Lecture 1: Introduction

Instructor: Anastasios Sidiropoulos

January 8, 2014
Geometry & algorithms

Geometry in algorithm design

- **Computational geometry.** Computing properties of geometric objects.

- Point sets, polygons, surfaces, terrains, polyhedra, etc.
- Diameter, volume, traversals, motion planning, etc.

- Geometric interpretation of data.
  - Treating input data set as a geometric object / space.

- Optimization / mathematical programming / geometric relaxations.
Geometry & algorithms

Geometry in algorithm design

- **Computational geometry.** Computing properties of geometric objects.
  - Point sets, polygons, surfaces, terrains, polyhedra, etc.
Geometry & algorithms

Geometry in algorithm design

- **Computational geometry.** Computing properties of geometric objects.
  - Point sets, polygons, surfaces, terrains, polyhedra, etc.
  - Diameter, volume, traversals, motion planning, etc.

- **Geometric interpretation of data.**
  - Treating input data set as a geometric object / space.
Geometry & algorithms

Geometry in algorithm design

- **Computational geometry.** Computing properties of geometric objects.
  - Point sets, polygons, surfaces, terrains, polyhedra, etc.
  - Diameter, volume, traversals, motion planning, etc.

- **Geometric interpretation of data.**
  - Treating input data set as a geometric object / space.
  - Optimization / mathematical programming / geometric relaxations.
Computational geometry

Examples of problems

- Given a set of points \( P \) in some ambient space \( S \)
Computational geometry

Examples of problems

- Given a set of points $P$ in some ambient space $S$
- Compute *efficiently* a property of $P$
  - Diameter
  - Closest Pair
  - Traveling Salesperson Problem (TSP)
  - Minimum Spanning Tree (MST)
Computational geometry

Examples of problems

- Given a set of points $P$ in some ambient space $S$
- Compute *efficiently* a property of $P$
  - Diameter
  - Closest Pair
  - Traveling Salesperson Problem (TSP)
  - Minimum Spanning Tree (MST)
- The *difficulty/complexity* of the problem depends on $S$.
  - Topology
  - Dimension
Geometric interpretation of data

- Often, data consists of a collection of records, each with multiple attributes.
**Geometric interpretation of data**

- Often, data consists of a collection of records, each with multiple attributes.
  - Computer vision (e.g. face recognition)
Geometric interpretation of data

- Often, data consists of a collection of records, each with multiple attributes.
  - Computer vision (e.g. face recognition)
  - Computational biology (e.g. DNA sequences)
Geometric interpretation of data

- Often, data consists of a collection of records, each with multiple attributes.
  - Computer vision (e.g. face recognition)
  - Computational biology (e.g. DNA sequences)
  - pandora.com (Music Genome Project: 400 attributes per song)
Geometric interpretation of data

- Often, data consists of a collection of records, each with multiple attributes.
  - Computer vision (e.g. face recognition)
  - Computational biology (e.g. DNA sequences)
  - pandora.com (Music Genome Project: 400 attributes per song)
  - Engineering, Medicine, Psychology, Finance, . . .
What do we want to compute?

Interesting problems on geometric data sets.
What do we want to compute?

Interesting problems on geometric data sets.

- **Similarity search:** Given a “query” record, find the most *similar* one in the data set, e.g.:
  - Find the most similar face.
  - Fingerprint recognition.
  - On-line dating.
  - Personalized medicine.
What do we want to compute?

Interesting problems on geometric data sets.

- **Similarity search**: Given a “query” record, find the most *similar* one in the data set, e.g.:
  - Find the most similar face.
  - Fingerprint recognition.
  - On-line dating.
  - Personalized medicine.

- **Clustering**: Partition the set of records into similar sets, e.g.:
  - Partition songs into music genres.
What do we want to compute?

Interesting problems on geometric data sets.

- **Similarity search**: Given a “query” record, find the most similar one in the data set, e.g.:
  - Find the most similar face.
  - Fingerprint recognition.
  - On-line dating.
  - Personalized medicine.

- **Clustering**: Partition the set of records into similar sets, e.g.:
  - Partition songs into music genres.

- **Compressed representations**:
  - Compute succinct approximate representation of the data.
  - Dimensionality reduction.
What do we want to compute?

Interesting problems on geometric data sets.

- **Similarity search:** Given a “query” record, find the most similar one in the data set, e.g.:
  - Find the most similar face.
  - Fingerprint recognition.
  - On-line dating.
  - Personalized medicine.

- **Clustering:** Partition the set of records into similar sets, e.g.:
  - Partition songs into music genres.

- **Compressed representations:**
  - Compute succinct approximate representation of the data.
  - Dimensionality reduction.

- **Sketching:** Summarization
  - Finding a (very small) subset of representative records.
What do we want to compute?

Interesting problems on geometric data sets.

- **Similarity search:** Given a “query” record, find the most similar one in the data set, e.g.:
  - Find the most similar face.
  - Fingerprint recognition.
  - On-line dating.
  - Personalized medicine.

- **Clustering:** Partition the set of records into similar sets, e.g.:
  - Partition songs into music genres.

- **Compressed representations:**
  - Compute succinct approximate representation of the data.
  - Dimensionality reduction.

- **Sketching:** Summarization
  - Finding a (very small) subset of representative records.

...
Dramatis personae

Most data comes in two possible forms:
  ▶ Metric spaces
  ▶ Graphs
A metric space is a pair \((X, \rho)\), where:

- \(X\) is the set of points.
- \(\rho : X \times X \rightarrow \mathbb{R}_{\geq 0}\) satisfies:
  - For all \(x, y \in X\), we have \(\rho(x, y) = 0\) if and only if \(x = y\).
  - For all \(x, y \in X\), we have \(\rho(x, y) = \rho(y, x)\).
  - For all \(x, y, z \in X\), we have \(\rho(x, y) \leq \rho(x, z) + \rho(z, y)\).
Metric spaces

A metric space is a pair \((X, \rho)\), where:

- \(X\) is the set of points.
- \(\rho : X \times X \rightarrow \mathbb{R}_{\geq 0}\) satisfies:
  - For all \(x, y \in X\), we have \(\rho(x, y) = 0\) if and only if \(x = y\).
  - For all \(x, y \in X\), we have \(\rho(x, y) = \rho(y, x)\).
  - For all \(x, y, z \in X\), we have \(\rho(x, y) \leq \rho(x, z) + \rho(z, y)\).

Examples of metric spaces?
Graphs as metric spaces

Let $G = (V, E)$ be a graph.
We will often endow $G$ with non-negative edge lengths

$$\text{length} : E \rightarrow \mathbb{R}_{\geq 0}$$
Graphs as metric spaces

Let $G = (V, E)$ be a graph. We will often endow $G$ with non-negative edge lengths

$$\text{length} : E \to \mathbb{R}_{\geq 0}$$

Then, $G$ gives rise to a *shortest-path metric* $d_G$, where for any $u, v \in V$,

$$d_G(u, v) = \min_{P:\text{path from } u \text{ to } v} \text{length}(P),$$

where

$$\text{length}(v_1, \ldots, v_k) = \sum_{i=1}^{k-1} \text{length}({v_i, v_{i+1}}).$$
Graphs as metric spaces

Let $G = (V, E)$ be a graph.
We will often endow $G$ with non-negative edge lengths

$$\text{length} : E \to \mathbb{R}_{\geq 0}$$

Then, $G$ gives rise to a shortest-path metric $d_G$, where for any $u, v \in V$,

$$d_G(u, v) = \min_{P: \text{path from } u \text{ to } v} \text{length}(P),$$

where

$$\text{length}(v_1, \ldots, v_k) = \sum_{i=1}^{k-1} \text{length}({v_i, v_{i+1}}).$$

Examples of shortest-path metrics?
Geometric interpretation

One possible interpretation (but not the only one!):

- Suppose each record has $d$ numerical attributes.
Geometric interpretation

One possible interpretation (but not the only one!):

- Suppose each record has \( d \) numerical attributes.
- Treat each record as a point in \( \mathbb{R}^d \).
Geometric interpretation

One possible interpretation (but not the only one!):

- Suppose each record has $d$ numerical attributes.
- Treat each record as a point in $\mathbb{R}^d$.
- $\ell_p$-distance corresponds to dissimilarity.
Geometric interpretation

One possible interpretation (but not the only one!):

▶ Suppose each record has $d$ numerical attributes.
▶ Treat each record as a point in $\mathbb{R}^d$.
▶ $\ell_p$-distance corresponds to dissimilarity.

▶ What is the right norm?
What is the right norm?

▶ The input might not always be Euclidean.
What is the right norm?

- The input might not always be Euclidean.
- E.g. edit-distance:
  - Metric space \((X, \rho)\).
  - \(X = \{0, 1\}^d\), for some \(d > 0\).
  - \(\rho(x, y) = \min \# \) of insertions/deletions to obtain \(y\) from \(x\).
What is the right norm?

- The input might not always be Euclidean.
- E.g. edit-distance:
  - Metric space \((X, \rho)\).
  - \(X = \{0, 1\}^d\), for some \(d > 0\).
  - \(\rho(x, y) = \min \# \) of insertions/deletions to obtain \(y\) from \(x\).
- Do we need completely different methods for each space?
Metric embeddings

Metric spaces $M = (X, \rho)$, $M' = (X', \rho')$.

A metric embedding is a mapping $f : X \rightarrow X'$.

The distortion of $f$ is a parameter that quantifies how good $f$ is.
Metric embeddings

Metric spaces $M = (X, \rho)$, $M' = (X', \rho')$.
A *metric embedding* is a mapping $f : X \to X'$.
The *distortion* of $f$ is a parameter that quantifies how *good* $f$ is.
Metric embeddings

Metric spaces $M = (X, \rho)$, $M' = (X', \rho')$.

A *metric embedding* is a mapping $f : X \to X'$.

The *distortion* of $f$ is a parameter that quantifies how *good* $f$ is.
Metric spaces $M = (X, \rho)$, $M' = (X', \rho')$.

A **metric embedding** is a mapping $f : X \to X'$.

$$\text{distortion}(f) = \left( \max_{x, y \in X} \frac{\rho'(f(x), f(y))}{\rho(x, y)} \right) \cdot \left( \max_{x', y' \in X} \frac{\rho(x', y')}{\rho'(f(x'), f(y'))} \right)$$
Metric embeddings & algorithm design

- Can we *simplify* a space $S$, while preserving its geometry?
Metric embeddings & algorithm design

- Can we simplify a space $S$, while preserving its geometry?
- Can we embed $S$ into a simpler space $S'$, with low distortion?
Metric embeddings & algorithm design

- Can we *simplify* a space $S$, while preserving its geometry?
- Can we embed $S$ into a *simpler* space $S'$, with low distortion?
- Is the embedding efficiently computable?
Metric embeddings & algorithm design

- Can we simplify a space $S$, while preserving its geometry?
- Can we embed $S$ into a simpler space $S'$, with low distortion?
- Is the embedding efficiently computable?
- If this is possible, then we can obtain faster algorithms!
Simplification via embeddings

Question: Can we embed a complicated space into some simpler space, with small distortion?
Simplification via embeddings
Simplification via embeddings

**Question:** Can we embed a complicated space into some simpler space, with small distortion?
All spaces are approximately Euclidean

Theorem (Bourgain ’85)

Any n-point metric space admits an embedding into Euclidean space with distortion $O(\log n)$.
All spaces are approximately Euclidean

Theorem (Bourgain ’85)

Any $n$-point metric space admits an embedding into Euclidean space with distortion $O(\log n)$.

- I.e. every point $x$ is mapped to some vector in $f(x) \in \mathbb{R}^d$, for some finite $d$. 

Corollary:

Every $n$-point metric space can be stored using linear space, with error/distortion $O(\log n)$.

This embedding is efficiently computable.

Problems in general metrics can be reduced to Euclidean space.
All spaces are approximately Euclidean

Theorem (Bourgain ’85)

Any n-point metric space admits an embedding into Euclidean space with distortion $O(\log n)$.

- I.e. every point $x$ is mapped to some vector in $f(x) \in \mathbb{R}^d$, for some finite $d$.
- The new distance is $\|f(x) - f(y)\|_2$. 

Corollary: Every $n$-point metric space can be stored using linear space, with error/distortion $O(\log n)$.

This embedding is efficiently computable. Problems in general metrics can be reduced to Euclidean space.
All spaces are approximately Euclidean

Theorem (Bourgain ’85)

Any n-point metric space admits an embedding into Euclidean space with distortion $O(\log n)$.

- I.e. every point $x$ is mapped to some vector in $f(x) \in \mathbb{R}^d$, for some finite $d$.
- The new distance is $\|f(x) - f(y)\|_2$.
- **Corollary**: Every $n$-point metric space can be stored using linear space, with error/distortion $O(\log n)$. 
All spaces are approximately Euclidean

Theorem (Bourgain ’85)

Any n-point metric space admits an embedding into Euclidean space with distortion \( O(\log n) \).

- I.e. every point \( x \) is mapped to some vector in \( f(x) \in \mathbb{R}^d \), for some finite \( d \).
- The new distance is \( \|f(x) - f(y)\|_2 \).
- **Corollary:** Every \( n \)-point metric space can be stored using linear space, with error/distortion \( O(\log n) \).
- This embedding is efficiently computable.
All spaces are approximately Euclidean

Theorem (Bourgain ’85)

Any $n$-point metric space admits an embedding into Euclidean space with distortion $O(\log n)$.

- I.e. every point $x$ is mapped to some vector in $f(x) \in \mathbb{R}^d$, for some finite $d$.
- The new distance is $\|f(x) - f(y)\|_2$.
- **Corollary:** Every $n$-point metric space can be stored using linear space, with error/distortion $O(\log n)$.
- This embedding is efficiently computable.
- Problems in general metrics can be reduced to Euclidean space.
Embedding metric space into graphs

Any $n$-point metric space can be embedded into a $n$-vertex graph, with distortion 1.
Embedding metric space into graphs

Any $n$-point metric space can be embedded into a $n$-vertex graph, with distortion 1.

Storing a graph on $n$ vertices requires $O(n^2)$ space.
Embedding metric space into graphs

Any $n$-point metric space can be embedded into a $n$-vertex graph, with distortion 1.

Storing a graph on $n$ vertices requires $O(n^2)$ space. Can we embed into sparse graphs?

Theorem ([Peleg and Schäffer])

For any $c \geq 1$, any $n$-point metric space admits an embedding with distortion $c$ into a graph with $O(n^{1+1/c})$ edges.

Corollary

Any $n$-point metric space admits an embedding with distortion $O(\log n)$ into a graph with $O(n)$ edges.
Constructing a sparse spanner

Let $G = (V, E)$, and suppose $|E| = \binom{n}{2}$.
Constructing a sparse spanner

Let $G = (V, E)$, and suppose $|E| = \binom{n}{2}$.
We will embed $G$ into some graph $G' = (V, E')$ with $|E'| \ll |E|$, with distortion at most some $c > 1$. 

Observation: We may assume that for any $\{u, v\} \in E$, we have $\text{length}(\{u, v\}) = d_G(u, v)$ (if not, setting $\text{length}(\{u, v\}) = d_G(u, v)$ does not change the shortest-path metric).

Sort $E$ in non-decreasing length, i.e. $\text{length}(e_1) \leq \text{length}(e_2) \leq \ldots \leq \text{length}(e_{|E|})$.

Initialize $E' = \emptyset$.

For $i = 1$ to $|E|$ if $G' \cup e_i$ does not contain a cycle with at most $c$ edges: add $e_i$ to $E'$. 

Constructing a sparse spanner

Let $G = (V, E)$, and suppose $|E| = \binom{n}{2}$.

We will embed $G$ into some graph $G' = (V, E')$ with $|E'| \ll |E|$, with distortion at most some $c > 1$.

Observation: We may assume that for any $\{u, v\} \in E$, we have

$$\text{length}(\{u, v\}) = d_G(u, v)$$

(if not, setting $\text{length}(\{u, v\}) = d_G(u, v)$ does not change the shortest-path metric).
Constructing a sparse spanner

Let $G = (V, E)$, and suppose $|E| = \binom{n}{2}$.

We will embed $G$ into some graph $G' = (V, E')$ with $|E'| \ll |E|$, with distortion at most some $c > 1$.

**Observation:** We may assume that for any $\{u, v\} \in E$, we have

$$\text{length}({u, v}) = d_G(u, v)$$

(if not, setting $\text{length}({u, v}) = d_G(u, v)$ does not change the shortest-path metric).

Sort $E$ in non-decreasing length, i.e.

$$\text{length}(e_1) \leq \text{length}(e_2) \leq \ldots \leq \text{length}(e_{|E|}).$$
Constructing a sparse spanner

Let $G = (V, E)$, and suppose $|E| = \binom{n}{2}$.
We will embed $G$ into some graph $G' = (V, E')$ with $|E'| \ll |E|$, with distortion at most some $c > 1$.

**Observation:** We may assume that for any $\{u, v\} \in E$, we have

$$\text{length}(\{u, v\}) = d_G(u, v)$$

(if not, setting $\text{length}(\{u, v\}) = d_G(u, v)$ does not change the shortest-path metric).

Sort $E$ in non-decreasing length, i.e.

$$\text{length}(e_1) \leq \text{length}(e_2) \leq \ldots \leq \text{length}(e_{|E|}).$$

Initialize $E' = \emptyset$. 
Constructing a sparse spanner

Let $G = (V, E)$, and suppose $|E| = \binom{n}{2}$.

We will embed $G$ into some graph $G' = (V, E')$ with $|E'| \ll |E|$, with distortion at most some $c > 1$.

**Observation:** We may assume that for any $\{u, v\} \in E$, we have

$$\text{length}(\{u, v\}) = d_G(u, v)$$

(if not, setting $\text{length}(\{u, v\}) = d_G(u, v)$ does not change the shortest-path metric).

Sort $E$ in non-decreasing length, i.e.

$$\text{length}(e_1) \leq \text{length}(e_2) \leq \ldots \leq \text{length}(e_{|E|}).$$

Initialize $E' = \emptyset$.

For $i = 1$ to $|E|$

  if $G' \cup e_i$ does not contain a cycle with at most $c$ edges:
    add $e_i$ to $E'$
Claim: $G'$ does not contain a cycle with at most $c$ edges.
Analysis

Claim: \( G' \) does not contain a cycle with at most \( c \) edges.

Why?
Claim: $G'$ does not contain a cycle with at most $c$ edges.

Why?

In other words, $G'$ has \textit{girth} at least $c + 1$. 
Lemma

The embedding of $G$ into $G'$ has distortion at most $c$.

Proof.

Let $\{u, v\} \in E$. If $\{u, v\} \in E'$, then $d_G(u, v) = d_{G'}(u, v)$. Otherwise, by construction, there exists a path with at most $c$ edges between $u$ and $v$ in $G'$ (since otherwise we would have added $\{u, v\}$ to $G'$). All these edges are considered before $\{u, v\}$, and thus their length is at most $\text{length}(\{u, v\})$. It follows that $d_{G'}(u, v) \leq c \cdot d_G(u, v)$.

It remains to consider the case $\{u, v\} \notin E$. Let $P = v_1, v_2, \ldots, v_k$ be a shortest-path in $G$ between $u$ and $v$. We have

$$d_{G'}(u, v) \leq \sum_{i=1}^{k-1} d_{G'}(v_i, v_{i+1}) \leq \sum_{i=1}^{k-1} c \cdot \text{length}(v_i, v_{i+1})$$

$$= \sum_{i=1}^{k-1} c \cdot d_G(v_i, v_{i+1}) = c \cdot d_G(u, v)$$
Lemma

Any graph with $n$ vertices, and girth at least $c + 1$, contains at most $n + n^{1 + 1/\lfloor c/2 \rfloor}$ edges.
Lemma

Any graph with $n$ vertices, and girth at least $c + 1$, contains at most $n + n^{1 + 1/\lceil c/2 \rceil}$ edges.

Corollary

$|E'| = O(n^{1 + 1/\lceil c/2 \rceil})$. 
The girth/density bound

**Lemma**

*Any graph $G'$ with $n$ vertices, and girth at least $c + 1$, contains at most* $n + n^{1+1/\lfloor c/2 \rfloor}$ *edges.*

**Proof.**

Assume $c = 2k$.
The girth/density bound

**Lemma**

Any graph $G'$ with $n$ vertices, and girth at least $c + 1$, contains at most $n + n^{1 + 1/\lfloor c/2 \rfloor}$ edges.

**Proof.**

Assume $c = 2k$.

Let $G' = (V, E')$. Suppose $|E'| = m$.

The average degree is $\bar{d} = 2m/n$.

There is a subgraph $H \subseteq G'$, with minimum degree at least $\delta = \bar{d}/2$. Why?
The girth/density bound

Lemma

Any graph $G'$ with $n$ vertices, and girth at least $c + 1$, contains at most $n + n^{1+1/\lfloor c/2 \rfloor}$ edges.

Proof.

Assume $c = 2k$.
Let $G' = (V, E')$. Suppose $|E'| = m$.

The average degree is $\bar{d} = 2m/n$.
There is a subgraph $H \subseteq G'$, with minimum degree at least $\delta = \bar{d}/2$. Why?

- Removing a vertex of degree $< \bar{d}/2$ does not decrease the average degree.
The girth/density bound

**Lemma**

Any graph $G'$ with $n$ vertices, and girth at least $c + 1$, contains at most $n + n^{1+1/\lfloor c/2 \rfloor}$ edges.

**Proof.**

Assume $c = 2k$.

Let $G' = (V, E')$. Suppose $|E'| = m$.

The average degree is $\bar{d} = 2m/n$.

There is a subgraph $H \subseteq G'$, with *minimum* degree at least $\delta = \bar{d}/2$. Why?

- Removing a vertex of degree $< \bar{d}/2$ does not decrease the average degree.

Let $v_0$ be a vertex in $H$. The $k$-neighborhood of $v_0$ is a tree. Why?
The girth/density bound

**Lemma**

Any graph $G'$ with $n$ vertices, and girth at least $c + 1$, contains at most $n + n^{1+1/\lceil c/2 \rceil}$ edges.

**Proof.**

Assume $c = 2k$.

Let $G' = (V, E')$. Suppose $|E'| = m$.

The average degree is $\bar{d} = 2m/n$.

There is a subgraph $H \subseteq G'$, with minimum degree at least $\delta = \bar{d}/2$. Why?

- Removing a vertex of degree $< \bar{d}/2$ does not decrease the average degree.

Let $v_0$ be a vertex in $H$. The $k$-neighborhood of $v_0$ is a tree. Why?

The number of vertices in this tree is at most

$$1 + \delta + \delta(\delta - 1) + \ldots + \delta(\delta - 1)^{k-1} \geq (\delta - 1)^k$$
The girth/density bound

**Lemma**

Any graph $G'$ with $n$ vertices, and girth at least $c + 1$, contains at most $n + n^{1+1/\lceil c/2 \rceil}$ edges.

**Proof.**

Assume $c = 2k$.

Let $G' = (V, E')$. Suppose $|E'| = m$.

The average degree is $\bar{d} = 2m/n$.

There is a subgraph $H \subseteq G'$, with minimum degree at least $\delta = \bar{d}/2$. Why?

- Removing a vertex of degree $< \bar{d}/2$ does not decrease the average degree.

Let $v_0$ be a vertex in $H$. The $k$-neighborhood of $v_0$ is a tree. Why?

The number of vertices in this tree is at most

$$1 + \delta + \delta(\delta - 1) + \ldots + \delta(\delta - 1)^{k-1} \geq (\delta - 1)^k$$

So, $n \geq (\delta - 1)^k$, and $m = \delta n/2 = \delta n \leq n^{1+1/k} + n$. 