Today.

Cuckoo hashing.

Today.

Cuckoo hashing.

Johnson-Lindenstrass.

Cuckoo hashing.

Hashing with two choices: max load $O(\log\log n)$.

Cuckoo hashing.

Hashing with two choices: max load $O(\log \log n)$.

Cuckoo hashing:

Cuckoo hashing.

Hashing with two choices: max load $O(\log \log n)$.

Cuckoo hashing:
Array.

Cuckoo hashing.

Hashing with two choices: max load $O(\log \log n)$.

Cuckoo hashing:
Array. Two hash functions h_1, h_2.

Cuckoo hashing.

Hashing with two choices: max load $O(\log \log n)$.

Cuckoo hashing:
Array. Two hash functions h_1, h_2.

Insert x: place in $h_1(x)$ or $h_2(x)$ if space.

Cuckoo hashing.

Hashing with two choices: max load $O(\log\log n)$.

Cuckoo hashing:
Array. Two hash functions h_1, h_2.

Insert x: place in $h_1(x)$ or $h_2(x)$ if space.
Else bump elt y in $h_i(x)$ u.a.r. for $i \in [1,2]$.

Cuckoo hashing.

Hashing with two choices: max load $O(\log\log n)$.

Cuckoo hashing:
Array. Two hash functions h_1, h_2.

Insert x: place in $h_1(x)$ or $h_2(x)$ if space.
  Else bump elt y in $h_i(x)$ u.a.r. for $i \in [1,2]$.
Bump y,x: place y in $h_j(y)$ where $j \neq i$ if space.

Cuckoo hashing.

Hashing with two choices: max load $O(\log\log n)$.

Cuckoo hashing:
Array. Two hash functions h_1, h_2.

Insert x: place in $h_1(x)$ or $h_2(x)$ if space.
  Else bump elt y in $h_i(x)$ u.a.r. for $i \in [1,2]$.
Bump y, x: place y in $h_j(y)$ where $j \neq i$ if space.
  Else bump y' in $h_j(y)$.

Cuckoo hashing.

Hashing with two choices: max load $O(\log\log n)$.

Cuckoo hashing:
Array. Two hash functions h_1, h_2.

Insert x: place in $h_1(x)$ or $h_2(x)$ if space.
  Else bump elt y in $h_i(x)$ u.a.r. for $i \in [1,2]$.
Bump y, x: place y in $h_j(y)$ where $j \neq i$ if space.
  Else bump y' in $h_j(y)$. And so on.

Cuckoo hashing.

Hashing with two choices: max load $O(\log\log n)$.

Cuckoo hashing:
Array. Two hash functions h_1, h_2.

Insert x: place in $h_1(x)$ or $h_2(x)$ if space.
  Else bump elt y in $h_i(x)$ u.a.r. for $i \in [1,2]$.
Bump y,x: place y in $h_j(y)$ where $j \neq i$ if space.
  Else bump y' in $h_j(y)$. And so on.

Cuckoo hashing.

Hashing with two choices: max load $O(\log\log n)$.

Cuckoo hashing:
Array. Two hash functions h_1, h_2.

Insert x: place in $h_1(x)$ or $h_2(x)$ if space.
Else bump elt y in $h_i(x)$ u.a.r. for $i \in [1,2]$.
Bump y, x: place y in $h_j(y)$ where $j \neq i$ if space.
Else bump y' in $h_j(y)$. And so on.

If go too long.

Cuckoo hashing.

Hashing with two choices: max load $O(\log\log n)$.

Cuckoo hashing:
Array. Two hash functions h_1, h_2.

Insert x: place in $h_1(x)$ or $h_2(x)$ if space.
  Else bump elt y in $h_i(x)$ u.a.r. for $i \in [1,2]$.
Bump y, x: place y in $h_j(y)$ where $j \neq i$ if space.
  Else bump y' in $h_j(y)$. And so on.

If go too long. Fail.

Cuckoo hashing.

Hashing with two choices: max load $O(\log\log n)$.

Cuckoo hashing:
Array. Two hash functions h_1, h_2.

Insert x: place in $h_1(x)$ or $h_2(x)$ if space.
Else bump elt y in $h_i(x)$ u.a.r. for $i \in [1,2]$.
Bump y, x: place y in $h_j(y)$ where $j \neq i$ if space.
Else bump y' in $h_j(y)$. And so on.

If go too long. Fail. Rehash entire hash table.

Cuckoo hashing.

Hashing with two choices: max load $O(\log \log n)$.

Cuckoo hashing:
Array. Two hash functions h_1, h_2.

Insert x: place in $h_1(x)$ or $h_2(x)$ if space.
Else bump elt y in $h_i(x)$ u.a.r. for $i \in [1,2]$.
Bump y,x: place y in $h_j(y)$ where $j \neq i$ if space.
Else bump y' in $h_j(y)$. And so on.

If go too long. Fail. Rehash entire hash table.
Fails if cycle for insert.

Cuckoo hashing.

Hashing with two choices: max load $O(\log\log n)$.

Cuckoo hashing:
Array. Two hash functions h_1, h_2.

Insert x: place in $h_1(x)$ or $h_2(x)$ if space.
Else bump elt y in $h_i(x)$ u.a.r. for $i \in [1,2]$.
Bump y, x: place y in $h_j(y)$ where $j \neq i$ if space.
Else bump y' in $h_j(y)$. And so on.

If go too long. Fail. Rehash entire hash table.
Fails if cycle for insert.

C_ℓ - event of cycle of length ℓ at a vertex.

Cuckoo hashing.

Hashing with two choices: max load $O(\log\log n)$.

Cuckoo hashing:
Array. Two hash functions h_1, h_2.

Insert x: place in $h_1(x)$ or $h_2(x)$ if space.
Else bump elt y in $h_i(x)$ u.a.r. for $i \in [1,2]$.
Bump y, x: place y in $h_j(y)$ where $j \neq i$ if space.
Else bump y' in $h_j(y)$. And so on.

If go too long. Fail. Rehash entire hash table.
Fails if cycle for insert.

C_ℓ - event of cycle of length ℓ at a vertex.

$$\Pr[C_\ell] \leq \binom{m}{\ell}\binom{n}{\ell}\left(\frac{\ell}{n}\right)^{2(\ell)} \leq \left(\frac{e^2}{8}\right)^{\ell} \tag{1}$$

Cuckoo hashing.

Hashing with two choices: max load $O(\log\log n)$.

Cuckoo hashing:
Array. Two hash functions h_1, h_2.

Insert x: place in $h_1(x)$ or $h_2(x)$ if space.
Else bump elt y in $h_i(x)$ u.a.r. for $i \in [1,2]$.
Bump y, x: place y in $h_j(y)$ where $j \neq i$ if space.
Else bump y' in $h_j(y)$. And so on.

If go too long. Fail. Rehash entire hash table.
Fails if cycle for insert.

C_ℓ - event of cycle of length ℓ at a vertex.

$$\Pr[C_\ell] \leq \binom{m}{\ell}\binom{n}{\ell}\left(\frac{\ell}{n}\right)^{2(\ell)} \leq \left(\frac{e^2}{8}\right)^{\ell} \qquad (1)$$

Probability that an insert hits a cycle of length $\ell \leq \frac{\ell}{n}\left(\frac{e^2}{8}\right)^{\ell}$

Cuckoo hashing.

Hashing with two choices: max load $O(\log\log n)$.

Cuckoo hashing:
Array. Two hash functions h_1, h_2.

Insert x: place in $h_1(x)$ or $h_2(x)$ if space.
Else bump elt y in $h_i(x)$ u.a.r. for $i \in [1,2]$.
Bump y, x: place y in $h_j(y)$ where $j \neq i$ if space.
Else bump y' in $h_j(y)$. And so on.

If go too long. Fail. Rehash entire hash table.
Fails if cycle for insert.

C_ℓ - event of cycle of length ℓ at a vertex.

$$\Pr[C_\ell] \leq \binom{m}{\ell}\binom{n}{\ell}\left(\frac{\ell}{n}\right)^{2(\ell)} \leq \left(\frac{e^2}{8}\right)^{\ell} \qquad (1)$$

Probability that an insert hits a cycle of length $\ell \leq \frac{\ell}{n}\left(\frac{e^2}{8}\right)^{\ell}$

Rehash every $\Omega(n)$ inserts (if $\leq n/8$ items in table.)

Cuckoo hashing.

Hashing with two choices: max load $O(\log\log n)$.

Cuckoo hashing:
Array. Two hash functions h_1, h_2.

Insert x: place in $h_1(x)$ or $h_2(x)$ if space.
Else bump elt y in $h_i(x)$ u.a.r. for $i \in [1,2]$.
Bump y, x: place y in $h_j(y)$ where $j \neq i$ if space.
Else bump y' in $h_j(y)$. And so on.

If go too long. Fail. Rehash entire hash table.
Fails if cycle for insert.

C_ℓ - event of cycle of length ℓ at a vertex.

$$\Pr[C_\ell] \leq \binom{m}{\ell}\binom{n}{\ell}\left(\frac{\ell}{n}\right)^{2(\ell)} \leq \left(\frac{e^2}{8}\right)^{\ell} \quad (1)$$

Probability that an insert hits a cycle of length $\ell \leq \frac{\ell}{n}\left(\frac{e^2}{8}\right)^{\ell}$

Rehash every $\Omega(n)$ inserts (if $\leq n/8$ items in table.)
$O(1)$ time on average.

Johnson-Lindenstrass

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

Johnson-Lindenstrass

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

Random $k = \frac{c \log n}{\varepsilon^2}$ dimensional subspace.

Johnson-Lindenstrass

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

Random $k = \frac{c \log n}{\varepsilon^2}$ dimensional subspace.

Claim: with probability $1 - \frac{1}{n^{c-2}}$,

$$(1-\varepsilon)\sqrt{\frac{k}{d}}|x_i - x_j| \le |y_i - y_j| \le (1+\varepsilon)\sqrt{\frac{k}{d}}|x_i - x_j|$$

Johnson-Lindenstrass

Points: $x_1, \dots, x_n \in \mathbb{R}^d$.

Random $k = \frac{c \log n}{\varepsilon^2}$ dimensional subspace.

Claim: with probability $1 - \frac{1}{n^{c-2}}$,

$$(1-\varepsilon)\sqrt{\frac{k}{d}}|x_i - x_j| \le |y_i - y_j| \le (1+\varepsilon)\sqrt{\frac{k}{d}}|x_i - x_j|$$

"Projecting and scaling by $\sqrt{\frac{d}{k}}$ preserves all pairwise distances w/in factor of $1 \pm \varepsilon$."

Random subspace.

Method 1:

Random subspace.

Method 1:
Pick unit v_1

Random subspace.

Method 1:
Pick unit v_1 ,

Random subspace.

Method 1:
Pick unit v_1 ,
v_2 orthogonal to v_1,

Random subspace.

Method 1:
Pick unit v_1 ,
v_2 orthogonal to v_1,
$\ldots$

Random subspace.

Method 1:

Pick unit v_1 ,

v_2 orthogonal to v_1,

$\ldots$

v_k orthogonal to previous vectors...

Random subspace.

Method 1:
Pick unit v_1 ,
v_2 orthogonal to v_1,
$\ldots$
v_k orthogonal to previous vectors...

Method 2:

Random subspace.

Method 1:
Pick unit v_1 ,
v_2 orthogonal to v_1,
$\ldots$
v_k orthogonal to previous vectors...

Method 2:
Choose k vectors $v_1, \ldots, v_k$

Random subspace.

Method 1:
Pick unit v_1 ,
v_2 orthogonal to v_1,
$\ldots$
v_k orthogonal to previous vectors...

Method 2:
Choose k vectors $v_1, \ldots, v_k$
Gram Schmidt orthonormalization of $k \times d$ matrix where rows are v_i.

Random subspace.

Method 1:
Pick unit v_1 ,
v_2 orthogonal to v_1,
...
v_k orthogonal to previous vectors...

Method 2:
Choose k vectors $v_1, \ldots, v_k$
Gram Schmidt orthonormalization of $k \times d$ matrix where rows are v_i.
remove projection onto previous subspace.

Projections.

Project x into subspace spanned by $v_1, v_2, \cdots, v_k$.

Projections.

Project x into subspace spanned by $v_1, v_2, \cdots, v_k$.

$y_1 = x \cdot v_1, y_2 = x \cdot, v_2, \cdots, y_k = x \cdot v_k$

Projections.

Project x into subspace spanned by $v_1, v_2, \cdots, v_k$.

$y_1 = x \cdot v_1, y_2 = x \cdot, v_2, \cdots, y_k = x \cdot v_k$

Projection: $(y_1, \ldots, y_k)$.

Projections.

Project x into subspace spanned by $v_1, v_2, \cdots, v_k$.

$y_1 = x \cdot v_1, y_2 = x \cdot, v_2, \cdots, y_k = x \cdot v_k$

Projection: $(y_1, \ldots, y_k)$.

Have: Arbitrary vector, random k-dimensional subspace.

Projections.

Project x into subspace spanned by $v_1, v_2, \cdots, v_k$.

$y_1 = x \cdot v_1, y_2 = x \cdot, v_2, \cdots, y_k = x \cdot v_k$

Projection: $(y_1, \ldots, y_k)$.

Have: Arbitrary vector, random k-dimensional subspace.

View As: Random vector, standard basis for k dimensions.

Projections.

Project x into subspace spanned by $v_1, v_2, \cdots, v_k$.

$y_1 = x \cdot v_1, y_2 = x \cdot, v_2, \cdots, y_k = x \cdot v_k$

Projection: $(y_1, \ldots, y_k)$.

Have: Arbitrary vector, random k-dimensional subspace.

View As: Random vector, standard basis for k dimensions.

Orthogonal U - rotates $v_1, \ldots, v_k$ onto $e_1, \ldots, e_k$

Projections.

Project x into subspace spanned by $v_1, v_2, \cdots, v_k$.

$y_1 = x \cdot v_1, y_2 = x \cdot, v_2, \cdots, y_k = x \cdot v_k$

Projection: $(y_1, \ldots, y_k)$.

Have: Arbitrary vector, random k-dimensional subspace.

View As: Random vector, standard basis for k dimensions.

Orthogonal U - rotates $v_1, \ldots, v_k$ onto $e_1, \ldots, e_k$

y_i

Projections.

Project x into subspace spanned by $v_1, v_2, \cdots, v_k$.

$y_1 = x \cdot v_1, y_2 = x \cdot, v_2, \cdots, y_k = x \cdot v_k$

Projection: $(y_1, \ldots, y_k)$.

Have: Arbitrary vector, random k-dimensional subspace.

View As: Random vector, standard basis for k dimensions.

Orthogonal U - rotates $v_1, \ldots, v_k$ onto $e_1, \ldots, e_k$

$y_i = \langle v_i | x \rangle$

Projections.

Project x into subspace spanned by $v_1, v_2, \cdots, v_k$.

$y_1 = x \cdot v_1, y_2 = x \cdot, v_2, \cdots, y_k = x \cdot v_k$

Projection: $(y_1, \ldots, y_k)$.

Have: Arbitrary vector, random k-dimensional subspace.

View As: Random vector, standard basis for k dimensions.

Orthogonal U - rotates $v_1, \ldots, v_k$ onto $e_1, \ldots, e_k$

$y_i = \langle v_i | x \rangle = \langle U v_i | U x \rangle$

Projections.

Project x into subspace spanned by $v_1, v_2, \cdots, v_k$.

$y_1 = x \cdot v_1, y_2 = x \cdot, v_2, \cdots, y_k = x \cdot v_k$

Projection: $(y_1, \ldots, y_k)$.

Have: Arbitrary vector, random k-dimensional subspace.

View As: Random vector, standard basis for k dimensions.

Orthogonal U - rotates $v_1, \ldots, v_k$ onto $e_1, \ldots, e_k$

$y_i = \langle v_i | x \rangle = \langle U v_i | U x \rangle = \langle e_i | U x \rangle$

Projections.

Project x into subspace spanned by $v_1, v_2, \cdots, v_k$.

$y_1 = x \cdot v_1, y_2 = x \cdot, v_2, \cdots, y_k = x \cdot v_k$

Projection: $(y_1, \ldots, y_k)$.

Have: Arbitrary vector, random k-dimensional subspace.

View As: Random vector, standard basis for k dimensions.

Orthogonal U - rotates $v_1, \ldots, v_k$ onto $e_1, \ldots, e_k$

$y_i = \langle v_i | x \rangle = \langle U v_i | U x \rangle = \langle e_i | U x \rangle = \langle e_i | z \rangle$

Projections.

Project x into subspace spanned by $v_1, v_2, \cdots, v_k$.

$y_1 = x \cdot v_1, y_2 = x \cdot, v_2, \cdots, y_k = x \cdot v_k$

Projection: $(y_1, \ldots, y_k)$.

Have: Arbitrary vector, random k-dimensional subspace.

View As: Random vector, standard basis for k dimensions.

Orthogonal U - rotates $v_1, \ldots, v_k$ onto $e_1, \ldots, e_k$

$y_i = \langle v_i | x \rangle = \langle U v_i | U x \rangle = \langle e_i | U x \rangle = \langle e_i | z \rangle$

Inverse of U maps e_i to random vector v_i

Projections.

Project x into subspace spanned by $v_1, v_2, \cdots, v_k$.

$y_1 = x \cdot v_1, y_2 = x \cdot, v_2, \cdots, y_k = x \cdot v_k$

Projection: $(y_1, \ldots, y_k)$.

Have: Arbitrary vector, random k-dimensional subspace.

View As: Random vector, standard basis for k dimensions.

Orthogonal U - rotates $v_1, \ldots, v_k$ onto $e_1, \ldots, e_k$

$y_i = \langle v_i | x \rangle = \langle U v_i | U x \rangle = \langle e_i | U x \rangle = \langle e_i | z \rangle$

Inverse of U maps e_i to random vector v_i

Projections.

Project x into subspace spanned by $v_1, v_2, \cdots, v_k$.

$y_1 = x \cdot v_1, y_2 = x \cdot, v_2, \cdots, y_k = x \cdot v_k$

Projection: $(y_1, \ldots, y_k)$.

Have: Arbitrary vector, random k-dimensional subspace.

View As: Random vector, standard basis for k dimensions.

Orthogonal U - rotates $v_1, \ldots, v_k$ onto $e_1, \ldots, e_k$

$y_i = \langle v_i | x \rangle = \langle Uv_i | Ux \rangle = \langle e_i | Ux \rangle = \langle e_i | z \rangle$

Inverse of U maps e_i to random vector v_i

$z = Ux$ is uniformly distributed on d sphere for unit $x \in \mathbb{R}^d$.

Projections.

Project x into subspace spanned by $v_1, v_2, \cdots, v_k$.

$y_1 = x \cdot v_1, y_2 = x \cdot, v_2, \cdots, y_k = x \cdot v_k$

Projection: $(y_1, \ldots, y_k)$.

Have: Arbitrary vector, random k-dimensional subspace.

View As: Random vector, standard basis for k dimensions.

Orthogonal U - rotates $v_1, \ldots, v_k$ onto $e_1, \ldots, e_k$

$y_i = \langle v_i | x \rangle = \langle Uv_i | Ux \rangle = \langle e_i | Ux \rangle = \langle e_i | z \rangle$

Inverse of U maps e_i to random vector v_i

$z = Ux$ is uniformly distributed on d sphere for unit $x \in \mathbb{R}^d$.

y_i is ith coordinate of random vector z.

Expected value of y_i.

Random projection: first k coordinates of random unit vector, z_i.

Expected value of y_i.

Random projection: first k coordinates of random unit vector, z_i.

$E[\sum_{i \in [d]} z_i^2] = 1$.

Expected value of y_i.

Random projection: first k coordinates of random unit vector, z_i.

$E[\sum_{i \in [d]} z_i^2] = 1$. Linearity of Expectation.

Expected value of y_i.

Random projection: first k coordinates of random unit vector, z_i.

$E[\sum_{i \in [d]} z_i^2] = 1$. Linearity of Expectation.

By symmetry, each z_i is identically distributed.

Expected value of y_i.

Random projection: first k coordinates of random unit vector, z_i.

$E[\sum_{i \in [d]} z_i^2] = 1$. Linearity of Expectation.

By symmetry, each z_i is identically distributed.

$E[\sum_{i \in [k]} z_i^2] = \frac{k}{d}$.

Expected value of y_i.

Random projection: first k coordinates of random unit vector, z_i.

$E[\sum_{i \in [d]} z_i^2] = 1$. Linearity of Expectation.

By symmetry, each z_i is identically distributed.

$E[\sum_{i \in [k]} z_i^2] = \frac{k}{d}$. Linearity of Expectation.

Expected value of y_i.

Random projection: first k coordinates of random unit vector, z_i.

$E[\sum_{i \in [d]} z_i^2] = 1$. Linearity of Expectation.

By symmetry, each z_i is identically distributed.

$E[\sum_{i \in [k]} z_i^2] = \frac{k}{d}$. Linearity of Expectation.

Expected length is $\sqrt{\frac{k}{d}}$.

Expected value of y_i.

Random projection: first k coordinates of random unit vector, z_i.

$E[\sum_{i \in [d]} z_i^2] = 1$. Linearity of Expectation.

By symmetry, each z_i is identically distributed.

$E[\sum_{i \in [k]} z_i^2] = \frac{k}{d}$. Linearity of Expectation.

Expected length is $\sqrt{\frac{k}{d}}$.

Johnson-Lindenstrass: close to expectation.

Expected value of y_i.

Random projection: first k coordinates of random unit vector, z_i.

$E[\sum_{i \in [d]} z_i^2] = 1$. Linearity of Expectation.

By symmetry, each z_i is identically distributed.

$E[\sum_{i \in [k]} z_i^2] = \frac{k}{d}$. Linearity of Expectation.

Expected length is $\sqrt{\frac{k}{d}}$.

Johnson-Lindenstrass: close to expectation.
k is large enough $\rightarrow$

Expected value of y_i.

Random projection: first k coordinates of random unit vector, z_i.

$E[\sum_{i \in [d]} z_i^2] = 1$. Linearity of Expectation.

By symmetry, each z_i is identically distributed.

$E[\sum_{i \in [k]} z_i^2] = \frac{k}{d}$. Linearity of Expectation.

Expected length is $\sqrt{\frac{k}{d}}$.

Johnson-Lindenstrass: close to expectation.
k is large enough $\rightarrow$
$\approx (1 \pm \varepsilon)\sqrt{\frac{k}{d}}$ with decent probability.

Concentration Bounds.

z is uniformly random unit vector.

Concentration Bounds.

z is uniformly random unit vector.

Random point on the unit sphere. $E[\sum_{i \in [k]} z_i^2] = \frac{k}{d}$.

Concentration Bounds.

z is uniformly random unit vector.

Random point on the unit sphere. $E[\sum_{i \in [k]} z_i^2] = \frac{k}{d}$.

Claim: $\Pr[|z_1| > \frac{t}{\sqrt{d}}] \leq e^{-t^2/2}$

Concentration Bounds.

z is uniformly random unit vector.

Random point on the unit sphere. $E[\sum_{i \in [k]} z_i^2] = \frac{k}{d}$.

Claim: $\Pr[|z_1| > \frac{t}{\sqrt{d}}] \leq e^{-t^2/2}$

Sphere view: surface "far" from equator defined by e_1.

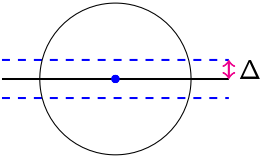
Concentration Bounds.

z is uniformly random unit vector.

Random point on the unit sphere. $E[\sum_{i\in[k]} z_i^2] = \frac{k}{d}$.

Claim: $\Pr[|z_1| > \frac{t}{\sqrt{d}}] \leq e^{-t^2/2}$

Sphere view: surface "far" from equator defined by e_1.



Concentration Bounds.

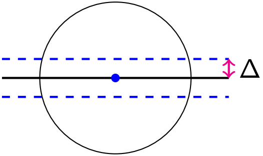
z is uniformly random unit vector.

Random point on the unit sphere. $E[\sum_{i \in [k]} z_i^2] = \frac{k}{d}$.

Claim: $\Pr[|z_1| > \frac{t}{\sqrt{d}}] \leq e^{-t^2/2}$

Sphere view: surface "far" from equator defined by e_1.

$|z_1| \geq \Delta$ if



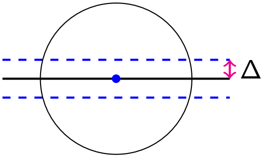
Concentration Bounds.

z is uniformly random unit vector.
  Random point on the unit sphere. $E[\sum_{i \in [k]} z_i^2] = \frac{k}{d}$.

Claim: $\Pr[|z_1| > \frac{t}{\sqrt{d}}] \leq e^{-t^2/2}$

  Sphere view: surface "far" from equator defined by e_1.

$|z_1| \geq \Delta$ if
$z \geq \Delta$ from equator of sphere.

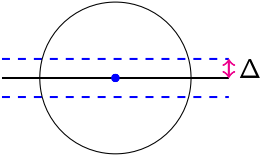


Concentration Bounds.

z is uniformly random unit vector.
Random point on the unit sphere. $E[\sum_{i\in[k]} z_i^2] = \frac{k}{d}$.

Claim: $\Pr[|z_1| > \frac{t}{\sqrt{d}}] \leq e^{-t^2/2}$

Sphere view: surface “far” from equator defined by e_1.



$|z_1| \geq \Delta$ if
$z \geq \Delta$ from equator of sphere.
Point on “Δ-spherical cap”.

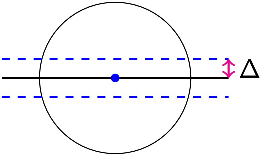
Concentration Bounds.

z is uniformly random unit vector.

Random point on the unit sphere. $E[\sum_{i \in [k]} z_i^2] = \frac{k}{d}$.

Claim: $\Pr[|z_1| > \frac{t}{\sqrt{d}}] \leq e^{-t^2/2}$

Sphere view: surface "far" from equator defined by e_1.



$|z_1| \geq \Delta$ if

$z \geq \Delta$ from equator of sphere.

Point on "Δ-spherical cap".

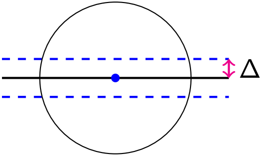
Area of caps

Concentration Bounds.

z is uniformly random unit vector.
Random point on the unit sphere. $E[\sum_{i \in [k]} z_i^2] = \frac{k}{d}$.

Claim: $\Pr[|z_1| > \frac{t}{\sqrt{d}}] \le e^{-t^2/2}$

Sphere view: surface "far" from equator defined by e_1.



$|z_1| \ge \Delta$ if
$z \ge \Delta$ from equator of sphere.
Point on "Δ-spherical cap".

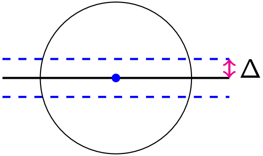
Area of caps
$\le$ S.A. of sphere of radius $\sqrt{1-\Delta^2}$

Concentration Bounds.

z is uniformly random unit vector.
Random point on the unit sphere. $E[\sum_{i\in[k]} z_i^2] = \frac{k}{d}$.

Claim: $\Pr[|z_1| > \frac{t}{\sqrt{d}}] \leq e^{-t^2/2}$

Sphere view: surface "far" from equator defined by e_1.



$|z_1| \geq \Delta$ if
$z \geq \Delta$ from equator of sphere.
Point on "Δ-spherical cap".

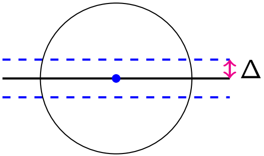
Area of caps
$\leq$ S.A. of sphere of radius $\sqrt{1-\Delta^2}$
$\propto r^d = \left(1-\Delta^2\right)^{d/2}$

Concentration Bounds.

z is uniformly random unit vector.
  Random point on the unit sphere. $E[\sum_{i \in [k]} z_i^2] = \frac{k}{d}$.

Claim: $\Pr[|z_1| > \frac{t}{\sqrt{d}}] \leq e^{-t^2/2}$

  Sphere view: surface "far" from equator defined by e_1.



$|z_1| \geq \Delta$ if
 $z \geq \Delta$ from equator of sphere.
 Point on "Δ-spherical cap".

Area of caps
  $\leq$ S.A. of sphere of radius $\sqrt{1-\Delta^2}$
  $\propto r^d = \left(1-\Delta^2\right)^{d/2}$
  $\propto \left(1 - \frac{t^2}{d}\right)^{d/2}$

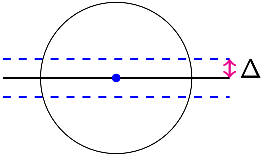
Concentration Bounds.

z is uniformly random unit vector.

Random point on the unit sphere. $E[\sum_{i\in[k]} z_i^2] = \frac{k}{d}$.

Claim: $\Pr[|z_1| > \frac{t}{\sqrt{d}}] \le e^{-t^2/2}$

Sphere view: surface "far" from equator defined by e_1.



$|z_1| \ge \Delta$ if

$z \ge \Delta$ from equator of sphere.

Point on "Δ-spherical cap".

Area of caps

$\le$ S.A. of sphere of radius $\sqrt{1-\Delta^2}$

$\propto r^d = \left(1-\Delta^2\right)^{d/2}$

$\propto \left(1-\frac{t^2}{d}\right)^{d/2} \approx e^{\frac{-t^2}{2}}$

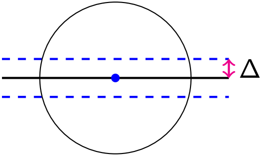
Concentration Bounds.

z is uniformly random unit vector.

Random point on the unit sphere. $E[\sum_{i \in [k]} z_i^2] = \frac{k}{d}$.

Claim: $\Pr[|z_1| > \frac{t}{\sqrt{d}}] \le e^{-t^2/2}$

Sphere view: surface "far" from equator defined by e_1.



$|z_1| \ge \Delta$ if

$z \ge \Delta$ from equator of sphere.

Point on "Δ-spherical cap".

Area of caps

$\le$ S.A. of sphere of radius $\sqrt{1-\Delta^2}$

$\propto r^d = \left(1-\Delta^2\right)^{d/2}$

$\propto \left(1 - \frac{t^2}{d}\right)^{d/2} \approx e^{\frac{-t^2}{2}}$

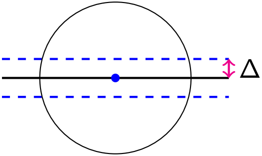
Constant of $\propto$ is unit sphere area. □

Concentration Bounds.

z is uniformly random unit vector.
Random point on the unit sphere. $E[\sum_{i \in [k]} z_i^2] = \frac{k}{d}$.

Claim: $\Pr[|z_1| > \frac{t}{\sqrt{d}}] \leq e^{-t^2/2}$

Sphere view: surface "far" from equator defined by e_1.



$|z_1| \geq \Delta$ if
$z \geq \Delta$ from equator of sphere.
Point on "Δ-spherical cap".

Area of caps
$\leq$ S.A. of sphere of radius $\sqrt{1-\Delta^2}$
$\propto r^d = \left(1-\Delta^2\right)^{d/2}$
$\propto \left(1 - \frac{t^2}{d}\right)^{d/2} \approx e^{\frac{-t^2}{2}}$
Constant of $\propto$ is unit sphere area. □

$\Pr[\text{any } z_i^2 > (2\log d)E[z_i^2]]$ is small.

Many coordinates.

Argued Pr[any $z_i^2 > (2\log d)E[z_i^2]$] is small.

Many coordinates.

Argued $\Pr[\text{any } z_i^2 > (2\log d)E[z_i^2]]$ is small.

Total Length?

Many coordinates.

Argued Pr[any $z_i^2 > (2\log d)E[z_i^2]$] is small.

Total Length? $z = \sqrt{z_1^2 + z_2^2 + \cdots z_k^2}$.

Many coordinates.

Argued $\Pr[\text{any } z_i^2 > (2\log d)E[z_i^2]]$ is small.

Total Length? $z = \sqrt{z_1^2 + z_2^2 + \cdots z_k^2}$.

$$\Pr\left[\left|\sqrt{(z_1^2 + z_2^2 + \cdots + z_k^2)} - \sqrt{\frac{k}{d}}\right| > t\right] \leq e^{-t^2 d/2}$$

Many coordinates.

Argued Pr[any $z_i^2 > (2\log d)E[z_i^2]]$ is small.

Total Length? $z = \sqrt{z_1^2 + z_2^2 + \cdots z_k^2}$.

$$\Pr[\left|\sqrt{(z_1^2 + z_2^2 + \cdots + z_k^2)} - \sqrt{\frac{k}{d}}\right| > t] \leq e^{-t^2 d/2}$$

Substituting $t = \varepsilon\sqrt{\frac{k}{d}}$, $k = \frac{c\log n}{\varepsilon^2}$.

Many coordinates.

Argued $\Pr[\text{any } z_i^2 > (2\log d)E[z_i^2]]$ is small.

Total Length? $z = \sqrt{z_1^2 + z_2^2 + \cdots z_k^2}$.

$$\Pr\left[\left|\sqrt{(z_1^2 + z_2^2 + \cdots + z_k^2)} - \sqrt{\frac{k}{d}}\right| > t\right] \le e^{-t^2 d/2}$$

Substituting $t = \varepsilon\sqrt{\frac{k}{d}}$, $k = \frac{c\log n}{\varepsilon^2}$.

$$\Pr\left[\left|\sqrt{z_1^2 + z_2^2 + \cdots + z_k^2} - \sqrt{\frac{k}{d}}\right|\right.$$

Many coordinates.

Argued Pr[any $z_i^2 > (2\log d)E[z_i^2]]$ is small.

Total Length? $z = \sqrt{z_1^2 + z_2^2 + \cdots z_k^2}$.

$$\Pr[\left|\sqrt{(z_1^2 + z_2^2 + \cdots + z_k^2)} - \sqrt{\frac{k}{d}}\right| > t] \leq e^{-t^2 d/2}$$

Substituting $t = \varepsilon\sqrt{\frac{k}{d}}$, $k = \frac{c\log n}{\varepsilon^2}$.

$$\Pr[\left|\sqrt{z_1^2 + z_2^2 + \cdots + z_k^2} - \sqrt{\frac{k}{d}}\right| > \varepsilon\sqrt{\frac{k}{d}}]$$

Many coordinates.

Argued Pr[any $z_i^2 > (2\log d)E[z_i^2]]$ is small.

Total Length? $z = \sqrt{z_1^2 + z_2^2 + \cdots z_k^2}$.

$$\Pr[\left|\sqrt{(z_1^2 + z_2^2 + \cdots + z_k^2)} - \sqrt{\frac{k}{d}}\right| > t] \leq e^{-t^2 d/2}$$

Substituting $t = \varepsilon\sqrt{\frac{k}{d}}$, $k = \frac{c\log n}{\varepsilon^2}$.

$$\Pr[\left|\sqrt{z_1^2 + z_2^2 + \cdots + z_k^2} - \sqrt{\frac{k}{d}}\right| > \varepsilon\sqrt{\frac{k}{d}}] \leq e^{-\varepsilon^2 k}$$

Many coordinates.

Argued Pr[any $z_i^2 > (2\log d)E[z_i^2]$] is small.

Total Length? $z = \sqrt{z_1^2 + z_2^2 + \cdots z_k^2}$.

$$\Pr[\left|\sqrt{(z_1^2 + z_2^2 + \cdots + z_k^2)} - \sqrt{\frac{k}{d}}\right| > t] \le e^{-t^2 d/2}$$

Substituting $t = \varepsilon\sqrt{\frac{k}{d}}$, $k = \frac{c\log n}{\varepsilon^2}$.

$$\Pr[\left|\sqrt{z_1^2 + z_2^2 + \cdots + z_k^2} - \sqrt{\frac{k}{d}}\right| > \varepsilon\sqrt{\frac{k}{d}}] \le e^{-\varepsilon^2 k} = e^{-c\log n} = \frac{1}{n^c}$$

Many coordinates.

Argued $\Pr[\text{any } z_i^2 > (2\log d)E[z_i^2]]$ is small.

Total Length? $z = \sqrt{z_1^2 + z_2^2 + \cdots z_k^2}$.

$$\Pr\left[\left|\sqrt{(z_1^2 + z_2^2 + \cdots + z_k^2)} - \sqrt{\frac{k}{d}}\right| > t\right] \leq e^{-t^2 d/2}$$

Substituting $t = \varepsilon\sqrt{\frac{k}{d}}$, $k = \frac{c\log n}{\varepsilon^2}$.

$$\Pr\left[\left|\sqrt{z_1^2 + z_2^2 + \cdots + z_k^2} - \sqrt{\frac{k}{d}}\right| > \varepsilon\sqrt{\frac{k}{d}}\right] \leq e^{-\varepsilon^2 k} = e^{-c\log n} = \frac{1}{n^c}$$

Johnson-Lindenstraus: For n points, $x_1, \ldots, x_n$, all distances preserved to within $1 \pm \varepsilon$ under $\sqrt{\frac{k}{d}}$-scaled projection above.

Many coordinates.

Argued Pr[any $z_i^2 > (2\log d)E[z_i^2]$] is small.

Total Length? $z = \sqrt{z_1^2 + z_2^2 + \cdots z_k^2}$.

$$\Pr[\left|\sqrt{(z_1^2 + z_2^2 + \cdots + z_k^2)} - \sqrt{\frac{k}{d}}\right| > t] \leq e^{-t^2 d/2}$$

Substituting $t = \varepsilon\sqrt{\frac{k}{d}}$, $k = \frac{c\log n}{\varepsilon^2}$.

$$\Pr[\left|\sqrt{z_1^2 + z_2^2 + \cdots + z_k^2} - \sqrt{\frac{k}{d}}\right| > \varepsilon\sqrt{\frac{k}{d}}] \leq e^{-\varepsilon^2 k} = e^{-c\log n} = \frac{1}{n^c}$$

Johnson-Lindenstraus: For n points, $x_1, \ldots, x_n$, all distances preserved to within $1 \pm \varepsilon$ under $\sqrt{\frac{k}{d}}$-scaled projection above.

View one pair $x_i - x_j$ as vector.

Many coordinates.

Argued $\Pr[\text{any } z_i^2 > (2\log d)E[z_i^2]]$ is small.

Total Length? $z = \sqrt{z_1^2 + z_2^2 + \cdots z_k^2}$.

$$\Pr[\left|\sqrt{(z_1^2 + z_2^2 + \cdots + z_k^2)} - \sqrt{\frac{k}{d}}\right| > t] \le e^{-t^2 d/2}$$

Substituting $t = \varepsilon\sqrt{\frac{k}{d}}$, $k = \frac{c\log n}{\varepsilon^2}$.

$$\Pr[\left|\sqrt{z_1^2 + z_2^2 + \cdots + z_k^2} - \sqrt{\frac{k}{d}}\right| > \varepsilon\sqrt{\frac{k}{d}}] \le e^{-\varepsilon^2 k} = e^{-c\log n} = \frac{1}{n^c}$$

Johnson-Lindenstraus: For n points, $x_1, \ldots, x_n$, all distances preserved to within $1 \pm \varepsilon$ under $\sqrt{\frac{k}{d}}$-scaled projection above.

View one pair $x_i - x_j$ as vector.
Scale to unit.

Many coordinates.

Argued Pr[any $z_i^2 > (2\log d)E[z_i^2]$] is small.

Total Length? $z = \sqrt{z_1^2 + z_2^2 + \cdots z_k^2}$.

$$\Pr[\left|\sqrt{(z_1^2 + z_2^2 + \cdots + z_k^2)} - \sqrt{\frac{k}{d}}\right| > t] \le e^{-t^2 d/2}$$

Substituting $t = \varepsilon\sqrt{\frac{k}{d}}$, $k = \frac{c\log n}{\varepsilon^2}$.

$$\Pr[\left|\sqrt{z_1^2 + z_2^2 + \cdots + z_k^2} - \sqrt{\frac{k}{d}}\right| > \varepsilon\sqrt{\frac{k}{d}}] \le e^{-\varepsilon^2 k} = e^{-c\log n} = \frac{1}{n^c}$$

Johnson-Lindenstraus: For n points, $x_1, \ldots, x_n$, all distances preserved to within $1 \pm \varepsilon$ under $\sqrt{\frac{k}{d}}$-scaled projection above.

View one pair $x_i - x_j$ as vector.
Scale to unit.
Projection fails to preserve $|x_i - x_j|$

Many coordinates.

Argued $\Pr[\text{any } z_i^2 > (2\log d)E[z_i^2]]$ is small.

Total Length? $z = \sqrt{z_1^2 + z_2^2 + \cdots z_k^2}$.

$$\Pr\left[\left|\sqrt{(z_1^2 + z_2^2 + \cdots + z_k^2)} - \sqrt{\frac{k}{d}}\right| > t\right] \le e^{-t^2 d/2}$$

Substituting $t = \varepsilon\sqrt{\frac{k}{d}}$, $k = \frac{c\log n}{\varepsilon^2}$.

$$\Pr\left[\left|\sqrt{z_1^2 + z_2^2 + \cdots + z_k^2} - \sqrt{\frac{k}{d}}\right| > \varepsilon\sqrt{\frac{k}{d}}\right] \le e^{-\varepsilon^2 k} = e^{-c\log n} = \frac{1}{n^c}$$

Johnson-Lindenstraus: For n points, $x_1, \ldots, x_n$, all distances preserved to within $1 \pm \varepsilon$ under $\sqrt{\frac{k}{d}}$-scaled projection above.

View one pair $x_i - x_j$ as vector.
Scale to unit.
Projection fails to preserve $|x_i - x_j|$
with probability $\le \frac{1}{n^c}$

Many coordinates.

Argued $\Pr[\text{any } z_i^2 > (2\log d)E[z_i^2]]$ is small.

Total Length? $z = \sqrt{z_1^2 + z_2^2 + \cdots z_k^2}$.

$$\Pr\left[\left|\sqrt{(z_1^2 + z_2^2 + \cdots + z_k^2)} - \sqrt{\frac{k}{d}}\right| > t\right] \le e^{-t^2 d/2}$$

Substituting $t = \varepsilon\sqrt{\frac{k}{d}}$, $k = \frac{c\log n}{\varepsilon^2}$.

$$\Pr\left[\left|\sqrt{z_1^2 + z_2^2 + \cdots + z_k^2} - \sqrt{\frac{k}{d}}\right| > \varepsilon\sqrt{\frac{k}{d}}\right] \le e^{-\varepsilon^2 k} = e^{-c\log n} = \frac{1}{n^c}$$

Johnson-Lindenstraus: For n points, $x_1, \ldots, x_n$, all distances preserved to within $1 \pm \varepsilon$ under $\sqrt{\frac{k}{d}}$-scaled projection above.

View one pair $x_i - x_j$ as vector.
Scale to unit.
Projection fails to preserve $|x_i - x_j|$
with probability $\le \frac{1}{n^c}$
Scaled vector length also preserved.

Many coordinates.

Argued $\Pr[\text{any } z_i^2 > (2\log d)E[z_i^2]]$ is small.

Total Length? $z = \sqrt{z_1^2 + z_2^2 + \cdots z_k^2}$.

$$\Pr[\left|\sqrt{(z_1^2 + z_2^2 + \cdots + z_k^2)} - \sqrt{\frac{k}{d}}\right| > t] \leq e^{-t^2 d/2}$$

Substituting $t = \varepsilon\sqrt{\frac{k}{d}}$, $k = \frac{c\log n}{\varepsilon^2}$.

$$\Pr[\left|\sqrt{z_1^2 + z_2^2 + \cdots + z_k^2} - \sqrt{\frac{k}{d}}\right| > \varepsilon\sqrt{\frac{k}{d}}] \leq e^{-\varepsilon^2 k} = e^{-c\log n} = \frac{1}{n^c}$$

Johnson-Lindenstraus: For n points, $x_1, \ldots, x_n$, all distances preserved to within $1 \pm \varepsilon$ under $\sqrt{\frac{k}{d}}$-scaled projection above.

View one pair $x_i - x_j$ as vector.
Scale to unit.
Projection fails to preserve $|x_i - x_j|$
with probability $\leq \frac{1}{n^c}$
Scaled vector length also preserved.

$\leq n^2$ pairs

Many coordinates.

Argued $\Pr[\text{any } z_i^2 > (2\log d)E[z_i^2]]$ is small.

Total Length? $z = \sqrt{z_1^2 + z_2^2 + \cdots z_k^2}$.

$$\Pr\left[\left|\sqrt{(z_1^2 + z_2^2 + \cdots + z_k^2)} - \sqrt{\frac{k}{d}}\right| > t\right] \le e^{-t^2 d/2}$$

Substituting $t = \varepsilon\sqrt{\frac{k}{d}}$, $k = \frac{c\log n}{\varepsilon^2}$.

$$\Pr\left[\left|\sqrt{z_1^2 + z_2^2 + \cdots + z_k^2} - \sqrt{\frac{k}{d}}\right| > \varepsilon\sqrt{\frac{k}{d}}\right] \le e^{-\varepsilon^2 k} = e^{-c\log n} = \frac{1}{n^c}$$

Johnson-Lindenstraus: For n points, $x_1, \ldots, x_n$, all distances preserved to within $1 \pm \varepsilon$ under $\sqrt{\frac{k}{d}}$-scaled projection above.

View one pair $x_i - x_j$ as vector.
Scale to unit.
Projection fails to preserve $|x_i - x_j|$
with probability $\le \frac{1}{n^c}$
Scaled vector length also preserved.

$\le n^2$ pairs plus union bound

Many coordinates.

Argued Pr[any $z_i^2 > (2\log d)E[z_i^2]]$ is small.

Total Length? $z = \sqrt{z_1^2 + z_2^2 + \cdots z_k^2}$.

$$\Pr[\left|\sqrt{(z_1^2 + z_2^2 + \cdots + z_k^2)} - \sqrt{\frac{k}{d}}\right| > t] \le e^{-t^2 d/2}$$

Substituting $t = \varepsilon\sqrt{\frac{k}{d}}$, $k = \frac{c\log n}{\varepsilon^2}$.

$$\Pr[\left|\sqrt{z_1^2 + z_2^2 + \cdots + z_k^2} - \sqrt{\frac{k}{d}}\right| > \varepsilon\sqrt{\frac{k}{d}}] \le e^{-\varepsilon^2 k} = e^{-c\log n} = \frac{1}{n^c}$$

Johnson-Lindenstraus: For n points, $x_1, \ldots, x_n$, all distances preserved to within $1 \pm \varepsilon$ under $\sqrt{\frac{k}{d}}$-scaled projection above.

View one pair $x_i - x_j$ as vector.
Scale to unit.
Projection fails to preserve $|x_i - x_j|$
with probability $\le \frac{1}{n^c}$
Scaled vector length also preserved.

$\le n^2$ pairs plus union bound
$\rightarrow$ prob any pair fails to be preserved with $\le \frac{1}{n^{c-2}}$.

Locality Preserving Hashing

Find nearby points in high dimensional space.

Locality Preserving Hashing

Find nearby points in high dimensional space.
  Points could be images!

Locality Preserving Hashing

Find nearby points in high dimensional space.
 Points could be images!

Hash function $h(\cdot)$ s.t. $h(x_i) = h(x_j)$ if $d(x_i, x_j) \leq \delta$.

Locality Preserving Hashing

Find nearby points in high dimensional space.
Points could be images!

Hash function $h(\cdot)$ s.t. $h(x_i) = h(x_j)$ if $d(x_i, x_j) \leq \delta$.

Low dimensions: grid cells give $\sqrt{d}$-approximation.

Locality Preserving Hashing

Find nearby points in high dimensional space.
  Points could be images!

Hash function $h(\cdot)$ s.t. $h(x_i) = h(x_j)$ if $d(x_i, x_j) \leq \delta$.

Low dimensions: grid cells give $\sqrt{d}$-approximation.
Not quite a solution.

Locality Preserving Hashing

Find nearby points in high dimensional space.
  Points could be images!

Hash function $h(\cdot)$ s.t. $h(x_i) = h(x_j)$ if $d(x_i, x_j) \leq \delta$.

Low dimensions: grid cells give $\sqrt{d}$-approximation.
Not quite a solution. Why?

Locality Preserving Hashing

Find nearby points in high dimensional space.
Points could be images!

Hash function $h(\cdot)$ s.t. $h(x_i) = h(x_j)$ if $d(x_i, x_j) \leq \delta$.

Low dimensions: grid cells give $\sqrt{d}$-approximation.
Not quite a solution. Why?
Close to grid boundary.

Locality Preserving Hashing

Find nearby points in high dimensional space.
  Points could be images!

Hash function $h(\cdot)$ s.t. $h(x_i) = h(x_j)$ if $d(x_i, x_j) \leq \delta$.

Low dimensions: grid cells give $\sqrt{d}$-approximation.
Not quite a solution. Why?
  Close to grid boundary.
Find close points to x:

Locality Preserving Hashing

Find nearby points in high dimensional space.
  Points could be images!

Hash function $h(\cdot)$ s.t. $h(x_i) = h(x_j)$ if $d(x_i, x_j) \leq \delta$.

Low dimensions: grid cells give $\sqrt{d}$-approximation.
Not quite a solution. Why?
  Close to grid boundary.
Find close points to x:
  Check grid cell and neighboring grid cells.

Locality Preserving Hashing

Find nearby points in high dimensional space.
  Points could be images!

Hash function $h(\cdot)$ s.t. $h(x_i) = h(x_j)$ if $d(x_i, x_j) \leq \delta$.

Low dimensions: grid cells give $\sqrt{d}$-approximation.
Not quite a solution. Why?
  Close to grid boundary.
Find close points to x:
  Check grid cell and neighboring grid cells.

Locality Preserving Hashing

Find nearby points in high dimensional space.
  Points could be images!

Hash function $h(\cdot)$ s.t. $h(x_i) = h(x_j)$ if $d(x_i, x_j) \leq \delta$.

Low dimensions: grid cells give $\sqrt{d}$-approximation.
Not quite a solution. Why?
  Close to grid boundary.
Find close points to x:
  Check grid cell and neighboring grid cells.

Project high dimensional points into low dimensions.

Locality Preserving Hashing

Find nearby points in high dimensional space.
  Points could be images!

Hash function $h(\cdot)$ s.t. $h(x_i) = h(x_j)$ if $d(x_i, x_j) \leq \delta$.

Low dimensions: grid cells give $\sqrt{d}$-approximation.
Not quite a solution. Why?
  Close to grid boundary.
Find close points to x:
  Check grid cell and neighboring grid cells.

Project high dimensional points into low dimensions.

  Use grid hash function.

Implementing Johnson-Lindenstraus

Random vectors

Implementing Johnson-Lindenstraus

Random vectors have many bits

Implementing Johnson-Lindenstraus

Random vectors have many bits

Use random bit vectors: $\{-1, +1\}^d$ instead.

Implementing Johnson-Lindenstraus

Random vectors have many bits

Use random bit vectors: $\{-1, +1\}^d$ instead.

Almost orthogonal.

Implementing Johnson-Lindenstraus

Random vectors have many bits

Use random bit vectors: $\{-1, +1\}^d$ instead.

Almost orthogonal.

Project z.

Implementing Johnson-Lindenstraus

Random vectors have many bits

Use random bit vectors: $\{-1,+1\}^d$ instead.

  Almost orthogonal.

Project z.

Coordinate for bit vector b.

Implementing Johnson-Lindenstraus

Random vectors have many bits

Use random bit vectors: $\{-1, +1\}^d$ instead.

Almost orthogonal.

Project z.

Coordinate for bit vector b.

$C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

Implementing Johnson-Lindenstraus

Random vectors have many bits

Use random bit vectors: $\{-1,+1\}^d$ instead.

Almost orthogonal.

Project z.

Coordinate for bit vector b.

$C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

$E[C_l^2] =$

Implementing Johnson-Lindenstraus

Random vectors have many bits

Use random bit vectors: $\{-1,+1\}^d$ instead.

Almost orthogonal.

Project z.

Coordinate for bit vector b.

$C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

$E[C_l^2] = E[\frac{1}{d} \sum_{i,j} b_i b_j z_i z_j]$

Implementing Johnson-Lindenstraus

Random vectors have many bits

Use random bit vectors: $\{-1,+1\}^d$ instead.

Almost orthogonal.

Project z.

Coordinate for bit vector b.

$C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

$E[C_l^2] = E[\frac{1}{d} \sum_{i,j} b_i b_j z_i z_j] = \frac{1}{d} \sum_{i,j} E[b_i b_j] z_i z_j = \frac{1}{d} \sum_i z_i^2$

Implementing Johnson-Lindenstraus

Random vectors have many bits

Use random bit vectors: $\{-1,+1\}^d$ instead.

Almost orthogonal.

Project z.

Coordinate for bit vector b.

$C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

$E[C_l^2] = E[\frac{1}{d} \sum_{i,j} b_i b_j z_i z_j] = \frac{1}{d} \sum_{i,j} E[b_i b_j] z_i z_j = \frac{1}{d} \sum_i z_i^2 = \frac{1}{d}$

Implementing Johnson-Lindenstraus

Random vectors have many bits

Use random bit vectors: $\{-1, +1\}^d$ instead.

Almost orthogonal.

Project z.

Coordinate for bit vector b.

$C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

$E[C_l^2] = E[\frac{1}{d} \sum_{i,j} b_i b_j z_i z_j] = \frac{1}{d} \sum_{i,j} E[b_i b_j] z_i z_j = \frac{1}{d} \sum_i z_i^2 = \frac{1}{d}$

Implementing Johnson-Lindenstraus

Random vectors have many bits

Use random bit vectors: $\{-1,+1\}^d$ instead.

Almost orthogonal.

Project z.

Coordinate for bit vector b.

$C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

$E[C_l^2] = E[\frac{1}{d} \sum_{i,j} b_i b_j z_i z_j] = \frac{1}{d} \sum_{i,j} E[b_i b_j] z_i z_j = \frac{1}{d} \sum_i z_i^2 = \frac{1}{d}$

$E[\sum_l C_l^2] = \frac{k}{d}$

Binary Johnson-Lindenstrass

Project onto $[-1, +1]$ vectors.

Binary Johnson-Lindenstrass

Project onto $[-1, +1]$ vectors.

$E[C] = E[\sum_l C_l^2] = \frac{k}{d}$

Binary Johnson-Lindenstrass

Project onto $[-1, +1]$ vectors.

$E[C] = E[\sum_l C_l^2] = \frac{k}{d}$

Concentration?

Binary Johnson-Lindenstrass

Project onto $[-1, +1]$ vectors.

$E[C] = E[\sum_l C_l^2] = \frac{k}{d}$

Concentration?

$$\Pr\left[|C - \frac{k}{d}| \geq \varepsilon \frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Binary Johnson-Lindenstrass

Project onto $[-1, +1]$ vectors.

$E[C] = E[\sum_l C_l^2] = \frac{k}{d}$

Concentration?

$$\Pr\left[|C - \frac{k}{d}| \geq \varepsilon \frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Choose $k = \frac{c \log n}{\varepsilon^2}$.

Binary Johnson-Lindenstrass

Project onto $[-1, +1]$ vectors.

$E[C] = E[\sum_l C_l^2] = \frac{k}{d}$

Concentration?

$$\Pr\left[|C - \frac{k}{d}| \geq \varepsilon \frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Choose $k = \frac{c \log n}{\varepsilon^2}$.

$\rightarrow$ failure probability $\leq 1/n^c$.

Analysis Idea.

$$\Pr\left[|C - \frac{k}{d}| \geq \varepsilon \frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Analysis Idea.

$$\Pr\left[|C - \frac{k}{d}| \geq \varepsilon \frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Variance of C^2?

Analysis Idea.

$$\Pr\left[|C - \frac{k}{d}| \geq \varepsilon \frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Variance of C^2? Recall $C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

Analysis Idea.

$$\Pr\left[|C - \frac{k}{d}| \geq \varepsilon \frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Variance of C^2? Recall $C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

$Var(C) \leq \left(\frac{k}{d^2}\right) (\sum_i {z_i}^4 + 4 \sum_{i,j} z_i^2 z_j^2)$

Analysis Idea.

$$\Pr\left[|C - \frac{k}{d}| \geq \varepsilon \frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Variance of C^2? Recall $C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

$Var(C) \leq \left(\frac{k}{d^2}\right) (\sum_i {z_i}^4 + 4 \sum_{i,j} z_i^2 z_j^2) \leq \left(\frac{k}{d^2}\right) 2 (\sum_i z_i^2)^2$

Analysis Idea.

$$\Pr\left[|C - \frac{k}{d}| \geq \varepsilon \frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Variance of C^2? Recall $C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

$Var(C) \leq \left(\frac{k}{d^2}\right)(\sum_i {z_i}^4 + 4\sum_{i,j} z_i^2 z_j^2) \leq \left(\frac{k}{d^2}\right) 2(\sum_i z_i^2)^2 \leq \frac{2k}{d^2}$.

Analysis Idea.

$$\Pr\left[|C - \frac{k}{d}| \geq \varepsilon \frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Variance of C^2? Recall $C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

$Var(C) \leq \left(\frac{k}{d^2}\right) (\sum_i {z_i}^4 + 4 \sum_{i,j} z_i^2 z_j^2) \leq \left(\frac{k}{d^2}\right) 2(\sum_i z_i^2)^2 \leq \frac{2k}{d^2}$.

Roughly normal (gaussian):

Analysis Idea.

$$\Pr\left[|C - \frac{k}{d}| \geq \varepsilon \frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Variance of C^2? Recall $C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

$Var(C) \leq \left(\frac{k}{d^2}\right) (\sum_i {z_i}^4 + 4 \sum_{i,j} z_i^2 z_j^2) \leq \left(\frac{k}{d^2}\right) 2(\sum_i z_i^2)^2 \leq \frac{2k}{d^2}$.

Roughly normal (gaussian):

Density $\propto e^{-t^2/2}$ for t std deviations away.

Analysis Idea.

$$\Pr\left[|C-\frac{k}{d}| \geq \varepsilon \frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Variance of C^2? Recall $C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

$Var(C) \leq \left(\frac{k}{d^2}\right)(\sum_i {z_i}^4 + 4\sum_{i,j} z_i^2 z_j^2) \leq \left(\frac{k}{d^2}\right) 2(\sum_i z_i^2)^2 \leq \frac{2k}{d^2}$.

Roughly normal (gaussian):

Density $\propto e^{-t^2/2}$ for t std deviations away.

So, assuming normality

Analysis Idea.

$$\Pr\left[|C-\frac{k}{d}| \geq \varepsilon\frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Variance of C^2? Recall $C_l = \frac{1}{\sqrt{d}}\sum_i b_i z_i$

$Var(C) \leq \left(\frac{k}{d^2}\right)(\sum_i {z_i}^4 + 4\sum_{i,j} z_i^2 z_j^2) \leq \left(\frac{k}{d^2}\right) 2(\sum_i z_i^2)^2 \leq \frac{2k}{d^2}$.

Roughly normal (gaussian):

Density $\propto e^{-t^2/2}$ for t std deviations away.

So, assuming normality

$\sigma = \frac{\sqrt{2k}}{d}$,

Analysis Idea.

$$\Pr\left[|C - \frac{k}{d}| \geq \varepsilon \frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Variance of C^2? Recall $C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

$Var(C) \leq \left(\frac{k}{d^2}\right)(\sum_i z_i{}^4 + 4\sum_{i,j} z_i^2 z_j^2) \leq \left(\frac{k}{d^2}\right) 2(\sum_i z_i^2)^2 \leq \frac{2k}{d^2}$.

Roughly normal (gaussian):

Density $\propto e^{-t^2/2}$ for t std deviations away.

So, assuming normality

$\sigma = \frac{\sqrt{2k}}{d}$, $t = \frac{\varepsilon \frac{k}{d}}{\frac{\sqrt{2k}}{d}}$

Analysis Idea.

$$\Pr\left[|C - \frac{k}{d}| \geq \varepsilon \frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Variance of C^2? Recall $C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

$Var(C) \leq \left(\frac{k}{d^2}\right) (\sum_i {z_i}^4 + 4 \sum_{i,j} z_i^2 z_j^2) \leq \left(\frac{k}{d^2}\right) 2(\sum_i z_i^2)^2 \leq \frac{2k}{d^2}$.

Roughly normal (gaussian):

Density $\propto e^{-t^2/2}$ for t std deviations away.

So, assuming normality

$\sigma = \frac{\sqrt{2k}}{d}$, $t = \frac{\varepsilon \frac{k}{d}}{\frac{\sqrt{2k}}{d}} = \varepsilon \sqrt{k}/\sqrt{2}$.

Analysis Idea.

$$\Pr\left[|C - \frac{k}{d}| \geq \varepsilon \frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Variance of C^2? Recall $C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

$Var(C) \leq \left(\frac{k}{d^2}\right)(\sum_i {z_i}^4 + 4\sum_{i,j} z_i^2 z_j^2) \leq \left(\frac{k}{d^2}\right) 2(\sum_i z_i^2)^2 \leq \frac{2k}{d^2}$.

Roughly normal (gaussian):

Density $\propto e^{-t^2/2}$ for t std deviations away.

So, assuming normality

$\sigma = \frac{\sqrt{2k}}{d}$, $t = \frac{\varepsilon \frac{k}{d}}{\frac{\sqrt{2k}}{d}} = \varepsilon\sqrt{k}/\sqrt{2}$.

Probability of failure roughly $\leq e^{-t^2/2}$

Analysis Idea.

$$\Pr\left[|C - \frac{k}{d}| \geq \varepsilon \frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Variance of C^2? Recall $C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

$Var(C) \leq \left(\frac{k}{d^2}\right)(\sum_i {z_i}^4 + 4\sum_{i,j} z_i^2 z_j^2) \leq \left(\frac{k}{d^2}\right) 2(\sum_i z_i^2)^2 \leq \frac{2k}{d^2}$.

Roughly normal (gaussian):

Density $\propto e^{-t^2/2}$ for t std deviations away.

So, assuming normality

$\sigma = \frac{\sqrt{2k}}{d}$, $t = \frac{\varepsilon \frac{k}{d}}{\frac{\sqrt{2k}}{d}} = \varepsilon\sqrt{k}/\sqrt{2}$.

Probability of failure roughly $\leq e^{-t^2/2}$

$\rightarrow e^{\varepsilon^2 k/4}$

Analysis Idea.

$$\Pr\left[|C-\frac{k}{d}| \geq \varepsilon\frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Variance of C^2? Recall $C_l = \frac{1}{\sqrt{d}}\sum_i b_i z_i$

$Var(C) \leq \left(\frac{k}{d^2}\right)(\sum_i {z_i}^4 + 4\sum_{i,j} z_i^2 z_j^2) \leq \left(\frac{k}{d^2}\right) 2(\sum_i z_i^2)^2 \leq \frac{2k}{d^2}$.

Roughly normal (gaussian):

Density $\propto e^{-t^2/2}$ for t std deviations away.

So, assuming normality

$\sigma = \frac{\sqrt{2k}}{d}$, $t = \frac{\varepsilon\frac{k}{d}}{\frac{\sqrt{2k}}{d}} = \varepsilon\sqrt{k}/\sqrt{2}$.

Probability of failure roughly $\leq e^{-t^2/2}$

$\rightarrow e^{\varepsilon^2 k/4}$

"Roughly normal."

Analysis Idea.

$$\Pr\left[|C - \frac{k}{d}| \geq \varepsilon \frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Variance of C^2? Recall $C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

$Var(C) \leq \left(\frac{k}{d^2}\right)(\sum_i {z_i}^4 + 4\sum_{i,j} z_i^2 z_j^2) \leq \left(\frac{k}{d^2}\right) 2(\sum_i z_i^2)^2 \leq \frac{2k}{d^2}$.

Roughly normal (gaussian):

Density $\propto e^{-t^2/2}$ for t std deviations away.

So, assuming normality

$\sigma = \frac{\sqrt{2k}}{d}$, $t = \frac{\varepsilon \frac{k}{d}}{\frac{\sqrt{2k}}{d}} = \varepsilon\sqrt{k}/\sqrt{2}$.

Probability of failure roughly $\leq e^{-t^2/2}$

$\rightarrow e^{\varepsilon^2 k/4}$

“Roughly normal.” Chernoff, Berry-Esseen, Central Limit Theorems.

Analysis Idea.

$$\Pr\left[|C - \frac{k}{d}| \geq \varepsilon \frac{k}{d}\right] \leq e^{-\varepsilon^2 k}$$

Variance of C^2? Recall $C_l = \frac{1}{\sqrt{d}} \sum_i b_i z_i$

$Var(C) \leq \left(\frac{k}{d^2}\right)(\sum_i {z_i}^4 + 4\sum_{i,j} z_i^2 z_j^2) \leq \left(\frac{k}{d^2}\right) 2(\sum_i z_i^2)^2 \leq \frac{2k}{d^2}$.

Roughly normal (gaussian):

Density $\propto e^{-t^2/2}$ for t std deviations away.

So, assuming normality

$\sigma = \frac{\sqrt{2k}}{d}$, $t = \frac{\varepsilon \frac{k}{d}}{\frac{\sqrt{2k}}{d}} = \varepsilon\sqrt{k}/\sqrt{2}$.

Probability of failure roughly $\leq e^{-t^2/2}$

$\rightarrow e^{\varepsilon^2 k/4}$

"Roughly normal." Chernoff, Berry-Esseen, Central Limit Theorems.

Summary

Cuckoo hashing.

Summary

Cuckoo hashing.

Two hash functions.

Summary

Cuckoo hashing.

Two hash functions. Few cycles in random sparse graph.

Summary

Cuckoo hashing.

Two hash functions. Few cycles in random sparse graph.

Chaining works!

Summary

Cuckoo hashing.

Two hash functions. Few cycles in random sparse graph.
Chaining works!

Johnson-Lindenstrass.

Summary

Cuckoo hashing.

Two hash functions. Few cycles in random sparse graph.
Chaining works!

Johnson-Lindenstrass.
$O(\log n)$ dimensions give good approximation of distances.