CS-GY 9223 D: Lecture 2 Supplemental
Finish MinHash, Exponential Tail Bounds

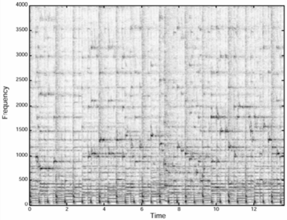
NYU Tandon School of Engineering, Prof. Christopher Musco

Abstract architecture of a sketching algorithm:

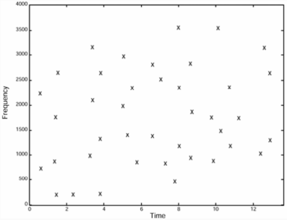
- Given a (high dimensional) dataset $D = d_1, \ldots, d_n$ with n pieces of data each in $\mathbb{R}^d$.
- **Sketch phase:** For each $i \in 1, \ldots, n$, compute $s_i = C(d_i)$, where C is some compression function and $s_i \in \mathbb{R}^k$ for $k \ll d$.
- **Process phase:** Use (more compact) dataset $s_1, \ldots, s_n$ to approximately compute something about D.

Sketching phase is easily distributed, parallelized, etc. Better space complexity, communication complexity, runtime, all at once.

How does **Shazam** match a song clip against a library of 8 million songs (32 TB of data) in a fraction of a second?



Spectrogram extracted from audio clip.



Processed spectrogram: used to construct audio “fingerprint” $\mathbf{q} \in \{0,1\}^d$.

Each clip is represented by a high dimensional binary vector $\mathbf{q}$.

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

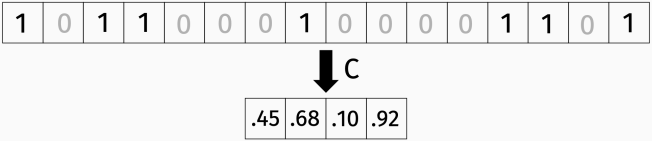
Given $\mathbf{q}$, find any nearby “fingerprint” $\mathbf{y}$ in a database – i.e. any $\mathbf{y}$ with $\text{dist}(\mathbf{y}, \mathbf{q})$ small.

Challenges:

- Database is possibly huge: $O(nd)$ bits.
- Expensive to compute $\text{dist}(\mathbf{y}, \mathbf{q})$: $O(d)$ time.

Goal: Design a more compact sketch for comparing $\mathbf{q}, \mathbf{y} \in \{0,1\}^d$. Ideally $\ll d$ space/time complexity.

$$C(\mathbf{q}) \in \mathbb{R}^k$$
$$C(\mathbf{y}) \in \mathbb{R}^k$$



Homomorphic Compression:

$C(\mathbf{q})$ should be similar to $C(\mathbf{y})$ if $\mathbf{q}$ is similar to $\mathbf{y}$.

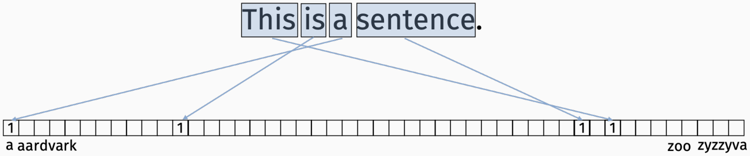
Definition (Jaccard Similarity)

$$J(\mathbf{q}, \mathbf{y}) = \frac{|\mathbf{q} \cap \mathbf{y}|}{|\mathbf{q} \cup \mathbf{y}|} = \frac{\text{\# of non-zero entries in common}}{\text{total \# of non-zero entries}}$$

Natural similarity measure for binary vectors. $0 \leq J(\mathbf{q}, \mathbf{y}) \leq 1$.

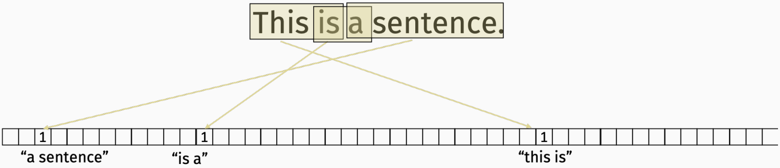
Can be applied to any data which has a natural binary representation (more than you might think).

"Bag-of-words" model:



How many words do a pair of documents have in common?

"Bag-of-words" model:

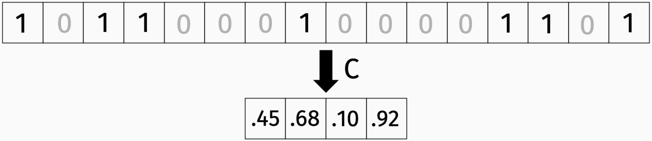


How many bigrams do a pair of documents have in common?

- Finding duplicate or new duplicate documents or webpages.
- Change detection for high-speed web caches.
- Finding near-duplicate emails or customer reviews which could indicate spam.

Other types of data with a natural binary representation?

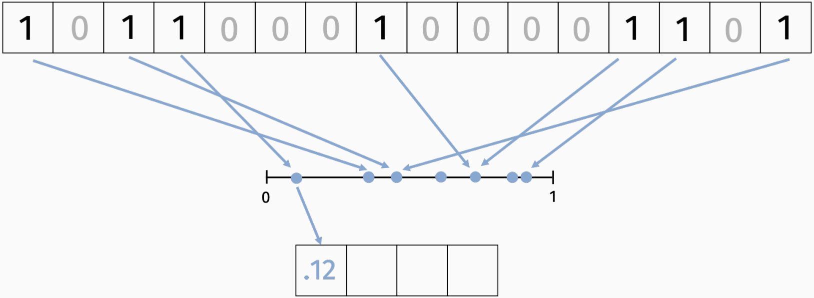
Goal: Design a compact sketch $C : \{0, 1\} \to \mathbb{R}^k$:



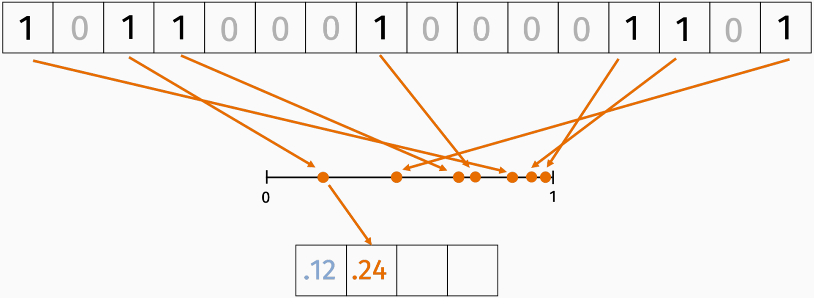
Homomorphic Compression: Want to use $C(\mathbf{q}), C(\mathbf{y})$ to approximately compute the Jaccard similarity $J(\mathbf{q}, \mathbf{y})$.

MinHash (Broder, '97):

- Choose k random hash functions $h_1, \ldots, h_k : \{1, \ldots, n\} \to [0, 1]$.
- For $i \in 1, \ldots, k$, let $c_i = \min_{j, \mathbf{q}_j = 1} h_i(j)$.
- $C(\mathbf{q}) = [c_1, \ldots, c_k]$.

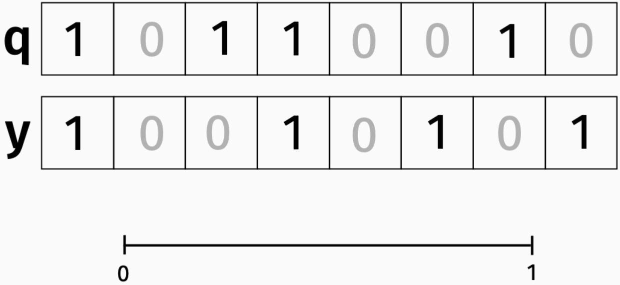


- Choose k random hash functions $h_1, \ldots, h_k : \{1, \ldots, n\} \to [0, 1]$.
- For $i \in 1, \ldots, k$, let $c_i = \min_{j, \mathbf{q}_j = 1} h_i(j)$.
- $C(\mathbf{q}) = [c_1, \ldots, c_k]$.

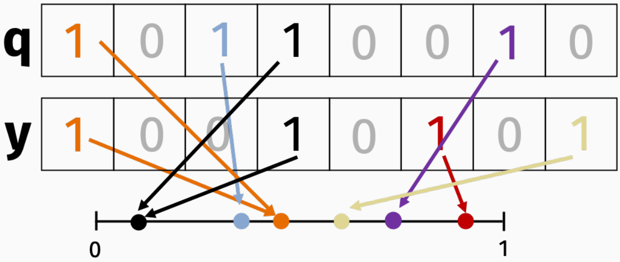


MINHASH ANALYSIS

Claim: $\Pr[c_i(\mathbf{q}) = c_i(\mathbf{y})] = J(\mathbf{q}, \mathbf{y})$.



Claim: $\Pr[c_i(\mathbf{q}) = c_i(\mathbf{y})] = J(\mathbf{q}, \mathbf{y})$.



Every non-zero index in $\mathbf{q} \cup \mathbf{y}$ is equally likely to produce the lowest hash value. $c_i(\mathbf{q}) = c_i(\mathbf{y})$ only if this index is 1 in <u>both</u> $\mathbf{q}$ and $\mathbf{y}$. There are $\mathbf{q} \cap \mathbf{y}$ such indices. So:

$$\Pr[c_i(\mathbf{q}) = c_i(\mathbf{y})] = \frac{\mathbf{q} \cap \mathbf{y}}{\mathbf{q} \cup \mathbf{y}} = J(\mathbf{q}, \mathbf{y})$$

Return: $\tilde{J} = \frac{1}{k}\sum_{i=1}^{k} \mathbb{1}[c_i(\mathbf{q}) = c_i(\mathbf{y})]$.

Unbiased estimate for Jaccard similarity:

$$\mathbb{E}\tilde{J} =$$

C(**q**)

.12	.24	.76	.35

C(**y**)

.12	.98	.76	.11

The more repetitions, the lower the variance.

Let $J = J(\mathbf{q}, \mathbf{y})$ denote the true Jaccard similarity.

Estimator: $\tilde{J} = \frac{1}{k}\sum_{i=1}^{k} \mathbb{1}[c_i(\mathbf{q}) = c_i(\mathbf{y})]$.

$$\text{Var}[\tilde{J}] =$$

Plug into Chebyshev inequality. How large does k need to be so that with probability $> 1 - \delta$:

$$|J - \tilde{J}| \leq \epsilon?$$

Chebyshev inequality: As long as $k = O\left(\frac{1}{\epsilon^2 \delta}\right)$, then with prob. $1 - \delta$,

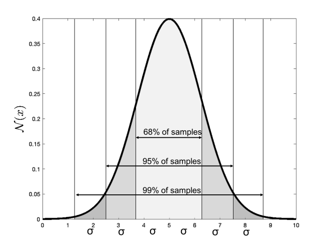
$$J(\mathbf{q}, \mathbf{y}) - \epsilon \leq \tilde{J}(C(\mathbf{q}), C(\mathbf{y})) \leq J(\mathbf{q}, \mathbf{y}) + \epsilon.$$

And $\tilde{J}$ only takes $O(k)$ time to compute! Independent of original fingerprint dimension d.

However, a linear dependence on $\frac{1}{\delta}$ is not good! Suppose we have a database of n songs slips, and Shazam wants to ensure the similarity between a query $\mathbf{q}$ and every song clip $\mathbf{y}$ is approximated well.

We would need $\delta \approx 1/n$. I.e. our compression need to use $k = O(n/\epsilon^2)$ dimensions, which is far too large!

Motivating question: Is Chebyshev's Inequality tight?



68-95-99 rule for Gaussian bell-curve. $X \sim N(0, \sigma^2)$

Chebyshev's Inequality:

$$\Pr(|X - \mathbb{E}[X]| \geq 1\sigma) \leq 100\%$$
$$\Pr(|X - \mathbb{E}[X]| \geq 2\sigma) \leq 25\%$$
$$\Pr(|X - \mathbb{E}[X]| \geq 3\sigma) \leq 11\%$$
$$\Pr(|X - \mathbb{E}[X]| \geq 4\sigma) \leq 6\%.$$

Truth:

$$\Pr(|X - \mathbb{E}[X]| \geq 1\sigma) \approx 32\%$$
$$\Pr(|X - \mathbb{E}[X]| \geq 2\sigma) \approx 5\%$$
$$\Pr(|X - \mathbb{E}[X]| \geq 3\sigma) \approx 1\%$$
$$\Pr(|X - \mathbb{E}[X]| \geq 4\sigma) \approx .01\%$$

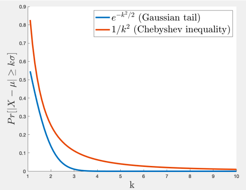
For $X \sim \mathcal{N}(\mu, \sigma^2)$:

$$\Pr[X = \mu \pm x] = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2}$$

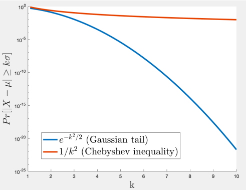
Lemma (Guassian Tail Bound)

For $X \sim \mathcal{N}(\mu, \sigma^2)$:

$$\Pr[|X - \mathbb{E}X| \geq \alpha \cdot \sigma] \leq O(e^{-\alpha^2/2}).$$



Standard y-scale.



Logarithmic y-scale.

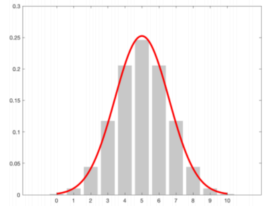
Takeaway: Gaussian random variables concentrate much tighter around their expectation than variance alone predicts.

Why does this matter for algorithm design?

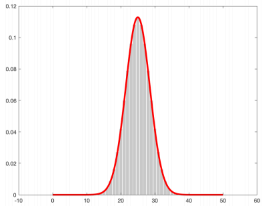
Theorem (CLT – Informal)

Any sum of independent, (identically distributed) r.v.'s $X_1, \ldots, X_k$ with mean μ and finite variance σ^2 converges to a Gaussian r.v. with mean $k \cdot \mu$ and variance $k \cdot \sigma^2$, as $k \to \infty$.

$$S = \sum_{i=1}^{n} X_i \Longrightarrow \mathcal{N}(k \cdot \mu, k \cdot \sigma^2).$$



(a) Distribution of # of heads after 10 coin flips, compared to a Gaussian.



(b) Distribution of # of heads after 50 coin flips, compared to a Gaussian.

Definition (Mutual Independence)

Random variables $X_1, \ldots, X_k$ are mutually independent if, for all possible values $v_1, \ldots, v_k$,

$$\Pr[X_1 = v_1, \ldots, X_k = v_k] = \Pr[X_1 = v_1] \cdot \ldots \cdot \Pr[X_k = v_k]$$

Strictly stronger than pairwise independence.

You have access to a coin and want to determine if it's ϵ-close to unbiased. To do so, you flip the coin repeatedly and check that the ratio of heads flips is between $1/2-\epsilon$ and $1/2+\epsilon$. If it is not, you reject the coin as overly biased.

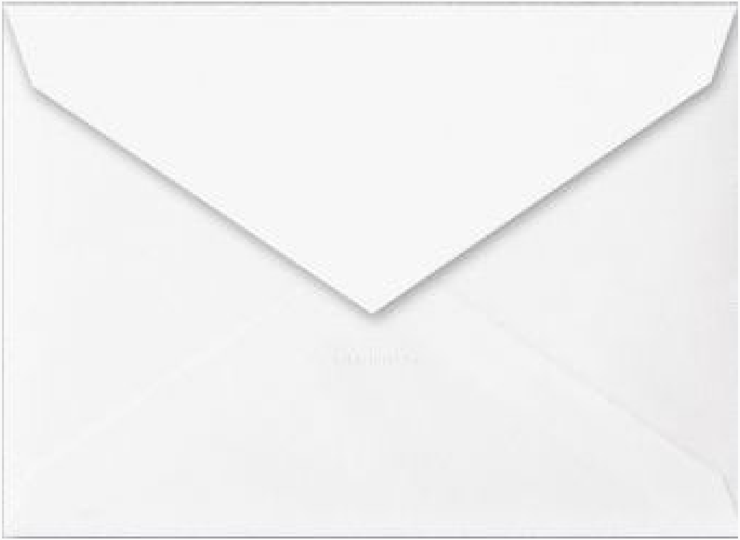
(a) How many flips k are required so that, with probability $(1-\delta)$, you do not accidentally reject a truly unbiased coin? The solution with depend on ϵ and δ.

For this problem, we will assume the CLT holds exactly for a sum of independent random variables – i.e., that this sum looks exactly like a Gaussian random variable.

Lemma (Guassian Tail Bound)

For $X \sim \mathcal{N}(\mu, \sigma^2)$:

$$\Pr[|X - \mathbb{E}X| \geq \alpha \cdot \sigma] \leq O(e^{-\alpha^2/2}).$$



These back-of-the-envelop calculations can be made rigorous! Lots of different "versions" of bound which do so.

- Chernoff bound
- Bernstein bound
- Hoeffding bound
- ...

Different assumptions on random varibles (e.g. binary, bounded, i.i.d), different forms (additive vs. multiplicative error), etc. **Wikipedia is your friend.**

Theorem (Chernoff Bound)

Let $X_1, X_2, \ldots, X_k$ be independent $\{0,1\}$-valued random variables and let $p_i = \mathbb{E}[X_i]$, where $0 < p_i < 1$. Then the sum $S = \sum_{i=1}^{k} X_i$, which has mean $\mu = \sum_{i=1}^{k} p_i$, satisfies

$$\Pr[S \geq (1+\epsilon)\mu] \leq e^{\frac{-\epsilon^2 \mu}{2+\epsilon}}.$$

and for $0 < \epsilon < 1$

$$\Pr[S \leq (1-\epsilon)\mu] \leq e^{\frac{-\epsilon^2 \mu}{2}}.$$

Theorem (Bernstein Inequality)

Let $X_1, X_2, \ldots, X_k$ be independent random variables with each $X_i \in [-1, 1]$. Let $\mu_i = \mathbb{E}[X_i]$ and $\sigma_i^2 = \mathrm{Var}[X_i]$. Let $\mu = \sum_i \mu_i$ and $\sigma^2 = \sum_i \sigma_i^2$. Then, for $\alpha \leq \frac{1}{2}\sigma$, $S = \sum_i X_i$ satisfies

$$\Pr[|S - \mu| > \alpha \cdot \sigma] \leq 2\exp(-\frac{\alpha^2}{4}).$$

Theorem (Hoeffding Inequality)

Let $X_1, X_2, \ldots, X_k$ be independent random variables with each $X_i \in [a_i, b_i]$. Let $\mu_i = \mathbb{E}[X_i]$ and $\mu = \sum_i \mu_i$. Then, for any $\alpha > 0$, $S = \sum_i X_i$ satisfies:

$$\Pr[|S - \mu| > \alpha] \leq 2\exp\left(-\frac{\alpha^2}{\sum_{i=1}^{k}(b_i - a_i)^2}\right).$$

Variance is a natural measure of central tendency, but there are others.

$$q^{\text{th}} \text{ central moment: } \mathbb{E}[(X - \mathbb{E}X)^q]$$

$k = 2$ gives the variance. Proof of Chebyshev's applies Markov's inequality to the random variable $(X - \mathbb{E}X)^2$).

Idea in brief: Apply Markov's inequality to $\mathbb{E}[(X - \mathbb{E}X)^q$ for larger q, or more generally to $f(X - \mathbb{E}X)$ for some other non-negative function f. E.g., to $\exp(X - \mathbb{E}X)$.

We will explore this approach in the next problem set.

Sample Application: Flip biased coin k times: i.e. the coin is heads with probability b. As long as $k \geq O\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$,

$$\Pr[|\text{\# heads} - b \cdot k| \geq \epsilon k] \leq \delta$$

Setup: Let $X_i = \mathbb{1}[i^{\text{th}} \text{ flip is heads}]$. Want bound probability that $\sum_{i=1}^{k} X_i$ deviates from it's expectation.

Corollary of Chernoff bound: Let $S = \sum_{i=1}^{k} X_i$ and $\mu = \mathbb{E}[S]$. For $0 < \Delta < 1$,

$$\Pr[|S - \mu| \geq \Delta\mu] \leq 2e^{-\Delta^2 \mu/3}$$

Sample Application: Flip biased coin k times: i.e. the coin is heads with probability b. As long as $k \geq O\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$,

$$\Pr[|\#\text{ heads} - b \cdot k| \geq \epsilon k] \leq \delta$$

Pay very little for higher probability – if you increase the number of coin flips by 2x, δ goes from $1/10 \rightarrow 1/100 \rightarrow 1/10000$

Let $J = J(\mathbf{q}, \mathbf{y})$ denote the true Jaccard similarity.

Estimator: $\tilde{J} = \frac{1}{k} \sum_{i=1}^{k} \mathbb{1}[c_i(\mathbf{q}) = c_i(\mathbf{y})]$.

By the analysis above,

$$\Pr[|\tilde{J} - J| \geq \epsilon] = \Pr[|\tilde{J} \cdot k - J \cdot k| \geq \epsilon k] \leq \delta$$

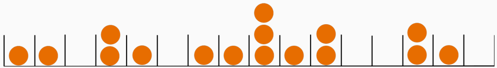
as long as $k = O\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$.

Much better than the $k = O\left(\frac{1}{\delta\epsilon^2}\right)$.

For example, if we had a data base of $n = 1,000,000$ songs, setting $\delta = \frac{1}{n}$ would only require space depending on $\log(n) \approx 14$, instead of on $n = 1,000,000$.

As in the first video lecture, we want to use concentration bounds to study the randomized load balancing problem. n jobs are distributed randomly to n servers using a hash function. Let S_i be the number of jobs sent to server i. What's the smallest B for which we can prove:

$$\Pr[max_i S_i \geq \mathbf{B}] \leq 1/10$$



Recall: Suffices to prove that, for any i, $\Pr[S_i \geq \mathbf{B}] \leq 1/10n$:

$$\begin{aligned}\Pr[max_i S_i \geq \mathbf{B}] &= \Pr[S_1 \geq \mathbf{B} \text{ or } \ldots \text{ or } S_1 \geq \mathbf{B}] \\ &\leq \Pr[S_1 \geq \mathbf{B}] + \ldots + \Pr[S_n \geq \mathbf{B}] \quad \text{(union bound).}\end{aligned}$$

Theorem (Chernoff Bound)

Let $X_1, X_2, \ldots, X_n$ be independent $\{0,1\}$-valued random variables and let $p_i = \mathbb{E}[X_i]$, where $0 < p_i < 1$. Then the sum $S = \sum_{j=1}^n X_i$, which has mean $\mu = \sum_{j=1}^n p_i$, satisfies

$$\Pr[X \geq (1+\epsilon)\mu] \leq e^{\frac{-\epsilon^2 \mu}{3+3\epsilon}}.$$

Consider a single bin. Let $X_j = \mathbb{1}[\text{ball } j \text{ lands in that bin}]$. $\mathbb{E}[X_j] = \frac{1}{n}$, so $\mu = 1$.

$$\Pr[S \geq (1 + c\log n)\mu] \leq e^{\frac{-c^2 \log^2 n}{c + c\log n}} \leq e^{\frac{-c \log^2 n}{2 \log n}} \leq e^{-.5c\log n} \leq \frac{1}{10n},$$

for sufficiently large c

So max load for randomized load balancing is $O(\log n)$! Best we could prove with Chebyshev's was $O(\sqrt{n})$.

Power of 2 Choices: Instead of assigning job to random server, choose 2 random servers and assign to the least loaded. With probability 1/10 the maximum load is bounded by:

(a) $O(\log n)$
(b) $O(\sqrt{\log n})$
(c) $O(\log logn)$
(d) $O(1)$