

# Efficient Estimation for High Similarities using Odd Sketches

**Ninh Pham**, IT University of Copenhagen

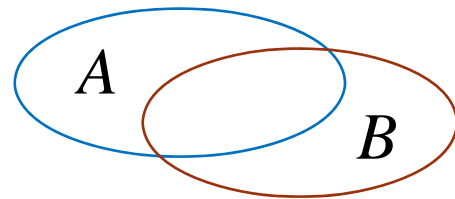
Joint work with **Rasmus Pagh**, IT University of Copenhagen  
and **Michael Mitzenmacher**, Harvard University

# Outline

- Set Similarity and Its Uses
- Minwise Hashing Schemes and Challenge
- Odd Sketch
  - Construction
  - Estimation
  - Quality of Estimation
- Experiments
- Conclusions

# Set Similarity and Its Uses

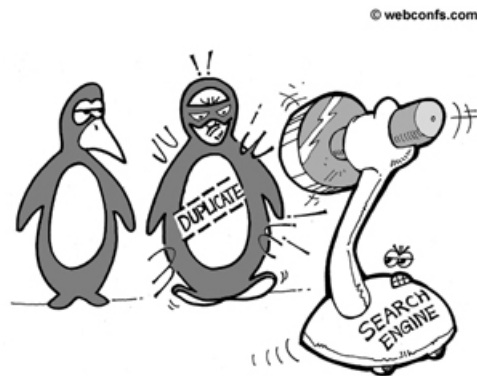
- Jaccard similarity coefficient:



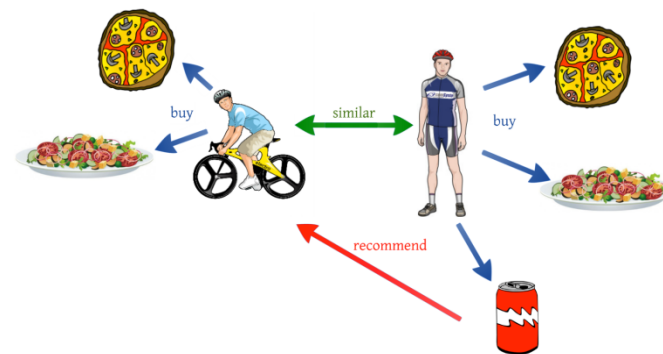
$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}, 0 \leq J \leq 1$$

- Some uses:

Web duplication detection



Collaborative filtering



# MinHash (Minwise Hashing)

Sets and its elements

| id | A | B |
|----|---|---|
| 1  | 1 | 1 |
| 2  | 0 | 0 |
| 3  | 1 | 0 |
| 4  | 0 | 0 |
| 5  | 1 | 1 |

$$J(A, B) = \frac{2}{3}$$

Random permutations

| $\pi_1$ | $\pi_2$ | $\pi_3$ |
|---------|---------|---------|
| 3       | 1       | 1       |
| 2       | 5       | 3       |
| 5       | 4       | 2       |
| 4       | 3       | 4       |
| 1       | 2       | 5       |

Rearrangement

| $\pi_1$ | A | B |
|---------|---|---|
| 3       | 1 | 0 |
| 2       | 0 | 0 |
| 5       | 1 | 1 |
| 4       | 0 | 0 |
| 1       | 1 | 1 |



MinHashes

| $S_1$ | $S_2$ |
|-------|-------|
| 3     | 5     |
| 1     | 1     |
| 1     | 1     |

$$\hat{J} = \frac{|S_1 \cap S_2|}{3} = \frac{2}{3}$$

# MinHash (Minwise Hashing)

- **MinHash Theorem:**

$$\Pr \left[ \min(\pi_i(A)) = \min(\pi_i(B)) \right] = J(A, B)$$

- Denote  $S_1$  and  $S_2$  by minhashes of  $A$  and  $B$  by considering  $k$  independent permutations  $\pi_1, \dots, \pi_k$

$$S_1 = \left\{ \min(\pi_i(A)) \mid i = 1, \dots, k \right\},$$

$$S_2 = \left\{ \min(\pi_i(B)) \mid i = 1, \dots, k \right\},$$

we obtain an **unbiased** estimator of  $J(A, B)$  and its variance:

$$\hat{J} = \frac{|S_1 \cap S_2|}{k}, \quad \text{Var}[\hat{J}] = \frac{J(1-J)}{k}.$$

# $b$ -Bit MinHash

MinHashes

| $S_1$ | $S_2$ |
|-------|-------|
| 3     | 5     |
| 1     | 1     |
| 1     | 1     |

$$\hat{j} = \frac{2}{3}$$

3-Bit MinHashes

| $S_1$       | $S_2$       |
|-------------|-------------|
| 01 <u>1</u> | 10 <u>1</u> |
| 00 <u>1</u> | 00 <u>1</u> |
| 00 <u>1</u> | 00 <u>1</u> |

$$\hat{j}^{b=3} = \frac{2}{3}$$

1-Bit MinHashes

| $S_1$ | $S_2$ |
|-------|-------|
| 1     | 1     |
| 1     | 1     |
| 1     | 1     |

$$\hat{j}^{b=1} = \frac{|S_1 \cap S_2|/3 - 1/2}{1 - 1/2} = 1$$

- **Intuition:**

- The same hash values give the same lowest  $b$  bits.
- Different hash values give different lowest  $b$  bits with probability  $1 - 1/2^b$ .

# $b$ -Bit MinHash

- $b$ -Bit MinHash Theorem

- Denote  $\min_b(\boldsymbol{\pi}(A))$  by the lowest  $b$  bits of the hash value  $\min(\boldsymbol{\pi}(A))$ , we obtain  $b$ -bit minhashes of  $A$  and  $B$

$$S_1^b = \left\{ \min_b(\pi_i(A)) \mid i = 1, \dots, k \right\},$$

$$S_2^b = \left\{ \min_b(\pi_i(B)) \mid i = 1, \dots, k \right\}.$$

We obtain an unbiased estimator for  $J(A, B)$  and its variance

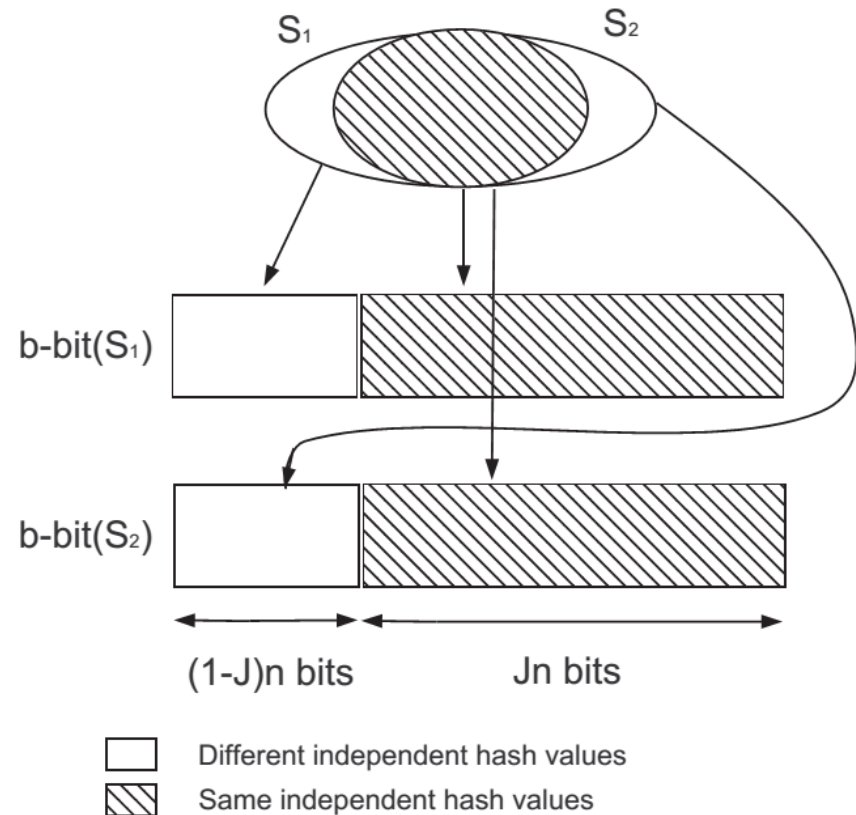
$$\hat{j}^b = \frac{|S_1^b \cap S_2^b| / k - 1/2^b}{1 - 1/2^b}, \quad \text{Var}[\hat{j}^b] = \frac{1 - J}{k} \left( J + \frac{1}{2^b - 1} \right).$$

- $b$ -bit MinHash uses more permutations than MinHash.

# Challenge

- When the Jaccard similarity is high,  $b$ -bit MinHash offers less information due to the two likely identical  $b$ -bit summaries.
- Inaccuracy in just a few bit positions (**white space**) will yield a large relative error of the estimate of  $J$ .

## $b$ -bit MinHash Construction





# Odd Sketch: Intuition

- **The Bloom filter principle:**
  - Wherever a list or set is used, and space is at a premium, consider using a Bloom filter if the effect of false positives can be mitigated.
- **The Odd Sketch:**
  - A Bloom filter using **one** hash function with an “**odd**” feature that the usual disjunction (OR) is replaced by an exclusive-or operation (XOR).
  - Constructed on the original minhashes ( $S_1$  and  $S_2$ ).

# Odd Sketch: Construction and Property

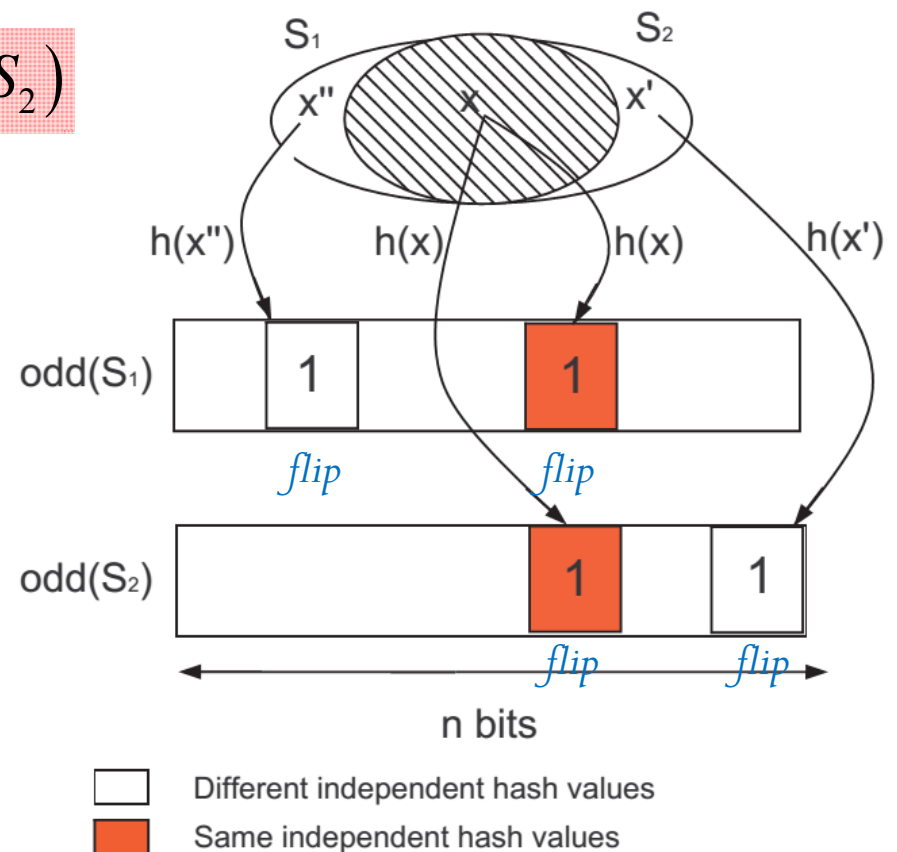
- **Property:**

$$\text{odd}(S_1) \oplus \text{odd}(S_2) = \text{odd}(S_1 \Delta S_2)$$

- When  $J$  is close to 1, we can use odd sketches of size  $n$  on minhashes of size **significantly above  $n$** . So the variance induced by the minhash step is reduced.

- Note:  $x = (i, \min(\pi_i(A)))$ ,  
 $h : [k] \times [\Omega] \rightarrow [n]$

## Odd Sketch Construction



# Odd Sketch: Estimation

- Jaccard similarity estimation:

1. Estimate  $|S_1 \Delta S_2|$  from its sketch  $odd(S_1 \Delta S_2)$
2. Estimate  $J(A, B)$  by

$$\hat{j}^{odd} = \frac{|S_1 \cap S_2|}{k} = 1 - \frac{|S_1 \Delta S_2|}{2k}.$$

- **Problem:** How to estimate the set's size from its odd sketch?

- Constructing  $odd(S)$  of size  $n$  (bits) of a set  $S$  of  $m$  elements as independently throwing  $m$  balls into  $n$  bins, and storing the **parity** of the number of balls in each bin.
- We will estimate  $m$  based on the observation of the number of “odd” bins in the sketch of size  $n$ .

# Estimate a set's size from its odd sketch

- **Poisson approximation (independent case):**

- When  $m$  balls are thrown into  $n$  bins, this is very approximately the same as independently giving each bin a number of balls that is Poisson distributed with mean  $m/n$ .

- **Lemma:**

Let  $Q$  be a random variable that has Poisson distribution with mean  $m/n$ . The probability  $p$  that  $Q$  is odd is  $(1 - e^{-\frac{2m}{n}})/2$ .

- Given  $Z$  odd bins in the sketch, we obtain an estimate

$$\hat{m} = -\frac{n}{2} \ln\left(1 - \frac{2Z}{n}\right)$$

# Estimate a set's size from its odd sketch

- **Two-state Markov chain (dependent case):**

- Changing of the parity of number of balls in any specific bin is a two-state Markov chain with the probability of changing state is  $1/n$ .

- Let  $p_i$  be the probability that any specific bin has an odd number of balls after  $i$  balls have been thrown, we have

$$p_i = \frac{1 - (1 - 2/n)^i}{2}$$

- Given  $Z$  odd bins in the sketch, we obtain an estimate

$$E[Z] = n \frac{1 - (1 - 2/n)^m}{2} \implies \hat{m} = \frac{\ln(1 - 2Z/n)}{\ln(1 - 2/n)}$$

# Estimate Jaccard similarity from odd sketches

- We construct odd sketches on the original minhashes.
  1.  $E[|S_1 \Delta S_2|] = 2k(1 - J)$ ,
  2.  $odd(S_1) \Delta odd(S_2) = odd(S_1 \Delta S_2)$ .
- We rely on the Poisson approximation approach, and estimate the symmetric difference  $|S_1 \Delta S_2|$

$$|S_1 \hat{\Delta} S_2| = -\frac{n}{2} \ln \left( 1 - \frac{2|odd(S_1) \Delta odd(S_2)|}{n} \right).$$

- We estimate the Jaccard similarity

$$\hat{j}^{odd} = 1 + \frac{n}{4k} \ln \left( 1 - \frac{2|odd(S_1) \Delta odd(S_2)|}{n} \right).$$

# Quality of Estimation: Concentration

- Concentration of  $Z/n$ , the fraction of odd bins:

$$\Pr(|Z/n - p| \geq \varepsilon) \leq (2e\sqrt{m})e^{-2n\varepsilon^2}$$

where  $p = \frac{1 - e^{-2m/n}}{2}$  (Poisson case) differs from the expectation  $\frac{1 - (1 - 2/n)^m}{2}$  (Markov chain case) by an  $o(1)$  amount.

By choosing  $n > c\varepsilon^{-2}\log k$  for some constant  $c$ , our estimator closely concentrates around the estimator of Poisson case  $p$  with probability  $1 - k^{-\omega(1)}$ .

# Quality of Estimation: Variance

- Poisson approximation (independent case):

$$\text{Var}[Z] = np(1-p) \text{ where } p = \frac{1 - e^{-2m/n}}{2}$$

- Two-state Markov chain (dependent case):

$$\begin{aligned} \text{Var}[Z] &= n^2 \frac{(1 - 4/n)^m - (1 - 2/n)^{2m}}{4} + n \frac{1 - (1 - 4/n)^m}{4} \\ &\leq n \frac{1 - (1 - 4/n)^m}{4} \approx n \frac{1 - e^{-4m/n}}{4} = np(1-p) \end{aligned}$$



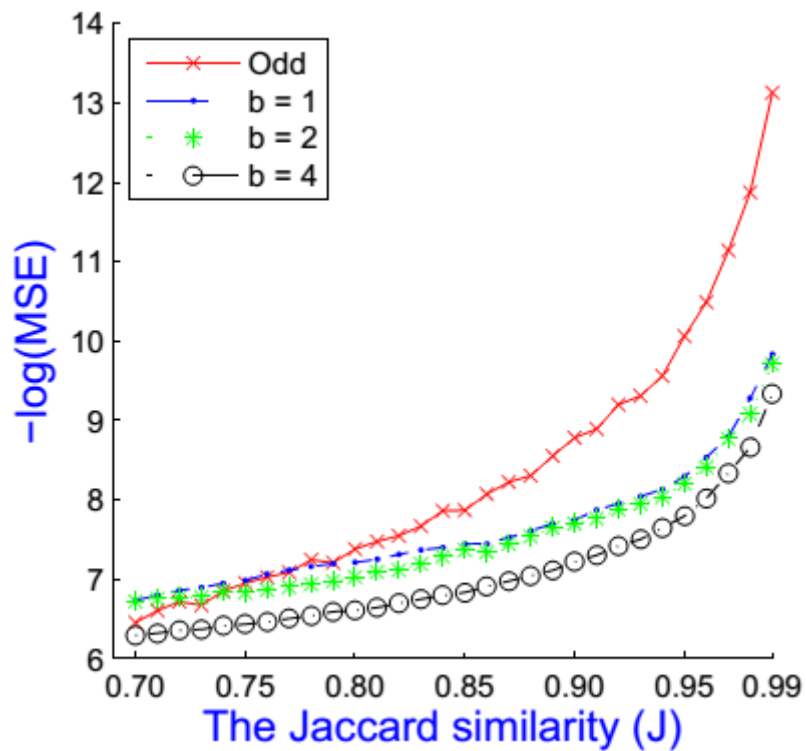
# Experiments

- **Parameter setting:**

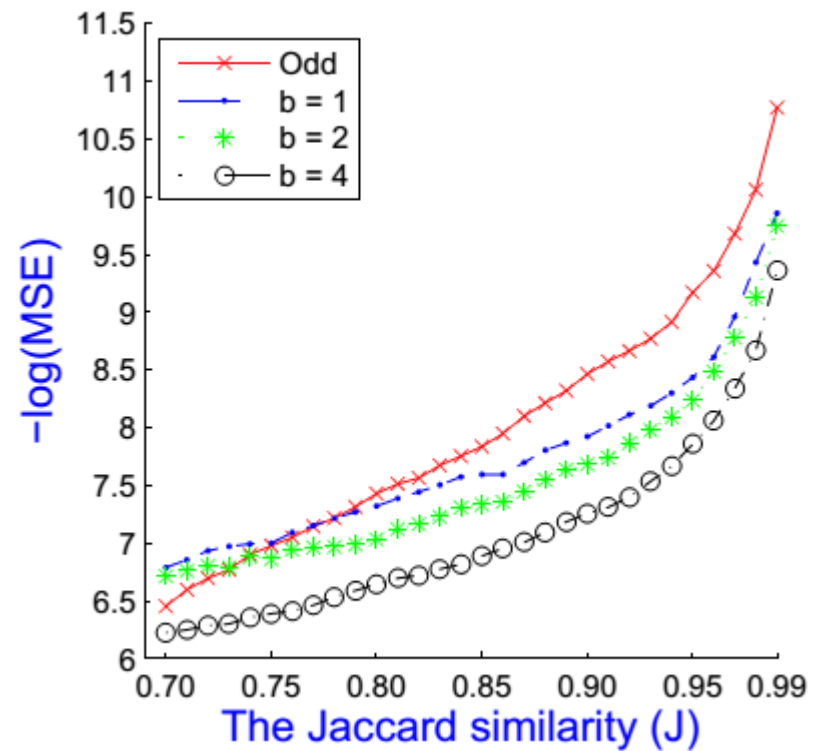
- Performance of both odd sketch and  $b$ -bit MinHash depends on the number of independent permutations  $k$ .
- $b$ -bit scheme uses  $k_b = \frac{n}{b}$  where  $n$  (bits) is the size of sketch.
- Odd sketch uses  $k_{odd} = \frac{n}{4(1-J_0)}$  where  $J_0$  is the user-defined similarity threshold (heuristic).
- **Observation:** When  $J_0 > 0.75 \rightarrow k_{odd} > k_b$ , odd sketch achieves better accuracy than  $b$ -bit scheme.

# Accuracy

Comparison of accuracy between Odd Sketch and  $b$ -bit MinHash on sparse synthetic dataset with sketch size of 512 bits.



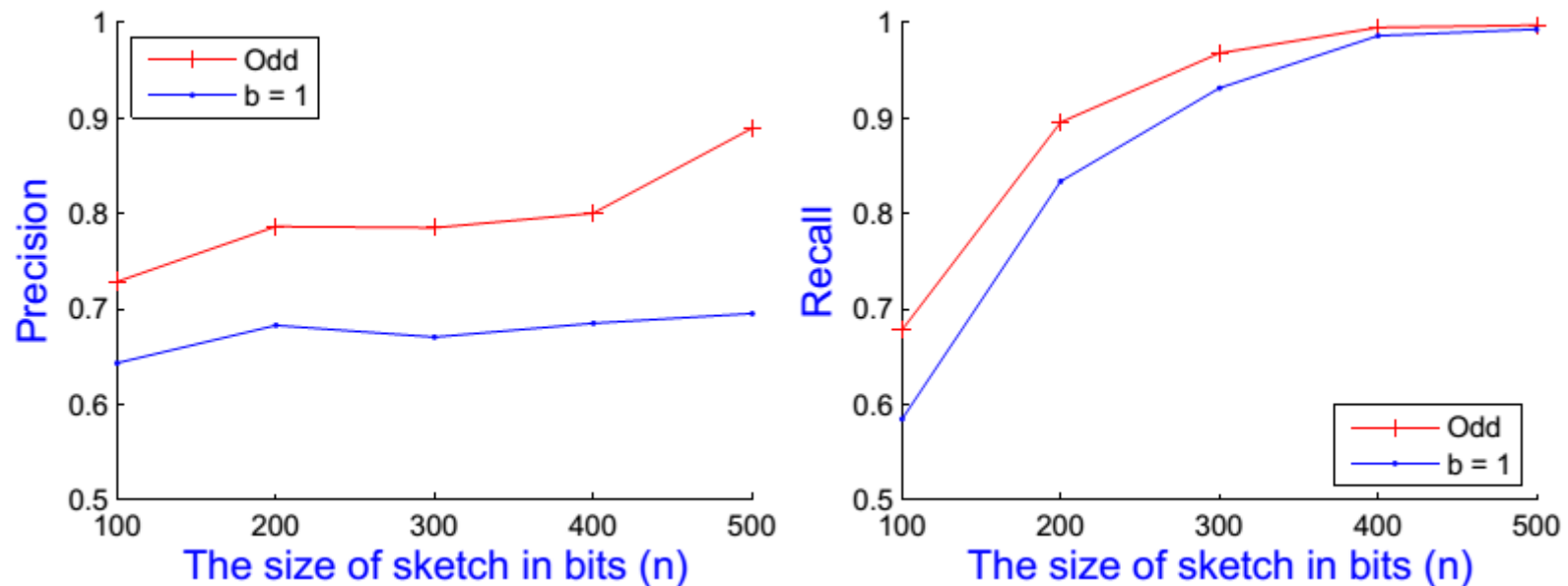
$$k_{\text{odd}} = \frac{n}{4(1-J)} \text{ and } k_b = \frac{n}{b}$$



$$k_{\text{odd}} = n \text{ and } k_b = \frac{n}{b}$$

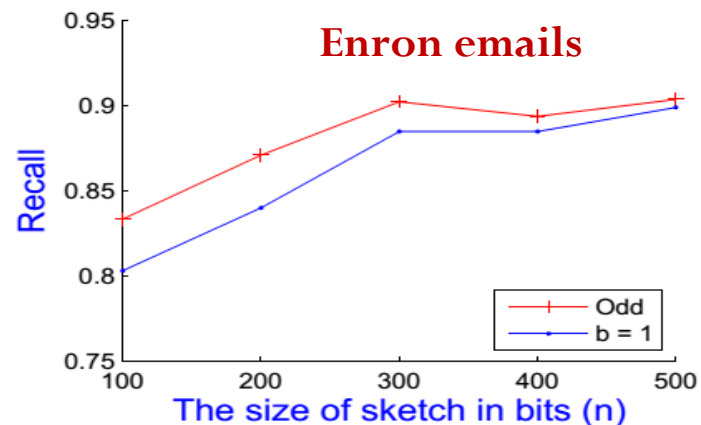
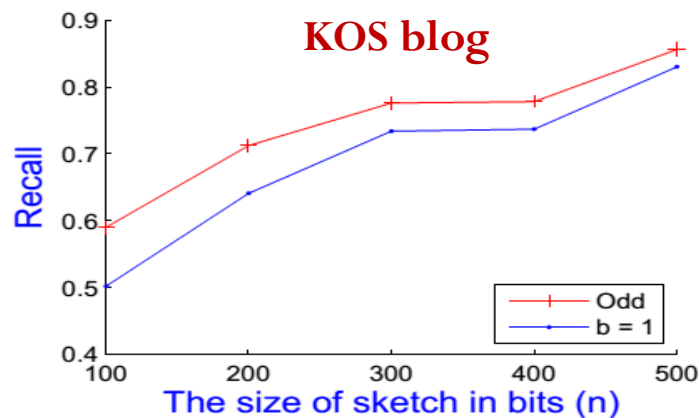
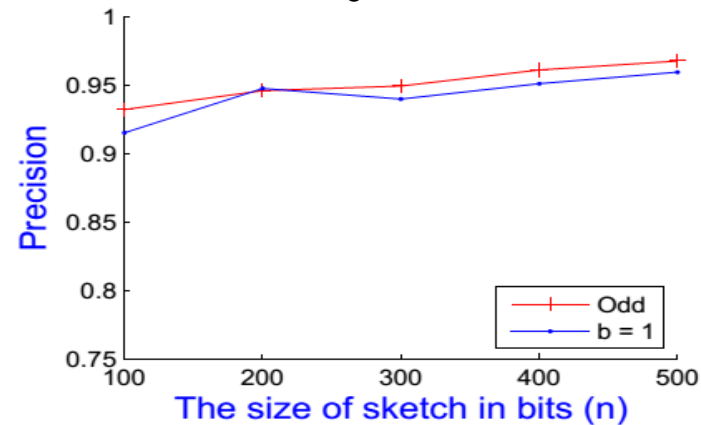
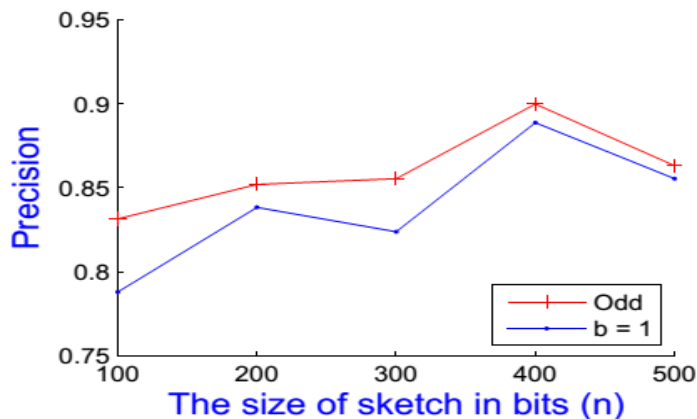
# Association rule learning

Comparison on precision/recall ratio between Odd Sketch and 1-bit MinHash on the mushroom dataset on detecting pairwise items that have  $J > J_0 = 0.9$ .



# Web duplication detection

Comparison on precision/recall ratio between Odd Sketch and 1-bit MinHash on the KOS blog entries and Enron emails datasets on detecting pairwise documents that have  $J > J_0 = 0.9$ .



# Conclusion

- Odd Sketch - a highly space-efficient sketch for the high similarity regime.
- Theoretical error analysis
- Experimental results on the accuracy and efficiency on real-world data sets.