Distance-Sensitive Hashing

Martin Aumüller, Tobias Christiani, Rasmus Pagh, Francesco Silvestri

IT University of Copenhagen & University of Padova

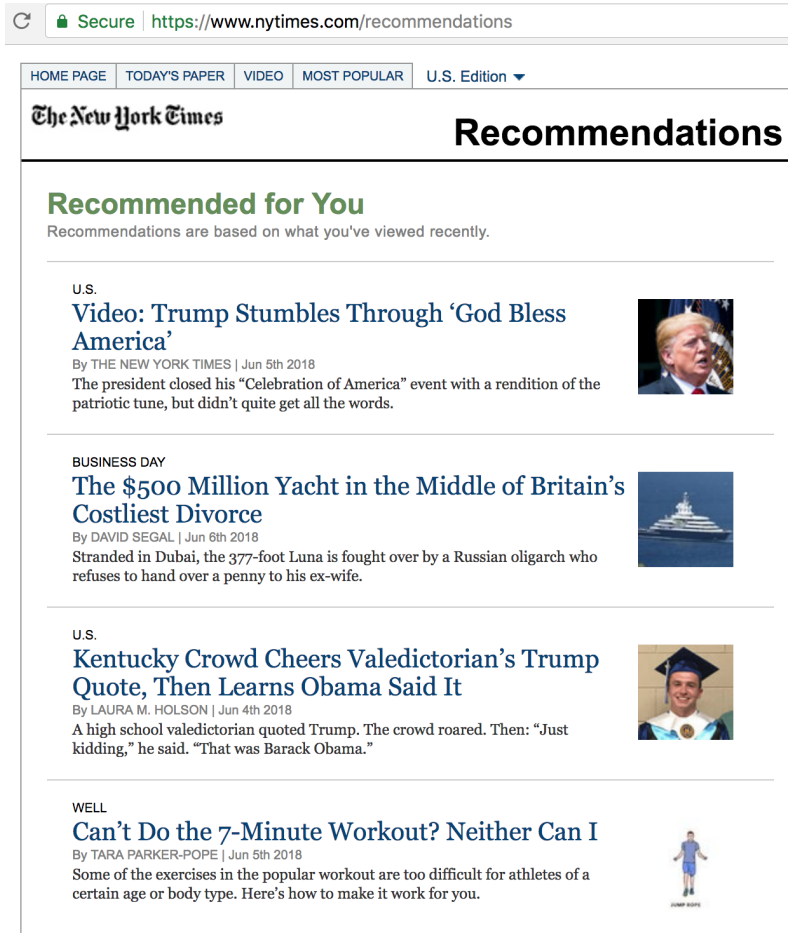
June 11, 2018

PODS 18

Supported by

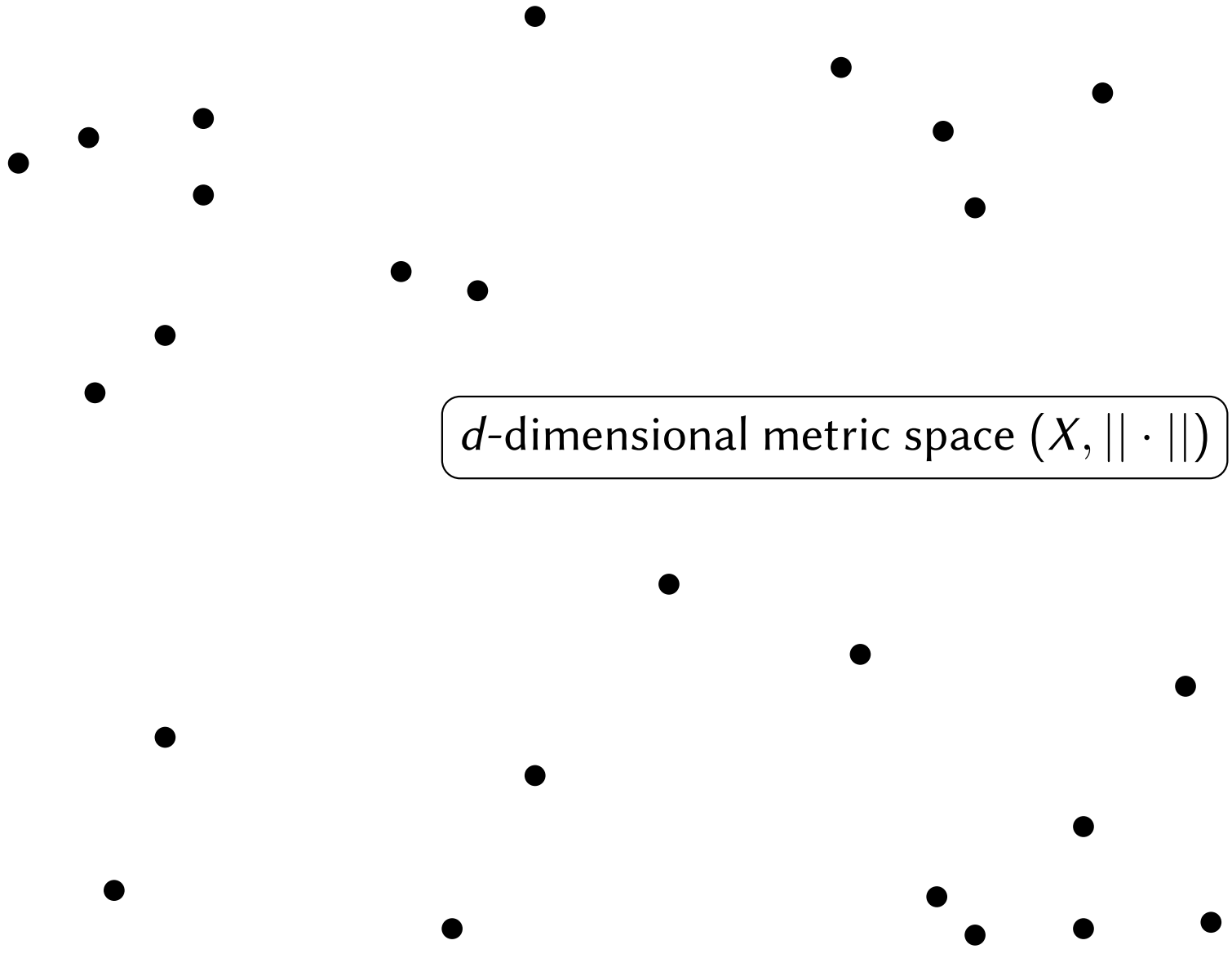
Motivation: Recommender Systems

- **Data**: Collection of newspaper articles S
- **Query**: User read article **x**, recommend "interesting" other articles from S
- **What is interesting?**
  - not too close to **x**
  - not too far away from **x**

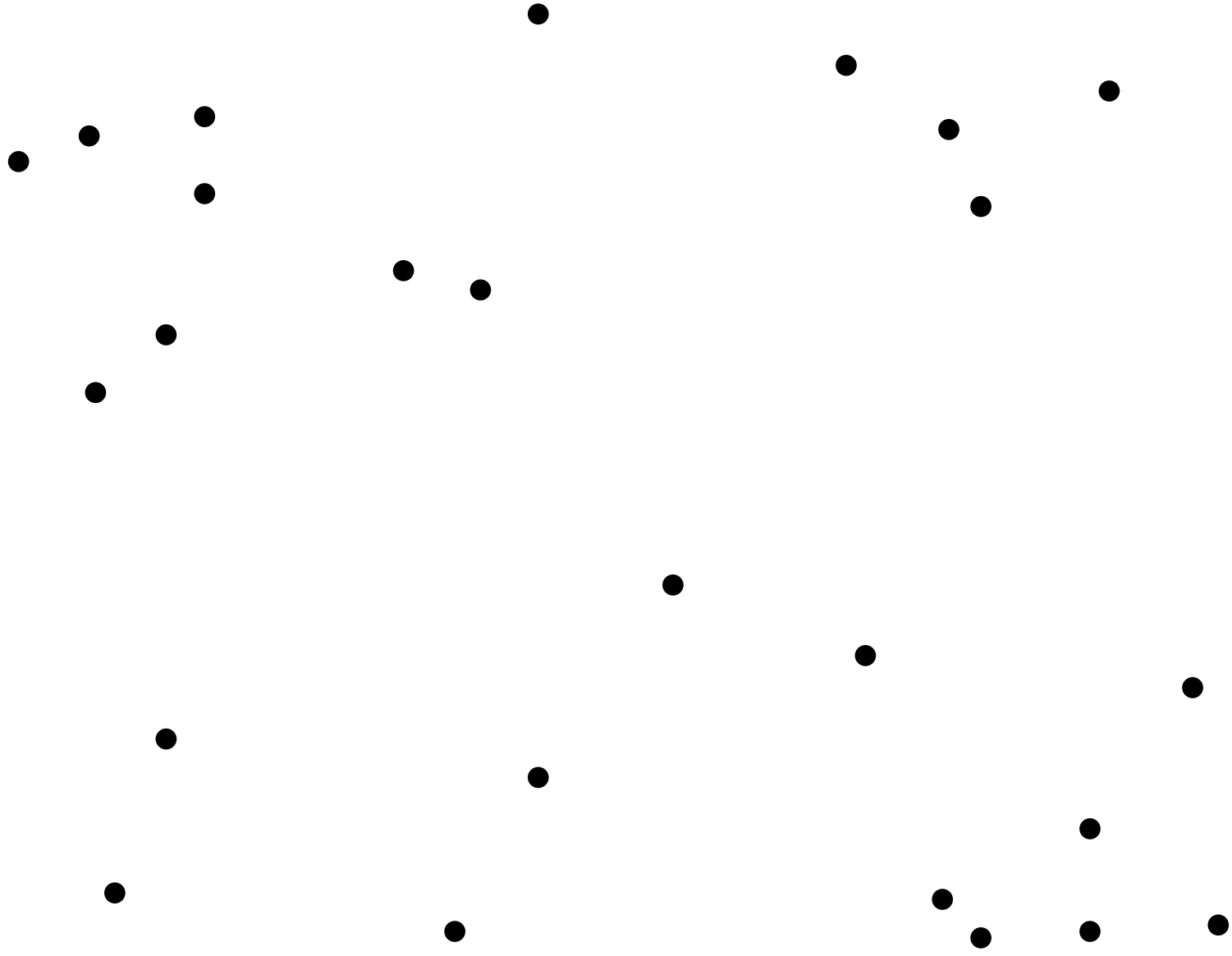


General setting: Approximate Similarity Search Problems

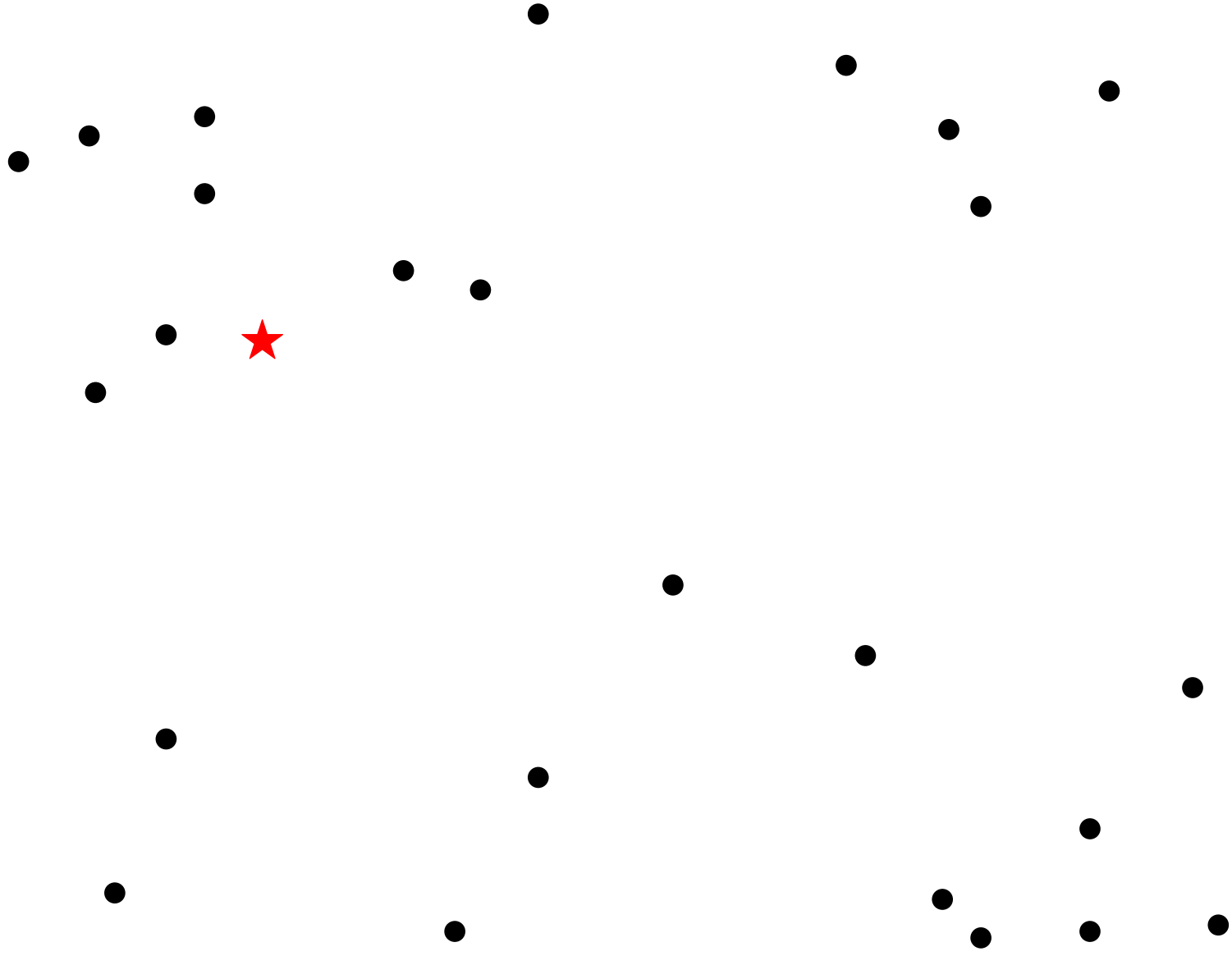
General setting: Approximate Similarity Search Problems



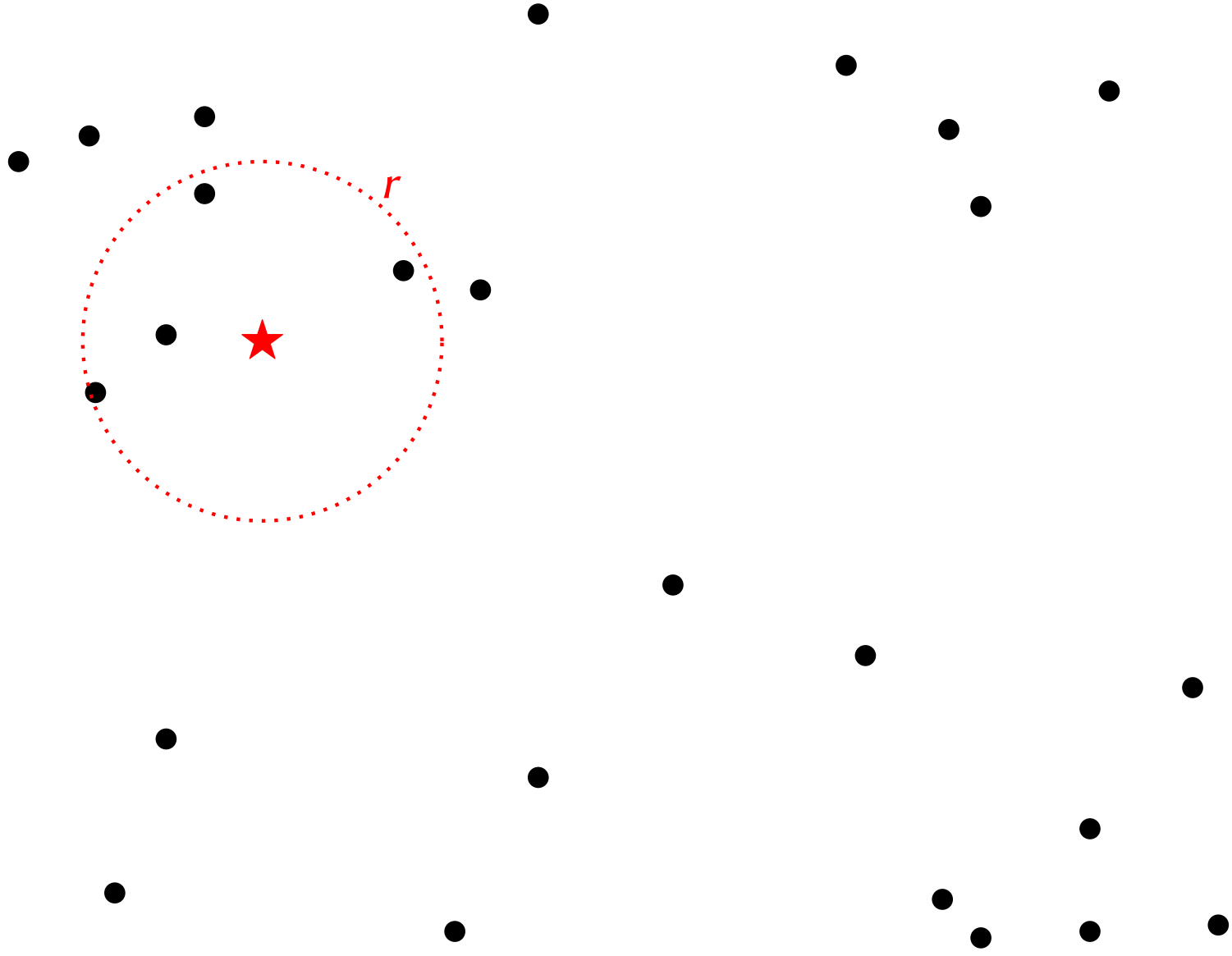
(c, r)-approximate Annulus Problem



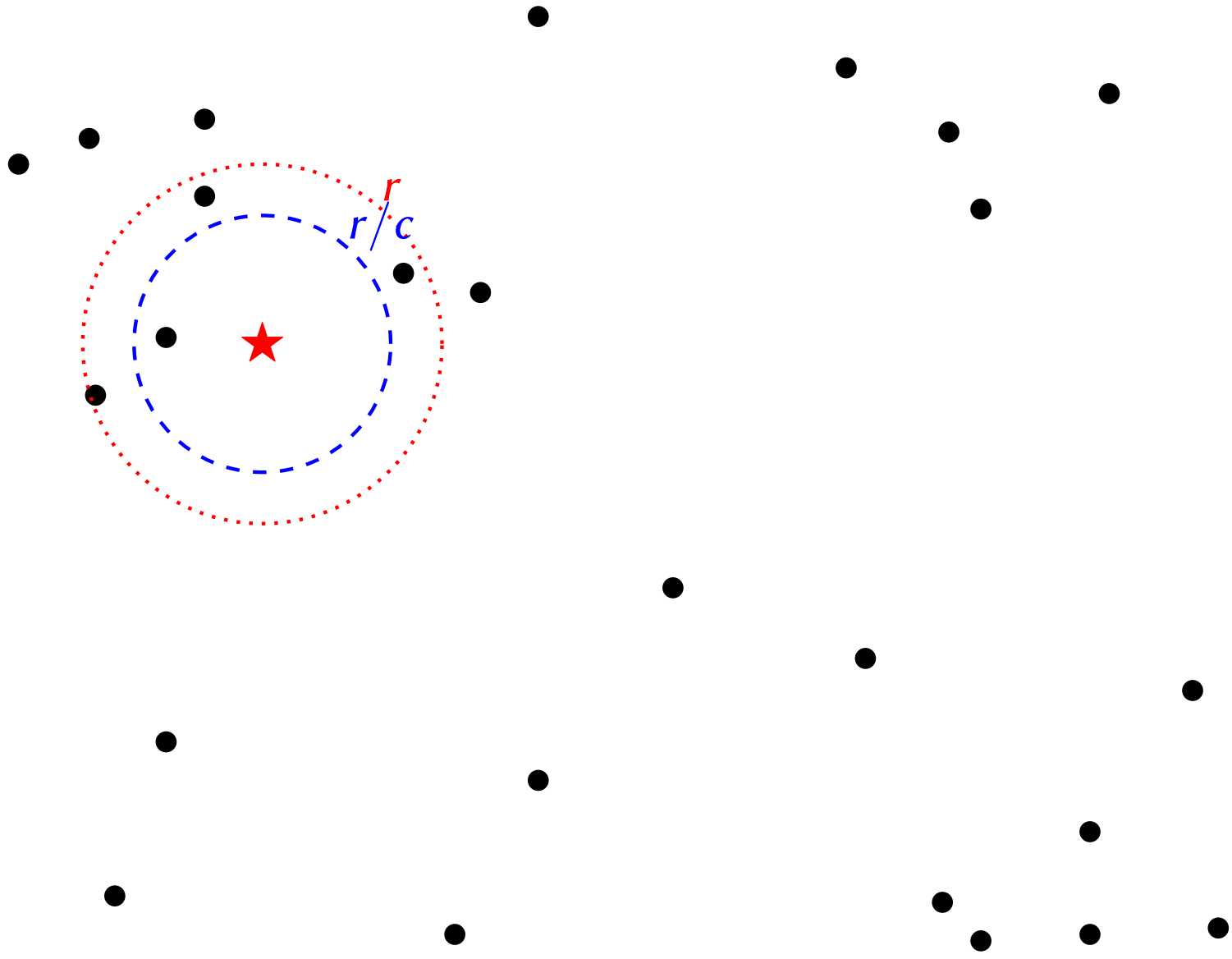
(c, r)-approximate Annulus Problem



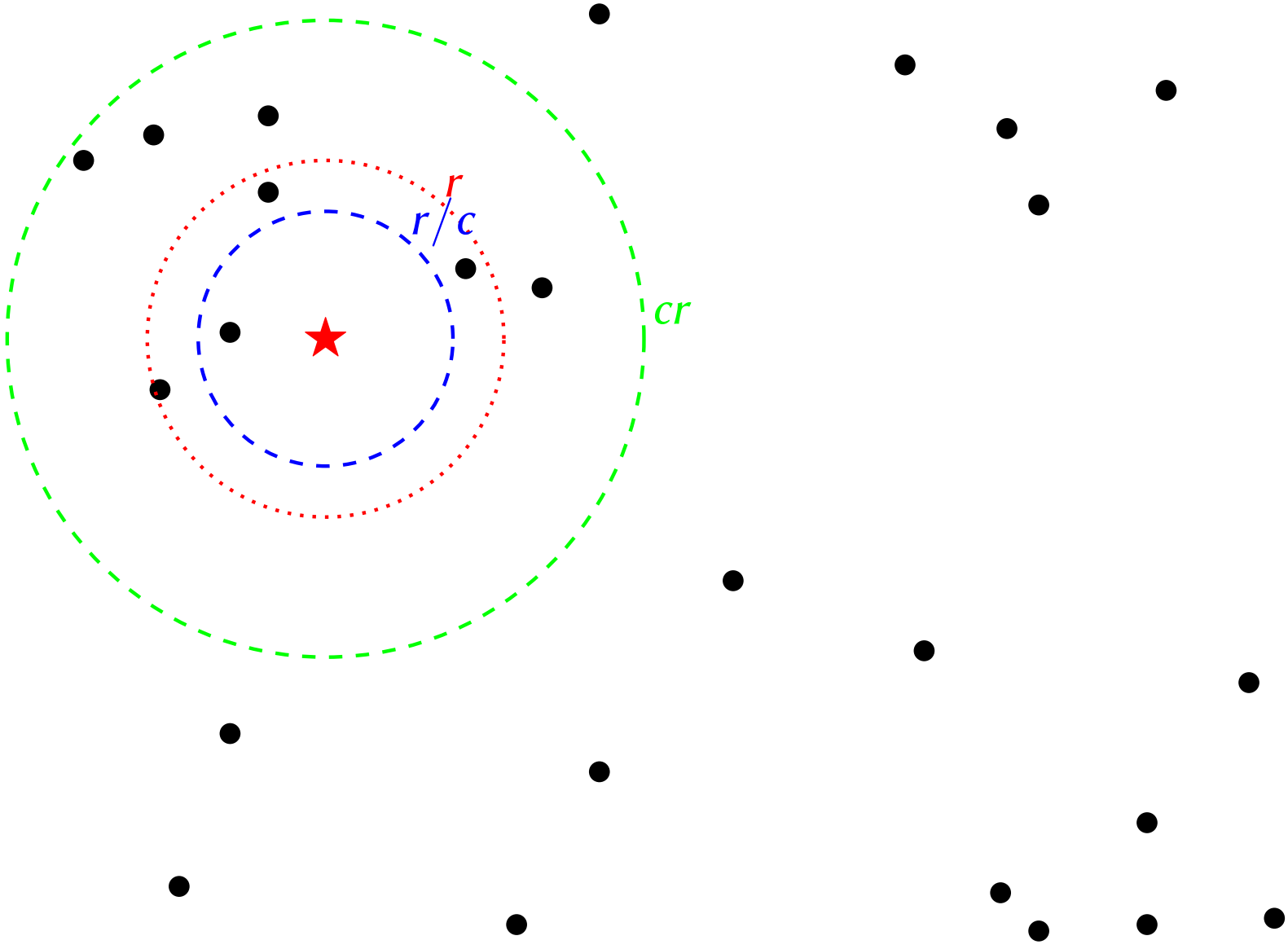
(c, r)-approximate Annulus Problem



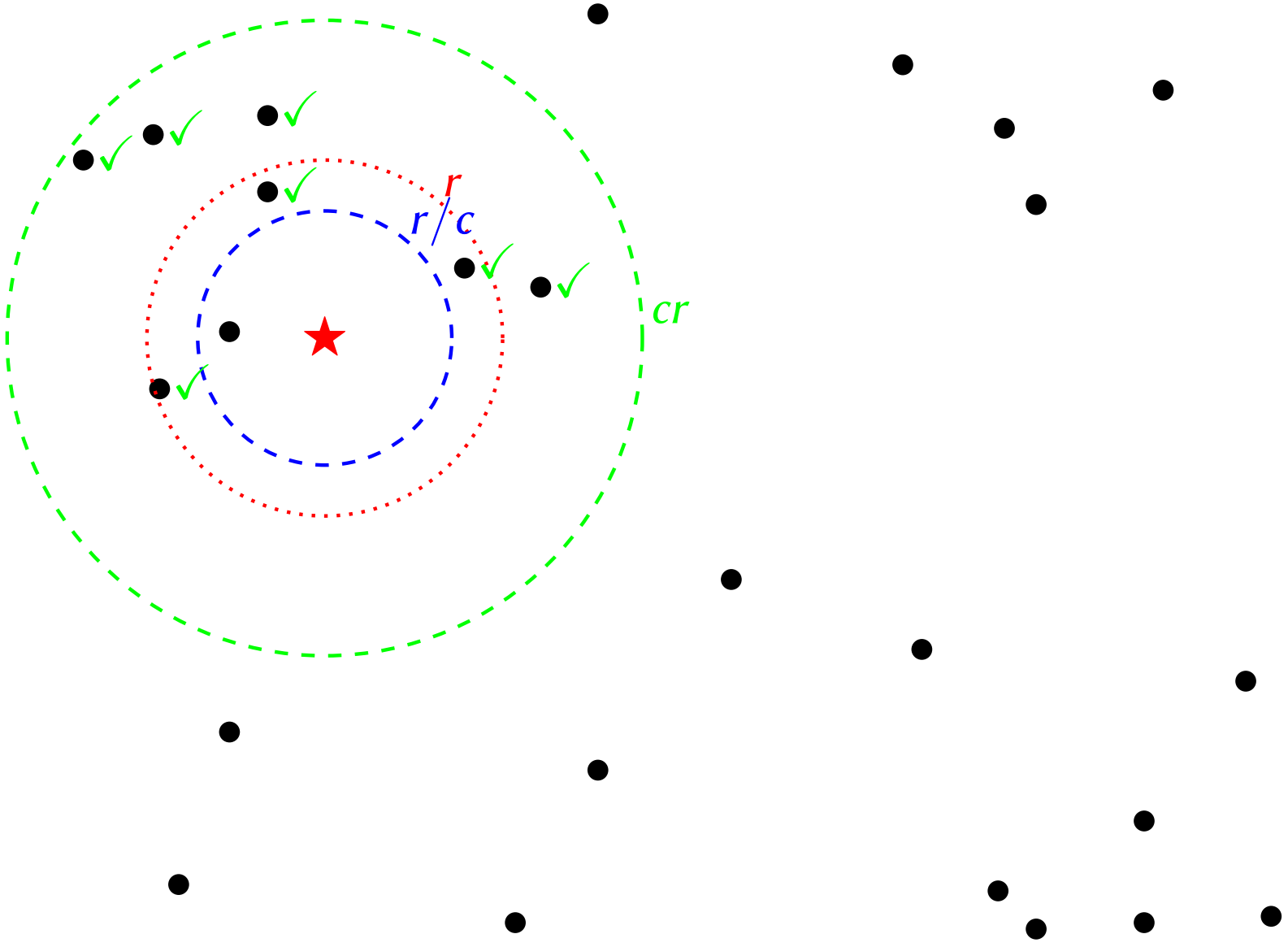
(c, r)-approximate Annulus Problem

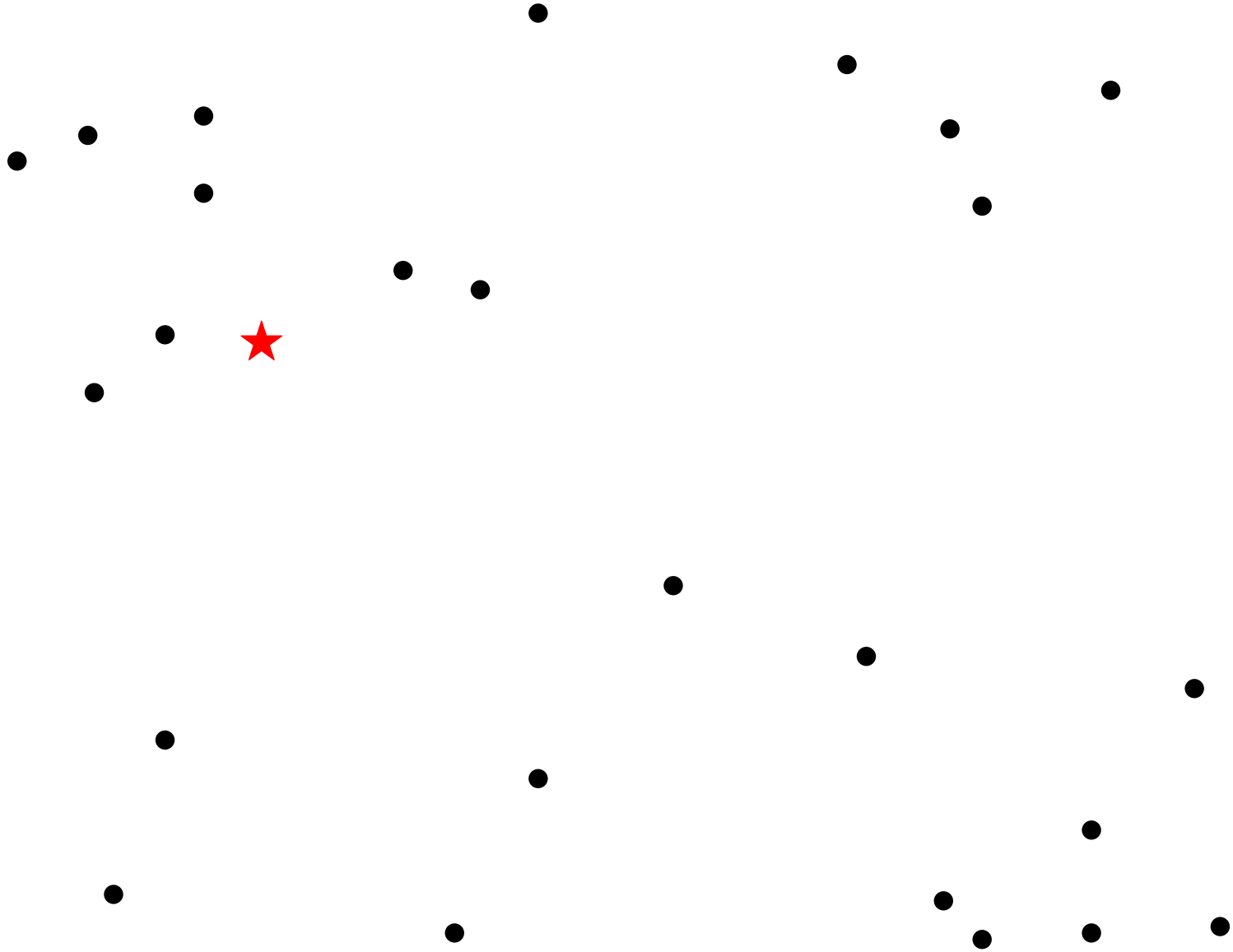


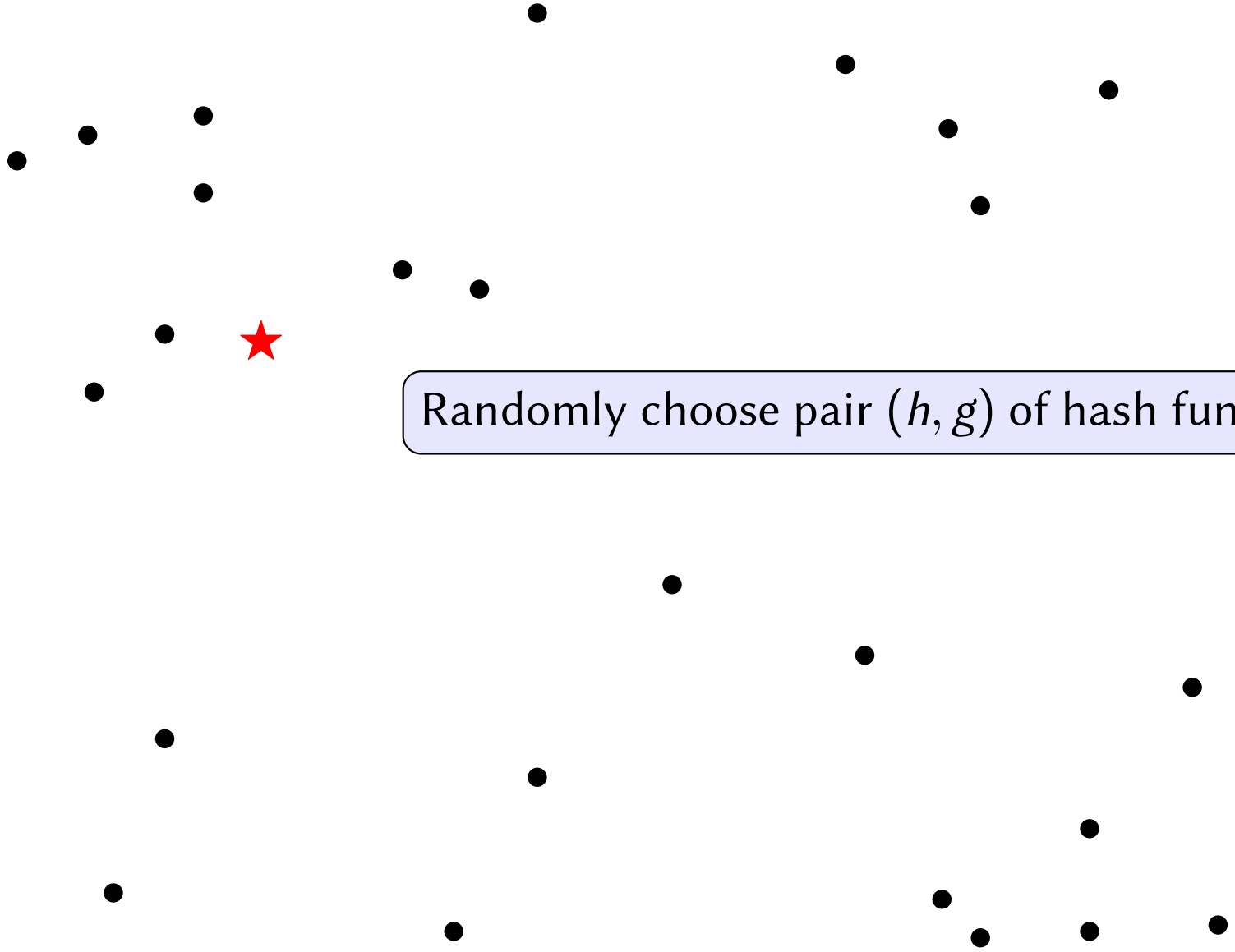
(c, r)-approximate Annulus Problem



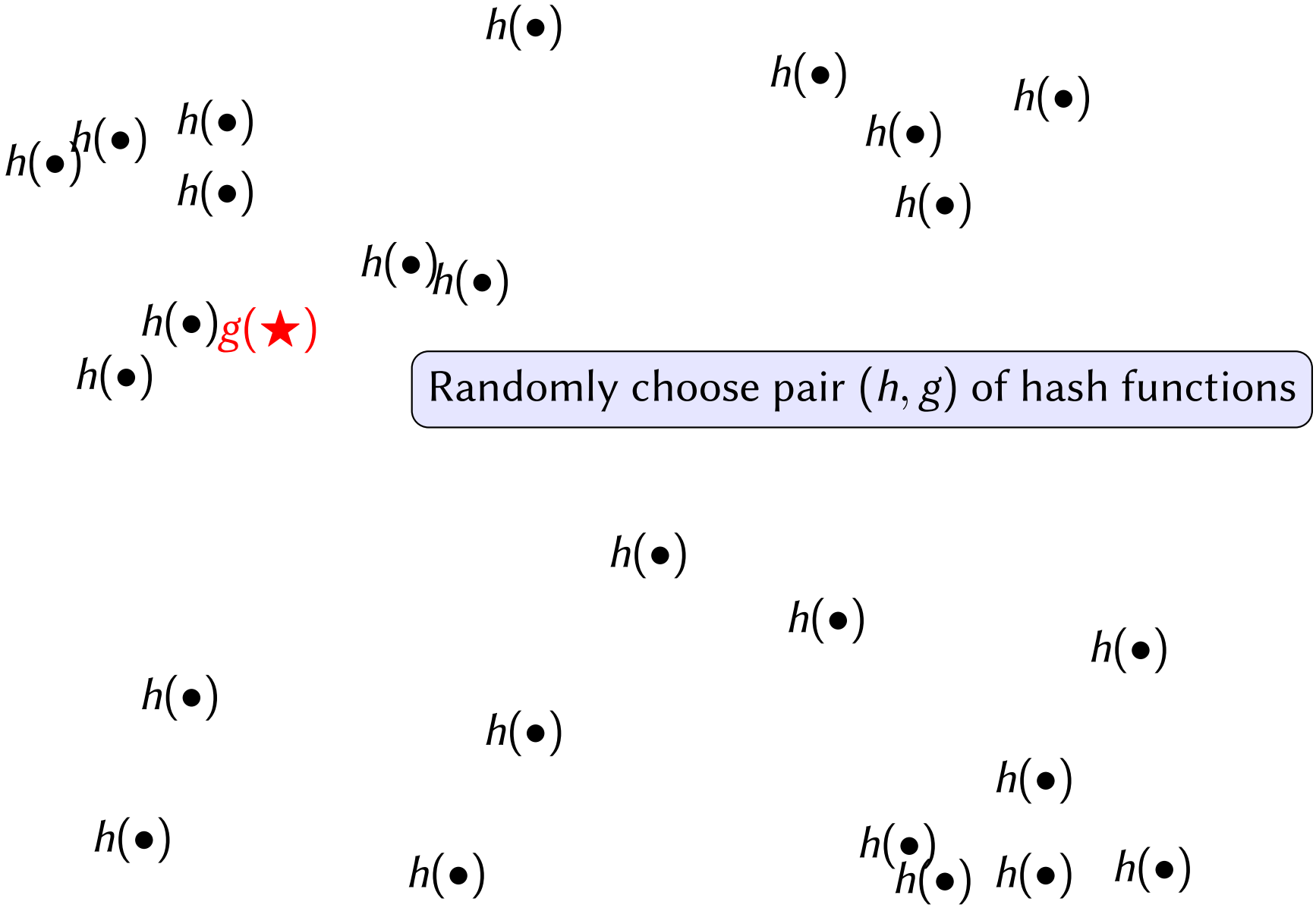
(c, r)-approximate Annulus Problem

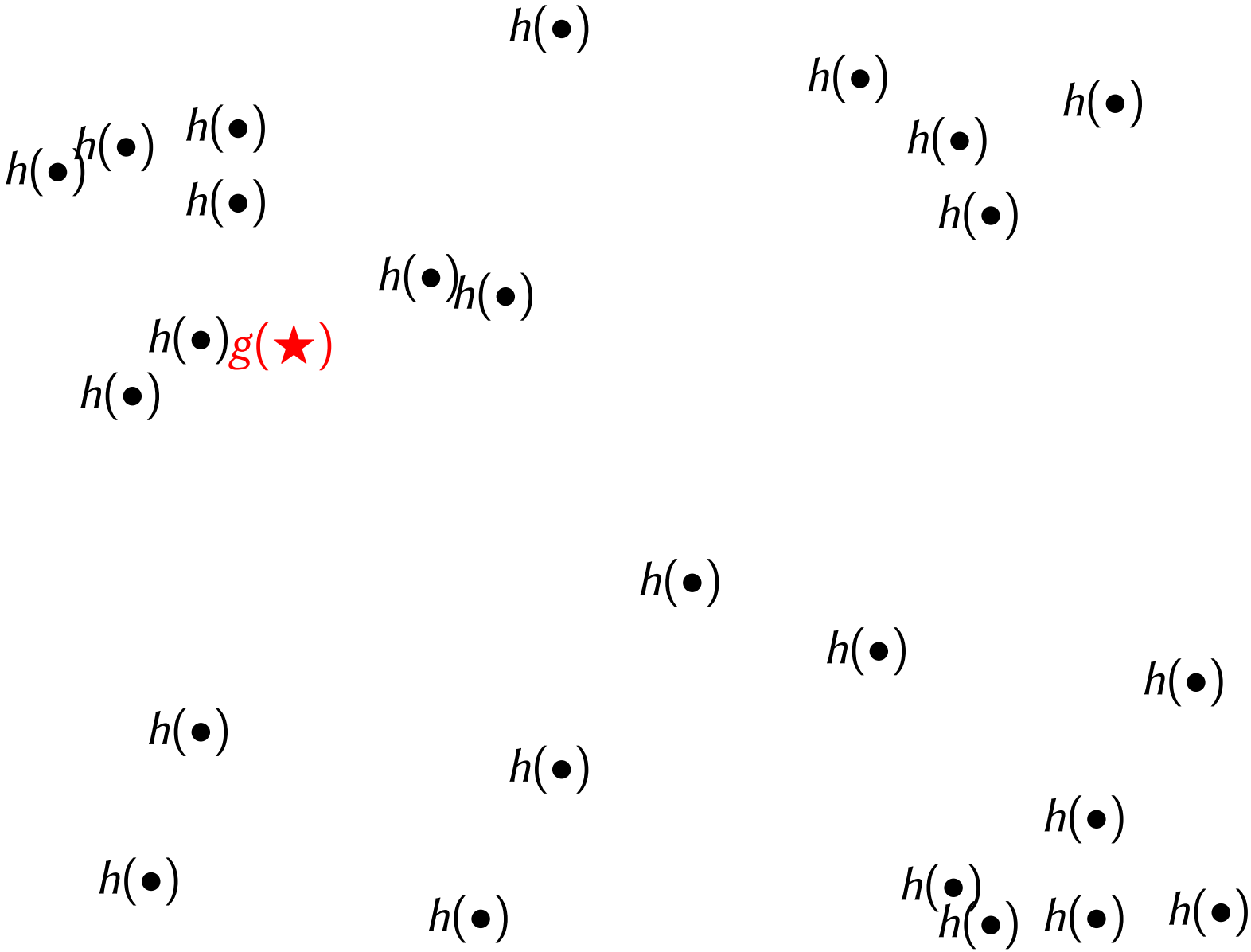




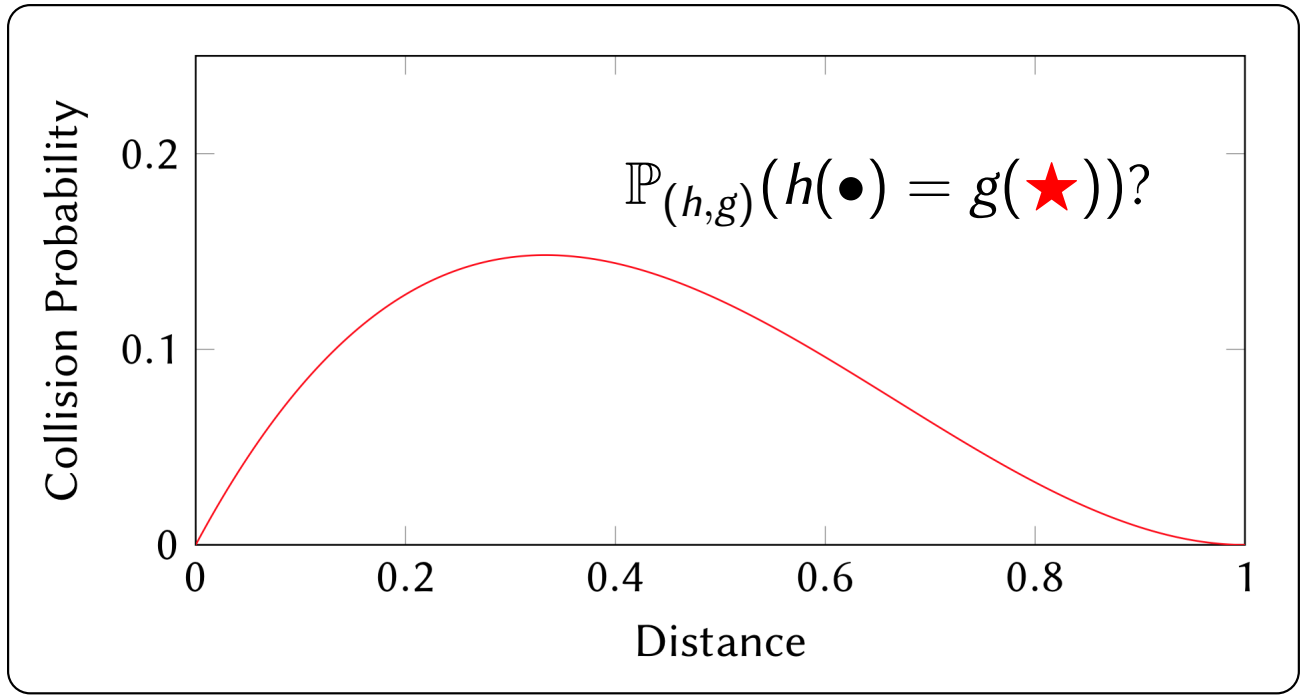


Randomly choose pair (h, g) of hash functions


h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
g(★)
h(•)
Randomly choose pair (h, g) of hash functions
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)


h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
g(★)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)
h(•)

$h(\bullet)$ $h(\bullet)$ $h(\bullet)$ $h(\bullet)$ $h(\bullet)$ $h(\bullet)$ $h(\bullet)$ $h(\bullet)$ $h(\bullet)$ $h(\bullet)$ $h(\bullet)$ $h(\bullet)$ $h(\bullet)$ $g(\bigstar)$ $h(\bullet)$ $h(\bullet)$ $h(\bullet)$

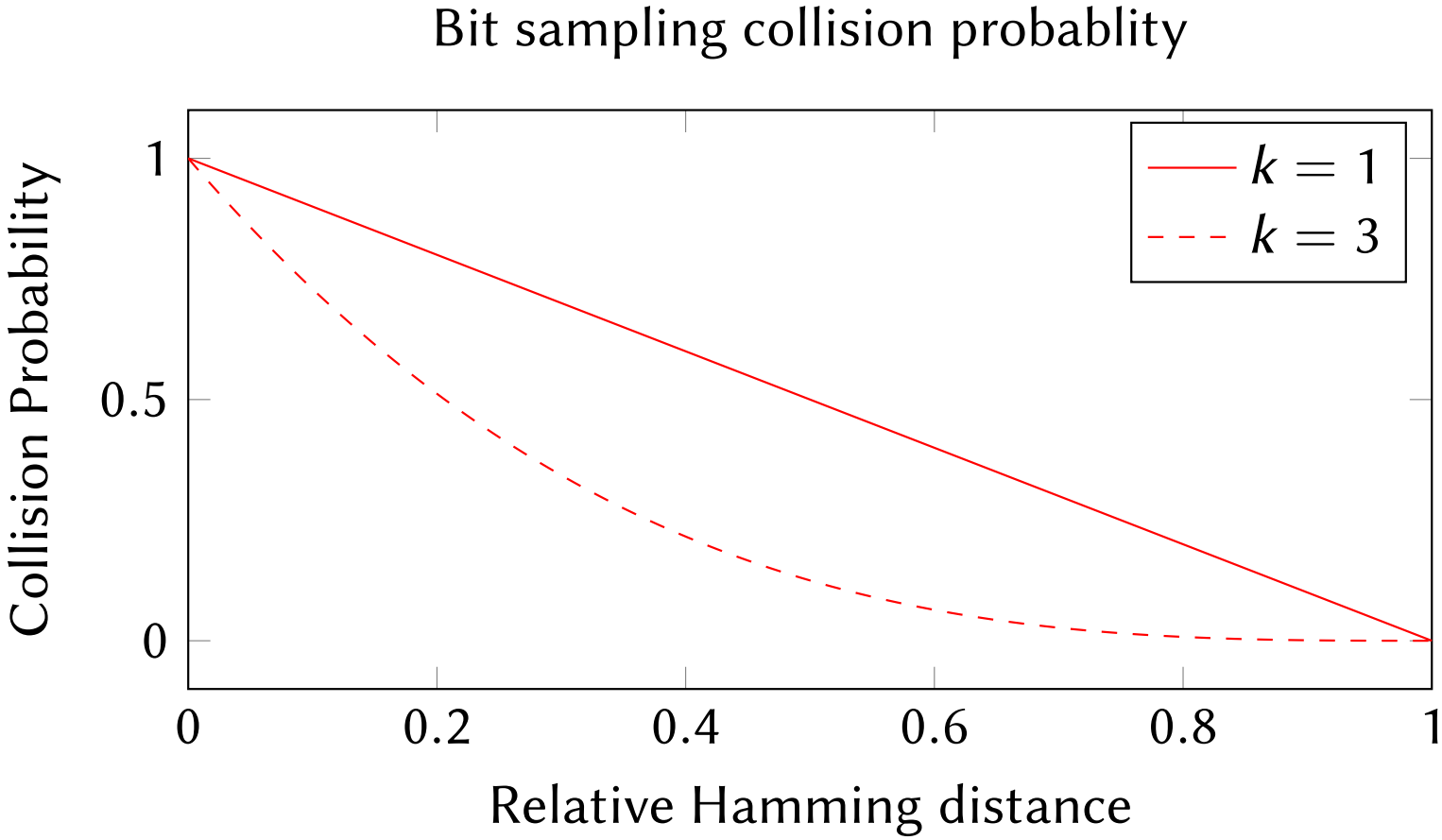


Bitsampling for $\{0,1\}^d$ (IM98)

- **space**: $\{0,1\}^d$ – length-d bitstrings
- $\mathrm{dist}(\mathbf{x},\mathbf{y})$ = (#positions where $\mathbf{x}$ and $\mathbf{y}$ differ)$/d$
  (relative Hamming distance)

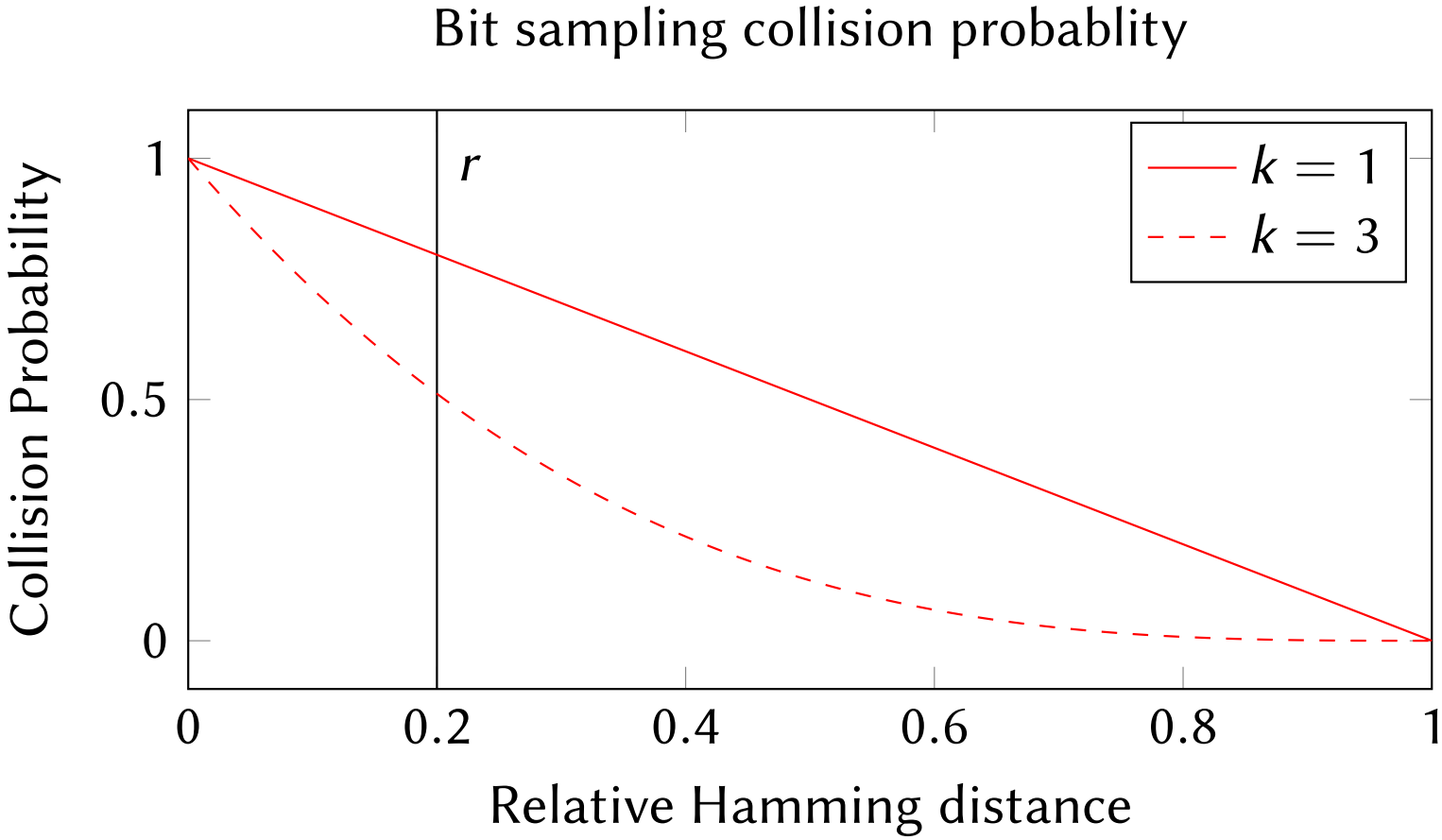
Bitsampling: Project to random subset of dimensions.

$\mathbf{x}$	$0\underline{01}0100\underline{1}0$	$\mathbf{y}$	$1\underline{01}0110\underline{0}\underline{1}0$
$h(\mathbf{x})$	011	$g(\mathbf{y})$	011



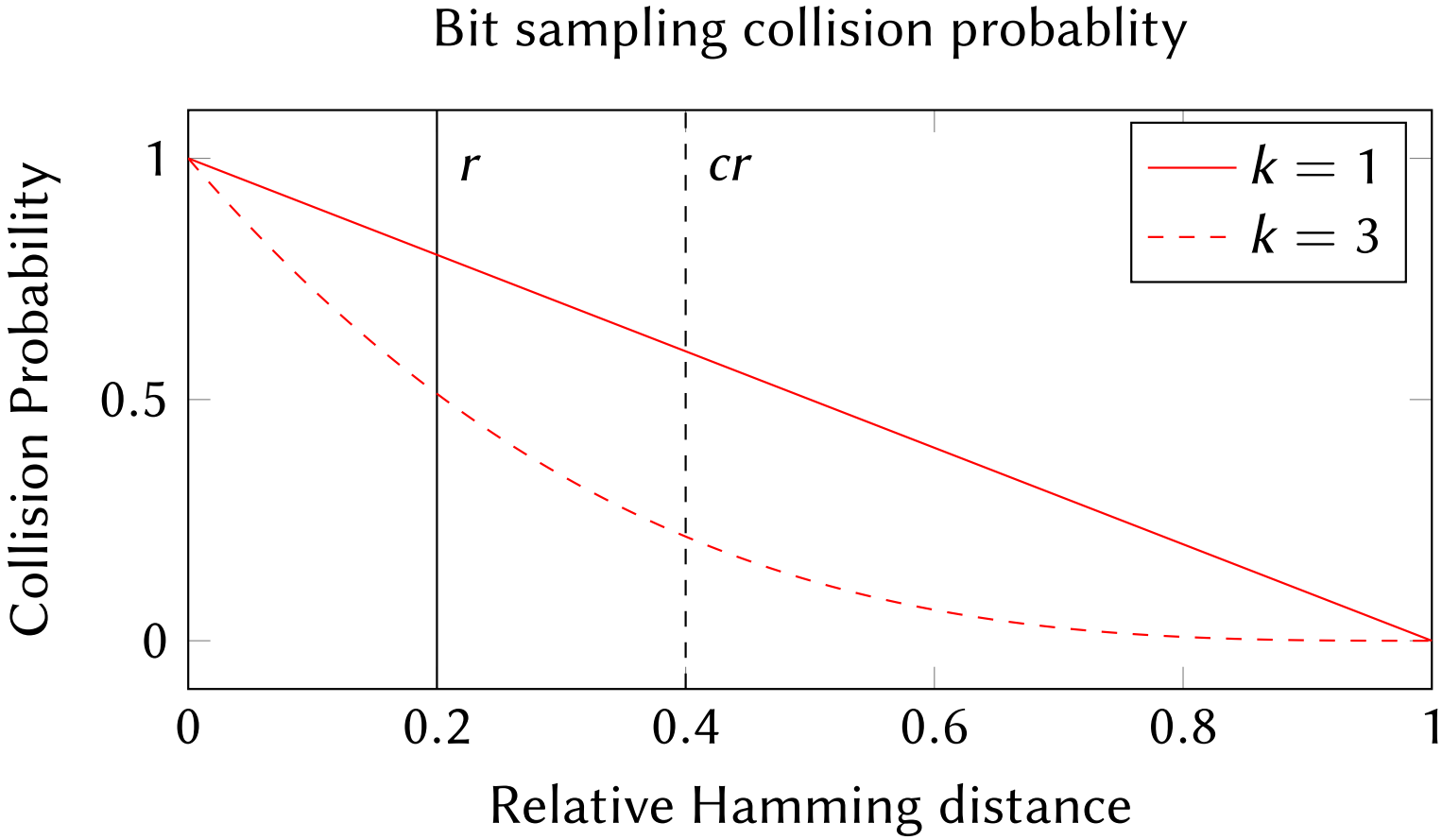
- If $\text{dist}(\mathbf{x}, \mathbf{y}) = r/d$ and we sample k positions, then

$$\mathbb{P}(h(\mathbf{x}) = g(\mathbf{y})) = (1 - r/d)^k.$$



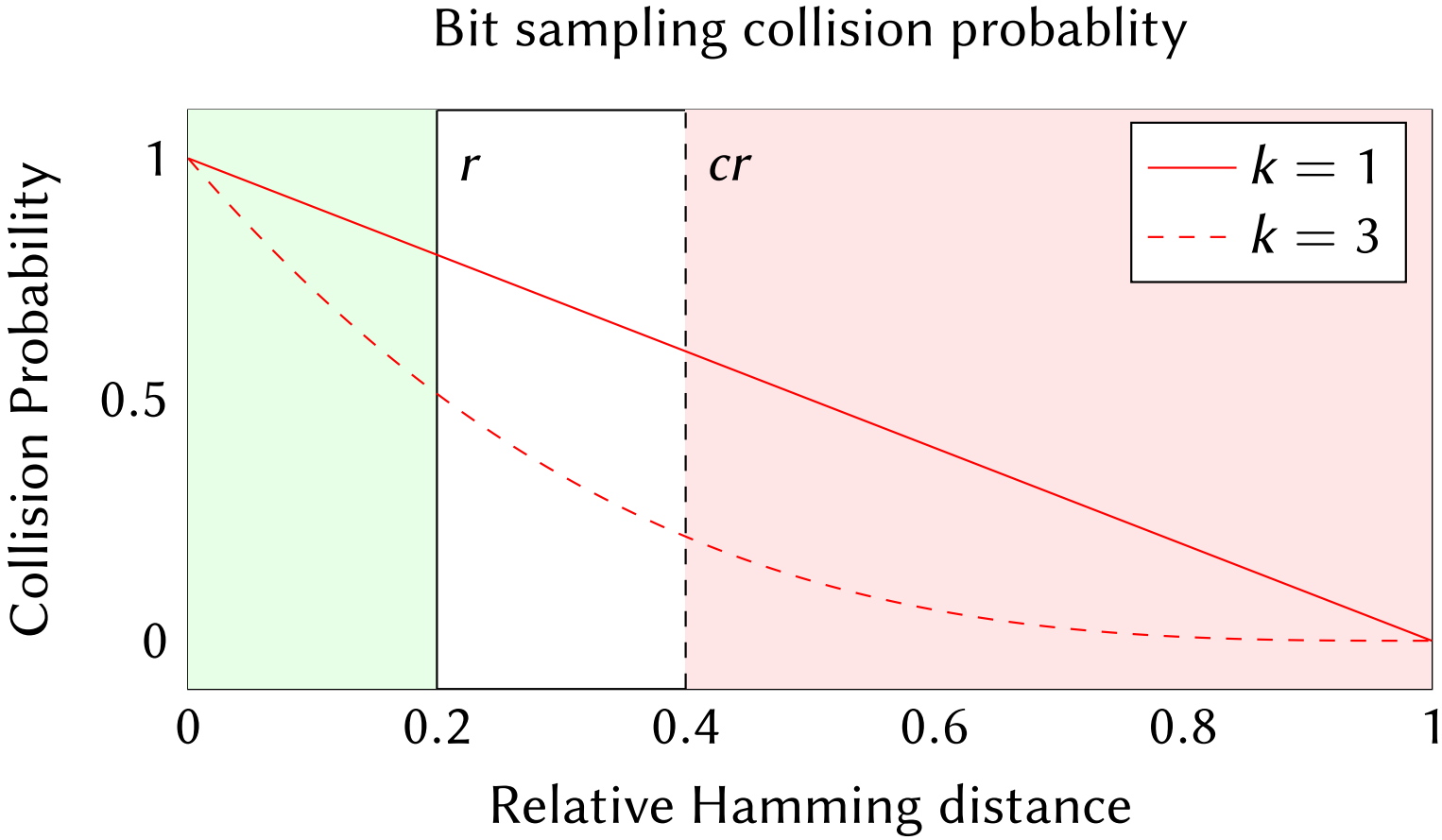
- If $\text{dist}(\mathbf{x}, \mathbf{y}) = r/d$ and we sample k positions, then

$$\mathbb{P}(h(\mathbf{x}) = g(\mathbf{y})) = (1 - r/d)^k.$$



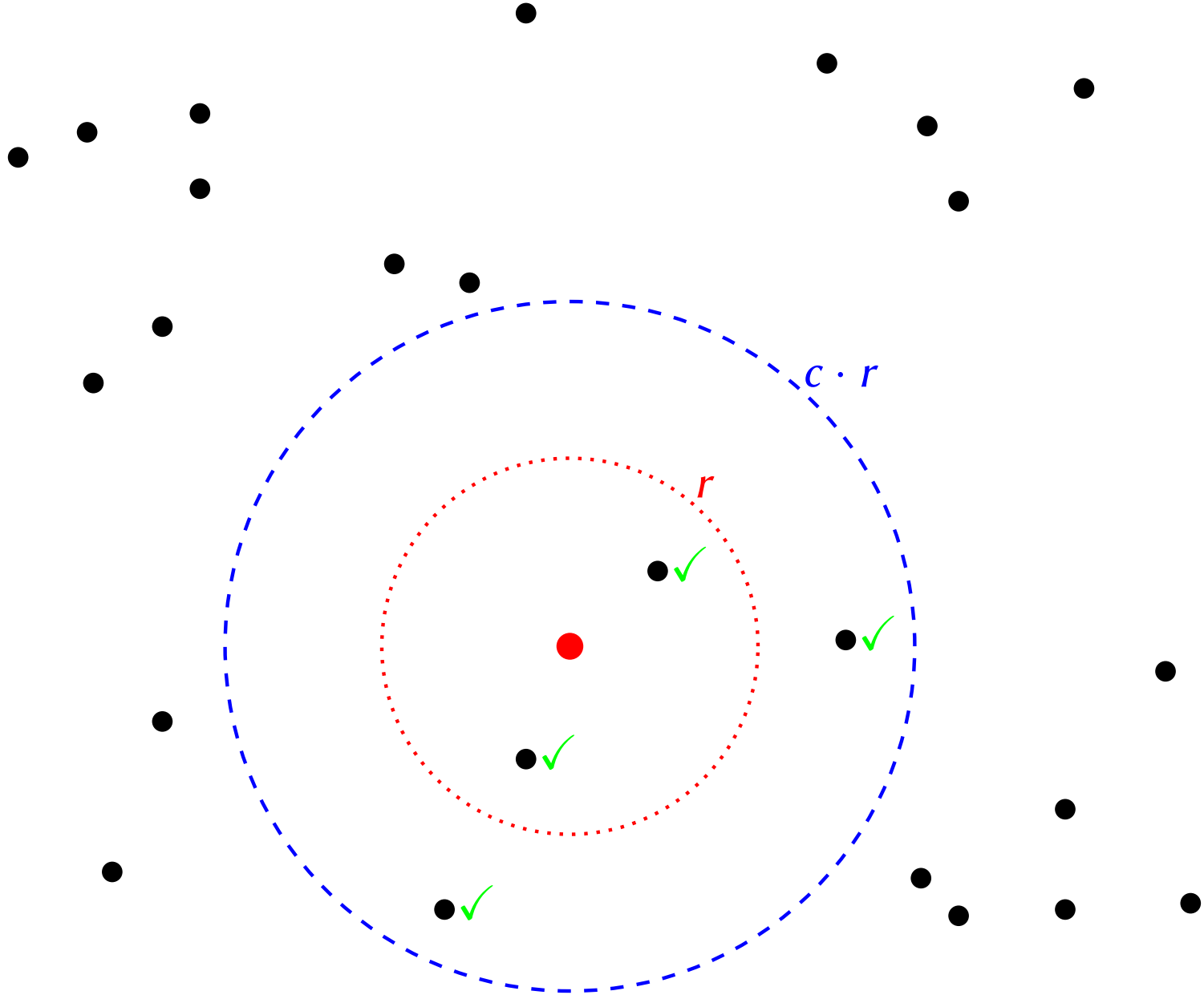
- If $\text{dist}(\mathbf{x}, \mathbf{y}) = r/d$ and we sample k positions, then

$$\mathbb{P}(h(\mathbf{x}) = g(\mathbf{y})) = (1 - r/d)^k.$$



- If $\text{dist}(\mathbf{x}, \mathbf{y}) = r/d$ and we sample k positions, then

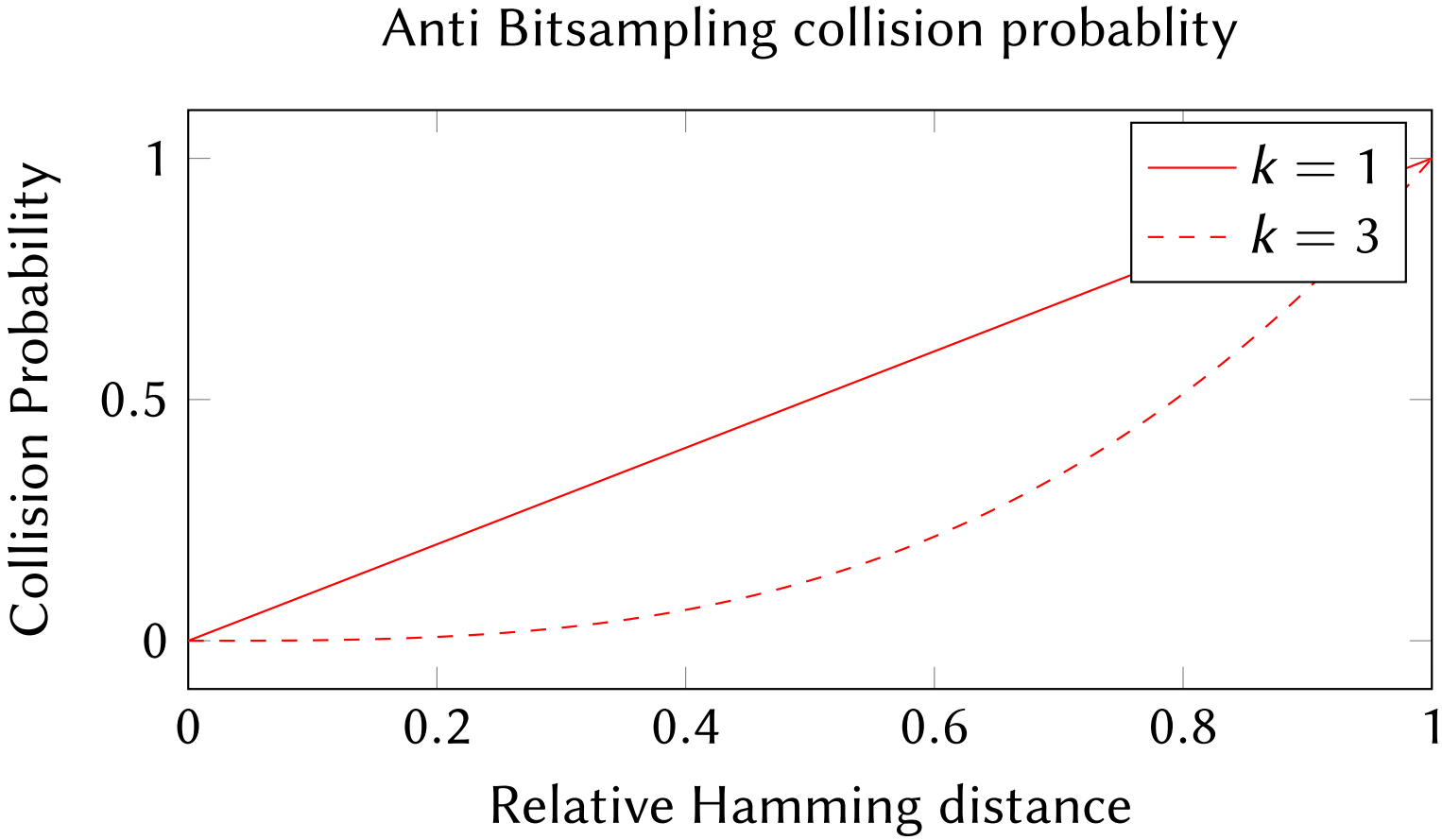
$$\mathbb{P}(h(\mathbf{x}) = g(\mathbf{y})) = (1 - r/d)^k.$$


$c \cdot r$
r

"Anti Bitsampling" for $\{0,1\}^d$

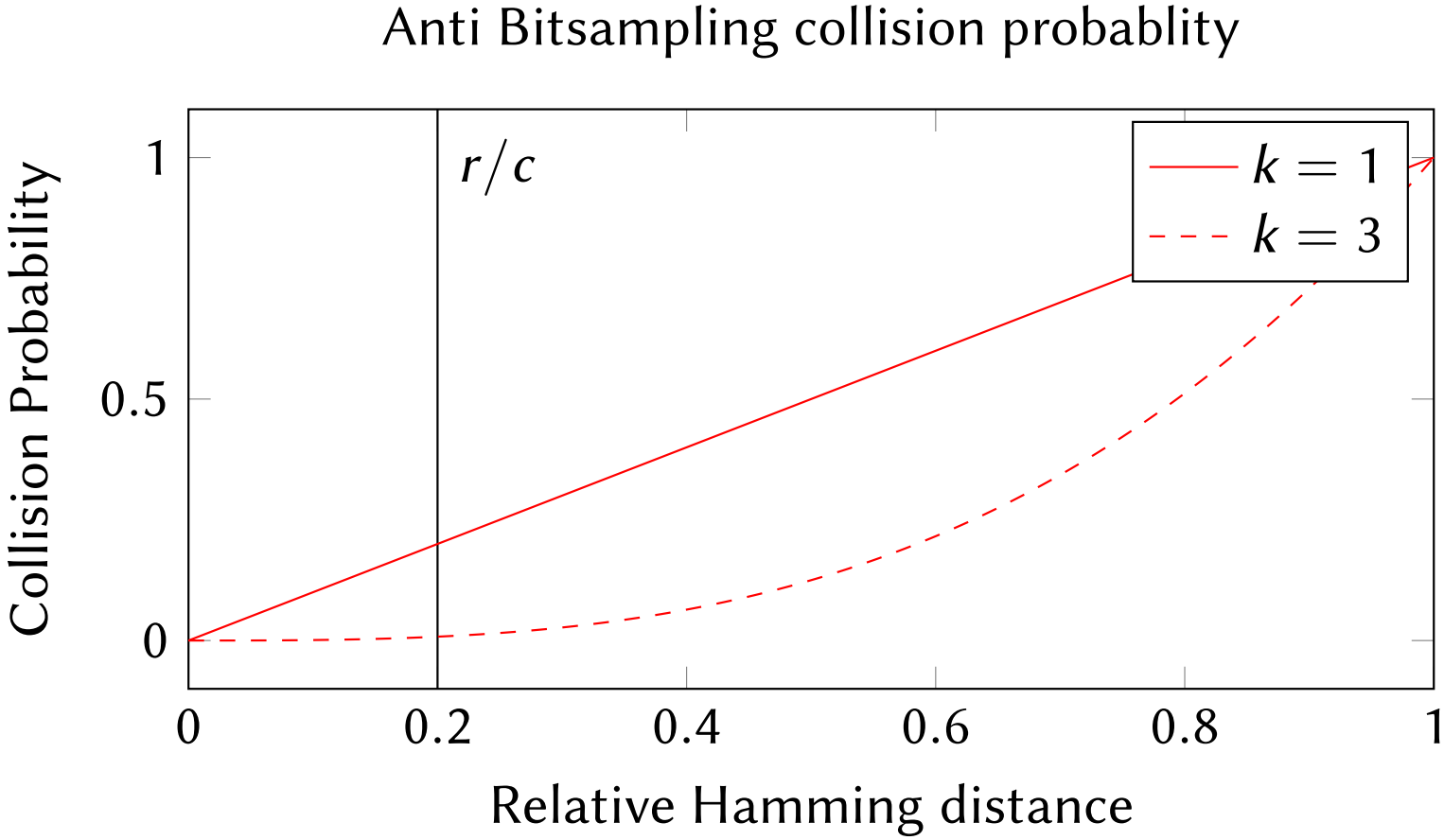
Anti Bitsampling: Project to random subset of dimensions, **flip all query hash code bits**.

$\mathbf{x}$	0<u>01</u>01001<u>0</u>1<u>0</u>	$\mathbf{y}$	1<u>01</u>01100<u>1</u>0
$h(\mathbf{x})$	011	$g(\mathbf{y})$	100



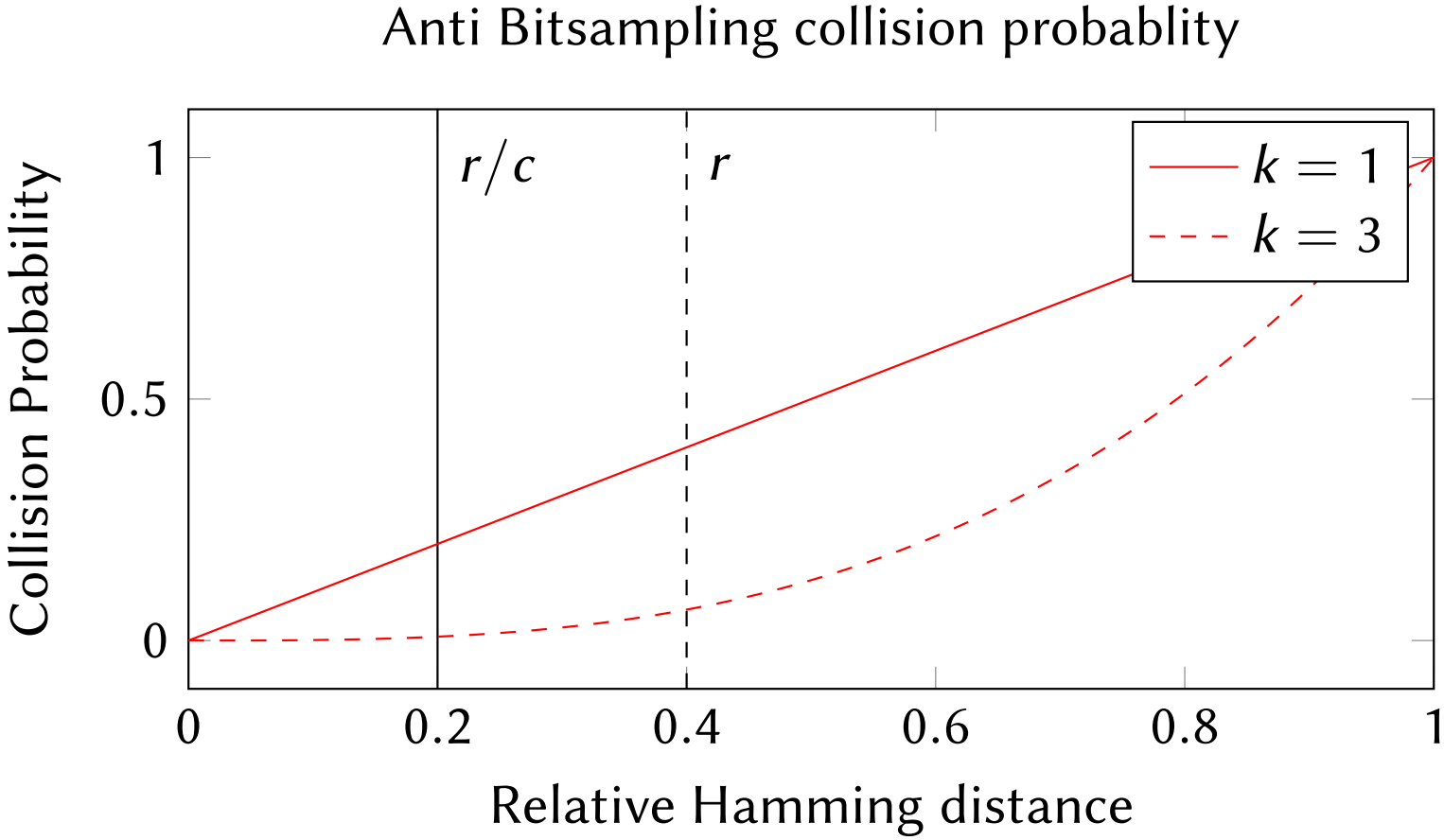
- If $\mathrm{dist}(\mathbf{x}, \mathbf{y}) = r/d$ and we sample k positions, then

$$\mathbb{P}(h(\mathbf{x}) = g(\mathbf{y})) = (r/d)^k.$$



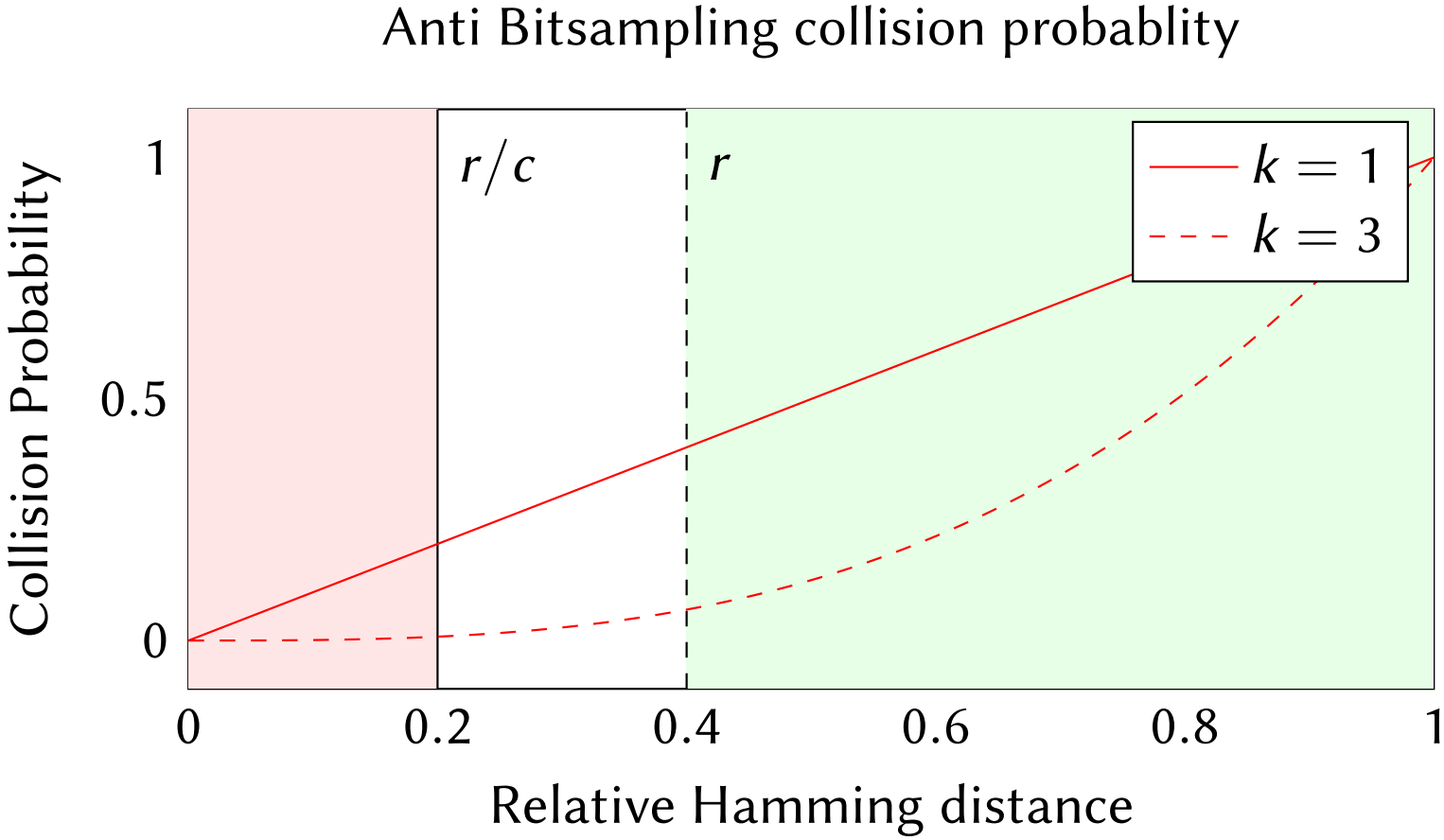
- If $\text{dist}(\mathbf{x}, \mathbf{y}) = r/d$ and we sample k positions, then

$$\mathbb{P}(h(\mathbf{x}) = g(\mathbf{y})) = (r/d)^k.$$



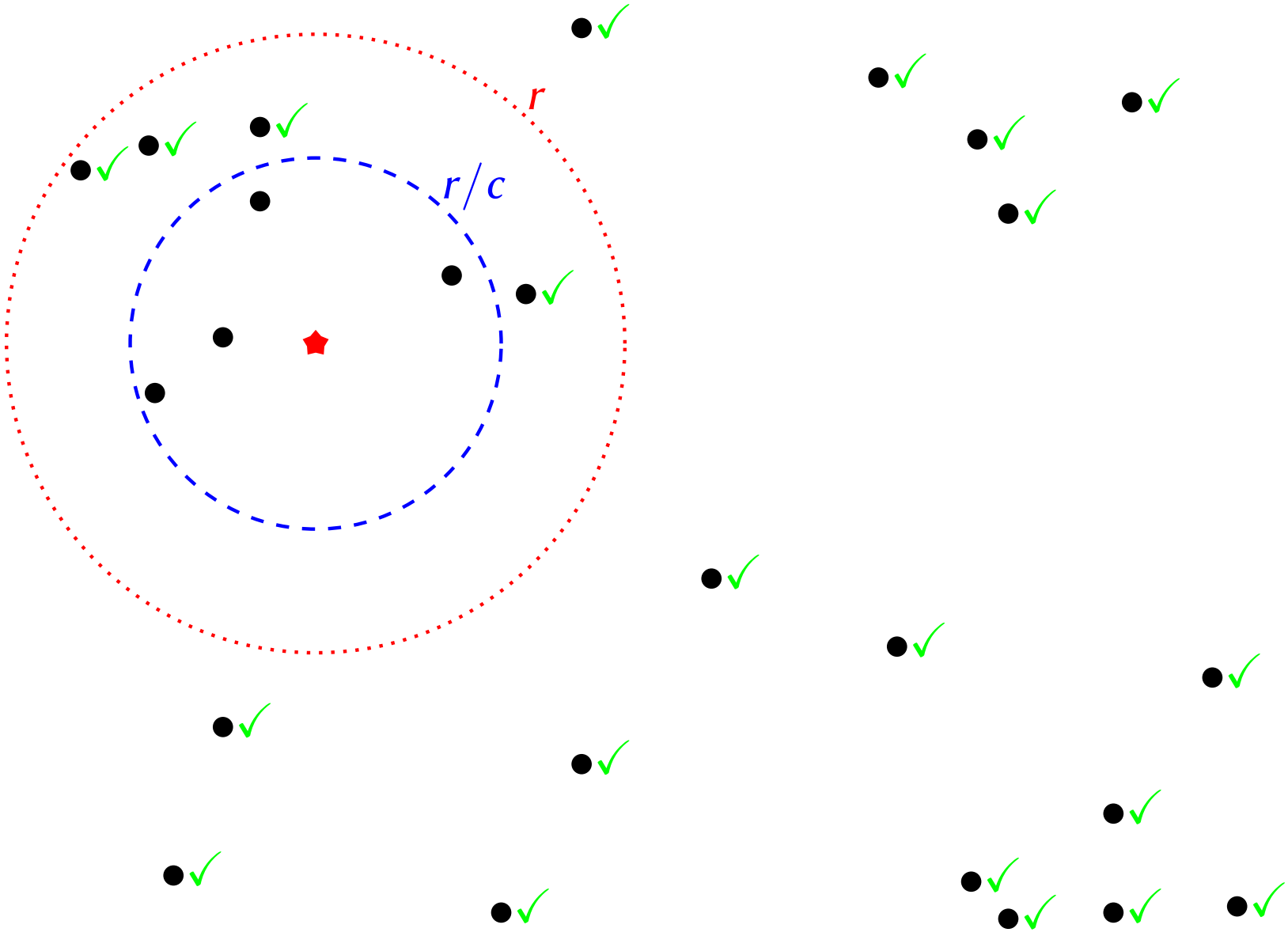
- If $\text{dist}(\mathbf{x}, \mathbf{y}) = r/d$ and we sample k positions, then

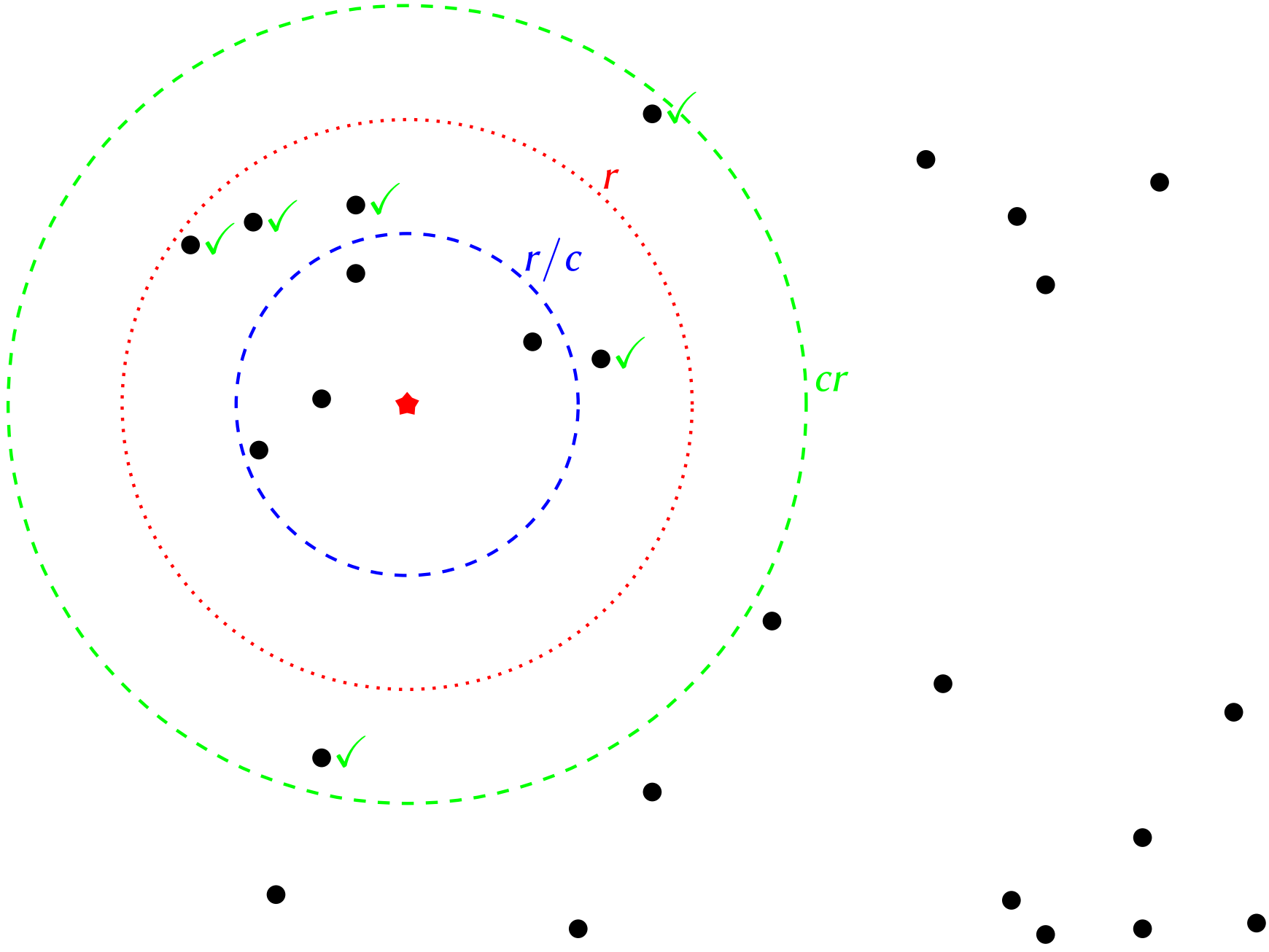
$$\mathbb{P}(h(\mathbf{x}) = g(\mathbf{y})) = (r/d)^k.$$



- If $\mathrm{dist}(\mathbf{x}, \mathbf{y}) = r/d$ and we sample k positions, then

$$\mathbb{P}(h(\mathbf{x}) = g(\mathbf{y})) = (r/d)^k.$$

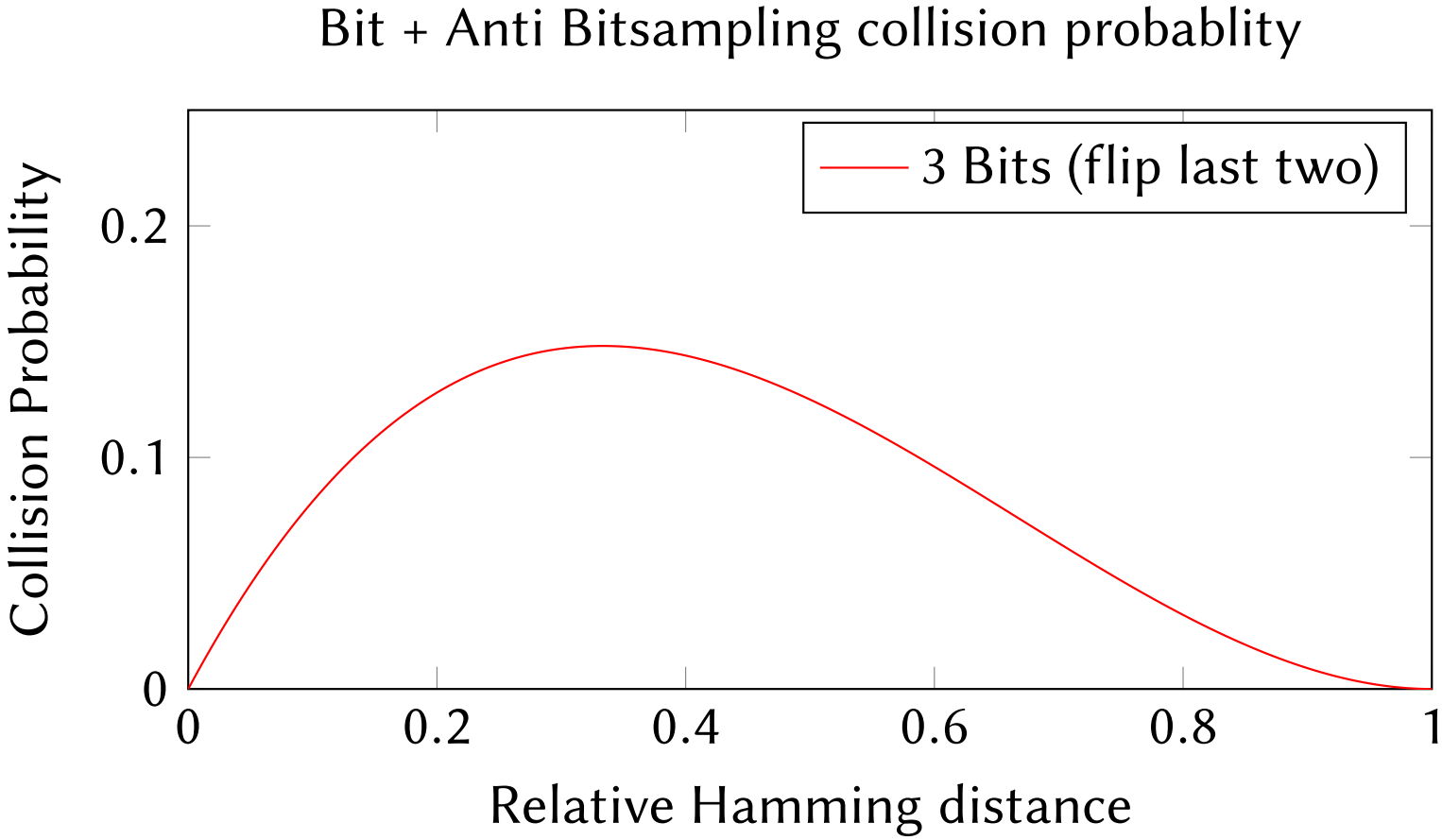

r
r/c


r
r/c
cr

Bitsampling + Anti Bitsampling for $\{0,1\}^d$

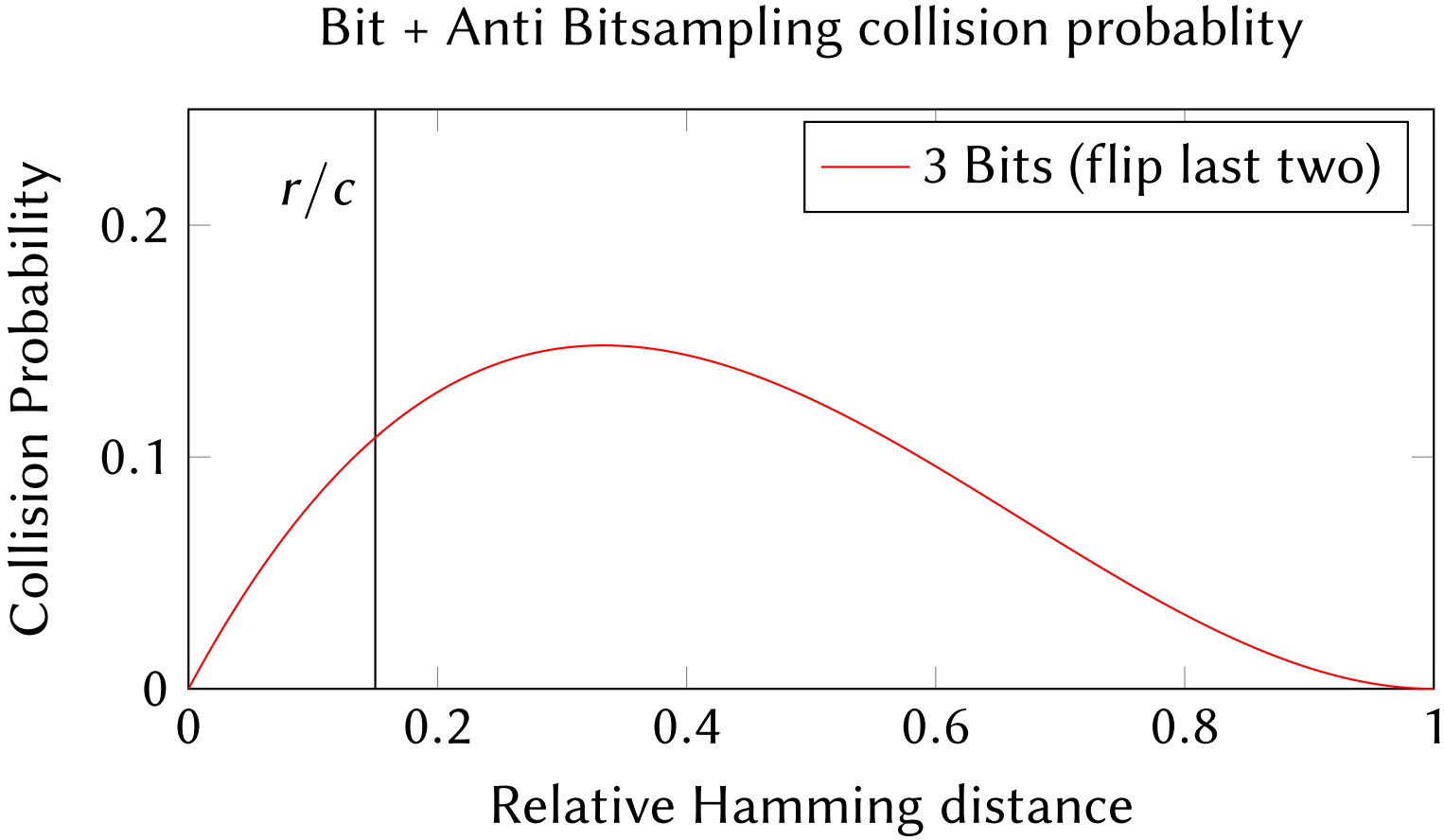
Bit + Anti Bitsampling: Project to random subset of dimensions, **flip some query hash code bits**. (Ex: last two flipped)

$\mathbf{x}$	$0\underline{01}01001\underline{0}1\underline{0}$	$\mathbf{y}$	$1\underline{\mathbf{01}}01100\underline{\mathbf{1}}\underline{0}$
$h(\mathbf{x})$	011	$g(\mathbf{y})$	000



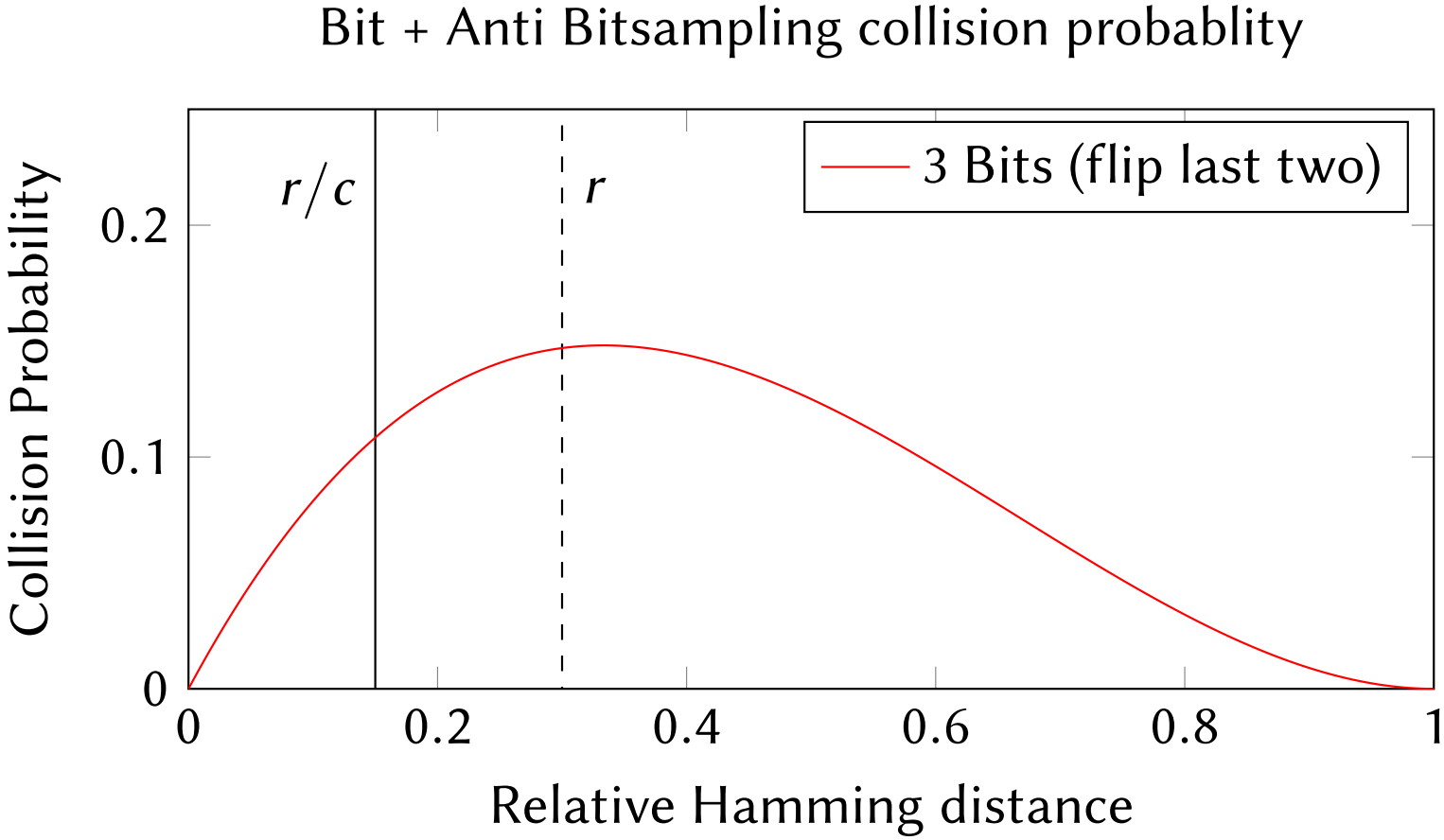
- If $\mathrm{dist}(\mathbf{x}, \mathbf{y}) = r/d$, we sample k positions and flip the last ℓ, then

$$\mathbb{P}(h(\mathbf{x}) = g(\mathbf{y})) = (1 - r/d)^{k-\ell}(r/d)^{\ell}.$$



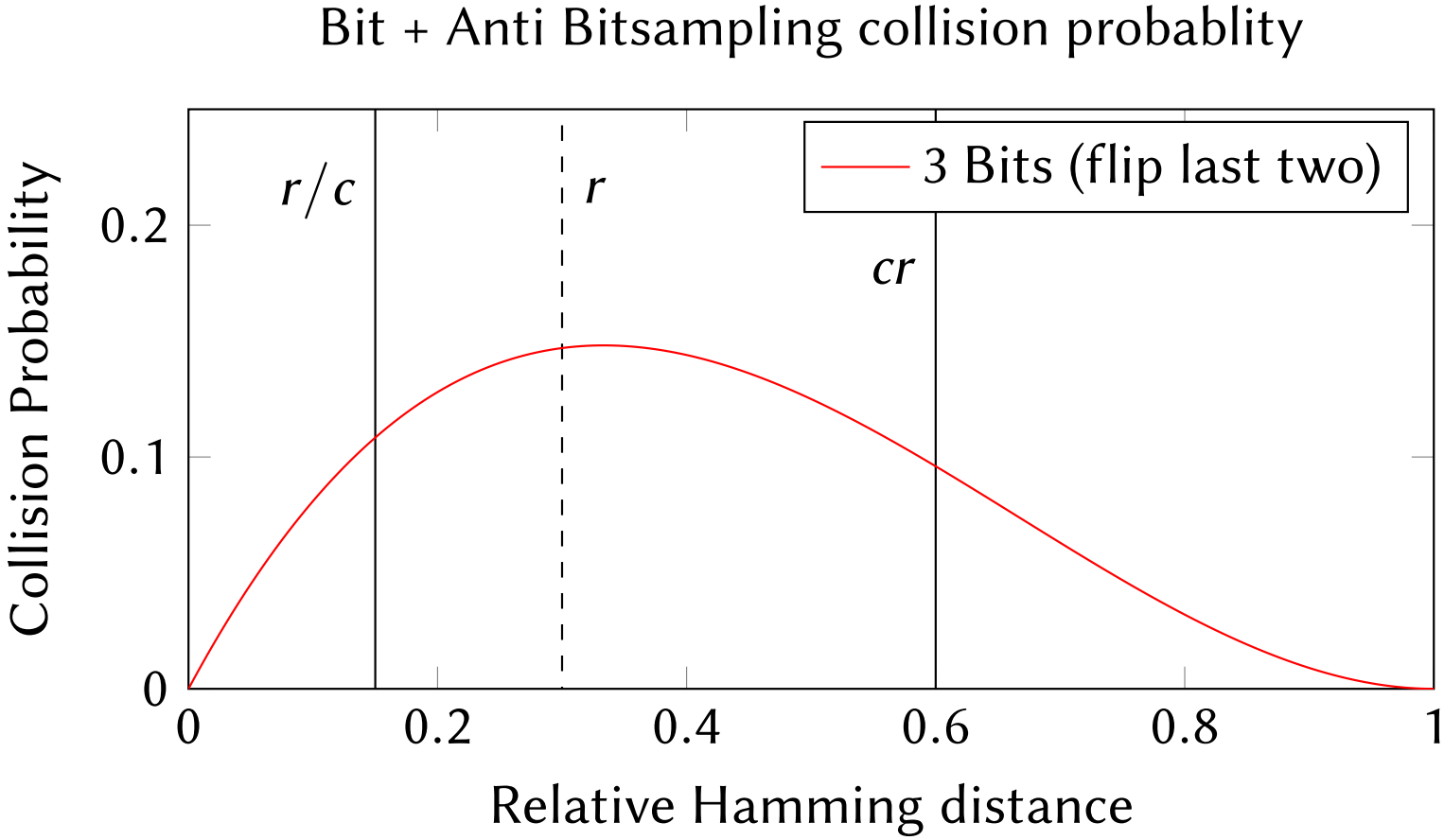
- If $\mathrm{dist}(\mathbf{x}, \mathbf{y}) = r/d$, we sample k positions and flip the last ℓ, then

$$\mathbb{P}(h(\mathbf{x}) = g(\mathbf{y})) = (1 - r/d)^{k-\ell}(r/d)^{\ell}.$$



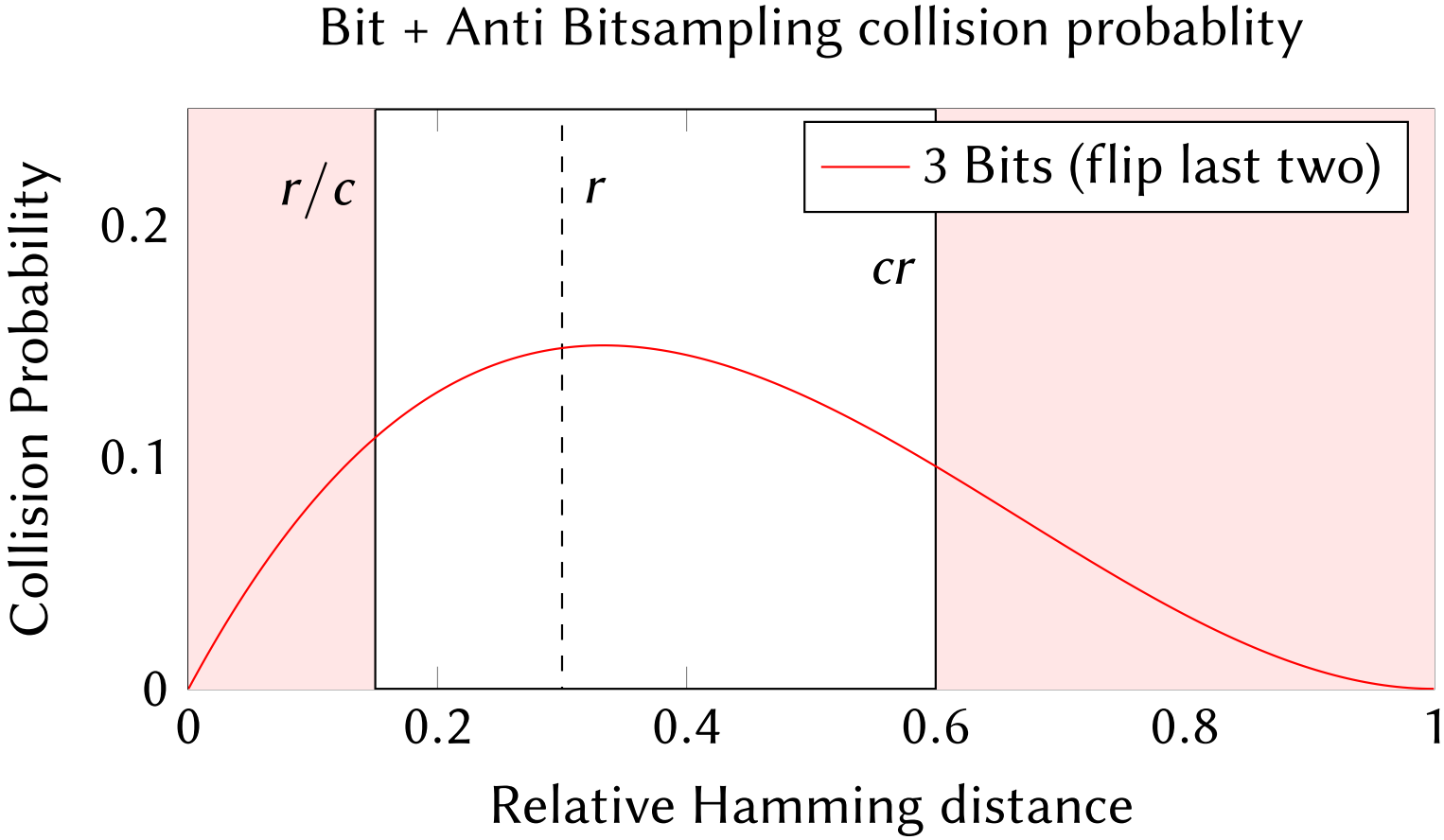
- If $\mathrm{dist}(\mathbf{x}, \mathbf{y}) = r/d$, we sample k positions and flip the last ℓ, then

$$\mathbb{P}(h(\mathbf{x}) = g(\mathbf{y})) = (1 - r/d)^{k-\ell}(r/d)^{\ell}.$$



- If $\mathrm{dist}(\mathbf{x}, \mathbf{y}) = r/d$, we sample k positions and flip the last ℓ, then

$$\mathbb{P}(h(\mathbf{x}) = g(\mathbf{y})) = (1 - r/d)^{k-\ell}(r/d)^{\ell}.$$



- If $\mathrm{dist}(\mathbf{x}, \mathbf{y}) = r/d$, we sample k positions and flip the last ℓ, then

$$\mathbb{P}(h(\mathbf{x}) = g(\mathbf{y})) = (1 - r/d)^{k-\ell}(r/d)^{\ell}.$$

Distance-sensitive Hashing

Assume space X with distance measure $\text{dist}: X \times X \to \mathbb{R}$.

Definition

A distribution $\mathcal{D}$ over pairs of functions $h, g: X \to R$ is called *distance-sensitive* with collision probability function (CPF) $f: \mathbb{R} \to [0, 1]$ if for each pair $\mathbf{x}, \mathbf{y} \in X$

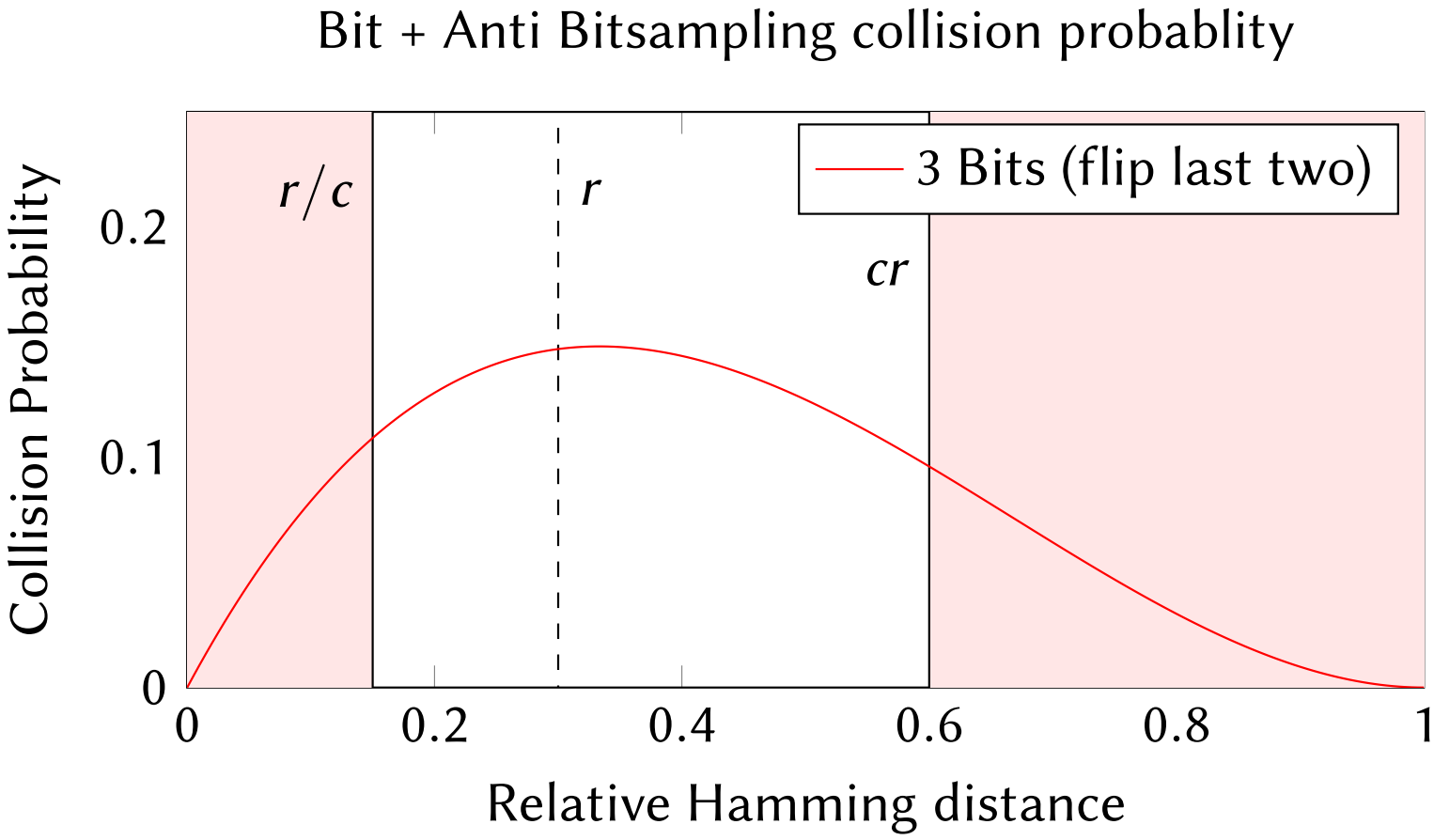
$$\mathbb{P}_{(h,g)\sim\mathcal{D}}[h(\mathbf{x}) = g(\mathbf{y})] = f(\text{dist}(\mathbf{x}, \mathbf{y})).$$

- LSH: f is monotonically decreasing
- Anti-LSH: f is monotonically increasing
- Annulus-LSH: f is unimodal and peaks in $[r/c, cr]$.

Performance for Annulus Queries

- $\rho_- = \frac{\log(f(r))}{\log(f(r/c))}$
- $\rho_+ = \frac{\log(f(r))}{\log(f(cr))}$

- Space: $O(n^{1+\rho_-+\rho_+})$
- Query time: $O(n^{\rho_-+\rho_+})$



Performance of Annulus Queries

Bitsampling

- "optimal" as LSH for $\{0,1\}^d$ under Hamming distance
- $\rho_+ = O(1/c)$

Anti Bitsampling

- $\rho_- = O(1/\log c)$
- Bad at distinguishing "close" and "very close" points.

Query time: $n^{O(1/\log c)}$, worse than LSH for near-neighbor queries.

Can we do better?

Performance of Annulus Queries

Bitsampling

- "optimal" as LSH for $\{0,1\}^d$ under Hamming distance
- $\rho_+ = O(1/c)$

Anti Bitsampling

- $\rho_- = O(1/\log c)$
- Bad at distinguishing "close" and "very close" points.

Query time: $n^{O(1/\log c)}$, worse than LSH for near-neighbor queries.

Can we do better?

$$0011010 \mapsto \frac{1}{\sqrt{d}}(-1,-1,+1,+1,-1,+1,-1)$$

Filter construction (Unit sphere)

(Becker et al. 2016, Christiani 2017, Andoni et al. 2017)

Fix $t > 0$. Sample $\mathbf{a} \sim \mathcal{N}^d(0, 1)$, and

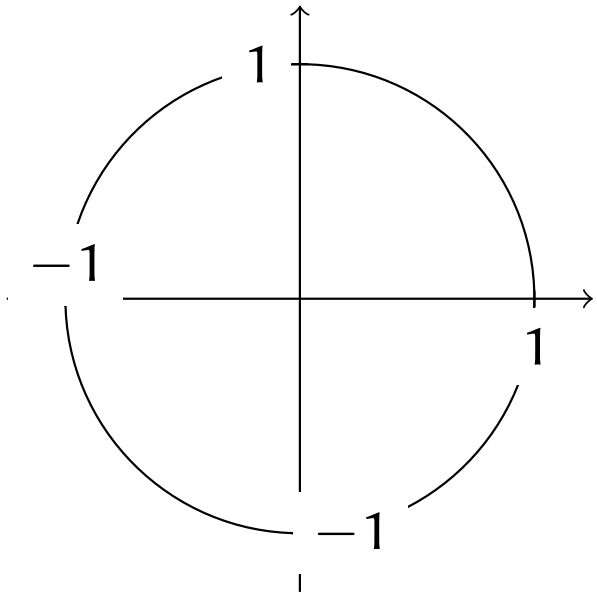
Store $\mathcal{U}_{a,t} = \{\mathbf{x} \in S \mid \langle \mathbf{a}, \mathbf{x} \rangle \geq t\}$

Filter construction (Unit sphere)

(Becker et al. 2016, Christiani 2017, Andoni et al. 2017)

Fix $t > 0$. Sample $\mathbf{a} \sim \mathcal{N}^d(0, 1)$, and

Store $\mathcal{U}_{a,t} = \{\mathbf{x} \in S \mid \langle \mathbf{a}, \mathbf{x} \rangle \geq t\}$

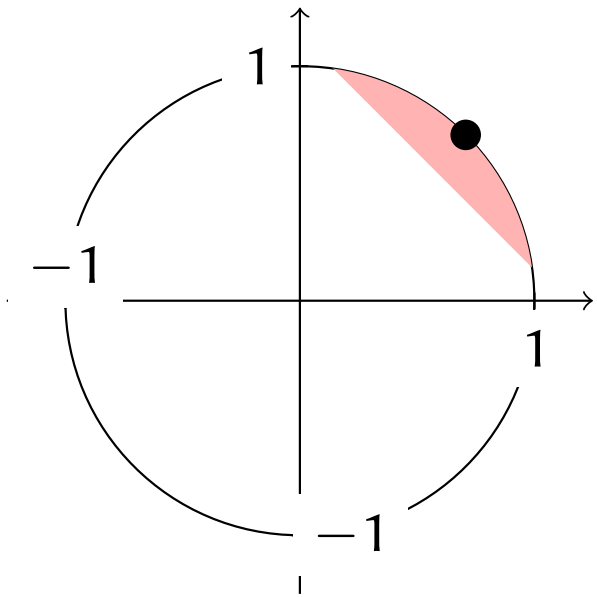


Filter construction (Unit sphere)

(Becker et al. 2016, Christiani 2017, Andoni et al. 2017)

Fix $t > 0$. Sample $\mathbf{a} \sim \mathcal{N}^d(0, 1)$, and

Store $\mathcal{U}_{a,t} = \{\mathbf{x} \in S \mid \langle \mathbf{a}, \mathbf{x} \rangle \geq t\}$

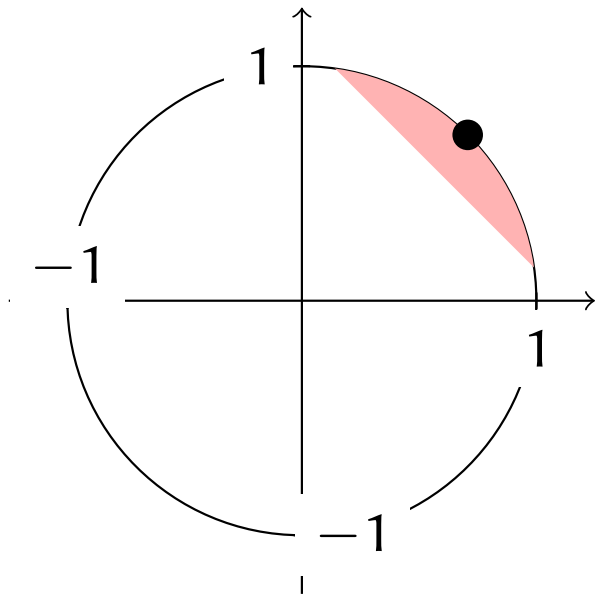


Filter construction (Unit sphere)

(Becker et al. 2016, Christiani 2017, Andoni et al. 2017)

Fix $t > 0$. Sample $\mathbf{a} \sim \mathcal{N}^d(0, 1)$, and

Store $\mathcal{U}_{a,t} = \{\mathbf{x} \in S \mid \langle \mathbf{a}, \mathbf{x} \rangle \geq t\}$ Inspect $\mathcal{U}_{a,t}$ if $\langle \mathbf{a}, \mathbf{y} \rangle \geq t$

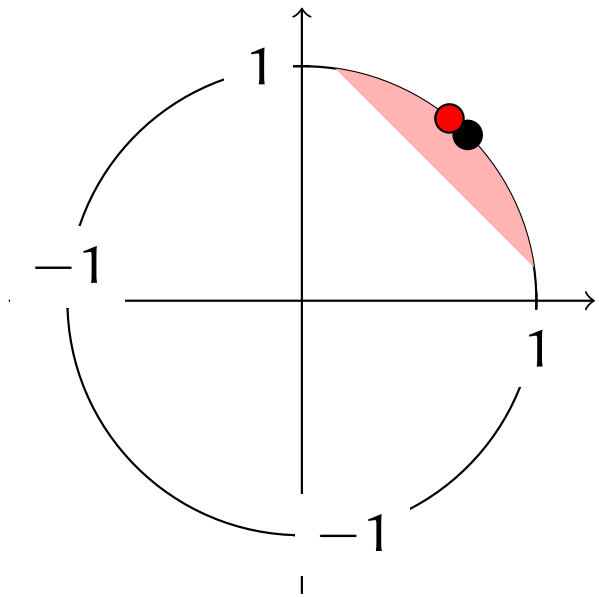


Filter construction (Unit sphere)

(Becker et al. 2016, Christiani 2017, Andoni et al. 2017)

Fix $t > 0$. Sample $\mathbf{a} \sim \mathcal{N}^d(0, 1)$, and

Store $\mathcal{U}_{a,t} = \{\mathbf{x} \in S \mid \langle \mathbf{a}, \mathbf{x} \rangle \geq t\}$ Inspect $\mathcal{U}_{a,t}$ if $\langle \mathbf{a}, \mathbf{y} \rangle \geq t$

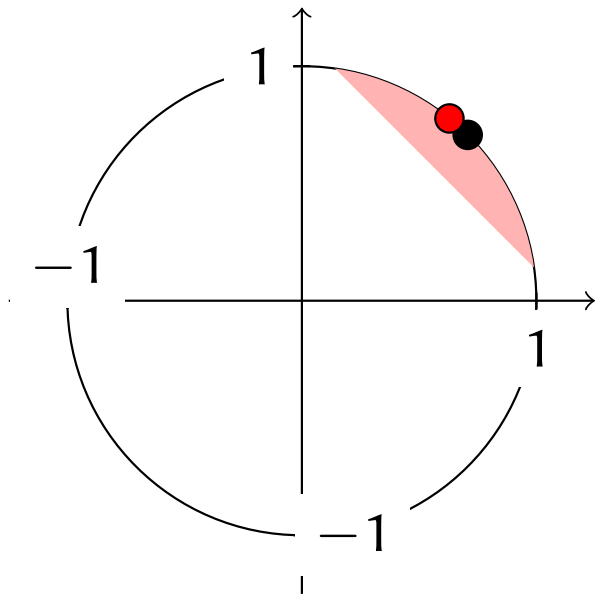


Filter construction (Unit sphere)

(Becker et al. 2016, Christiani 2017, Andoni et al. 2017)

Fix $t > 0$. Sample $\mathbf{a} \sim \mathcal{N}^d(0, 1)$, and

Store $\mathcal{U}_{a,t} = \{\mathbf{x} \in S \mid \langle \mathbf{a}, \mathbf{x} \rangle \geq t\}$ Inspect $\mathcal{U}_{a,t}$ if $\langle \mathbf{a}, \mathbf{y} \rangle \leq -t$

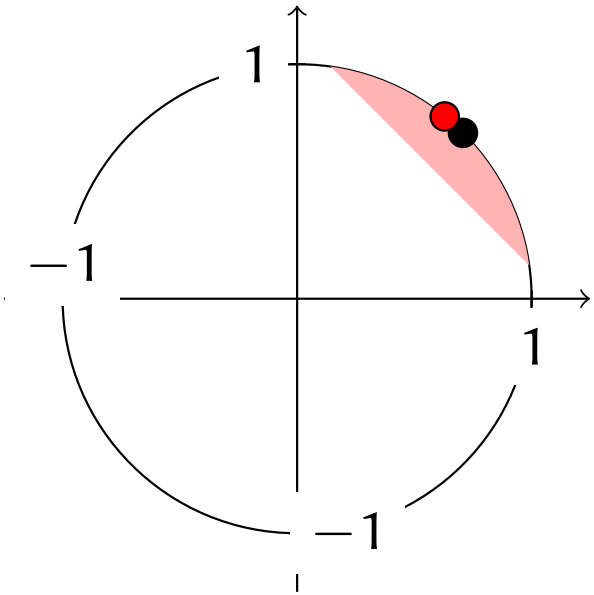


Filter construction (Unit sphere)

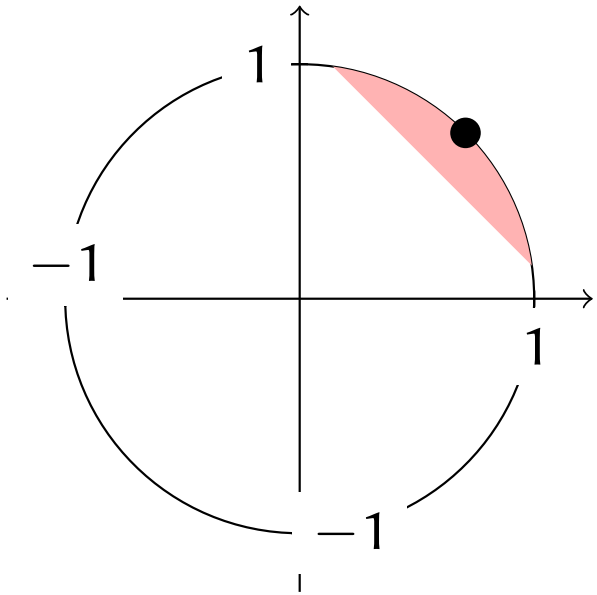
(Becker et al. 2016, Christiani 2017, Andoni et al. 2017)

Fix $t > 0$. Sample $\mathbf{a} \sim \mathcal{N}^d(0, 1)$, and

Store $\mathcal{U}_{a,t} = \{\mathbf{x} \in S \mid \langle \mathbf{a}, \mathbf{x} \rangle \geq t\}$



Inspect $\mathcal{U}_{a,t}$ if $\langle \mathbf{a}, \mathbf{y} \rangle \leq -t$

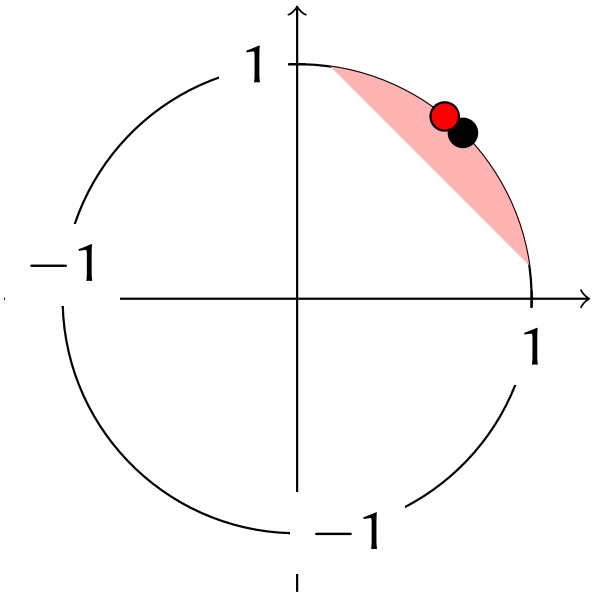


Filter construction (Unit sphere)

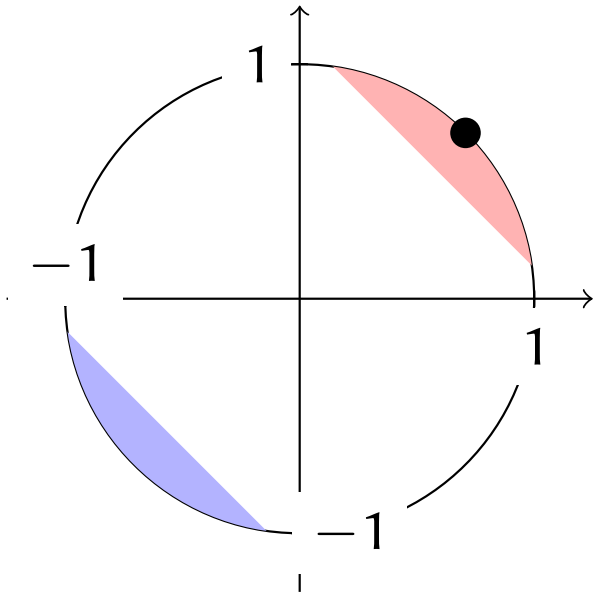
(Becker et al. 2016, Christiani 2017, Andoni et al. 2017)

Fix $t > 0$. Sample $\mathbf{a} \sim \mathcal{N}^d(0, 1)$, and

Store $\mathcal{U}_{a,t} = \{\mathbf{x} \in S \mid \langle \mathbf{a}, \mathbf{x} \rangle \geq t\}$



Inspect $\mathcal{U}_{a,t}$ if $\langle \mathbf{a}, \mathbf{y} \rangle \leq -t$

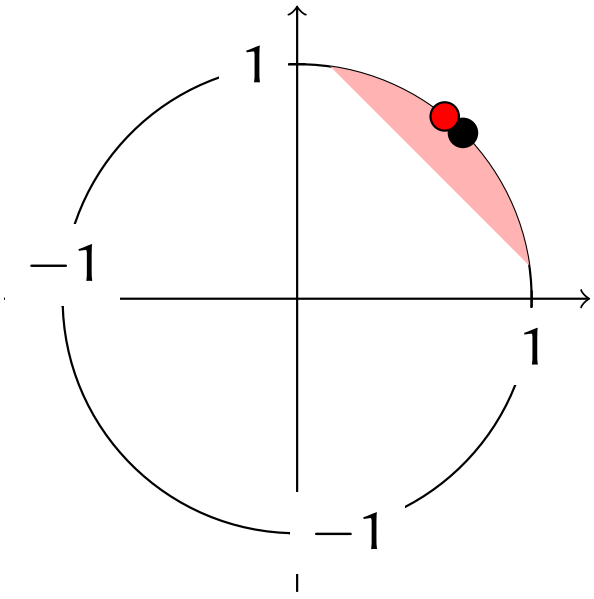


Filter construction (Unit sphere)

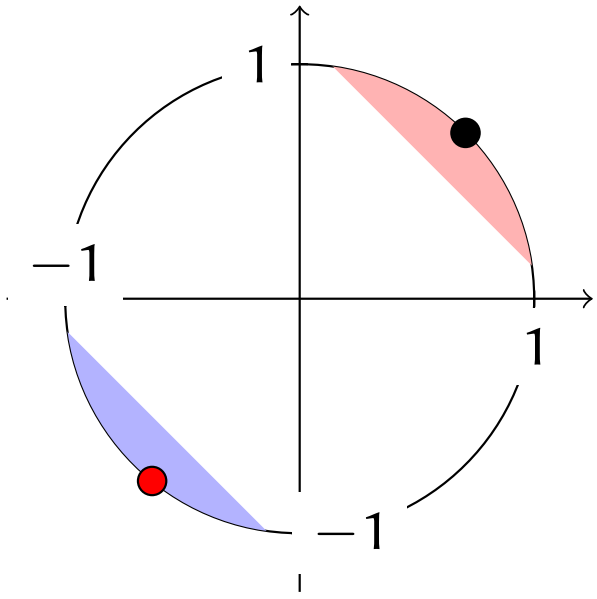
(Becker et al. 2016, Christiani 2017, Andoni et al. 2017)

Fix $t > 0$. Sample $\mathbf{a} \sim \mathcal{N}^d(0, 1)$, and

Store $\mathcal{U}_{a,t} = \{\mathbf{x} \in S \mid \langle \mathbf{a}, \mathbf{x} \rangle \geq t\}$



Inspect $\mathcal{U}_{a,t}$ if $\langle \mathbf{a}, \mathbf{y} \rangle \leq -t$



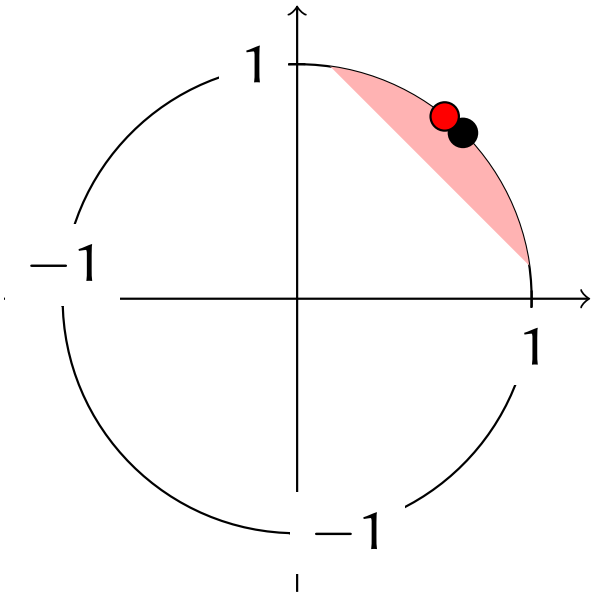
Filter construction (Unit sphere)

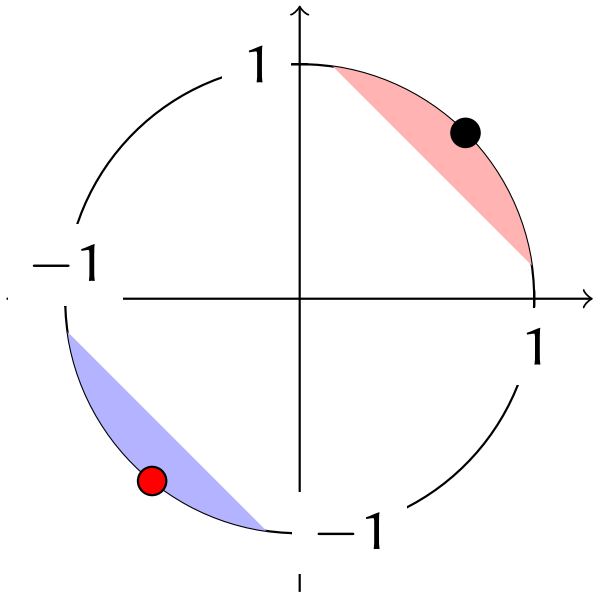
(Becker et al. 2016, Christiani 2017, Andoni et al. 2017)

Fix $t > 0$. Sample $\mathbf{a} \sim \mathcal{N}^d(0, 1)$, and

Store $\mathcal{U}_{a,t} = \{\mathbf{x} \in S \mid \langle \mathbf{a}, \mathbf{x} \rangle \geq t\}$

Inspect $\mathcal{U}_{a,t}$ if $\langle \mathbf{a}, \mathbf{y} \rangle \leq -t$





LSF:

- $\rho_+ = O(1/c^2)$
- optimal

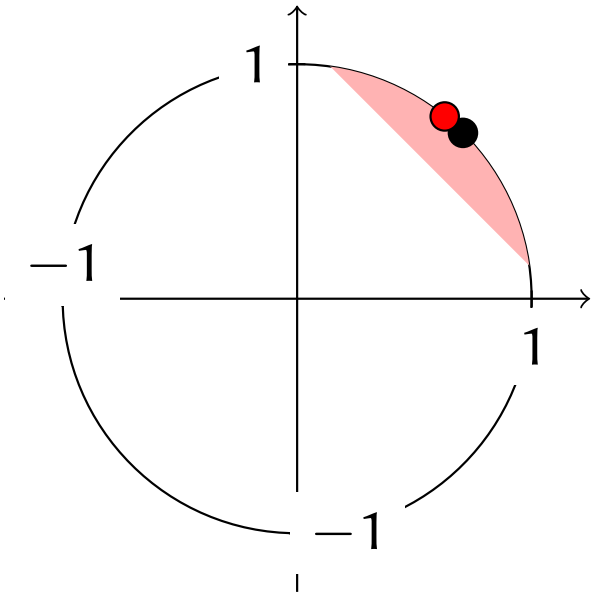
Filter construction (Unit sphere)

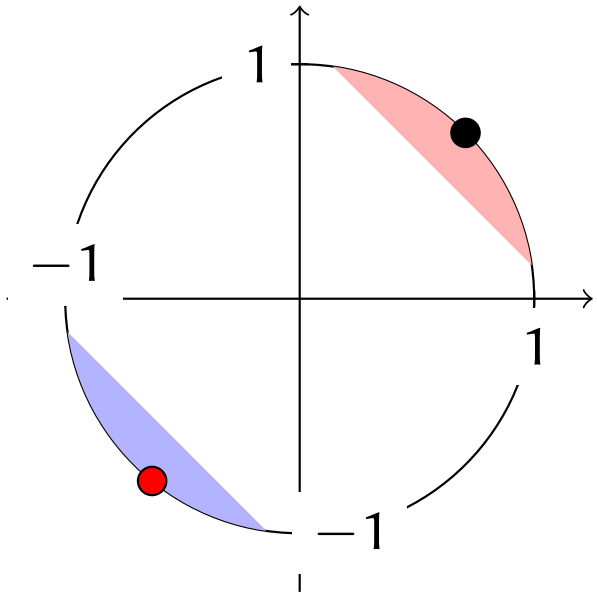
(Becker et al. 2016, Christiani 2017, Andoni et al. 2017)

Fix $t > 0$. Sample $\mathbf{a} \sim \mathcal{N}^d(0, 1)$, and

Store $\mathcal{U}_{a,t} = \{\mathbf{x} \in S \mid \langle \mathbf{a}, \mathbf{x} \rangle \geq t\}$

Inspect $\mathcal{U}_{a,t}$ if $\langle \mathbf{a}, \mathbf{y} \rangle \leq -t$





LSF:

- $\rho_+ = O(1/c^2)$
- optimal

Anti-LSF:

- $\rho_- = O(1/c^2)$
- optimal?

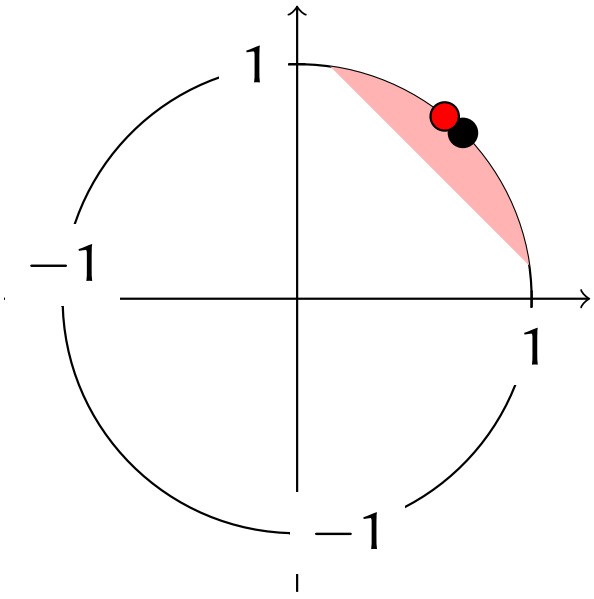
Filter construction (Unit sphere)

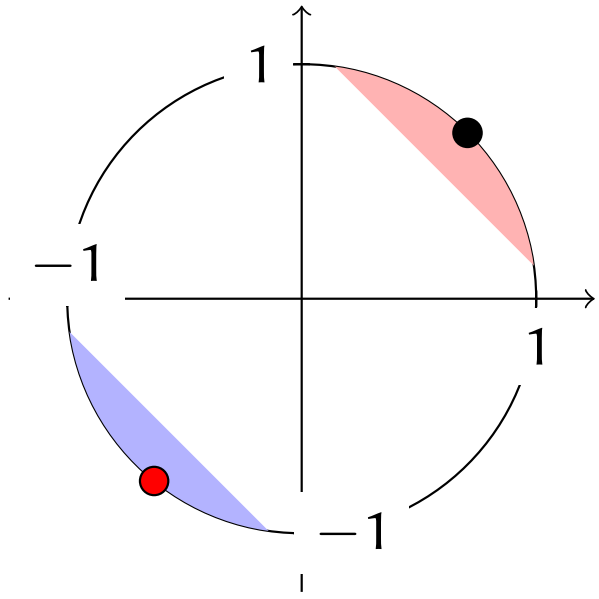
(Becker et al. 2016, Christiani 2017, Andoni et al. 2017)

Fix $t > 0$. Sample $\mathbf{a} \sim \mathcal{N}^d(0, 1)$, and

Store $\mathcal{U}_{a,t} = \{\mathbf{x} \in S \mid \langle \mathbf{a}, \mathbf{x} \rangle \geq t\}$

Inspect $\mathcal{U}_{a,t}$ if $\langle \mathbf{a}, \mathbf{y} \rangle \leq -t$





LSF:

- $\rho_+ = O(1/c^2)$
- optimal

Annulus-LSF:

- $\rho_+ + \rho_- = O(1/c^2)$
- optimal?

Anti-LSF:

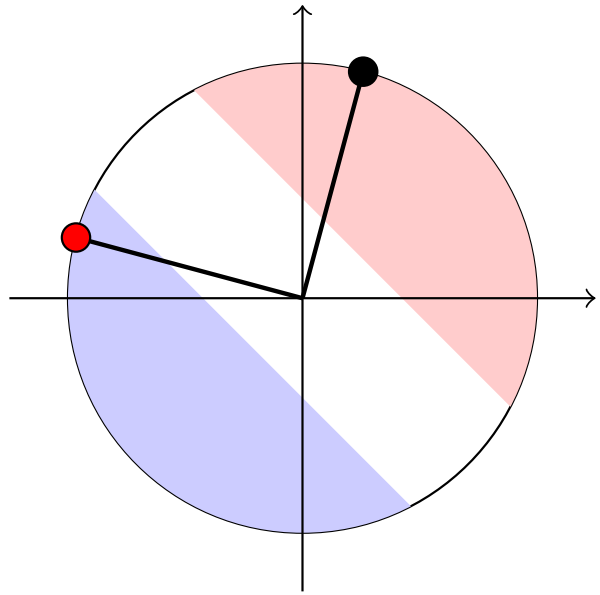
- $\rho_- = O(1/c^2)$
- optimal?

Filter construction (Unit sphere)

(Becker et al. 2016, Christiani 2017, Andoni et al. 2017)

Fix $t > 0$. Sample $\mathbf{a} \sim \mathcal{N}^d(0, 1)$, and

Store $\mathcal{U}_{a,t} = \{\mathbf{x} \in S \mid \langle \mathbf{a}, \mathbf{x} \rangle \geq t\}$ Inspect $\mathcal{U}_{a,t}$ if $\langle \mathbf{a}, \mathbf{y} \rangle \leq -t$



LSF:

- $\rho_+ = O(1/c^2)$
- optimal

Annulus-LSF:

- $\rho_+ + \rho_- = O(1/c^2)$
- optimal?

Anti-LSF:

- $\rho_- = O(1/c^2)$
- optimal?

A Lower Bound on the Unit Sphere

- Setting: distance $\sqrt{2}$ vs. distance $\sqrt{2}/c$ (random vs. correlated).

Theorem

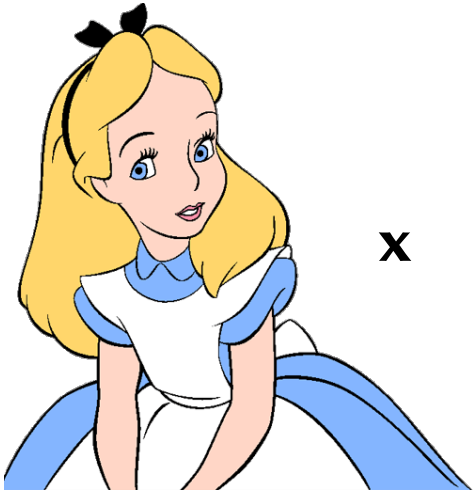
For every distance-sensitive family with CPF f we have

$$\rho_- = \frac{\log(f(\sqrt{2}))}{\log(f(\sqrt{2}/c))} \geq \frac{1}{2c^2 - 1}.$$

(Matches LSH lower bounds, but is **dual** to them.)

Privacy-preserving distance estimation

Problem inspired by (Riazi et al. '16)



$\mathbf{x} \quad \rightarrow \quad \text{dist}(\mathbf{x}, \mathbf{y}) \leq r? \quad \leftarrow \quad \mathbf{y}$

Alice

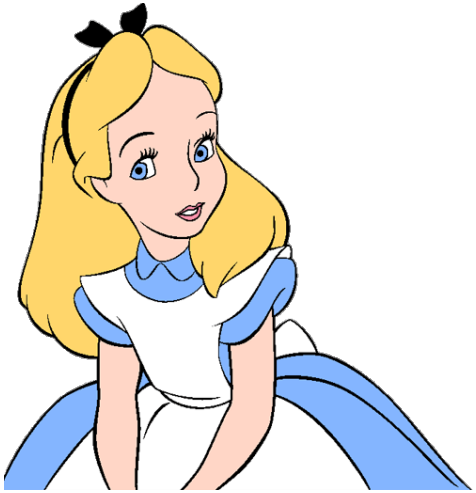
Bob

Goals: Don't reveal vectors, don't reveal actual distance.

For an approximation factor $c > 1$, and parameters $\varepsilon, \delta > 0$:

- $\text{dist}(\mathbf{x}, \mathbf{y}) \leq r$, we say "Yes" with probability at least $1 - \varepsilon$.
- $\text{dist}(\mathbf{x}, \mathbf{y}) \geq cr$, we say "No" with probability at least $1 - \delta$.

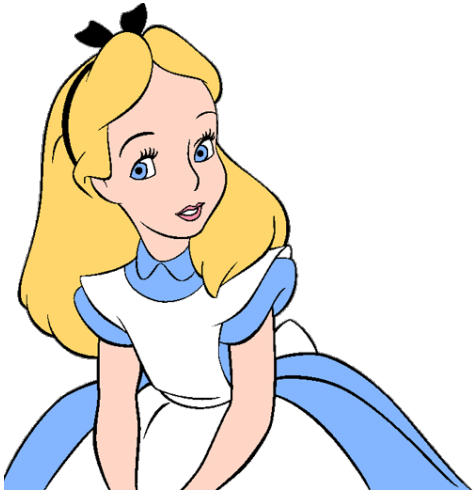
Solution via “step-function CPFs”

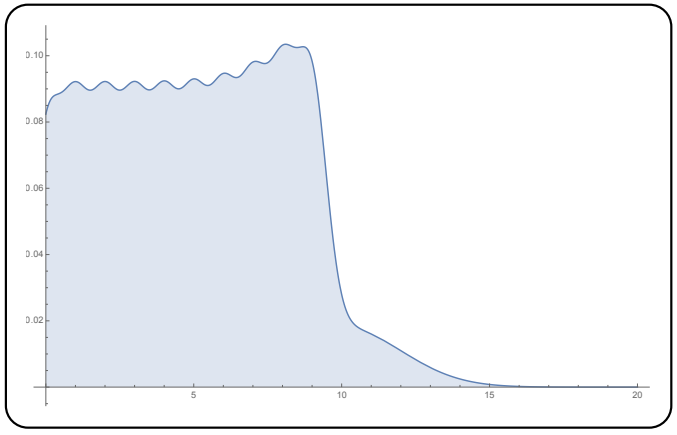


x

y

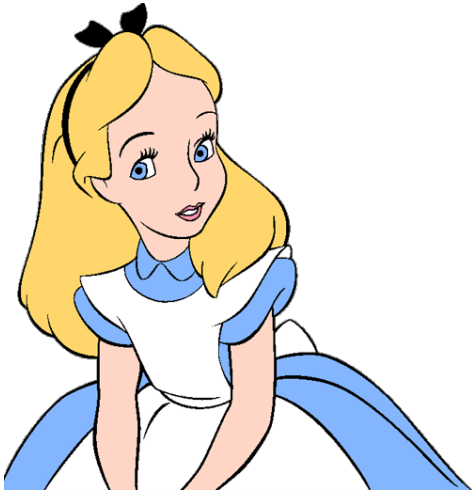
Solution via “step-function CPFs”

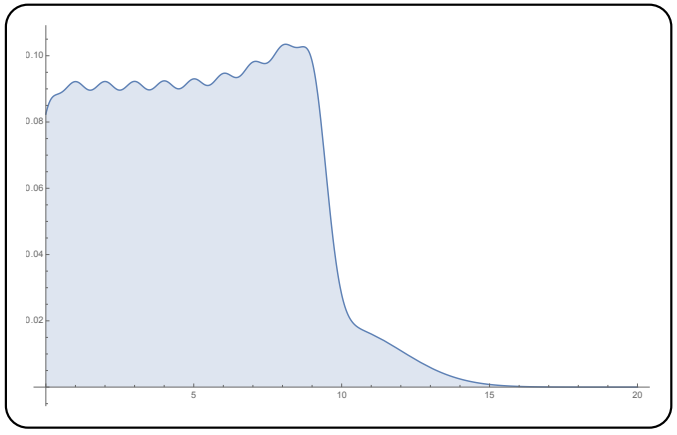




x **y**

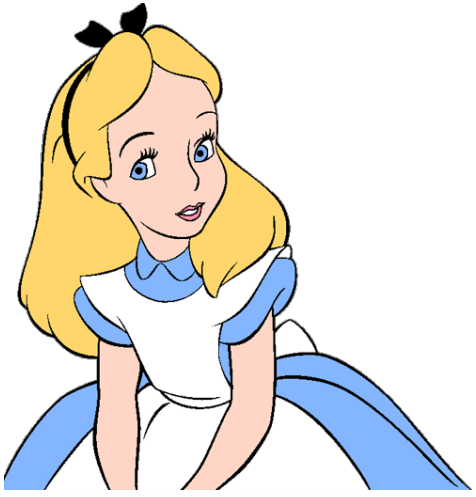
Solution via “step-function CPFs”

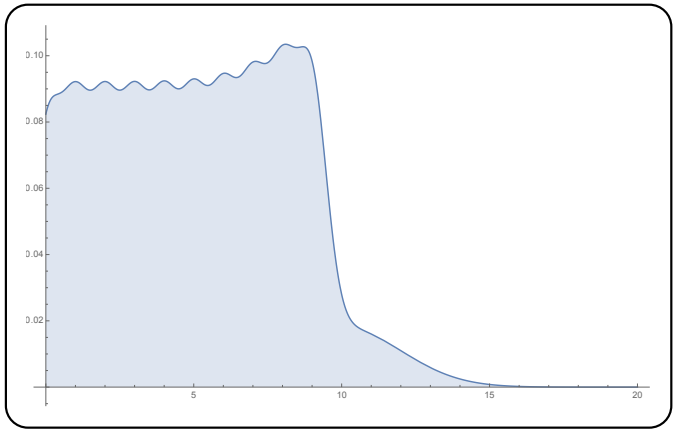




x Choose $(h_1, g_1), \ldots, (h_t, g_t) \sim \mathcal{D}$ **y**

Solution via "step-function CPFs"

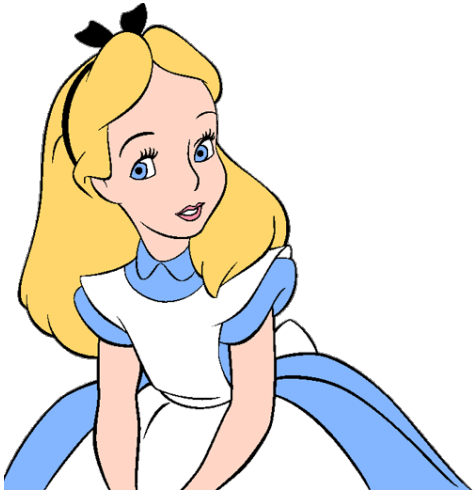


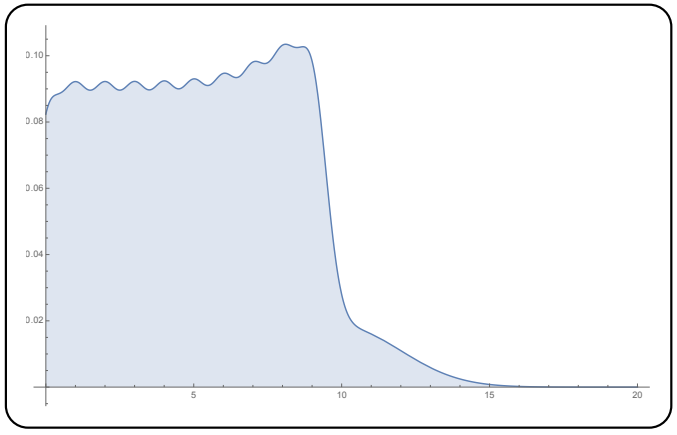


x Choose $(h_1, g_1), \ldots, (h_t, g_t) \sim \mathcal{D}$ **y**

↙ ↘

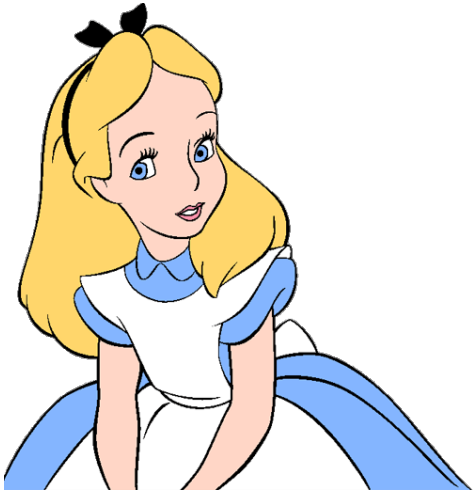
Solution via "step-function CPFs"

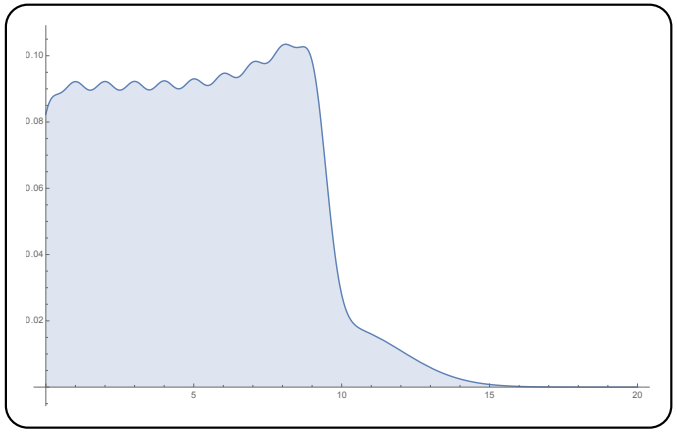




x Choose $(h_1, g_1), \ldots, (h_t, g_t) \sim \mathcal{D}$ **y**

Solution via "step-function CPFs"



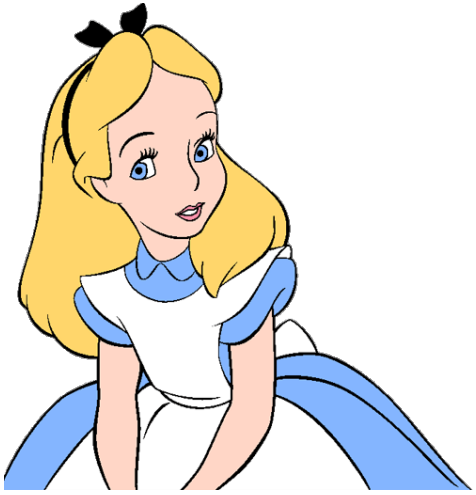


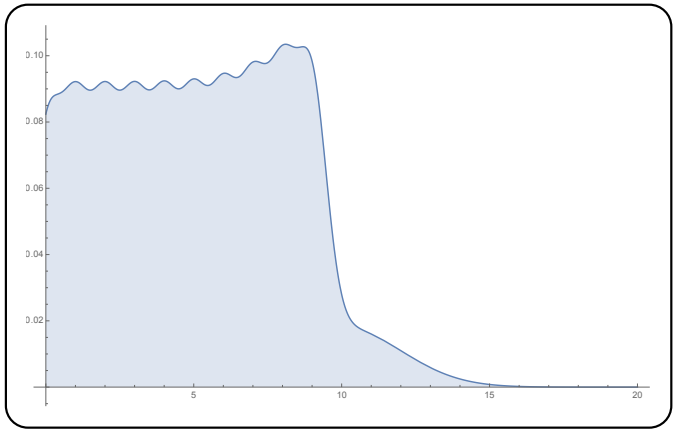
$\mathbf{x}$ Choose $(h_1, g_1), \ldots, (h_t, g_t) \sim \mathcal{D}$ $\mathbf{y}$

$\Downarrow$ $\swarrow$ $\searrow$ $\Downarrow$

$A = \{h_1(\mathbf{x}), \ldots, h_t(\mathbf{x})\}$ $B = \{g_1(\mathbf{y}), \ldots, g_t(\mathbf{y})\}$

Solution via "step-function CPFs"





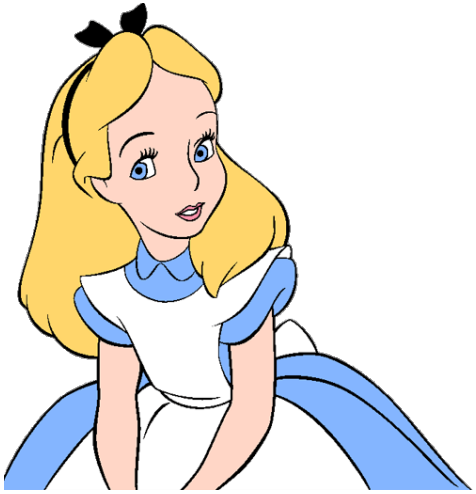
$\mathbf{x}$ Choose $(h_1, g_1), \ldots, (h_t, g_t) \sim \mathcal{D}$ $\mathbf{y}$

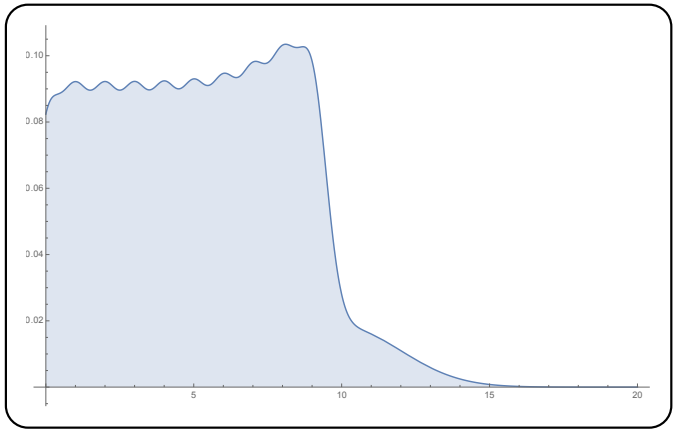
$\Downarrow$ $\swarrow$ $\searrow$ $\Downarrow$

$A = \{h_1(\mathbf{x}), \ldots, h_t(\mathbf{x})\}$ $B = \{g_1(\mathbf{y}), \ldots, g_t(\mathbf{y})\}$

$\searrow$ $\swarrow$

Solution via "step-function CPFs"



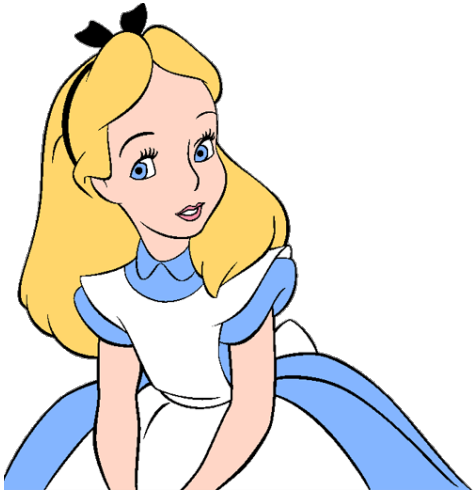


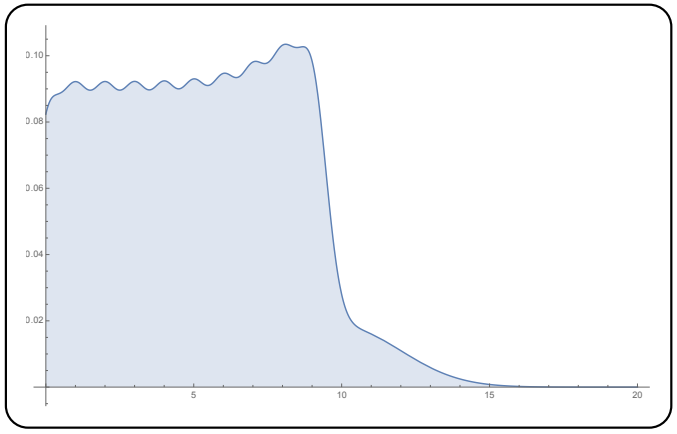
$\mathbf{x}$ Choose $(h_1, g_1), \ldots, (h_t, g_t) \sim \mathcal{D}$ $\mathbf{y}$

$$A = \{h_1(\mathbf{x}), \ldots, h_t(\mathbf{x})\} \qquad B = \{g_1(\mathbf{y}), \ldots, g_t(\mathbf{y})\}$$

"Yes" $\Leftrightarrow$: $A \cap B \neq \emptyset$

Solution via "step-function CPFs"





$\mathbf{x}$ Choose $(h_1, g_1), \ldots, (h_t, g_t) \sim \mathcal{D}$ $\mathbf{y}$

$A = \{h_1(\mathbf{x}), \ldots, h_t(\mathbf{x})\}$ $B = \{g_1(\mathbf{y}), \ldots, g_t(\mathbf{y})\}$

"Yes" $\Leftrightarrow$: $A \cap B \neq \emptyset$

Checked using "privacy-preserving set intersection"

Conclusion & Open Problems

- Introduced notion of "distance-sensitive" hash families
  ... generalization of locality-sensitive hashing
- Applications:
  - annulus queries
  - estimating a function
  - privacy-preserving distance estimation
- Provided some initial constructions and a lower bound on "anti-LSH"

Conclusion & Open Problems

- Introduced notion of "distance-sensitive" hash families
  ... generalization of locality-sensitive hashing
- Applications:
  - annulus queries
  - estimating a function
  - privacy-preserving distance estimation
- Provided some initial constructions and a lower bound on "anti-LSH"

Open:

- Which functions f admit a distance-sensitive hash family?
- Better solution to annulus queries than combining LSH + anti-LSH (which gives $O(n^{2\rho})$ query time)?

Conclusion & Open Problems

- Introduced notion of "distance-sensitive" hash families
  ... generalization of locality-sensitive hashing
- Applications:
  - annulus queries
  - estimating a function
  - privacy-preserving distance estimation
- Provided some initial constructions and a lower bound on "anti-LSH"

Open:

- Which functions f admit a distance-sensitive hash family?
- Better solution to annulus queries than combining LSH + anti-LSH (which gives $O(n^{2\rho})$ query time)?

Thank you!