Locality-sensitive Hashing without False Negatives

Rasmus Pagh
IT University of Copenhagen

SODA, January 10, 2016

European Research Council
Established by the European Commission

SCALABLE SIMILARITY SEARCH

1-2 post-doc positions available starting fall 2016 - contact me for details!

Talk outline

- Hamming similarity search
- Approximate similarity search using LSH
- Recent developments
- New result: Avoiding false negatives

Hamming similarity search

vanilla version

- Build data structure for set $S \subseteq \{0,1\}^d$ s.t. given query vector q and radius r, can decide

$$\exists x \in S : ||x - q|| \leq r$$

where $||x - q||$ = Hamming distance between x and q.

Hamming similarity search

vanilla version

- Build data structure for set $S \subseteq \{0,1\}^d$ s.t. given query vector q and radius r, can decide

$$\exists x \in S : ||x - q|| \leq r$$

where $||x - q||$ = Hamming distance between x and q.

- [Williams '04], [Alman & Williams '15]:

Hamming similarity search in time $n^{0.99}\, 2^{o(d)} \Longrightarrow$

k-SAT w. n variables can be solved in time α^n, $\alpha < 2$

Hamming similarity search

vanilla version

- Build data structure for set $S \subseteq \{0,1\}^d$ s.t. given query vector q and radius r, can decide

$$\exists x \in S : ||x - q|| \leq r$$

where $||x - q||$ = Hamming distance between x and q.

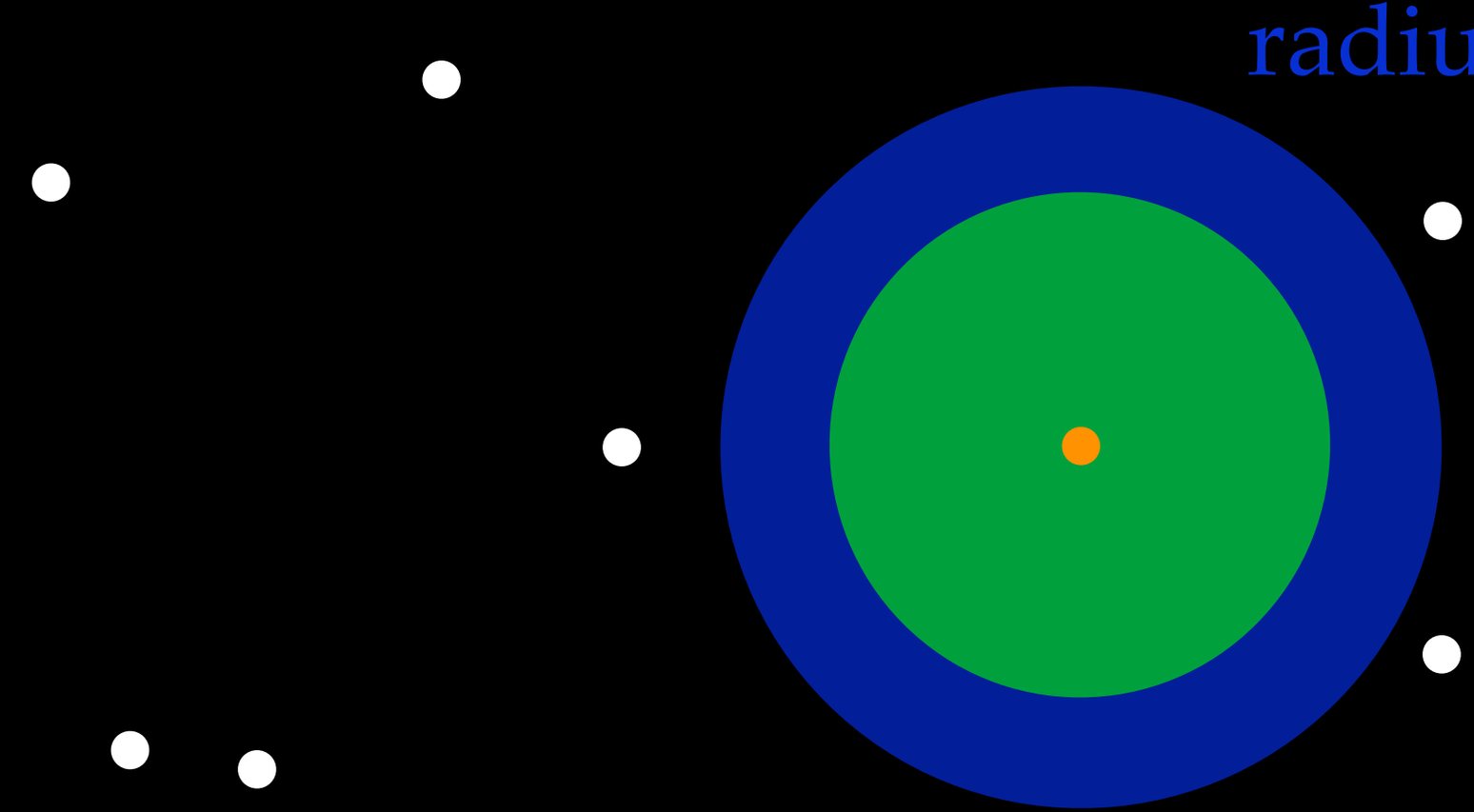
- [Williams '04], [Alman & Williams '15]:

Strong ETH states: Not possible!

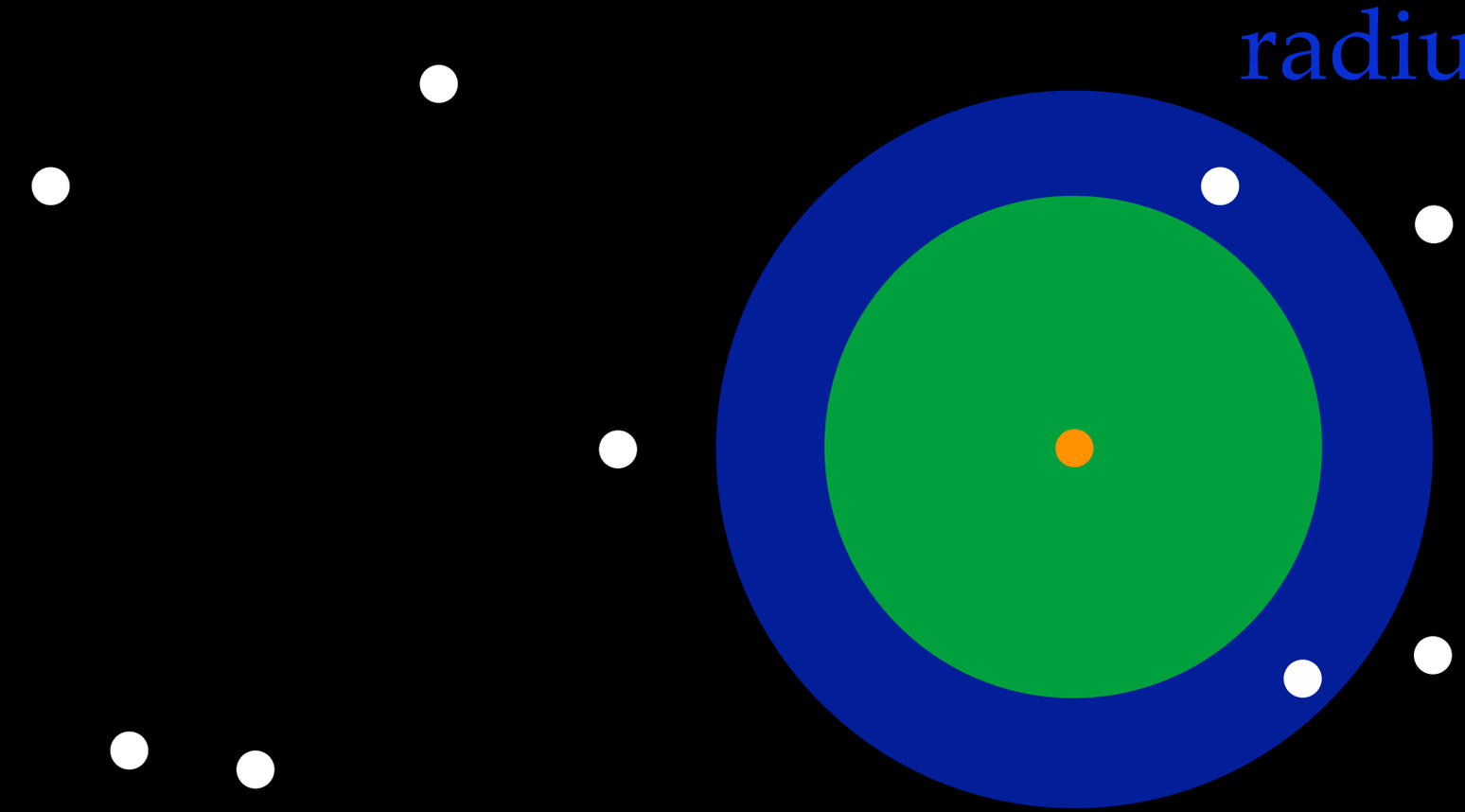
Hamming similarity search in time $n^{0.99}\, 2^{o(d)} \Longrightarrow$

k-SAT w. n variables can be solved in time α^n, $\alpha < 2$

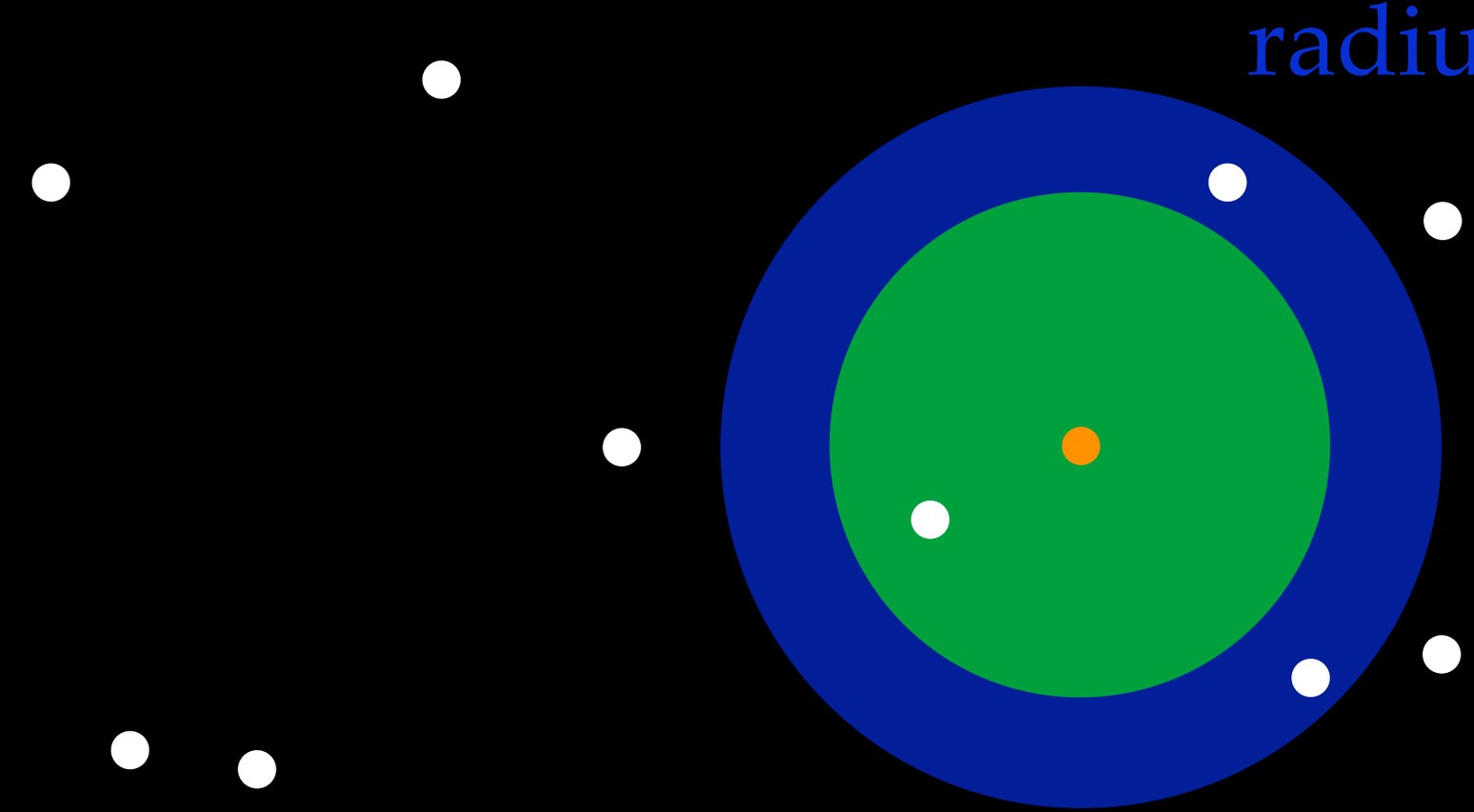
Approximate similarity search



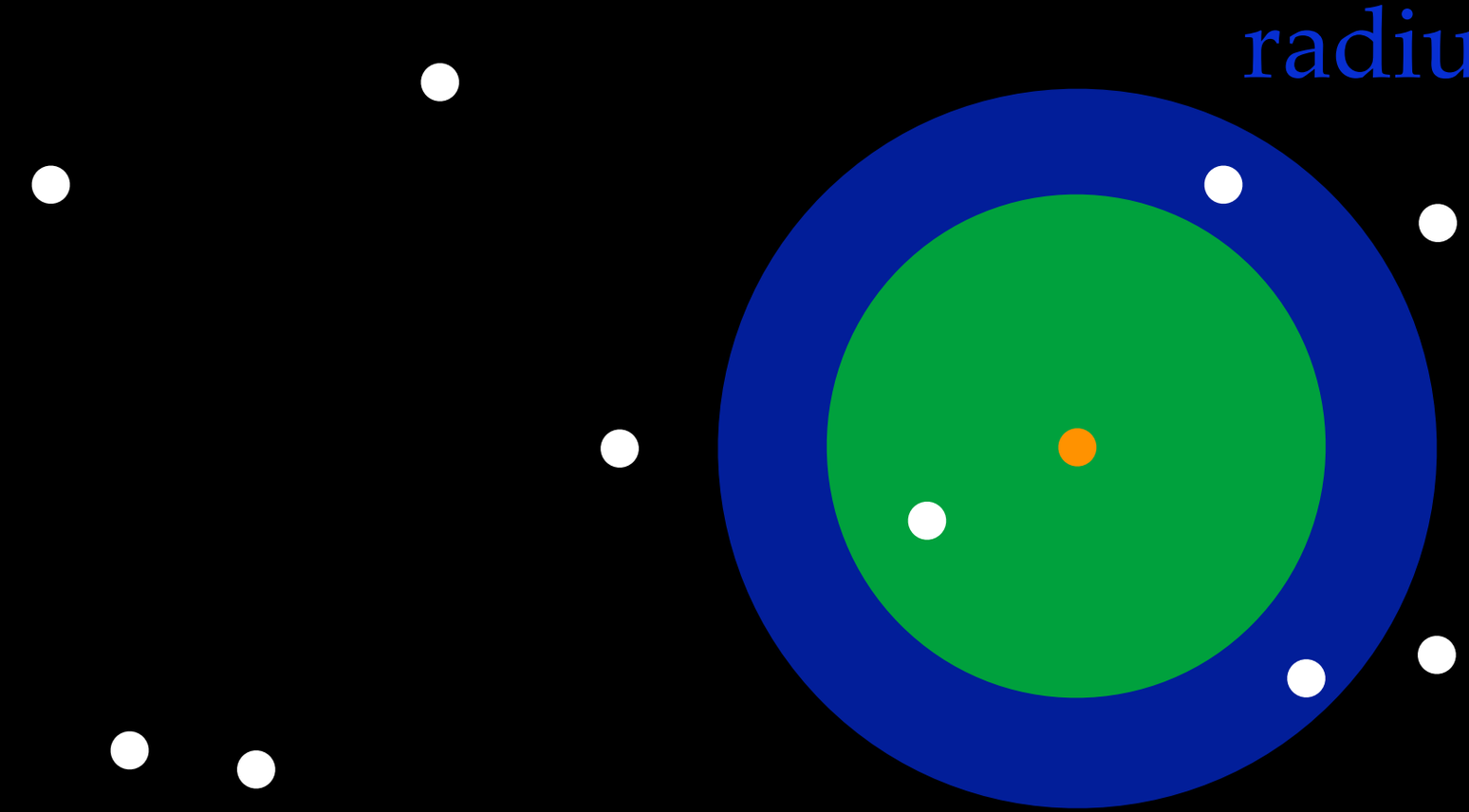
Approximate similarity search



Approximate similarity search



Approximate similarity search



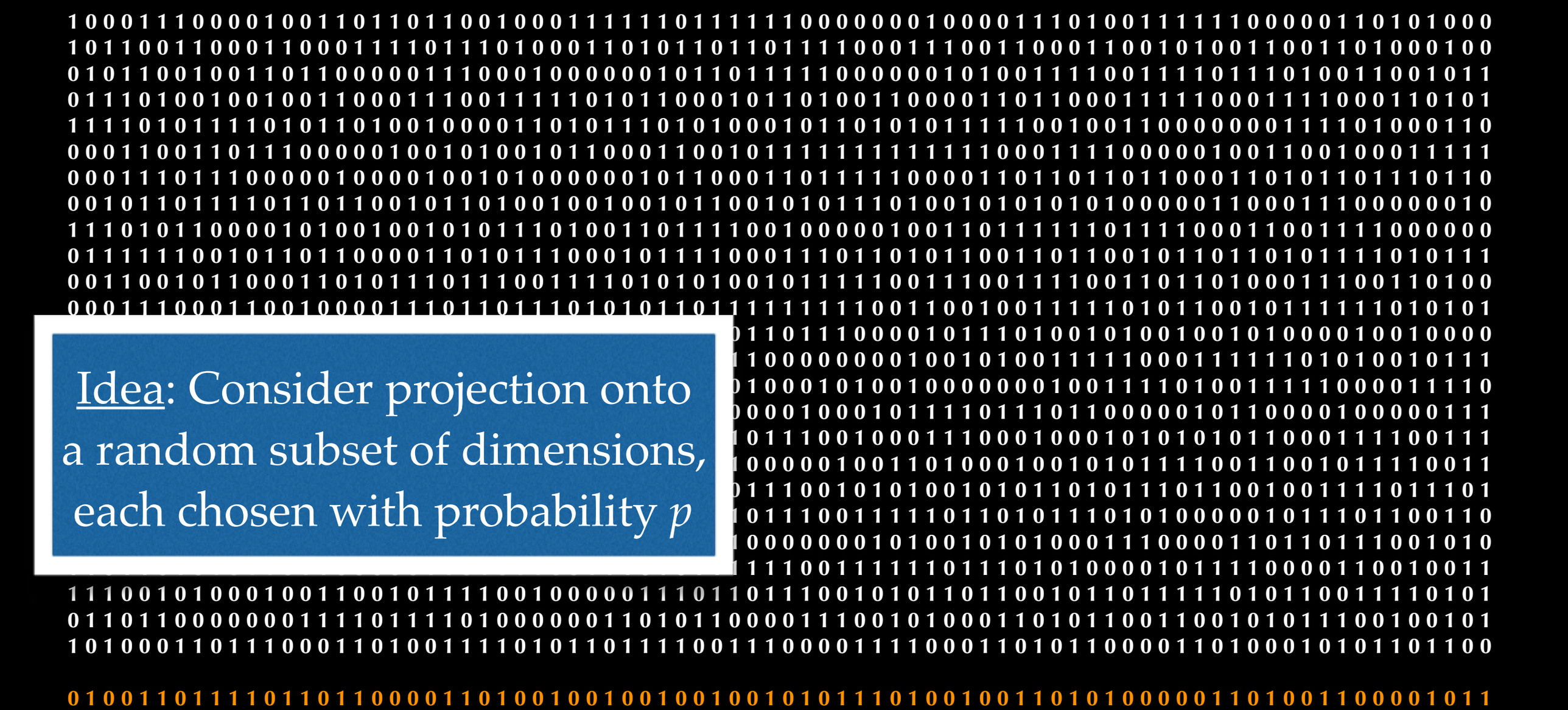
[Indyk & Motwani, STOC '98]:
Time $O(dn^{1/c})$, space $O(n^{1+1/c} + dn)$

[Indyk & Motwani ’98]

Locality-sensitive hashing

[Indyk & Motwani '98]

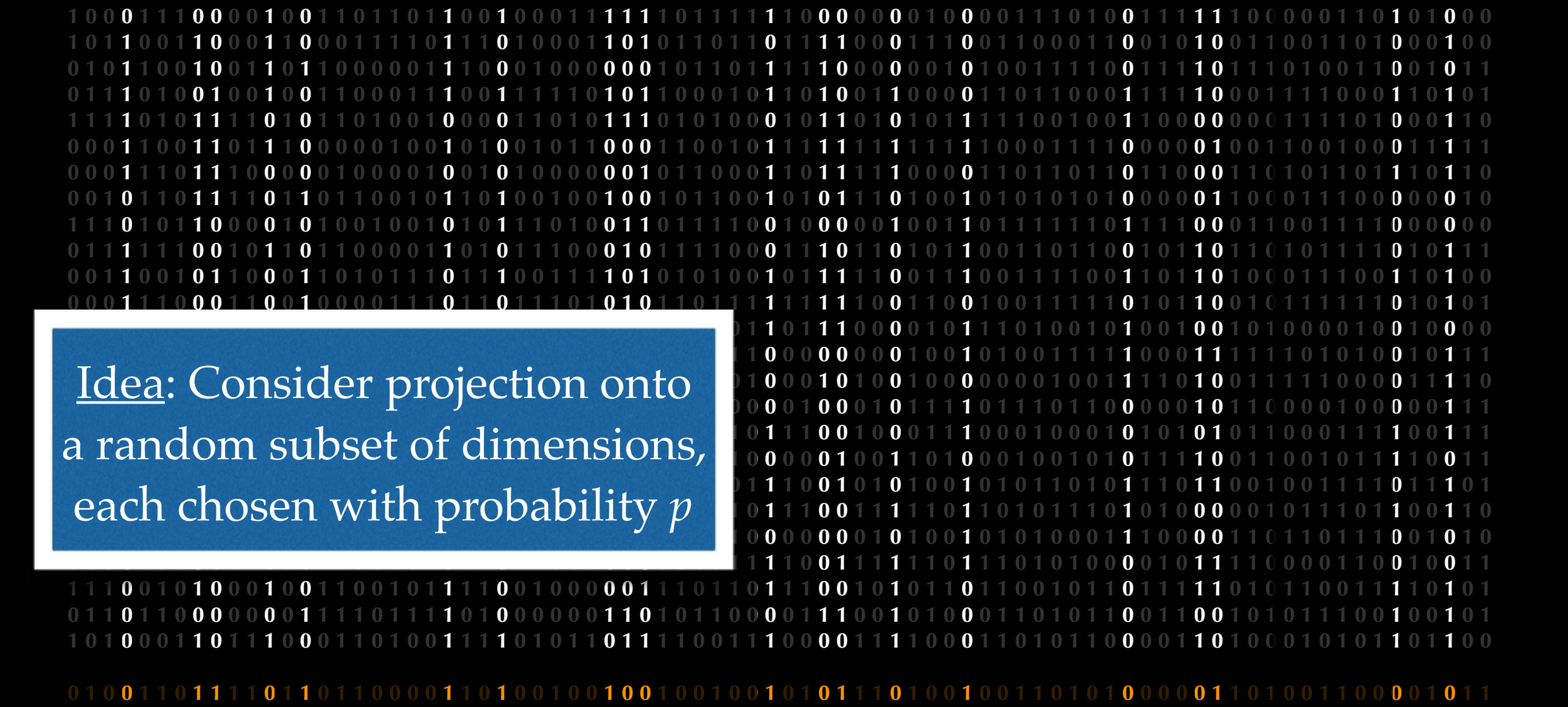
Locality-sensitive hashing



Idea: Consider projection onto a random subset of dimensions, each chosen with probability p

[Indyk & Motwani ’98]

Locality-sensitive hashing



Idea: Consider projection onto a random subset of dimensions, each chosen with probability p

[Indyk & Motwani '98]

Locality-sensitive hashing

$$h(x) = x \wedge a$$

Idea: Consider projection onto a random subset of dimensions, each chosen with probability p

[Indyk & Motwani '98]

Locality-sensitive hashing

$$h(x) = x \wedge a$$

Idea: Consider projection onto a random subset of dimensions, each chosen with probability p

Candidate match

[Indyk & Motwani '98]

Locality-sensitive hashing

$$h_i(x) = x \wedge a_i$$

Idea: Consider projection onto a random subset of dimensions, each chosen with probability p

Candidate match

[Indyk & Motwani '98]

Locality-sensitive hashing

$$h_i(x) = x \wedge a_i$$

Idea: Consider projection onto a random subset of dimensions, each chosen with probability p

Candidate match

$t = n^{1/c}$ repetitions ensure constant success probability

[Indyk & Motwani '98]

Locality-sensitive hashing

$$h_i(x) = x \wedge a_i$$

Idea: Consider projection onto a random subset of dimensions, each chosen with probability p

Candidate match

Possibility of *false negatives*

$$t = n^{1/c}$$

repetitions ensure constant success probability

Some recent developments*

Reference	Exponent, search time	Comment
Linear search	1	
Indyk & Motwani STOC '98	$1/c$	

* Focus on subquadratic space; lower order terms ignored.

Some recent developments*

Reference	Exponent, search time	Comment
Linear search	1	
Indyk & Motwani STOC '98	$1/c$	
Andoni & Razenshteyn STOC '15	$1/(2c-1)$	Data dependent LSH

* Focus on subquadratic space; lower order terms ignored.

Some recent developments*

Reference	Exponent, search time	Comment
Linear search	1	
Indyk & Motwani STOC '98	$1/c$	
Andoni & Razenshteyn STOC '15	$1/(2c\text{-}1)$	Data dependent LSH
Laarhoven arXiv '15	$(2c\text{-}1)/c^2$	Linear space

* Focus on subquadratic space; lower order terms ignored.

Some recent developments*

Reference	Exponent, search time	Comment
Linear search	1	
Indyk & Motwani STOC '98	$1/c$	
Andoni & Razenshteyn STOC '15	$1/(2c\text{-}1)$	Data dependent LSH
Laarhoven arXiv '15	$(2c\text{-}1)/c^2$	Linear space
Alman & Williams FOCS '15	$1-\tilde{\Omega}(\log(n)/d)$	Batched, c=1
Karppa, Kaski & Kohonen SODA '16	$\frac{2\omega-3}{3}$	Batched [c>1, random data, $\omega > 2.25$]

* Focus on subquadratic space; lower order terms ignored.

Recent developments*

Reference	Exponent, search time	Comment
Linear search	1	
Indyk & Motwani STOC '98	$1/c$	
Andoni & Razenshteyn STOC '15	$1/(2c\text{-}1)$	Data dep
Laarhoven arXiv '15	$(2c\text{-}1)/c^2$	
Alman & Williams FOCS '15	$1-\tilde{\Omega}(\log(n)/d)$	ed, c=1
Karppa, Kaski & Kohonen SODA '16	$\frac{2\omega-3}{3}$	Batched , random data, $\omega > 2.25$]

Possibility of *false negatives*

* Focus on subquadratic space; lower order terms ignored.

Similarity search without false negatives

Reference	Exponent, search time	Comment
Arasu, Ganti & Kaushik VLDB '06	$\approx 3/c$ (analysis not in their paper)	
This paper	$\ln(4)/c$	
This paper	$1/c$	For $cr = \log(n)$

[Arasu, Ganti & Kaushik '06]

Basic correlated LSH

partitioning

[Arasu, Ganti & Kaushik '06]

Basic correlated LSH

partitioning

h_1

[Arasu, Ganti & Kaushik '06]

Basic correlated LSH

partitioning

h_2

[Arasu, Ganti & Kaushik ’06]

Basic correlated LSH

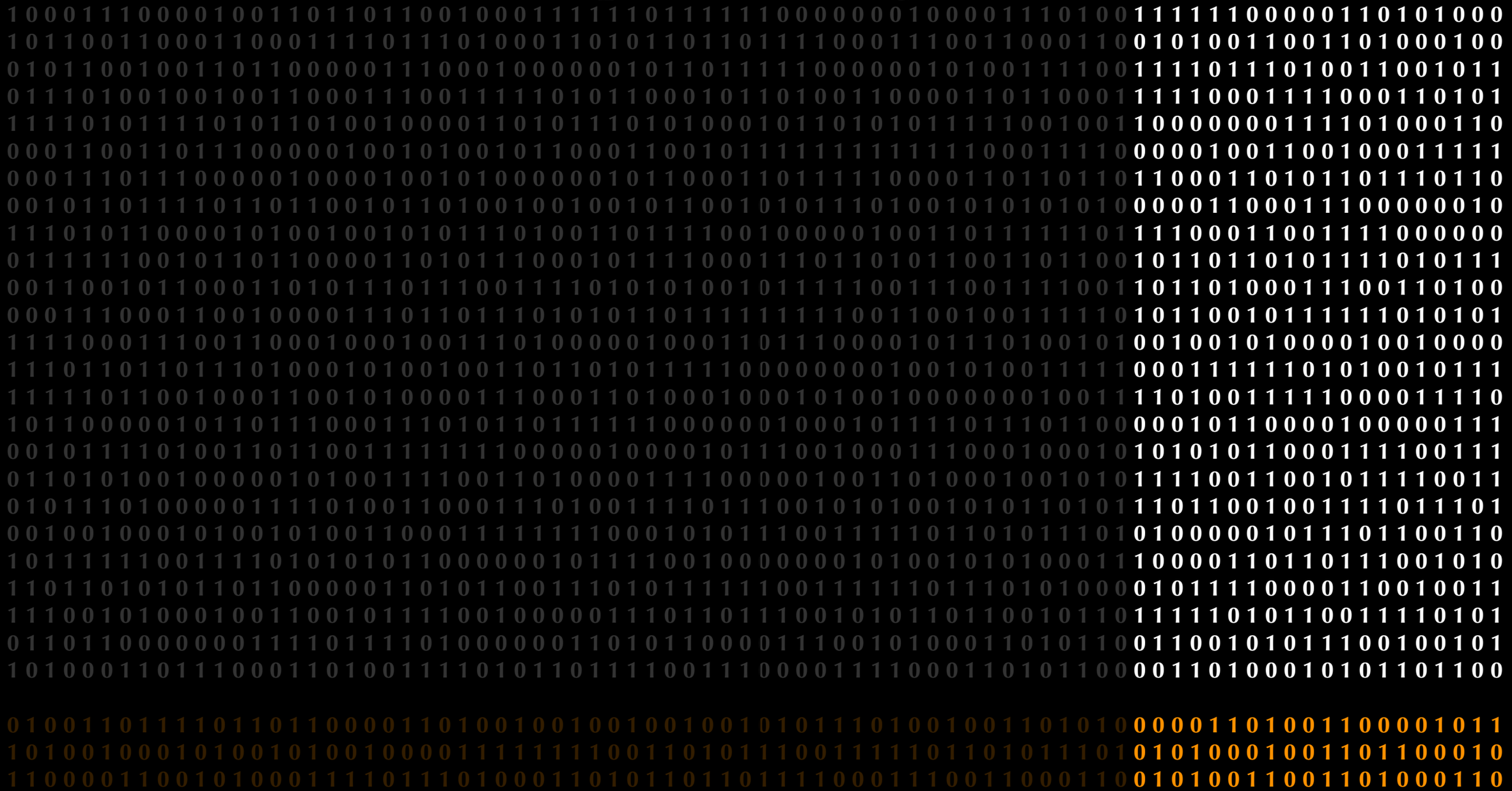
partitioning

h_3

[Arasu, Ganti & Kaushik ’06]

Basic correlated LSH

partitioning



h_4

[Arasu, Ganti & Kaushik '06]

Basic correlated LSH
partitioning

For Hamming distance ≤ 3,
a collision is guaranteed!

[Arasu, Ganti & Kaushik '06]

Basic correlated LSH
partitioning

For Hamming distance ≤ 3,
a collision is guaranteed!

[Arasu et al. '06]:
To bound probability of collision for distance > 3 *randomly permute* the dimensions

[Arasu, Ganti & Kaushik '06]

Basic correlated LSH

enumeration

h_{12}

[Arasu, Ganti & Kaushik '06]

Basic correlated LSH

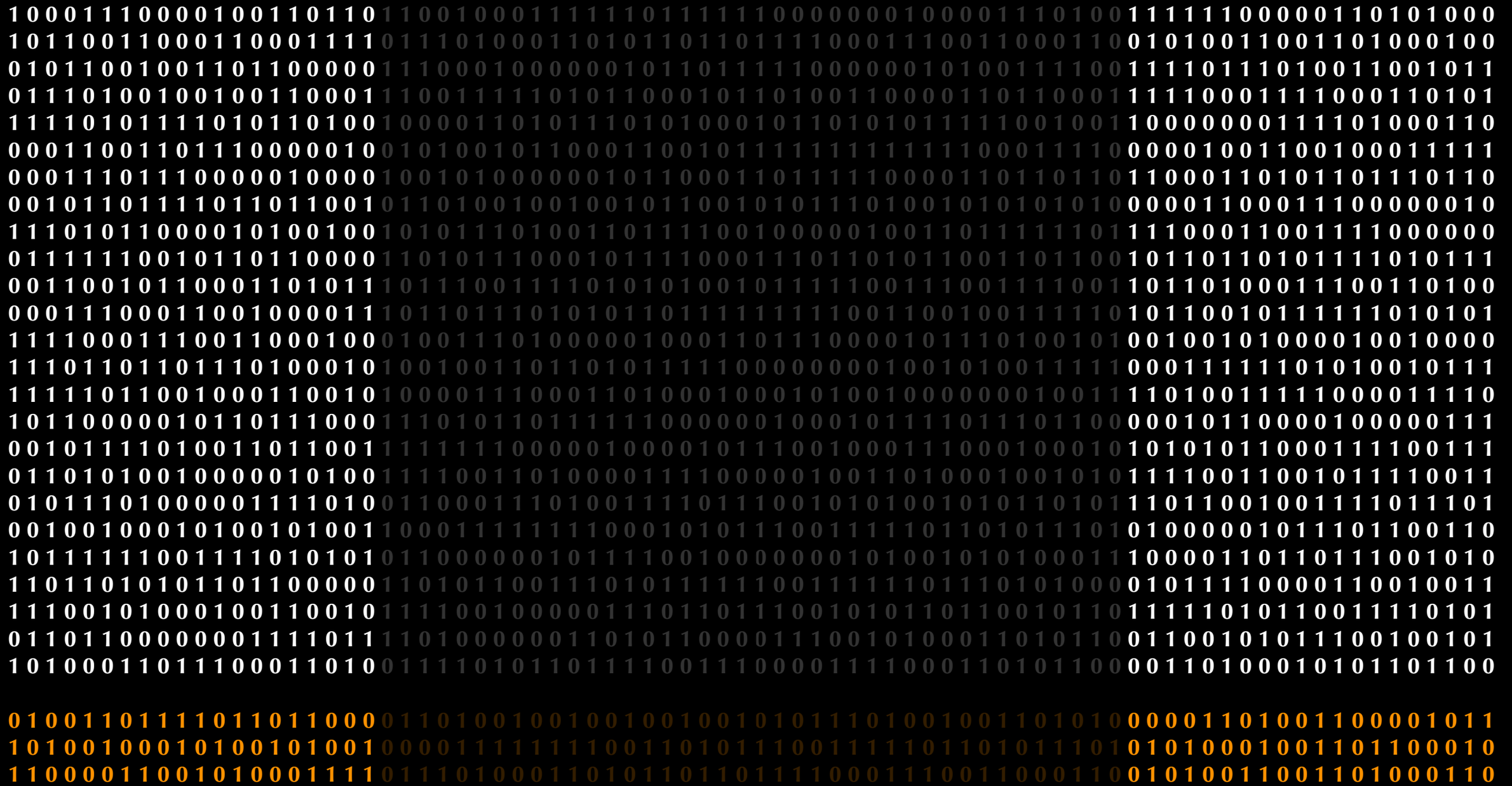
enumeration

h_{13}

[Arasu, Ganti & Kaushik ’06]

Basic correlated LSH

enumeration



h_{14}

[Arasu, Ganti & Kaushik '06]

Basic correlated LSH
enumeration

For Hamming distance ≤ 2,
a collision is guaranteed in

$h_{12}, h_{13}, h_{14}, h_{23}, h_{24}, h_{34}$

h_{14}

[Arasu, Ganti & Kaushik ’06]

Basic correlated LSH
enumeration

For Hamming distance ≤ 2, a collision is guaranteed in

$h_{12}, h_{13}, h_{14}, h_{23}, h_{24}, h_{34}$

Probabilistic argument suggests that it is possible to do much better. But how?

h_{14}

Mathematical question

- A (*p,r*)-***covering*** matrix of dim. *d* satisfies:
  - Has (1-*p*)*d* zeros and *pd* ones in each row;
  - for every set of *r* columns there exists a row with 0s in all of them.

Mathematical question

- A (*p*,*r*)-***covering*** matrix of dim. *d* satisfies:
  - Has (1-*p*)*d* zeros and *pd* ones in each row;
  - for every set of *r* columns there exists a row with 0s in all of them.

Small collision probability

Mathematical question

- A (*p*,*r*)-***covering*** matrix of dim. *d* satisfies:
  - Has (1-*p*)*d* zeros and *pd* ones in each row;
  - for every set of *r* columns there exists a row with 0s in all of them.

Small collision probability

Collision guarantee

Mathematical question

- A (*p*,*r*)-***covering*** matrix of dim. *d* satisfies:
  - Has (1-*p*)*d* zeros and *pd* ones in each row;
  - for every set of *r* columns there exists a row with 0s in all of them.

Small collision probability

Collision guarantee

- <u>Question</u>:
  How few rows can such a matrix have?

Number of hash functions

Mathematical question

- A (*p*,*r*)-***covering*** matrix of dim. *d* satisfies:
  - Has (1-*p*)*d* zeros and *pd* ones in each row;
  - for every set of *r* columns there exists a row with 0s in all of them.

Small collision probability

Collision guarantee

- <u>Question</u>:
  How few rows can such a matrix have?

Number of hash functions

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$p = 4/7$
$r = 2$

Mathematical question

- A (p,r)-***covering*** matrix of dim. d satisfies:
  - Has $(1-p)d$ zeros and pd ones in each row;
  - for every set of r columns there exists a row with 0s in all of them.

Small collision probability

Collision guarantee

- <u>Question</u>:
  How few rows can such a matrix have?

Number of hash functions

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$p = 4/7$
$r = 2$

Mathematical question

- A (p,r)-***covering*** matrix of dim. d satisfies:
  - Has $(1-p)d$ zeros and pd ones in each row;
  - for every set of r columns there exists a row with 0s in all of them.

Small collision probability

Collision guarantee

- Question:
  How few rows can such a matrix have?

Number of hash functions

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$p = 4/7$

$r = 2$

Mathematical question

- A (p,r)-***covering*** matrix of dim. d satisfies:
  - Has $(1-p)d$ zeros and pd ones in each row;
  - for every set of r columns there exists a row with 0s in all of them.

Small collision probability

Collision guarantee

- <u>Question</u>:
  How few rows can such a matrix have?

Number of hash functions

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$p = 4/7$
$r = 2$

In the mathematics literature such a construction is referred to as a *covering design*

- A (p,r)-***covering*** matrix of dim. d satisfies:
  - Has $(1-p)d$ zeros and pd ones in each row;
  - for every set of r columns there exists a row with 0s in all of them.

Small collision probability

Collision guarantee

- Question:
  How few rows can such a matrix have?

Number of hash functions

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$p = 4/7$

$r = 2$

[Gordon, Patashnik, Kuperberg '95]

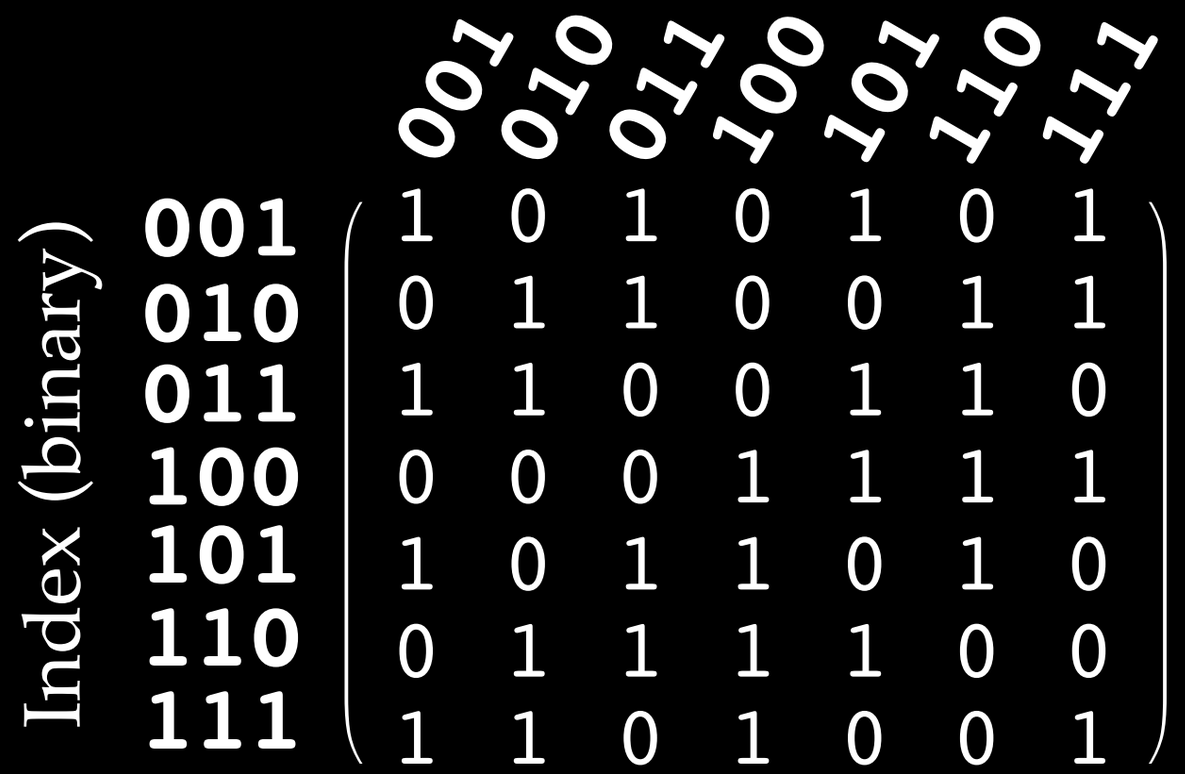
A covering design

- Parameters: $d = 2^{r+1}\text{-}1$, $pd = 2^r\text{+}1 \approx d/2$

[Gordon, Patashnik, Kuperberg '95]

A covering design

- Parameters: $d = 2^{r+1}\text{-}1$, $pd = 2^r\text{+}1 \approx d/2$



[Gordon, Patashnik, Kuperberg '95]

A covering design

- Parameters: $d = 2^{r+1}\text{-}1$, $pd = 2^r\text{+}1 \approx d/2$

Idea: Entry is dot product (mod 2) of row / column ID vectors (Hadamard code)

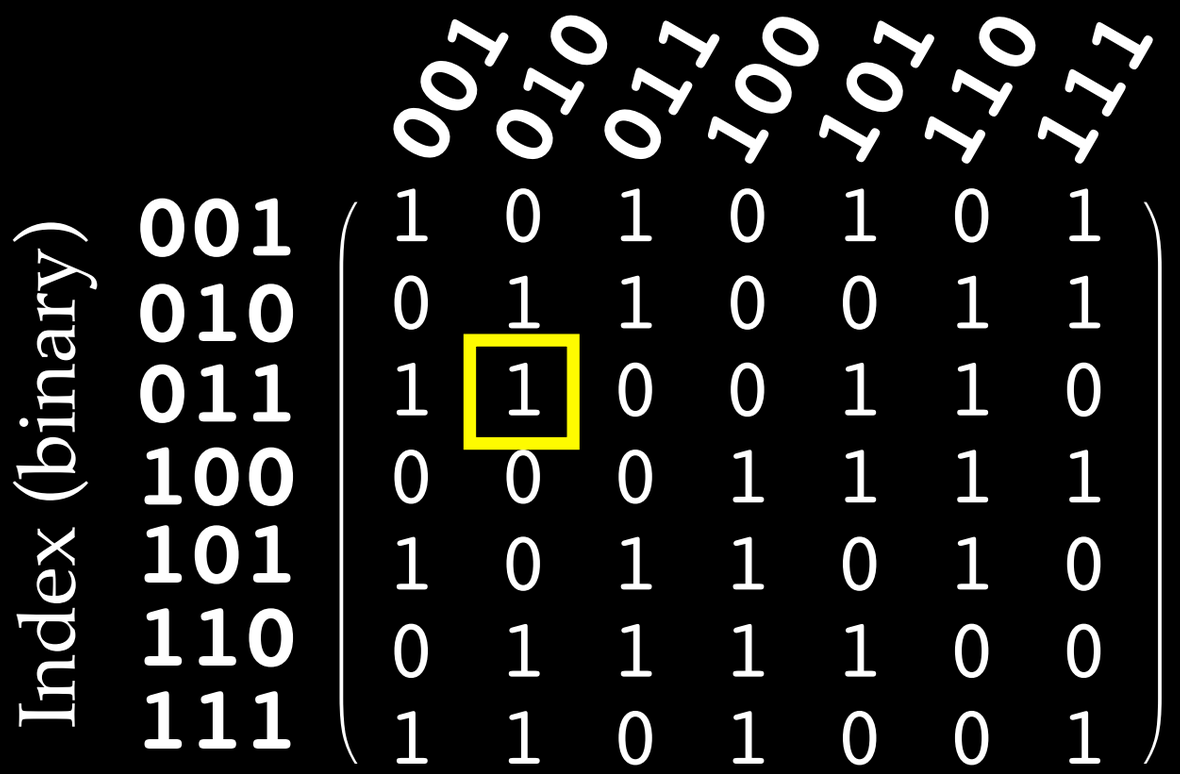
Index (binary)	001	010	011	100	101	110	111
001	1	0	1	0	1	0	1
010	0	1	1	0	0	1	1
011	1	1	0	0	1	1	0
100	0	0	0	1	1	1	1
101	1	0	1	1	0	1	0
110	0	1	1	1	1	0	0
111	1	1	0	1	0	0	1

[Gordon, Patashnik, Kuperberg '95]

A covering design

- Parameters: $d = 2^{r+1}-1$, $pd = 2^r+1 \approx d/2$

Idea: Entry is dot product (mod 2) of row / column ID vectors (Hadamard code)



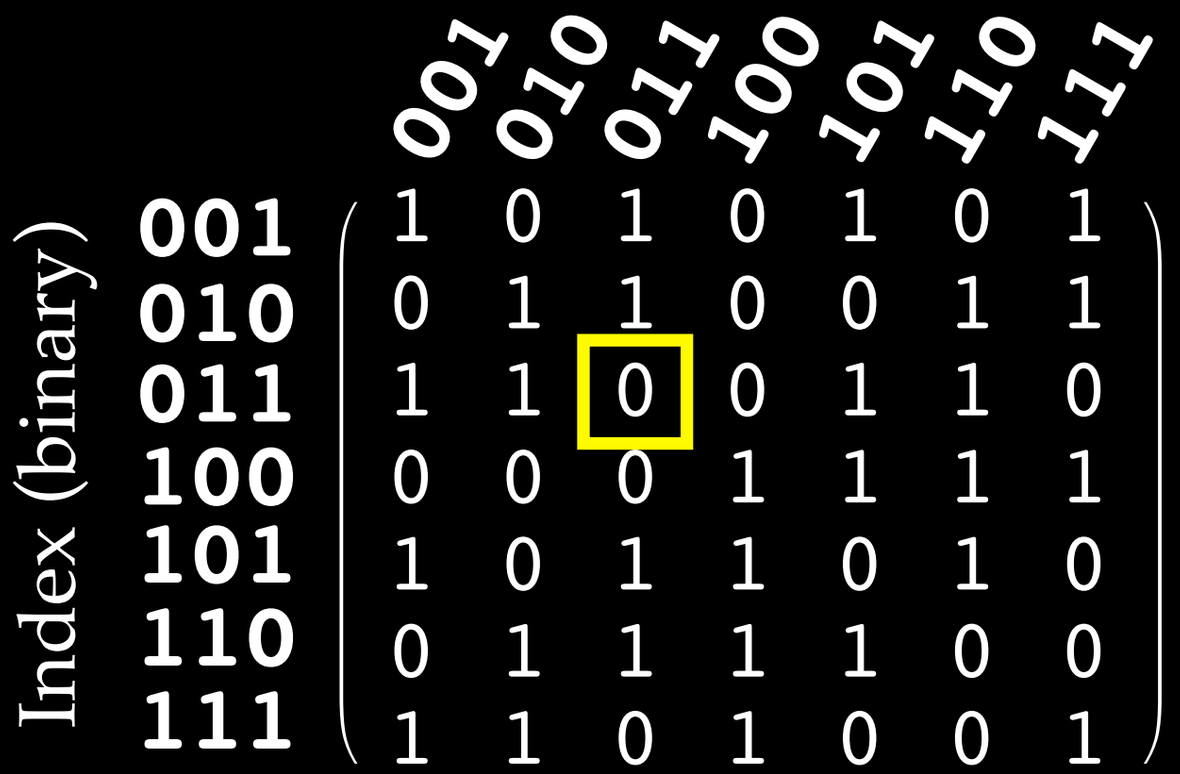
$(0,1,1) \cdot (0,1,0)=1$

[Gordon, Patashnik, Kuperberg '95]

A covering design

- Parameters: $d = 2^{r+1}-1$, $pd = 2^r+1 \approx d/2$

Idea: Entry is dot product (mod 2) of row / column ID vectors (Hadamard code)



$(0,1,1) \cdot (0,1,1)=0$

[Gordon, Patashnik, Kuperberg '95]

A covering design

- Parameters: $d = 2^{r+1}\text{-}1$, $pd = 2^r + 1 \approx d/2$

Idea: Entry is dot product (mod 2) of row / column ID vectors (Hadamard code)

Index (binary)	001	010	011	100	101	110	111
001	1	0	1	0	1	0	1
010	0	1	1	0	0	1	1
011	1	1	0	0	1	1	0
100	0	0	0	1	1	1	1
101	1	0	1	1	0	1	0
110	0	1	1	1	1	0	0
111	1	1	0	1	0	0	1

[Gordon, Patashnik, Kuperberg '95]

A covering design

- Parameters: $d = 2^{r+1}-1$, $pd = 2^r+1 \approx d/2$

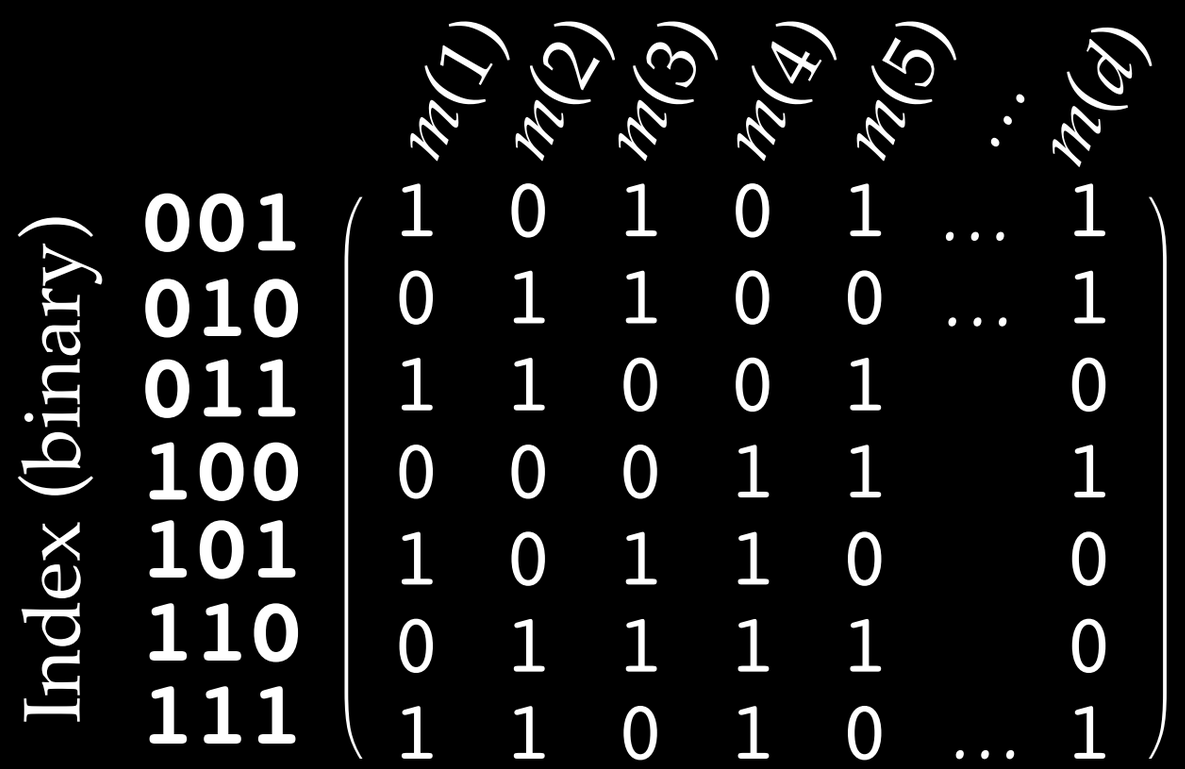
Idea: Entry is dot product (mod 2) of row / column ID vectors (Hadamard code)

Lemma:
For every set of r vectors in $\{0,1\}^{r+1}$ there exists a nonzero vector that is orthogonal to all

Sampling the covering design

- Parameters: $d = 2^{r+1}\text{-}1$, $pd = 2^r + 1 \approx d/2$

Idea: Choose columns of covering design using a hash function m: $\{1,\ldots,d\} \longrightarrow \{0,1\}^{r+1}$.



Sampling the covering design

- Parameters: $d = 2^{r+1}\text{-}1$, $pd = 2^r + 1 \approx d/2$

Idea: Choose columns of covering design using a hash function m: $\{1,\ldots,d\} \longrightarrow \{0,1\}^{r+1}$.

Index (binary)	$m(1)$	$m(2)$	$m(3)$	$m(4)$	$m(5)$	$\ldots$	$m(d)$
001	1	0	1	0	1	…	1
010	0	1	1	0	0	…	1
011	1	1	0	0	1		0
100	0	0	0	1	1		1
101	1	0	1	1	0		0
110	0	1	1	1	1		0
111	1	1	0	1	0	…	1

Lemma:

$\Pr[x \wedge a_i = y \wedge a_i] \leq 2^{-||x-y||}$

Implementation in Python

```
def buildCovering(d,r):
    for v in xrange(1,2**(r+1)): A[v] = 0
    for i in xrange(d):
        m = randint(1, 2**(r+1)-1)
        for v in xrange(1,2**(r+1)):
            A[v] = A[v] + (1<<i) * (popcnt(m & v) % 2)

def buildDataStructure(S,r):
    D = {}
    for x in S:
        for v in xrange(1,2**(r+1)):
            if not (x & A[v]) in D: D[x & A[v]] = Set()
            D[x & A[v]].add(x)
    return D

def nearestNeighbor(D,r,y):
    best, nn = infinity, None
    for v in xrange(1,2**(r+1)):
        if (y & A[v]) in D:
            for x in D[y & A[v]]:
                if dist(x,y) < best:
                    best, nn = dist(x,y), x
        if best <= floor(log(v+1,2)): return nn
    return None
```

Too many collisions?

- Can map vectors from $\{0,1\}^d$ to $\{0,1\}^{td}$, increasing all distances by an integer factor t.

$q^t =$ 110**0**10111**0**10111**0**10111**0**101

$x^t =$ 110**1**10111**1**10111**1**10111**1**101

- Try to "hit" sweet spot $tr = \log(n)/c$

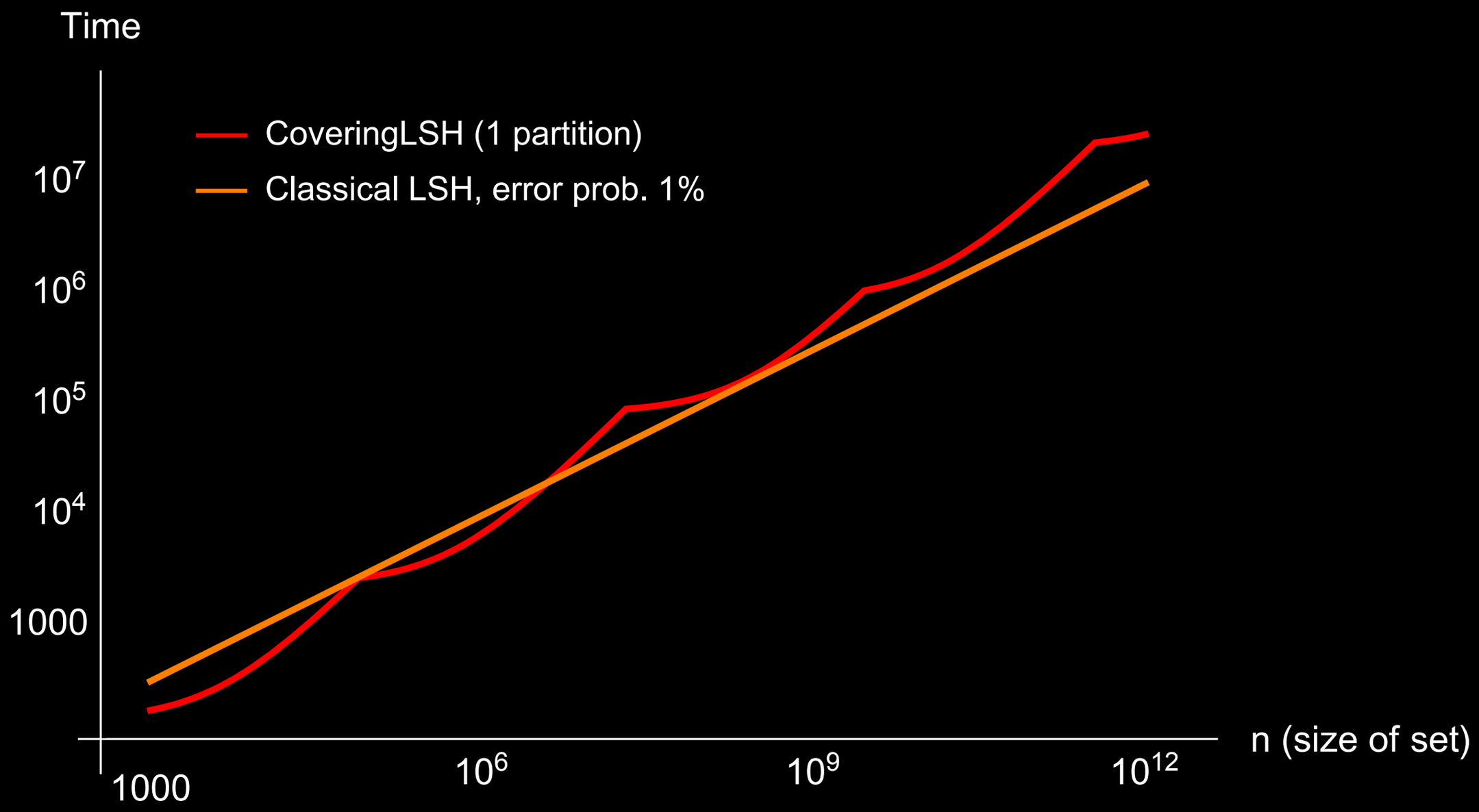
Too many collisions?

- Can map vectors from $\{0,1\}^d$ to $\{0,1\}^{td}$, increasing all distances by an integer factor t.

- For $cr=o(\log n/\log\log n)$, not in SODA version:
  - Use mod p (rather than mod 2) linear algebra to produce denser coverings.
  - Matches Indyk-Matwani up to polylog(n) factor.

Example

r=6, c=2



Time 2^r too much?

- Partition dimensions (randomly):

Time 2^r too much?

- Partition dimensions (randomly):

- Distance in *some* part will be $\leq r/s$
  - Use CoveringLSH with radius r/s on each part

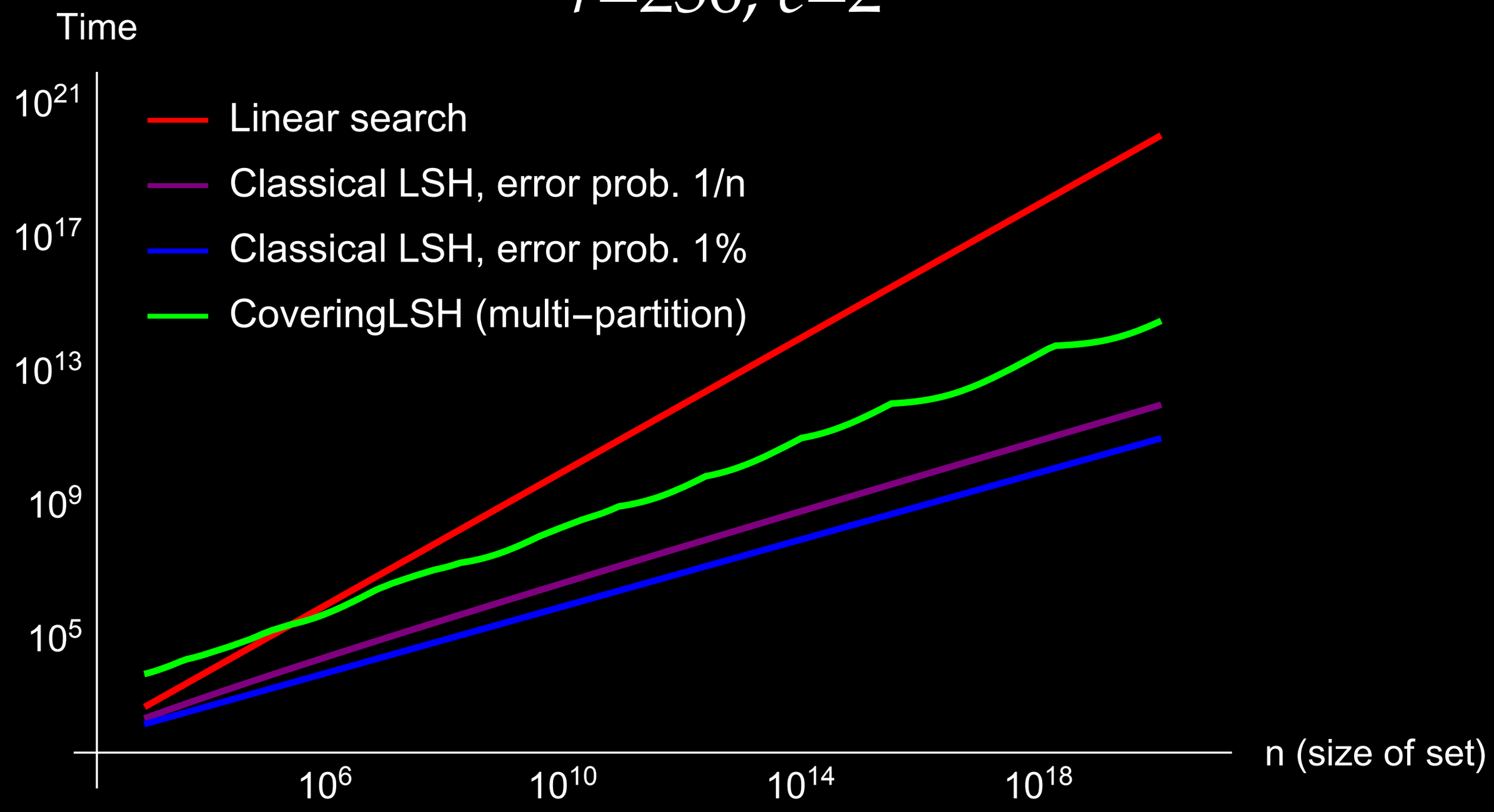
$$\begin{pmatrix}
1&0&1&0&1&0&1&0&0&0&0&0&0&0&0\\
0&1&1&0&0&1&1&0&0&0&0&0&0&0&0\\
1&1&0&0&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&1&1&1&1&0&0&0&0&0&0&0&0\\
1&0&1&1&0&1&0&0&0&0&0&0&0&0&0\\
0&1&1&1&1&0&0&0&0&0&0&0&0&0&0\\
1&1&0&1&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&1&0&1&0&1\\
0&0&0&0&0&0&0&0&1&1&0&0&1&1\\
0&0&0&0&0&0&0&1&1&0&0&1&1&0\\
0&0&0&0&0&0&0&0&0&0&1&1&1&1\\
0&0&0&0&0&0&0&1&0&1&1&0&1&0\\
0&0&0&0&0&0&0&0&1&1&1&1&0&0\\
0&0&0&0&0&0&0&1&1&0&1&0&0&1
\end{pmatrix}$$

$s = 2$

$r = 5$

Example

r=256, c=2



Some open questions

Match performance of classical (or data dep.) LSH without false neg.?

Some open questions

Match performance of classical (or data dep.) LSH without false neg.?

Linear space near neighbor search without false negatives: Matching known bounds?

Some open questions

Match performance of classical (or data dep.) LSH without false neg.?

Linear space near neighbor search without false negatives: Matching known bounds?

Conditional lower bounds for *approximate* similarity search? (Based on SETH, 3SUM,…)

Thank you for your attention!

Acknowledgement of economic support:

European Research Council

Established by the European Commission