CSCE 689: Special Topics in Modern Algorithms for Data Science

Lecture 7

Samson Zhou

Recall: Concentration Inequalities

- Concentration inequalities bound the probability that a random variable is "far away" from its expectation

- Looking at the k^{th} moment for sufficiently high k gives a number of very strong (and useful!) concentration inequalities with exponential tail bounds

- Chernoff bounds, Bernstein's inequality, Hoeffding's inequality, etc.

Recall: Concentration Inequalities

- Suppose we flip a fair coin $n = 100$ times and let H be the total number of heads

- Markov's inequality: $\Pr[H \geq 60] \leq 0.833$
- Chebyshev's inequality: $\Pr[H \geq 60] \leq 0.25$
- 4^{th} moment: $\Pr[H \geq 60] \leq 0.186$
- Bernstein's inequality: $\Pr[H \geq 60] \leq 0.15$
- Truth: $\Pr[H \geq 60] \approx 0.0284$

Last Time: Chernoff Bounds

- Useful variant of Bernstein's inequality when the random variables are binary

- Chernoff bounds: Let $X_1, \dots, X_n \in \{0, 1\}$ be independent random variables and let $X = X_1 + \cdots + X_n$ have mean μ. Then for any $\delta \geq 0$:

$$\Pr[|X - \mu| \geq \delta\mu] \leq 2 \exp\left(-\frac{\delta^2 \mu}{2 + \delta}\right)$$

Last Time: Median-of-Means Framework

- Suppose we design a randomized algorithm A to estimate a hidden statistic Z of a dataset and we know $0 < Z \leq 1000$.
- Suppose each time we use the algorithm A, it outputs a number X such that $\mathrm{E}[X] = Z$ and $\mathrm{Var}[X] = 100Z^2$
- Suppose we want to estimate Z to accuracy ε, with probability $1 - \delta$

- Accuracy boosting: Repeat A a total of $\frac{10^{12}}{\varepsilon^2}$ times and take the mean
- Success boosting: Find the mean a total of $O\left(\log\frac{1}{\delta}\right)$ times and take the median, to be correct with probability $1 - \delta$

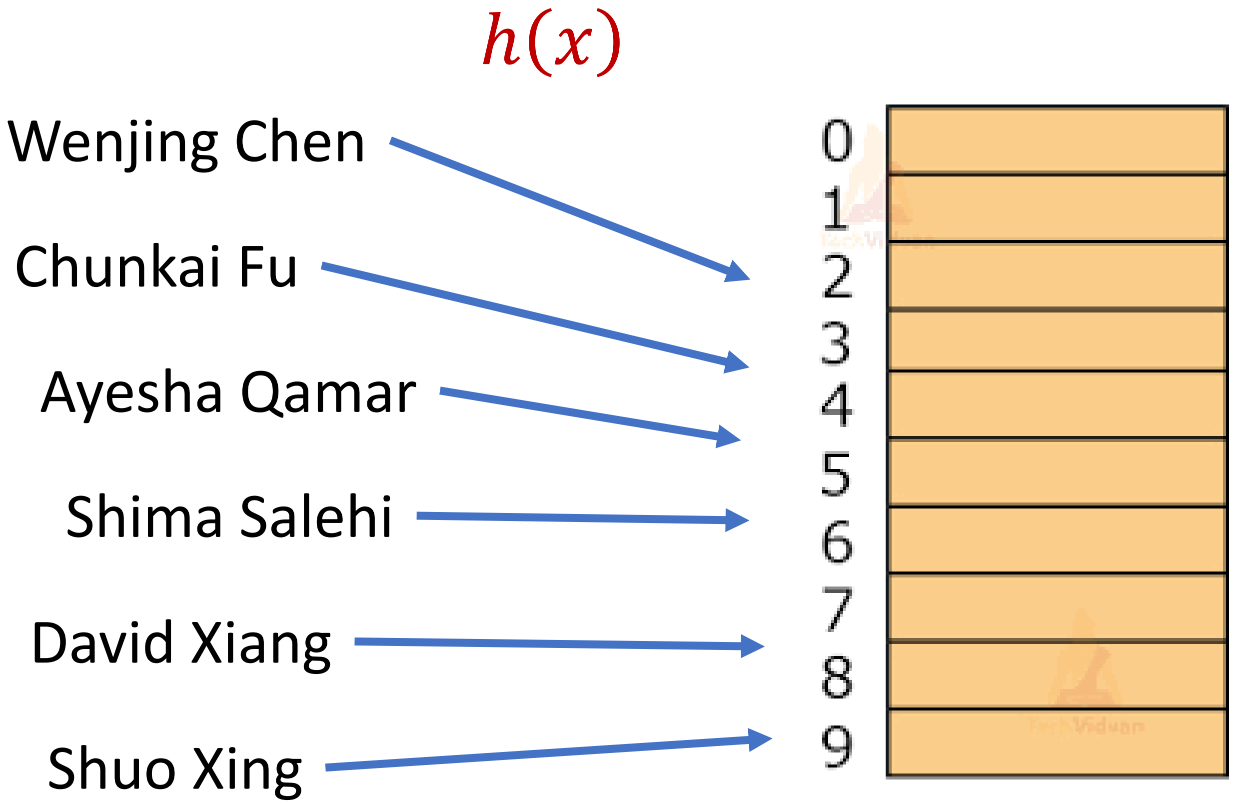
Last Time: Max Load

- Recall we fixed a value $k \in [n]$
- $\Pr[X \geq 3 \log n] \leq \frac{1}{n^2}$ means that with probability at least $1 - \frac{1}{n^2}$, we will get fewer than $3 \log n$ rolls with value k

- Union bound: With probability at least $1 - \frac{1}{n}$, no outcome will be rolled more than $3 \log n$ times

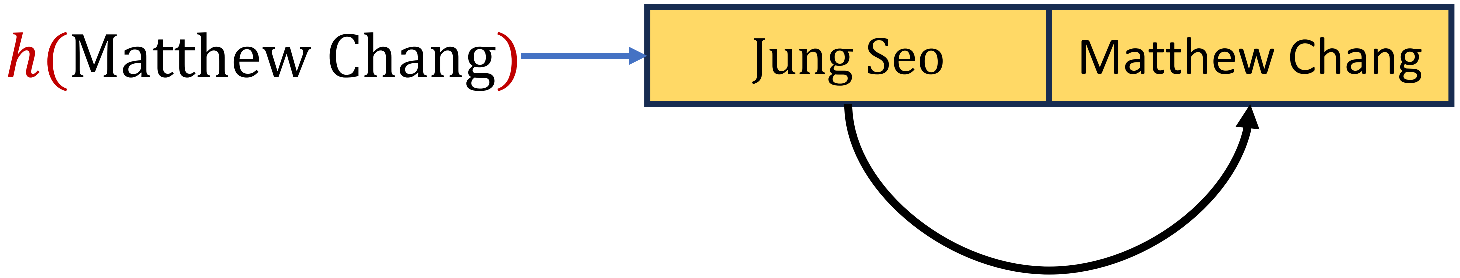
Hashing

- Suppose we have a number of files, how do we consistently store them in memory?
- If we hash n items, we require $\Theta(n^2)$ slots to avoid collisions



Dealing with Collisions

- Suppose we store multiple items in the same location as a linked list



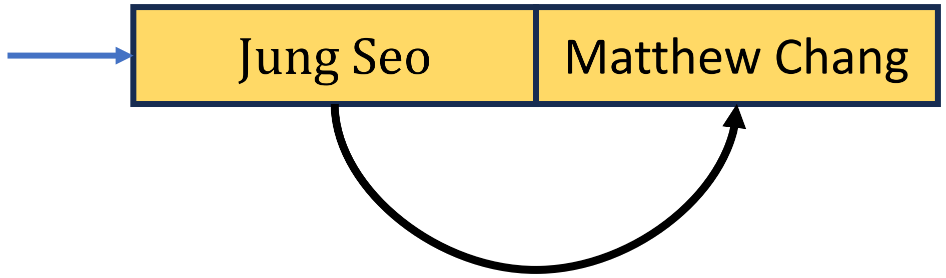
- If the maximum number of collisions in a location is c, then could traverse a linked list of size c for a query

- Query runtime: $O(c)$

Collisions and Max Load

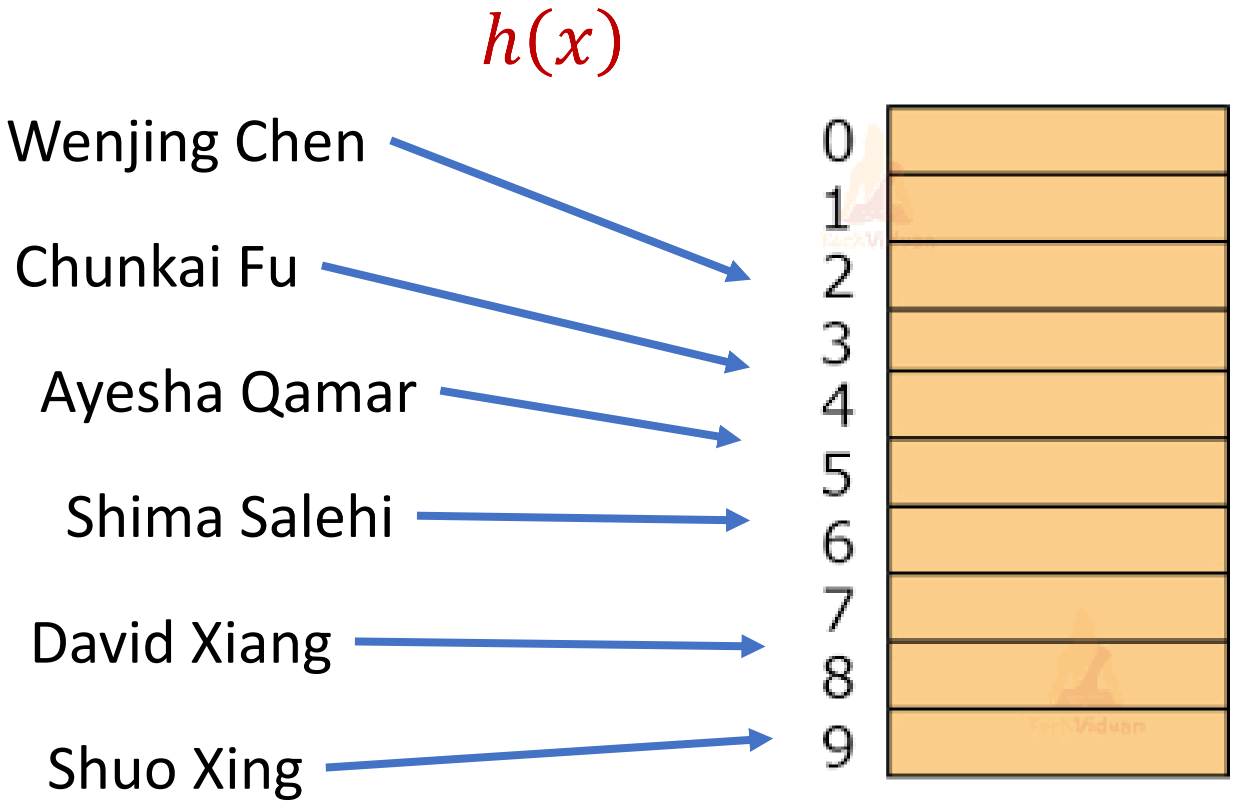
- With probability at least $1 - \frac{1}{n}$, no outcome will be rolled more than $3 \log n$ times

- Worst case query time: $O(\log n)$



Hashing

- For $O(1)$ query time, use $\Theta(n^2)$ slots to avoid collisions
- For $O(\log n)$ query time, use $\Theta(n)$ slots with linked lists



Coupon Collector

- Suppose we have a fair n-sided die. "On average", how many times should we roll the die before we see all possible outcomes among the rolls? Example: $1, 5, 2, 4, 1, 3, 1, 6$ for $n = 6$

- Consider r rolls
- Fix a specific outcome $k \in [n]$
- Let $X_i = 1$ if the i-th roll is k and $X_i = 0$ otherwise

Coupon Collector

- The total number of rolls with value k is $X = X_1 + \cdots + X_r$
- $\mathrm{E}[X] = \frac{r}{n} = 6 \log n$ for $r = 6n \log n$
- Recall Chernoff bounds:

$$\Pr[X \leq (1-\delta)\mu] \leq \exp\left(-\frac{\delta^2 \mu}{2}\right)$$

- $\Pr[X \leq \log n] \leq \frac{1}{n^2}$

Coupon Collector

- Recall we fixed a value $k \in [n]$
- $\Pr[X \leq \log n] \leq \frac{1}{n^2}$ means that with probability at least $1 - \frac{1}{n^2}$, we will at least $\log n$ rolls with value k

- Union bound: With probability at least $1 - \frac{1}{n}$, all outcomes will be rolled at least $\log n$ times

End of Probability Unit

Trivia Question #1 (Birthday Paradox)

- Suppose we have a fair n-sided die. “On average”, how many times should we roll the die before we see a repeated outcome among the rolls? Example: 1, 5, 2, 4, 5

- $\Theta(1)$
- $\Theta(\log n)$
- $\Theta(\sqrt{n})$
- $\Theta(n)$

Trivia Question #3 (Max Load)

- Suppose we have a fair n-sided die that we roll n times. “On average”, what is the largest number of times any outcome is rolled? Example: $1, 5, 2, 4, 1, 3, 1$ for $n = 7$

- $\Theta(1)$
- $\tilde{\Theta}(\log n)$
- $\tilde{\Theta}(\sqrt{n})$
- $\tilde{\Theta}(n)$

Trivia Question #4 (Coupon Collector)

- Suppose we have a fair n-sided die. "On average", how many times should we roll the die before we see all possible outcomes among the rolls? Example: $1, 5, 2, 4, 1, 3, 1, 6$ for $n = 6$

- $\Theta(n)$
- $\Theta(n \log n)$
- $\Theta(n\sqrt{n})$
- $\Theta(n^2)$

Dimensionality Reduction

Many images from:

Cameron Musco's

COMPSCI 514: Algorithms for Data Science

Big Data

- Not only many data points, but also many measurements per data point, i.e., very high dimensional data

Big Data

- Not only many data points, but also many measurements per data point, i.e., very high dimensional data

- Twitter has 450 million active monthly users (as of 2022), 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...

Big Data

- Not only many data points, but also many measurements per data point, i.e., very high dimensional data

- 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

Big Data

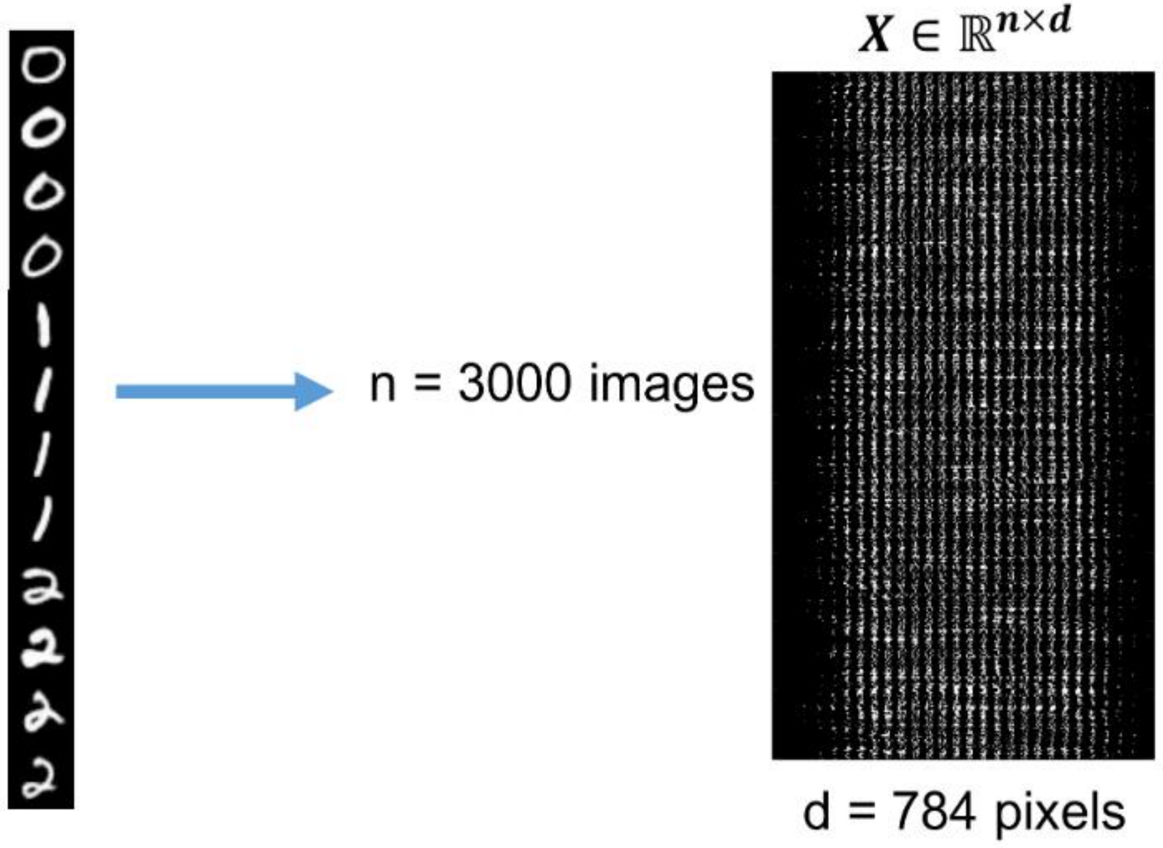
- Not only many data points, but also many measurements per data point, i.e., very high dimensional data

- The human genome contains 3 billion+ base pairs. Genetic datasets often contain information on 100s of thousands+ mutations and genetic markers

Visualizing Big Data

- Data points are interpreted as high dimensional vectors, with real valued entries: $x_1, \dots, x_n \in R^d$

- Dataset is interpreted as a matrix: $X \in R^{n \times d}$ with k-th row x_k



Dimensionality Reduction

- Dimensionality Reduction: Transform the data points so that they have much smaller dimension

$$x_1, \dots, x_n \in R^d \longrightarrow y_1, \dots, y_n \in R^m \quad \text{for} \quad m \ll d$$

$$\longrightarrow x_i = (0,1,0,0,1,0,1,1) \longrightarrow y_i = (-1,2,1)$$

- Transformation should still capture the key aspects of $x_1, \dots, x_n$

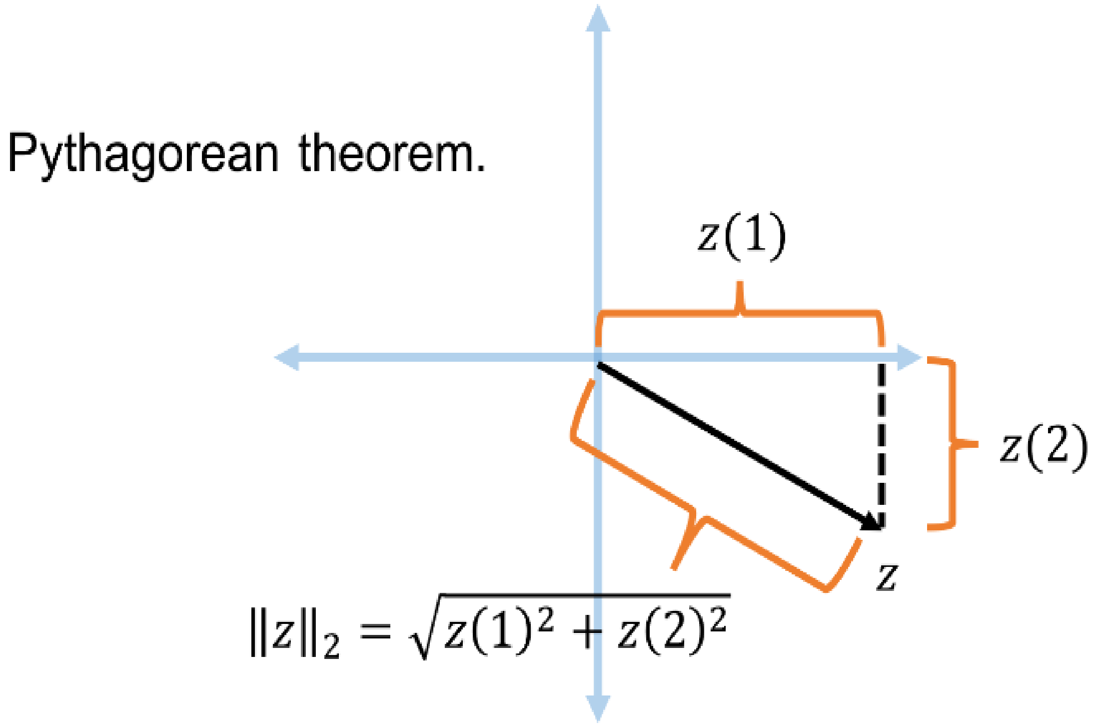
Low Distortion Embedding

- Given $x_1, \dots, x_n \in R^d$, a distance function D, and an accuracy parameter $\varepsilon \in [0,1)$, a low-distortion embedding of $x_1, \dots, x_n$ is a set of points $y_1, \dots, y_n$, and a distance function D' such that for all $i, j \in [n]$
$$(1-\varepsilon)D\left(x_i, x_j\right) \le D'\left(y_i, y_j\right) \le (1+\varepsilon)D\left(x_i, x_j\right)$$

Euclidean Space

- For $z \in R^d$, the ℓ_2 norm of z is denoted by $\|z\|_2$ and defined as:

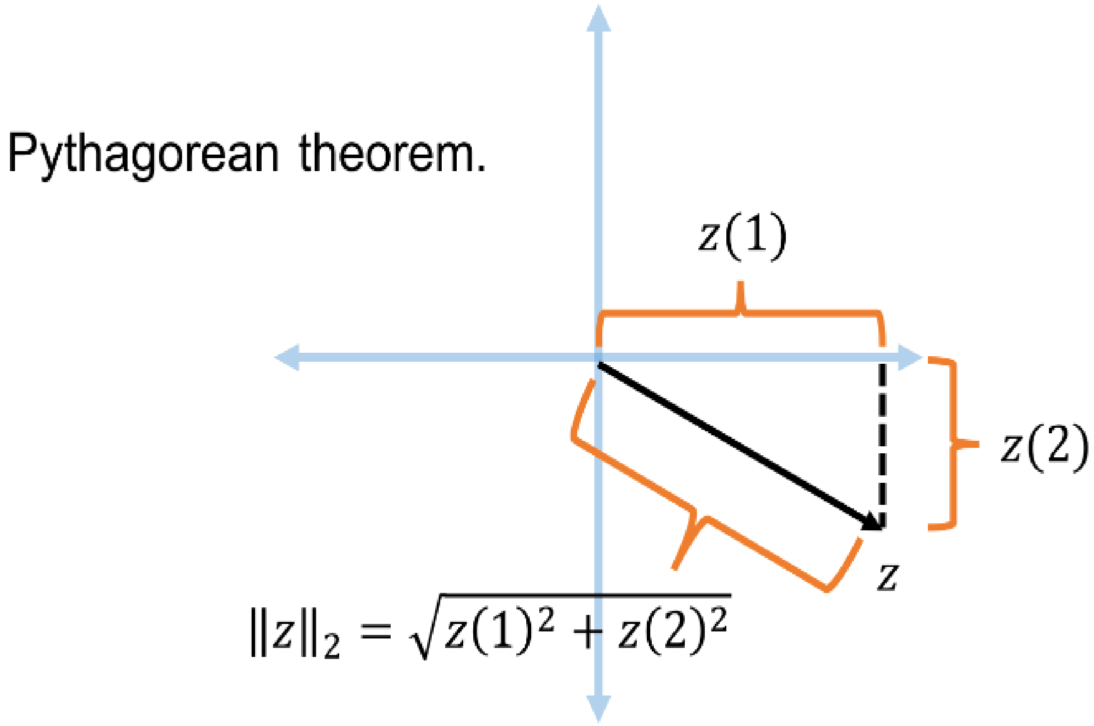
$$\|z\|_2 = \sqrt{z_1^2 + z_2^2 + \cdots + z_d^2}$$



Euclidean Space

- For $z \in R^d$, the ℓ_2 norm of z is denoted by $\|z\|_2$ and defined as:

$$\|z\|_2 = \sqrt{z_1^2 + z_2^2 + \cdots + z_d^2}$$

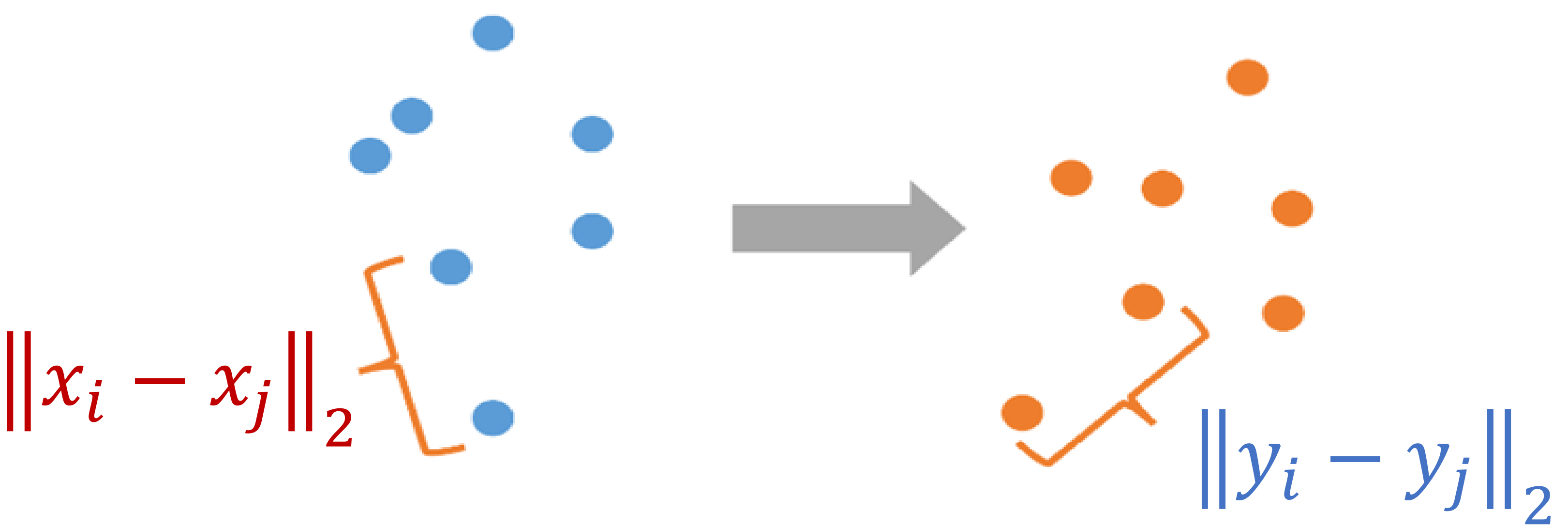


- For $x, y \in R^d$, the distance function D is denoted by $\|\cdot\|_2$ and defined as $\|x - y\|_2$

Low Distortion Embedding for Euclidean Space

- Given $x_1, \dots, x_n \in R^d$ and an accuracy parameter $\varepsilon \in [0,1)$, a low-distortion embedding of $x_1, \dots, x_n$ is a set of points $y_1, \dots, y_n$ such that for all $i, j \in [n]$

$$(1-\varepsilon)\left\|x_i - x_j\right\|_2 \leq \left\|y_i - y_j\right\|_2 \leq (1+\varepsilon)\left\|x_i - x_j\right\|_2$$



Examples: Embeddings for Euclidean Space

- Suppose $x_1, \ldots, x_n \in R^d$ all lie on the 1^{st} - axis
- Take $m = 1$ and y_i to be the first coordinate of x_i
- Then $\left\|y_i - y_j\right\|_2 = \left\|x_i - x_j\right\|_2$ for all $i, j \in [n]$
- Embedding has no distortion