on a n-dimensional sphere.

The n-sphere is the generalization of the ordinary sphere to spaces of arbitrary dimension. It is an n-dimensional manifold that can be embedded in Euclidean $(n+1)$-space.

It is defined by:

$$\mathbb{S}^n = \{x \in \mathbb{R}^{n+1} \colon |x| = r\}$$

So our "usual" sphere is really a 2-sphere, or $\mathbb{S}^2$.

Directions and directional widths: formal definition of directional widths

Let $\mathbb{S}^{d-1}$ be the unit sphere centered at the origin in $\mathbb{R}^d$. For any direction $u \in \mathbb{S}^{d-1}$ and a point set $P \subseteq \mathbb{R}^d$, we define the **directional width** as:

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

where $\langle \cdot, \cdot \rangle$ is the usual inner product.

Directions and directional widths: **an approximation of directional widths**

Let's again introduce a parameter $\mu > 0$ and set $\Delta \subseteq \mathbb{R}^d$.

A set $Q \subseteq P$ is a μ-approximation of P within $\Delta \subseteq \mathbb{R}^d$ if, for each $u \in \Delta$:

$$(1 - \mu)\omega(u, P) \leq \omega(u, Q)$$

4. A simple coreset construction

A simple coreset construction: roadmap

In the litterature, it is shown that:

1. Any point set can be turned into an α-fat point set using a linear non-singular transform.

> *"Let P be a set of n points in $\mathbb{R}^d$ and let ε be a parameter. There exists a linear non-singular transform T such that $T(P)$ is α_d-fat, where α_d is a constant depending only on d."*

2. There exists an algorithm for computing coresets of α-fat point sets.

> *"Let P be a α-fat point set contained in C. For any $\varepsilon > 0$, we can compute, in $O(n + 1/(\alpha\varepsilon)^{d-1})$ time, a subset $Q \subseteq P$ of $O(1/(\alpha\varepsilon)^{d-1})$ points that ε-approximates P."*

A simple coreset construction: fat point sets

A point set P is α-fat if there exists a point $p \in \mathbb{R}^d$ and a hypercube $\mathbb{C}$ centered at the origin so that:

$$p + \alpha\mathbb{C} \subset CH(P) \subset p + \mathbb{C}$$

So in this sense, an α-fat point set is just a non-empty set of points whose convex hull which:

- contains the hypercube of edge length α centered at the origin,
- is contained in the unit hypercube.

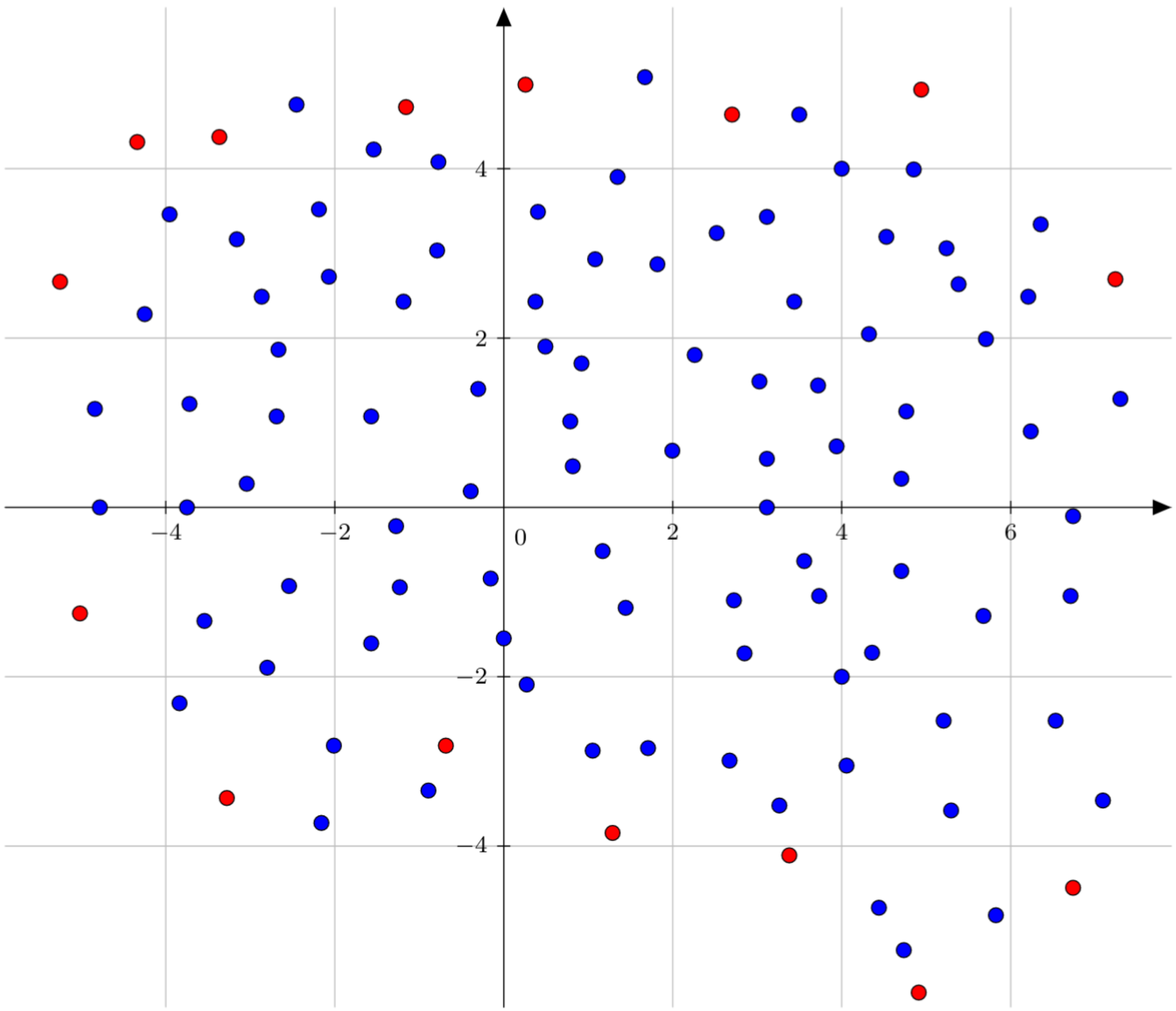
A simple coreset construction: the algorithm

Consider a d-dimensional grid $\mathbb{Z}$ of size $\delta = \frac{\varepsilon\alpha}{6\sqrt{d}}$. That is:

$$\mathbb{Z} = \{(\delta i_1, ..., \delta i_d) | i_1, ..., i_d \in \mathbb{Z}\}$$

- For each column along the x_d-axis in $\mathbb{Z}$, pick one point from both extreme nonempty cells and add them to Q.
- Clearly, the Hausdorff distance between Q and P is smaller than $\frac{\varepsilon\alpha}{6}$ and Q makes an ε-approximation of P.
- It can be shown that $|Q| = O(1/(\alpha\varepsilon)^{d-1})$, so Q can be constructed in time $O(n + 1/(\alpha\varepsilon)^{d-1})$.

A simple coreset construction: an example



5. An improved construction

An improved construction: the algorithm

Let S be the sphere of radius $\sqrt{d+1}$ centered at the origin and set $\delta = \sqrt{\varepsilon\alpha} \leq 1/2$.

- One can construct a set J of $O(1/\delta^{d-1})$ points on S so that:

$$\forall x \in S,\ \exists y \in J : ||x - y|| \leq \delta$$

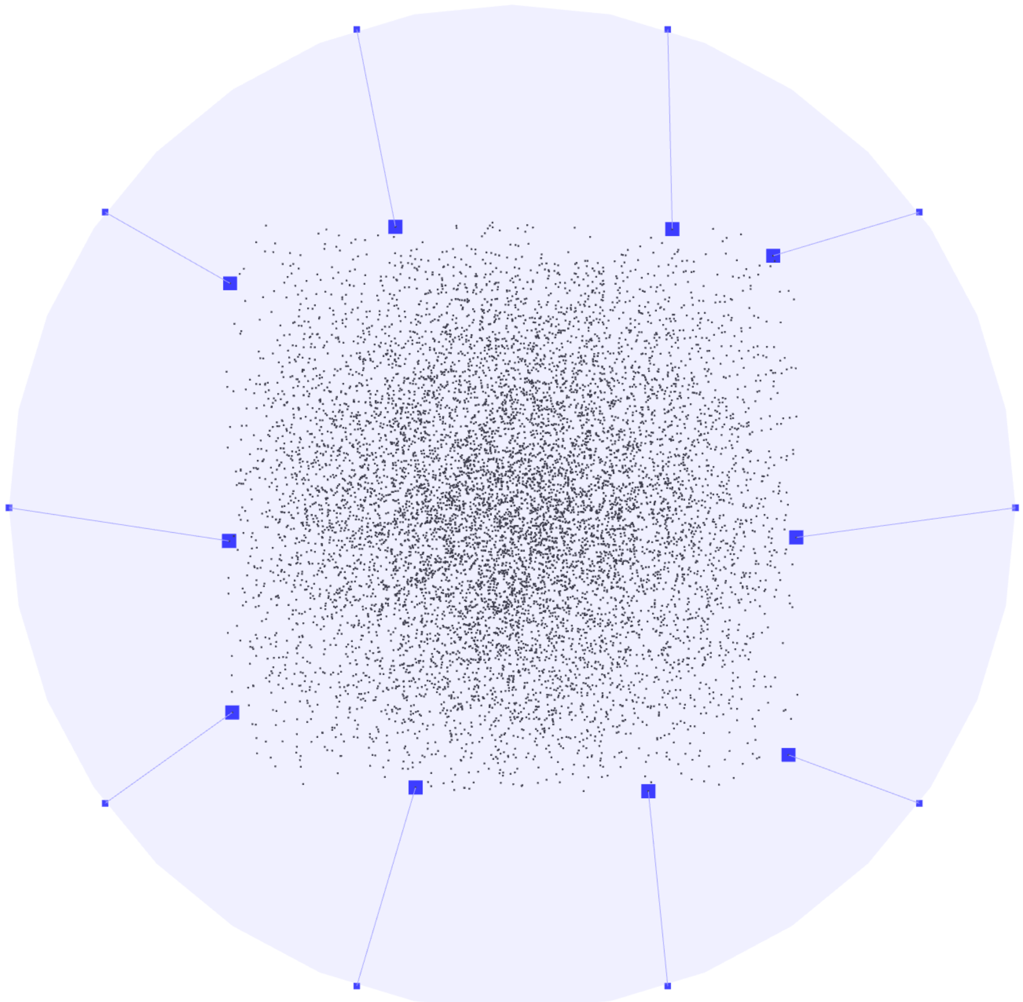
- This can be done by processing P into a data structure that can answer ε-approximate nearest-neighbor queries.
- K-d trees allow exact nearest-neighbor queries.

An improved construction: the algorithm (2)

For a query point q, let $\phi(q)$ be the point of P returned by this data structure.

- For each point $y \in J$, compute $\phi(y)$
- Return the set $Q = \{\phi(y) | y \in J\}$.

An improved construction: an example



6. Implementation

Implementation: computing the error

For a set of directions Δ in $\mathbb{S}^1$ (2D circle), the error is defined as:

$$err(Q, P) = \max_{u \in \Delta} \frac{\omega(u, P) - \omega(u, Q)}{\omega(u, P)}$$

- Intuitively, it is the maximum value amongst all relatives errors on the directional width for all directions u in Δ.
- Accuracy rises with the number of directions considered.
- Authors use 1000 directions in a 4D space.
- I implemented 4 directions in a 2D space: the horizontal, the vertical and both diagonals.

Implementation: demo

Questions?