CSE 312

Foundations of Computing II

Lecture 9: Pairwise-Independent Hashing

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

Stefano Tessaro

tessaro@cs.washington.edu

This week – Applications + Random Variables

- **Today:** Data structures!
  - The power of pairwise-independence
- **Wednesday:** (Simple) Machine Learning
  - Naïve Bayes Learning
  - (Optional) Project
- **Friday:** Random Variables

Last time – Refresher

Definition. The events $\mathcal{A}_1, \dots, \mathcal{A}_n$ are **independent** if for every $k \leq n$ and $1 \leq j_1 < j_2 < \cdots < j_k \leq n$,

$$\mathbb{P}\left(\mathcal{A}_{j_1} \cap \mathcal{A}_{j_2} \cap \cdots \cap \mathcal{A}_{j_k}\right) = \mathbb{P}\left(\mathcal{A}_{j_1}\right) \cdot \mathbb{P}\left(\mathcal{A}_{j_2}\right) \cdots \mathbb{P}\left(\mathcal{A}_{j_k}\right).$$

Last time – Refresher

Definition. The events $\mathcal{A}_1, \dots, \mathcal{A}_n$ are **independent** if for every $k \leq n$ and $1 \leq j_1 < j_2 < \cdots < j_k \leq n$,

$$\mathbb{P}\left(\mathcal{A}_{j_1} \cap \mathcal{A}_{j_2} \cap \cdots \cap \mathcal{A}_{j_k}\right) = \mathbb{P}\left(\mathcal{A}_{j_1}\right) \cdot \mathbb{P}\left(\mathcal{A}_{j_2}\right) \cdots \mathbb{P}\left(\mathcal{A}_{j_k}\right).$$

Definition. The events $\mathcal{A}_1, \dots, \mathcal{A}_n$ are **pairwise-independent** if for all distinct $i, j \in [n]$,

$$\mathbb{P}\left(\mathcal{A}_i \cap \mathcal{A}_j\right) = \mathbb{P}(\mathcal{A}_i) \cdot \mathbb{P}(\mathcal{A}_j).$$

Today: Application to CS of pairwise-independence!

Basic Problem

Problem: Store a subset S of a large set X.

Example. $X =$ set of all US ZIP codes $|X| \approx 42000$

$S =$ set of ZIP codes of CSE 312 students $|S| \approx 50$

Two goals:

1. **Constant-time** answering of queries "Is $x \in S$?"
2. **Minimize storage** requirements.

Imagine for simplicity $X = \{1, \dots, K\} = [K]$

Naïve Solution – Constant Time

Idea: Represent S as an array a with K entries.

$$a[i] = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{if } i \notin S \end{cases}$$

$S = \{1,3, \ldots, K-1\}$

1	2	3	4	5	…	$K-1$	K
1	0	1	0	0	…	1	0

Membership test: To check $i \in S$ just check whether $a[i] = 1$.

$\rightarrow$ **constant time!**

Storage: Require storing K bits, even for small S.

Naïve Solution – Small Storage

Idea: Represent S as a list with $|S|$ entries.

$S = \{1,3,\ldots,K-1\}$

Storage: Grows with $|S|$ <u>only</u> 👍 😀

Membership test: Check $i \in S$ requires time linear in $|S|$

(Can be made logarithmic by using a tree) 👎 😢

Today – Hash Table

$$a[\mathbf{h}(i)] = \begin{cases} i & \text{if } i \in S \\ 0 & \text{if } i \notin S \end{cases}$$

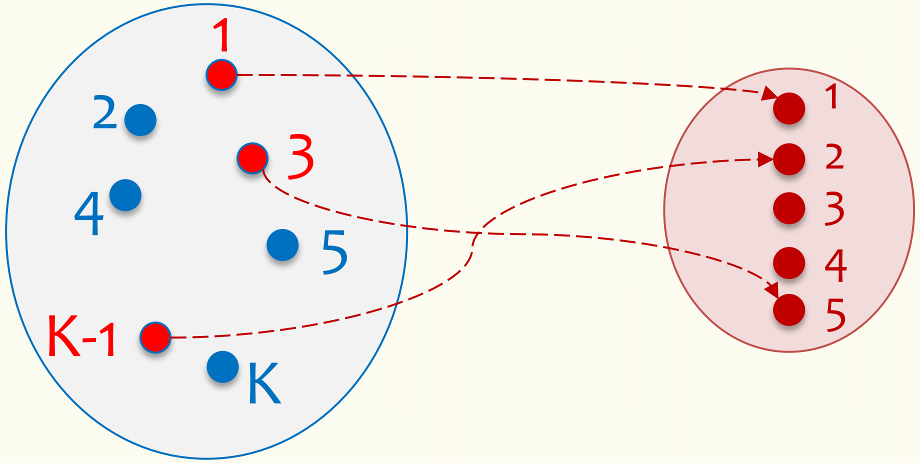
Idea: Represent S as an array a with $M \ll K$ entries.

$S = \{1,3, \ldots, K-1\}$

1	2	3	4	5
1	$K-1$	0	0	3

$M = 5$

Membership test: To check $i \in S$ just check whether $a[\mathbf{h}(i)] = i$.



Storage: M elements from $\{0\} \cup [K]$

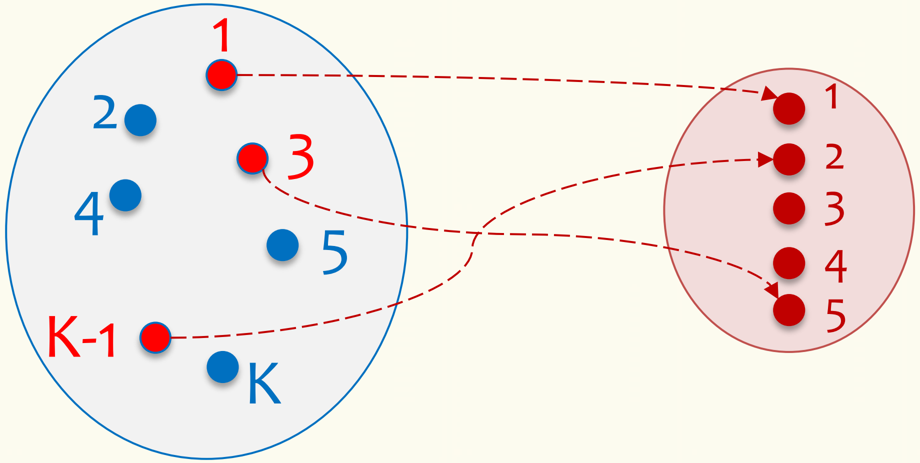
hash function h: $[\mathrm{K}] \to [M]$

Our Solut

Challenge 1: Ensure $\mathbf{h}(i) \neq \mathbf{h}(j)$ for all $i, j \in S$

$$a[\mathbf{h}(i)] = \begin{cases} i & \text{if } i \in S \\ 0 & \text{if } i \notin S \end{cases}$$

Membership test: To check $i \in S$ just check whether $a[\mathbf{h}(i)] = i$.



Storage: M elements from $\{0\} \cup [K]$

hash function h: $[\mathrm{K}] \rightarrow [M]$

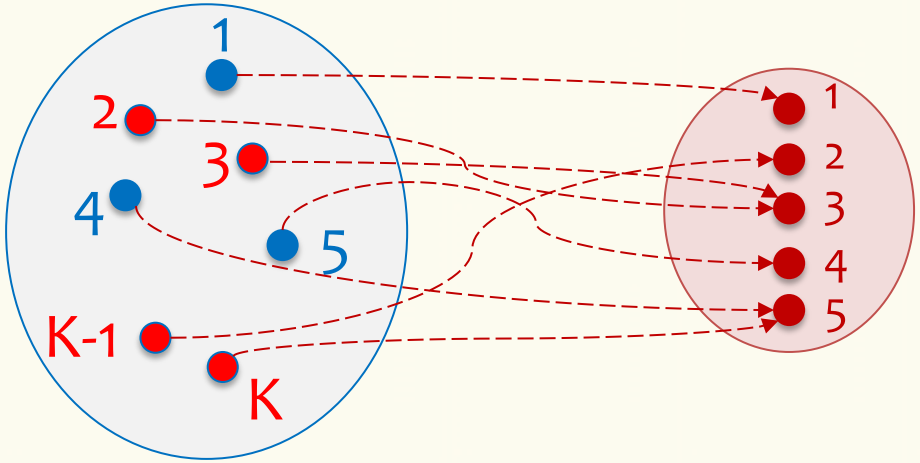
Challenge 2: Ensure $M \approx |S|$

We will show today $M \approx |S|^2$

Our Solut

Challenge 1: Ensure $\mathbf{h}(i) \neq \mathbf{h}(j)$ for all $i, j \in S$

hash function h: $[K] \to [M]$



Membership test: To check $i \in S$ just check whether $a[\mathbf{h}(i)] = i$.

Impossible! Because $M < K$, for every $\mathbf{h}$, we can <u>always</u> come up with a set S where this is not true!

(By the pigeonhole principle)

Solution: We will pick $\mathbf{h}$ <u>randomly</u> and show it is good for S with good probability (e.g., $\geq 1/2$)

How to choose **h**?

Fix set $S \subseteq [K]$ with n elements. Wlog $S = \{1, \ldots, n\}$

First idea: Pick $\mathbf{h}: [K] \to [M]$ randomly from the set of all functions.

Theorem. $\mathbb{P}(\exists i \neq j: \mathbf{h}(i) = \mathbf{h}(j)) \leq \frac{n(n-1)}{2M}$

Set $M = n^2 = |S|^2$ for probability $< \frac{1}{2}$

Note: This will not be a good idea in the end. Why? We need to store entire description of **h**! Let's stick with it for now.

Proof – Random Hash

$\Omega = \{\mathbf{h} \mid \mathbf{h}: [K] \to [M]\}$

$\mathbb{P}(\mathbf{h}) = \dfrac{1}{M^K}$

$\mathcal{C} = \{\mathbf{h} \mid \exists i \neq j: \mathbf{h}(i) = \mathbf{h}(j)\}$

For every $i < j$: $\mathcal{C}_{i,j} = \{\mathbf{h} \mid \mathbf{h}(i) = \mathbf{h}(j)\}$

Claim. $\mathcal{C} = \mathcal{C}_{1,2} \cup \mathcal{C}_{1,3} \cup \cdots \mathcal{C}_{n-1,n} = \bigcup_{i<j} \mathcal{C}_{i,j}$

"Proof": $\mathcal{C}$ happens if and only if ($\mathbf{h}(1) = \mathbf{h}(2)$ or $\mathbf{h}(1) = \mathbf{h}(3)$ or $\mathbf{h}(1) = \mathbf{h}(4)$ or … or $\mathbf{h}(n-1) = \mathbf{h}(n)$)

Proof – Random Hash

$\Omega = \{\mathbf{h} \mid \mathbf{h}: [K] \to [M]\}$

$\mathbb{P}(\mathbf{h}) = \dfrac{1}{M^K}$

For every $i < j$: $\mathcal{C}_{i,j} = \{\mathbf{h} \mid \mathbf{h}(i) = \mathbf{h}(j)\}$

Claim. For all $i < j$, $\mathbb{P}(\mathcal{C}_{i,j}) = \frac{1}{M}$

Proof: Let $\mathcal{A}_i(y) = \{\mathbf{h} \mid \mathbf{h}(i) = y\}$ [i.e., we pick a function that maps i to y.]

$$\mathbb{P}(\mathcal{C}_{i,j}) = \sum_y \mathbb{P}(\mathcal{A}_i(y) \cap \mathcal{A}_j(y))$$

Note that $\mathbb{P}(\mathcal{A}_i(y)) = \mathbb{P}\big(\mathcal{A}_j(y)\big) = \dfrac{M^{K-1}}{M^K} = \dfrac{1}{M}$

$$\mathbb{P}\Big(\mathcal{A}_i(y) \cap \mathcal{A}_j(y)\Big) = \frac{M^{K-2}}{M^K} = \frac{1}{M^2} = \frac{1}{M} \cdot \frac{1}{M}$$

Independent!

Proof – Random Hash

$\Omega = \{\mathbf{h} \mid \mathbf{h}: [K] \to [M]\}$

$\mathbb{P}(\mathbf{h}) = \dfrac{1}{M^K}$

For every $i < j$: $\mathcal{C}_{i,j} = \{\mathbf{h} \mid \mathbf{h}(i) = \mathbf{h}(j)\}$

Claim. For all $i < j$, $\mathbb{P}(\mathcal{C}_{i,j}) = \frac{1}{M}$

Proof: Let $\mathcal{A}_i(y) = \{\mathbf{h} \mid \mathbf{h}(i) = y\}$ [i.e., we pick a function that i maps to y.]

$$\mathbb{P}(\mathcal{C}_{i,j}) = \sum_y \mathbb{P}(\mathcal{A}_i(y) \cap \mathcal{A}_j(y)) = \sum_y \mathbb{P}(\mathcal{A}_i(y)) \cdot \mathbb{P}(\mathcal{A}_j(y))$$

$$= \sum_y \frac{1}{M^2} = M \times \frac{1}{M^2} = \frac{1}{M}$$

Proof – Random Hash

$$\mathcal{C} = \bigcup_{i<j} \mathcal{C}_{i,j} \qquad \mathbb{P}(\mathcal{C}_{i,j}) = \frac{1}{M}$$

Claim. For all $i < j$, $\mathbb{P}(\mathcal{C}_{i,j}) = 1/M$

$$\mathbb{P}(\mathcal{C}) = \mathbb{P}(\cup_{i<j} \mathcal{C}_{i,j}) \leq \sum_{i<j} \mathbb{P}(\mathcal{C}_{i,j}) = \sum_{i<j} \frac{1}{M} = \binom{n}{2}\frac{1}{M} = \frac{n(n-1)}{2M}$$

Union bound: $\mathbb{P}(\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_n) \leq \mathbb{P}(\mathcal{A}_1) + \cdots + \mathbb{P}(\mathcal{A}_n)$

$\blacksquare$

Theorem. $\mathbb{P}(\exists i \neq j : \mathbf{h}(i) = \mathbf{h}(j)) \leq \frac{n(n-1)}{2M}$

Back to Data Structures

Problem: Description of **h**: $[K] \rightarrow [M]$ needs to be stored along with the set S.

Need to store K elements from $[M]$. 👎 😢

Our proof did not need **h** to be picked at random from all functions …

Claim. For all $i < j$, $\mathbb{P}(\mathcal{C}_{i,j}) = 1/M$

$$\mathbb{P}(\mathcal{C}_{i,j}) = \sum_y \mathbb{P}(\mathcal{A}_i(y) \cap \mathcal{A}_j(y)) = \sum_y \mathbb{P}\big(\mathcal{A}_i(y)\big)\, \mathbb{P}(\mathcal{A}_j(y))$$

$$= \sum_y \frac{1}{M^2} = M \times \frac{1}{M^2} = \frac{1}{M}$$

This only requires pairwise independence of the $\mathcal{A}_i(y)$'s

Pairwise-Independent Functions

Definition. A <u>set</u> H of functions $[K] \to [M]$ is **pairwise independent** if for all distinct $i \neq j$, and all $y, y' \in [M]$

$$|\{\mathbf{h} \in H \mid \mathbf{h}(i) = y \wedge \mathbf{h}(j) = y'\}| = \frac{|H|}{M^2}$$

Now: Pick $\mathbf{h}$: $[K] \to [M]$ randomly from pairwise-independent H .

Theorem. $\mathbb{P}(\exists i \neq j: \mathbf{h}(i) = \mathbf{h}(j)) \leq \frac{n(n-1)}{2M}$

Proof as before: Only one step different (next slide)

Pairwise-Independent Functions

Definition. A set H of functions $[K] \to [M]$ is **pairwise independent** if for all distinct $i \neq j$, and all $y, y' \in [M]$

$$|\{\mathbf{h} \in H \mid \mathbf{h}(i) = y \wedge \mathbf{h}(j) = y'\}| = \frac{|H|}{M^2}$$

Let $\mathcal{A}_i(y) = \{\mathbf{h} \in H \mid \mathbf{h}(i) = y\}$

$$\mathbb{P}\left(\mathcal{A}_i(y) \cap \mathcal{A}_j(y)\right) = \frac{|\{\mathbf{h} \in H \mid \mathbf{h}(i) = y \wedge \mathbf{h}(j) = y'\}|}{|H|} = \frac{1}{M^2}$$

This is all we needed!

Pairwise-Independent Functions

Fact: The set of all functions $[K] \rightarrow [M]$ is pairwise independent

– Size M^K

Pairwise-Independent Functions

Fact (informal)*: There exists a pairwise-independent set H of functions $[K] \to [M]$ with size $|H| = K^2$

- Described by two elements of $[K]$.
- Idea*: $x \to (ax + b \bmod K) \bmod M$ i.e., function described by a,b in $[K]$.
- Overall solution takes storing $|S|^2 + 2$ elements from $[K] \cup \{0\}$ (i.e., array + description of a chosen good function)

Several other applications: Data structures, algorithms, cryptography, …

*Some cheating here, as usually one gets an approximation of a pairwise independent hash function, where $\mathbb{P}\left(\mathcal{A}_i(y) \cap \mathcal{A}_j(y)\right) \approx \mathbb{P}(\mathcal{A}_i(y)) \cdot \mathbb{P}\left(\mathcal{A}_j(y)\right)$