

Efficient Online Locality Sensitive Hashing via Reservoir Counting

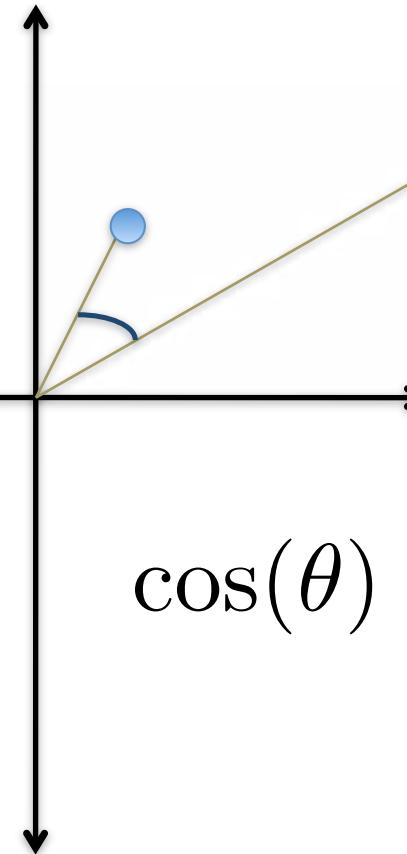
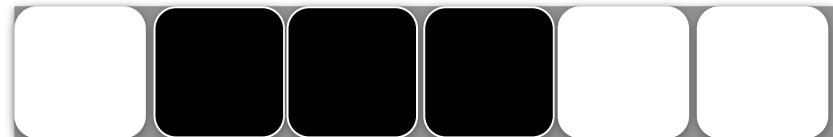
Benjamin Van Durme and Ashwin Lall



human language technology
center of excellence
JOHNS HOPKINS
UNIVERSITY

**DENISON
UNIVERSITY**

Locality Sensitive Hashing

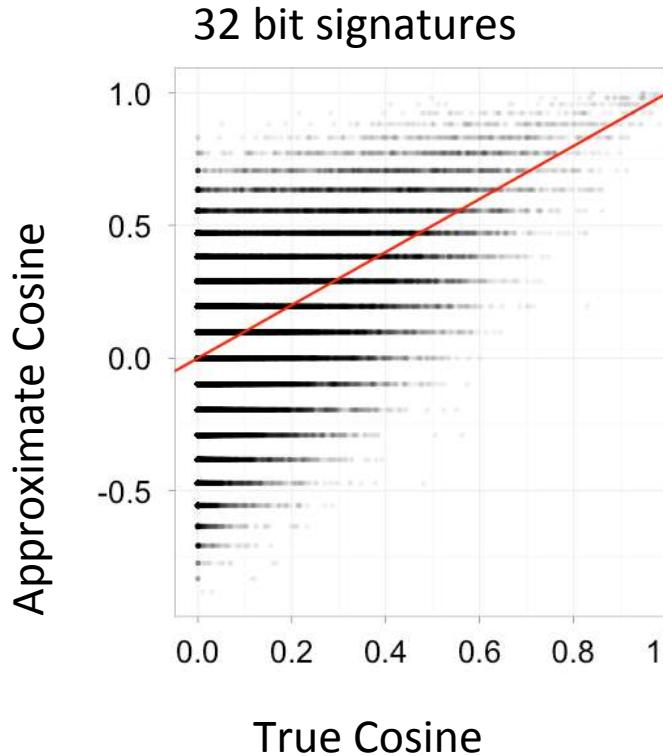


$$\cos(\theta) \approx \cos\left(\frac{h}{b}\pi\right)$$

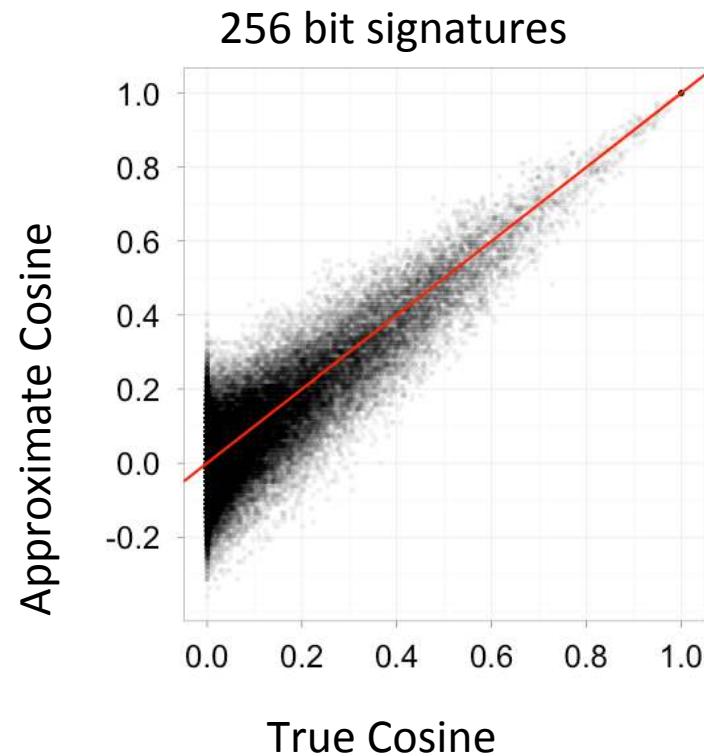
$$= \cos\left(\frac{1}{6}\pi\right)$$

Hamming Distance := $h = 1$
Signature Length := $b = 6$

Accuracy as function of bits



Cheap



Accurate

Some uses in Comp Ling

Noun Clustering

Ravichandran, Hovy and Pantel (2005)

Paraphrase Acquisition

Bhagat and Ravichandran (2008)

Chan, Callison-Burch and Van Durme (2011)

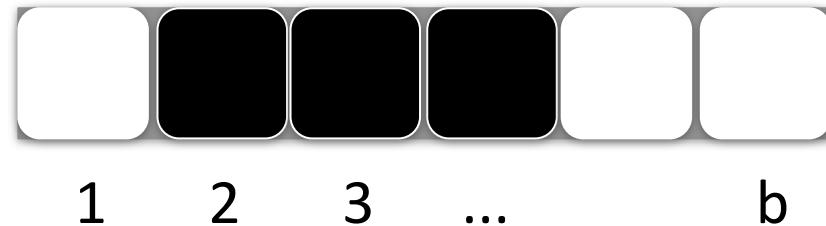
Lexicon Induction

Bergsma and Van Durme (2011)

Topic Detection and Tracking (TDT)

Petrovic, Osborne and Lavrenko (2010)

LSH



LSH



$$\vec{v} \cdot \vec{r}_i \geq 0$$

A gray square frame containing a solid black square, representing a positive inner product result.

$$\vec{v} \cdot \vec{r}_i < 0$$

A gray square frame containing a solid white square, representing a negative inner product result.

Online LSH



$$\sum_t \vec{v}_t \cdot \vec{r}_i \geq 0$$

A gray square frame containing a solid black square. This visual representation indicates that the dot product of the vector sum and the query vector is non-negative.

$$\sum_t \vec{v}_t \cdot \vec{r}_i < 0$$

A gray square frame containing a solid white square. This visual representation indicates that the dot product of the vector sum and the query vector is negative.

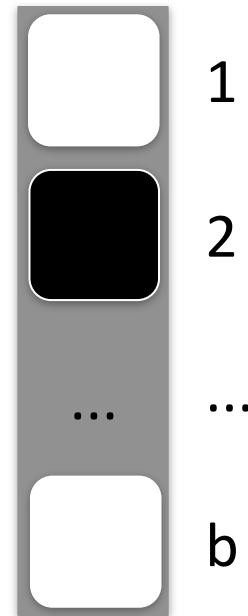
Online LSH : b parallel streams of numbers

$$3 + -2 + 4 + 1 + -1 + \dots < 0$$

$$-5 + -1 + 7 + -2 + 3 + \dots \geq 0$$

...

$$1 + 2 + -4 + 5 + -2 + \dots < 0$$



Stream of numbers

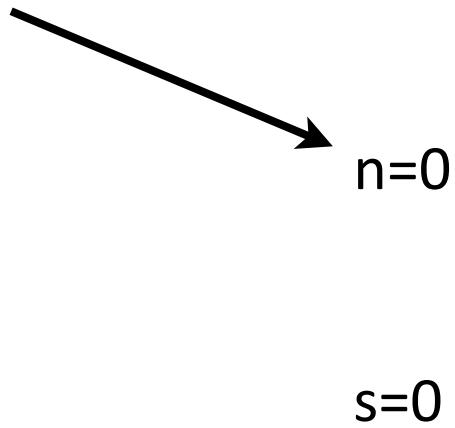
3, -2, 4, 1, -1, ...

Stream of numbers

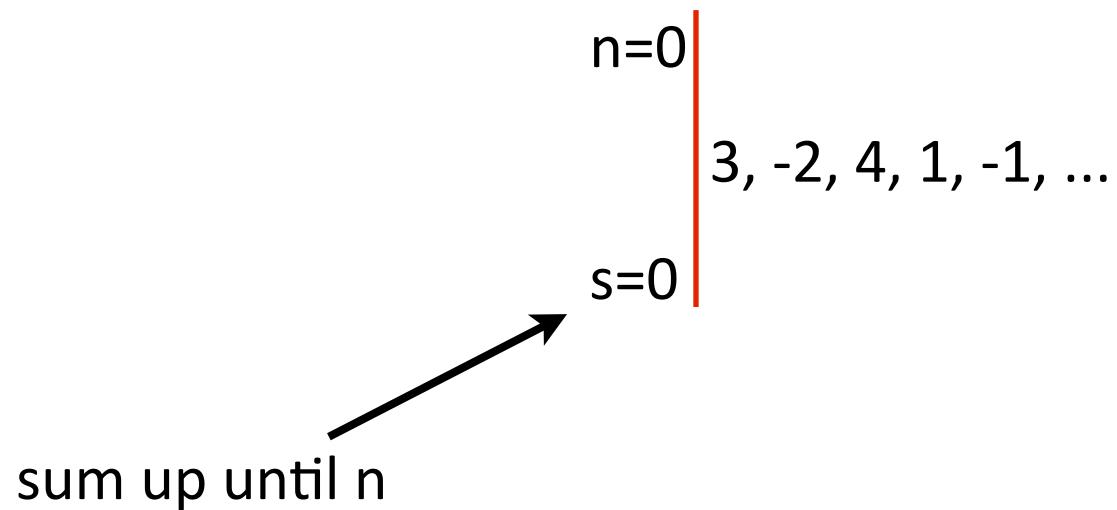
$n=0$ |
3, -2, 4, 1, -1, ...
 $s=0$ |

Stream of numbers

position in stream



Stream of numbers



Stream of numbers

$n=0$ |
3, -2, 4, 1, -1, ...
 $s=0$ |

Stream of numbers

$n=1$
3, -2, 4, 1, -1, ...
 $s=3$

Stream of numbers

$n=2$ |
3, -2, 4, 1, -1, ...
 $s=1$ |

Stream of numbers

$n=3$
3, -2, 4, | 1, -1, ...
 $s=5$

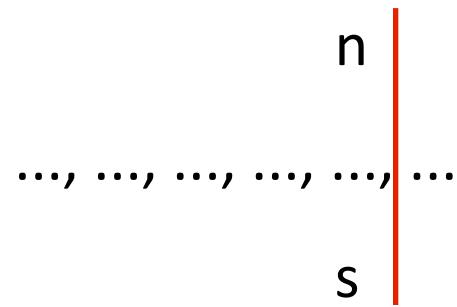
Stream of numbers

$n=4$
3, -2, 4, 1, -1, ...
 $s=6$

Stream of numbers

$n=5$
3, -2, 4, 1, -1, ...
 $s=5$

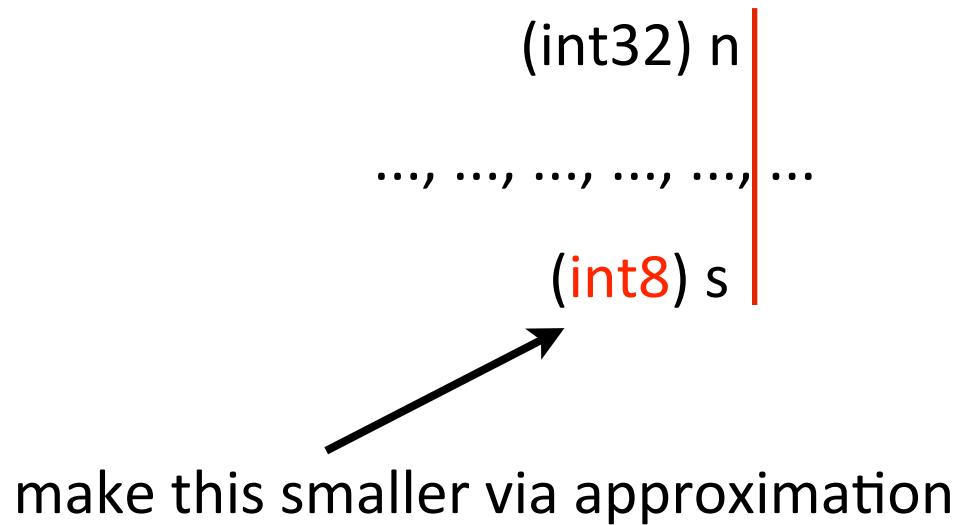
Size of variables



Size of variables

(int32) n
..., ..., ..., ..., ..., ...
(int32) s

Size of variables



Reservoir Sampling

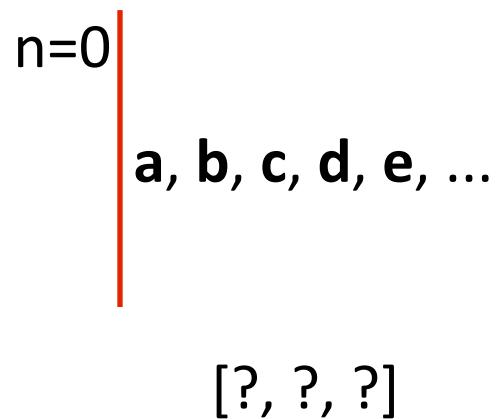
a, b, c, d, e, ...

Reservoir Sampling

a, b, c, d, e, ...

reservoir of size k=3 → [?, ?, ?]

Reservoir Sampling



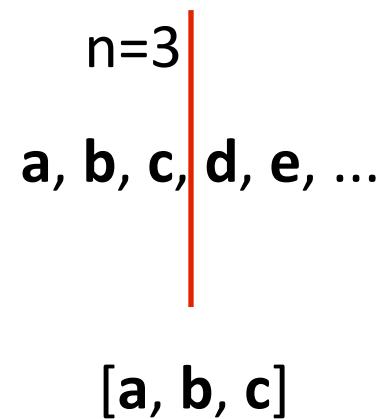
Reservoir Sampling

n=1
|
a, b, c, d, e, ...
|
[a, ?, ?]

Reservoir Sampling

n=2 |
a, b, | c, d, e, ...
[a, b, ?]

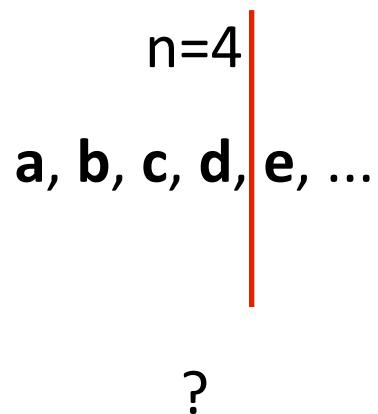
Reservoir Sampling



Reservoir Sampling

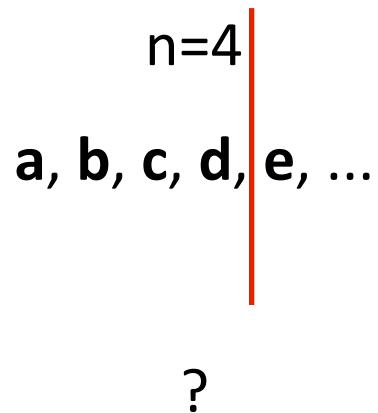
$n=4$
a, b, c, d, e, ...
?

Reservoir Sampling



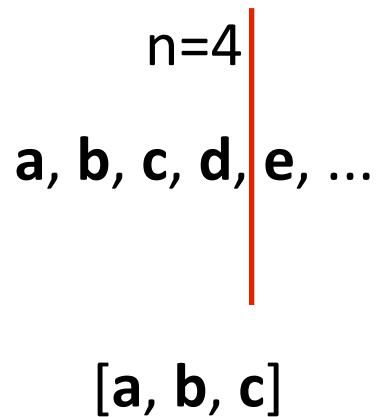
1. accept **d** with probability: $k/n = 3/4$

Reservoir Sampling



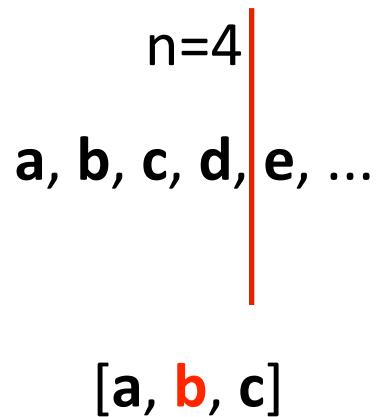
1. **accept d** with probability: $k/n = 3/4$
2. if accept, then **evict** a random element from reservoir

Reservoir Sampling



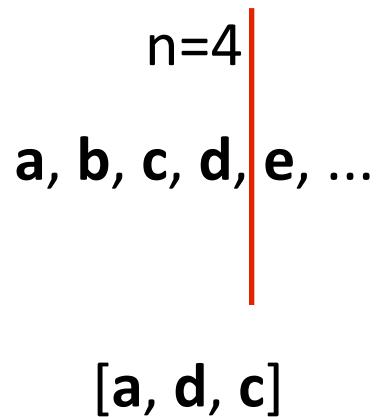
1. **accept d** with probability: $k/n = 3/4$
2. if accept, then **evict** a random element from reservoir

Reservoir Sampling



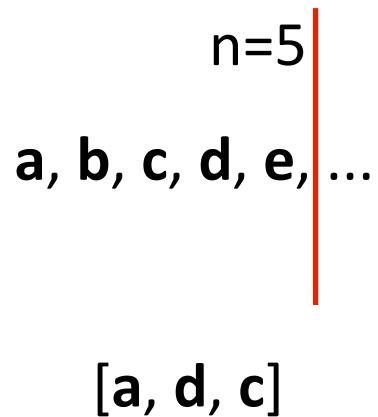
1. **accept d** with probability: $k/n = 3/4$
2. if accept, then **evict** a random element from reservoir

Reservoir Sampling



1. **accept d** with probability: $k/n = 3/4$
2. if accept, then **evict** a random element from reservoir

Reservoir Sampling



1. accept **e** with probability: $k/n = 3/5$

Reservoir Sampling

n=26

a, b, c, d, e, ...

[p, h, z]

Stream of numbers

3, -2, 4, 1, -1, ...

Stream of numbers in +/- unary

3, -2, 4, 1, -1, ...

1, 1, 1

Stream of numbers in +/- unary

3, -2, 4, 1, -1, ...

1, 1, 1, -1, -1

Stream of numbers in +/- unary

3, -2, 4, 1, -1, ...

1, 1, 1, -1, -1, 1, 1, 1, 1

Stream of numbers in +/- unary

3, -2, 4, 1, -1, ...

Stream of numbers in +/- unary

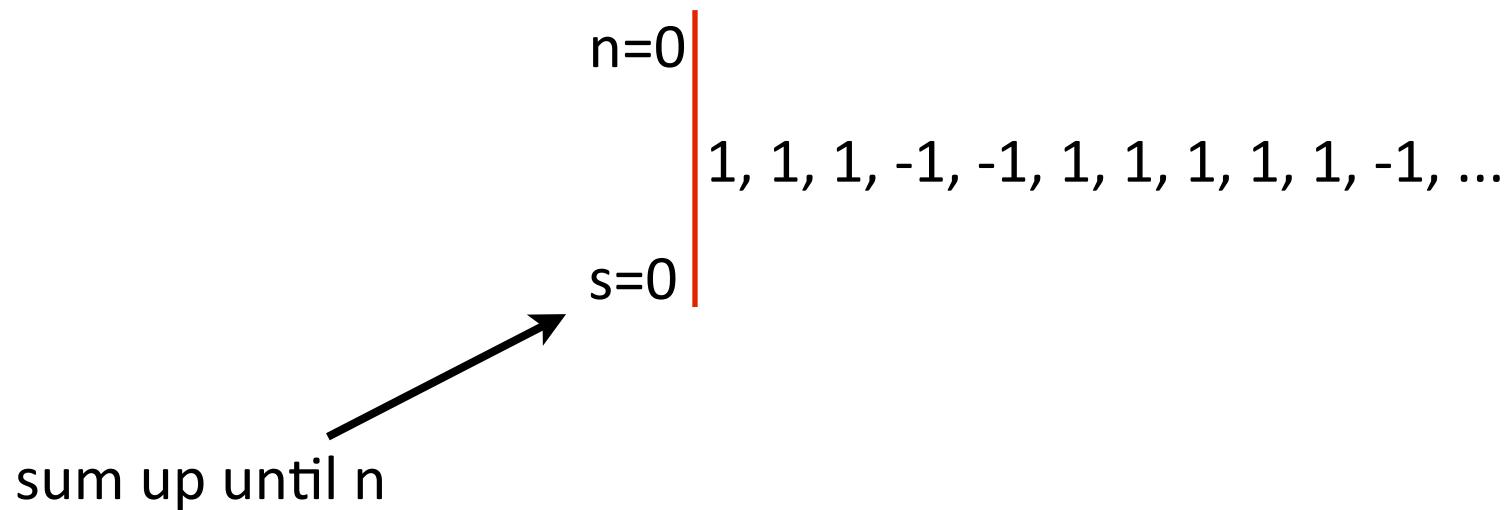
3, -2, 4, 1, -1, ...

1, 1, 1, -1, -1, 1, 1, 1, 1, 1, -1, ...

Stream of numbers in +/- unary

1, 1, 1, -1, -1, 1, 1, 1, 1, 1, -1, ...

Stream of numbers in +/- unary



Stream of numbers in +/- unary

$n=1$ |
1, 1, 1, -1, -1, 1, 1, 1, 1, 1, -1, ...
 $s=1$ |

Stream of numbers in +/- unary

$n=2$ |
1, 1, | 1, -1, -1, 1, 1, 1, 1, 1, -1, ...
 $s=2$ |

Stream of numbers in +/- unary

... |
1, 1, 1, -1, -1, 1, 1, 1, 1, 1, -1, ...
... |

Stream of numbers in +/- unary

$n=6$
1, 1, 1, -1, -1, 1, | 1, 1, 1, 1, -1, ...
 $s=2$

Stream of numbers in +/- unary

1, 1, 1, -1, -1, 1, 1, 1, 1, | 1, -1, ...
... | ...

Stream of numbers in +/- unary

$n=11$
1, 1, 1, -1, -1, 1, 1, 1, 1, 1, -1, ...
 $s=5$

Sampling from stream of numbers in +/- unary

1, 1, 1, -1, -1, 1, 1, 1, 1, 1, -1, ...

Sampling from stream of numbers in +/- unary

n=0

1, 1, 1, -1, -1, 1, 1, 1, 1, 1, -1, ...

[?, ?, ?]

Sampling from stream of numbers in +/- unary

$n=3$
1, 1, 1, -1, -1, 1, 1, 1, 1, 1, -1, ...
[1,1,1]

Sampling from stream of numbers in +/- unary

n=5 |
1, 1, 1, -1, -1, | 1, 1, 1, 1, 1, -1, ...
[1, -1, 1]

Sampling from stream of numbers in +/- unary

n=9 |
1, 1, 1, -1, -1, 1, 1, 1, 1, 1, -1, ...
[1, **1**, 1]

Exchangeability of elements

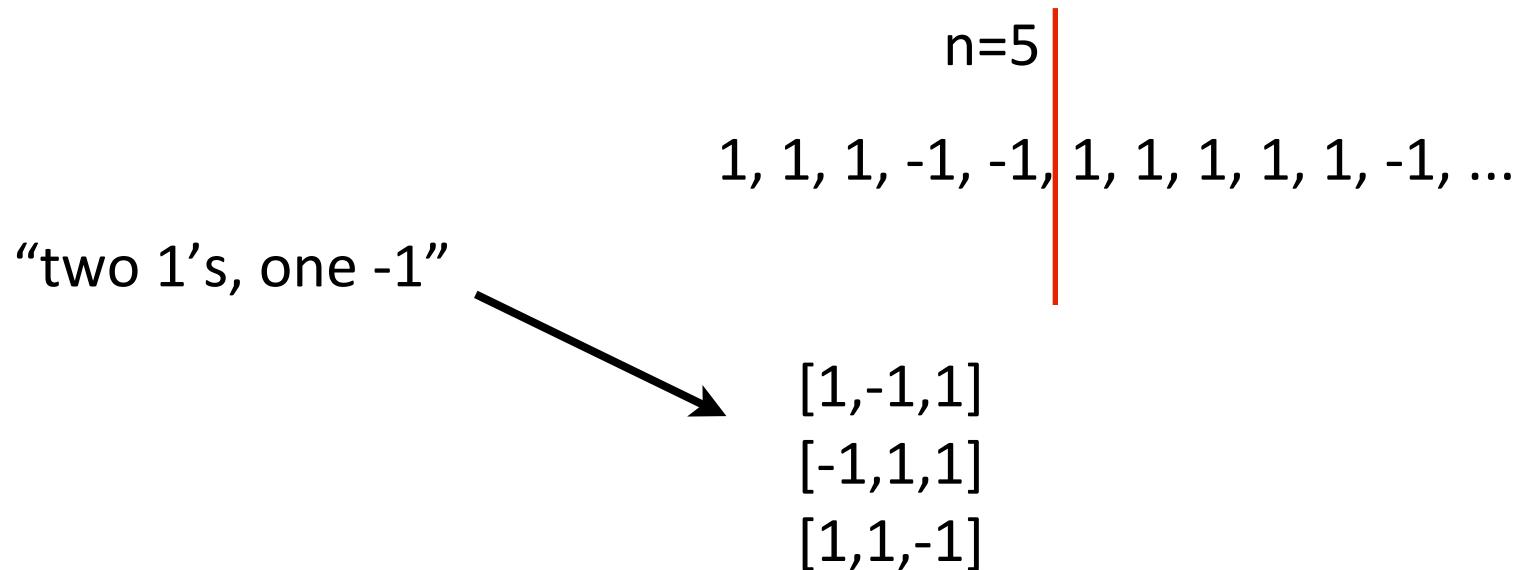
n=5 |
1, 1, 1, -1, -1, | 1, 1, 1, 1, 1, -1, ...
[1,-1,1]

Exchangeability of elements

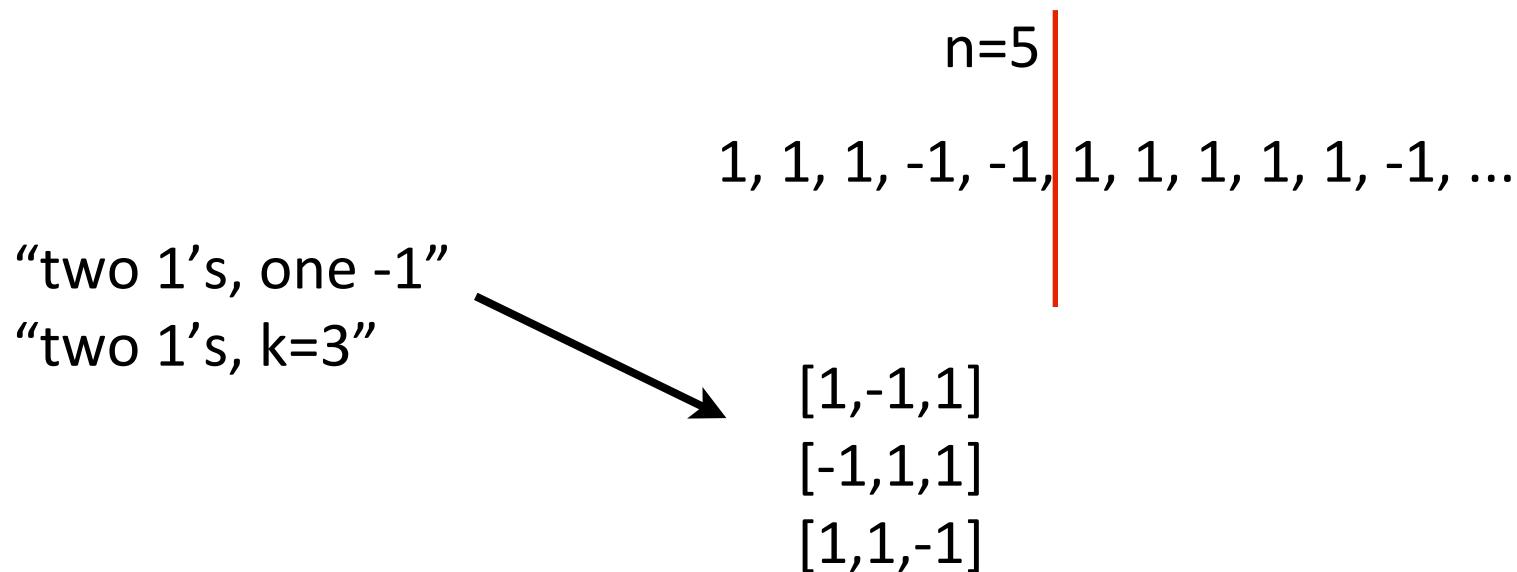
n=5 |
1, 1, 1, -1, -1, | 1, 1, 1, 1, 1, -1, ...

[1,-1,1]
[-1,1,1]
[1,1,-1]

Implicit reservoir



Implicit reservoir



Implicit reservoir

If $k=3$, there can be:
zero 1's,
one 1,
two 1's, or
three 1's.

Implicit reservoir

If $k=3$, there can be:
zero 1's,
one 1,
two 1's, or
three 1's.

For a given k , there can be:
 $k+1$ different states,
representable with $\lg(k+1)$ bits.

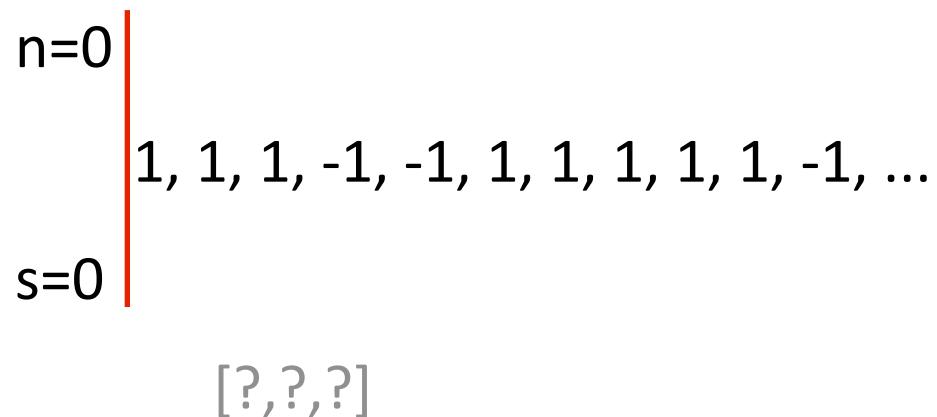
Implicit reservoir

If $k=3$, there can be:
zero 1's,
one 1,
two 1's, or
three 1's.

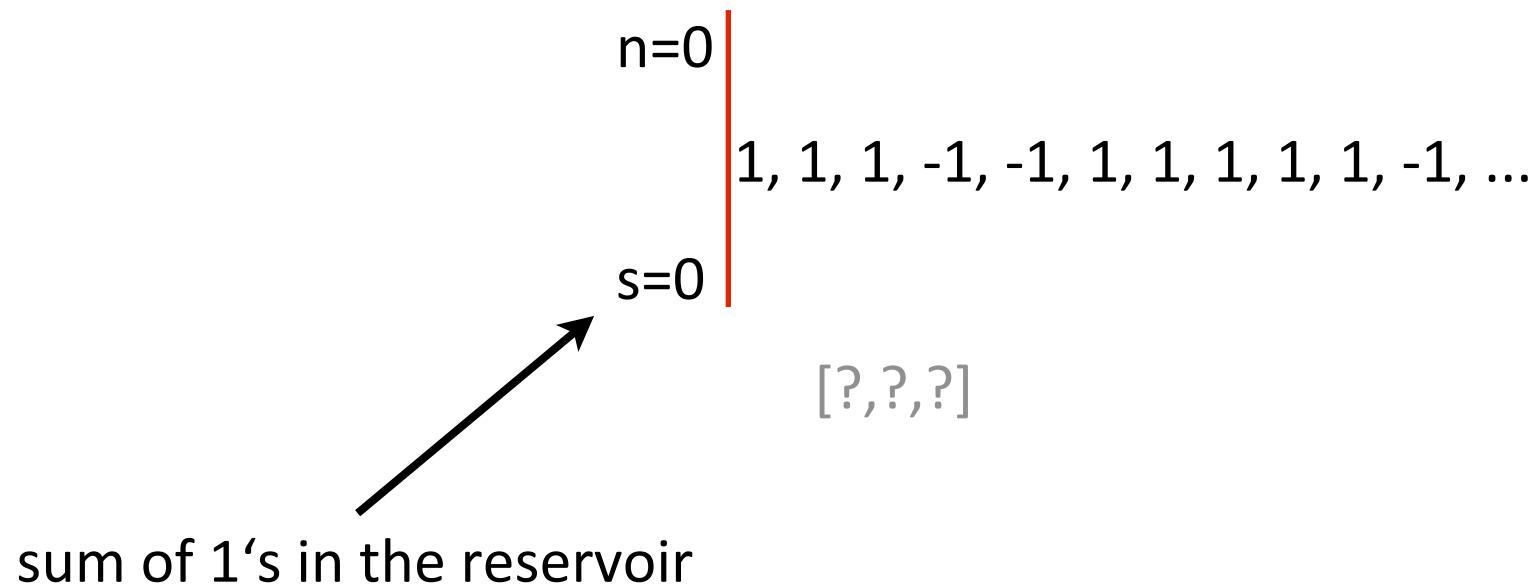
For a given k , there can be:
 $k+1$ different states,
representable with $\lg(k+1)$ bits.

When $k=255$, need $\lg(256) = 8$ bits.

Sampling from stream of numbers in +/- unary



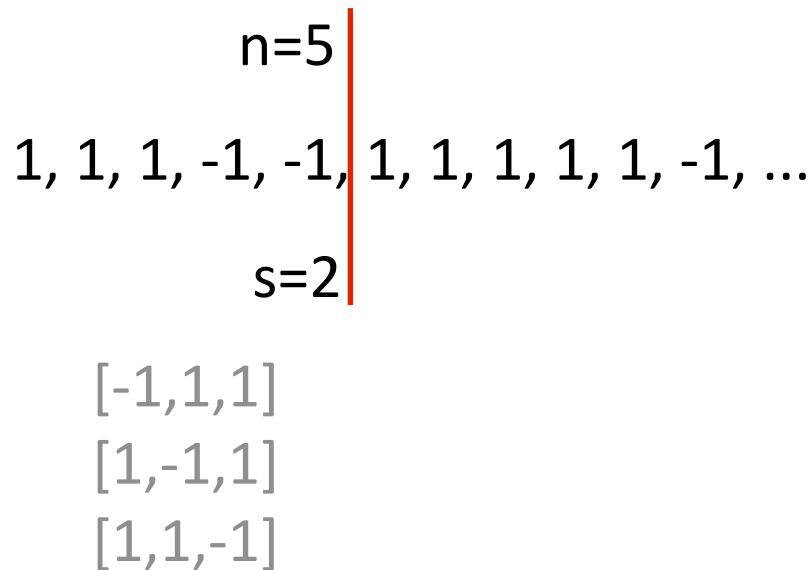
Sampling from stream of numbers in +/- unary



Sampling from stream of numbers in +/- unary

$n=3$ |
1, 1, 1, -1, -1, 1, 1, 1, 1, 1, -1, ...
 $s=3$ |
[1,1,1]

Sampling from stream of numbers in +/- unary



Sampling from stream of numbers in +/- unary

$n=9$
1, 1, 1, -1, -1, 1, 1, 1, 1, 1, -1, ...
 $s=3$

[1,1,1]

Sampling from stream of numbers

3, -2, 4, 1, -1, ...

1, 1, 1, -1, -1, 1, 1, 1, 1, 1, -1, ...

Sampling from stream of numbers

3, -2, 4, 1, -1, ...

n=5

1, 1, 1, -1, -1, 1, 1, 1, 1, 1, -1, ...

s=2

Sampling from stream of numbers

Compute **expected** number of accepts,
then **expected** change to reservoir.

3, -2, **4**, 1, -1, ...

n=5
1, 1, 1, -1, -1, **1, 1, 1, 1, 1, -1, ...**
s=2

Sampling from stream of numbers

Compute expected number of accepts, $k \log((n+m)/n)$
then expected change to reservoir.

3, -2, 4, 1, -1, ...

n=5
1, 1, 1, -1, -1, | 1, 1, 1, 1, 1, -1, ...
s=2 |

Sampling from stream of numbers

Compute expected number of accepts,
then expected change to reservoir. [see paper]

3, -2, 4, 1, -1, ...

n=5
1, 1, 1, -1, -1, | 1, 1, 1, 1, 1, -1, ...
s=2 |

Online LSH : b parallel streams

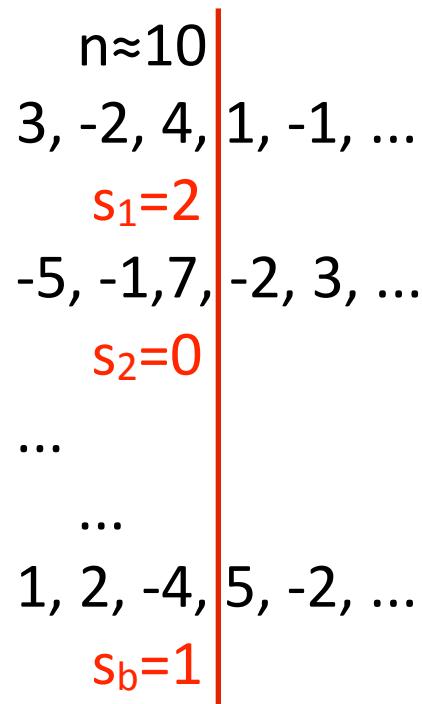
3, -2, 4, 1, -1, ...

-5, -1, 7, -2, 3, ...

...

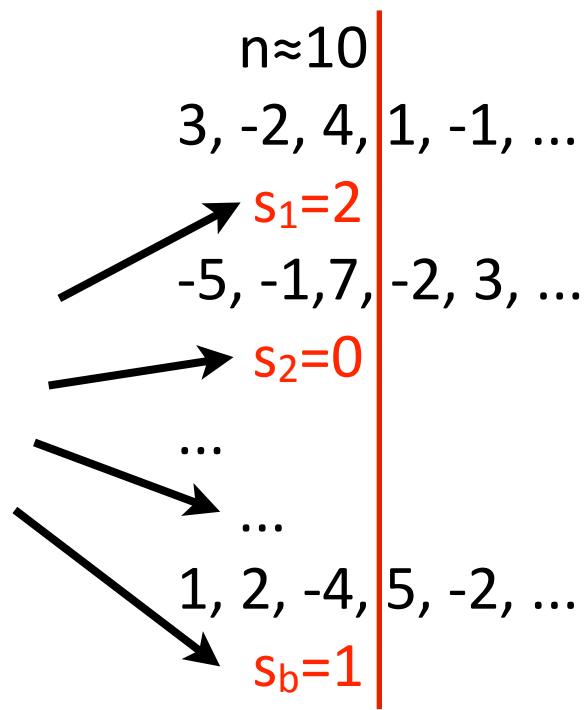
1, 2, -4, 5, -2, ...

Online LSH : b parallel streams

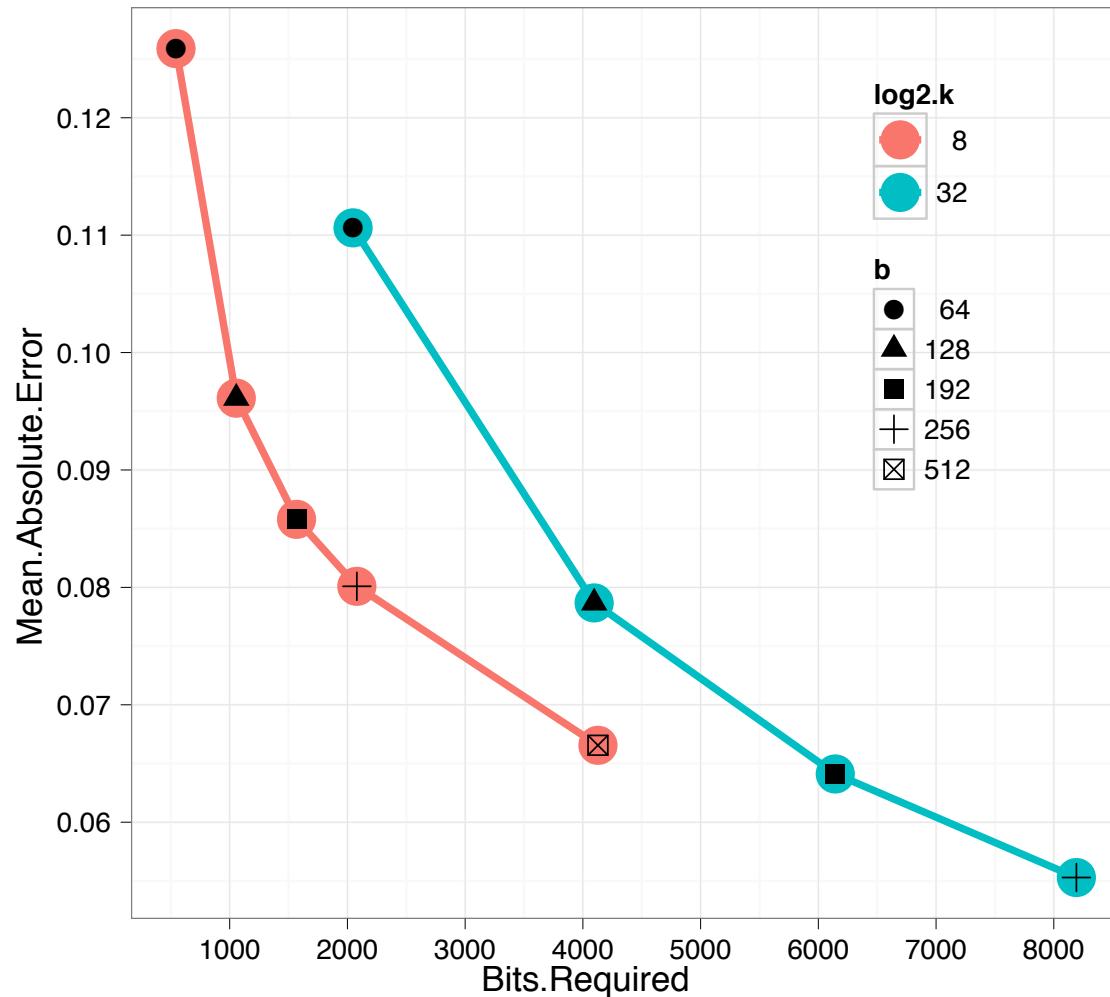


Online LSH : b parallel streams

from int32
to $\lg(k+1)$ bits,
e.g., int8

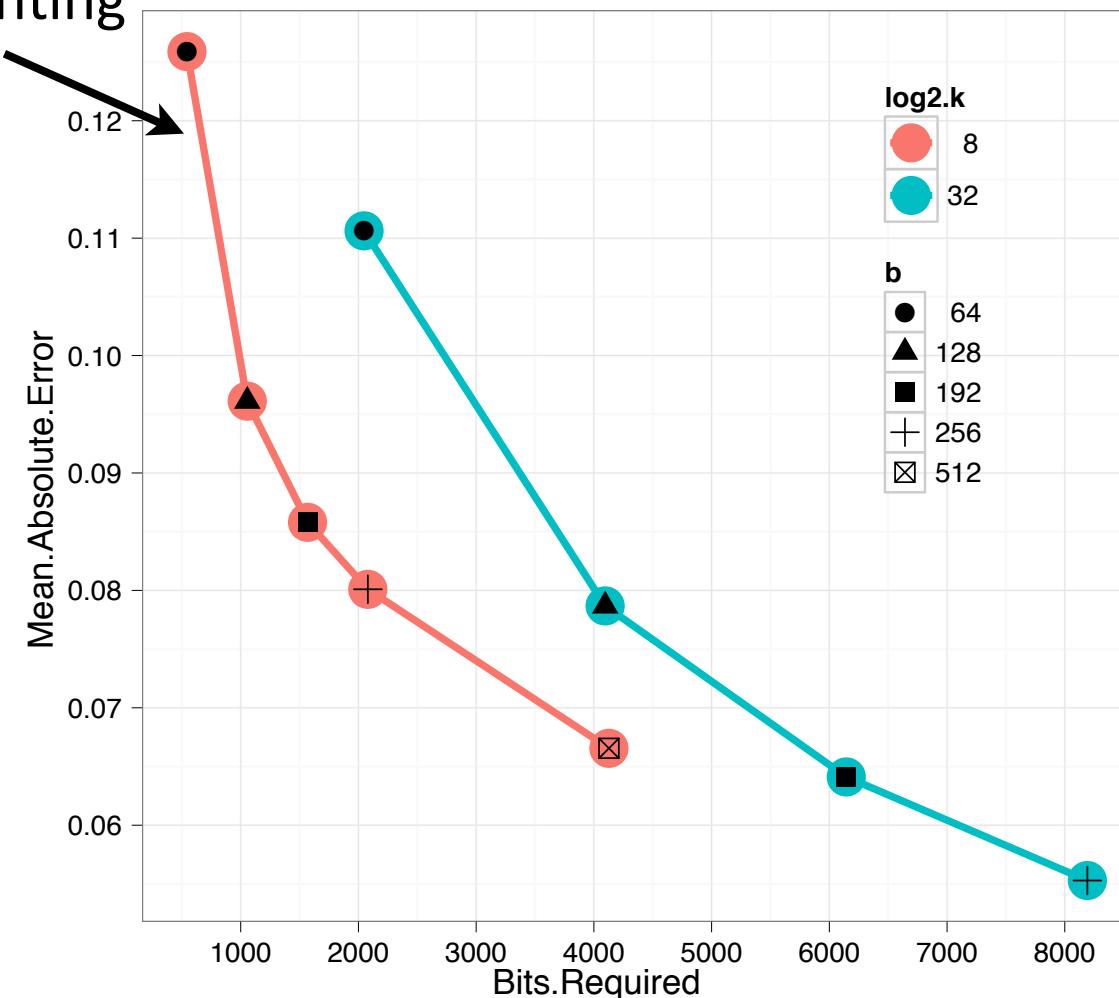


More bang for the bit

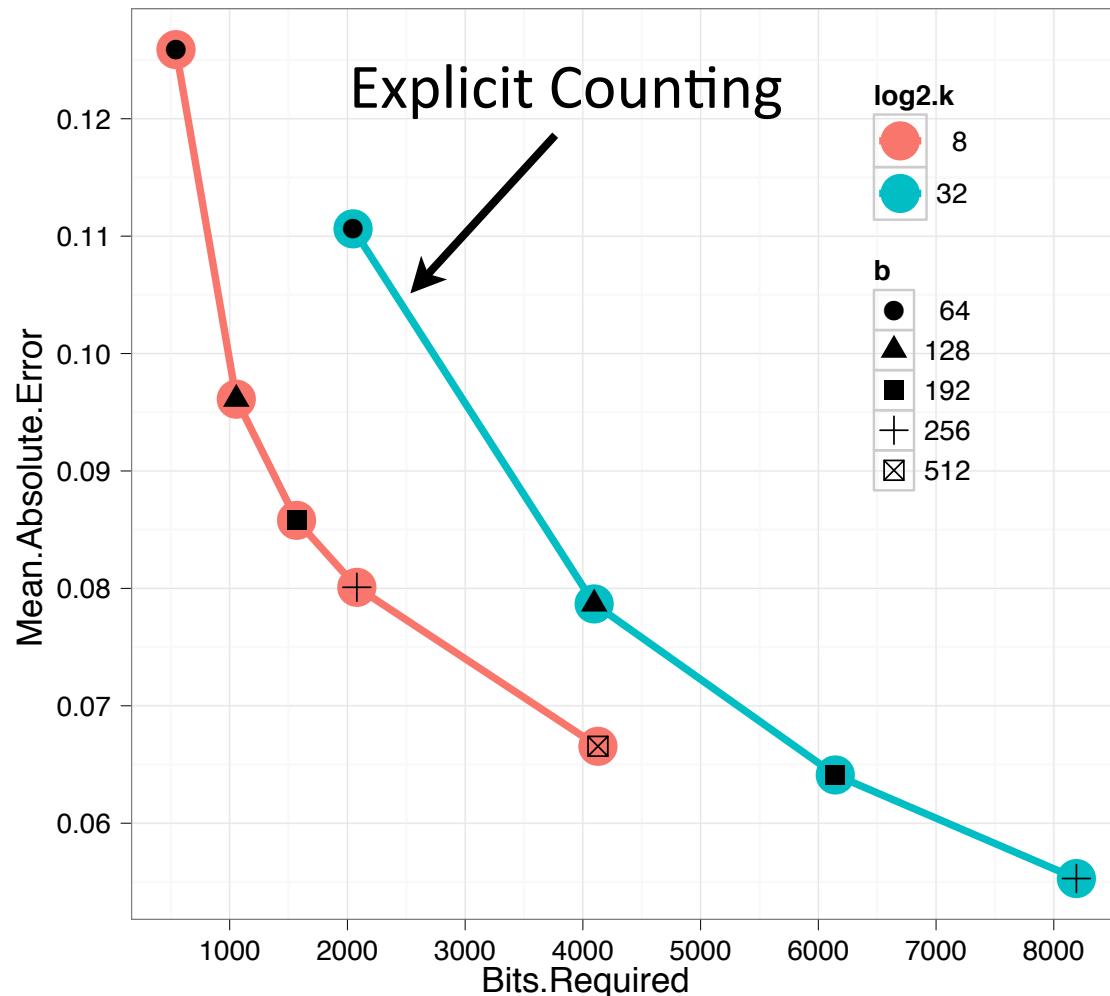


More bang for the bit

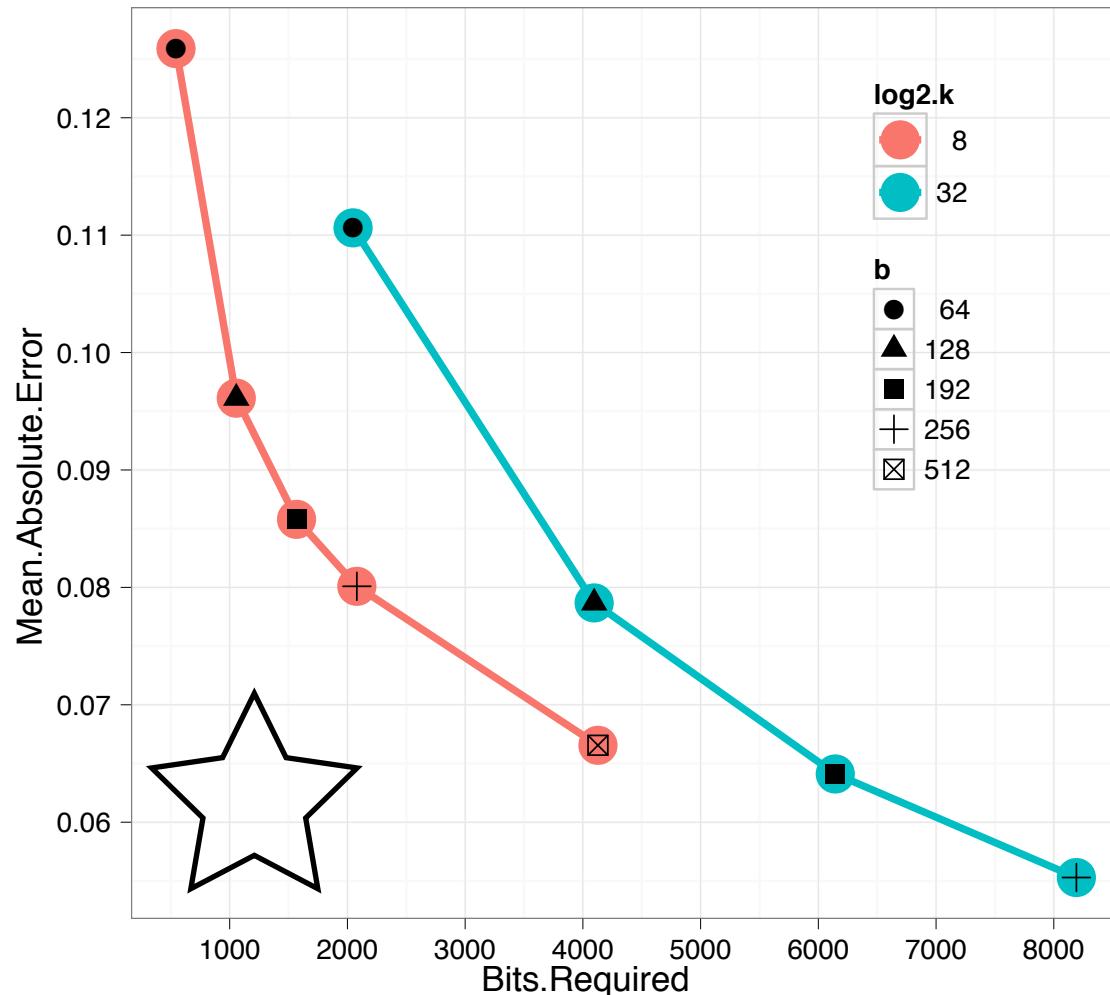
Reservoir Counting



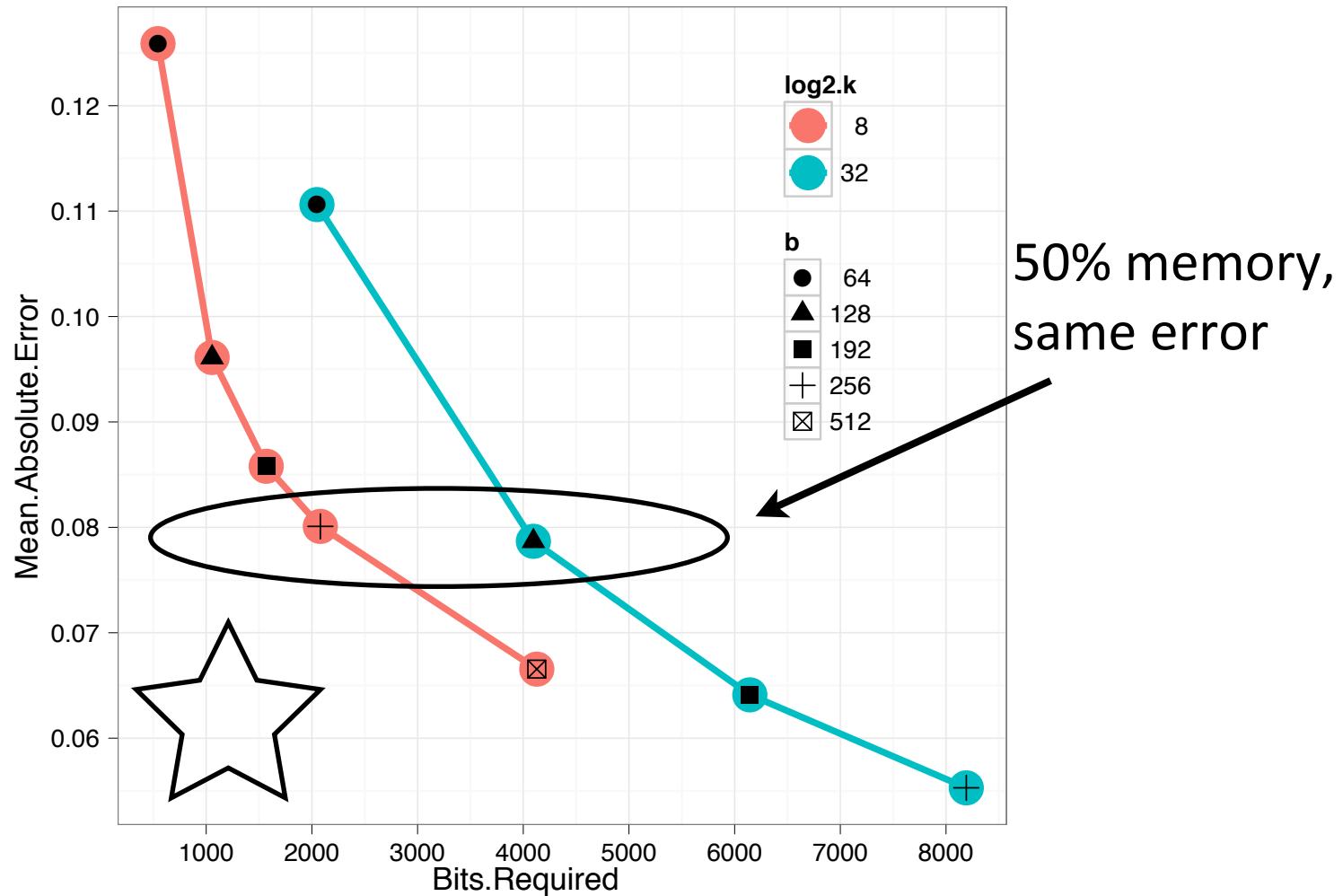
More bang for the bit



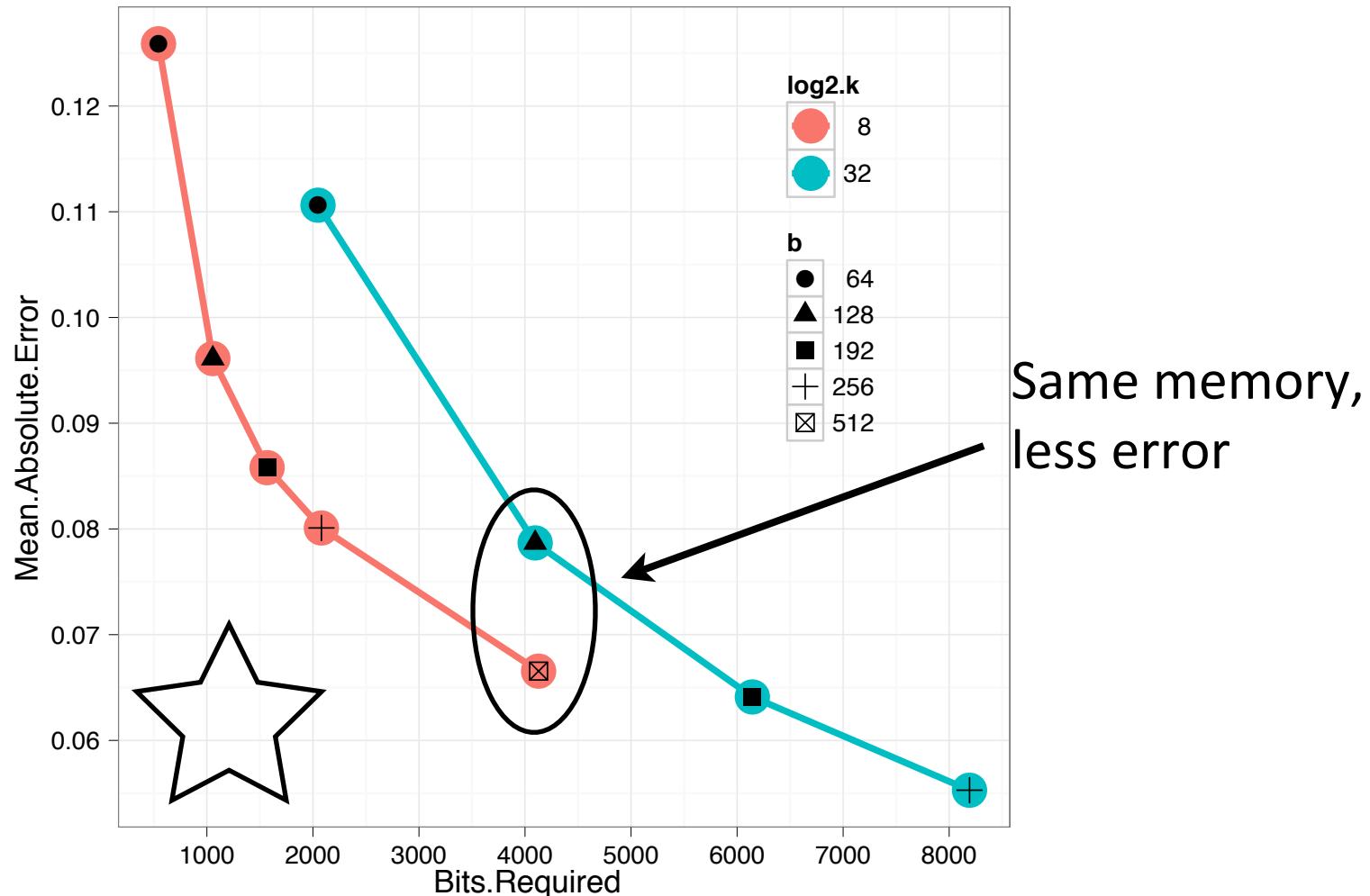
More bang for the bit



More bang for the bit



More bang for the bit



Thanks



human language technology
center of excellence
JOHNS HOPKINS
UNIVERSITY