COMPSCI 514: ALGORITHMS FOR DATA SCIENCE

Cameron Musco

University of Massachusetts Amherst. Fall 2019.

Lecture 9

LOGISTICS

- Problem Set 2 was released on 9/28. **Due Friday 10/11**.
- Problem Set 1 should be graded by the end of this week.
- Midterm on Thursday 10/17. Will cover material through this week, but not material next week (10/8 and 10/10).
- This Thursday, will have a MAP (Midterm Assessment Process).
  - Someone from the Center for Teaching & Learning will collect feedback from you during the first 20 minutes of class.
  - Will be summarized and relayed to me anonymously, so I can make any adjustments and incorporate suggestions to help you learn the material better.

Last Class: The Frequent Elements Problem

- Given a stream of items $x_1, \ldots, x_n$ and a parameter k, identify all elements that appear at least n/k times in the stream.
- Deterministic algorithms: Boyer-Moore majority algorithm and Misra-Gries summaries.
- Randomized algorithm: Count-Min sketch
- Analysis via Markov's inequality and repetition. 'Min trick' similar to median trick.

This Class: Randomized dimensionality reduction.

- The extremely powerful Johnson-Lindenstrauss Lemma and random projection.
- Linear algebra warm up.

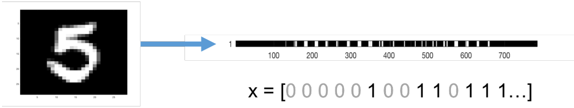
'Big Data' means not just many data points, but many measurements per data point. I.e., very high dimensional data.

- Twitter has 321 active monthly users. Records (tens of) thousands of measurements per user: who they follow, who follows them, when they last visited the site, timestamps for specific interactions, how many tweets they have sent, the text of those tweets, etc...
- A 3 minute Youtube clip with a resolution of 500 x 500 pixels at 15 frames/second with 3 color channels is a recording of ≥ 2 billion pixel values. Even a 500 x 500 pixel color image has $750,000$ pixel values.
- The human genome contains 3 billion+ base pairs. Genetic datasets often contain information on 100s of thousands+ mutations and genetic markers.

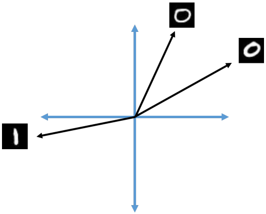
DATASETS AS VECTORS AND MATRICES

In data analysis and machine learning, data points with many attributes are often stored, processed, and interpreted as high dimensional vectors, with real valued entries.

ATAGCCGTAGT → x = [1 2 1 3 4 4 3 2 1 3 4]



x = [0 0 0 0 0 1 0 0 1 1 0 1 1 1...]

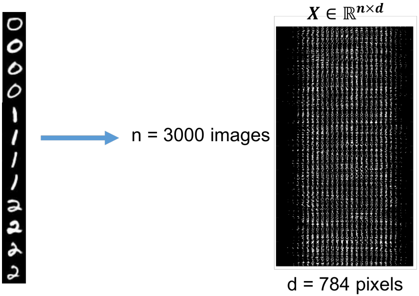


Similarities/distance between vectors (e.g., $\langle x, y \rangle$, $\|x - y\|_2$) have meaning for underlying datapoints.

Data points are interpreted as high dimensional vectors, with real valued entries. Dataset is interpreted as a matrix.

Data Points: $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$

Data Set: $X \in \mathbb{R}^{n \times d}$ with i^{th} row equal to x_i.



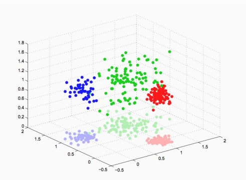
Many data points $n \implies$ tall. Many dimensions $d \implies$ wide.

Dimensionality Reduction: Compress data points so that they lie in many fewer dimensions.

$$x_1, x_2, \ldots, x_n \in \mathbb{R}^d \to \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n \in \mathbb{R}^{d'} \to \text{ for } d' \ll d.$$

x = [0 0 0 0 0 1 0 0 1 1 0 1 1 1...] → $\tilde{x}$ = [-5.5 4 3.2 -1]

'Lossy compression' that still preserves important information about the relationships between $x_1, \ldots, x_n$.

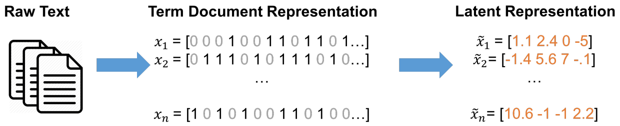


Generally will not consider directly how well $\tilde{x}_i$ approximates x_i.

DIMENSIONALITY REDUCTION

Dimensionality reduction is a ubiquitous technique in data science.

- Principal component analysis
- Latent semantic analysis (LSA)



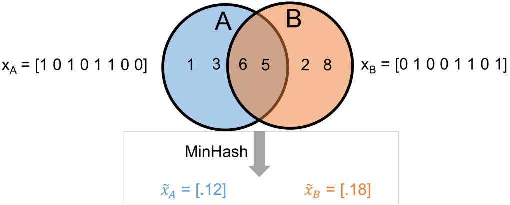
- Linear discriminant analysis
- Autoencoders

Compressing data makes it more efficient to work with. May also remove extraneous information/noise.

Low Distortion Embedding: Given $x_1, \ldots, x_n \in \mathbb{R}^d$, distance function D, and error parameter $\epsilon \geq 0$, find $\tilde{x}_1, \ldots \tilde{x}_n \in \mathbb{R}^{d'}$ (where $d' \ll d$) and distance function $\tilde{D}$ such that for all $i, j \in [n]$:

$$(1-\epsilon)D(x_i, x_j) \leq \tilde{D}(\tilde{x}_i, \tilde{x}_j) \leq (1+\epsilon)D(x_i, x_j)$$

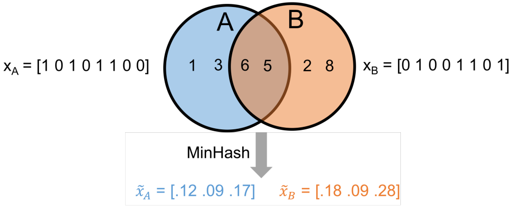
Have already seen one example in class: MinHash



Low Distortion Embedding: Given $x_1, \ldots, x_n \in \mathbb{R}^d$, distance function D, and error parameter $\epsilon \geq 0$, find $\tilde{x}_1, \ldots \tilde{x}_n \in \mathbb{R}^{d'}$ (where $d' \ll d$) and distance function $\tilde{D}$ such that for all $i, j \in [n]$:

$$(1-\epsilon)D(x_i, x_j) \leq \tilde{D}(\tilde{x}_i, \tilde{x}_j) \leq (1+\epsilon)D(x_i, x_j)$$

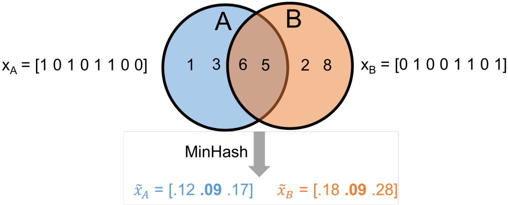
Have already seen one example in class: MinHash



Low Distortion Embedding: Given $x_1, \ldots, x_n \in \mathbb{R}^d$, distance function D, and error parameter $\epsilon \geq 0$, find $\tilde{x}_1, \ldots \tilde{x}_n \in \mathbb{R}^{d'}$ (where $d' \ll d$) and distance function $\tilde{D}$ such that for all $i, j \in [n]$:

$$(1-\epsilon)D(x_i, x_j) \leq \tilde{D}(\tilde{x}_i, \tilde{x}_j) \leq (1+\epsilon)D(x_i, x_j)$$

Have already seen one example in class: MinHash



With large enough signature size r, $\frac{\#\text{ matching entries in } \tilde{x}_A, \tilde{x}_B}{r} \approx J(x_A, x_B)$.

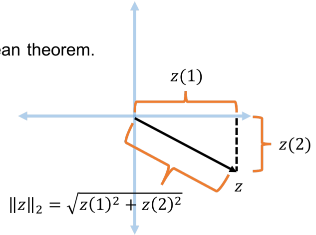
- Reduce dimension from $d = |U|$ to r. Note: here $J(x_A, x_B)$ is a similarity rather than a distance, so not quire a low distortion embedding. But closely related.

Low Distortion Embedding for Euclidean Space: Given $x_1, \ldots, x_n \in \mathbb{R}^d$ and error parameter $\epsilon \geq 0$, find $\tilde{x}_1, \ldots \tilde{x}_n \in \mathbb{R}^{d'}$ (where $d' \ll d$) such that for all $i, j \in [n]$:

$$(1-\epsilon)\|x_i - x_j\|_2 \leq \|\tilde{x}_i - \tilde{x}_j\|_2 \leq (1+\epsilon)\|x_i - x_j\|_2$$

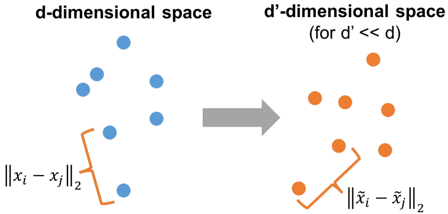
Recall that for $z \in \mathbb{R}^m$, $\|z\|_2 = \sqrt{\sum_{i=1}^m z(i)^2}$.

Pythagorean theorem.



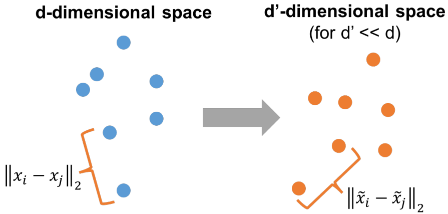
Low Distortion Embedding for Euclidean Space: Given $x_1, \ldots, x_n \in \mathbb{R}^d$ and error parameter $\epsilon \geq 0$, find $\tilde{x}_1, \ldots \tilde{x}_n \in \mathbb{R}^{d'}$ (where $d' \ll d$) such that for all $i, j \in [n]$:

$$(1-\epsilon)\|x_i - x_j\|_2 \leq \|\tilde{x}_i - \tilde{x}_j\|_2 \leq (1+\epsilon)\|x_i - x_j\|_2$$



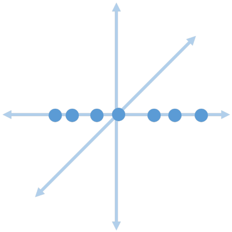
Low Distortion Embedding for Euclidean Space: Given $x_1, \ldots, x_n \in \mathbb{R}^d$ and error parameter $\epsilon \geq 0$, find $\tilde{x}_1, \ldots \tilde{x}_n \in \mathbb{R}^{d'}$ (where $d' \ll d$) such that for all $i, j \in [n]$:

$$(1-\epsilon)\|x_i - x_j\|_2 \leq \|\tilde{x}_i - \tilde{x}_j\|_2 \leq (1+\epsilon)\|x_i - x_j\|_2$$



Can use $\tilde{x}_1, \ldots, \tilde{x}_n$ in place of $x_1, \ldots, x_n$ in many applications: clustering, SVM, near neighbor search, etc.

A very easy case: Assume that $x_1, \ldots, x_n$ all lie on the 1^{st}-axis in $\mathbb{R}^d$.



Set $d' = 1$ and $\tilde{x}_i = x_i(1)$ (i.e., $\tilde{x}_i$ is just a single number.).

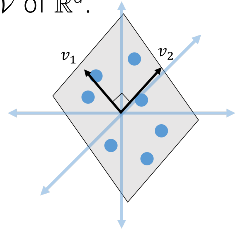
- For all i, j:

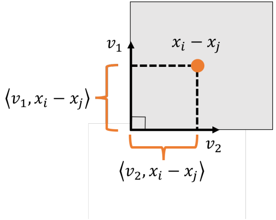
$$\|\tilde{x}_i - \tilde{x}_j\|_2 = \sqrt{[x_i(1) - x_j(1)]^2} = |x_i(1) - x_j(1)| = \|x_i - x_j\|_2.$$

- An embedding with no distortion from any d into $d' = 1$.

An easy case: Assume that $x_1, \ldots, x_n$ lie in any k-dimensional subspace $\mathcal{V}$ of $\mathbb{R}^d$.

An easy case: Assume that $x_1, \ldots, x_n$ lie in any k-dimensional subspace $\mathcal{V}$ of $\mathbb{R}^d$.





- Let $v_1, v_2, \ldots v_k$ be an orthonormal basis for $\mathcal{V}$ and $\mathbf{V} \in \mathbb{R}^{d\times k}$ be the matrix with these vectors as its columns.
- For all i, j, we have $x_i - x_j \in \mathcal{V}$ and (a good exercise to show)

$$\|x_i - x_j\|_2 = \sqrt{\sum_{\ell=1}^{k} \langle v_\ell, x_i - x_j \rangle^2} = \|\mathbf{V}^T(x_i - x_j)\|_2.$$

An easy case: Assume that $x_1, \ldots, x_n$ lie in any k-dimensional subspace $\mathcal{V}$ of $\mathbb{R}^d$.

- Let $v_1, v_2, \ldots v_k$ be an orthonormal basis for $\mathcal{V}$ and $\mathbf{V} \in \mathbb{R}^{d \times k}$ be the matrix with these vectors as its columns.
- For all i, j, we have $x_i - x_j \in \mathcal{V}$ and (a good exercise to show)

$$\|x_i - x_j\|_2 = \sqrt{\sum_{\ell=1}^{k} \langle v_\ell, x_i - x_j \rangle^2} = \|\mathbf{V}^T(x_i - x_j)\|_2.$$

- If we set $\tilde{x}_i \in \mathbb{R}^k$ to $\tilde{x}_i = \mathbf{V}^T x_i$ we have:

$$\|\tilde{x}_i - \tilde{x}_j\|_2 = \|\mathbf{V}^T x_i - \mathbf{V}^T x_j\|_2 = \|\mathbf{V}^T(x_i - x_j)\|_2 = \|x_i - x_j\|_2.$$

- An embedding with no distortion from any d into $d' = k$.
- $\mathbf{V}^T : \mathbb{R}^d \to \mathbb{R}^k$ is a linear map giving our dimension reduction.

What about when we don't make any assumptions on $x_1, \ldots, x_n$. I.e., they can be scattered arbitrarily around d-dimensional space?

- Can we find a no-distortion embedding into $d' \ll d$ dimensions? No! Require $d' = d$.
- Can we find an ϵ-distortion embedding into $d' \ll d$ dimensions for $\epsilon > 0$? Yes! Always, with d' depending on ϵ.

$$\text{For all } i, j: \; (1-\epsilon)\|x_i - x_j\|_2 \leq \|\tilde{x}_i - \tilde{x}_j\|_2 \leq (1+\epsilon)\|x_i - x_j\|_2.$$

Johnson-Lindenstrauss Lemma: For any set of points $x_1, \ldots, x_n \in \mathbb{R}^d$ and $\epsilon > 0$ there exists a linear map $\mathbf{\Pi} : \mathbb{R}^d \to R^{d'}$ such that $d' = O\left(\frac{\log n}{\epsilon^2}\right)$ and letting $\tilde{x}_i = \mathbf{\Pi} x_i$:

For all i, j : $(1-\epsilon)\|x_i - x_j\|_2 \leq \|\tilde{x}_i - \tilde{x}_j\|_2 \leq (1+\epsilon)\|x_i - x_j\|_2$.

Further, if $\mathbf{\Pi}$ has each entry chosen i.i.d. as $\frac{1}{\sqrt{d'}} \cdot \mathcal{N}(0,1)$, it satisfies the guarantee with high probability.

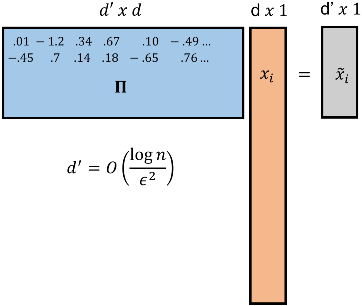
For $d = 1$ trillion, $\epsilon = .05$, and $n = 100000$, $d' \approx 6600$.

Very surprising! Powerful result with a simple (naive) construction: applying a random linear transformation to a set of points preserves the distances between all those points with high probability.

RANDOM PROJECTION

For any $x_1, \ldots x_n$, and $\mathbf{\Pi} \in \mathbb{R}^{d \times d'}$ chosen with each entry chosen i.i.d. as $\frac{1}{\sqrt{d'}} \cdot \mathcal{N}(0,1)$, with high probability, letting $\tilde{x}_i = \mathbf{\Pi} x_i$:

$$\text{For all } i,j: \ (1-\epsilon)\|x_i - x_j\|_2 \leq \|\mathbf{\Pi}(x_i - x_j)\|_2 \leq (1+\epsilon)\|x_i - x_j\|_2.$$



- $\mathbf{\Pi}$ is known as a random projection.
- Data oblivious transformation. Stark contrast to methods like PCA.

Algorithmic Considerations:

- Many alternative constructions: ± 1 entries, sparse (most entries 0), structured, etc. $\implies$ more efficient computation of $\tilde{x}_i = \mathbf{\Pi} x_i$.
- Data oblivious property means that once $\mathbf{\Pi}$ is chosen, $\tilde{x}_1, \ldots, \tilde{x}_n$ can be computed in a stream using little memory
  - For $i = 1, \ldots, n$
    - $\tilde{x}_i := \mathbf{\Pi} x_i$.
  - Memory needed is $O(d + n \cdot d')$ vs. $O(nd)$ to store all the data.
- Compression can also be easily performed in parallel on different servers.
- When new data points are added, can be easily compressed, without updating existing points.