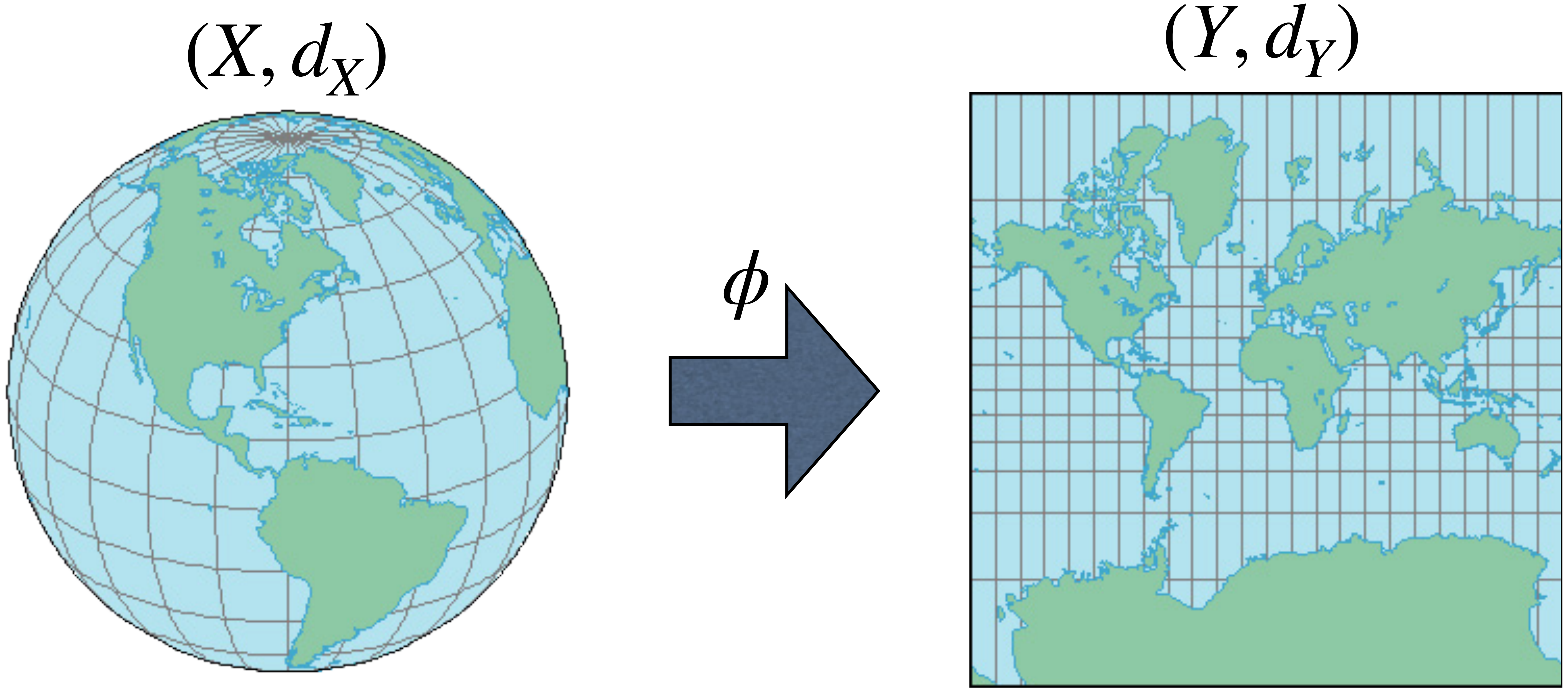
Advanced Algorithms

Dimensionality Reduction

刘明谋 Nanjing University, Suzhou, 2025

Metric Embedding

- Two metric spaces: (X, d_X) and (Y, d_Y)



low-distortion: for small $\alpha \geq 1$

$$\forall x_1, x_2 \in X : \quad \frac{1}{\alpha} d_X(x_1, x_2) \leq d_Y(\phi(x_1), \phi(x_2)) \leq \alpha d_X(x_1, x_2)$$

Dimension Reduction

Input: n points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$

Output: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in \mathbb{R}^k$ s.t. $\forall 1 \leq i, j \leq n$:

$$(1-\epsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\| \leq \|\boldsymbol{y}_i - \boldsymbol{y}_j\| \leq (1+\epsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$$

- Usually we want $k \ll d$.
- How small can k be?
- For what distance $||\cdot||$?
- The embedding should be efficiently constructible.

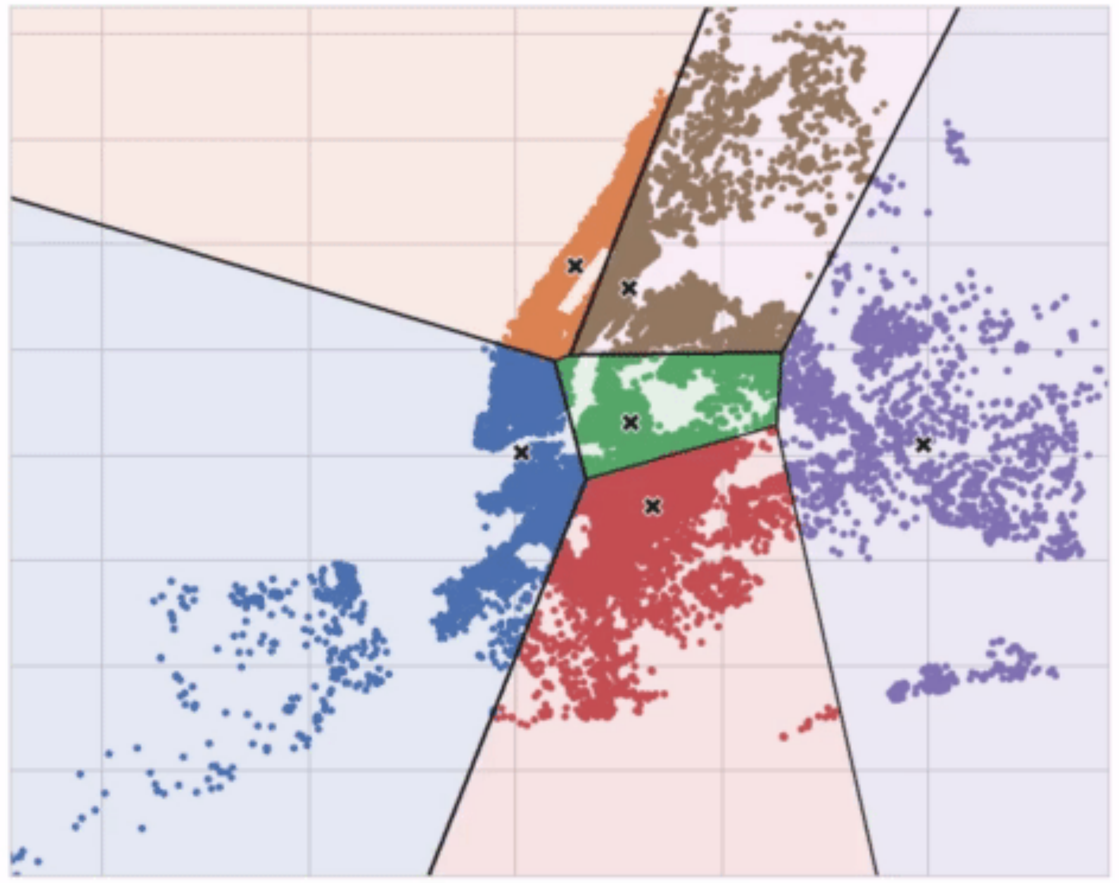
k-means Clustering

Input: database $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ and parameter $k \in \mathbb{N}^+$

Output: $y_1, \ldots, y_k \in \mathbb{R}^d$ minimizing

$$\sum_i \min_{j \in [k]} \|x_i - y_i\|_2^2$$

- Equivalently , $\min\limits_{k\text{-parts } \mathscr{P}} \min\limits_{y_1 \in \mathscr{P}_1, \ldots} \sum\limits_{j \in [k]} \sum\limits_{i \in \mathscr{P}_j} \|x_i - y_j\|_2^2$
- Fix $\mathscr{P}$, minimizer $y_j = \sum\limits_{i \in \mathscr{P}_j} x_i / |\mathscr{P}_j|$ is the centroid
- Plug in, $\min\limits_{k\text{-parts } \mathscr{P}} \sum\limits_{j \in [k]} \sum\limits_{i \neq i' \in \mathscr{P}_j} \|x_i - x_{i'}\|_2^2$



Voronoi partition

Johnson-Lindenstrauss Theorem/Transformation (JLT)

Johnson-Lindenstrauss Theorem

(Johnson-Lindenstrauss 1984)

"In Euclidian space, it is always possible to embed a set of n points in *arbitrary* dimension to $\mathrm{O}(\log n)$ dimension with constant distortion."

Theorem (Johnson-Lindenstrauss 1984):

$\forall 0 < \epsilon < 1$, for any set S of n points from $\mathbb{R}^d$, there is a $\phi : \mathbb{R}^d \to \mathbb{R}^k$ with $k = O(\epsilon^{-2} \log n)$, such that $\forall \boldsymbol{x}, \boldsymbol{y} \in S :$

$$(1 - \epsilon)\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 \leq \|\phi(\boldsymbol{x}) - \phi(\boldsymbol{y})\|_2^2 \leq (1 + \epsilon)\|\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

Johnson-Lindenstrauss Theorem

(Johnson-Lindenstrauss 1984)

"In Euclidian space, it is always possible to embed a set of n points in *arbitrary* dimension to $\mathrm{O}(\log n)$ dimension with constant distortion."

Theorem (Johnson-Lindenstrauss 1984):

$\forall 0 < \epsilon < 1$, for any set S of n points from $\mathbb{R}^d$, there is a $A \in \mathbb{R}^{k\times d}$ with $k = O(\epsilon^{-2}\log n)$, such that $\forall \boldsymbol{x}, \boldsymbol{y} \in S$:

$$(1-\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 \le \|A\boldsymbol{x}-A\boldsymbol{y}\|_2^2 \le (1+\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$$

- The **probabilistic method**: for random $A \in \mathbb{R}^{k\times d}$

$$\Pr\left[\forall \boldsymbol{x}, \boldsymbol{y} \in S : (1-\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 \le \|A\boldsymbol{x}-A\boldsymbol{y}\|_2^2 \le (1+\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2\right] = \underbrace{1 - O\left(\frac{1}{n}\right)}_{\text{w.h.p.}}$$

Theorem (Johnson-Lindenstrauss 1984):

$\forall 0 < \epsilon < 1$, for any set S of n points from $\mathbb{R}^d$, there is a $A \in \mathbb{R}^{k \times d}$ with $k = O(\epsilon^{-2} \log n)$, such that $\forall \boldsymbol{x}, \boldsymbol{y} \in S$:

$$(1-\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 \le \|A\boldsymbol{x}-A\boldsymbol{y}\|_2^2 \le (1+\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$$

- The **probabilistic method**: for random $A \in \mathbb{R}^{k \times d}$

$$\Pr\left[\forall \boldsymbol{x}, \boldsymbol{y} \in S : (1-\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 \le \|A\boldsymbol{x}-A\boldsymbol{y}\|_2^2 \le (1+\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2\right] = 1 - O\left(\frac{1}{n}\right)$$

- **Efficient construction** of random $A \in \mathbb{R}^{k \times d}$:
  - projection onto uniform random k-dimensional subspace; (Johnson-Lindenstrauss; Dasgupta-Gupta)
  - independent Gaussian entries; (Indyk-Motwani)
  - i.i.d. -1/+1 entries; (Achlioptas)

Dimension Reduction

Input: n points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$

Output: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in \mathbb{R}^k$ s.t. $\forall 1 \le i, j \le n$:

$$(1-\epsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 \le \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2 \le (1+\epsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2$$

- for some suitable $k = O(\epsilon^{-2} \log n)$:

J-L Transformation (i.i.d. Gaussian entries)**:**

Entries of $A \in \mathbb{R}^{k \times d}$ are chosen i.i.d. from $\mathcal{N}(0,1/k)$;

(Gaussian distribution with mean 0 and variance $1/k$)

For $i = 1,2,\ldots,n$: let $\boldsymbol{y}_i = A\boldsymbol{x}_i$;

- Gaussian random variable $X \sim \mathcal{N}(\mu, \sigma^2)$:

$$\Pr[X \le t] = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}} \mathrm{d}x$$

$$\mathbb{E}[X] = \mu$$

$$\mathbf{Var}[X] = \sigma^2$$

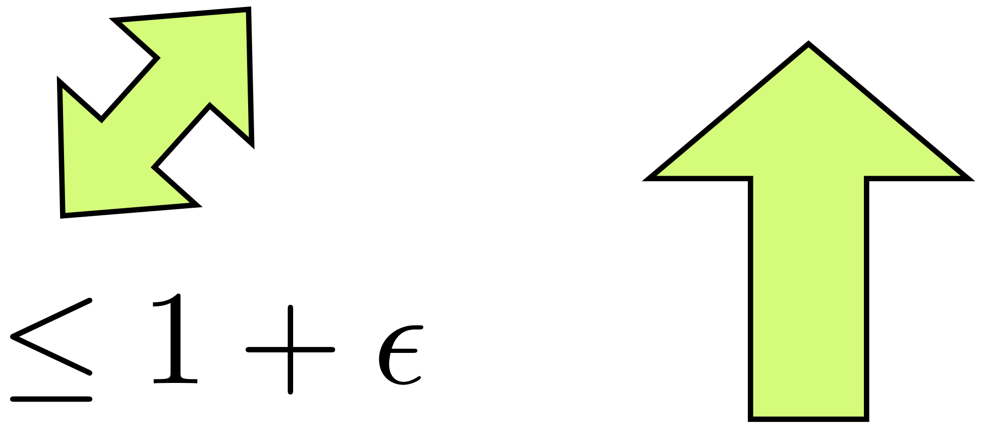
Norm Preservation

- $\forall 0 \le \epsilon \le 1, \forall$ set S of n points from $\mathbb{R}^d$
- Random matrix $A \in \mathbb{R}^{k \times d}$ with $k = (\epsilon^{-2} \log n)$:

Johnson-Lindenstrauss Theorem:

With high probability ($\ge 1 - O(1/n)$): $\forall \boldsymbol{x}, \boldsymbol{y} \in S$,

$$(1-\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 \le \|A\boldsymbol{x}-A\boldsymbol{y}\|_2^2 \le (1+\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$$



$$1-\epsilon \le \left\| A \boxed{\frac{(\boldsymbol{x}-\boldsymbol{y})}{\|\boldsymbol{x}-\boldsymbol{y}\|_2}} \right\|_2^2 \le 1+\epsilon$$

unit vector!

union bound over all $\mathrm{O}(n^2)$ pairs of $\boldsymbol{x}, \boldsymbol{y} \in S$

for any unit vector $\boldsymbol{u} \in \mathbb{R}^d$:

$$\Pr\left[\left|\|A\boldsymbol{u}\|_2^2 - 1\right| > \epsilon\right] < \frac{1}{n^3}$$

$A \in \mathbb{R}^{k\times d}$: each entry of A is chosen i.i.d. from $\mathcal{N}\left(0, \frac{1}{k}\right)$

for any unit vector $\boldsymbol{u} \in \mathbb{R}^d$:

$$\Pr\left[\left|\|A\boldsymbol{u}\|_2^2 - 1\right| > \epsilon\right] < \frac{1}{n^3}$$

$$\|A\boldsymbol{u}\|_2^2 = \sum_{i=1}^{k} (A\boldsymbol{u})_i^2 \xrightarrow{\text{linearity of expectation}} \mathbb{E}\left[\|A\boldsymbol{u}\|_2^2\right] = \sum_{i=1}^{k} \mathbb{E}\left[(A\boldsymbol{u})_i^2\right]$$

$$(A\boldsymbol{u})_i^2 = \left(\sum_{j=1}^{d} A_{ij} u_j\right)^2$$

each $A_{ij} \sim \mathcal{N}\left(0, \frac{1}{k}\right)$ i.i.d.

recall:

$$X \sim \mathcal{N}\left(\mu_1, \sigma_1^2\right), Y \sim \mathcal{N}\left(\mu_2, \sigma_2^2\right) \Longrightarrow X + Y \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$$

$$\Longrightarrow \quad (A\boldsymbol{u})_i = \sum_{j=1}^{d} A_{ij} u_j \sim \mathcal{N}\left(0, \frac{\sum_{j=1}^{d} u_j^2}{k}\right) = \mathcal{N}\left(0, \frac{1}{k}\right)$$

independently!

$A \in \mathbb{R}^{k \times d}$: each entry of A is chosen i.i.d. from $\mathcal{N}\left(0, \frac{1}{k}\right)$

for any unit vector $\boldsymbol{u} \in \mathbb{R}^d$:

$$\Pr\left[\left|\|A\boldsymbol{u}\|_2^2 - 1\right| > \epsilon\right] < \frac{1}{n^3}$$

$$\|A\boldsymbol{u}\|_2^2 = \sum_{i=1}^{k} (A\boldsymbol{u})_i^2 \xrightarrow{\text{linearity of expectation}} \mathbb{E}\left[\|A\boldsymbol{u}\|_2^2\right] = \sum_{i=1}^{k} \mathbb{E}\left[(A\boldsymbol{u})_i^2\right]$$

$(A\boldsymbol{u})_i \sim \mathcal{N}\left(0, \frac{1}{k}\right)$ i.i.d.

$$\Rightarrow \quad \mathbb{E}\left[(A\boldsymbol{u})_i^2\right] = \mathbf{Var}[(A\boldsymbol{u})_i] + \mathbb{E}\left[(A\boldsymbol{u})_i\right]^2 = \frac{1}{k}$$

$$\Rightarrow \quad \mathbb{E}\left[\|A\boldsymbol{u}\|_2^2\right] = \sum_{i=1}^{k} \mathbb{E}\left[(A\boldsymbol{u})_i^2\right] = 1$$

$A \in \mathbb{R}^{k\times d}$: each entry of A is chosen i.i.d. from $\mathcal{N}\left(0, \frac{1}{k}\right)$

for any unit vector $\boldsymbol{u} \in \mathbb{R}^d$:

$$\Pr\left[\left|\|A\boldsymbol{u}\|_2^2 - 1\right| > \epsilon\right] < \frac{1}{n^3}$$

$$\|A\boldsymbol{u}\|_2^2 = \sum_{i=1}^{k} (A\boldsymbol{u})_i^2$$

$(A\boldsymbol{u})_i \sim \mathcal{N}\left(0, \frac{1}{k}\right)$ i.i.d.

$$\mathbb{E}\left[\|A\boldsymbol{u}\|_2^2\right] = 1$$

⇨ for i.i.d. $Y_1, Y_2, \ldots, Y_k \sim \mathcal{N}\left(0, \frac{1}{k}\right)$

$$\Pr\left[\left|\|A\boldsymbol{u}\|_2^2 - 1\right| > \epsilon\right] = \Pr\left[\left|\sum_{i=1}^{k} Y_i^2 - \mathbb{E}\left[\sum_{i=1}^{k} Y_i^2\right]\right| > \epsilon\right]$$

$A \in \mathbb{R}^{k \times d}$: each entry of A is chosen i.i.d. from $\mathcal{N}\left(0, \frac{1}{k}\right)$

for any unit vector $\boldsymbol{u} \in \mathbb{R}^d$:

$$\Pr\left[\left|\|A\boldsymbol{u}\|_2^2 - 1\right| > \epsilon\right] < \frac{1}{n^3}$$

for i.i.d. $Y_1, Y_2, \ldots, Y_k \sim \mathcal{N}\left(0, \frac{1}{k}\right)$ consider $X_i = \sqrt{k} \cdot Y_i$

$$\Pr\left[\left|\|A\boldsymbol{u}\|_2^2 - 1\right| > \epsilon\right] = \Pr\left[\left|\sum_{i=1}^{k} Y_i^2 - \mathbb{E}\left[\sum_{i=1}^{k} Y_i^2\right]\right| > \epsilon\right]$$

Chernoff bound for χ^2-distributions:

For independent $X_1, \ldots, X_k \in \mathcal{N}(0,1)$ $\implies$

$$\begin{cases} \Pr\left[\sum_{i=1}^{k} X_i^2 > (1+\epsilon)k\right] < e^{-\epsilon^2 k/8} \\ \Pr\left[\sum_{i=1}^{k} X_i^2 < (1-\epsilon)k\right] < e^{-\epsilon^2 k/8} \end{cases}$$

$A \in \mathbb{R}^{k\times d}$: each entry of A is chosen i.i.d. from $\mathcal{N}\left(0, \frac{1}{k}\right)$

for any unit vector $\boldsymbol{u} \in \mathbb{R}^d$:

$$\Pr\left[\left|\|A\boldsymbol{u}\|_2^2 - 1\right| > \epsilon\right] < \frac{1}{n^3}$$

for i.i.d. $X_1, X_2, \ldots, X_k \sim \mathcal{N}(0,1)$

$$\Pr\left[\left|\|A\boldsymbol{u}\|_2^2 - 1\right| > \epsilon\right] = \Pr\left[\sum_{i=1}^{k} X_i^2 > (1+\epsilon)k \text{ or } \sum_{i=1}^{k} X_i^2 < (1-\epsilon)k\right]$$

$< \frac{1}{n^3}$ for suitable $k = O(\varepsilon^{-2}\log n)$

Chernoff bound for χ^2-distributions:

For independent $X_1, \ldots, X_k \in \mathcal{N}(0,1) \implies$

$$\begin{cases} \Pr\left[\sum_{i=1}^{k} X_i^2 > (1+\epsilon)k\right] < e^{-\epsilon^2 k/8} \\ \Pr\left[\sum_{i=1}^{k} X_i^2 < (1-\epsilon)k\right] < e^{-\epsilon^2 k/8} \end{cases}$$

Chernoff bound for χ^2-distributions:

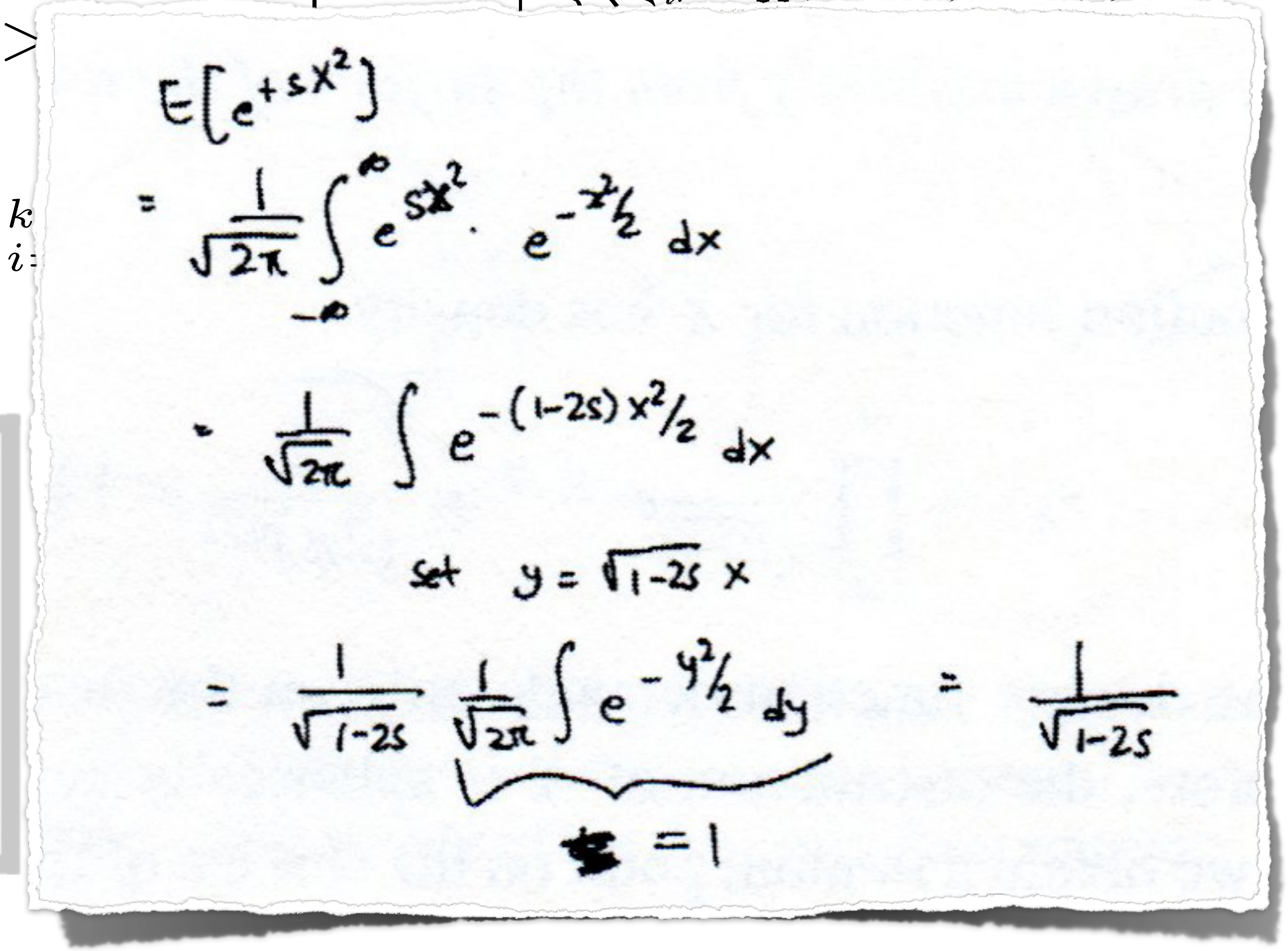
For independent
$X_1, \ldots, X_k \in \mathcal{N}(0,1) \implies \begin{cases} \Pr\left[\sum_{i=1}^k X_i^2 > (1+\epsilon)k\right] < e^{-\epsilon^2 k/8} \\ \Pr\left[\sum_{i=1}^k X_i^2 < (1-\epsilon)k\right] < e^{-\epsilon^2 k/8} \end{cases}$

for all λ>0: $\Pr\left[\sum_{i=1}^{k} X_i^2 >$

$\leq e^{-(1+\epsilon)\lambda k} \cdot \mathbb{E}\left[e^{\lambda \sum_{i}^{k}}\right.$

$X \sim \mathcal{N}(0,1)$ ⇨

$$\mathbb{E}\left[e^{sX^2}\right] = \frac{1}{\sqrt{1-2s}}$$



$$E\left[e^{+sX^2}\right]$$

$$= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{sx^2} \cdot e^{-x^2/2}\, dx$$

$$= \frac{1}{\sqrt{2\pi}} \int e^{-(1-2s)x^2/2}\, dx$$

set $y = \sqrt{1-2s}\, x$

$$= \frac{1}{\sqrt{1-2s}} \underbrace{\frac{1}{\sqrt{2\pi}} \int e^{-y^2/2}\, dy}_{=1} = \frac{1}{\sqrt{1-2s}}$$

Chernoff bound for χ^2-distributions:

For independent

$$X_1, \ldots, X_k \in \mathcal{N}(0,1) \implies \begin{cases} \Pr\left[\sum_{i=1}^k X_i^2 > (1+\epsilon)k\right] < \mathrm{e}^{-\epsilon^2 k/8} \\ \Pr\left[\sum_{i=1}^k X_i^2 < (1-\epsilon)k\right] < \mathrm{e}^{-\epsilon^2 k/8} \end{cases}$$

for all λ>0: $\Pr\left[\sum_{i=1}^{k} X_i^2 > (1+\epsilon)k\right] = \Pr\left[\mathrm{e}^{\lambda \sum_{i=1}^k X_i^2} > \mathrm{e}^{(1+\epsilon)\lambda k}\right]$

$$\leq \mathrm{e}^{-(1+\epsilon)\lambda k} \cdot \mathbb{E}\left[\mathrm{e}^{\lambda \sum_{i=1}^k X_i^2}\right] = \mathrm{e}^{-(1+\epsilon)\lambda k} \cdot \prod_{i=1}^{k} \mathbb{E}\left[\mathrm{e}^{\lambda X_i^2}\right]$$

$X \sim \mathcal{N}(0,1)$ ⇨

$$\mathbb{E}\left[\mathrm{e}^{sX^2}\right] = \frac{1}{\sqrt{1-2s}}$$

$$\leq \mathrm{e}^{-\epsilon\lambda k} \left(\frac{\mathrm{e}^{-\lambda}}{\sqrt{1-2\lambda}}\right)^k$$

$\leq \mathrm{e}^{-\epsilon\lambda k + 2\lambda^2 k}$ for λ<1/4

$= \mathrm{e}^{-\epsilon^2 k/8}$ choosing $\lambda = \varepsilon/4$

Johnson-Lindenstrauss Theorem

(Johnson-Lindenstrauss 1984)

"In Euclidian space, it is always possible to embed a set of n points in *arbitrary* dimension to $\mathrm{O}(\log n)$ dimension with constant distortion."

Theorem (Johnson-Lindenstrauss 1984):

$\forall 0 < \epsilon < 1$, for any set S of n points from $\mathbb{R}^d$, there is a $A \in \mathbb{R}^{k \times d}$ with $k = O(\epsilon^{-2} \log n)$, such that $\forall \boldsymbol{x}, \boldsymbol{y} \in S$:

$$(1-\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 \le \|A\boldsymbol{x}-A\boldsymbol{y}\|_2^2 \le (1+\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$$

- The **probabilistic method**: for random $A \in \mathbb{R}^{k \times d}$

$$\Pr\left[\forall \boldsymbol{x}, \boldsymbol{y} \in S : (1-\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 \le \|A\boldsymbol{x}-A\boldsymbol{y}\|_2^2 \le (1+\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2\right] = 1 - O\left(\frac{1}{n}\right)$$

w.h.p.

Theorem (Johnson-Lindenstrauss 1984):

$\forall 0 < \epsilon < 1$, for any set S of n points from $\mathbb{R}^d$, there is a $A \in \mathbb{R}^{k \times d}$ with $k = O(\epsilon^{-2} \log n)$, such that $\forall \boldsymbol{x}, \boldsymbol{y} \in S$:

$$(1-\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 \le \|A\boldsymbol{x}-A\boldsymbol{y}\|_2^2 \le (1+\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$$

- The probabilistic method: for random $A \in \mathbb{R}^{k \times d}$

$$\Pr\left[\forall \boldsymbol{x}, \boldsymbol{y} \in S : (1-\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 \le \|A\boldsymbol{x}-A\boldsymbol{y}\|_2^2 \le (1+\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2\right] = 1 - O\left(\frac{1}{n}\right)$$

- **Efficient construction** of random $A \in \mathbb{R}^{k \times d}$:
  - projection onto uniform random k-dimensional subspace; (Johnson-Lindenstrauss; Dasgupta-Gupta)
  - independent Gaussian entries; (Indyk-Motwani)
  - i.i.d. -1/+1 entries; (Achlioptas)

Dimension Reduction

Input: n points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$

Output: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in \mathbb{R}^k$ s.t. $\forall 1 \le i, j \le n$:

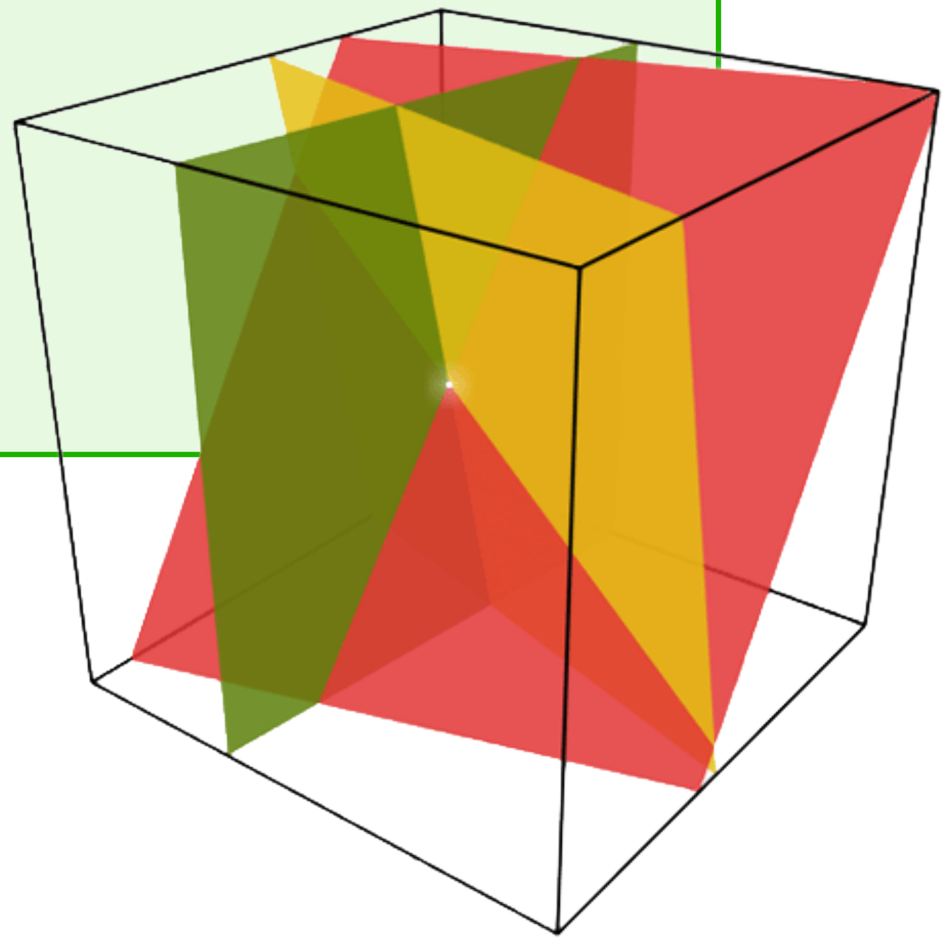
$$(1-\epsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 \le \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2 \le (1+\epsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2$$

- for some suitable $k = O(\epsilon^{-2}\log n)$:

J-L Transformation (uniform *k*-dim subspace)**:**

The k rows $A_1, \ldots, A_k$ of $A \in \mathbb{R}^{k\times d}$ are **orthogonal unit** vectors $\in \mathbb{R}^d$ chosen uniform at random;

For $i = 1,2,\ldots,n$: let $\boldsymbol{y}_i = \sqrt{\frac{d}{k}} A\boldsymbol{x}_i$;



- $A \in \mathbb{R}^{k\times d}$: projection onto a uniform *k*-dimensional subspace

$A \in \mathbb{R}^{k \times d}$: projection onto uniform random k-dim subspace

for any unit vector $\boldsymbol{u} \in \mathbb{R}^d$:

$$\Pr\left[\left|\left\|\sqrt{\frac{d}{k}}A\boldsymbol{u}\right\|_2^2 - 1\right| > \epsilon\right] < \frac{1}{n^3}$$

$$\Uparrow$$

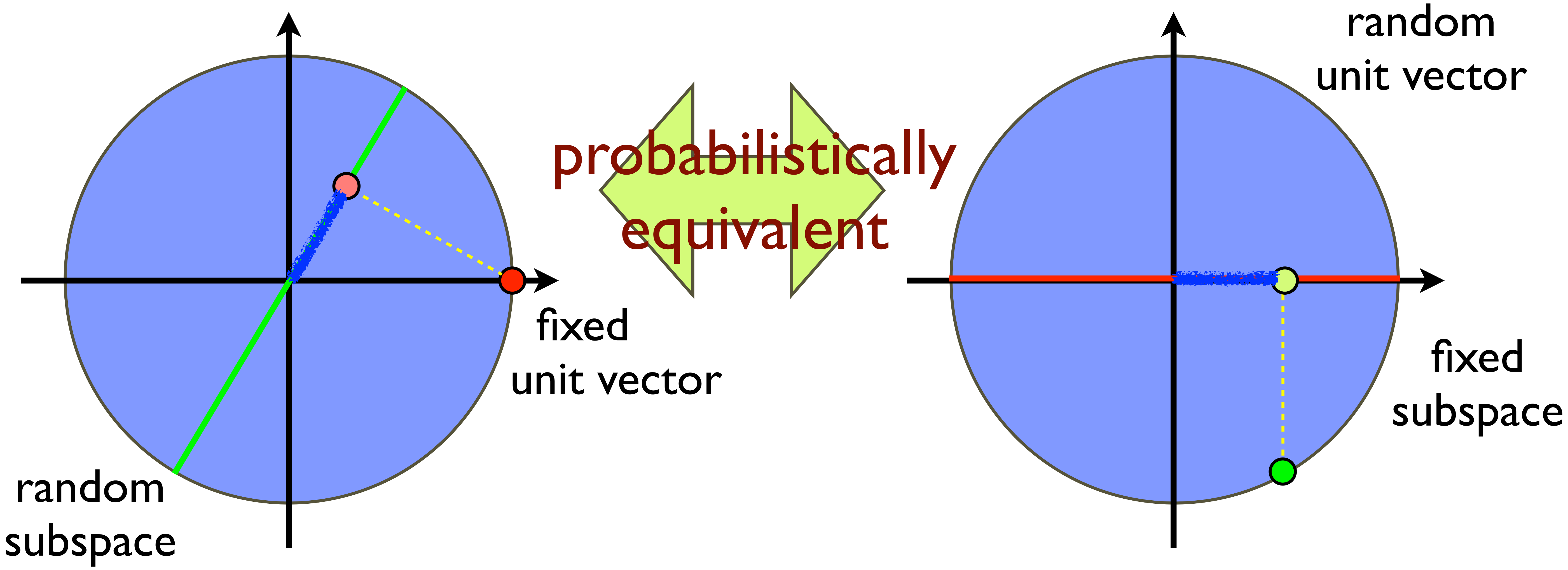
$$\Pr\left[\|A\boldsymbol{u}\|_2^2 > (1+\epsilon)\frac{k}{d} \text{ or } \|A\boldsymbol{u}\|_2^2 < (1-\epsilon)\frac{k}{d}\right] < \frac{1}{n^3}$$

$A \in \mathbb{R}^{k\times d}$: projection onto uniform random k-dim subspace

for any unit vector $\boldsymbol{u} \in \mathbb{R}^d$:

$$\Pr\left[\, \| A\boldsymbol{u} \|_2^2 > (1+\epsilon)\frac{k}{d} \right] < \frac{1}{2n^3}$$

$$\Pr\left[\, \| A\boldsymbol{u} \|_2^2 < (1-\epsilon)\frac{k}{d} \right] < \frac{1}{2n^3}$$



"inner-products are invariant under rotations"

$A \in \mathbb{R}^{k \times d}$: projection onto uniform random k-dim subspace

for any unit vector $\boldsymbol{u} \in \mathbb{R}^d$:

$$\Pr\left[\| A\boldsymbol{u} \|_2^2 > (1+\epsilon)\frac{k}{d}\right] < \frac{1}{2n^3}$$

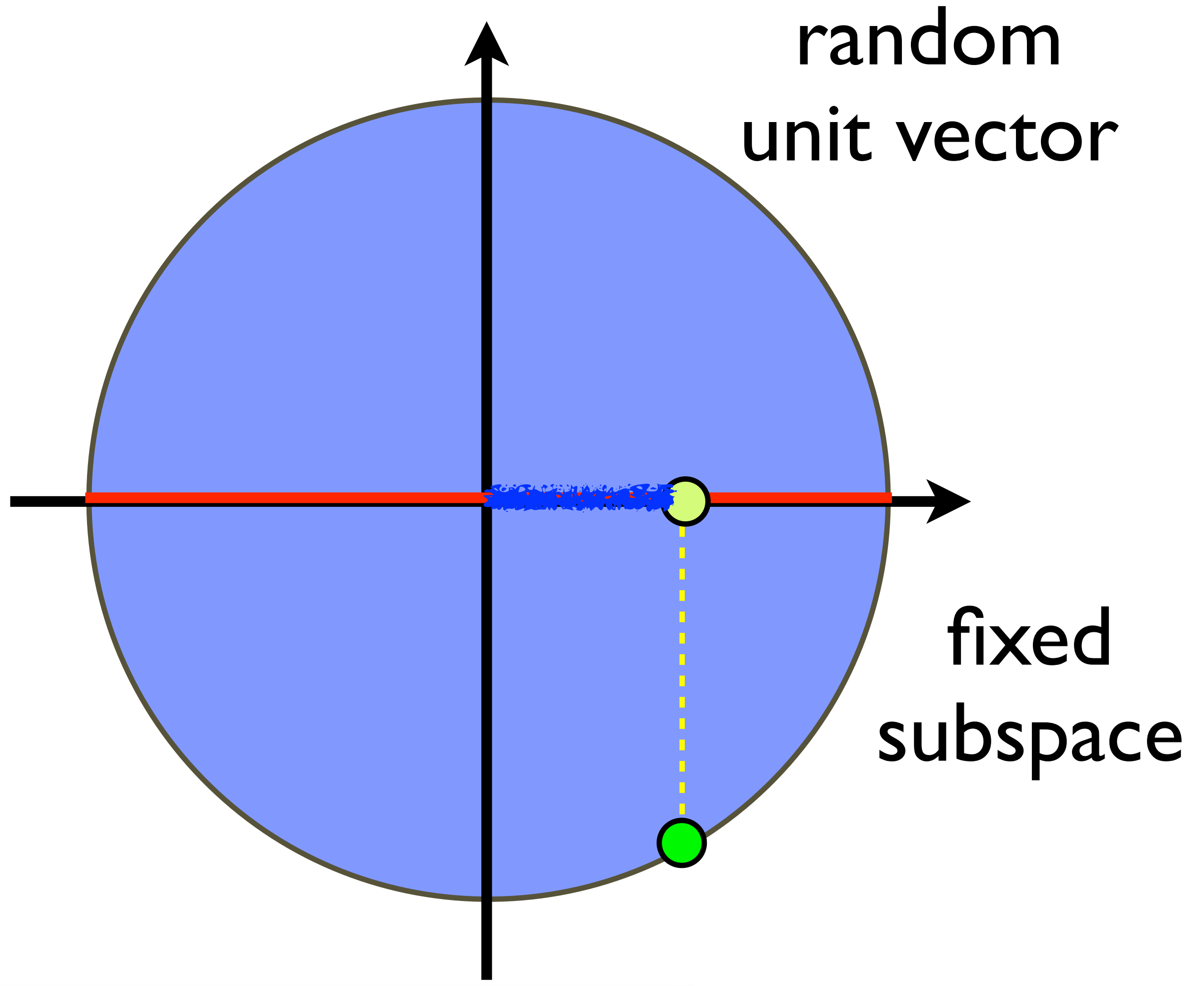
$$\Pr\left[\| A\boldsymbol{u} \|_2^2 < (1-\epsilon)\frac{k}{d}\right] < \frac{1}{2n^3}$$

uniform random unit vector $\in \mathbb{R}^d$:

$$Y = (Y_1, \ldots, Y_k, Y_{k+1}, \ldots, Y_d)$$

fixed k-dimensional subspace:

$$Z = (Y_1, \ldots, Y_k)$$



$\|A\boldsymbol{u}\|$ is identically distributed as $\|Z\|$

uniform random unit vector $\in \mathbb{R}^d$: $\quad Y = (Y_1, \ldots, Y_d)$

sample $X = (X_1, \ldots, X_d) \in \mathbb{R}^d$ where $X_i \sim \mathcal{N}(0,1)$ *i.i.d.*;

let $Y = \dfrac{X}{\|X\|}$;

density: $$\Pr[X = \boldsymbol{x}] = \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-x_i^2/2} = (2\pi)^{-d/2} \mathrm{e}^{-\|\boldsymbol{x}\|_2^2/2}$$

Spherically symmetric!

for some suitable $k = O(\varepsilon^{-2} \log n)$:

$$\Pr\left[\sum_{i=1}^{k} Y_i^2 > (1+\epsilon)\frac{k}{d}\right] < \frac{1}{2n^3}$$

$$\Pr\left[\sum_{i=1}^{k} Y_i^2 < (1-\epsilon)\frac{k}{d}\right] < \frac{1}{2n^3}$$

i.i.d. Gaussian random variables $X_1, X_2, \ldots, X_d \sim \mathcal{N}(0,1)$

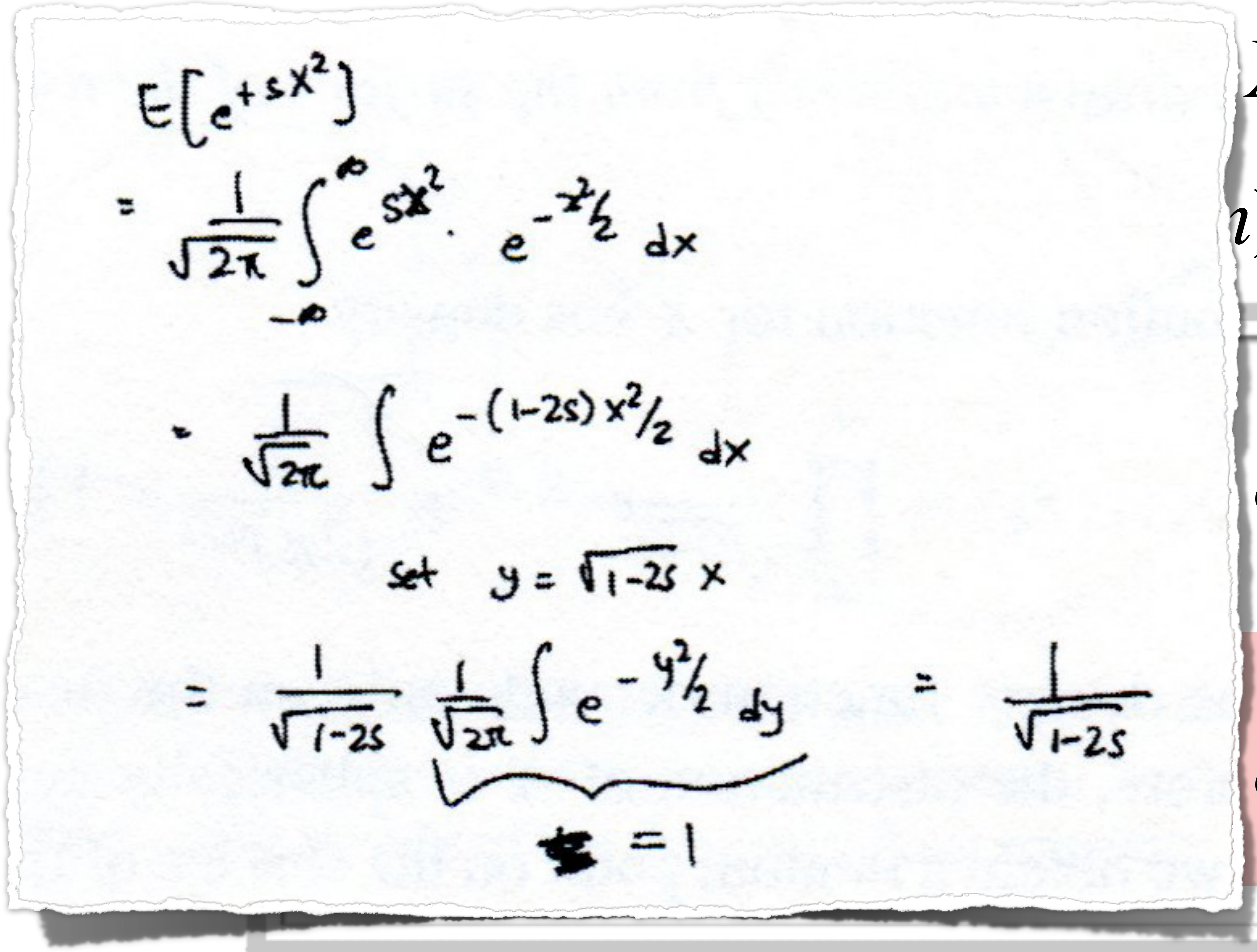
for some suitable $k = \mathrm{O}(\varepsilon^{-2}\log n)$:

$$\Pr\left[\sum_{i=1}^{k} X_i^2 > (1+\epsilon)\frac{k}{d}\sum_{i=1}^{d} X_i^2\right] < \frac{1}{2n^3}$$

$$\Pr\left[\sum_{i=1}^{k} X_i^2 < (1-\epsilon)\frac{k}{d}\sum_{i=1}^{d} X_i^2\right] < \frac{1}{2n^3}$$

$$\Pr\left[(d-(1+\epsilon)k)\sum_{i=1}^{k} X_i^2 - (1+\epsilon)k\sum_{i=k+1}^{d} X_i^2 > 0\right] < \frac{1}{2n^3}$$

$$\Pr\left[(d-(1-\epsilon)k)\sum_{i=1}^{k} X_i^2 - (1-\epsilon)k\sum_{i=k+1}^{d} X_i^2 < 0\right] < \frac{1}{2n^3}$$



$$\mathbb{E}\left[e^{+sX^2}\right]$$

$$= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{sx^2} \cdot e^{-x^2/2}\, dx$$

$$= \frac{1}{\sqrt{2\pi}} \int e^{-(1-2s)x^2/2}\, dx$$

set $y = \sqrt{1-2s}\, x$

$$= \frac{1}{\sqrt{1-2s}} \underbrace{\frac{1}{\sqrt{2\pi}} \int e^{-y^2/2}\, dy}_{=1} = \frac{1}{\sqrt{1-2s}}$$

$X_1, X_2, \ldots, X_d \sim \mathcal{N}(0,1)$

n):

$$\epsilon)k \sum_{i=k+1}^{d} X_i^2 > 0\Bigg] < \frac{1}{2n^3}$$

$$\epsilon)k \sum_{i=k+1}^{d} X_i^2 < 0\Bigg] < \frac{1}{2n^3}$$

🟥 $$= \Pr\left[\exp\left\{\lambda\left(((1-\epsilon)k-d)\sum_{i=1}^{k} X_i^2 + (1-\epsilon)k\sum_{i=k+1}^{d} X_i^2\right)\right\} > 1\right]$$ (arbitrary $\lambda > 0$)

$$\leq \mathbb{E}\left[\exp\left\{\lambda\left(((1-\epsilon)k-d)\sum_{i=1}^{k} X_i^2 + (1-\epsilon)k\sum_{i=k+1}^{d} X_i^2\right)\right\}\right]$$ (Markov's inequality)

$$= \prod_{i=1}^{k} \mathbb{E}\left[e^{\lambda((1-\epsilon)k-d)X_i^2}\right] \prod_{i=k+1}^{d} \mathbb{E}\left[e^{\lambda(1-\epsilon)kX_i^2}\right]$$

i.i.d. Gaussian random variables $X_1, X_2, \ldots, X_d \sim \mathcal{N}(0,1)$

for some suitable $k = \mathrm{O}(\varepsilon^{-2}\log n)$:

$$\Pr\left[(d-(1+\epsilon)k)\sum_{i=1}^{k} X_i^2 - (1+\epsilon)k\sum_{i=k+1}^{d} X_i^2 > 0\right] < \frac{1}{2n^3}$$

$$\Pr\left[(d-(1-\epsilon)k)\sum_{i=1}^{k} X_i^2 - (1-\epsilon)k\sum_{i=k+1}^{d} X_i^2 < 0\right] < \frac{1}{2n^3}$$

🟥 $= \Pr\left[\exp\left\{\lambda\left(((1-\epsilon)k-d)\sum_{i=1}^{k} X_i^2 + (1-\epsilon)k\sum_{i=k+1}^{d} X_i^2\right)\right\} > 1\right]$ (arbitrary $\lambda > 0$)

$$\leq (1-2\lambda((1-\epsilon)k-d))^{-\frac{k}{2}}(1-2\lambda(1-\epsilon)k)^{-\frac{d-k}{2}}$$

set:

$$\lambda = \frac{\epsilon}{2(1-\epsilon)(d-(1-\epsilon)k)}$$

$$\leq (1-\epsilon)^{-\frac{k}{2}}\left(1-\frac{\epsilon k}{d-k}\right)^{\frac{d-k}{2}} \leq \exp\left(-\frac{\epsilon^2 k}{4}\right)$$

Theorem (Johnson-Lindenstrauss 1984):

$\forall 0 < \epsilon < 1$, for any set S of n points from $\mathbb{R}^d$, there is a $A \in \mathbb{R}^{k\times d}$ with $k = O(\epsilon^{-2}\log n)$, such that $\forall \boldsymbol{x}, \boldsymbol{y} \in S$:

$$(1-\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 \le \|A\boldsymbol{x}-A\boldsymbol{y}\|_2^2 \le (1+\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$$

- The **probabilistic method**: for random $A \in \mathbb{R}^{k\times d}$

$$\Pr\left[\forall \boldsymbol{x}, \boldsymbol{y} \in S : (1-\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 \le \|A\boldsymbol{x}-A\boldsymbol{y}\|_2^2 \le (1+\epsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2\right] = 1 - O\left(\frac{1}{n}\right)$$

- **Efficient construction** of random $A \in \mathbb{R}^{k\times d}$:
  - projection onto uniform random k-dimensional subspace; (Johnson-Lindenstrauss; Dasgupta-Gupta)
  - independent Gaussian entries; (Indyk-Motwani)
  - i.i.d. -1/+1 entries; (Achlioptas)

Tug-Of-War Algorithm

Count Sketch: z

Upon each x_i: $z \leftarrow z + \sigma(x_i)$

Query: return z^2

Count Sketch+: $\mathrm{CS}[k]$ (initialized to all 0's)

Upon each x_i: $\mathrm{CS}[j] \leftarrow \mathrm{CS}[j] + \sigma_j(x_i)$, for all $j \le k$

Query: return $z_+^2 = \sum \mathrm{CS}[j]^2 / k$

Chebyshev's Inequality

For random variable X, for any $t > 0$,

$$\Pr\left[\,|X - \mathbb{E}[X]| \ge t\,\right] \le \frac{\mathbf{Var}[X]}{t^2}$$

for any unit vector $\mathbf{u} \in \mathbb{R}^d$:

$$\Pr\left[\left|\,\|A\mathbf{u}\|_2^2 - 1\right| \ge \epsilon\right] \le \delta$$

- Unbiased: $\mathbb{E}[z^2] = \|x\|_2^2$
- Claim: $\mathbf{Var}(z^2) = O(\|x\|_2^2)$ for $A \in \{-1, +1\}^{1\times d}$
- $\mathbf{Var}(z_+^2) = O(\|x\|_2^2/k)$ for $A \in \left\{-1/\sqrt{k}, +1/\sqrt{k}\right\}^{k\times d}$
- $\Pr\left[\left|\,\|A\mathbf{u}\|_2^2 - 1\right| \ge \epsilon\right] \le \mathbf{Var}(z_+^2)/\epsilon^2 = O(1/k\epsilon^2) =: \delta$

Any issue?

Chernoff bound for χ^2-distributions:

For independent $X_1, \dots, X_k \in \mathcal{N}(0,1) \implies \begin{cases} \Pr\left[\sum_{i=1}^k X_i^2 > (1+\epsilon)k\right] < \mathrm{e}^{-\epsilon^2 k/8} \\ \Pr\left[\sum_{i=1}^k X_i^2 < (1-\epsilon)k\right] < \mathrm{e}^{-\epsilon^2 k/8} \end{cases}$

for all λ>0: $\Pr\left[\sum_{i=1}^k X_i^2 > (1+\epsilon)k\right] = \Pr\left[\mathrm{e}^{\lambda \sum_{i=1}^k X_i^2} > \mathrm{e}^{(1+\epsilon)\lambda k}\right]$

$$\leq \mathrm{e}^{-(1+\epsilon)\lambda k} \cdot \mathbb{E}\left[\mathrm{e}^{\lambda \sum_{i=1}^k X_i^2}\right] = \mathrm{e}^{-(1+\epsilon)\lambda k} \cdot \prod_{i=1}^k \mathbb{E}\left[\mathrm{e}^{\lambda X_i^2}\right]$$

$X \sim \mathcal{N}(0,1)$ ⇨

$$\mathbb{E}\left[\mathrm{e}^{sX^2}\right] = \frac{1}{\sqrt{1-2s}}$$

$$\leq \mathrm{e}^{-\epsilon\lambda k}\left(\frac{\mathrm{e}^{-\lambda}}{\sqrt{1-2\lambda}}\right)^k$$

$\leq \mathrm{e}^{-\epsilon\lambda k + 2\lambda^2 k}$ for λ<1/4

$= \mathrm{e}^{-\epsilon^2 k/8}$ choosing $\lambda = \varepsilon/4$

Count Sketch versus Gaussian

Count Sketch+: $\mathrm{CS}[k]$ (initialized to all 0's)

Upon each x_i: $\mathrm{CS}[j] \leftarrow \mathrm{CS}[j] + \sigma_j(x_i)$, for all $j \leq k$

Query: return $z_+^2 = \sum \mathrm{CS}[j]^2 / k$

$$M_{X^2}(t) \triangleq \mathbb{E}\left[\exp(tX^2)\right] = \sum_{k \geq 0} \frac{t^k}{k!} \mathbb{E}\left[X^{2k}\right]$$

- Count Sketch is not worse than Gaussian. Let $\mathbf{w} \triangleq (1,\ldots,1), \mathbf{Z} \sim \mathcal{N}^d(0,1), \mathbf{r} \sim \{-1,+1\}^d$
  - Claim: $\mathbb{E}\left[\langle \mathbf{x}, \mathbf{r} \rangle^{2k}\right] \leq \mathbb{E}\left[\langle \mathbf{w}, \mathbf{r} \rangle^{2k}\right] \leq \mathbb{E}\left[\langle \mathbf{w}, \mathbf{Z} \rangle^{2k}\right]$ for all $k \in \mathbb{N}$, $\mathbf{x} \in \mathbb{R}^d$ with $\|\mathbf{x}\|_2^2 = d$
- Eq(1): intuitively provable by symmetry.
- Eq(2): $\mathbb{E}\left[\langle \mathbf{w}, \mathbf{r} \rangle^{2k}\right] = \sum_{i_1,\ldots,i_{2k}} \mathbb{E}[r_{i_1} \cdot \ldots \cdot r_{i_{2k}}]$ is zero if any multiplicity is odd
  - Suffices to consider only $\sum_{i_1,\ldots,i_{2k}} \mathbb{E}[r_{i_1}^{2\ell_1}] \cdot \ldots \cdot \mathbb{E}[r_{i_{2k}}^{2\ell_{2k}}]$, so be $\mathbb{E}\left[\langle \mathbf{w}, \mathbf{Z} \rangle^{2k}\right]$.

Count Sketch versus Gaussian

Count Sketch+: $\mathrm{CS}[k]$ (initialized to all 0's)

Upon each x_i: $\mathrm{CS}[j] \leftarrow \mathrm{CS}[j] + \sigma_j(x_i)$, for all $j \leq k$

Query: return $z_+^2 = \sum \mathrm{CS}[j]^2 / k$

$$M_{X^2}(t) \triangleq \mathbb{E}\left[\exp(tX^2)\right] = \sum_{k \geq 0} \frac{t^k}{k!} \mathbb{E}\left[X^{2k}\right]$$

- Count Sketch is not worse than Gaussian. Let $\mathbf{w} \triangleq (1,\ldots,1), \mathbf{Z} \sim \mathcal{N}^d(0,1), \mathbf{r} \sim \{-1,+1\}^d$
  - Claim: $\mathbb{E}\left[\langle \mathbf{x}, \mathbf{r} \rangle^{2k}\right] \leq \mathbb{E}\left[\langle \mathbf{w}, \mathbf{r} \rangle^{2k}\right] \leq \mathbb{E}\left[\langle \mathbf{w}, \mathbf{Z} \rangle^{2k}\right]$ for all $k \in \mathbb{N}$, $\mathbf{x} \in \mathbb{R}^d$ with $\|\mathbf{x}\|_2^2 = d$
- Eq(2): Suffices to consider only $\displaystyle\sum_{i_1,\ldots,i_{2k}} \mathbb{E}\left[r_{i_1}^{2\ell_1}\right] \cdot \ldots \cdot \mathbb{E}\left[r_{i_{2k}}^{2\ell_{2k}}\right]$, so be $\mathbb{E}\left[\langle \mathbf{w}, \mathbf{Z} \rangle^{2k}\right]$.
  - Claim: $\mathbb{E}\left[r_{i_1}^{2\ell_1}\right] \leq \mathbb{E}\left[Z_{i_1}^{2\ell_1}\right]$
  - $\mathbb{E}\left[r_{i_1}^{2\ell_1}\right] = 1, \mathbb{E}\left[Z_{i_1}^{2\ell_1}\right] = \dfrac{(2\ell_1)!}{(\ell!2^\ell)} \geq 1$

Nearest Neighbor Search (NNS)

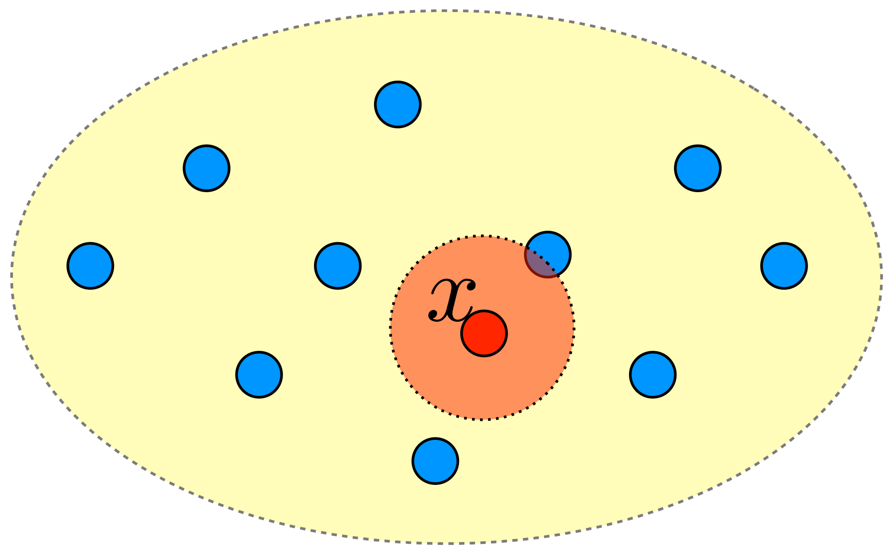
Nearest Neighbor Search (NNS)

- Metric space (X, dist):

Data: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in X$

Query: a point $\boldsymbol{x} \in X$

Find the datapoint $\boldsymbol{y}_i$ that is closest to $\boldsymbol{x}$.



Applications in:

- database systems
- pattern matching
- machine learning
- image processing
- bioinformatics
- … …

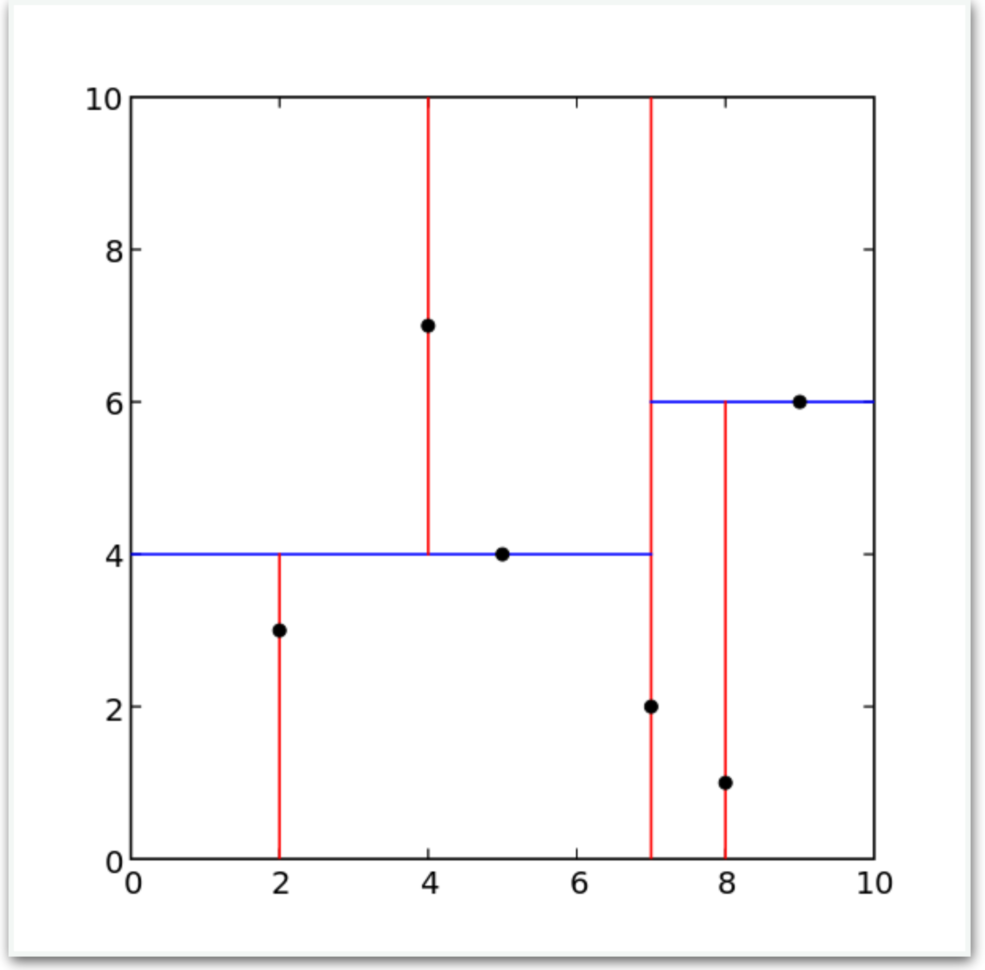
Nearest Neighbor Search (NNS)

Data: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in [N]^d$

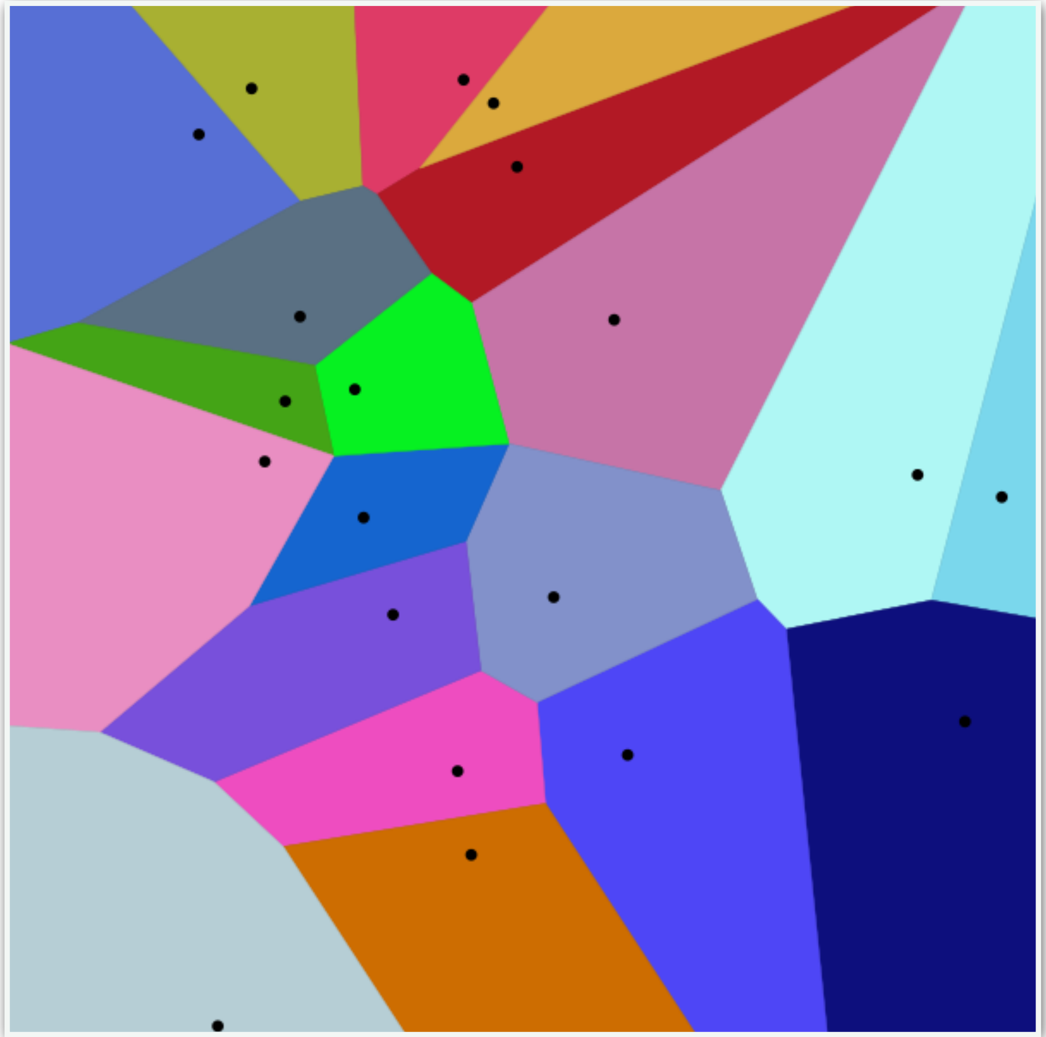
Query: a point $\boldsymbol{x} \in [N]^d$

Find the datapoint $\boldsymbol{y}_i$ that is closest to $\boldsymbol{x}$.

when dimension d is small:



k-d tree



Voronoi diagram

Nearest Neighbor Search (NNS)

- Hamming space $\{0,1\}^d$:

Data: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in \{0,1\}^d$

Query: a point $\boldsymbol{x} \in \{0,1\}^d$

Find the datapoint $\boldsymbol{y}_i$ that is closest to $\boldsymbol{x}$.

when dimension d is high:

say $d \gg \log n$

Curse of dimensionality:

It is conjectured that to solve NNS in high dimension requires either super-polynomial(*n*) space or super-polynomial(*d*) time.

Blessing: **randomization + approximation**

Approximate Near Neighbor (ANN)

- Metric space (X, dist):

Data: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in X$

Query: a point $\boldsymbol{x} \in X$

c-ANN (*Approximate* ***Nearest*** *Neighbor*):

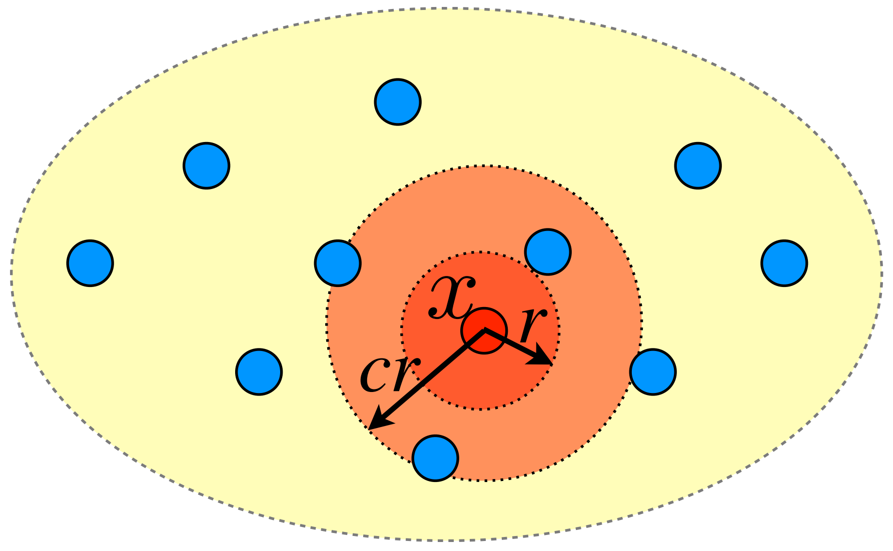
Find a $\boldsymbol{y}_i$ such that $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) \leq c \cdot \min_{1 \leq j \leq n} \mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_j)$

(c, r)-ANN (*Approximate* ***Near*** *Neighbor*):

return a $\boldsymbol{y}_i$ that $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) \leq c \cdot r$ if $\exists \boldsymbol{y}_j$ s.t. $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_j) \leq r$

"no" if $\forall \boldsymbol{y}_i$, $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) > c \cdot r$

arbitrary if otherwise



$$r_0 = D_{\min} = \min_{1 \leq i < j \leq n} \mathrm{dist}(\boldsymbol{y}_i, \boldsymbol{y}_j)$$

$$r_k = \sqrt{c} \cdot r_{k-1}$$

$$r_{\log_c(D_{\max}/D_{\min})} = D_{\max} = \max_{1 \leq i < j \leq n} \mathrm{dist}(\boldsymbol{y}_i, \boldsymbol{y}_j)$$

- Metric space (X, dist):

Data: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in X$

Query: a point $\boldsymbol{x} \in X$

c-ANN (*Approximate* ***N**earest* *Neighbor*):

Find a $\boldsymbol{y}_i$ such that $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) \le c \cdot \min\limits_{1 \le j \le n} \mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_j)$

(c, r)-ANN (*Approximate* ***N**ear* *Neighbor*):

return a $\boldsymbol{y}_i$ that $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) \le c \cdot r$ if $\exists \boldsymbol{y}_j$ s.t. $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_j) \le r$

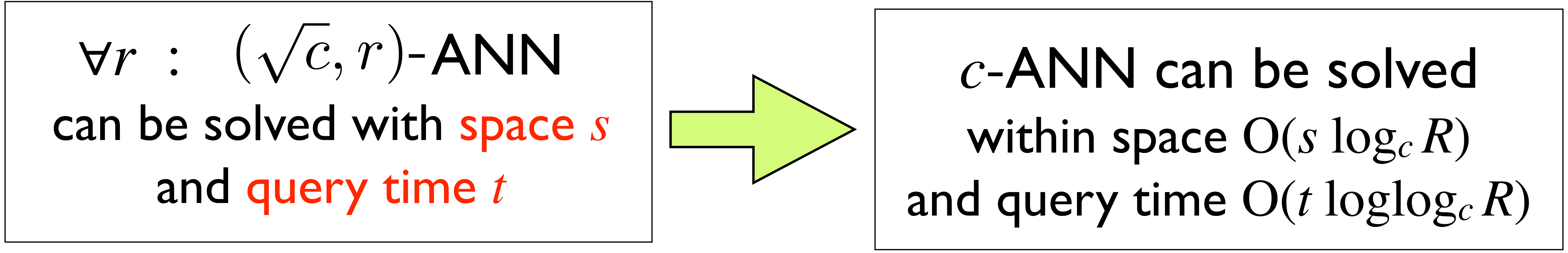
"no" if $\forall \boldsymbol{y}_i$, $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) > c \cdot r$

arbitrary if otherwise

let $R = \dfrac{D_{\max}}{D_{\min}}$

$$D_{\max} = \max_{1 \le i < j \le n} \mathrm{dist}(\boldsymbol{y}_i, \boldsymbol{y}_j)$$

$$D_{\min} = \min_{1 \le i < j \le n} \mathrm{dist}(\boldsymbol{y}_i, \boldsymbol{y}_j)$$



$\forall r$: $(\sqrt{c}, r)$-ANN

can be solved with space s

and query time t

c-ANN can be solved

within space $\mathrm{O}(s \log_c R)$

and query time $\mathrm{O}(t \log\log_c R)$

- Hamming space $\{0,1\}^d$:

Data: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in \{0,1\}^d$

Query: a point $\boldsymbol{x} \in \{0,1\}^d$

(c, r)-**ANN** (*Approximate Near Neighbor*):

return a $\boldsymbol{y}_i$ that $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) \le c \cdot r$ if $\exists \boldsymbol{y}_j$ s.t. $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_j) \le r$

answer "**no**" if $\forall \boldsymbol{y}_i$, $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) > c \cdot r$

arbitrary if otherwise

- High dimension: $d \gg \log n$

$$\boldsymbol{z}_i(j) = (A\boldsymbol{y}_i)_j = \left(\sum_{\ell=1}^{d} A_{j\ell}\boldsymbol{y}_i(\ell)\right) \bmod 2$$

Dimension Reduction:

Let k, p and s to be fixed later;

sample $k \times d$ Boolean matrix A with *i.i.d.* entries $\in$ Bernoulli(p);

for $i = 1,2,\ldots,n$: let $\boldsymbol{z}_i = A\boldsymbol{y}_i \in \{0,1\}^k$ on finite field $\mathrm{GF}(2)$;

store all s-balls $B_s(\boldsymbol{u}) = \{\boldsymbol{y}_i \mid \mathrm{dist}(\boldsymbol{u}, \boldsymbol{z}_i) \le s\}$ for all $\boldsymbol{u} \in \{0,1\}^k$;

- Hamming space $\{0,1\}^d$:

Data: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in \{0,1\}^d$

Query: a point $\boldsymbol{x} \in \{0,1\}^d$

- High dimension: $d \gg \log n$

$$\boldsymbol{z}_i(j) = (A\boldsymbol{y}_i)_j = \left(\sum_{\ell=1}^{d} A_{j\ell}\,\boldsymbol{y}_i(\ell)\right) \bmod 2$$

Dimension Reduction:

Let k, p and s to be fixed later;

sample $k \times d$ Boolean matrix A with *i.i.d.* entries $\in$ Bernoulli(p);

for $i = 1,2,\ldots,n$: let $\boldsymbol{z}_i = A\boldsymbol{y}_i \in \{0,1\}^k$ on finite field $\mathrm{GF}(2)$;

store all s-balls $B_s(\boldsymbol{u}) = \{\boldsymbol{y}_i \mid \mathrm{dist}(\boldsymbol{u}, \boldsymbol{z}_i) \le s\}$ for all $\boldsymbol{u} \in \{0,1\}^k$;

To answer **query** $\boldsymbol{x} \in \{0,1\}^d$: retrieve $B_s(A\boldsymbol{x})$;

if $B_s(A\boldsymbol{x}) = \emptyset$ return "**no**"

else return any $\boldsymbol{y}_i \in B_s(A\boldsymbol{x})$

space: $\mathrm{O}(n2^k)$ query time: $\mathrm{O}(kd)$ computation + $\mathrm{O}(1)$ memory access

- Hamming space $\{0,1\}^d$:

Data: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in \{0,1\}^d$

Query: a point $\boldsymbol{x} \in \{0,1\}^d$

- High dimension: $d \gg \log n$

$$\boldsymbol{z}_i(j) = (A\boldsymbol{y}_i)_j = \left(\sum_{\ell=1}^{d} A_{j\ell}\boldsymbol{y}_i(\ell)\right) \bmod 2$$

Dimension Reduction:

Let k, p and s to be fixed later;

sample $k \times d$ Boolean matrix A with *i.i.d.* entries $\in$ Bernoulli(p);

for $i = 1,2,\ldots,n$: let $\boldsymbol{z}_i = A\boldsymbol{y}_i \in \{0,1\}^k$ on finite field $\mathrm{GF}(2)$;

for suitable $k = O(\log n)$, p and s:

$$\forall \boldsymbol{x},\boldsymbol{y} \in \{0,1\}^d: \quad \begin{aligned} \mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) \le r \quad &\Rightarrow \quad \Pr[\ \mathrm{dist}(A\boldsymbol{x}, A\boldsymbol{y}) > s] < 1/n^2 \\ \mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) > c{\cdot}r \quad &\Rightarrow \quad \Pr[\ \mathrm{dist}(A\boldsymbol{x}, A\boldsymbol{y}) \le s] < 1/n^2 \end{aligned}$$

union bound (c,r)-ANN is solved w.h.p.

random $k \times d$ Boolean matrix A with *i.i.d.* entries $\in$ Bernoulli(p);

computation on GF(2): $$(A\boldsymbol{x})_i = \left(\sum_{j=1}^{d} A_{ij} x_i\right) \bmod 2$$

for suitable p and s:

$\forall \boldsymbol{x},\boldsymbol{y} \in \{0,1\}^d$: dist($\boldsymbol{x},\boldsymbol{y}$) $\le r$ $\Rightarrow$ Pr[dist($A\boldsymbol{x}$, $A\boldsymbol{y}$) $> s$] $< e^{-\Omega(k)}$

dist($\boldsymbol{x},\boldsymbol{y}$) $> c\cdot r$ $\Rightarrow$ Pr[dist($A\boldsymbol{x}$, $A\boldsymbol{y}$) $\le s$] $< e^{-\Omega(k)}$

row vector $A_{i\cdot}$: *i.i.d.* entries $\in$ Bernoulli(p)

$$\Pr[(A\boldsymbol{x})_i \neq (A\boldsymbol{y})_i] = \Pr[\langle A_{i\cdot}, \boldsymbol{x}\rangle \neq \langle A_{i\cdot}, \boldsymbol{y}\rangle] = \frac{1}{2}\left(1 - (1-2p)^{\mathrm{dist}(\boldsymbol{x},\boldsymbol{y})}\right)$$

Why?

For uniform $\boldsymbol{u} \in \{0,1\}^d$: $$\Pr[\langle \boldsymbol{u}, \boldsymbol{x}\rangle \neq \langle \boldsymbol{u}, \boldsymbol{y}\rangle] = \frac{1}{2}$$

generate $A_{i\cdot}$ as:

1. each $j \in [d]$ joins $D \subseteq [d]$ independently with probability $2p$;
2. for each $j \in D$: samples a uniform and independent $A_{ij} \in \{0,1\}$;
3. for each $j \notin D$: $A_{ij}=0$;

$A_{i\cdot}$ restricted on D is a uniform Boolean vector!

random $k \times d$ Boolean matrix A with *i.i.d.* entries $\in$ Bernoulli(p);

computation on GF(2): $$(A\boldsymbol{x})_i = \left(\sum_{j=1}^{d} A_{ij} x_i\right) \bmod 2$$

for suitable p and s:

$$\forall \boldsymbol{x},\boldsymbol{y} \in \{0,1\}^d: \quad \begin{array}{ll} \mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) \le r & \Rightarrow \Pr[\,\mathrm{dist}(A\boldsymbol{x}, A\boldsymbol{y}) > s] < e^{-\Omega(k)} \\ \mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) > c \cdot r & \Rightarrow \Pr[\,\mathrm{dist}(A\boldsymbol{x}, A\boldsymbol{y}) \le s] < e^{-\Omega(k)} \end{array}$$

row vector $A_{i\cdot}$: *i.i.d.* entries $\in$ Bernoulli(p)

$$\Pr[(A\boldsymbol{x})_i \neq (A\boldsymbol{y})_i] = \Pr[\langle A_{i\cdot}, \boldsymbol{x}\rangle \neq \langle A_{i\cdot}, \boldsymbol{y}\rangle] = \frac{1}{2}\left(1 - (1-2p)^{\mathrm{dist}(\boldsymbol{x},\boldsymbol{y})}\right)$$

choose p to satisfy $(1-2p) = 2^{-1/r}$

$$\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) \le r \quad \Rightarrow \quad \Pr[\,(A\boldsymbol{x})_i \neq (A\boldsymbol{y})_i\,] \le 1/4$$

$$\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) > c \cdot r \quad \Rightarrow \quad \Pr[\,(A\boldsymbol{x})_i \neq (A\boldsymbol{y})_i\,] > 1/2 - 2^{-(c+1)}$$

random k×d Boolean matrix A with *i.i.d.* entries ∈ Bernoulli(p);

computation on GF(2): $(A\boldsymbol{x})_i = \left(\sum_{j=1}^{d} A_{ij} x_i\right) \bmod 2$

for suitable p and s:

$$\boxed{\begin{array}{lll} \forall \boldsymbol{x},\boldsymbol{y}\in\{0,1\}^d: & \mathrm{dist}(\boldsymbol{x},\boldsymbol{y})\le r & \Rightarrow\ \Pr[\,\mathrm{dist}(A\boldsymbol{x},A\boldsymbol{y})>s] < \mathrm{e}^{-\Omega(k)} \\ & \mathrm{dist}(\boldsymbol{x},\boldsymbol{y})>c\cdot r & \Rightarrow\ \Pr[\,\mathrm{dist}(A\boldsymbol{x},A\boldsymbol{y})\le s] < \mathrm{e}^{-\Omega(k)} \end{array}}$$

choose p to satisfy $(1\text{-}2p) = 2^{-1/r}$

$$\mathrm{dist}(\boldsymbol{x},\boldsymbol{y})\le r \quad\Rightarrow\quad \Pr[\,(A\boldsymbol{x})_i \neq (A\boldsymbol{y})_i\,] \le 1/4$$

$$\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) > c\cdot r \quad\Rightarrow\quad \Pr[\,(A\boldsymbol{x})_i \neq (A\boldsymbol{y})_i\,] > 1/2 - 2^{-(c+1)}$$

$$\mathrm{dist}(A\boldsymbol{x},A\boldsymbol{y}) = X = \sum_{i=1}^{k} X_i \qquad \text{where}\quad X_i = \begin{cases} 1 & \text{if } (A\boldsymbol{x})_i \neq (A\boldsymbol{y})_i \\ 0 & \text{otherwise} \end{cases}$$

independent trials

random $k\times d$ Boolean matrix A with *i.i.d.* entries $\in$ Bernoulli(p);

computation on GF(2): $$(A\boldsymbol{x})_i = \left(\sum_{j=1}^{d} A_{ij}x_i\right) \bmod 2$$

for suitable p and s:

$$\forall \boldsymbol{x},\boldsymbol{y} \in \{0,1\}^d: \quad \begin{array}{ll} \mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) \le r & \Rightarrow \Pr[\ \mathrm{dist}(A\boldsymbol{x}, A\boldsymbol{y}) > s] < e^{-\Omega(k)} \\ \mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) > c\cdot r & \Rightarrow \Pr[\ \mathrm{dist}(A\boldsymbol{x}, A\boldsymbol{y}) \le s] < e^{-\Omega(k)} \end{array}$$

choose p to satisfy $(1\text{-}2p) = 2^{-1/r}$

$$\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) \le r \quad \Rightarrow \quad \Pr[\ X_i = 1\ ] \le 1/4 \qquad \Rightarrow \quad \mathbf{E}[X] \le k/4$$

$$\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) > c\cdot r \quad \Rightarrow \quad \Pr[\ X_i = 1\ ] > 1/2 - 2^{-(c+1)} \quad \Rightarrow \quad \mathbf{E}[X] \le (1/2 - 2^{-(c+1)})k$$

$$\mathrm{dist}(A\boldsymbol{x}, A\boldsymbol{y}) = X = \sum_{i=1}^{k} X_i \qquad \text{where} \quad X_i = \begin{cases} 1 & \text{if } (A\boldsymbol{x})_i \ne (A\boldsymbol{y})_i \\ 0 & \text{otherwise} \end{cases}$$

choose $s = (1/4 + 1/2 - 2^{-(c+1)})k/2 = (3/8 - 2^{-(c+2)})k$

$$\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) \le r \quad \Rightarrow \quad \Pr[\ \mathrm{dist}(A\boldsymbol{x}, A\boldsymbol{y}) > s] \le \Pr[\ X > \mathbf{E}X + (1/8 - 2^{-(c+2)})k\ ]$$

$$\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) > c\cdot r \quad \Rightarrow \quad \Pr[\ \mathrm{dist}(A\boldsymbol{x}, A\boldsymbol{y}) \le s] \le \Pr[\ X < \mathbf{E}X - (1/8 - 2^{-(c+2)})k\ ]$$

Chernoff-Hoeffding Bound

Chernoff Bound:

For $X = \sum_{i=1}^{n} X_i$, where $X_1, \ldots, X_n \in \{0,1\}$ are *independent* (or *negatively associated*),

for any $t > 0$:

$$\Pr\left[X \geq \mathbb{E}[X] + t\right] \leq \exp\left(-\frac{2t^2}{n}\right)$$

$$\Pr\left[X \leq \mathbb{E}[X] - t\right] \leq \exp\left(-\frac{2t^2}{n}\right)$$

random $k \times d$ Boolean matrix A with *i.i.d.* entries $\in$ Bernoulli(p);

computation on GF(2): $$(A\boldsymbol{x})_i = \left(\sum_{j=1}^{d} A_{ij} x_i\right) \bmod 2$$

for suitable p and s:

$\forall \boldsymbol{x},\boldsymbol{y} \in \{0,1\}^d$: $\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) \le r \Rightarrow \Pr[\, \mathrm{dist}(A\boldsymbol{x}, A\boldsymbol{y}) > s] < e^{-\Omega(k)}$

$\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) > c\cdot r \Rightarrow \Pr[\, \mathrm{dist}(A\boldsymbol{x}, A\boldsymbol{y}) \le s] < e^{-\Omega(k)}$

choose p to satisfy $(1\text{-}2p) = 2^{-1/r}$

$\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) \le r \Rightarrow \Pr[\, X_i = 1\,] \le 1/4 \Rightarrow \mathbf{E}[X] \le k/4$

$\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) > c\cdot r \Rightarrow \Pr[\, X_i = 1\,] > 1/2 - 2^{-(c+1)} \Rightarrow \mathbf{E}[X] \le (1/2 - 2^{-(c+1)})k$

$$\mathrm{dist}(A\boldsymbol{x}, A\boldsymbol{y}) = X = \sum_{i=1}^{k} X_i$$ where $$X_i = \begin{cases} 1 & \text{if } (A\boldsymbol{x})_i \neq (A\boldsymbol{y})_i \\ 0 & \text{otherwise} \end{cases}$$

choose $s = (1/4 + 1/2 - 2^{-(c+1)})k/2 = (3/8 - 2^{-(c+2)})k$

$\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) \le r \Rightarrow \Pr[\, \mathrm{dist}(A\boldsymbol{x}, A\boldsymbol{y}) > s] \le \Pr[\, X > \mathbf{E}X + (1/8\text{-}2^{-(c+2)})k\,] < \exp(-2(1/8\text{-}2^{-(c+2)})^2 k)$

$\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) > c\cdot r \Rightarrow \Pr[\, \mathrm{dist}(A\boldsymbol{x}, A\boldsymbol{y}) \le s] \le \Pr[\, X < \mathbf{E}X - (1/8\text{-}2^{-(c+2)})k\,] < \exp(-2(1/8\text{-}2^{-(c+2)})^2 k)$

- Hamming space $\{0,1\}^d$:

Data: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in \{0,1\}^d$

Query: a point $\boldsymbol{x} \in \{0,1\}^d$

- High dimension: $d \gg \log n$

Dimension Reduction:

Let $k = \frac{\ln n}{(1/8-2^{-(c+2)})^2}$, $p = \frac{1}{2} - 2^{-1-1/r}$, $s = (\frac{3}{8} - 2^{-(c+2)})k$

sample $k \times d$ Boolean matrix A with *i.i.d.* entries $\in$ Bernoulli(p);

for $i = 1,2,\ldots,n$: let $\boldsymbol{z}_i = A\boldsymbol{y}_i \in \{0,1\}^k$ on finite field $\mathrm{GF}(2)$;

store all s-balls $B_s(\boldsymbol{u}) = \{\boldsymbol{y}_i \mid \mathrm{dist}(\boldsymbol{u}, \boldsymbol{z}_i) \leq s\}$ for all $\boldsymbol{u} \in \{0,1\}^k$;

To answer **query** $\boldsymbol{x} \in \{0,1\}^d$: retrieve $B_s(A\boldsymbol{x})$;

if $B_s(A\boldsymbol{x}) = \emptyset$ return "**no**"

else return any $\boldsymbol{y}_i \in B_s(A\boldsymbol{x})$

space: $n^{\mathrm{O}(1)}$ query time: $\mathrm{O}(d \ln n)$ solve (c,r)-ANN w.h.p.

Locality Sensitive Hashing (LSH)

Locality-Sensitive Hashing (LSH)

(Indyk-Motwani 1998)

- Metric space (X, dist):

Data: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in X$

Query: a point $\boldsymbol{x} \in X$

(c, r)-ANN (*Approximate Near Neighbor*):

return a $\boldsymbol{y}_i$ that $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) \leq c \cdot r$ if $\exists \boldsymbol{y}_j$ s.t. $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_j) \leq r$

"no" if $\forall \boldsymbol{y}_i$, $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) > c \cdot r$

arbitrary if otherwise

Locality-sensitive hashing (LSH):

A random $h : X \to U$ is an (r, cr, p, q)-LSH if $\forall x, y \in X$:

$$\mathrm{dist}(x, y) \leq r \implies \Pr[h(x) = h(y)] \geq p$$

$$\mathrm{dist}(x, y) > c \cdot r \implies \Pr[h(x) = h(y)] \leq q$$

Locality-Sensitive Hashing (LSH)

- Metric space (X, dist):

Locality-sensitive hashing (LSH):

A random $h : X \to U$ is an (r, cr, p, q)-**LSH** if $\forall x, y \in X$:

$$\mathrm{dist}(x, y) \le r \implies \Pr[h(x) = h(y)] \ge p$$

$$\mathrm{dist}(x, y) > c \cdot r \implies \Pr[h(x) = h(y)] \le q$$

Proposition (*tensorization*):

$\exists$ an (r, cr, p, q)-LSH $h : X \to U$ $\implies$ $\exists$ an $\left(r, cr, p^k, q^k\right)$-LSH $h : X \to U^k$

draw independent $h_1, \ldots, h_k$ according to distribution of h

$$g(x) = (h_1(x), h_2(x), \ldots, h_k(x)) \in U^k$$

- Metric space (X, dist):

Data: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in X$

Query: a point $\boldsymbol{x} \in X$

return a $\boldsymbol{y}_i$ that $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) \leq c \cdot r$ if $\exists \boldsymbol{y}_j$ s.t. $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_j) \leq r$

"no" if $\forall \boldsymbol{y}_i$, $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) > c \cdot r$

suppose we have $(r, cr, p^*, 1/n)$-LSH g: $X \longrightarrow U$

$\forall \boldsymbol{x}, \boldsymbol{y} \in X$: $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}) \leq r \quad \Rightarrow \Pr[\ g(\boldsymbol{x}) = g(\boldsymbol{y})\ ] \geq p^*$

$\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}) > c \cdot r \quad \Rightarrow \Pr[\ g(\boldsymbol{x}) = g(\boldsymbol{y})\ ] \leq 1/n$

Data structure: a dictionary for all key-value pairs $\langle g(\boldsymbol{y}_i), y_i \rangle$;

Upon **query** $\boldsymbol{x} \in X$: find all $\boldsymbol{y}_i$ with $g(\boldsymbol{x}) = g(\boldsymbol{y}_i)$;

if encounter a $\boldsymbol{y}_i$ that $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) \leq c \cdot r$ return this $\boldsymbol{y}_i$;

else return "no";

if real answer is "no": always correct

if real answer is not "no": correct with probability $\geq p^*$

(Use FKS) space: $O(n)$ time: $O(1) + O(1)$ in expectation

- Metric space (X, dist):

Data: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in X$

Query: a point $\boldsymbol{x} \in X$

return a $\boldsymbol{y}_i$ that $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) \leq c \cdot r$ if $\exists \boldsymbol{y}_j$ s.t. $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_j) \leq r$

"no" if $\forall \boldsymbol{y}_i$, $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) > c \cdot r$

suppose we have $(r, cr, p^*, 1/n)$-LSH $g: X \to U$

Hash functions: *i.i.d.* instances $g_1, \ldots, g_\ell$ of g, where $\ell = 1/p^*$;

Data structure: ℓ dictionaries, where the jth dictionary stores all key-value pairs $\langle g_j(\boldsymbol{y}_i), y_i \rangle$;

Upon **query** $\boldsymbol{x} \in X$:

find $\leq 10\ell$ such $\boldsymbol{y}_i$ that $\exists j, g_j(\boldsymbol{x}) = g_j(\boldsymbol{y}_i)$;

if encounter a $\boldsymbol{y}_i$ that $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) \leq c \cdot r$ then return this $\boldsymbol{y}_i$;
else return "no";

metric space (X, dist) $(r, cr, p^*, 1/n)$-LSH $g: X \to U$

Data: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in X$ **Query**: $\boldsymbol{x} \in X$

Hash functions: *i.i.d.* instances $g_1, \ldots, g_\ell$ of g, where $\ell = 1/p^*$;

Data structure: ℓ dictionaries, where the jth dictionary stores
all key-value pairs $\langle g_j(\boldsymbol{y}_i), y_i \rangle$;

Upon **query** $\boldsymbol{x} \in X$:

find $\leq 10\ell$ such $\boldsymbol{y}_i$ that $\exists j, g_j(\boldsymbol{x}) = g_j(\boldsymbol{y}_i)$;

if encounter a $\boldsymbol{y}_i$ that $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) \leq c \cdot r$ then return this $\boldsymbol{y}_i$;
else return “no”;

space: $O(nl) = O(n/p^*)$ time: $O(l) = O(1/p^*)$ (use FKS)

if real answer is “no” : $\forall\, \boldsymbol{y}_i$, $\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}_i) > c{\cdot}r$ ⇨ always correct

if $\exists \boldsymbol{y}_s$ s.t. $\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}_s) \leq r$

$\Pr[$ answer “no” $] \leq ?$

metric space (X, dist) $(r, cr, p^*, 1/n)$-LSH $g: X \to U$

Data: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in X$ **Query**: $\boldsymbol{x} \in X$

Hash functions: *i.i.d.* instances $g_1, \ldots, g_\ell$ of g, where $\ell = 1/p^*$;

Data structure: ℓ dictionaries, where the jth dictionary stores all key-value pairs $\langle g_j(\boldsymbol{y}_i), y_i \rangle$;

Upon **query** $\boldsymbol{x} \in X$:

find $\le 10\ell$ such $\boldsymbol{y}_i$ that $\exists j, g_j(\boldsymbol{x}) = g_j(\boldsymbol{y}_i)$;

if encounter a $\boldsymbol{y}_i$ that $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) \le c \cdot r$ then return this $\boldsymbol{y}_i$;

else return “no”;

if $\exists \boldsymbol{y}_s$ s.t. $\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}_s) \le r$

Pr[answer “no”] $\le \boxed{\Pr[\forall j, g_j(\boldsymbol{x}) \neq g_j(\boldsymbol{y}_s)]} \le (1-p^*)^\ell \le 1/e$

$+ \underline{\Pr[>10l \text{ bad } \boldsymbol{y}_i \text{ that } \mathrm{dist}(\boldsymbol{x},\boldsymbol{y}_i) > c{\cdot}r \text{ but } \exists j \text{ s.t. } g_j(\boldsymbol{x})=g_j(\boldsymbol{y}_i)]}$

Markov inequality $\le \mathbf{E}[\ \#\text{ of such bad } \boldsymbol{y}_i] \,/\, 10l \le \frac{\ell \cdot n \cdot (1/n)}{10\ell} \le 0.1$

linearity of expectation

metric space (X, dist) $(r, cr, p^*, 1/n)$-LSH $g: X \to U$

Data: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in X$ **Query**: $\boldsymbol{x} \in X$

Hash functions: *i.i.d.* instances $g_1, \ldots, g_\ell$ of g, where $\ell = 1/p^*$;

Data structure: ℓ dictionaries, where the jth dictionary stores
all key-value pairs $\langle g_j(\boldsymbol{y}_i), y_i \rangle$;

Upon **query** $\boldsymbol{x} \in X$:
find $\leq 10\ell$ such $\boldsymbol{y}_i$ that $\exists j, g_j(\boldsymbol{x}) = g_j(\boldsymbol{y}_i)$;
if encounter a $\boldsymbol{y}_i$ that $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) \leq c \cdot r$ then return this $\boldsymbol{y}_i$;
else return "no";

space: $O(nl) = O(n/p^*)$ time: $O(l) = O(1/p^*)$

if real answer is "no" : $\forall\, \boldsymbol{y}_i$, $\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}_i) > c{\cdot}r$ ⇒ always correct

if $\exists \boldsymbol{y}_s$ s.t. $\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}_s) \leq r$

⇒ Pr[answer "no"] $\leq 1/e + 0.1 < 0.5$

- (c, r)-**ANN** in metric space (X, dist):

Data: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in X$

Query: a point $\boldsymbol{x} \in X$

return a $\boldsymbol{y}_i$ that $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) \leq c \cdot r$ if $\exists \boldsymbol{y}_j$ s.t. $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_j) \leq r$

"no" if $\forall \boldsymbol{y}_i$, $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) > c \cdot r$

suppose we have (r, cr, p, q)-LSH h: $X \rightarrow U$

we have $(r, cr, p^k, 1/n)$-LSH g: $X \rightarrow U^k$

for $k = \log_{(1/q)} n$ so $p^k = p^{\log_{1/q} n} = n^{-\rho}$

where $$\boxed{\rho = \frac{\log p}{\log q}}$$

solve (c, r)-**ANN** with space $O(n^{1+\rho})$

query time $\mathrm{O}(n^{\varrho} \cdot \log n)$ and one-sided error <0.5

- (c, r)-**ANN** in metric space (X, dist):

Data: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in X$

Query: a point $\boldsymbol{x} \in X$

return a $\boldsymbol{y}_i$ that $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) \le c \cdot r$ if $\exists \boldsymbol{y}_j$ s.t. $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_j) \le r$

"no" if $\forall \boldsymbol{y}_i$, $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) > c \cdot r$

suppose we have (r, cr, p, q)-LSH h: $X \rightarrow U$

$$\boxed{\rho = \frac{\log p}{\log q}}$$

⇨ solve (c, r)-**ANN** with space $O(n^{1+\rho})$

query time $O(n^{\rho} \cdot \log n)$ and one-sided error <0.5

- (c, r)-**ANN** in Hamming space $\{0,1\}^d$:

Data: n points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \in \{0,1\}^d$

Query: a point $\boldsymbol{x} \in \{0,1\}^d$

return a $\boldsymbol{y}_i$ that $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) \leq c \cdot r$ if $\exists \boldsymbol{y}_j$ s.t. $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_j) \leq r$

"no" if $\forall \boldsymbol{y}_i$, $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}_i) > c \cdot r$

$\forall\, \boldsymbol{x} \in \{0, 1\}^d$: $h(\boldsymbol{x}) = x_i$ for uniform random $i \in [d]$

$\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) \leq r \quad \Rightarrow \Pr[\, h(\boldsymbol{x}) = h(\boldsymbol{y})\,] \geq 1\text{-}r/d$

$\mathrm{dist}(\boldsymbol{x},\boldsymbol{y}) > c{\cdot}r \quad \Rightarrow \Pr[\, h(\boldsymbol{x}) = h(\boldsymbol{y})\,] \leq 1\text{-}cr/d$

h: $\{0, 1\}^d \longrightarrow \{0,1\}$ is an $(r, cr, 1\text{-}r/d, 1\text{-}cr/d)$-LSH

$$\boxed{\rho = \frac{\log(1 - r/d)}{\log(1 - cr/d)} \leq \frac{1}{c}}$$

⇨ solve (c, r)-**ANN** in Hamming space with space $O(n^{1+1/c})$ query time $O(n^{1/c}{\cdot}\log n)$ and one-sided error <0.5

COMPUTER SCIENCE

A neural algorithm for a fundamental computing problem

Sanjoy Dasgupta,[1] Charles F. Stevens,[2,3] Saket Navlakha[4]*

Similarity search—for example, identifying similar images in a database or similar documents on the web—is a fundamental computing problem faced by large-scale information retrieval systems. We discovered that the fruit fly olfactory circuit solves this problem with a variant of a computer science algorithm (called locality-sensitive hashing). The fly circuit assigns similar neural activity patterns to similar odors, so that behaviors learned from one odor can be applied when a similar odor is experienced. The fly algorithm, however, uses three computational strategies that depart from traditional approaches. These strategies can be translated to improve the performance of computational similarity searches. This perspective helps illuminate the logic supporting an important sensory function and provides a conceptually new algorithm for solving a fundamental computational problem.

An essential task of many neural circuits is to generate neural activity patterns in response to input stimuli, so that different inputs can be specifically identified. We studied the circuit used to process odors in the fruit fly olfactory system and uncovered computational strategies for solving a fundamental machine learning problem: approximate similarity (or nearest-neighbors) search.

The fly olfactory circuit generates a "tag" for each odor, which is a set of neurons that fire when that odor is presented (*1*). This tag is critical for learning behavioral responses to different odors (*2*). For example, if a reward (e.g., sugar water) or a punishment (e.g., electric shock) is associated with an odor, that odor becomes attractive (a fly pendence is removed (*7*, *8*); that is, the distribution of firing rates across the 50 PN types is exponential, with close to the same mean for all odors and all odor concentrations (*1*). Thus, the first step in the circuit essentially "centers the mean"—a standard preprocessing step in many computational pipelines—using a technique called divisive normalization (*8*). This step is important so that the fly does not mix up odor intensity with odor type.

The second step, where the main algorithmic insight begins, involves a 40-fold expansion in the number of neurons: Fifty PNs project to 2000 Kenyon cells (KCs), connected by a sparse, binary random connection matrix (*9*). Each KC receives and sums the firing rates from about six randomly out of the billions of images on the web—that look most similar to your elephant image. This is called the nearest-neighbors search problem, which is of fundamental importance in information retrieval, data compression, and machine learning (*10*). Each image is typically represented as a *d*-dimensional vector of feature values. (Each odor that a fly processes is a 50-dimensional feature vector of firing rates.) A distance metric is used to compute the similarity between two images (feature vectors), and the goal is to efficiently find the nearest neighbors of any query image. If the web contained only a few images, then brute force linear search could easily be used to find the exact nearest neighbors. If the web contained many images, but each image was represented by a low-dimensional vector (e.g., 10 or 20 features), then space-partitioning methods (*12*) would similarly suffice. However, for large databases with high-dimensional data, neither approach scales (*11*).

In many applications, returning an approximate set of nearest neighbors that are "close enough" to the query is adequate, so long as they can be found quickly. This has motivated an approach for finding approximate nearest neighbors by LSH (*10*). For the fly, as noted, the locality-sensitive property states that two odors that generate similar ORN responses will be represented by two tags that are themselves similar (Fig. 1B). Likewise, for image search, the tag of an elephant image will be more similar to the tag of another elephant image than to the tag of a skyscraper image.

Unlike a traditional (non-LSH) hash function, where the input points are scattered randomly and uniformly over the range, a LSH function provides a distance-preserving embedding of points from *d*-dimensional space into *m*-dimensional

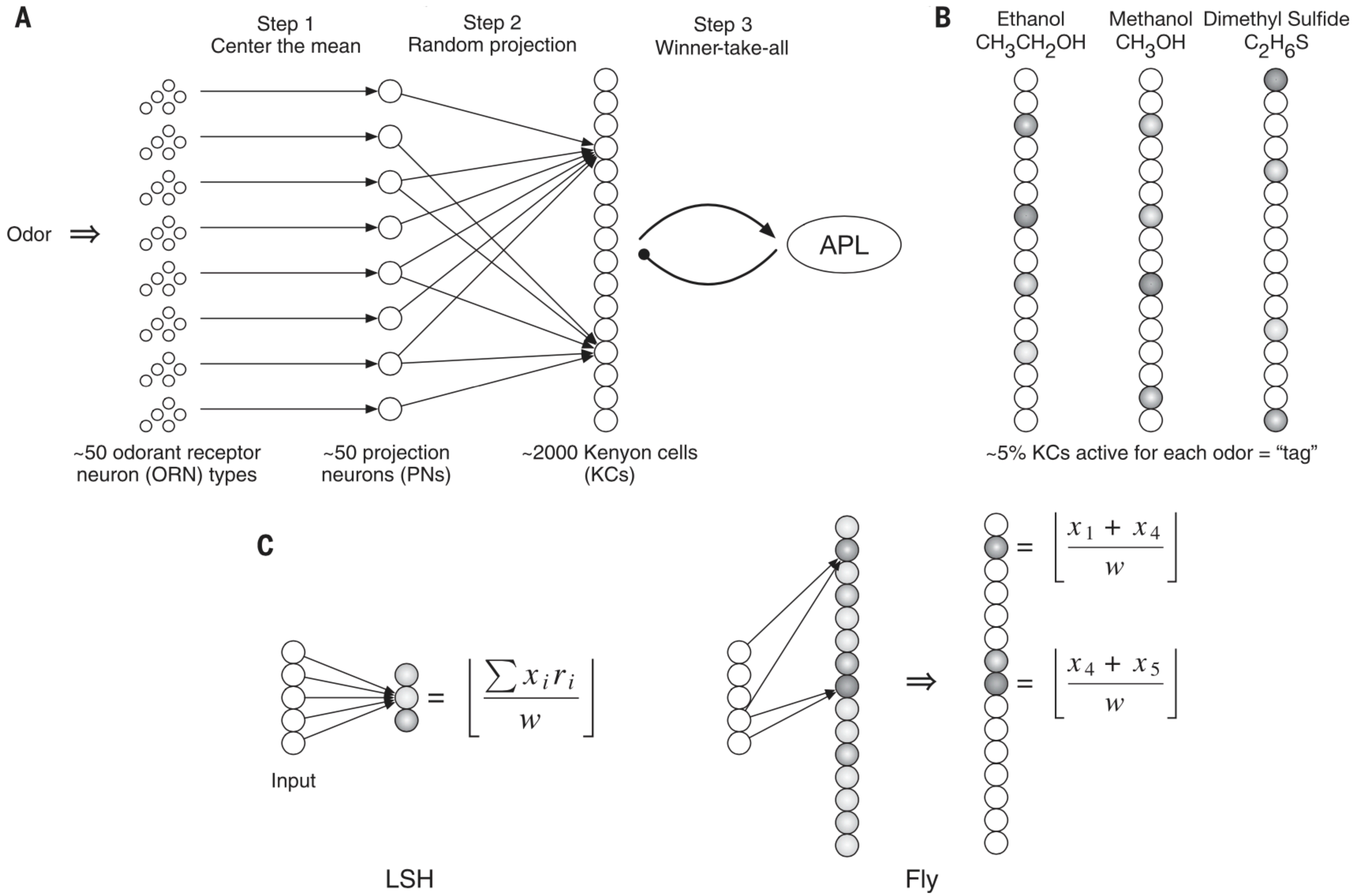

A
Step 1
Center the mean
Step 2
Random projection
Step 3
Winner-take-all
Odor ⇒
APL
~50 odorant receptor neuron (ORN) types
~50 projection neurons (PNs)
~2000 Kenyon cells (KCs)
B
Ethanol
CH_3CH_2OH
Methanol
CH_3OH
Dimethyl Sulfide
C_2H_6S
~5% KCs active for each odor = "tag"
C
Input
$= \left\lfloor \frac{\sum x_i r_i}{w} \right\rfloor$
LSH
⇒
$= \left\lfloor \frac{x_1 + x_4}{w} \right\rfloor$
$= \left\lfloor \frac{x_4 + x_5}{w} \right\rfloor$
Fly

Fig. 1. Mapping between the fly olfactory circuit and locality-sensitive hashing (LSH). (**A**) Schematic of the fly olfactory circuit. In step 1, 50 ORNs in the fly's nose send axons to 50 PNs in the glomeruli; as a result of this projection, each odor is represented by an exponential distribution of firing rates, with the same mean for all odors and all odor concentrations. In step 2, the PNs expand the dimensionality, projecting to 2000 KCs connected by a sparse, binary random projection matrix. In step 3, the KCs receive feedback inhibition from the anterior paired lateral (APL) neuron, which leaves only the top 5% of KCs to remain firing spikes for the odor. This 5% corresponds to the tag (hash) for the odor. (**B**) Illustrative odor responses. Similar pairs of odors (e.g., methanol and ethanol) are assigned more similar tags than are dissimilar odors. Darker shading denotes higher activity. (**C**) Differences between conventional LSH and the fly algorithm. In the example, the computational complexity for LSH and the fly are the same. The input dimensionality $d = 5$. LSH computes $m = 3$ random projections, each of which requires 10 operations (five multiplications plus five additions). The fly computes $m = 15$ random projections, each of which requires two addition operations. Thus, both require 30 total operations. $\mathbf{x}$, input feature vector; r, Gaussian random variable; w, bin width constant for discretization.

Fast Johnson-Lindenstrauss Transformation (FJLT)

Subspace Embedding