

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 $\|x - q\| = \text{Hamming distance between } x \text{ and } q.$

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 $\|x - q\| = \text{Hamming distance between } x \text{ and } q$.

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

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

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

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

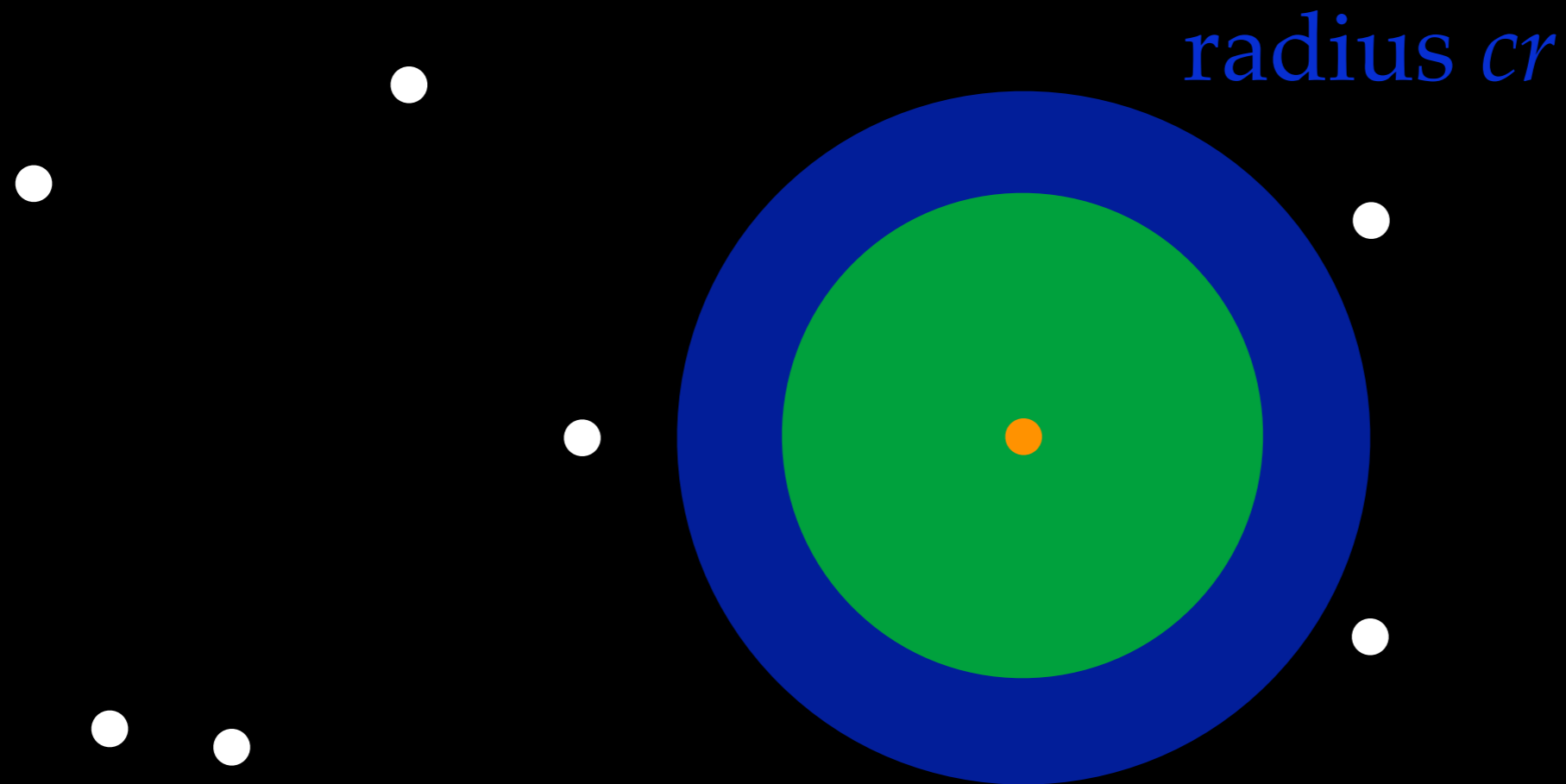
where $\|x - q\| = \text{Hamming distance between } x \text{ and } q$.

- [Williams '04], [Alman & Williams '15]: **Strong ETH states: Not possible!**

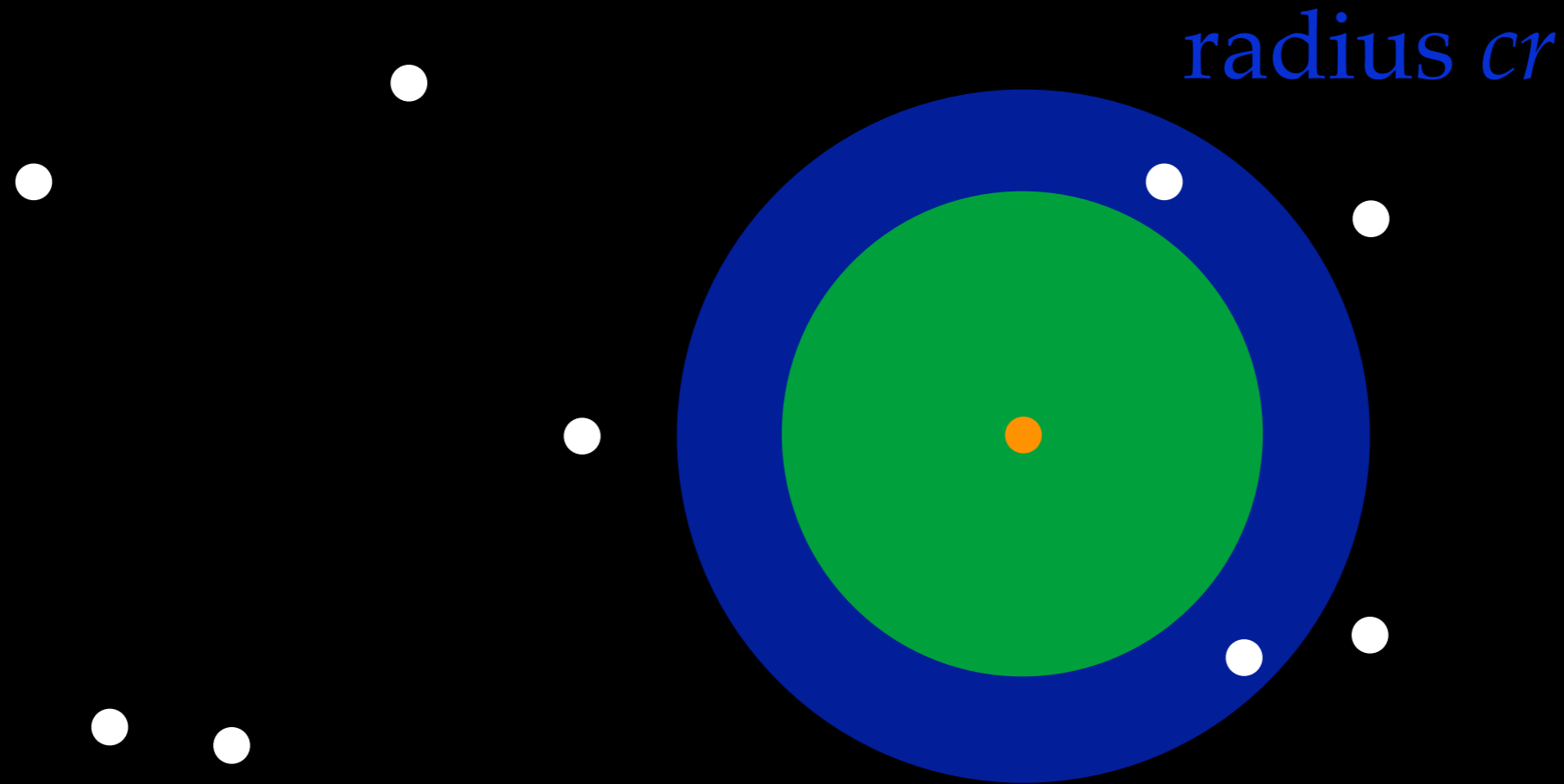
Hamming similarity search in time $n^{0.99} 2^{o(d)} \Rightarrow$

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

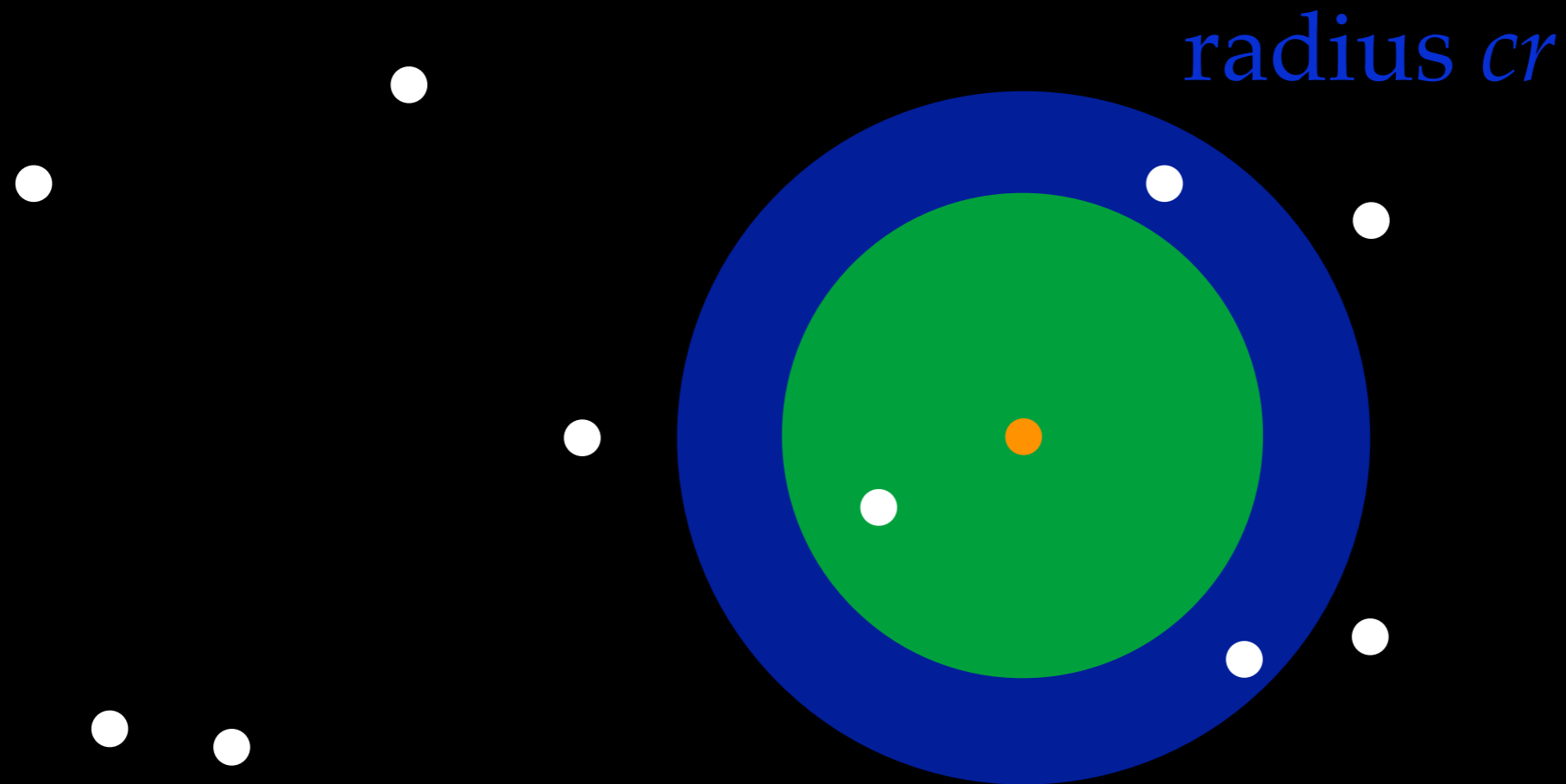
Approximate similarity search



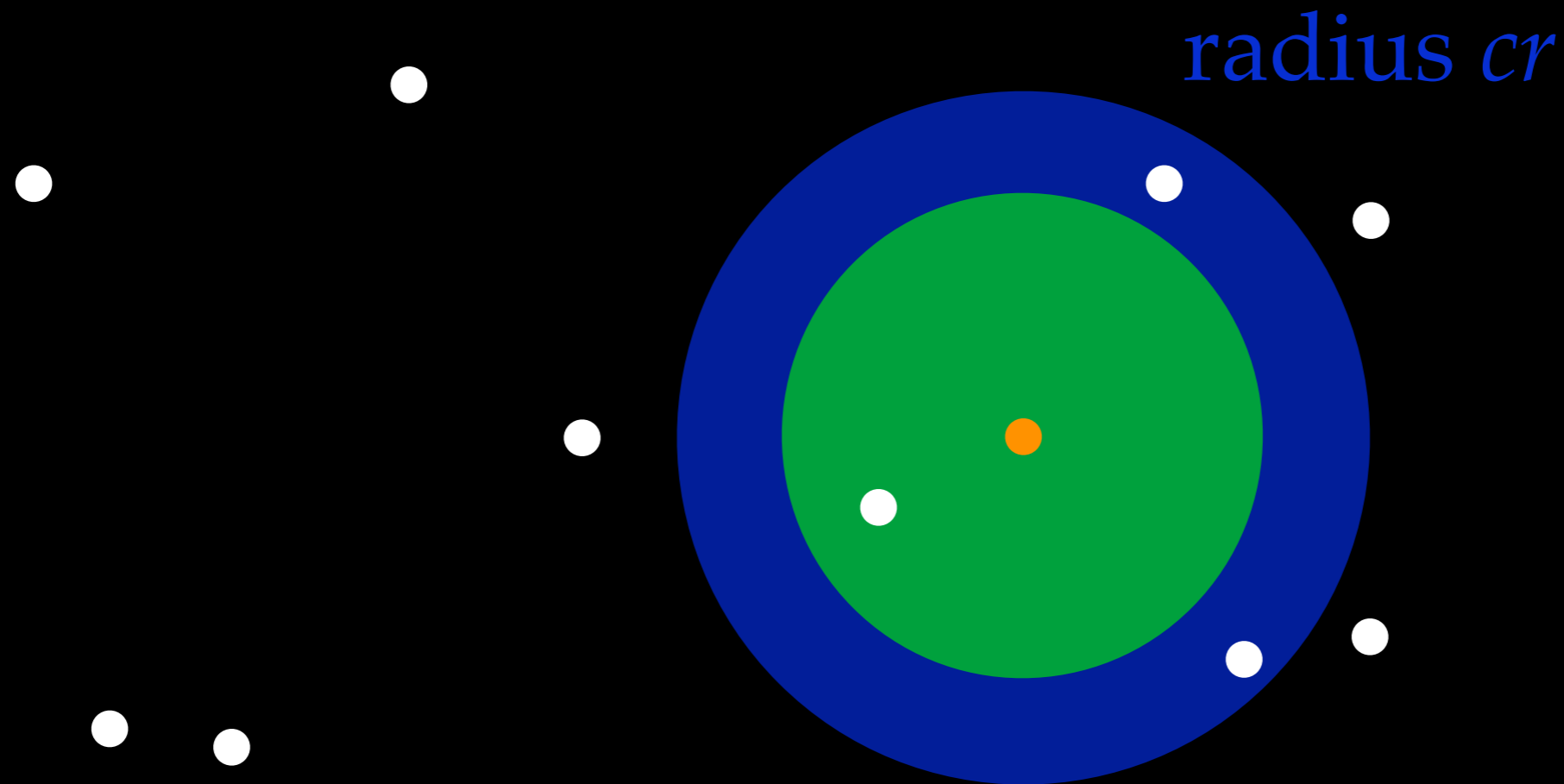
Approximate similarity search



Approximate similarity search



Approximate similarity search



[Indyk & Motwani, STOC '98]:

Time $O(dn^{1/c})$, space $O(n^{1+1/c} + dn)$

Locality-sensitive hashing

1000111000010011011011001000111111011111100000010000111010011111100000110101000
10110011000110001111011101000110101101101111000111001100011001010011001101000100
01011001001101100000111000100000010110111110000001010011110011110111010011001011
01110100100100110001110011111010110001011010011000011011000111110001111000110101
11110101111010110100100001101011101010001011010101111100100110000000111101000110
0001100110111000001001010010110001100101111111111110001111000001001100100011111
00011101110000010000100101000000101100011011111000011011011011000110101101110110
00101101111011011001011010010010010110010101110100101010101000001100011100000010
11101011000010100100101011101001101111001000001001101111110111100011001111000000
01111110010110110000110101110001011110001110110101100110110010110110101111010111
001100101100011010111101110011110101010010111110011100111100110110100011100110100
000111000110010000111011011110101011011111111100110010011111010110010111111010101
11110001110011000100010011101000001000110111000010111010010100100101000010010000
11101101101110100010100100110110101111100000000100101001111100011111101010010111
11111011001000110010100001110001101000100010100100000001001111010011111000011110
10110000010110111000111010110111111000000100010111101110110000010110000100000111
001011110100110110011111111100000100001011100100011100010001010101011000111100111
01101010010000010100111100110100001111000001001101000100101011110011001011110011
01011101000001111010011000111010011110111001010100101011010111011001001111011101
00100100010100101001100011111111000101011100111110110101110101000001011101100110
10111111001111010101011000000101111001000000010100101010001110000110110111001010
11011010101101100000110101100111010111111001111110111010100001011110000110010011
11100101000100110010111100100000111011011100101011011001011011111010110011110101
01101100000001111011110100000011010110000111001010001101011001100101011100100101
10100011011100011010011110101101111001110000111100011010110000110100010101101100

01001101111011011000011010010010010010010010101110100100110101000001101001100001011

Locality-sensitive hashing

1000111000010011011011001000111111011111100000010000111010011111100000110101000
10110011000110001111011101000110101101101111000111001100011001010011001101000100
01011001001101100000111000100000010110111110000001010011110011110111010011001011
01110100100100110001110011111010110001011010011000011011000111110001111000110101
11110101111010110100100001101011101010001011010101111100100110000000111101000110
0001100110111000001001010010110001100101111111111110001111000001001100100011111
00011101110000010000100101000000101100011011111000011011011011000110101101110110
00101101111011011001011010010010010110010101110100101010101000001100011100000010
11101011000010100100101011101001101111001000001001101111110111100011001111000000
01111110010110110000110101110001011110001110110101100110110010110110101111010111
00110010110001101011101110011110101010010111110011100111100110110100011100110100
0001110001100100001110110111010101101111111100110010011111010110010111111010101

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

011011100001011011000011010111000101111000111011010110011011001011011010111101011
000100010111101110110000010110000100000111
011100100011100010001010101011000111100111
000001001101000100101011110011001011110011
0111001010100101011010111011001001111011101
011100111110110101110101000001011101100110
000000010100101010001110000110110111001010
111001111110111010100001011110000110010011

11100101000100110010111100100000111011011100101011011001011011111010110011110101
01101100000001111011110100000011010110000111001010001101011001100101011100100101
10100011011100011010011110101101111001110000111100011010110000110100010101101100

01001101111011011000011010010010010010010101110100100110101000001101001100001011

Locality-sensitive hashing

1000111000010011011011001000111111011111100000010000111010011111100000110101000
10110011000110001111011101000110101101101111000111001100011001010011001101000100
01011001001101100000111000100000010110111110000001010011110011110111010011001011
01110100100100110001110011111010110001011010011000011011000111110001111000110101
111101011110101101001000011010111101010001011010101111100100110000000111101000110
000110011011100000100101001011000110010111111111110001111000001001100100011111
00011101110000010000100101000000101100011011111000011011011011000110101101110110
00101101111011011001011010010010010110010101110100101010101000001100011100000010
11101011000010100100101011101001101111001000001001101111110111100011001111000000
011111100101101100001101011100010111100011101101011001101100101101101101111010111
001100101100011010111101110011110101010010111110011100111100110110100011100110100
0001110001100100001110110111010101101111111001100100111110101100101111111010101
011011100001011101001010000
100000000100101001111100011111101010010111
100010100100000001001111010011111000011110
000100010111101110110000010110000100000111
011100100011100010001010101011000111100111
000001001101000100101011110011001011110011
111001010100101011010111011001001111011101
011100111110110101110101000001011101100110
000000010100101010001110000110110111001010
111001111110111010100001011110000110010011
111001010001001100101111001011011111010101
0110110000001111011110100000011010110000111001010001101011100100101
101000110111000110100111110101101111001110000111100011010110000110100010101101100
01001101111011011000011010010010010010010101110100100110101000001101001100001011

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

Locality-sensitive hashing

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

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

```

100011100001001101101100100
101100110001100011110111010
010110010011011000001110001
011101001001001100011100111
111101011110101101001000011
00011001101110000010010100101111111111110001111000001001100100011111
00011101110000010000100101000000101100011011111000011011011011000110101101110110
00101101111011011001011010010010010110010101110100101010101000001100011100000010
11101011000010100100101011101001101111001000001001101111110111100011001111000000
011111100101101100001101011100010111100011101101011001101100101101101101111010111
001100101100011010111101110011110101010010111110011100111100110110100011100110100
0001110001100100001110110111010101101111111001100100111110101100101111111010101
0110111000010111010010100100100100100001001000010010000
100000000100101001111100011111101010010111
010010100100000001001111010011111000011110
000100010111101110110000010110000100000111
011100100011100010001010101011000111100111
000001001101000100101011110011001011110011
111001010100101011010111011001001111011101
011100111110110101110101000001011101100110
000000010100101010001110000110110111001010
111001111110111010100001011110000110010011
1110010100010011001011110010000011101101101011111010110011110101
0110110000001111011110100000011010110000111001010001101011001100101011100100101
101000110111000110100111110101101111001110000111100011010110000110100010101101100
01001101111011011000011010010010010010010101110100100110101000001101001100001011

```

Locality-sensitive hashing

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

100011100001001101101100100
101100110001100011110111010
010110010011011000001110001
011101001001001100011100111
111101011110101101001000011
00011001101110000010010100101111111111110001111000001001100100011111
00011101110000010000100101000000101100011011111000011011011011000110101101110110
00101101111011011001011010010010010110010101110100101010101000001100011100000010
11101011000010100100101011101001101111000010011011111101111000110011111000000
0111111001011011000011010111000101111001101101011001101101101101101111010111
0011001011000110101110111001111010101110011100111001101101000111001101100
00011100011001000011101101110101011100110011111010110101101011011010110101
11000010111010010100100100100100001001000010010000
00000100101001111100011111101010010111
10100100000001001111010011111000011110
0001011110111011000010110000100000111
001000111000100010101011000111100111
01001101000100101011110011001011110011
01010100101011010111011001001111011101
00111110110101110101000001011101100110
00010100101010001110000110110111001010
01111110111010100001011110000110010011
001011011001011011111010110011110101
0010100011010110011001100100101011100101
0001111000110101100001101000110100101100

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

Candidate match

01001101111011011000011010010010010010010101110100100110101000001101001100001011

Locality-sensitive hashing

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

100011100001001101101100100
101100110001100011110111010
010110010011011000001110001
011101001001001100011100111
111101011110101101001000011
00011001101110000010010100101111111111110001111000001001100100011111
00011101110000010000100101000000101100011011111000011011011011000110101101110110
00101101111011011001011010010010010110010101110100101010101000001100011100000010
11101011000010100100101011101001101111000010011011111101111000110011111000000
01111110010110110000110101110001011110011011011001101101101101101111010111
0011001011000110101110111001111010101110011100111001101101000111001101100
0001110001100100001110110111010101110011001111101011010110101101101010101
110000101110100101001001001001000010010000
00000100101001111100011111101010010111
10100100000001001111010011111000011110
0001011110111011000010110000100000111
001000111000100010101011000111100111
01001101000100101011110011001011110011
01010100101011010111011001001111011101
00111110110101110101000001011101100110
00010100101010001110000110110111001010
01111110111010100001011110000110010011
00101011011001011011111010110011110101
00101000110101100110011001100101011100101
0001111000110101100001101000110100101100

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

Candidate match

01001101111011011000011010010010010010010101110100100110101000001101001100001011

Locality-sensitive hashing

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

100011100001001101101100100
 101100110001100011110111010
 010110010011011000001110001
 011101001001001100011100111
 111101011110101101001000011
 00011001101110000010010100101111111111110001111000001001100100011111
 00011101110000010000100101000000101100011011111000011011011011000110101101110110
 00101101111011011001011010010010010110010101110100101010101000001100011100000010
 111010110000101001001010111010011011110000100110111110111100011001111000000
 0111111001011011000011010111000101111001101101011001101100101101101101111010111
 0011001011000110101110111001111010101110011100111100110110100011100110100
 0001110001100100001110110111010101110011001001

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

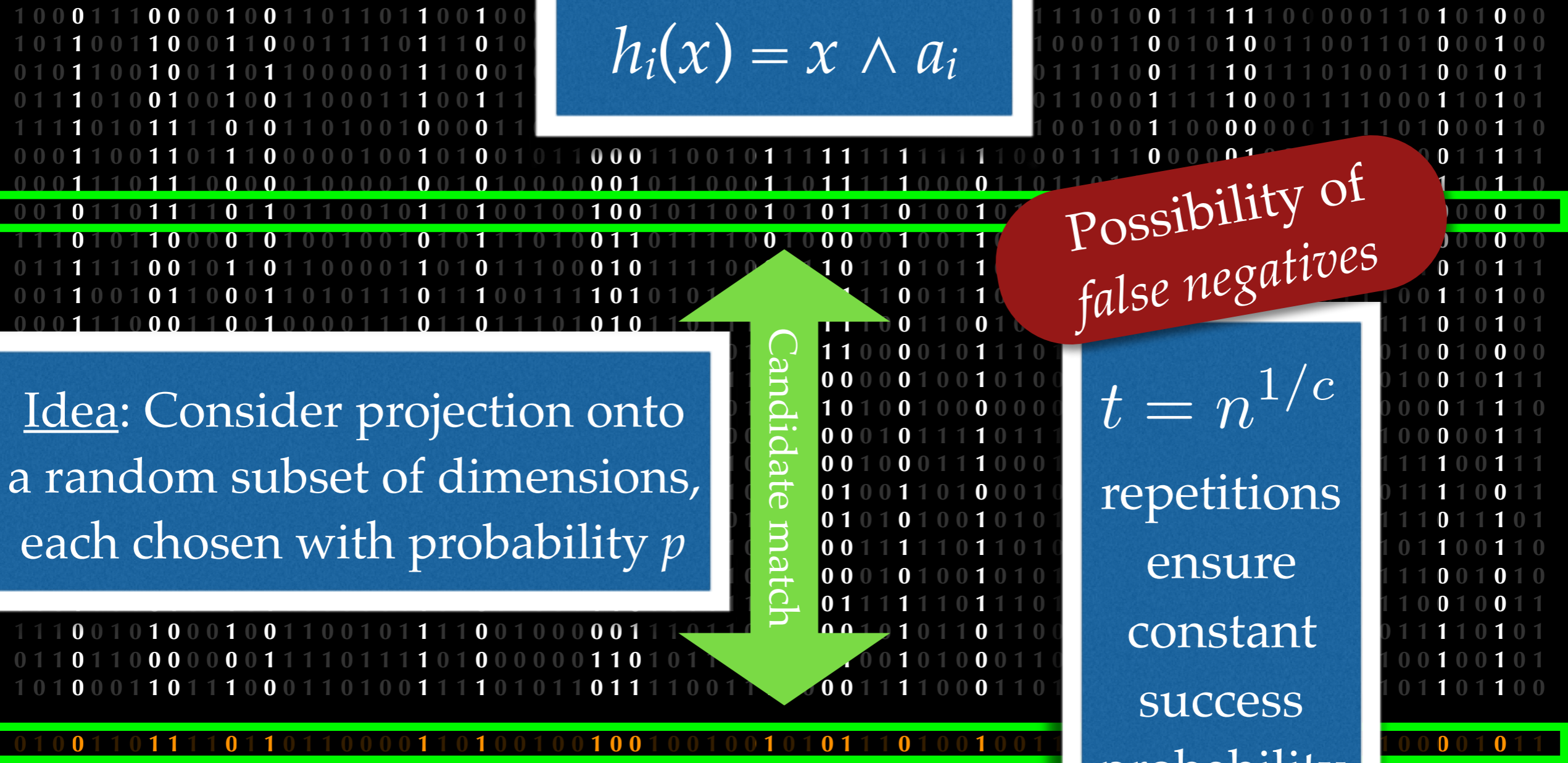
Candidate match

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

111001010001001100101111001000001110111
 011011000000111101111010000001101011
 1010001101110001101001111101011011110011
 0100110111101101100001101001001001
 000001011

Locality-sensitive hashing

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



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

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	$1/(2c-1)$	Data dependent LSH
Laarhoven arXiv '15	$(2c-1)/c^2$	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	$1/(2c-1)$	Data dependent LSH
Laarhoven arXiv '15	$(2c-1)/c^2$	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	$1/(2c-1)$	Data dependent
Laarhoven arXiv '15	$(2c-1)/c^2$	
Alman & Williams FOCS '15	$1 - \tilde{\Omega}(\log(n)/d)$	Batched, $c=1$
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 $cr = \log(n)$

Basic correlated LSH partitioning

10001110000100110110110010001111110111111000000010000111010011111100000110101000
10110011000110001111011101000110101101101111000111001100011001010011001101000100
01011001001101100000111000100000010110111110000001010011110011110111010011001011
01110100100100110001110011111010110001011010011000011011000111110001111000110101
11110101111010110100100001101011101010001011010101111100100110000000111101000110
0001100110111000001001010010110001100101111111111110001111000001001100100011111
00011101110000010000100101000000101100011011111000011011011011000110101101110110
00101101111011011001011010010010010110010101110100101010101000001100011100000010
1110101100001010010010101110100110111100100000100110111110111100011001111000000
01111110010110110000110101110001011110001110110101100110110010110110101111010111
001100101100011010111101110011110101010010111110011100111100110110100011100110100
00011100011001000011101101111010101101111111100110010011111010110010111111010101
11110001110011000100010011101000001000110111000010111010010100100101000010010000
11101101101110100010100100110110101111100000000100101001111100011111101010010111
11111011001000110010100001110001101000100010100100000001001111010011111000011110
10110000010110111000111010110111111000000100010111101110110000010110000100000111
00101111010011011001111111100000100001011100100011100010001010101011000111100111
01101010010000010100111100110100001111000001001101000100101011110011001011110011
01011101000001111010011000111010011110111001010100101011010111011001001111011101
00100100010100101001100011111111000101011100111110110101110101000001011101100110
10111111001111010101011000000101111001000000010100101010001110000110110111001010
11011010101101100000110101100111010111111001111110111010100001011110000110010011
11100101000100110010111100100000111011011100101011011001011011111010110011110101
01101100000001111011110100000011010110000111001010001101011001100101011100100101
1010001101110001101001111101011011110011110000111100011010110000110100010101101100

01001101111011011000011010010010010010010101110100100110101000001101001100001011
1010010001010010100100001111111001101011100111110110101110101010001001101100010
110000110010100011110111101000110101101101111000111001100011001010011001101000110

Basic correlated LSH partitioning

10001110000100110110110010001111110111111000000010000111010011111100000110101000
10110011000110001111011101000110101101101111000111001100011001010011001101000100
01011001001101100000111000100000010110111110000001010011110011110111010011001011
01110100100100110001110011111010110001011010011000011011000111110001111000110101
11110101111010110100100001101011101010001011010101111100100110000000111101000110
0001100110111000001001010010110001100101111111111110001111000001001100100011111
00011101110000010000100101000000101100011011111000011011011011000110101101110110
00101101111011011001011010010010010110010101110100101010101000001100011100000010
1110101100001010010010101110100110111100100000100110111110111100011001111000000
01111110010110110000110101110001011110001110110101100110110010110110101111010111
00110010110001101011101110011110101010010111110011100111100110110100011100110100
0001110001100100001110110111010101101111111100110010011111010110010111111010101
1111000111001100010001000100111010000010001101110000101110100101001001001000010010000
11101101101110100010100100110110101111100000000100101001111100011111101010010111
11111011001000110010100001110001101000100010100100000001001111010011111000011110
10110000010110111000111010110111111000000100010111101110110000010110000100000111
00101111010011011001111111100000100001011100100011100010001010101011000111100111
01101010010000010100111100110100001111000001001101000100101011110011001011110011
010111010000011110100110000111010011110111001010100101011010111011001001111011101
00100100010100101001100011111111000101011100111110110101110101000001011101100110
10111111001111010101011000000101111001000000010100101010001110000110110111001010
11011010101101100000110101100111010111111001111110111010100001011110000110010011
11100101000100110010111100100000111011011100101011011001011011111010110011110101
01101100000001111011110100000011010110000111001010001101011001100101011100100101
1010001101110001101001111101011011110011110000111100011010110000110100010101101100

01001101111011011000011010010010010010010101110100100110101000001101001100001011
1010010001010010100100001111111001101011100111110110101110101010001001101100010
11000011001010001111011101000110101101101111000111001100011001010011001101000110

h_1

Basic correlated LSH partitioning

10001110000100110110110010001111110111111000000010000111010011111100000110101000
10110011000110001111011101000110101101101111000111001100011001010011001101000100
01011001001101100000111000100000010110111110000001010011110011110111010011001011
01110100100100110001110011111010110001011010011000011011000111110001111000110101
11110101111010110100100001101011101010001011010101111100100110000000111101000110
00011001101110000010010100101100011001011111111110001111000001001100100011111
00011101110000010000100101000000101100011011111000011011011011000110101101110110
00101101111011011001011010010010010110010101110100101010101000001100011100000010
1110101100001010010010101110100110111100100000100110111110111100011001111000000
01111110010110110000110101110001011110001110110101100110110010110110101111010111
001100101100011010111101110011110101010010111110011100111100110110100011100110100
000111000110010000111011011110101011011111111100110010011111010110010111111010101
1111000111001100010001000100111010000010001101110000101110100101001001001000010010000
1110110110111010001010010011011010111110000000100101001111100011111101010010111
11111011001000110010100001110001101000100010100100000001001111010011111000011110
10110000010110111000111010110111111000000100010111101110110000010110000100000111
00101111010011011001111111100000100001011100100011100010001010101011000111100111
01101010010000010100111100110100001111000001001101000100101011110011001011110011
01011101000001111010011000111010011110111001010100101011010111011001001111011101
00100100010100101001100011111111000101011100111110110101110101000001011101100110
10111111001111010101011000000101111001000000010100101010001110000110110111001010
11011010101101100000110101100111010111111001111110111010100001011110000110010011
11100101000100110010111100100000111011011100101011011001011011111010110011110101
01101100000001111011110100000011010110000111001010001101011001100101011100100101
101000110111000110100111110101101111001110000111100011010110000110100010101101100

01001101111011011000011010010010010010010101110100100110101000001101001100001011
1010010001010010100100001111111001101011100111110110101110101010001001101100010
110000110010100011110111010001101011011011110001110011001010011001101000110

h_2

Basic correlated LSH partitioning

10001110000100110110110010001111110111111000000010000111010011111100000110101000
10110011000110001111011101000110101101101111000111001100011001010011001101000100
01011001001101100000111000100000010110111110000001010011110011110111010011001011
01110100100100110001110011111010110001011010011000011011000111110001111000110101
11110101111010110100100001101011101010001011010101111100100110000000111101000110
00011001101110000010010100101100011001011111111111000111100001001100100011111
00011101110000010000100101000000101100011011111000011011011011000110101101110110
00101101111011011001011010010010010110010101110100101010101000001100011100000010
11101011000010100100101011101001101111001000001001101111110111100011001111000000
01111110010110110000110101110001011110001110110101100110110010110110101111010111
001100101100011010111101110011110101010010111110011100111100110110100011100110100
00011100011001000011101101111010101101111111100110010011111010110010111111010101
11110001110011000100010011101000001000110111000010111010010100100101000010010000
11101101101110100010100100110110101111100000000100101001111100011111101010010111
11111011001000110010100001110001101000100010100100000001001111010011111000011110
10110000010110111000111010110111111000000100010111101110110000010110000100000111
00101111010011011001111111100000100001011100100011100010001010101011000111100111
01101010010000010100111100110100001111000001001101000100101011110011001011110011
010111010000011110100110001110100111101111001010100101011010111011001001111011101
00100100010100101001100011111111000101011100111110110101110101000001011101100110
10111111001111010101011000000101111001000000010100101010001110000110110111001010
11011010101101100000110101100111010111111001111110111010100001011110000110010011
11100101000100110010111100100000111011011100101011011001011011111010110011110101
01101100000001111011110100000011010110000111001010001101011001100101011100100101
1010001101110001101001111101011011110011110000111100011010110000110100010101101100

01001101111011011000011010010010010010010010101110100100110101000001101001100001011
1010010001010010100100001111111001101011100111110110101110101010001001101100010
1100001100101000111101111010001101011011011111000111001100011001010011001000110

h_3

Basic correlated LSH partitioning

10001110000100110110110010001111110111111000000010000111010011111100000110101000
10110011000110001111011101000110101101101111000111001100011001010011001101000100
01011001001101100000111000100000010110111110000001010011110011110111010011001011
01110100100100110001110011111010110001011010011000011011000111110001111000110101
11110101111010110100100001101011101010001011010101111100100110000000111101000110
0001100110111000001001010010110001100101111111111110001111000001001100100011111
00011101110000010000100101000000101100011011111000011011011011000110101101110110
00101101111011011001011010010010010110010101110100101010101000001100011100000010
11101011000010100100101011101001101111001000001001101111110111100011001111000000
01111110010110110000110101110001011110001110110101100110110010110110101111010111
001100101100011010111101110011110101010010111110011100111100110110100011100110100
00011100011001000011101101111010101101111111100110010011111010110010111111010101
11110001110011000100010011101000001000110111000010111010010100100101000010010000
11101101101110100010100100110110101111100000000100101001111100011111101010010111
11111011001000110010100001110001101000100010100100000001001111010011111000011110
10110000010110111000111010110111111000000100010111101110110000010110000100000111
00101111010011011001111111100000100001011100100011100010001010101011000111100111
01101010010000010100111100110100001111000001001101000100101011110011001011110011
01011101000001111010011000111010011110111001010100101011010111011001001111011101
00100100010100101001100011111111000101011100111110110101110101000001011101100110
10111111001111010101011000000101111001000000010100101010001110000110110111001010
11011010101101100000110101100111010111111001111110111010100001011110000110010011
11100101000100110010111100100000111011011100101011011001011011111010110011110101
01101100000001111011110100000011010110000111001010001101011001100101011100100101
1010001101110001101001111101011011110011110000111100011010110000110100010101101100

01001101111011011000011010010010010010010101110100100110101000001101001100001011
1010010001010010100100001111111001101011100111110110101110101010001001101100010
110000110010100011110111101000110101101101111000111001100011001010011001101000110

h_4

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

Basic correlated LSH enumeration

10001110000100110110110010001111110111111000000010000111010011111100000110101000
10110011000110001111011101000110101101101111000111001100011001010011001101000100
01011001001101100000111000100000010110111110000001010011110011110111010011001011
01110100100100110001110011111010110001011010011000011011000111110001111000110101
11110101111010110100100001101011101010001011010101111100100110000000111101000110
00011001101110000010010100101100011001011111111110001111000001001100100011111
00011101110000010000100101000000101100011011111000011011011011000110101101110110
00101101111011011001011010010010010110010110010101110100101010101000001100011100000010
1110101100001010010010101110100110111100100000100110111110111100011001111000000
01111110010110110000110101110001011110001110110101100110110010110110101111010111
00110010110001101011101110011110101010010111110011100110110100011100110100
00011100011001000011101101111010101101111111110011001001111101011001011111010101
111100011100110001000100111010000010001101110000101110100101001001001000010010000
1110110110111010001010010011011010111110000000100101001111100011111101010010111
111110110010001100101000011100011010001000101001000100000001001111010011111000011110
10110000010110111000111010110111111000000100010111101110110000010110000100000111
001011110100110110011111111100000100001011100100011100010001010101011000111100111
01101010010000010100111100110100001111000001001101000100101011110011001011110011
01011101000001111010011000111010011110111001010100101011010111011001001111011101
001001000101001010011000111111111000101011100111110110101110101000001011101100110
1011111100111101010101100000010111100100000010100101010001110000110110111001010
11011010101101100000110101100111010111111001111110111010100001011110000110010011
11100101000100110010111100100000111011011100101011011001011011111010110011110101
01101100000001111011110100000011010110000111001010001101011001100101011100100101
101000110111000110100111110101101111001110000111100011010110000110100010101101100

01001101111011011000011010010010010010010010101110100100110101000001101001100001011
1010010001010010100100001111111001101011100111110110101110101010001001101100010
110000110010100011110111010001101011011011110001110011001010011001101000110

h_{12}

Basic correlated LSH enumeration

1000111000010011011011001000111111011111100000010000111010011111100000110101000
10110011000110001111011101000110101101101111000111001100011001010011001101000100
01011001001101100000111000100000010110111110000001010011110011110111010011001011
01110100100100110001110011111010110001011010011000011011000111110001111000110101
11110101111010110100100001101011101010001011010101111100100110000000111101000110
000110011011100000100101001011000110010111111111111000111100001001100100011111
00011101110000010000100101000000101100011011111000011011011011000110101101110110
00101101111011011001011010010010010110010101110100101010101000001100011100000010
11101011000010100100101011101001101111001000001001101111110111100011001111000000
01111110010110110000110101110001011110001110110101100110110010110110101111010111
001100101100011010111101110011110101010010111110011100111100110110100011100110100
00011100011001000011101101110101011011111111100110010011111010110010111111010101
1111000111001100010001000111010000010001101110000101110100101001001001000010010000
1110110110111010001010010011011010111110000000100101001111100011111101010010111
11111011001000110010100001110001101000100010100100000001001111010011111000011110
10110000010110111000111010110111111000000100010111101110110000010110000100000111
00101111010011011001111111100000100001011100100011100010001010101011000111100111
01101010010000010100111100110100001111000001001101000100101011110011001011110011
01011101000001111010011000111010011110111001010100101011010111011001001111011101
001001000101001010011000111111111000101011100111110110101110101000001011101100110
10111111001111010101011000000101111001000000010100101010001110000110110111001010
11011010101101100000110101100111010111111001111110111010100001011110000110010011
11100101000100110010111100100000111011011100101011011001011011111010110011110101
01101100000001111011110100000011010110000111001010001101011001100101011100100101
1010001101110001101001111101011011110011110000111100011010110000110100010101101100

01001101111011011000011010010010010010010010101110100100110101000001101001100001011
1010010001010010100100001111111001101011100111110110101110101010001001101100010
11000011001010001111011101000110101101101111000111001100011001010011001101000110

h_{13}

Basic correlated LSH enumeration

10001110000100110110110010001111110111111000000010000111010011111100000110101000
10110011000110001111011101000110101101101111000111001100011001010011001101000100
01011001001101100000111000100000010110111110000001010011110011110111010011001011
01110100100100110001110011111010110001011010011000011011000111110001111000110101
11110101111010110100100001101011101010001011010101111100100110000000111101000110
0001100110111000001001010010110001100101111111111110001111000001001100100011111
00011101110000010000100101000000101100011011111000011011011011000110101101110110
00101101111011011001011010010010010110010101110100101010101000001100011100000010
11101011000010100100101011101001101111001000001001101111110111100011001111000000
01111110010110110000110101110001011110001110110101100110110010110110101111010111
001100101100011010111101110011110101010010111110011100111100110110100011100110100
00011100011001000011101101111010101101111111100110010011111010110010111111010101
111100011100110001000100011101000001000110111000010111010010100100101000010010000
1110110110111010001010010011011010111110000000100101001111100011111101010010111
11111011001000110010100001110001101000100010100100000001001111010011111000011110
10110000010110111000111010110111111000000100010111101110110000010110000100000111
00101111010011011001111111100000100001011100100011100010001010101011000111100111
01101010010000010100111100110100001111000001001101000100101011110011001011110011
01011101000001111010011000111010011110111001010100101011010111011001001111011101
00100100010100101001100011111111000101011100111110110101110101000001011101100110
10111111001111010101011000000101111001000000010100101010001110000110110111001010
11011010101101100000110101100111010111111001111110111010100001011110000110010011
11100101000100110010111100100000111011011100101011011001011011111010110011110101
01101100000001111011110100000011010110000111001010001101011001100101011100100101
101000110111000110100111110101101111001110000111100011010110000110100010101101100

01001101111011011000011010010010010010010101110100100110101000001101001100001011
1010010001010010100100001111111001101011100111110110101110101010001001101100010
11000011001010001111011101000110101101101111000111001100011001010011001101000110

h_{14}

Basic correlated LSH enumeration

For Hamming distance ≤ 2 ,
a collision is guaranteed in
 $h_{12}, h_{13}, h_{14}, h_{23}, h_{24}, h_{34}$

10001110000100110110110010001111110111111000000010000111010011111100000110101000
10110011000110001111011101000110101101101111000111001100011001010011001101000100
01011001001101100000111000100000010110111110000001010011110011110111010011001011
01110100100100110001110011111010110001011010011000011011000111110001111000110101
11110101111010110100100001101011101010001011010101111100100110000000111101000110
0001100110111000001001010010110001100101111111111110001111000001001100100011111
0001110111000001000010000100001000010000100001000010000100001000010000100001000010000
00101101111011011001011001011001011001011001011001011001011001011001011001011001011
111010110000101001001001011001011001011001011001011001011001011001011001011001011
0111111001011011000011001011001011001011001011001011001011001011001011001011001011
001100101100011010111010111010111010111010111010111010111010111010111010111010111
000111000110010000111010111010111010111010111010111010111010111010111010111010111
1111000111001100010001000100010001000100010001000100010001000100010001000100010001000
11101101101110100010100101001010010100101001010010100101001010010100101001010010100
11111011001000110010100101001010010100101001010010100101001010010100101001010010100
10110000010110111000111000111000111000111000111000111000111000111000111000111000111
00101111010011011001111001111001111001111001111001111001111001111001111001111001111
0110101001000001010011110100111010011110111001010100101011010111011001001111011101
0101110100000111101001110001111010011110111001010100101011010111011001001111011101
001001000101001010011000111111111000101011100111110110101110101000001011101100110
10111111001111010101011000000101111001000000010100101010001110000110110111001010
11011010101101100000110101100111010111111001111110111010100001011110000110010011
11100101000100110010111100100000111011011100101011011001011011111010110011110101
01101100000001111011110100000011010110000111001010001101011001100101011100100101
1010001101110001101001111101011011110011110000111100011010110000110100010101101100

01001101111011011000011010010010010010010101110100100110101000001101001100001011
1010010001010010100100001111111001101011100111110110101110101010001001101100010
11000011001010001111011101000110101101101111000111001100011001010011001101000110

h_{14}

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

- Question:
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

- 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

- 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

- 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

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

Number of hash functions

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

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

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

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

$$p = 4/7$$

$$r = 2$$

A covering design

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

A covering design

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

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

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)

		001	010	011	100	101	110	111
Index (binary)	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

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)

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

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

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)

		001	010	011	100	101	110	111
Index (binary)	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

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

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)

		001	010	011	100	101	110	111
Index (binary)	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

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)

		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	

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

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

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

Sampling the covering design

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

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

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

Lemma:

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

```
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 .

$$q^t = 110\mathbf{0}101110\mathbf{0}101110\mathbf{0}101110\mathbf{0}101$$

$$x^t = 110\mathbf{1}101110\mathbf{1}101110\mathbf{1}101110\mathbf{1}101$$

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

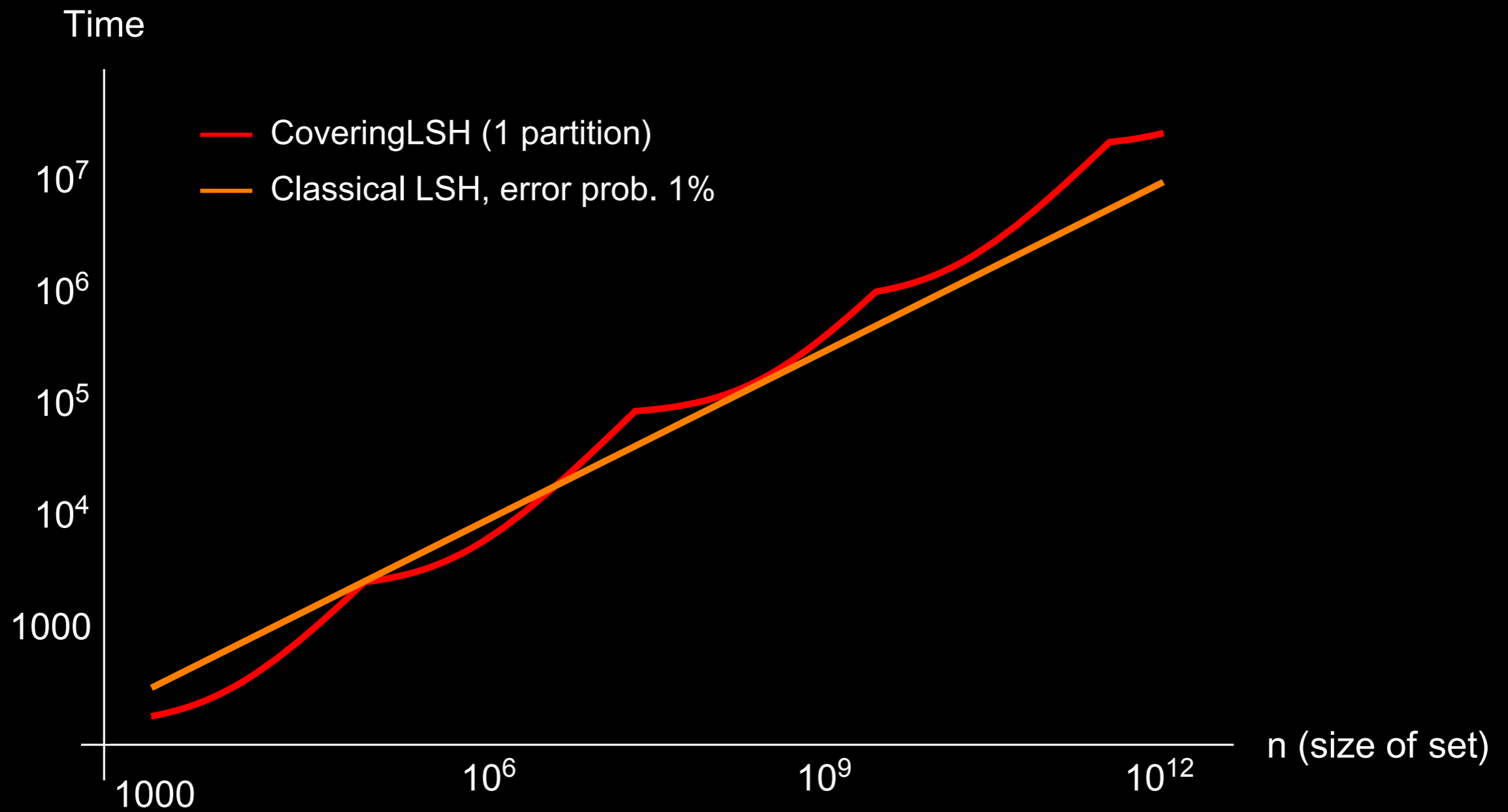
Too many collisions?

- 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

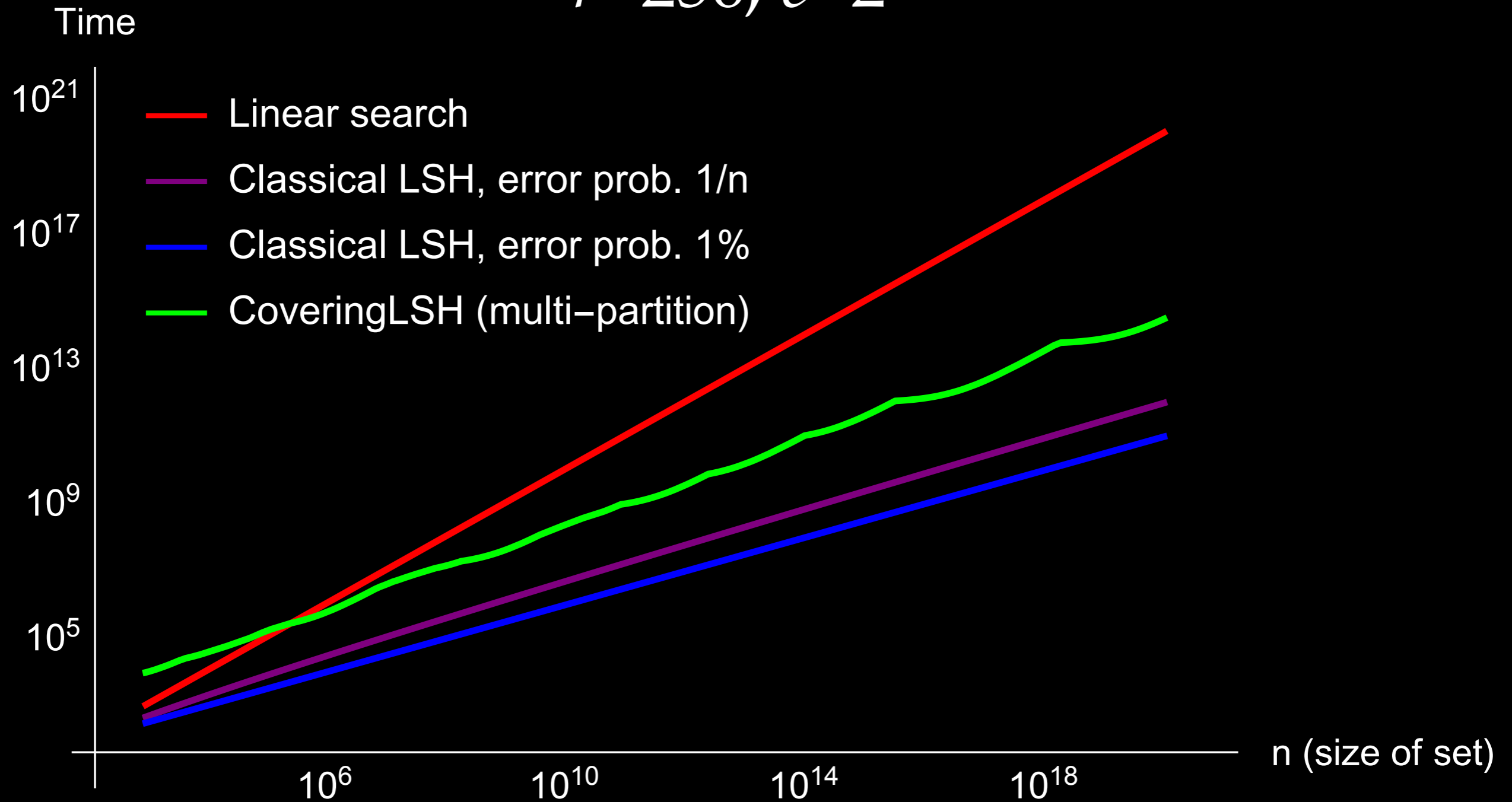
1	0	1	0	1	0	1	0	0	0	0	0	0	0
0	1	1	0	0	1	1	0	0	0	0	0	0	0
1	1	0	0	1	1	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	0	0	0
1	0	1	1	0	1	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	1	0	1	0	1
0	0	0	0	0	0	0	0	1	1	0	0	1	1
0	0	0	0	0	0	0	1	1	0	0	1	1	0
0	0	0	0	0	0	0	0	0	0	1	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

$$s = 2$$

$$r = 5$$

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

