# Geometric Data analysis

## Randomized projections and Dimensionality reduction

Ioannis Emiris

Dept Informatics & Telecoms, National Kapodistrian U. Athens
ATHENA Research & Innovation Center, Greece
INRIA Sophia-Antipolis France

Fall 2022

# Outline

1. Dimensionality reduction
   - Proof of JL Lemma
2. Random projections in Euclidean space
   - Projections and k-ANNs
   - Decision problem
3. LSH-able metrics
4. Experimental results

# Approaches

- Trees (and AVDs): $S = O(dn)$, $Q = o(n) \cdot \exp(d)$.
- LSH: $S = O(dn^{1+\rho})$, $Q = O(dn^{\rho})$, $\rho = 1/(1+\epsilon)^2$.
- Dimensionality reduction

  ...and $k$-ANNs beat the curse in optimal space [Anagnostopoulos,E,Psarros:15-17]
  - $S = O(dn)$, $Q = O^*(dn^{\rho})$, $\rho = 1 - \epsilon^2/(\log\log n - \log\epsilon)$.
  - $S = O^*(dn)$, $Q = O^*(dn^{\rho})$, $\rho = 1 + \epsilon^2/\log\epsilon < 1$.

  ...for LSH-able metrics [Avarikioti,E,Psarros,Samaras'17]:
  - $S = O^*(dn)$, $Q = O^*(dn^{\rho})$, $\rho = 1 - \Theta(\epsilon^2)$.

# Dimensionality reduction

## Lemma (Johnson,Lindenstrauss'82)

*Given pointset* $P \subset \mathbb{R}^d$, $|P| = n$, $0 < \epsilon < 1$, *there exists a distribution over linear maps*

$$f : \mathbb{R}^d \to \mathbb{R}^{d'}$$

*with* $d' = O\left(\log n/\epsilon^2\right)$ *s.t., for any* $p, q \in \mathbb{R}^d$, *w/probability* $\geq 2/3$:

$$(1-\epsilon)\|p-q\|_2 \leq \|f(p)-f(q)\|_2 \leq (1+\epsilon)\|p-q\|_2.$$

Proofs (Constructive): Random orthogonal projection [JL'84], Gaussian matrix [Indyk,Motwani'98], i.i.d. entries $\in \{-1,1\}$ [Achlioptas'03], etc.

$f$ oblivious to $P$ i.e. defined over entire space.

Fast JL Transform using structured matrices [Chazelle et al.]

# Outline

1. Dimensionality reduction
   - Proof of JL Lemma
2. Random projections in Euclidean space
   - Projections and k-ANNs
   - Decision problem
3. LSH-able metrics
4. Experimental results

# Gaussian combinations

## Lemma

*Let $g \sim N(0,1)^d$, i.e. with iid normal coordinates, $x \in \mathbb{S}^{d-1}$. Then, their innner product is normally distributed: $\langle x, g\rangle \sim N(0,1)$.*

## Proof.

A linear combination of gaussian variables follows the gaussian distribution. Hence, it suffices to compute the expectation and variance:

$$\mathbb{E}\langle x, g\rangle = \sum_{j=1}^{d} \mathbb{E}[g_j] \cdot x_j = 0,$$

$$\mathbb{E}\langle x, g\rangle^2 = \sum_{k \neq j}^{d} \mathbb{E}[g_j] \cdot \mathbb{E}[g_k] \cdot x_j \cdot x_k + \sum_{j=1}^{d} \mathbb{E}[g_j^2] \cdot x_j^2 = 1,$$

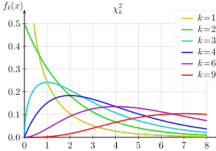
because the $g_j$, $j = 1, \ldots, d$ are independent and $x \in \mathbb{S}^{d-1}$. $\square$

# Squared gaussians

Let each $G_i \sim N(0,1)^d$, $x \in \mathbb{S}^{d-1}$, and $X = G \cdot x$.

**Sum of squares**

For $X_1, \ldots, X_k$ i.i.d. r.v.: $X_i = \langle x, G_i \rangle \sim N(0,1)$, and $Y_k = \sum_{i=1}^{k} X_i^2$, we know $Y_k$ follows the $\chi^2$ distribution with $k$ dof. Clearly $\mathbb{E}[Y_k] = k$.




For r.v. $s$, and $t \in \mathbb{R}$, $\mathbb{E}[e^{ts}]$ is the moment generating function of $s$.

**Fact**

*Let $X \sim N(0,1)$ and $Y_k$ as above. Then, if $t \in (0, 1/2)$,*

$$\mathbb{E}[e^{tX^2}] = \frac{1}{\sqrt{1-2t}} \Rightarrow \mathbb{E}[e^{tY_k}] = \frac{1}{\sqrt{1-2t}^{\,k}}.$$

# Proof of JL Lemma (I)

## Lemma

*Let* $Y = \|X\|_2^2$ : $Y_k = \sum_{i=1}^{k} X_i^2$, $X_i \sim N(0,1)$, *so* $\mathbb{E}[Y_k] = k$. *Then,*

- $\mathrm{P}[Y_k \geq (1+\epsilon)k] < e^{-(\epsilon^2-\epsilon^3)k/4}$,
- $\mathrm{P}[Y_k \leq (1-\epsilon)k] < e^{-(\epsilon^2-\epsilon^3)k/4}$.

## Proof of first bound.

Markov's bound: $\mathrm{P}[x \geq a] \leq \mathbb{E}[x]/a$, $x \geq 0$. Then, for $t \in (0, 1/2)$:

$$\mathrm{P}[Y_k \geq (1+\epsilon)k] = \mathrm{P}[e^{tY_k} \geq e^{(1+\epsilon)tk}] \leq \frac{\mathbb{E}[e^{tY_k}]}{e^{(1+\epsilon)tk}} =$$

$$= \frac{1}{(1-2t)^{k/2} \cdot e^{(1+\epsilon)tk}} \overset{t=\epsilon/2(1+\epsilon)}{=} ((1+\epsilon)e^{-\epsilon})^{k/2} < e^{-(\epsilon^2-\epsilon^3)k/4},$$

using $1 + x \leq exp(x - x^2/2 + x^3/3)$, for $x \in (-1, 1)$. $\square$

# Proof of JL Lemma (II)

## Theorem

*Let $G \in N(0,1)^{k \times d}$ i.e. the elements are i.i.d. r.v.'s that follow $N(0,1)$. Let $A = \frac{1}{\sqrt{k}} G$. Then, for a fixed vector $x \in \mathbb{R}^d$,*

$$\mathrm{P}\left[\|Ax\|^2 \notin [(1-\epsilon)\|x\|^2, (1+\epsilon)\|x\|^2]\right] < 2 \cdot e^{-(\epsilon^2 - \epsilon^3)k/4}.$$

## Proof.

We apply the union bound. Notice that the stated probability equals

$$\mathrm{P}\left[\frac{\|Ax\|^2}{\|x\|^2} \notin [1-\epsilon, 1+\epsilon]\right].$$

In other words, $k \cdot \frac{\|Ax\|^2}{\|x\|^2} = \|G(x/\|x\|)\|^2$ follows the $\chi^2$ distribution with $k$ dof, where $\|x\|$ is fixed. □

# Consequences

## Dimension vs set size

Can always assume $d = o(n)$ or $d = O(\log n)$, otherwise apply JL Lemma to get $d' = O(\log n/\epsilon^2)$.

## Does not remedy the curse for ANN

- BBD-trees still require query time linear in $n$.
- AVDs require $n^{O(-\log \epsilon/\epsilon^2)}$ space, prohibitive if $\epsilon \ll 1$ [HarPeled et al.12]

# Nearest-neighbor Preserving Embedding

## Definition (Indyk,Naor'07)

*Let $X, Y$ be metric spaces, and $P \subseteq X$. A distribution over mappings*

$$f : X \to Y$$

*is a NN-preserving embedding with distortion $D \geq 1$ if, for any $\epsilon > 0$ and query $q \in X$, s.t. $f(p)$ is an $\epsilon$-ANN of $f(q), p \in P$ then, with constant probability,*

$$p \text{ is a } D\epsilon\text{-ANN of } q.$$

## Does it remedy the curse for ANN?

- Yes, for low doubling dim (ddim). Not in general.
- ddim$= \delta$ iff $2^\delta$ balls cover double-radius ball; $\text{ddim}(\ell_p^d) = \Theta(d), p > 1$

# $k$-ANNs

## Definition ($k$-ANNs)

*Given query $q$, find a sequence $S = [p_1, \cdots, p_k] \subset P$ of distinct points s.t. $p_i$ is an $\epsilon$-ANN of the $i$-th exact NN of $q$.*

## Property of tree-based search (*)

The solution to $k$-ANNs using BBD-trees implies, for every point $x \in P$ not visited during the search, $(1 + \epsilon) dist(x, q) > dist(p_k, q)$.

# Outline

# Outline

# Locality-preserving Embedding

## Definition

*Let $X, Y$ be metric spaces, and $P \subseteq X$. A distribution over mappings*

$$f : X \to Y$$

*is a locality-preserving embedding with parameter $k$, distortion $D \geq 1$, and success probability $\delta$ if, for $\epsilon > 0$ and query $q \in X$, when $[f(p_1), \cdots, f(p_k)]$ is a solution to $k$-ANNs of $f(q)$ satisfying the property of tree-based search (*) above then, with probability $\geq \delta$,*

$$\exists\, i \in \{1, \ldots, k\} : p_i \text{ is a } D\epsilon\text{-ANN of } q.$$

[Anagnostopoulos,E,Psarros:SoCG'15-TALG17]

# Low quality embedding

Locality-preserving embeddings lead to an “aggressive” JL-type projection

**Theorem**

*There exists a randomized mapping $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ satisfying the definition of locality-preserving embedding with parameter $k$ for*

$$d' = O\left(\frac{\log(n/k)}{\epsilon^2}\right),$$

*distortion $D = 1 + \epsilon$, $\epsilon \in (0, 1)$, and failure probability $1/3$.*

Eventually $d' \sim \log n/(\epsilon^2 + \log\log n)$.

# Proof ingredients

## Proof of JL by probabilistic argument [Dasgupta,Gupta'03]

For the Euclidean metric $\|\cdot\|$, $\exists$ distribution over linear maps

$$f : \mathbb{R}^d \to \mathbb{R}^{d'},$$

s.t. for $p \in \mathbb{R}^d$, $\|p\| = 1$: If $\beta^2 \neq 1$, then

$$\mathrm{P}[\, \|f(p)\|^2 \leq \beta^2 d'/d \,] \leq exp(\frac{d'}{2}(1 - \beta^2 + 2\ln\beta)).$$

## Two bad cases

- $\#\{$"far-away" $p \in P : f(p)$ within distance $\simeq \beta^2 d'/d\,\} \geq k$,
- nearest neighbor $p^*$: $f(p^*)$ at distance $\geq (1+\epsilon/2)^2 d'/d$.

# Euclidean ANN

Recall: With BBD trees, find $k$-ANNs in $O^*(((1+\frac{d'}{\epsilon})^{d'}+k)\log n)$.

## Lemma

*There exists $k$ s.t., for fixed $\epsilon$, $\lceil 1+6d'/\epsilon \rceil^{d'}+k=O(n^\rho)$, where*

$$\rho = 1-\Theta(\frac{\epsilon^2}{\log\log n}).$$

## Theorem (Main)

*Given $n$ points in $\mathbb{R}^d$, our method employs a BBD-tree to report an $(2\epsilon+\epsilon^2)$-ANN in $O(dn^\rho\log n)$, using space $O(dn)$. Preprocessing takes $O(dn\log n)$ and, for each query, it succeeds with constant probability.*

# Outline

# Projection and $k$ Near neighbors

## Definition ($k$ Approximate Near Neighbors)

*Given $P \subset \mathbb{R}^d$, $\epsilon > 0$, $R > 0$, build a data structure which, for any query point $q \in \mathbb{R}^d$:*

- *if* $|\{p \in P \mid dist(q,p) \leq R\}| \geq k$, *report* $S \subseteq \{p \in P \mid dist(q,p) \leq (1+\epsilon)R\}$: $|S| = k$,
- *if* $|\{p \in P \mid dist(q,p) \leq R\}| < k$, *report* $S \subseteq \{p \in P \mid dist(q,p) \leq (1+\epsilon)R\}$ *s.t.* $|\{p \in P \mid dist(q,p) \leq R\}| \leq |S| \leq k$.

## Theorem

*There exists a linear space and linear preprocessing-time grid-based randomised data structure reporting an Approximate Near Neighbor (or failure) in $\mathbb{R}^d$ with query time in $O(dn^\rho)$, $\rho \simeq 1 + \epsilon^2 / \log \epsilon$.*

# Putting everything together

## Corollary

*The $\epsilon$-ANN optimization problem in $\mathbb{R}^d$ is solved using space $= O^*(dn)$, query time*

$$O^*(dn^\rho),\ \rho = 1 + \epsilon^2 / \log \epsilon < 1,$$

*by a randomized algorithm with constant success probability.*

## Open

Exploit the sequence of $k$-ANNs: It's not a set!

# Outline

# Locality-Sensitive Hashing

Recall LSH.

**Definition (Indyk,Motwani)**

*Let $r \in \mathbb{R}, 0 < \epsilon < 1$ and $1 > p_1 > p_2 > 0$. We call a family $F$ of hash functions $(p_1, p_2, r, (1+\epsilon)r)$-sensitive for a metric space $X$ if, for any $x, y \in X$, and $h_i$ distributed uniformly in $F$:*

- $dist(x, y) \leq r \implies Pr[h_i(x) = h_i(y)] \geq p_1$,
- $dist(x, y) \geq (1+\epsilon)r \implies Pr[h_i(x) = h_i(y)] \leq p_2$.

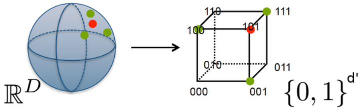
This definition is suitable for the $(\epsilon, r)$-Approximate Near Neighbor decision problem.

# Hamming (0/1) Hypercube

## Projection

- Input: Metric space admitting family of LSH functions $h_i$.
- For each $h_i$ "hashtable": let $f_i$ map buckets to $\{0, 1\}$ uniformly
- Overall projection $f : x \mapsto [\, f_1(h_1(x)), \ldots, f_{d'}(h_{d'}(x))\,] \in \{0, 1\}^{d'}$.
- Preprocess: Project points to vertices of cube, dimension $d' = \lfloor \lg n \rfloor$.

Here $d'$ is like $k$ in LSH.




## Approximate Near Neighbor

- Query: Project query, check points in same and nearby vertices.
- Visit all 0/1 vertices $v$, s.t. $\|v - f(q)\|_1 \leq \frac{1}{2} d'(1 - p_1)$, until: $x$ found, s.t. $\mathrm{dist}(x, q) \leq (1 + \epsilon) r$, or threshold #points checked.

# Important topologies

## Theorem

*For $\ell_1$ and $\ell_2$ metrics, this solves the Approximate Near Neighbor decision problem efficiently, thus yielding a solution for the $\epsilon$-ANN optimization problem with space and preprocessing in $O^*(dn)$, and query time in $O^*(dn^\rho)$, $\rho = 1 - \Theta(\epsilon^2)$.*
*The data structure succeeds with constant probability.*

## Sketch for $\ell_2$

Recall LSH family, for $w \in \mathbb{R}$:

$$x \mapsto h_{vt}(x) = \lfloor \frac{x \cdot v + t}{w} \rfloor,$$

for $v \sim \mathcal{N}(0,1)^d, t \in_R [0, w)$.

# Basic property

## Lemma (General, Technical)

*Given a $(p_1, p_2, r, (1+\epsilon)r)$-sensitive hash family for metric space $X$, there exists a randomized data structure for the $(\epsilon, r)$-Near-Neighbor using space $O(dn)$, preprocessing time $O(dn)$, and query time*

$$O(dn^{1-\Theta((p_1-p_2)^2)} + n^{-\log(p_1(1-p_1))}).$$

*Given a query, preprocessing succeeds with constant probability.*

## Proof sketch

Let $f : X \to \{0,1\}^{d'}$ be the projection defined above. Then for $x, y \in X$:

- $dist(x,y) \le r \implies E[\,\|f_i(h_i(x)) - f_i(h_i(y))\|_1] \le 0.5(1-p_1) \implies E[\|f(x) - f(y)\|_1] \le 0.5 \cdot d' \cdot (1-p_1)$,
- $dist(x,y) \ge c \cdot r \implies E[\,\|f(x) - f(y)\|_1] \ge 0.5 \cdot d' \cdot (1-p_2)$.

# Implementation for $\mathbb{R}^d$

## Parameters

- $d'$: larger implies finer mapping so search can stop earlier; increases storage and preprocessing.
- Threshold #points to be checked in $\mathbb{R}^d$

## Distance computation

- $\|x - q\|^2 = \|x\|^2 + \|q\|^2 - 2q \cdot x$, where the first two can be stored. May offer up to 10% speedup. Slight slowdown on MNIST.

Project idea: $\|x - q\|^2 - \|y - q\|^2$ reduces to $2q \cdot (y - x)$.

https://github.com/gsamaras/Dolphinn

# Outline

# Hypercube

- Implements projection to hypercube, for Approximate Near Neighbors.
- 8-80 times faster than brute force.

`Falconn` implements hyperplane/crosspoly LSH (4748 lines) [AILRS'15].
Hypercube is worse/same in build, same/better in space, query (716 lines)

| | sift | SIFT | MNIST | GIST |
|---|---|---|---|---|
| $d, n$ | 128, $10^4$ | 128, $10^6$ | 784, $6 \cdot 10^4$ | 960, $10^6$ |
| `F (c)` | 2.5e-4 | 1.5e-2 | 3.0e-3 | .34 |
| `F (h)` | 8.6e-5 | 9.0e-3 | 6.2e-4 | .13 |
| `D` | 9.0e-5 | 9.0e-3 | 5.0e-4 | .13 |

Range search, in sec

# DolphinnPy

- `https://github.com/ipsarros/DolphinnPy` [Psarros]
- Python 2.7, NumPy (pip install numpy)
- Hardcoded parameters (main.py):
  $K =$ new (projection) dimension,
  num_of_probes $=$ max #buckets searched,
  $M =$ max #candidate points examined.
- `python main.py`: preprocesses data, runs Dolphinn (hyperplane LSH) and exhaustive search on queries.
- Print $K$, preprocessing and average-query time; multiplicative error (approximation), #exact-answers.

# Tests

- Fix $K$, vary $num\_of\_probes$, $M$ so as to improve accuracy (#exact-answers), decrease multiplicative error.
- Fix $num\_of\_probes$, $M$, vary $K$ for same goal.
- After reading files, the script calls `isotropize` on both sets (data, queries). Compare algorithm after commenting out both lines.

- `siftsmall.tar.gz` from `http://corpus-texmex.irisa.fr/`
- contains datafile and queryfile in `fvecs` format, $d = 128, n = 10^4$.