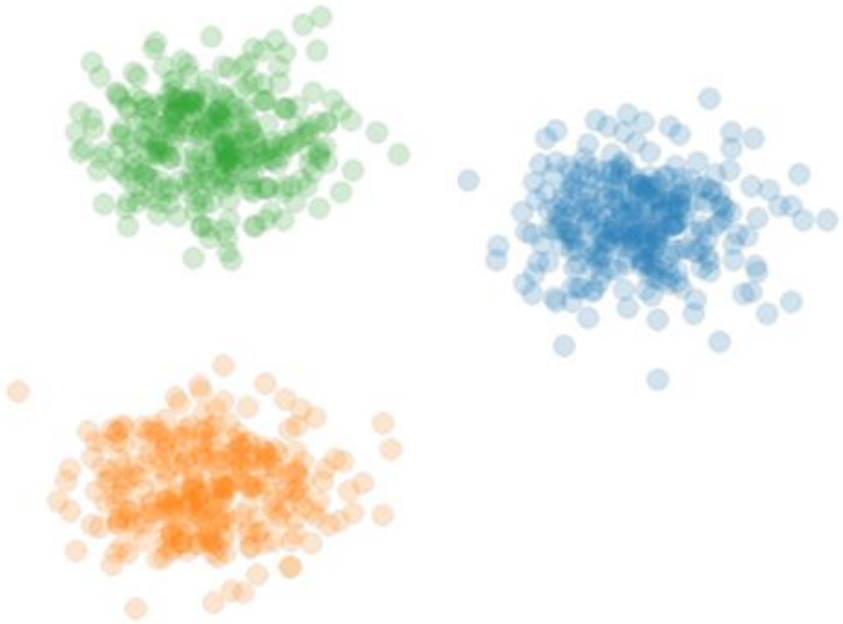
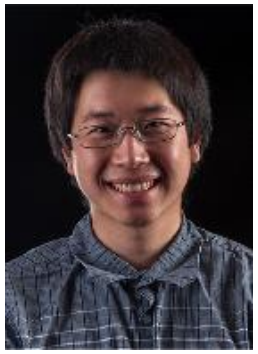
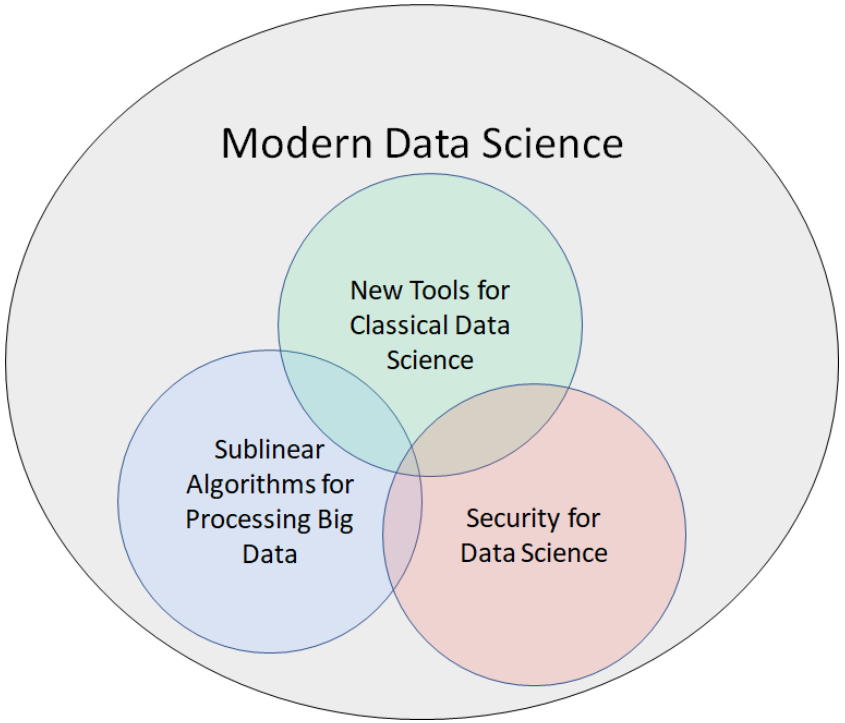
Berkeley
UNIVERSITY OF CALIFORNIA

Theoretical Foundations of Modern Data Science



Samson Zhou





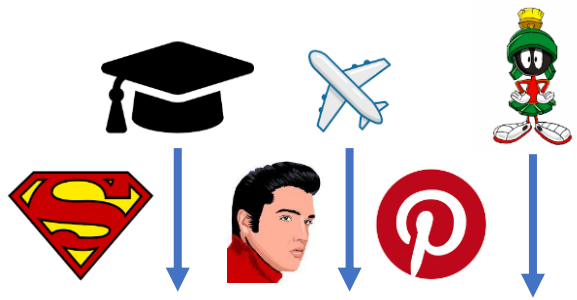
Background

- Post-doc at UC Berkeley / Rice
- Previous: Post-doc at Carnegie Mellon
- PhD in Computer Science from Purdue
- Dual Bachelors in Math, Computer Science from MIT

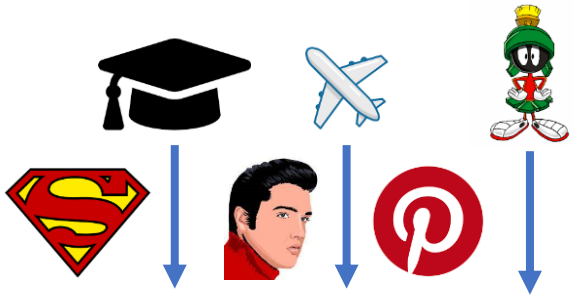
- Research areas: Data science, sublinear algorithms, security and privacy

Data Science


LAMBORGHINI
GOLDEN STATE
WARRIORS
ESPN
COLLEGE
GAMEDAY

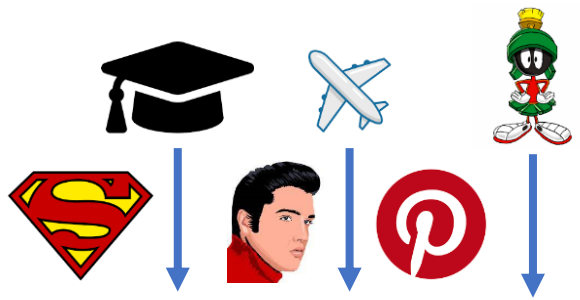



LAMBORGHINI
GOLDEN STATE WARRIORS
ESPN COLLEGE GAMEDAY



1 1 0 1 0 1

0

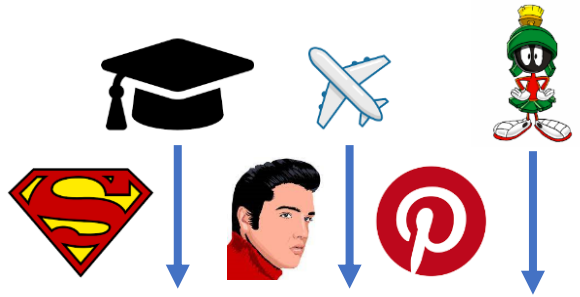


1 1 0 1 0 1

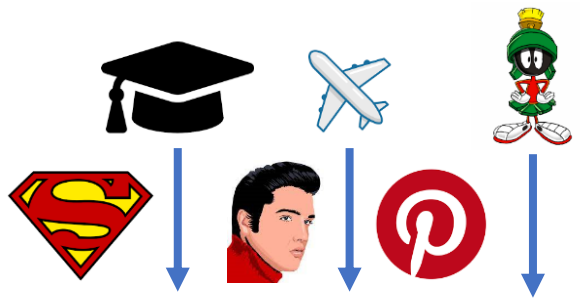
0 1 1 1 0 0

0

1



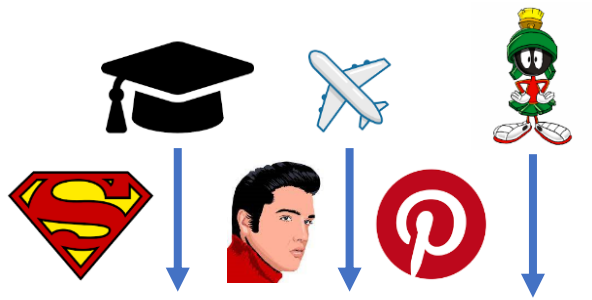
1	1	0	1	0	1	0
0	1	1	1	0	0	1
0	1	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	1	1	1	0
0	1	1	1	0	1	1
0	0	1	0	1	1	0



$$A = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

n

d



$$A = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}_{n \times d} \qquad b = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Mark Rober

@MarkRober
23.3M subscribers

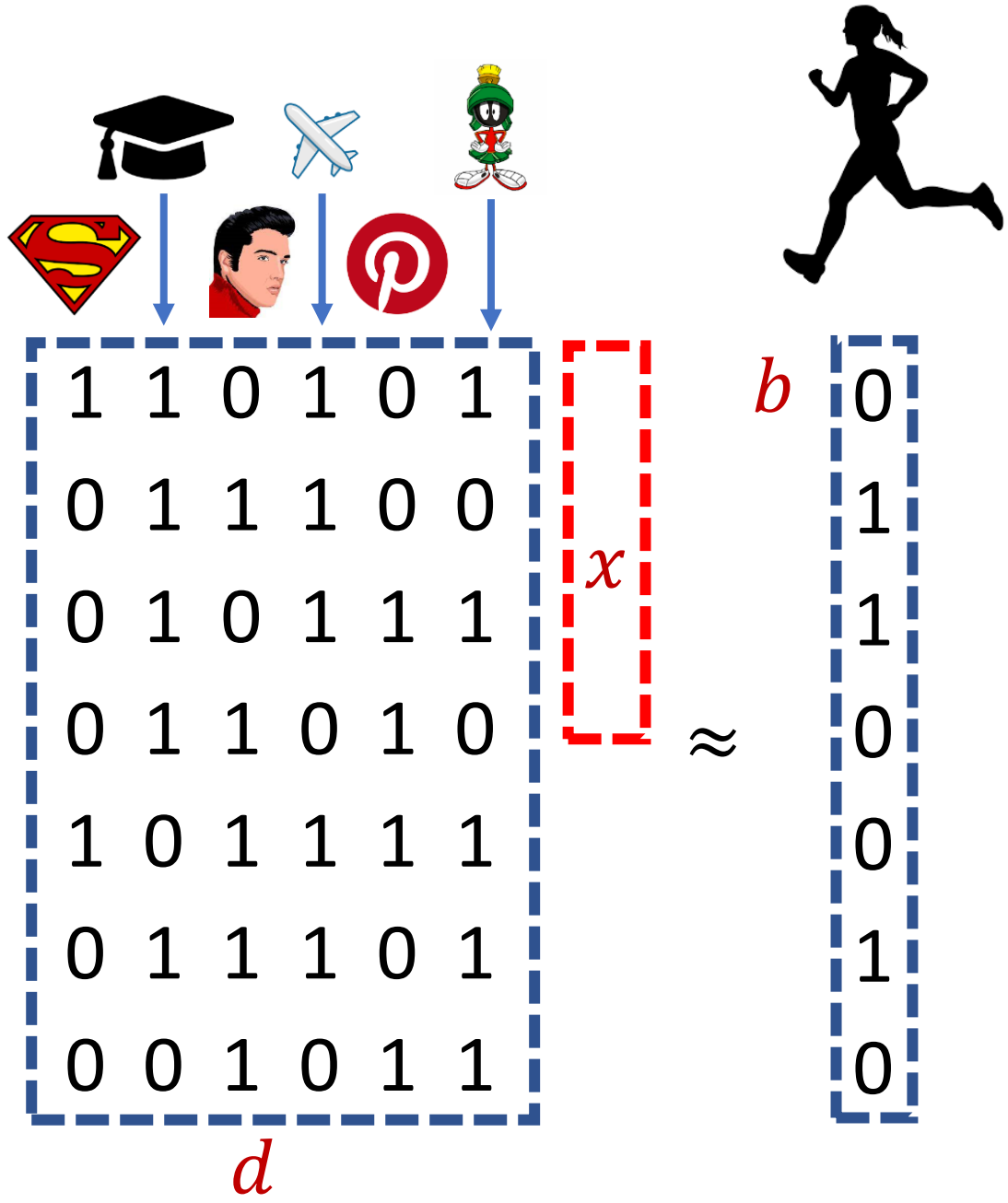
HORTS LIVE PLAYLISTS COMMUNITY STORE CHANNELS ABOUT

Stealing Baseball Signs with a Phone (Machine Learning)

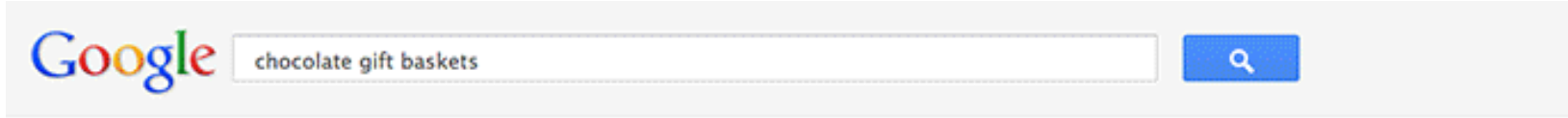
Mark Rober 24M views • 3 years ago

I always sucked at baseball... until now... ok, I still probably suck. Go subscribe to Jabril's channel!!! https://www.youtube.com/channel/UCQALLeQPoZdZC4JNUboVEUg Simple app (it's actually...

CC


A
1 1 0 1 0 1
0 1 1 1 0 0
0 1 0 1 1 1
n
0 1 1 0 1 0
1 0 1 1 1 1
0 1 1 1 0 1
0 0 1 0 1 1
d
x
≈
b
0
1
1
0
0
1
0
1


LAMBORGHINI
GOLDEN STATE WARRIORS
ESPN COLLEGE GAMEDAY



Search

About 13,400,000 results (0.44 seconds)

Warning: This page is a tool for AdWords advertisers to test their ads. For full Google functionality return to the Google homepage.

Web
Images
Maps
Videos
News
More

Auckland
Change location

The web
Pages from New Zealand

More search tools

Ads related to **chocolate gift baskets**

Gift Baskets | Beautiful**Baskets**.co.nz
www.beautiful**baskets**.co.nz/
NZ's Favourite **Gift Baskets** Fast Nationwide NZ Delivery

Chocolates Delivered Fast - Delicious Handmade NZ **Chocolates**.
www.devonport**chocolates**.co.nz/
Browse Our Website & Order Online.
Devonport, 17 Wynyard Street, Auckland - 0800 002 462 - Directions
Chocolate Bars - Weddings & Special Occasions - Birthday Gifts - Corporate Gifts

Great **Gift Baskets** Online - Stunning Hampers for all Occasions.
www.wineplus.co.nz/**gift-baskets**
Fast Delivery NZ-wide - Buy Online.

Chocolate Gift Baskets - Auckland Flowers and Gifts
www.nzflower.co.nz/.../**gift_baskets_chocolate**_new_zealand_auckla...
Chocolate Gift Baskets: Specializing in Chocolate Baskets, Chocolate Gifts, Gourmet **Chocolate Gift Baskets**, Chocolate Lovers Gift Baskets. Easy and Secure ...

Chocolate Gift Baskets | Bliss Baskets and Gifts
www.bliss**giftbaskets**.co.nz/style/**chocolate-gift-baskets**
With their wide range of variety and versatility, **chocolate gift baskets** are a popular gift that is always appreciated by the recipient.

Gift Ideas, **Chocolate** Bouquets | Edible Blooms NZ
www.edibleblooms.co.nz/
Edible Blooms New Zealand offers a unique twist on flowers and **gift hampers**. Our range of **chocolate** bouquets, fresh fruit bouquets and gourmet **gift baskets** ...

Gift Baskets for **Chocolate** Lovers::My Goodness **Gift Baskets** New ...

Ads

Order **Gift Baskets** Online
www.hamperbiz.co.nz/**Gifts**
Stunning Range of **Gifts**, Hampers Wine, Flowers, Free Delivery.

Gift Baskets Gourmet Food
www.**gift**barn.co.nz/
Fine Wine & **Chocolates**
Next day delivery New Zealand Wide

Chocolate delivery
www.edibleblooms.co.nz/
Chocolate Bouquets The Perfect **Gift** Order New Zealand Wide Online Now

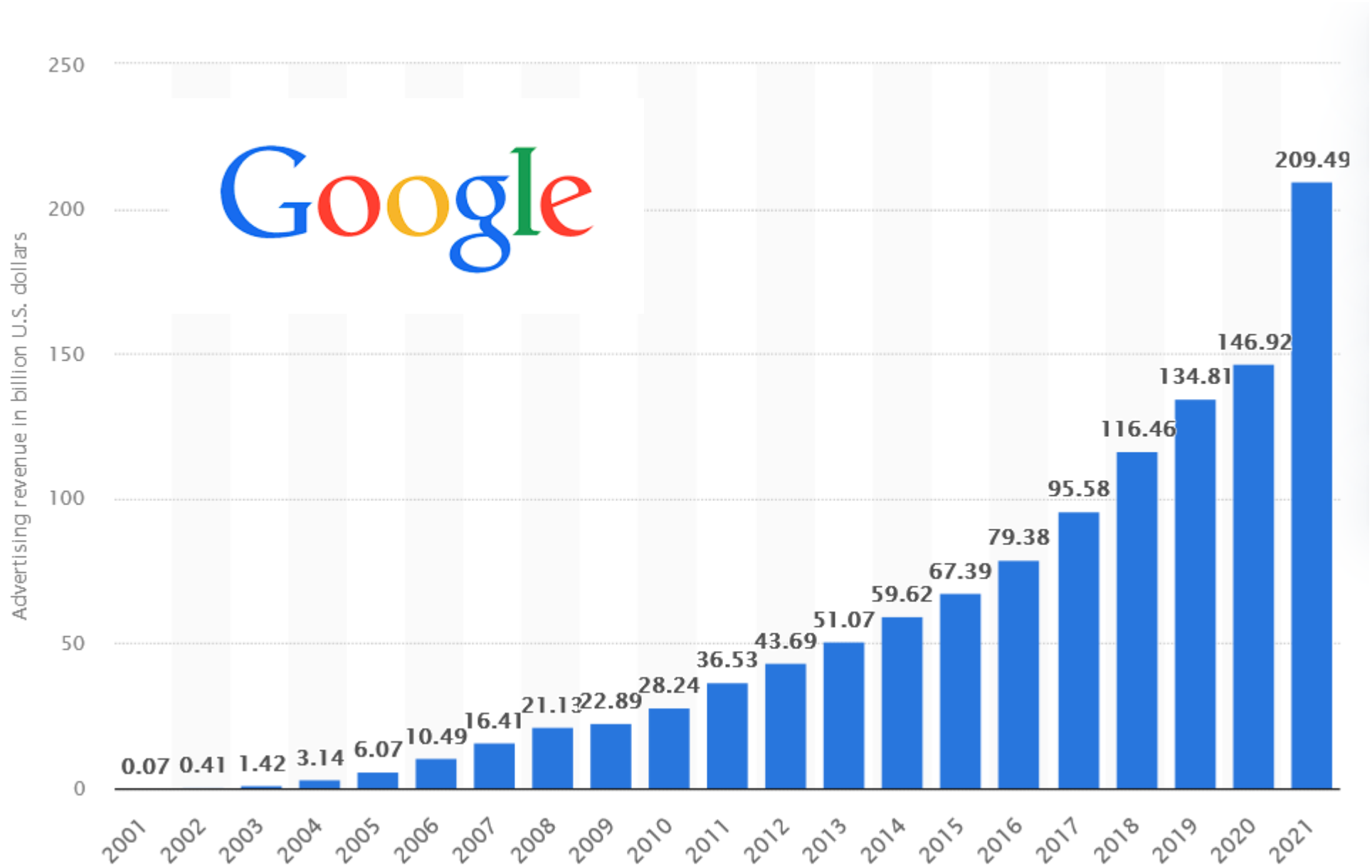
Top Christmas Hampers
www.champershampers.co.nz/
The highest quality food and wine corporate hampers, Christmas hams

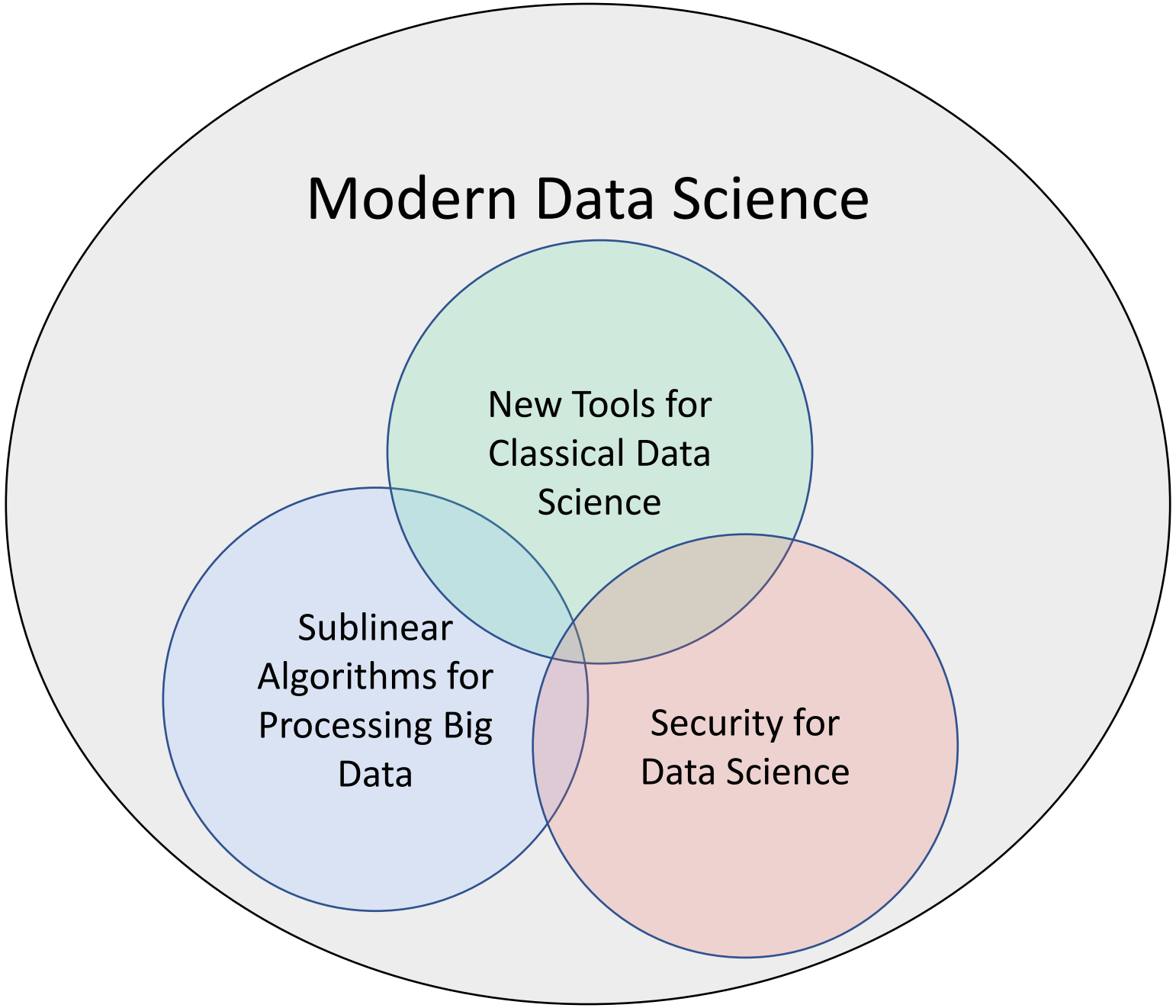
NZ's Top **Gift Baskets**
www.mygoodness.co.nz/
Quality, Stylish Hampers from Award Winning, Great Service Kiwi Company

Gift Baskets Delivered NZ
www.raptabout**gifts**.co.nz/**Gifts**
Huge Selection of **Gift Baskets**
Browse our Online Store Now!

Fine **Chocolate** Delivery

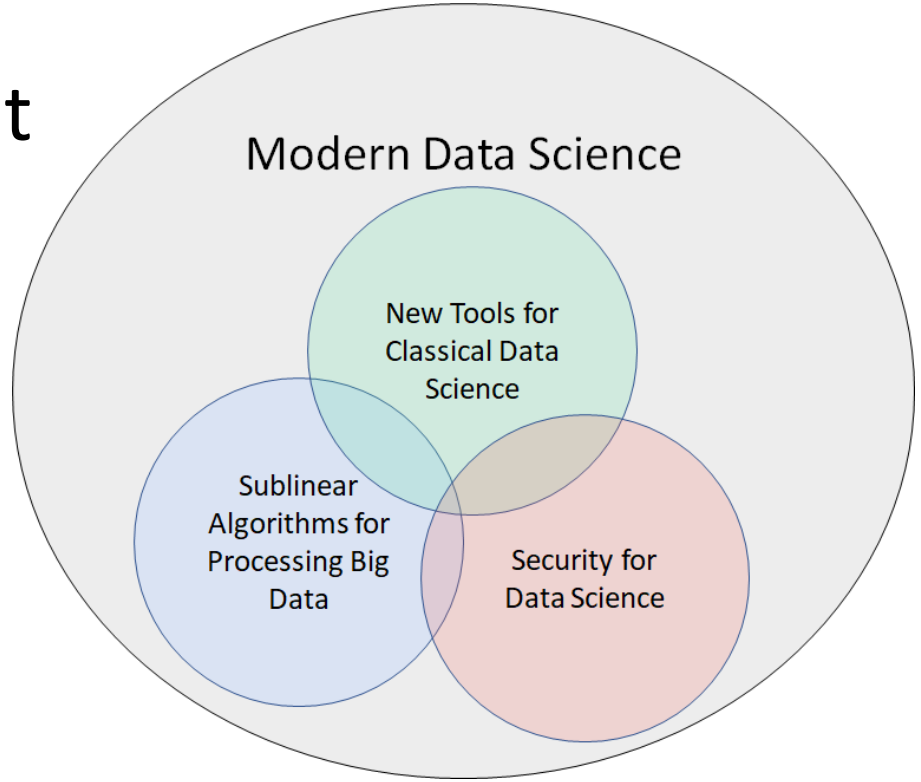
Google Ad Revenue (2001-2021, billion USD)




Modern Data Science
New Tools for Classical Data Science
Sublinear Algorithms for Processing Big Data
Security for Data Science

Evolving Demands

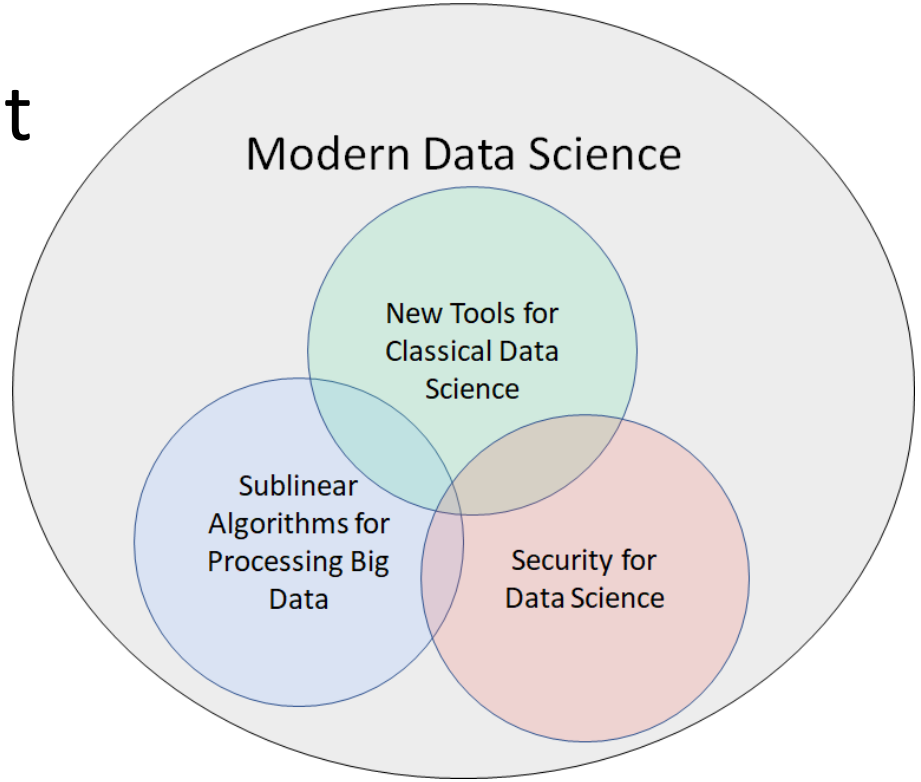
- Sublinear-time or sublinear-space algorithms
- Incorporation of advice
- Security for data science
- Robustness to noise or adversarial input
- Ability to handle time-sensitive data

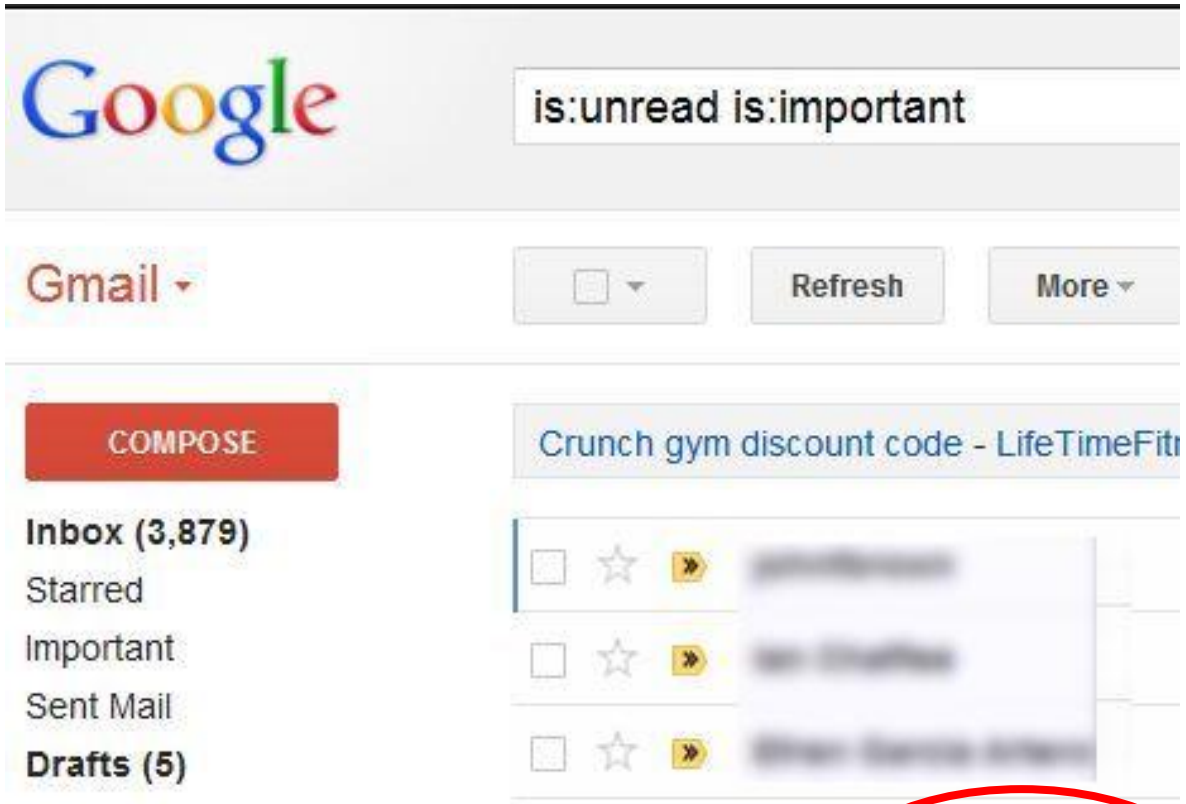


New Tools for Classical Data Science

Evolving Demands

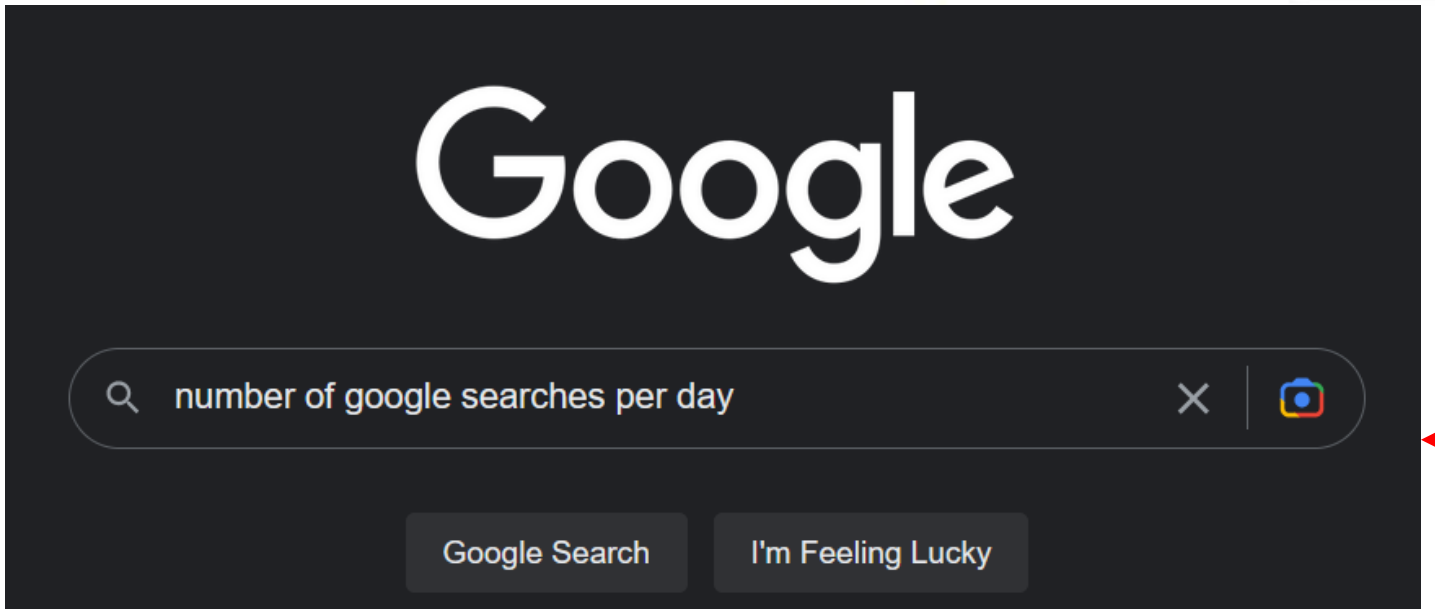
- Sublinear-time or sublinear-space algorithms
- Incorporation of advice
- Security for data science
- Robustness to noise or adversarial input
- Ability to handle time-sensitive data



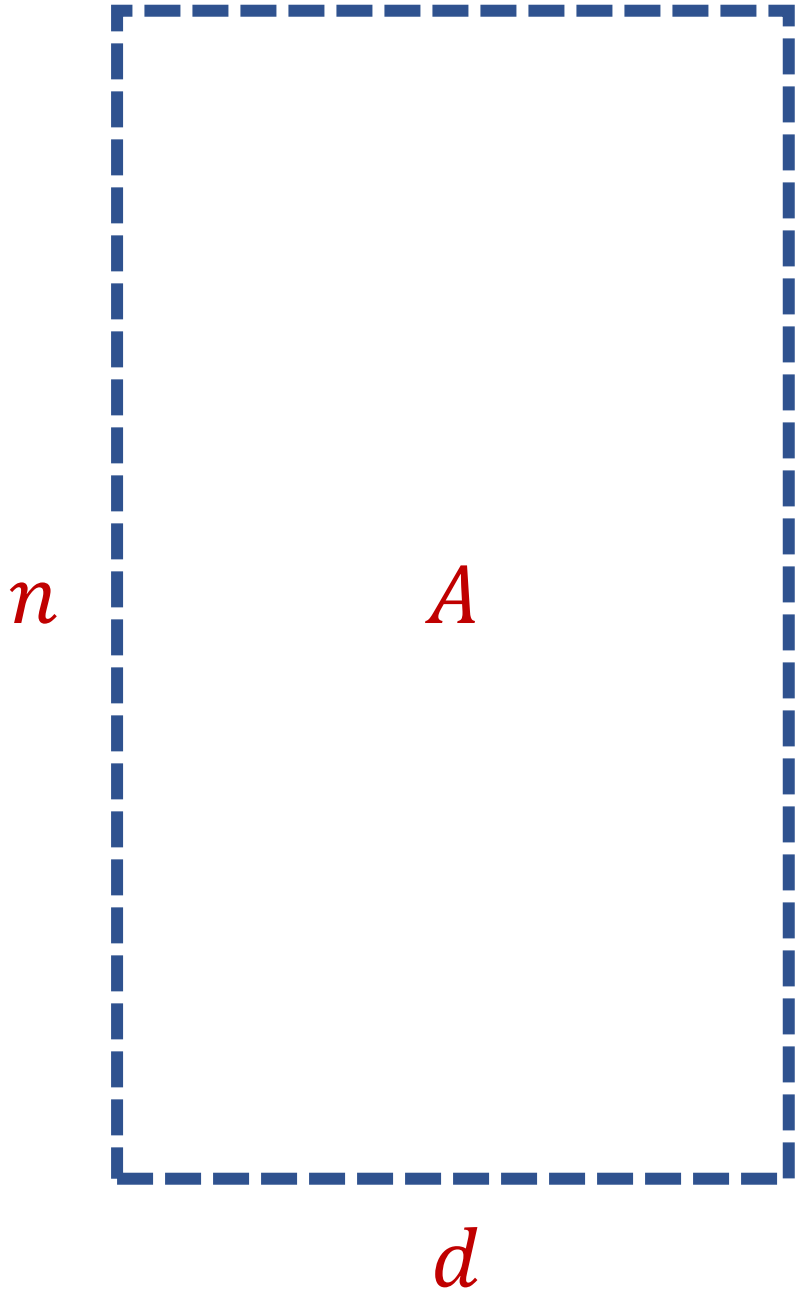


3 billion monthly active users

330 billion daily e-mails



8.5 billion daily Google searches



Dimensionality Reduction

- Reduce d
- Reduce n

Sampling, Sketching, Johnson-Lindenstrauss, IsoMap

Sampling, Sketching, Coresets

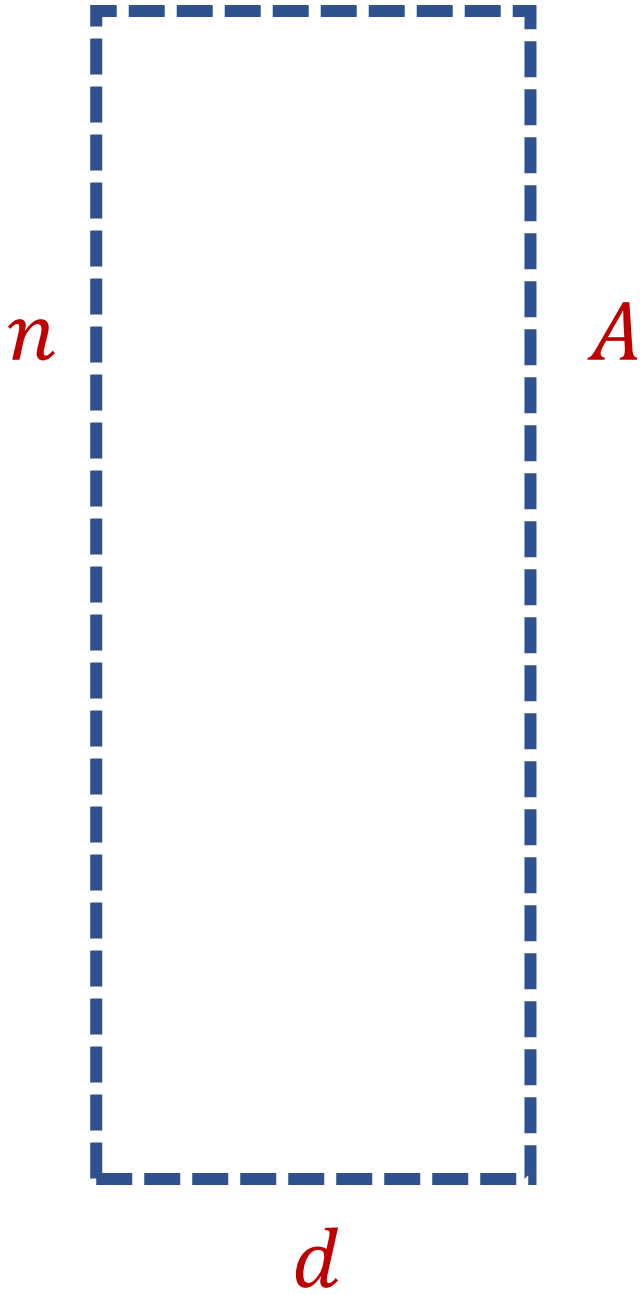
- Application depends on task!

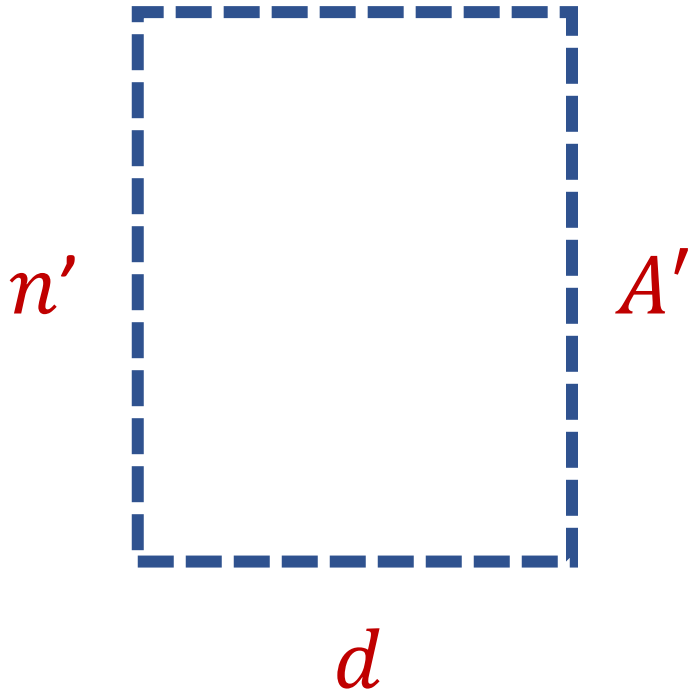
New Dimensionality Reduction Results

- Dimensionality reduction for Wasserstein barycenter [ISZ21]
- Sketching for valuation functions, e.g., additive, submodular, Lipschitz [YZ19]
- Kernel density estimation on kernel matrices [BKISZ22]

Coreset

- Subset A' of representative rows of A for a given task with "score" function f
- $f(A,\cdot) \approx f(A',\cdot)$



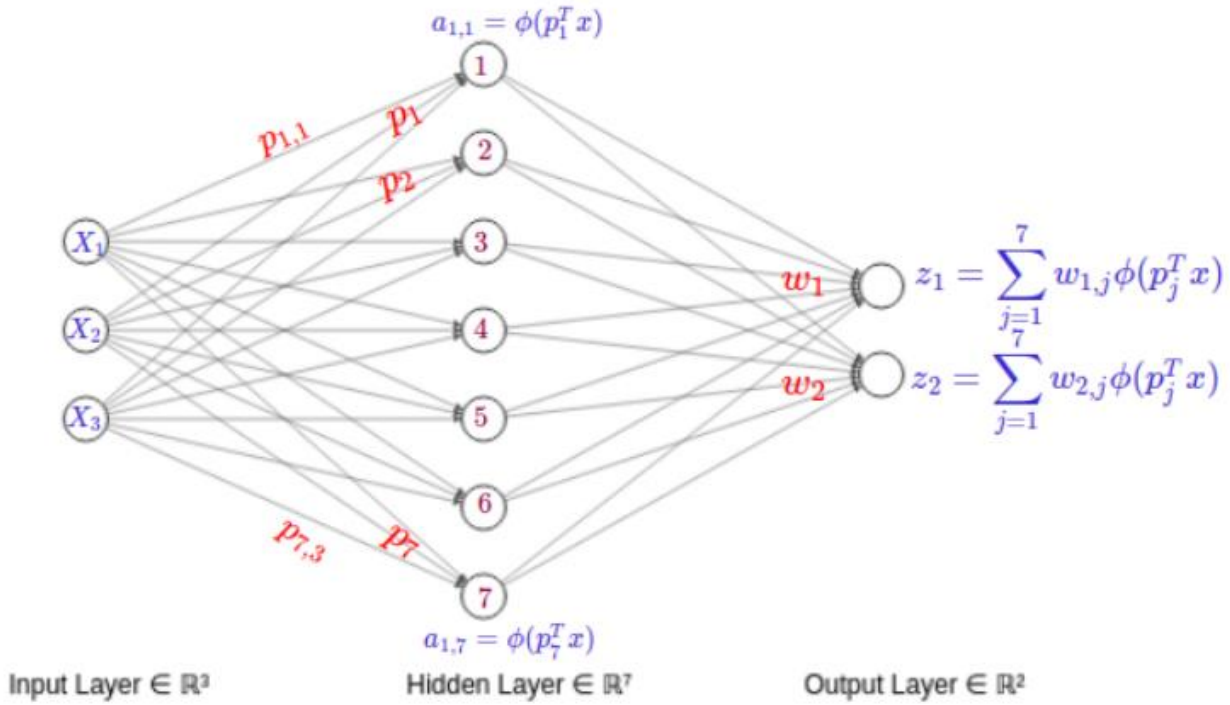


Coreset

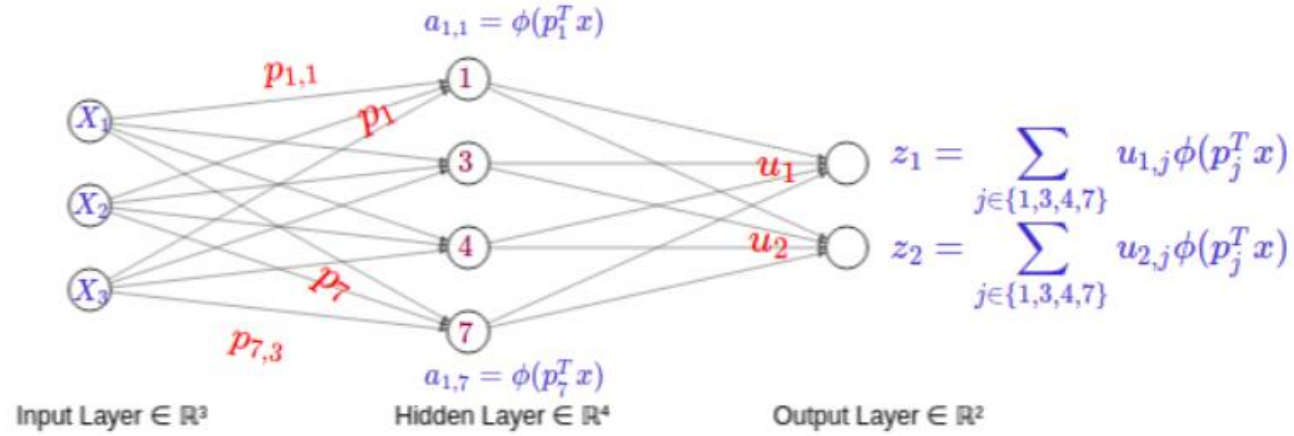
- Lots of different constructions with various tradeoffs
  - Size n' vs. accuracy
  - Size n' vs. computation time
  - Size n' vs. interpretability
  - Average-case vs. worst-case performance

New Coreset Constructions

- Coresets for data-independent neural pruning [MOBZF20]
- Coresets for projective clustering [TWZBF22]
- Coresets for regression on structured matrices [MMMWZF22]
- Online coresets for linear algebra [BDMMUWZ20]
- General coreset construction framework through sensitivity sampling [BFLSZ21]


$a_{1,1} = \phi(p_1^T x)$
$p_{1,1}$
p_1
p_2
$p_{7,3}$
p_7
X_1
X_2
X_3
1
2
3
4
5
6
7
w_1
w_2
$z_1 = \sum_{j=1}^{7} w_{1,j}\phi(p_j^T x)$
$z_2 = \sum_{j=1}^{7} w_{2,j}\phi(p_j^T x)$
$a_{1,7} = \phi(p_7^T x)$
Input Layer ∈ ℝ³
Hidden Layer ∈ ℝ⁷
Output Layer ∈ ℝ²

(a)


$a_{1,1} = \phi(p_1^T x)$
$p_{1,1}$
p_1
p_7
$p_{7,3}$
X_1
X_2
X_3
1
3
4
7
u_1
u_2
$z_1 = \sum_{j\in\{1,3,4,7\}} u_{1,j}\phi(p_j^T x)$
$z_2 = \sum_{j\in\{1,3,4,7\}} u_{2,j}\phi(p_j^T x)$
$a_{1,7} = \phi(p_7^T x)$
Input Layer ∈ ℝ³
Hidden Layer ∈ ℝ⁴
Output Layer ∈ ℝ²

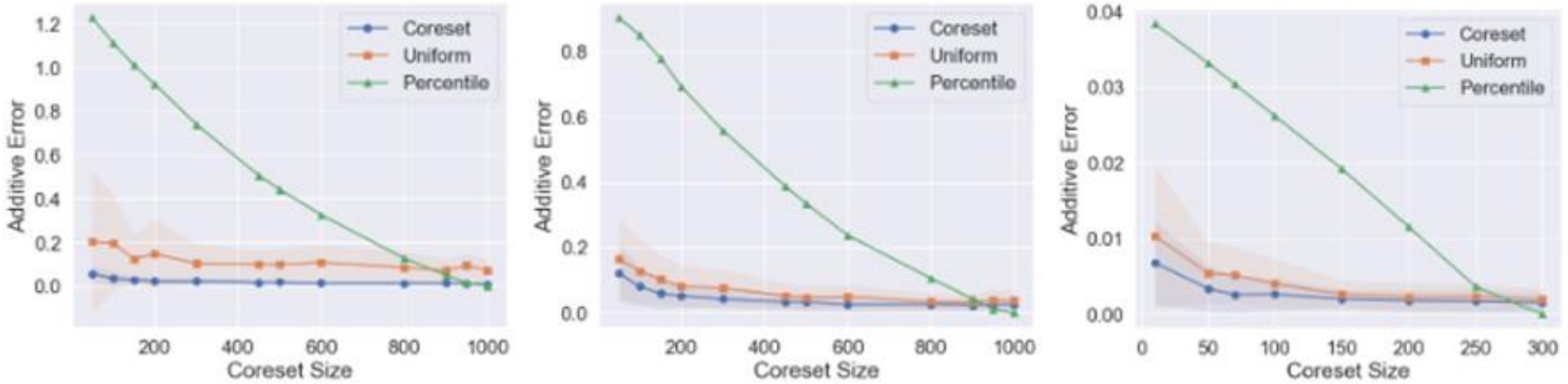
(b)

[MOBZF20]

Network	Error(%)	# Parameters	Compression Ratio
LeNet-300-100	2.16	267K	
LeNet-300-100 Pruned	**2.03**	**26K**	**90%**
VGG-16	8.95	1.4M	
VGG-16 Pruned	**8.16**	**350K**	**75%**

Table 2: Empirical evaluations of our coresets on existing architectures for MNIST and CIFAR-10. Note the improvement of accuracy in both cases!

[MOBZF20]

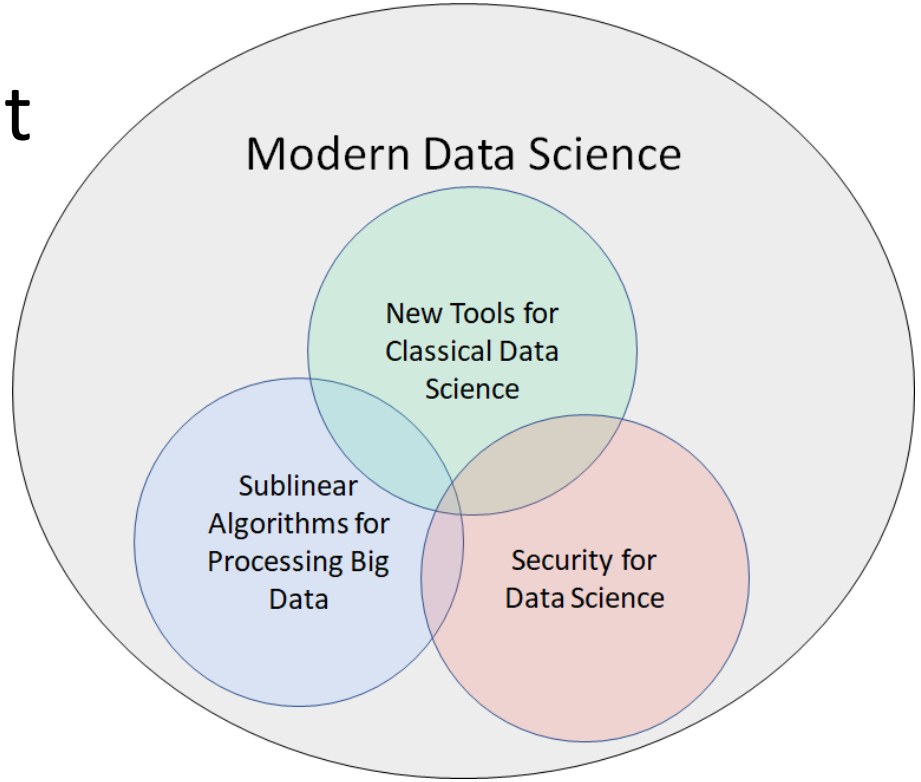


(a) (b) (c)

The weights of the points in (a) are drawn from the Gaussian distribution, in (b) from the Uniform distribution and in (c) we used the trained weights from LeNet-300-100.

Evolving Demands

- Sublinear-time or sublinear-space algorithms
- Incorporation of advice
- Security for data science
- Robustness to noise or adversarial input
- Ability to handle time-sensitive data

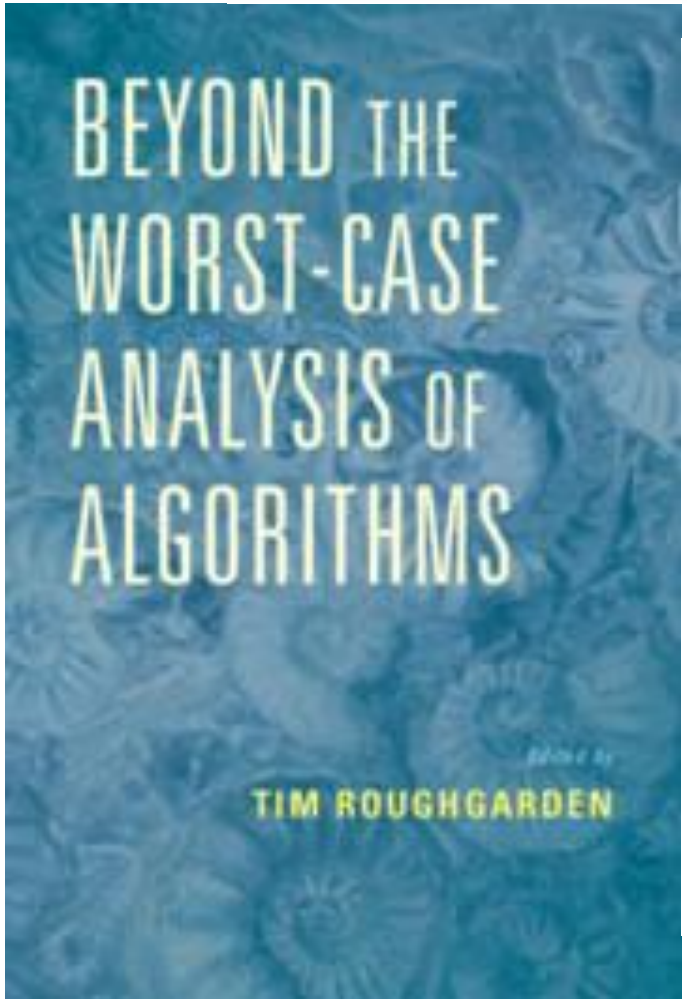



7:08
LTE
Mahabalipuram
73°
Partly Cloudy
H:86° L:72°
Sunny conditions expected around 9AM.
Now 8AM 9AM 10AM 11AM 12P
73° 76° 79° 82° 84° 85
10-DAY FORECAST
Today 72° 86°
Fri 72° 86°
Sat 71° 85°
7:08
LTE
Mahabalipuram
77°
Partly Cloudy
H:84° L:76°
Sunny conditions expected around 9AM.
Now 8AM 9AM 10AM 11AM 12F
77° 78° 80° 82° 83° 83
10-DAY FORECAST
Today 76° 84°
Fri 75° 84°
Sat 74° 85°
Sun 74° 85°
7:08
LTE
Mahabalipuram
74°
Partly Cloudy
H:85° L:73°
Sunny conditions expected around 9AM.
Now 8AM 9AM 10AM 11AM 12PM
74° 76° 79° 81° 83° 84°
10-DAY FORECAST
Today 73° 85°
Fri 73° 85°
Sat 72° 84°
Sun 72° 84°

Stellar™ 6.890 Learning-Augmented Algorithms

LOGIN

Course : » Course 6 : » Spring 2019 : » 6.890 : »Materials



29 Data-Driven Algorithm Design 626
Maria-Florina Balcan
29.1 Motivation and Context 626
29.2 Data-Driven Algorithm Design via Statistical Learning 628
29.3 Data-Driven Algorithm Design via Online Learning 639
29.4 Summary and Discussion 644

30 Algorithms with Predictions 646
Michael Mitzenmacher and Sergei Vassilvitskii
30.1 Introduction 646
30.2 Counting Sketches 649
30.3 Learned Bloom Filters 650
30.4 Caching with Predictions 652
30.5 Scheduling with Predictions 655
30.6 Notes 660

https://algorithms-with-predictions.github.io

Algorithms with Predictions

PAPER LIST FURTHER MATERIAL ABOUT

'07 '09 '10 '17 '18 '19 '20 '21 '22

Newest first

122 papers

Graph Searching with Predictions Banerjee, Cohen-Addad, Gupta, Li arXiv '22 exploration online search

Scheduling with Predictions Cho, Henderson, Shmoys arXiv '22 online scheduling

On the Power of Learning-Augmented BSTs Chen, Chen arXiv '22 data structure search

Algorithms with Prediction Portfolios Dinitz, Im, Lavastida, Moseley, Vassilvitskii arXiv '22 load balancing matching multiple predictions online scheduling

Private Algorithms with Private Predictions Amin, Dick, Khodak, Vassilvitskii arXiv '22 differential privacy

Paging with Succinct Predictions Antoniadis, Boyar, Eliáš, Favrholdt, Hoeksma, Larsen, Polak, Simon arXiv '22 caching/paging online

Proportionally Fair Online Allocation of Public Goods with Predictions Banerjee, Gkatzelis, Hossain, Jin, Micha, Shah arXiv '22 allocation online

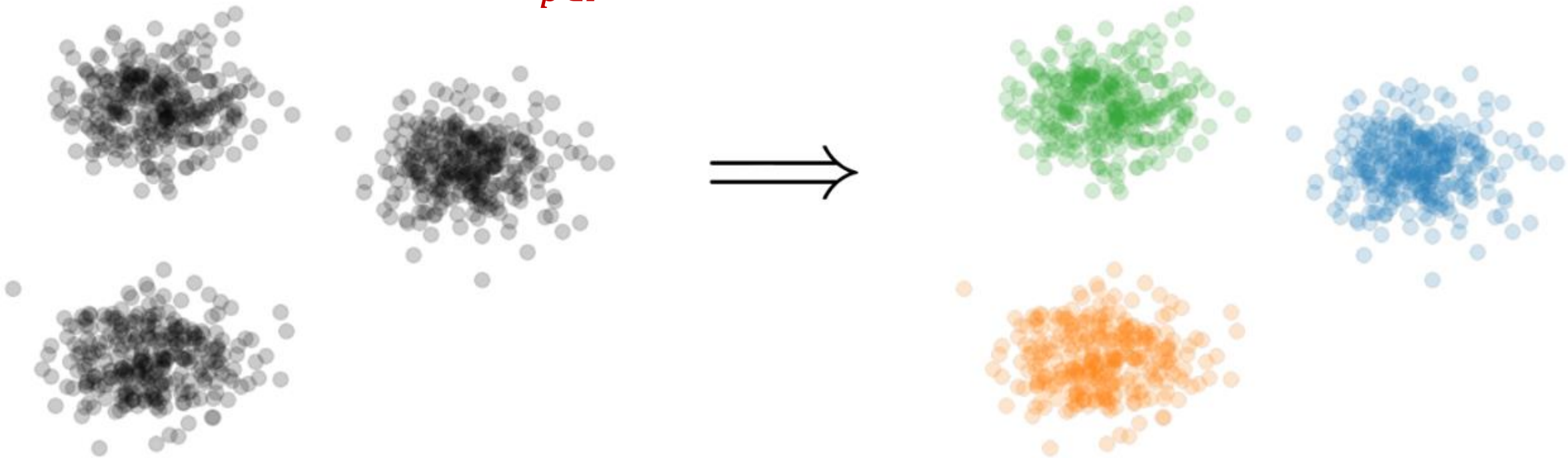
Canadian Traveller Problem with Predictions Bampis, Escoffier, Xefteris arXiv '22 WAOA '22 online routing

Learning-Augmented Algorithms for Online Linear and Semidefinite Programming Grigorescu, Lin, Silwal, Song, Zhou arXiv '22 covering problems online SDP

Learning-Augmented Clustering

- Goal: Given dataset P in d dimensions, output a set C of k centers to minimize

$$\sum_{p \in P} \min_{c \in C} \|p - c\|_2^2$$



Learning-Augmented Clustering

- Goal: Given dataset P in d dimensions, output a set C of k centers to minimize

$$\sum_{p \in P} \min_{c \in C} \|p - c\|_2^2$$

- NP-hard to even approximate within a factor of 1.07 [CC20, LSW17]
- Beyond worst case: Clustering on similar inputs or inputs with auxiliary information
- ML can guide the clustering!
- Our main message: We can avoid worst case with advice!

Predictor

- Suppose Π outputs noisy labels according to a $(1+\alpha)$ approximate clustering C and error rate $\lambda \leq \alpha$

What is the label of x_1?

x_1 belongs to cluster 3

What is the label of x_2?

x_2 belongs to cluster 7

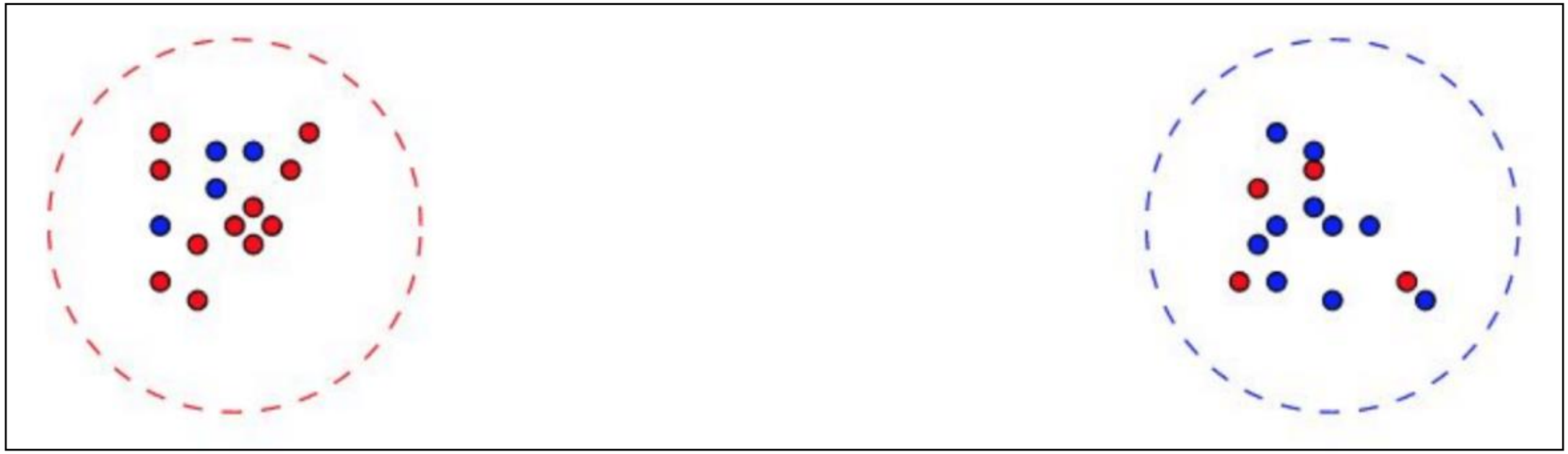
Theoretical Guarantee

- Suppose Π outputs noisy labels according to a $(1+\alpha)$ approximate clustering C and error rate $\lambda \leq \alpha$
- Main result [EFSWZ22]: Algorithm that outputs a $(1+O(\alpha))$ approximate k-means clustering in nearly linear time

- "Predictions can overcome complexity hardness barriers!"

Algorithmic Intuition

- Not enough to blindly follow predictions!



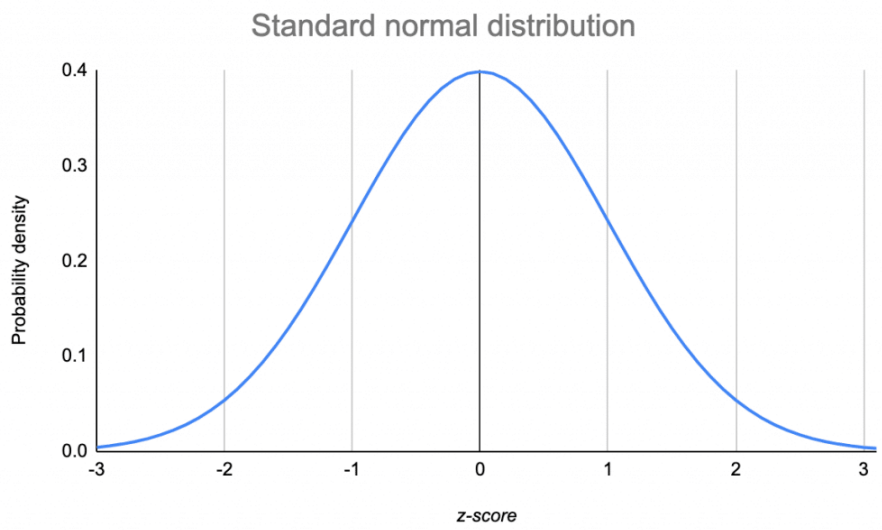
- Optimal cost ≈ 0
- Predictor with arbitrary small error has large cost!

Algorithmic Intuition

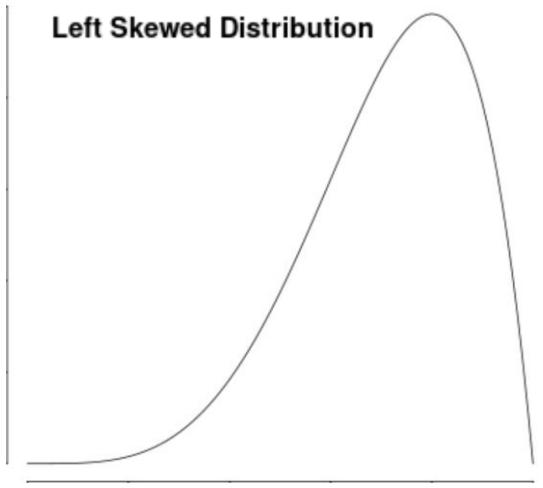
- Not enough to blindly follow predictions!

- Our approach: Use ideas from robust mean estimation

$(1-\varepsilon)P$



εQ



Algorithmic Intuition

- Not enough to blindly follow predictions!

- Our approach: Use ideas from robust mean estimation

Algorithmic Intuition

- Not enough to blindly follow predictions!

- Our approach: Use ideas from robust mean estimation

Algorithmic Intuition

- Not enough to blindly follow predictions!
- Our approach: Use ideas from robust mean estimation
- We exploit the fact that cluster centers are means of points!

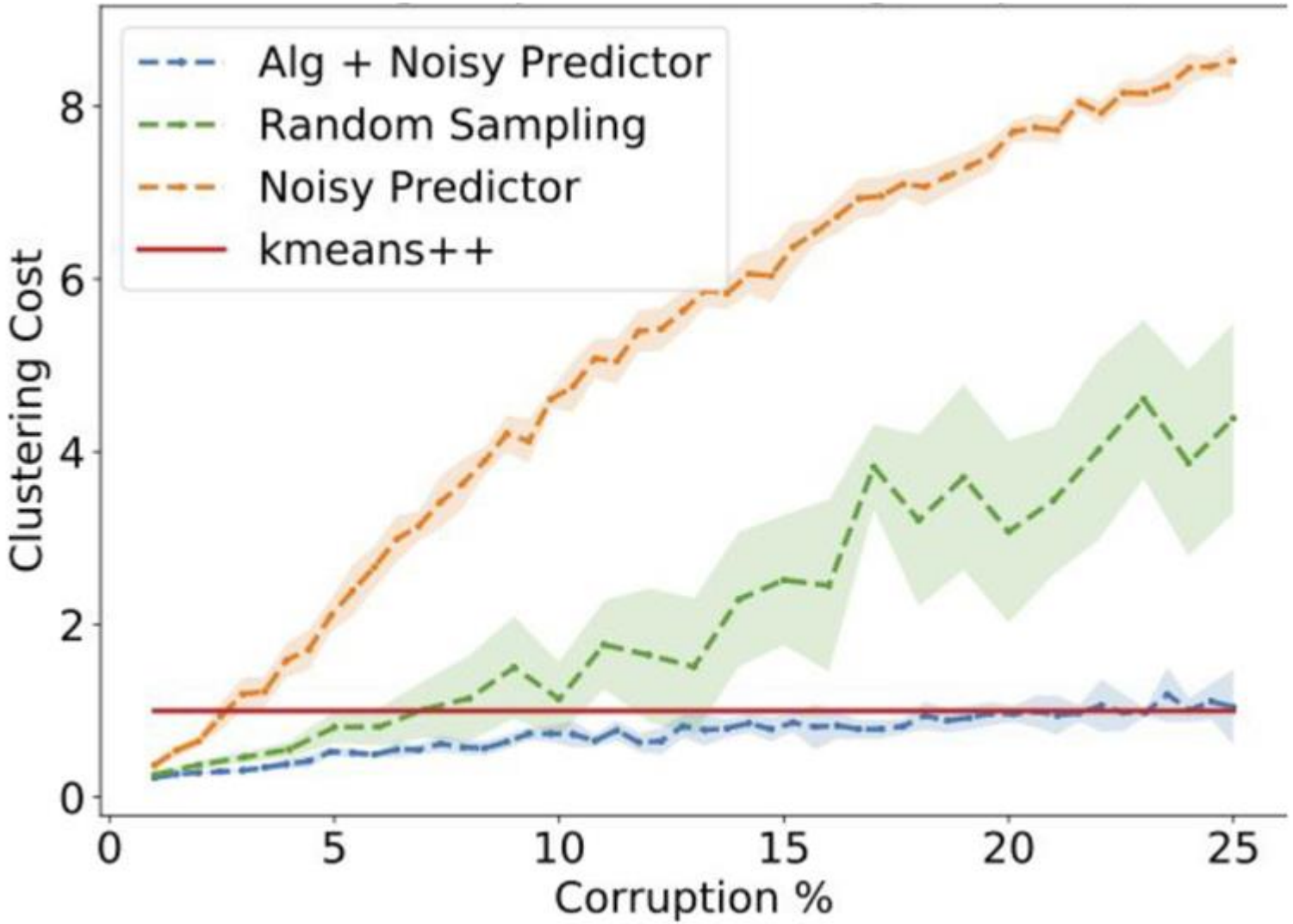
Algorithmic Intuition

- Not enough to blindly follow predictions!
- Our approach: Use ideas from robust mean estimation
- We exploit the fact that cluster centers are means of points!
- Introduces a new set of tools for k-means clustering

Experimental Results

- Case Study: Spectral clustering on graphs varying over time
- Dataset: Internet router graph varying over the course of a year
- Methodology: Compare to standard benchmarks while using various natural predictors, i.e., noisily perturb true labels and compare to baselines as function of error

Dataset: Internet router graph varying over the course of a year, $k = 10$



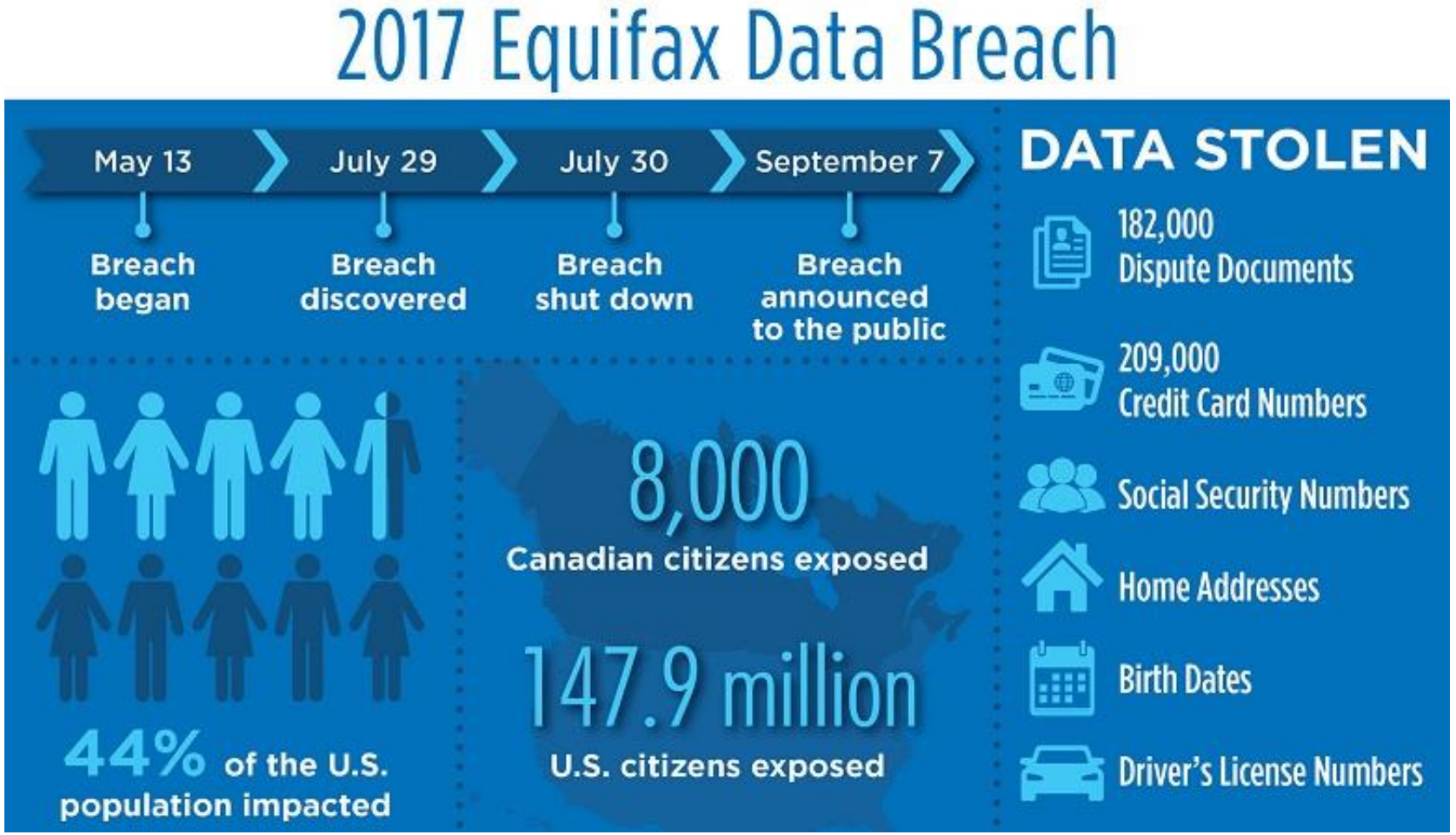
Conclusion: Our algorithm (using predictor) outperforms benchmarks such as k-means ++ for low error while staying competitive with high corruptions

My Recent Work on New Data Science Tools

- Learning-Augmented k-means Clustering (ICLR 2022)
- Fast Regression for Structured Inputs (ICLR 2022)
- New Coresets for Projective Clustering and Applications (AISTATS 2022)
- Dimensionality Reduction for Wasserstein Barycenter (NeurIPS 2021)
- Learning a Latent Simplex in Input Sparsity Time (ICLR 2021)
- Near Optimal Linear Algebra in the Online and Sliding Window Models (FOCS 2020)
- Data-Independent Neural Pruning via Coresets (ICLR 2020)
- Adversarially Robust Submodular Maximization under Knapsack Constraints (KDD 2019)

Questions?

Security for Data Science

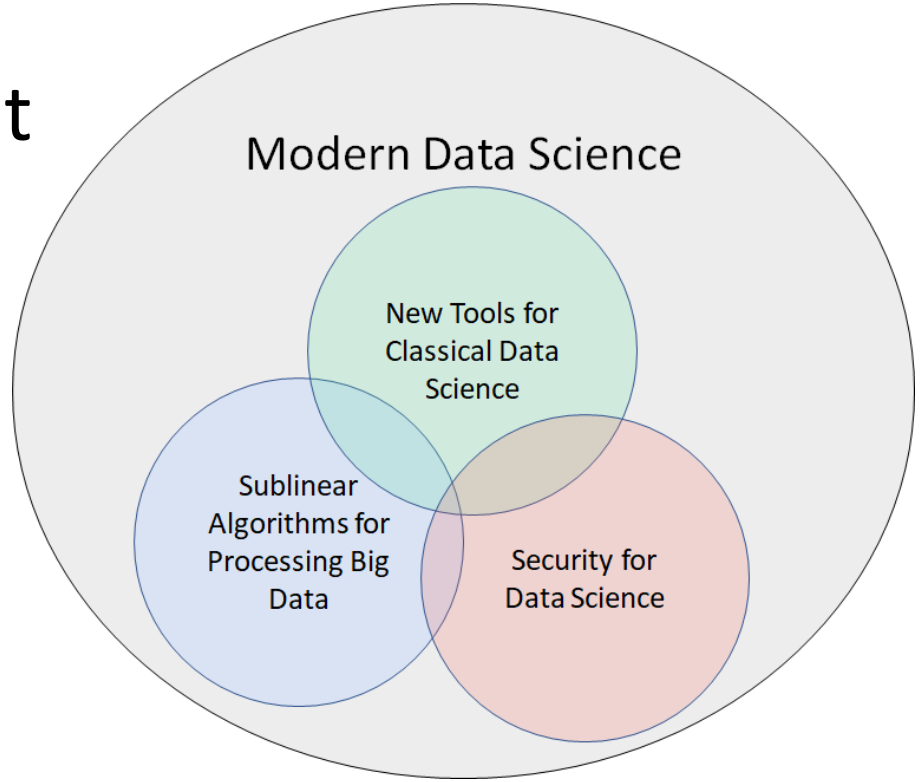


"Equifax agreed to a $700 million settlement over the privacy breach, but $425 million of that was set aside to repay consumers as a restitution fund."

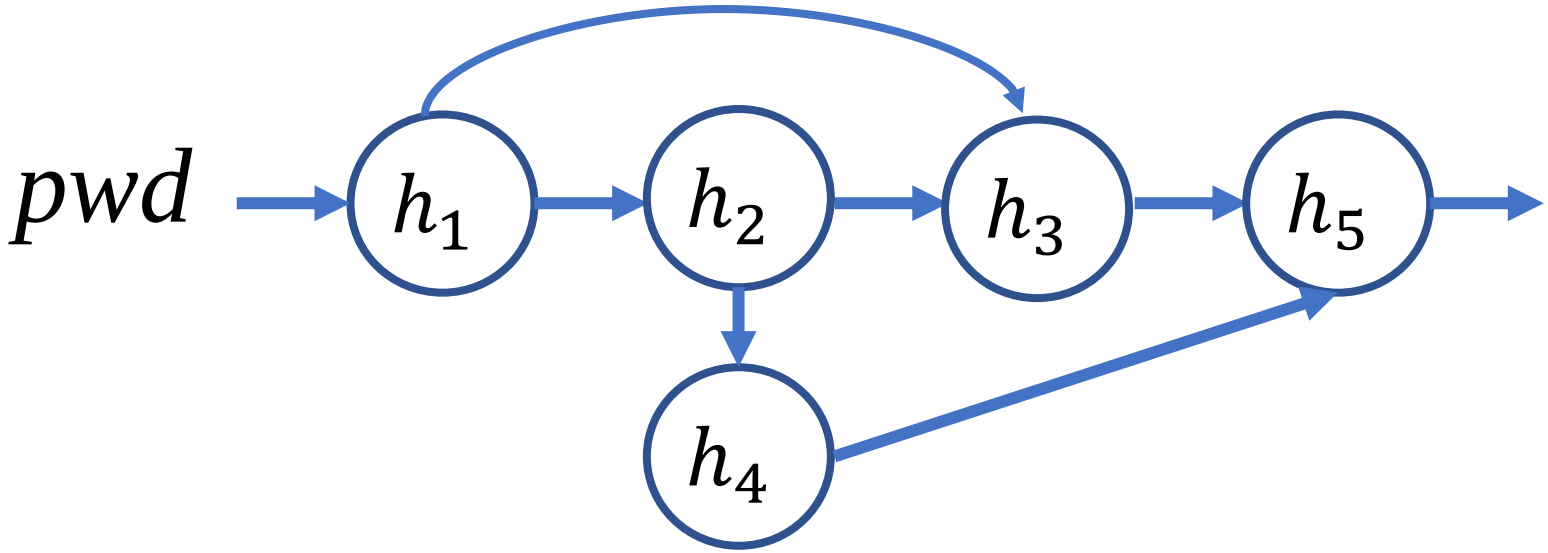

YAHOO!
UBER
Gmail
by Google
Adobe
BLIZZARD
ENTERTAINMENT
eHarmony
AMERICA
Online
U.S. DEPARTMENT OF HOMELAND SECURITY
LastPass ****
ebay
Dropbox
LinkedIn
PlayStation

Evolving Demands

- Sublinear-time or sublinear-space algorithms
- Incorporation of advice
- Security for data science
- Robustness to noise or adversarial input
- Ability to handle time-sensitive data

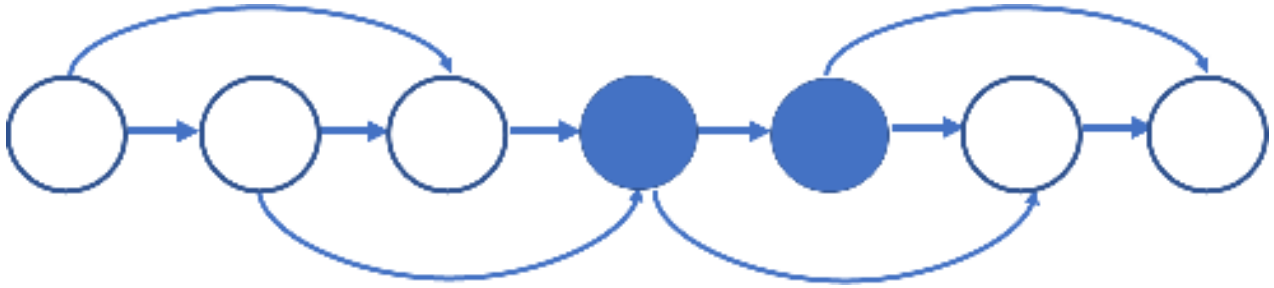


Graph Theory for Memory Hard Functions



Graph Theory for Memory Hard Functions

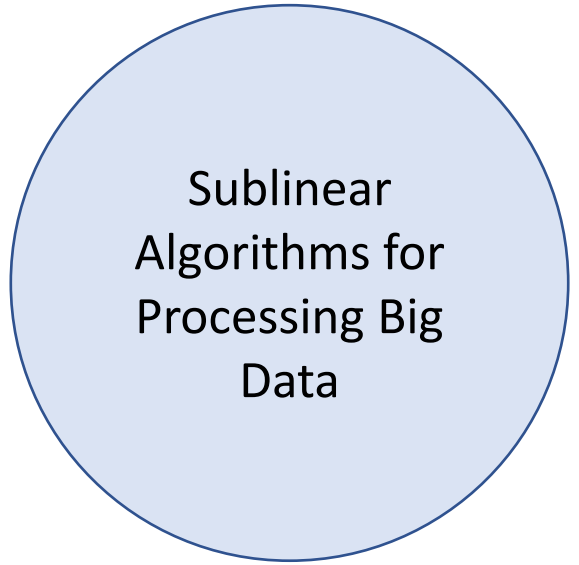
- Proved the security benefits of memory-hard functions using an economic marginal cost-revenue analysis [BHZ18]
- Showed the hardness of computing [BZ18] and approximating the computational cost [BLZ20] of graph pebbling
- Applied graph theory to cryptanalyze the computational complexity of memory-hard functions [BZ18, BRZ18]



My Recent Work on Security for Data Science

- On the Security of Proofs of Sequential Work in a Post-Quantum World (ITC 2021)
- Approximating Cumulative Pebbling Cost is Unique Games Hard (ITCS 2020)
- Computationally Data-Independent Memory Hard Functions (ITCS 2020)
- Data-Independent Memory Hard Functions: New Attacks and Stronger Constructions (CRYPTO 2019)
- Bandwidth-Hard Functions: Reductions and Lower Bounds (CCS 2018)
- On the Economics of Offline Password Cracking (Security and Privacy 2018)
- On the Depth-Robustness and Cumulative Pebbling Cost of Argon2i (TCC 2017)

Sublinear Algorithms for Processing Big Data



Model #1: Streaming Model

- Input: Elements of an underlying data set S, which arrives sequentially
- Output: Evaluation (or approximation) of a given function
- Goal: Use space *sublinear* in the size m of the input S

1 0 1 1 1 0 0 1

Frequency Vector

- Given a set S of m elements from $[n]$, let f_i be the frequency of element i. (How often it appears)

$$1\ 1\ 2\ 1\ 2\ 1\ 1\ 2\ 3 \rightarrow [5, 3, 1, 0] \coloneqq f$$

Heavy-Hitters (Frequent Items)

- Given a set S of m elements from $[n]$, let f_i be the frequency of element i. (How often it appears)
- Let L_2 be the norm of the frequency vector:

$$L_2 = \sqrt{f_1^2 + f_2^2 + \cdots + f_n^2}$$

- Goal: Given a set S of m elements from $[n]$ and a threshold ε, output the elements i such that $f_i > \varepsilon\, L_2$...and no elements j such that $f_j < \frac{\varepsilon}{16}\, L_2$
- Motivation: DDoS prevention, iceberg queries

Frequency Moments (L_p Norm)

- Given a set S of m elements from $[n]$, let f_i be the frequency of element i. (How often it appears)
- Let F_p be the frequency moment of the vector:

$$F_p = f_1^p + f_2^p + \cdots + f_n^p$$

- Goal: Given a set S of m elements from $[n]$ and an accuracy parameter ε, output a $(1+\varepsilon)$-approximation to F_p
- Motivation: Entropy estimation, linear regression

Distinct Elements (F_0 Estimation)

- Given a set S of m elements from $[n]$, let f_i be the frequency of element i. (How often it appears)
- Let F_0 be the frequency moment of the vector:

$$F_0 = |\{i : f_i \neq 0\}|$$

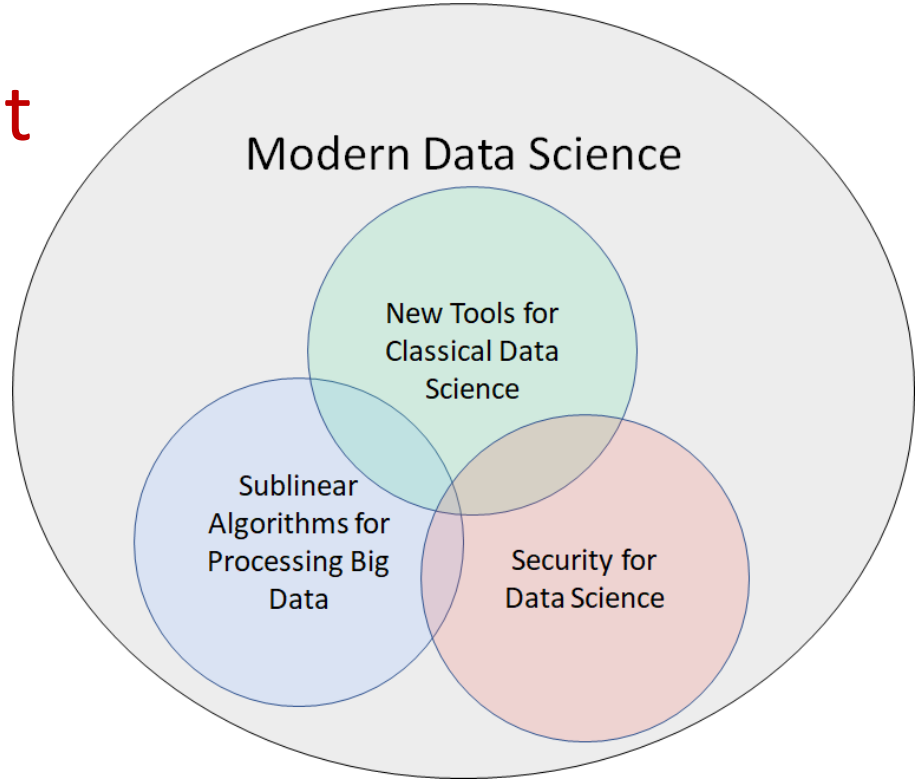
- Goal: Given a set S of m elements from $[n]$ and an accuracy parameter ε, output a $(1+\varepsilon)$-approximation to F_0
- Motivation: Traffic monitoring

$(1+\varepsilon)$-Approximation Streaming Algorithms

- $\Theta\left(\frac{1}{\varepsilon^2}\right)$ space-accuracy tradeoff for distinct elements, i.e., F_0 [KNW10, Blasiok20]
- $\Theta\left(\frac{1}{\varepsilon^2}\right)$ space-accuracy tradeoff for F_p with $p \in (0,2]$ [BDN17]
- $\Theta\left(\frac{1}{\varepsilon^2}\right)$ space-accuracy tradeoff for F_p with $p > 2$ [Ganguly11, GW18]
- $\Theta\left(\frac{1}{\varepsilon^2}\right)$ space-accuracy tradeoff for L_2-heavy hitters [BCINWW17]

Evolving Demands

- Sublinear-time or sublinear-space algorithms
- Incorporation of advice
- Security for data science
- Robustness to noise or adversarial input
- Ability to handle time-sensitive data



skier_adv.png

Dog	91%
Dog Like Mammal	87%
Snow	84%
Arctic	70%
Winter	67%
Ice	65%
Fun	60%
Freezing	60%
Glacial Landform	50%

Human readers will easily identify the image as showing two men on skis. Google's Cloud Vision service reported being 91 percent certain it saw a dog. Other stunts have shown how to make stop signs invisible, or audio that sounds benign to humans but is transcribed by software as "OK Google browse to evil dot com."

Model #2: Adversarially Robust Streaming

- Input: Elements of an underlying data set S, which arrives sequentially and *adversarially*
- Output: Evaluation (or approximation) of a given function
- Goal: Use space *sublinear* in the size m of the input S

1

1

Model #2: Adversarially Robust Streaming

- Input: Elements of an underlying data set S, which arrives sequentially and *adversarially*
- Output: Evaluation (or approximation) of a given function
- Goal: Use space *sublinear* in the size m of the input S

10

1

Model #2: Adversarially Robust Streaming

- Input: Elements of an underlying data set S, which arrives sequentially and *adversarially*
- Output: Evaluation (or approximation) of a given function
- Goal: Use space *sublinear* in the size m of the input S

101

2

Model #2: Adversarially Robust Streaming

- Input: Elements of an underlying data set S, which arrives sequentially and *adversarially*
- Output: Evaluation (or approximation) of a given function
- Goal: Use space *sublinear* in the size m of the input S

1010

3

Model #2: Adversarially Robust Streaming

- Input: Elements of an underlying data set S, which arrives sequentially and *adversarially*
- Output: Evaluation (or approximation) of a given function
- Goal: Use space *sublinear* in the size m of the input S

1010

3

$(1+\varepsilon)$-Robust Algorithms [WZ21]

- $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space-accuracy tradeoff for distinct elements, i.e., F_0
- $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space-accuracy tradeoff for F_p with $p \in (0,2]$
- $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space-accuracy tradeoff for F_p with integer $p > 2$
- $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space-accuracy tradeoff for L_2-heavy hitters

"No losses* are necessary!"

15-859 Algorithms for Big Data — Fall 2022

Problem Set 3

Due: Tuesday, November 8, 11:59pm

Problem 2: F_2-Difference Estimator in a Stream (25 points)

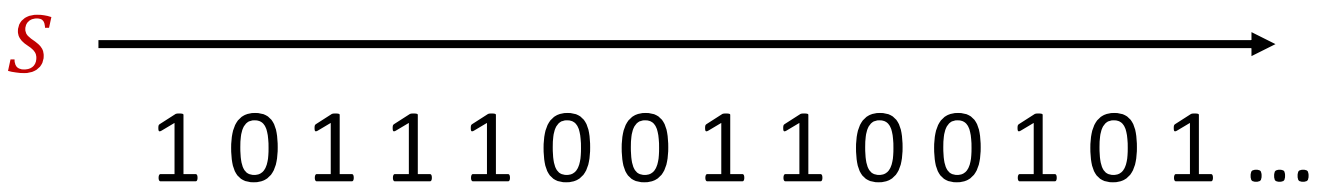
In this problem we consider insertion-only streams, meaning that we just see positive updates to an underlying vector $x \in \{0, 1, 2, \ldots, M\}^n$ for some $M = \text{poly}(n)$. That is, no negative changes to coordinates of x are allowed. We will consider estimating $\|x\|_2^2$ up to a $(1+\epsilon)$-multiplicative factor.

Often x is *slowly-changing*, meaning that at some point in the stream $x = u$ and then at some later point in the stream $x = u + v$ for some $v \in \{0, 1, 2, \ldots, M\}^n$, and we have $\|u+v\|_2^2 - \|u\|_2^2 \leq \gamma\|u\|_2^2$ and $\|v\|_2^2 \leq \gamma\|u\|_2^2$ for some $0 < \gamma < 1$. You would like to estimate $\|u+v\|_2^2$ using your previous estimate $\|u\|_2^2$ and using very little space. Show how to estimate $\|u+v\|_2^2$ up to a $(1 \pm \epsilon)$-multiplicative factor with probability at least $9/10$, given a $(1 \pm \epsilon/2)$-approximation to $\|u\|_2^2$ and a sketch which uses space $O(\gamma\epsilon^{-2} \log n)$ bits. You can assume that you initialize $x = 0$ and after processing some number of updates you will have $x = u$ at which point you are given a $1 \pm \epsilon/2$ approximation to $\|u\|_2^2$. Then you start processing the rest of the updates and finally end up setting $x = u + v$ and at the end of the stream you want to output a $1 \pm \epsilon$ approximation to $\|u+v\|_2^2$.

Note that if $\gamma = \Theta(1)$, then there is no improvement over just estimating $\|u+v\|_2^2$ directly using the sketch from class. However, if for example, $\gamma = \Theta(\epsilon)$, then the memory is only $O(\epsilon^{-1} \log n)$ bits, which is a significant savings.

Robust Algorithms

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

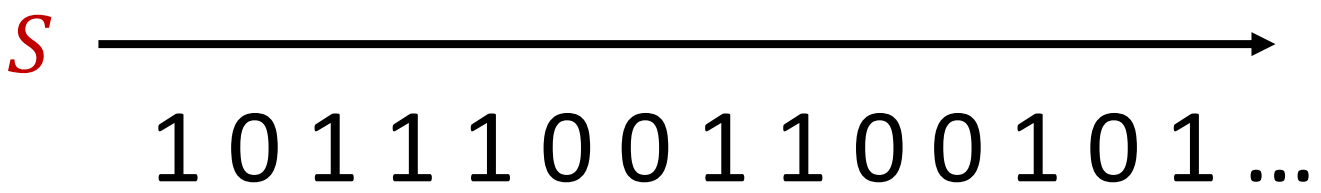


Robust Algorithms

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

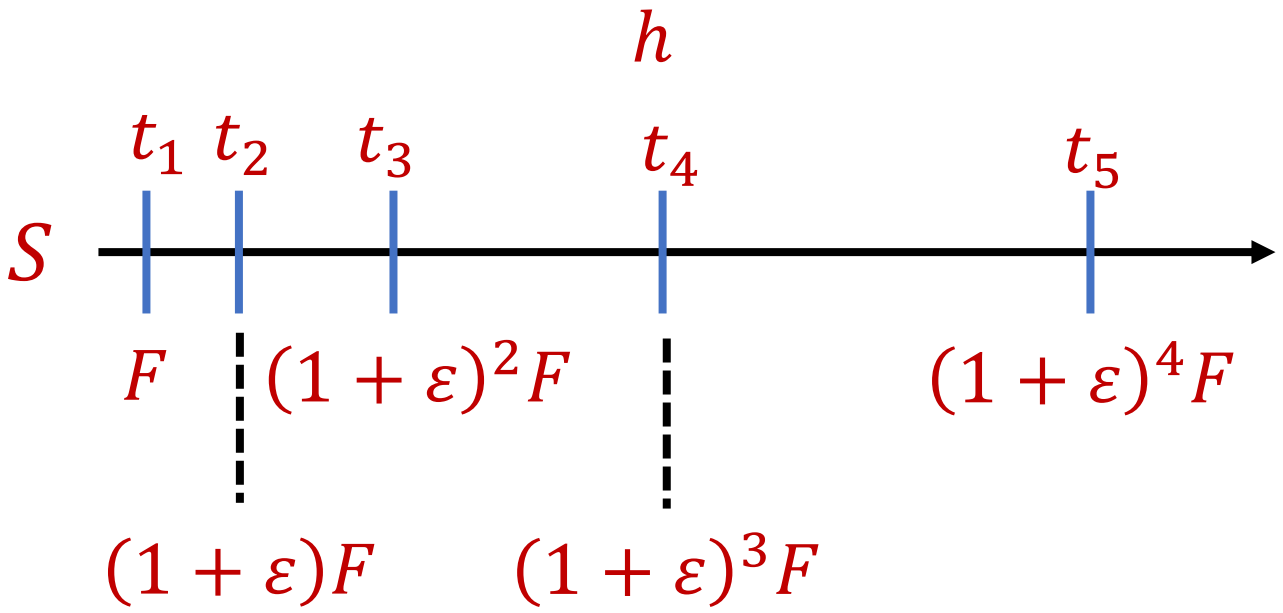
$(1+\varepsilon)$ approximation to h

- Monotonic function h



Robust Algorithms

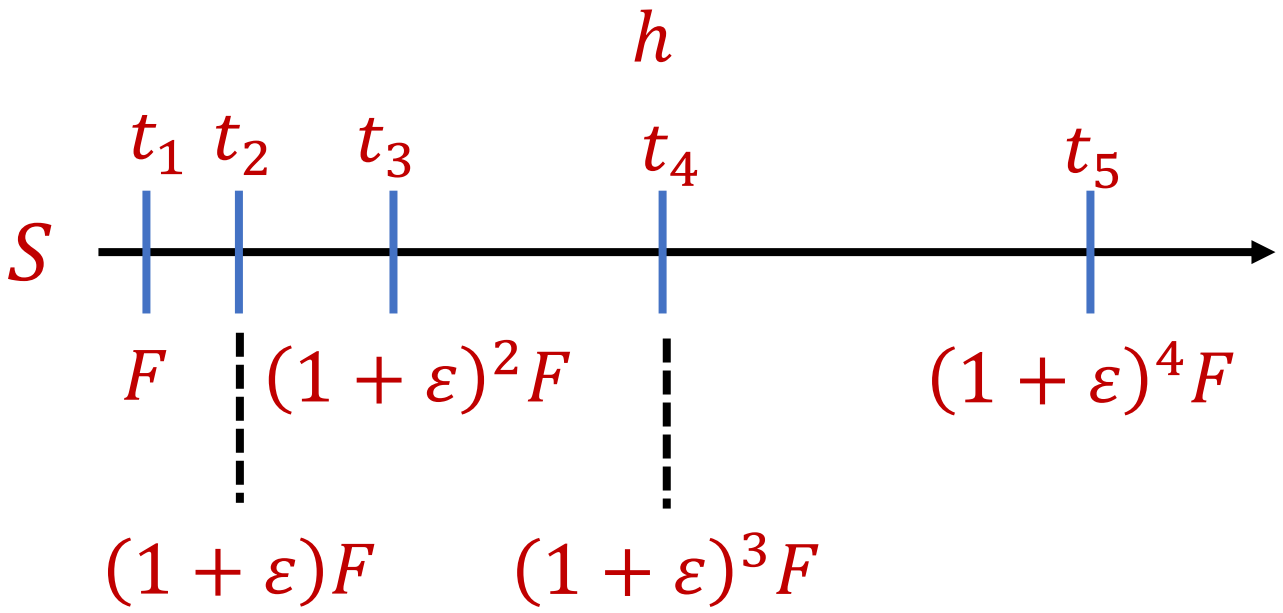
Goal: $\tilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space



- Monotonic function h
- Consider times when h increases by $(1+\varepsilon)$

Robust Algorithms

Goal: $\tilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

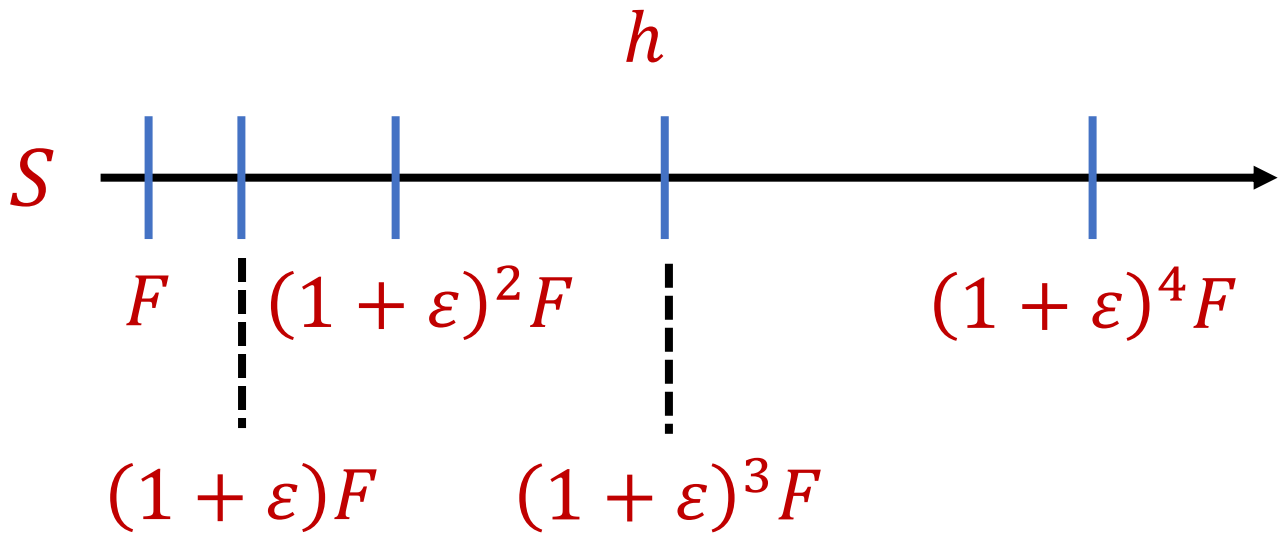


- Monotonic function h
- Consider times when h increases by $(1+\varepsilon)$
- Correctness at these times suffice!

Robust Algorithms

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

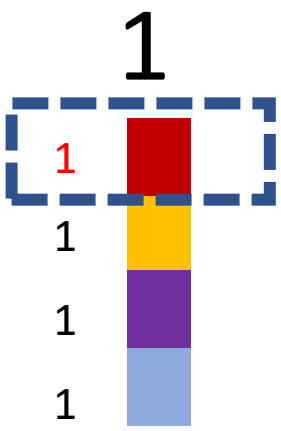
- Initialize many instances of the streaming algorithm
- Use a new instance of the algorithm each time the function increases by $(1+\varepsilon)$



Robust Algorithms

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Initialize many instances of the streaming algorithm
- Use a new instance of the algorithm each time the function increases by $(1+\varepsilon)$

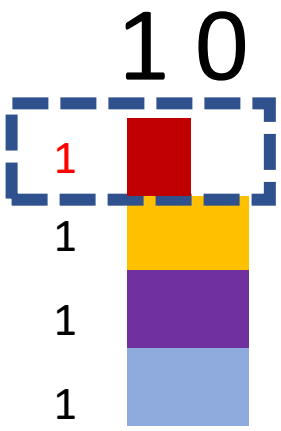


- Example: Number of ones in the stream (2-approximation)

Robust Algorithms

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Initialize many instances of the streaming algorithm
- Use a new instance of the algorithm each time the function increases by $(1+\varepsilon)$

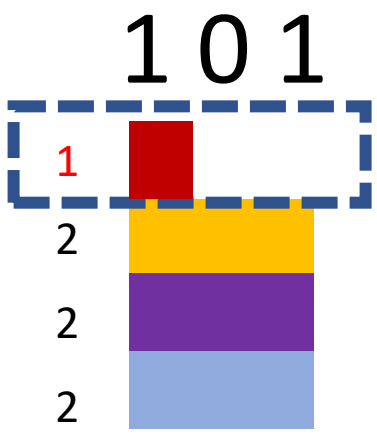


- Example: Number of ones in the stream (2-approximation)

Robust Algorithms

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Initialize many instances of the streaming algorithm
- Use a new instance of the algorithm each time the function increases by $(1+\varepsilon)$



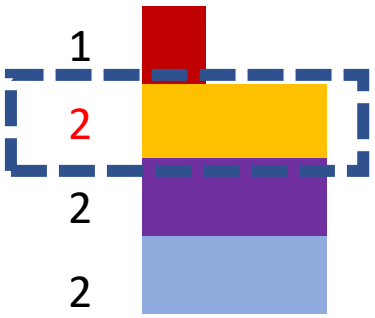
- Example: Number of ones in the stream (2-approximation)

Robust Algorithms

Goal: $\tilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Initialize many instances of the streaming algorithm
- Use a new instance of the algorithm each time the function increases by $(1+\varepsilon)$

1 0 1

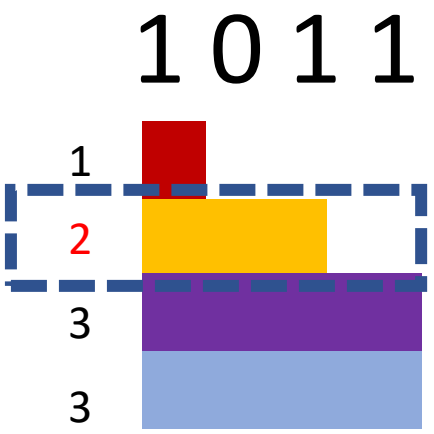


- Example: Number of ones in the stream (2-approximation)

Robust Algorithms

Goal: $\tilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Initialize many instances of the streaming algorithm
- Use a new instance of the algorithm each time the function increases by $(1+\varepsilon)$

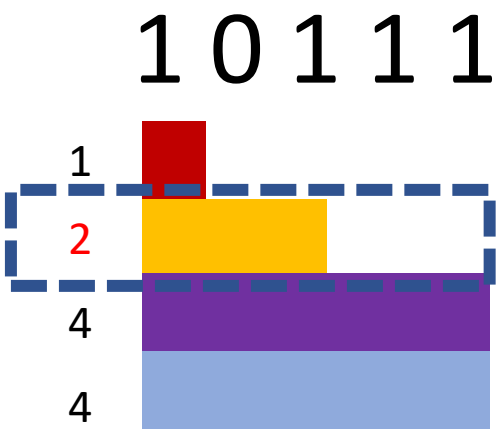


- Example: Number of ones in the stream (2-approximation)

Robust Algorithms

Goal: $\tilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Initialize many instances of the streaming algorithm
- Use a new instance of the algorithm each time the function increases by $(1+\varepsilon)$



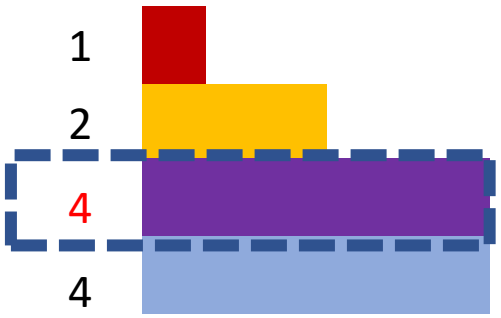
- Example: Number of ones in the stream (2-approximation)

Robust Algorithms

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Initialize many instances of the streaming algorithm
- Use a new instance of the algorithm each time the function increases by $(1+\varepsilon)$

1 0 1 1 1



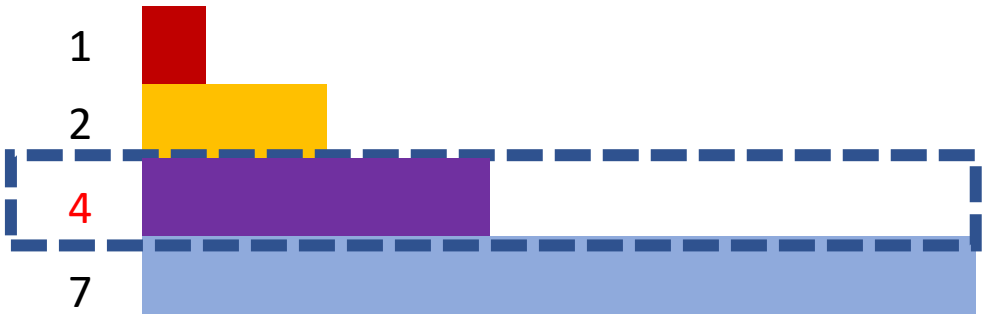
- Example: Number of ones in the stream (2-approximation)

Robust Algorithms

Goal: $\tilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Initialize many instances of the streaming algorithm
- Use a new instance of the algorithm each time the function increases by $(1+\varepsilon)$

1 0 1 1 1 0 0 1 1 0 1



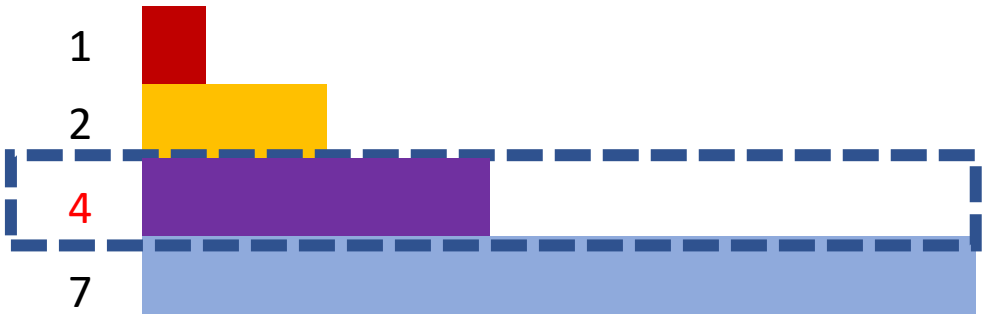
- Example: Number of ones in the stream (2-approximation)
- Number of ones stream is at least 4 and at most 8
- 4 is a good approximation

Intuition

Goal: $\tilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Approach uses $\frac{1}{\varepsilon^2}$ space each time function increases by $(1+\varepsilon)$ and function increases $\frac{1}{\varepsilon}$ times
- Total space: $\frac{1}{\varepsilon^2} \times \frac{1}{\varepsilon} = \tilde{O}\left(\frac{1}{\varepsilon^3}\right)$

1 0 1 1 1 0 0 1 1 0 1



Intuition

Goal: $\tilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Approach uses $\frac{1}{\varepsilon^2}$ space each time function increases by $(1+\varepsilon)$ and function increases $\frac{1}{\varepsilon}$ times for total space $\tilde{O}\left(\frac{1}{\varepsilon^3}\right)$
- Do we really need to pay $\frac{1}{\varepsilon^2}$ space each time function increases by $(1+\varepsilon)$?

Intuition

Goal: $\tilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Approach uses $\frac{1}{\varepsilon^2}$ space each time function increases by $(1+\varepsilon)$ and function increases $\frac{1}{\varepsilon}$ times for total space $\tilde{O}\left(\frac{1}{\varepsilon^3}\right)$
- Do we really need to pay $\frac{1}{\varepsilon^2}$ space each time function increases by $(1+\varepsilon)$?

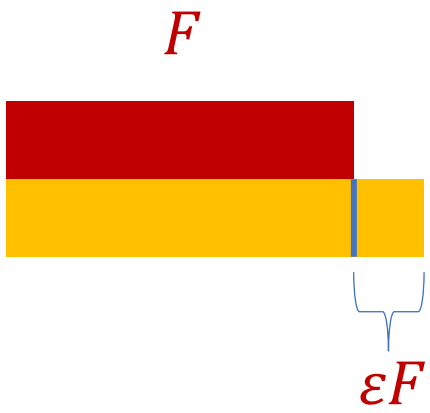
Intuition

Goal: $\tilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Approach uses $\frac{1}{\varepsilon^2}$ space each time function increases by $(1+\varepsilon)$ and function increases $\frac{1}{\varepsilon}$ times for total space $\tilde{O}\left(\frac{1}{\varepsilon^3}\right)$
- Do we really need to pay $\frac{1}{\varepsilon^2}$ space each time function increases by $(1+\varepsilon)$?

Intuition

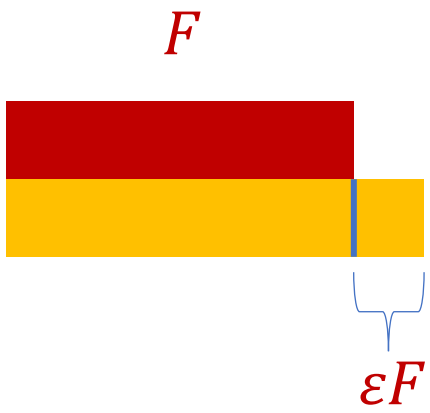
Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space



Intuition

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

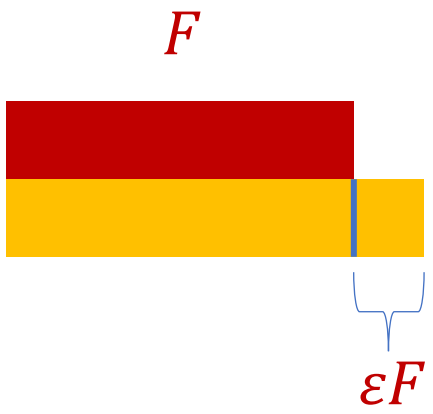
- Only need constant factor approximation to εF



Intuition

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

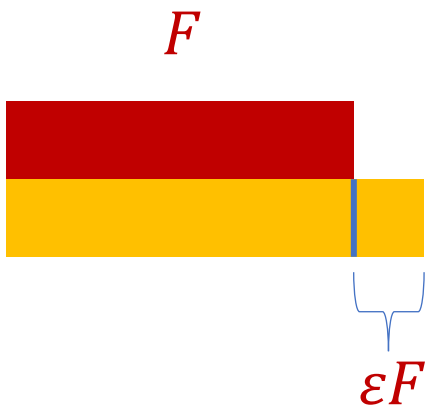
- Only need constant factor approximation to εF
- Only need to pay $\frac{1}{\varepsilon}$ space each time function increases by $(1+\varepsilon)$



Intuition

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

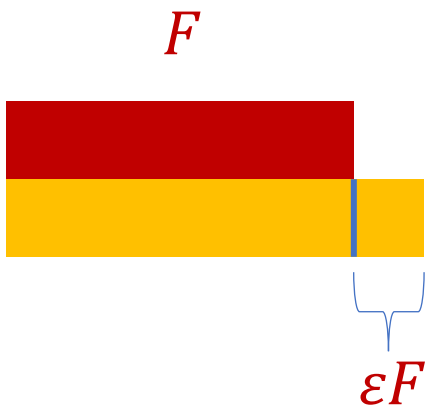
- Only need constant factor approximation to εF
- Only need to pay $\frac{1}{\varepsilon}$ space each time function increases by $(1+\varepsilon)$
- Function increases by $(1+\varepsilon)$ a total of $\frac{1}{\varepsilon}$ times



Intuition

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Only need constant factor approximation to εF
- Only need to pay $\frac{1}{\varepsilon}$ space each time function increases by $(1+\varepsilon)$
- Function increases by $(1+\varepsilon)$ a total of $\frac{1}{\varepsilon}$ times
- Total space: $\frac{1}{\varepsilon} \times \frac{1}{\varepsilon} = \widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$



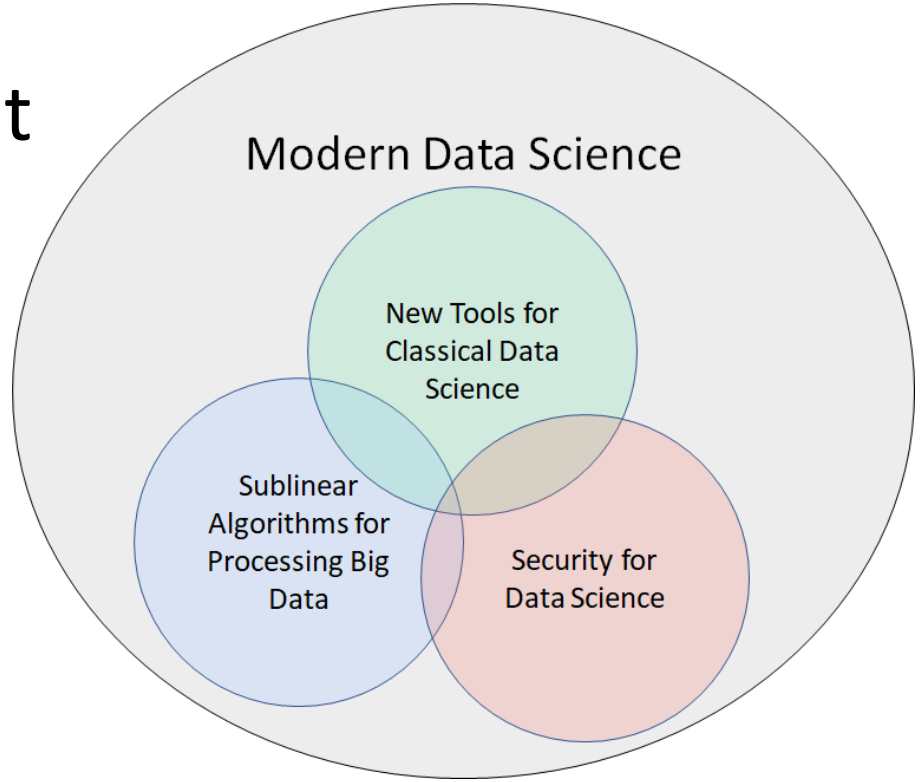
Additional Work

- [BHMSSZ22] showed that “sampling-based” algorithms achieve adversarial robustness “for-free”
- [ABJSSWZ22] showed impossibility results for many natural problems for “white-box” adversaries
- [ABJSSWZ22] gave a sublinear-space algorithm for estimating F_0 when the white-box adversary has polynomial computation time

- Open Questions: 1) Is anything non-trivial possible for “black-box” adversaries that are allowed to delete elements? 2) Can lattice-based crypto be used for other algorithms robust to white-box adversaries?

Evolving Demands

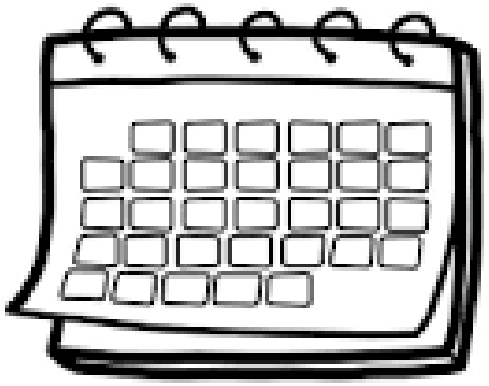
- Sublinear-time or sublinear-space algorithms
- Incorporation of advice
- Security for data science
- Robustness to noise or adversarial input
- Ability to handle time-sensitive data



“Facebook data policy retains user search histories for 6 months, the Apple differential privacy overview states that collected data is retained for 3 months, the Google data retention policy states that browser information may be stored for up to 9 months”

What is the data retention policy?

Data retention policies **concern what data should be stored or archived, where that should happen, and for exactly how long**. Once the retention time period for a particular data set expires, it can be deleted or moved as historical data to secondary or tertiary storage, depending on the requirements.



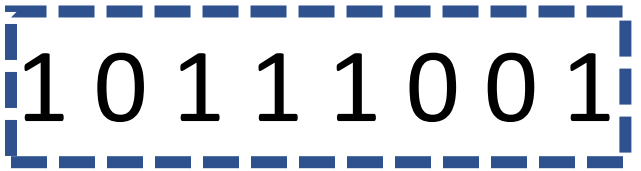
Do I need a data retention policy?

There are numerous benefits to establishing a solid data retention policy. Some of the more compelling benefits include: Automated compliance. With an established policy, organizations can ensure they comply with regulatory requirements mandating the retention of various types of data.

Model #3: Sliding Window Model

- Input: Elements of an underlying data set S, which arrives sequentially
- Output: Evaluation (or approximation) of a given function
- Goal: Use space *sublinear* in the size m of the input S

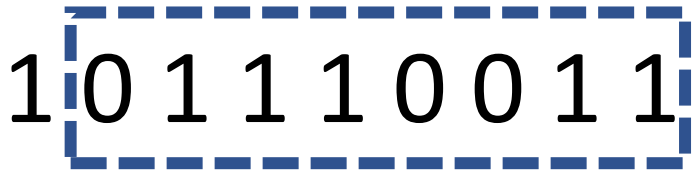
- Sliding Window: "Only the m most recent updates form the underlying data set S"



Model #3: Sliding Window Model

- Input: Elements of an underlying data set S, which arrives sequentially
- Output: Evaluation (or approximation) of a given function
- Goal: Use space *sublinear* in the size m of the input S

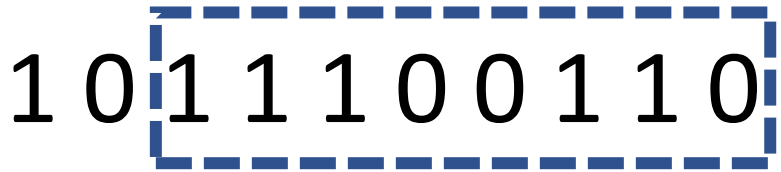
- Sliding Window: "Only the m most recent updates form the underlying data set S"



Model #3: Sliding Window Model

- Input: Elements of an underlying data set S, which arrives sequentially
- Output: Evaluation (or approximation) of a given function
- Goal: Use space *sublinear* in the size m of the input S

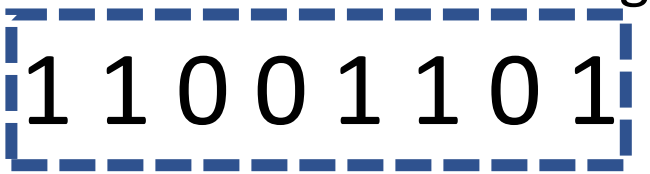
- Sliding Window: "Only the m most recent updates form the underlying data set S"



Model #3: Sliding Window Model

- Input: Elements of an underlying data set S, which arrives sequentially
- Output: Evaluation (or approximation) of a given function
- Goal: Use space *sublinear* in the size m of the input S

- Sliding Window: "Only the m most recent updates form the underlying data set S"
  - Emphasizes recent interactions, appropriate for time sensitive settings



$(1+\varepsilon)$-Approximation Sliding Window Algorithms [WZ21]

- Space $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ algorithm for F_p with $p \in (0,2]$ [WZ21]

Problem	[BO07] Space	Our Result
L_p Estimation, $p \in (0,1)$	$\tilde{\mathcal{O}}\left(\frac{\log^3 n}{\varepsilon^3}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^3 n}{\varepsilon^2}\right)$
L_p Estimation, $p \in (1,2]$	$\tilde{\mathcal{O}}\left(\frac{\log^3 n}{\varepsilon^{2+p}}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^3 n}{\varepsilon^2}\right)$
L_p Estimation, integer $p > 2$	$\tilde{\mathcal{O}}\left(\frac{n^{1-2/p}}{\varepsilon^{2+p}}\right)$	$\tilde{\mathcal{O}}\left(\frac{n^{1-2/p}}{\varepsilon^2}\right)$
Entropy Estimation	$\tilde{\mathcal{O}}\left(\frac{\log^5 n}{\varepsilon^4}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^5 n}{\varepsilon^2}\right)$

"No losses* are necessary!"

$(1+\varepsilon)$-Approximation Sliding Window Algorithms [BGLWZ18]

- $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space-accuracy tradeoff for F_0, i.e., distinct elements
- $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space-accuracy tradeoff for L_2-heavy hitters

- We give nearly tight new upper and lower bounds for these problems!

RandNLA in the Sliding Window Model

- First algorithms for problems in numerical linear algebra in the sliding window model [BDMMUWZ20]

- $O(d^2)$ space randomized sliding window algorithm for spectral sparsification (space optimal, up to lower order terms)
- $O(dk)$ space randomized sliding window algorithm for low-rank approximation (space optimal, up to lower order terms)
- $O(d^3)$ space randomized sliding window algorithm for ℓ_1 subspace embedding

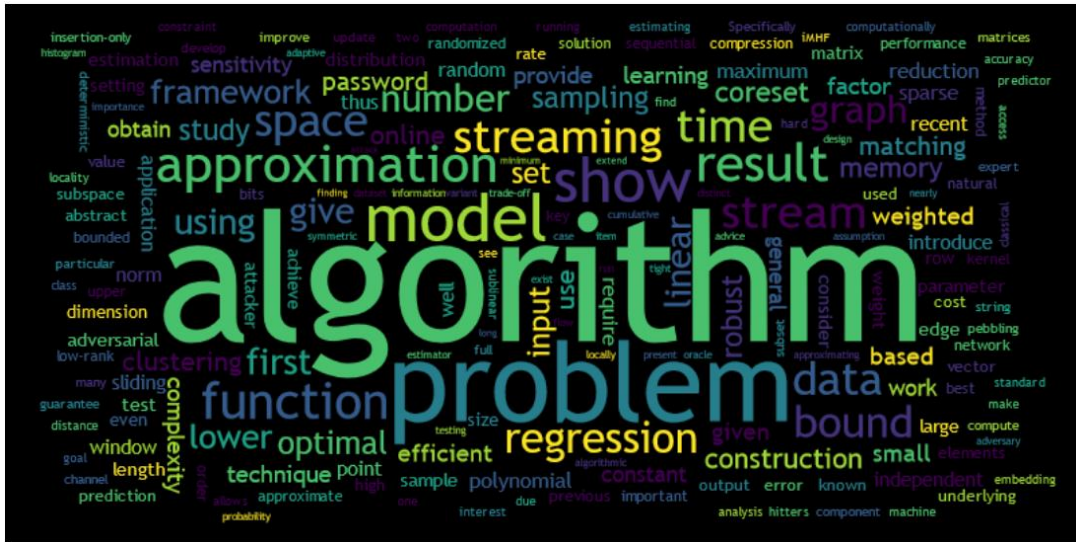
My Recent Work on Sublinear Algorithms

- Memory Bounds for the Experts Problem (STOC 2022)
- The White-Box Adversarial Data Stream Model (PODS 2022)
- Truly Perfect Samplers for Data Streams and Sliding Windows (PODS 2022)
- Noisy Boolean Hidden Matching with Applications (ITCS 2022)
- Adversarial Robustness of Streaming Algorithms through Importance Sampling (NeurIPS 2021)
- Tight Bounds for Adversarially Robust Streams and Sliding Windows via Difference Estimators (FOCS 2021)
- Separations for Estimating Large Frequency Moments on Data Streams (ICALP 2021)
- Non-Adaptive Adaptive Sampling on Turnstile Streams (STOC 2020)

Future Research Directions

- Adversarial robustness for data science/ML
- Limits of learning-augmented algorithms, beyond NP-hardness, better utility bounds for DP
- Data science with fairness considerations

- Evolving demands of the future!



Future Agenda

- Department research: interdisciplinary collaborations
- Department activities: social hour, reading group, lunch seminar
- Regional activities: Boston Theory Day, Boston Data Science Day

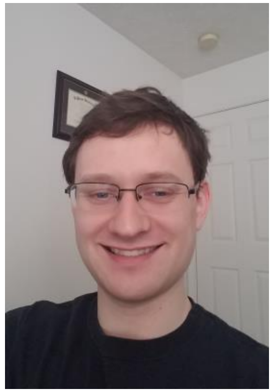
- Teaching: Randomized algorithms / mathematical toolkit (hashing, concentration inequalities, linear programming, convex optimization), "modern" data science (dimensionality reduction, clustering, sublinear algorithms)
- Funding!: NSF, NIH, DARPA, Apple, Google, Meta, Microsoft

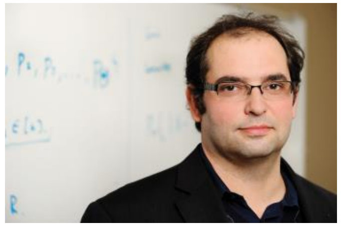
Thank You!



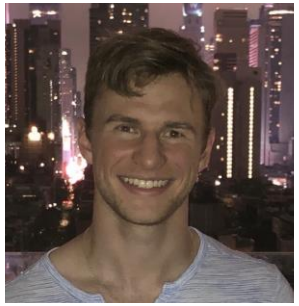






Thank You!



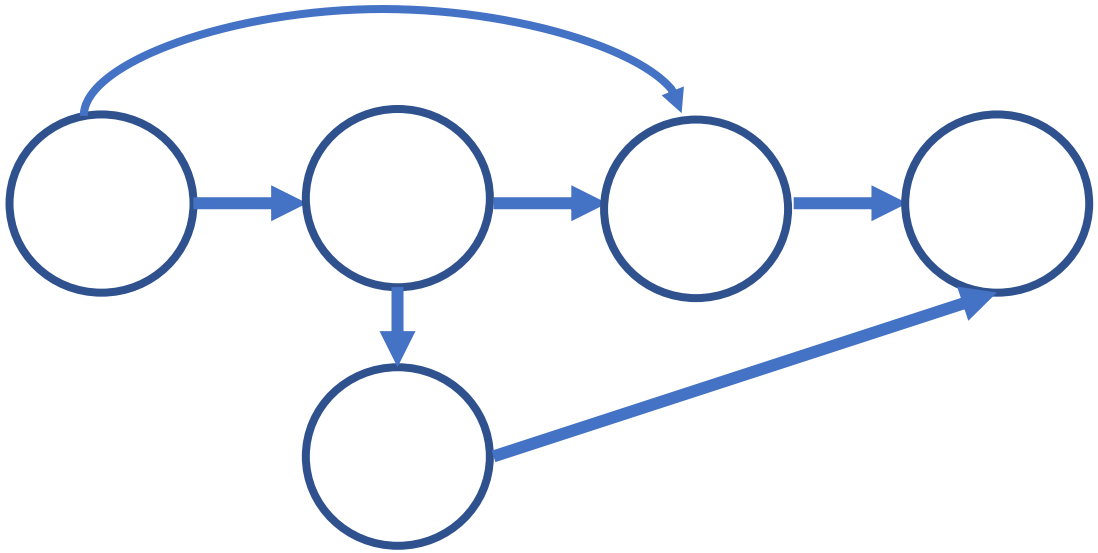






iMHFs

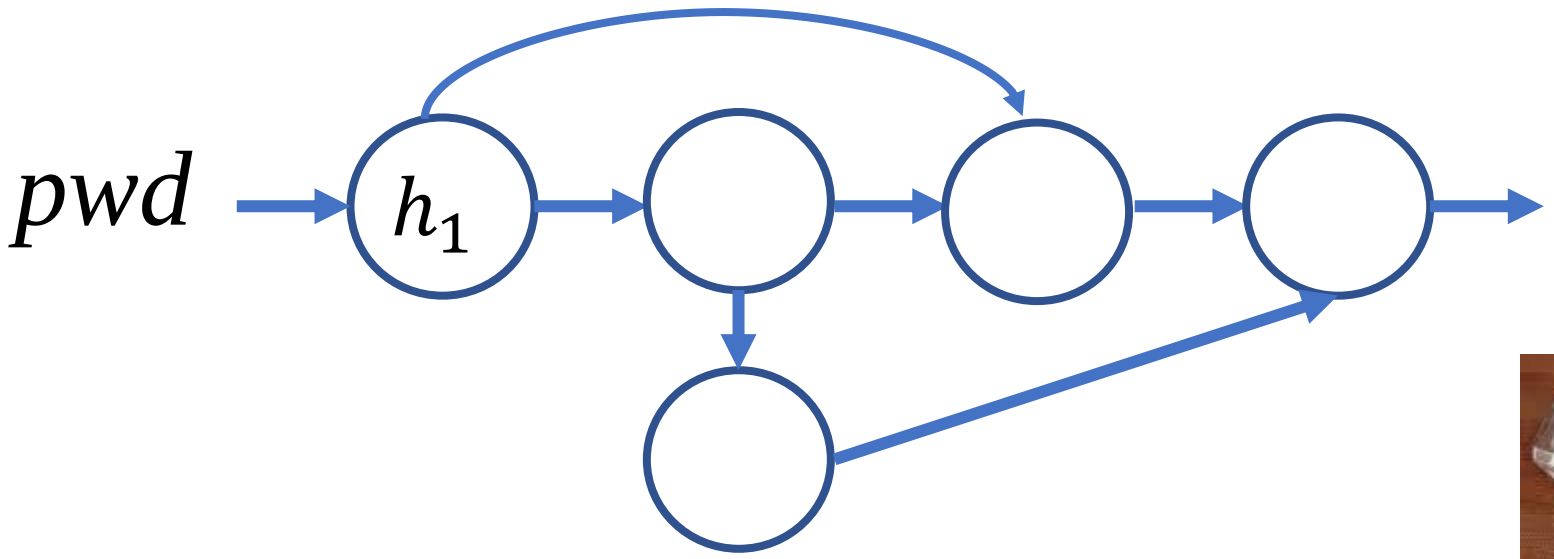
$f_{G,H}$



Hash function: H

iMHFs

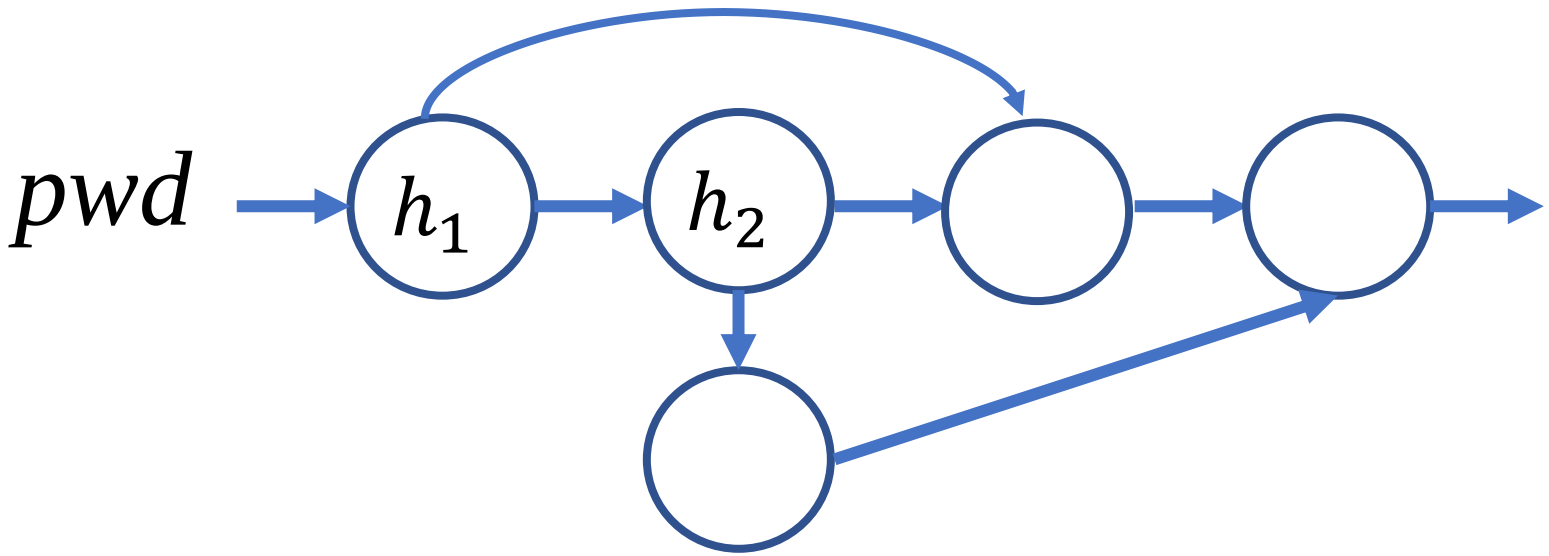
$f_{G,H}$



$$h_1 = H(pwd, salt)$$

iMHFs

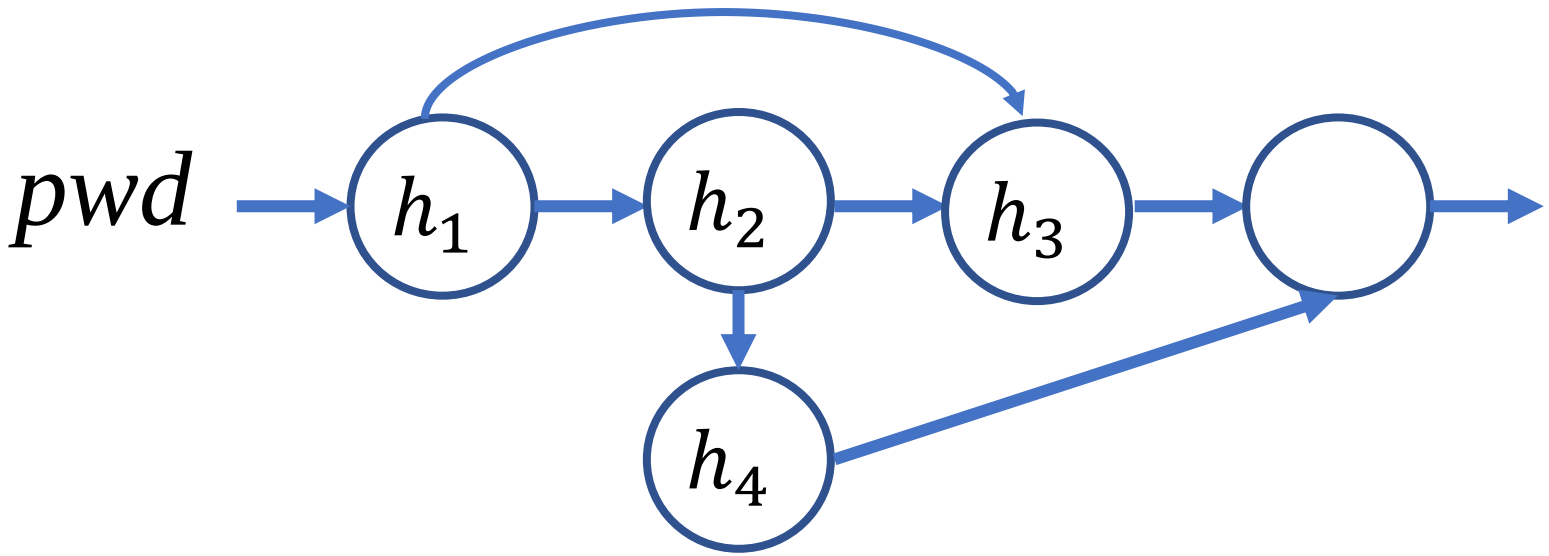
$f_{G,H}$



$$h_2 = H(h_1)$$

iMHFs

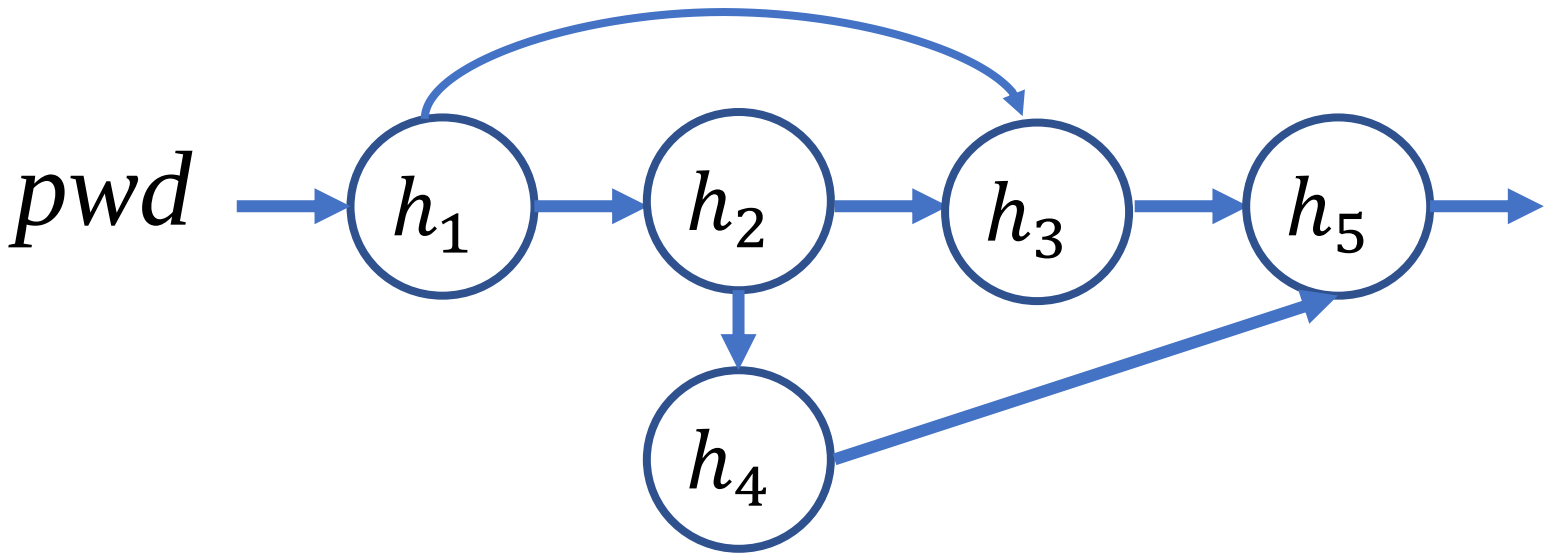
$f_{G,H}$



$h_3 = H(h_1, h_2),$ $h_4 = H(h_2)$

iMHFs

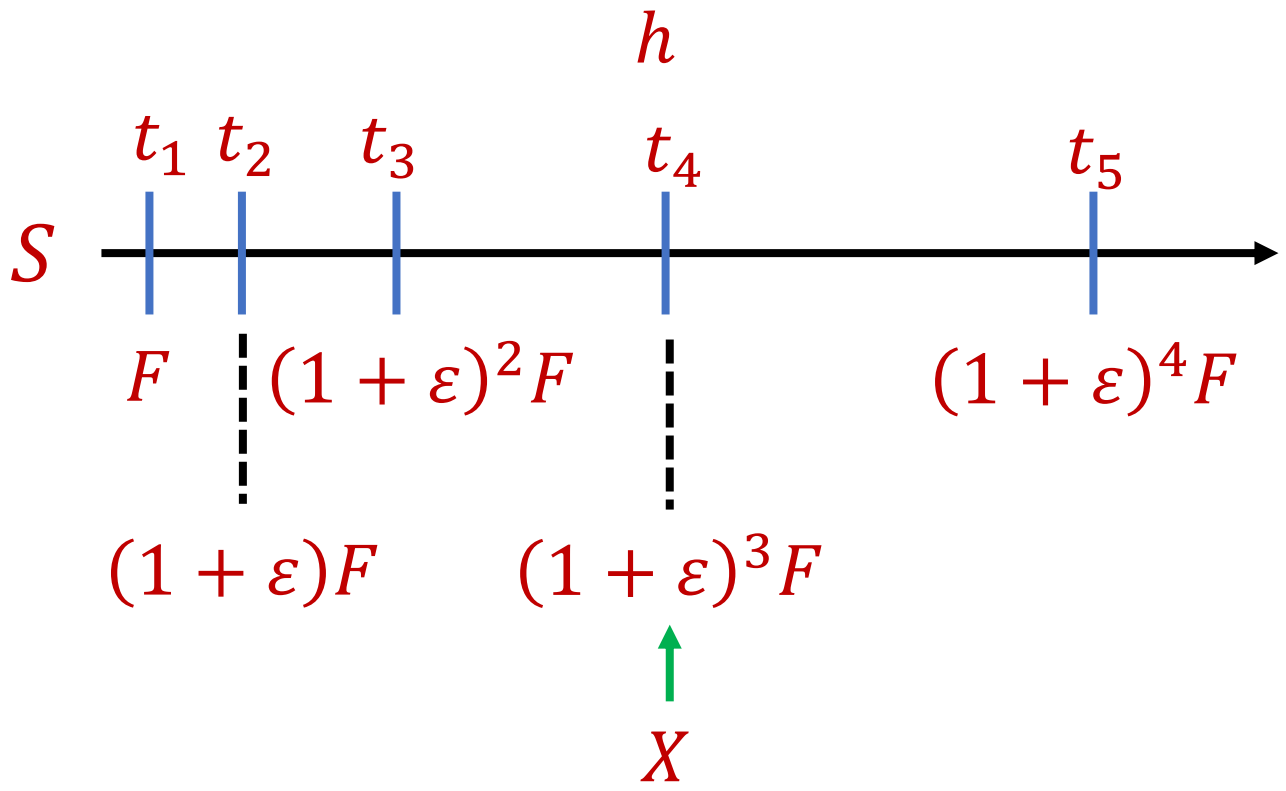
$f_{G,H}$



$$h_5 = H(h_3, h_4)$$

Robust Algorithms

Goal: $\tilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

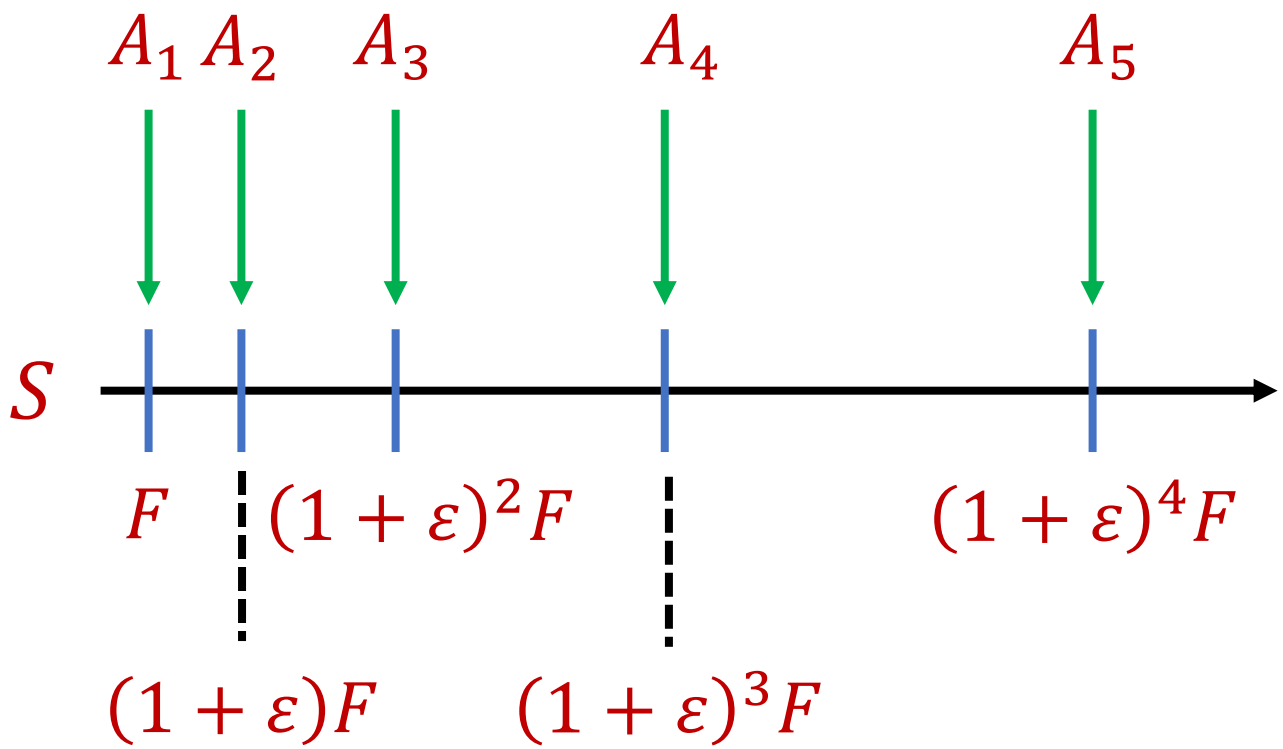


- Monotonic function h
- Consider times when h increases by $(1+\varepsilon)$
- Correctness at these times suffice!

If X is a $(1+\varepsilon)$-approximation to $(1+\varepsilon)^3F$, then X is a $(1+\varepsilon)^2$ approximation to all times before $(1+\varepsilon)^4F$

Robust Algorithms

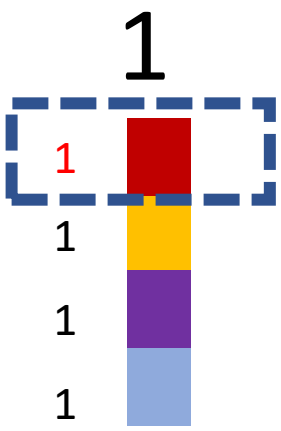
Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space



Robust Algorithms

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Start many instances of the streaming algorithm at the beginning
- Use an instance of the algorithm but "freeze" the output
- Each time the next instance has value $(1 + O(\varepsilon))$ more than the "frozen" output, use the next instance and "freeze" its output

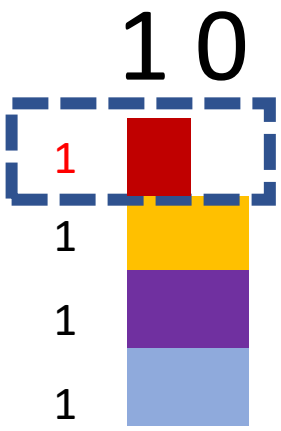


- Example: Number of ones in the stream (2-approximation)

Robust Algorithms

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Start many instances of the streaming algorithm at the beginning
- Use an instance of the algorithm but "freeze" the output
- Each time the next instance has value $(1 + O(\varepsilon))$ more than the "frozen" output, use the next instance and "freeze" its output

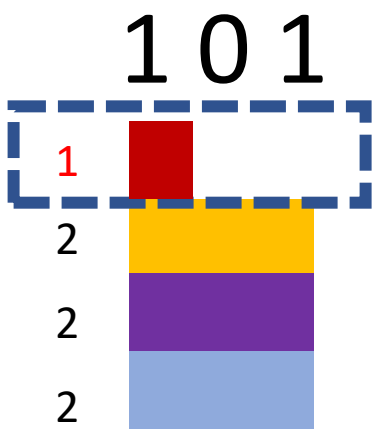


- Example: Number of ones in the stream (2-approximation)

Robust Algorithms

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Start many instances of the streaming algorithm at the beginning
- Use an instance of the algorithm but “freeze” the output
- Each time the next instance has value $(1 + O(\varepsilon))$ more than the “frozen” output, use the next instance and “freeze” its output



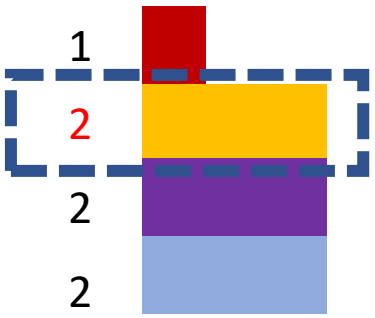
- Example: Number of ones in the stream (2-approximation)

Robust Algorithms

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Start many instances of the streaming algorithm at the beginning
- Use an instance of the algorithm but "freeze" the output
- Each time the next instance has value $(1 + O(\varepsilon))$ more than the "frozen" output, use the next instance and "freeze" its output

1 0 1



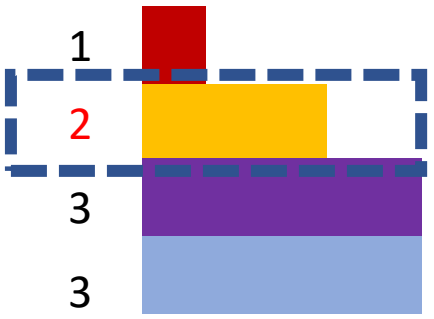
- Example: Number of ones in the stream (2-approximation)

Robust Algorithms

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Start many instances of the streaming algorithm at the beginning
- Use an instance of the algorithm but "freeze" the output
- Each time the next instance has value $(1 + O(\varepsilon))$ more than the "frozen" output, use the next instance and "freeze" its output

1 0 1 1



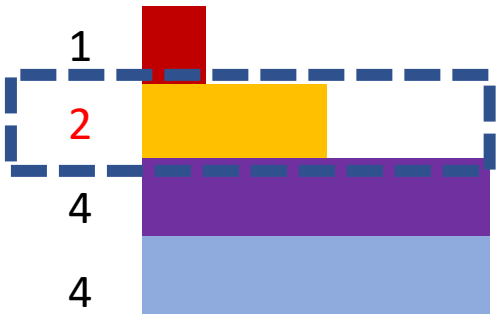
- Example: Number of ones in the stream (2-approximation)

Robust Algorithms

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Start many instances of the streaming algorithm at the beginning
- Use an instance of the algorithm but "freeze" the output
- Each time the next instance has value $(1 + O(\varepsilon))$ more than the "frozen" output, use the next instance and "freeze" its output

1 0 1 1 1



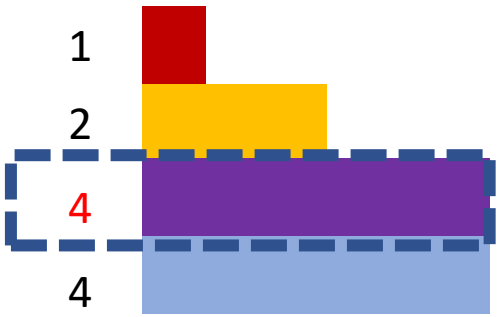
- Example: Number of ones in the stream (2-approximation)

Robust Algorithms

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Start many instances of the streaming algorithm at the beginning
- Use an instance of the algorithm but “freeze” the output
- Each time the next instance has value $(1 + O(\varepsilon))$ more than the “frozen” output, use the next instance and “freeze” its output

1 0 1 1 1



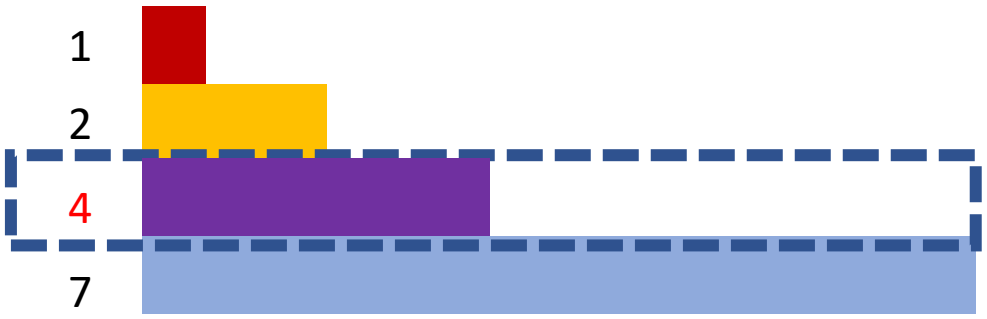
- Example: Number of ones in the stream (2-approximation)

Robust Algorithms

Goal: $\widetilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Start many instances of the streaming algorithm at the beginning
- Use an instance of the algorithm but "freeze" the output
- Each time the next instance has value $(1 + O(\varepsilon))$ more than the "frozen" output, use the next instance and "freeze" its output

1 0 1 1 1 0 0 1 1 0 1



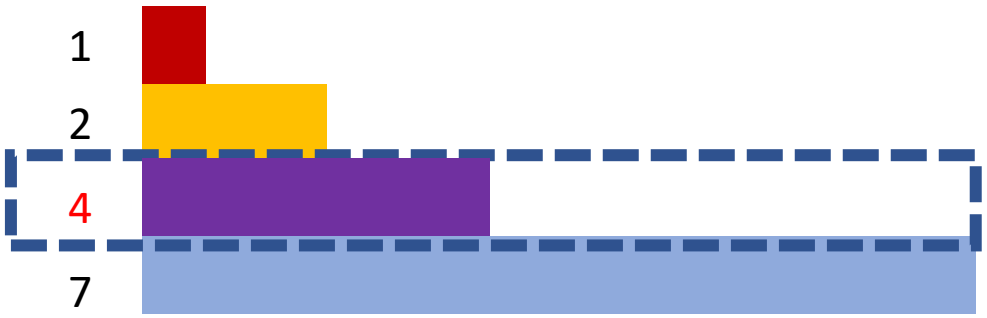
- Example: Number of ones in the stream (2-approximation)
- Number of ones stream is at least 4 and at most 8
- 4 is a good approximation

Intuition

Goal: $\tilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Approach uses $\frac{1}{\varepsilon^2}$ space each time function increases by $(1+\varepsilon)$ and function increases $\frac{1}{\varepsilon}$ times
- Total space: $\tilde{O}\left(\frac{1}{\varepsilon^3}\right)$

1 0 1 1 1 0 0 1 1 0 1



Intuition

Goal: $\tilde{\Theta}\left(\frac{1}{\varepsilon^2}\right)$ space

- Initialize many instances of the streaming algorithm
- Use a new instance of the algorithm each time the function increases by $(1+\varepsilon)$
- Approach uses $\frac{1}{\varepsilon^2}$ space each time function increases by $(1+\varepsilon)$ and function increases $\frac{1}{\varepsilon}$ times
- Total space: $\frac{1}{\varepsilon^2} \times \frac{1}{\varepsilon} = \tilde{O}\left(\frac{1}{\varepsilon^3}\right)$

census.gov:

Privacy & Confidentiality

Federal Law Protects Your Information. The U.S. Census Bureau is bound by Title 13 of the United States Code. This law not only provides authority for the work we do, but also provides strong protection for the information we collect from individuals and businesses. As a result, the Census Bureau has one of the strongest confidentiality guarantees in the federal government.

It is against the law for any Census Bureau employee to disclose or publish any census or survey information that identifies an individual or business. This is true even for inter-agency communication: the FBI and other government entities do not have the legal right to access this information. In fact, when these protections have been challenged, Title 13's confidentiality guarantee has been upheld.

For more information about how the Census Bureau safeguards the data it collects, visit the agency's Data Protection and Disclosure Avoidance Working Papers Web sites.

Anonymizing Data

Age	Zip Code	Employer	Has Pet
56	77005	Apple	Yes
32	77005	Microsoft	No
71	77005	Amazon	Yes
44	77005	Petsmart	Yes
25	77005	Netflix	No
61	77005	Google	No

Anonymizing Data

Age	Zip Code	Employer	Has Pet
56	77005	Apple	Yes
32	77005	Microsoft	No
71	77005	Amazon	Yes
44	77005	Petsmart	Yes
25	77005	Netflix	No
61	77005	Google	No

Name	Age	Gender	Employer
Alice	56	Female	Apple
Bob	32	Male	Microsoft
Carol	71	Female	Amazon
Dale	44	Male	Petsmart
Erin	25	Female	Netflix
Frank	61	Male	Google

Reconstruction Attack

Name	Age	Zip Code	Gender	Employer	Has Pet
Alice	56	77005	Female	Apple	Yes
Bob	32	77005	Male	Microsoft	No
Carol	71	77005	Female	Amazon	Yes
Dale	44	77005	Male	Petsmart	Yes
Erin	25	77005	Female	Netflix	No
Frank	61	77005	Male	Google	No

Implications of the simulated attack

The Census Bureau believed in 2010 that it was necessary to coarsen geographic identifiers in microdata such that the minimum population in any published geography was at least 100,000 persons (Public-Use Microdata Areas).

Our simulated reconstruction-abetted re-identification attack demonstrated that the tabular summaries from the 2010 Census can be converted into a 100% microdata file with geographic precision to the census block-level.

Our simulated attack demonstrated that, depending on the quality of the external data used, between 52 and 179 million respondents to the 2010 Census can be correctly re-identified from the reconstructed microdata.

Stronger privacy protections, such as those in the 2020 Census Disclosure Avoidance System, are necessary to protect against reconstruction-abetted attacks.

Differential Privacy

- [DMNS06] Given $\varepsilon > 0$ and $\delta \in (0,1)$, a randomized algorithm $A: U^* \to Y$ is (ε, δ)-differentially private if, for every neighboring frequency vectors f and f' and for all $E \subseteq Y$,

$$\Pr[A(f) \in E] \leq e^{\varepsilon} \Pr[A(f') \in E] + \delta$$

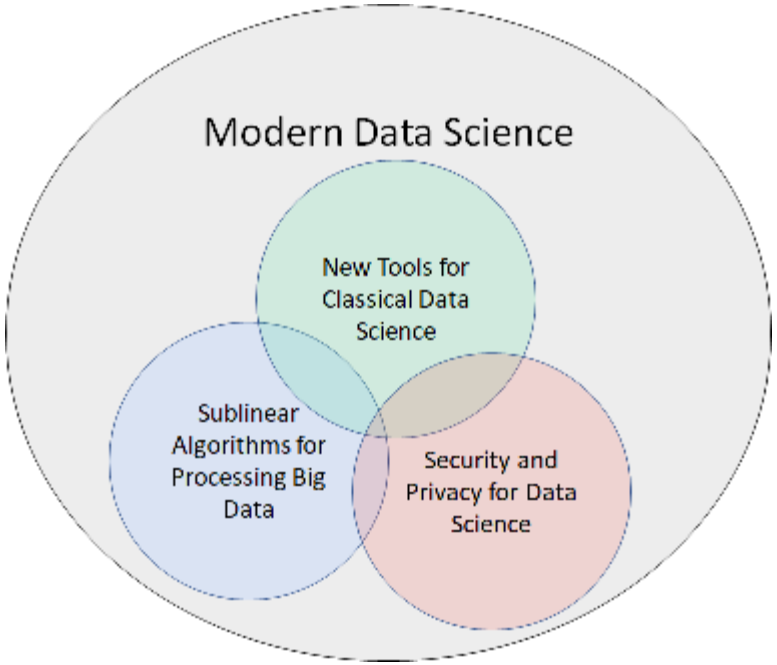
$A($    $) \approx A($    $)$

Differential Privacy Result Highlights

- Framework for converting FPTAS/FPRAS algorithms to private FPTAS/FPRAS algorithms [BGMZ22]
- Privacy has a price in memory cost [DSWZ22]
- No overhead in communication cost for summation in the differentially oblivious shuffle model [GKNMZ22]

Evolving Demands

- Sublinear-time or sublinear-space algorithms
- Ability to handle time-sensitive data
- Robustness to noise or adversarial input
- Incorporation of advice
- Differential privacy
- Fairness

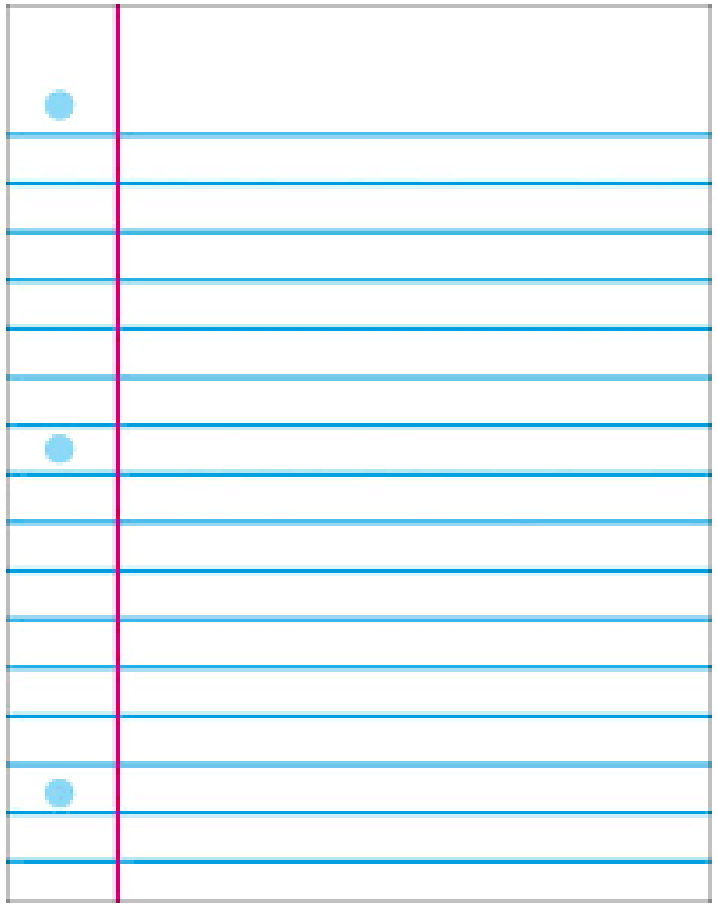



Swingline

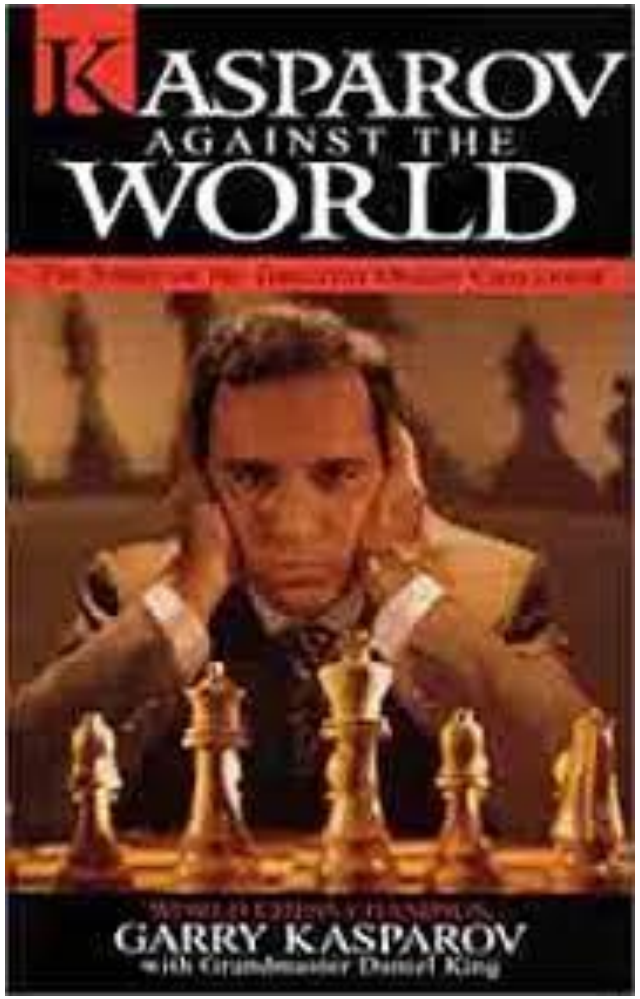

12 Assorted Colors
Crayola
MARKERS
12


Paper Mate
Pink Pearl
Latex FREE!


ELMER'S
WASHABLE, NO RUN
SCHOOL
GLUE




Scotch



Kasparov versus the World

From Wikipedia, the free encyclopedia

Kasparov versus the World was a game of chess played in 1999 over the Internet.[1] It was a consultation game, in which a World Team of thousands decided each move for the black pieces by plurality vote, while Garry Kasparov conducted the white pieces by himself. More than 50,000 people from over 75 countries participated in the game.

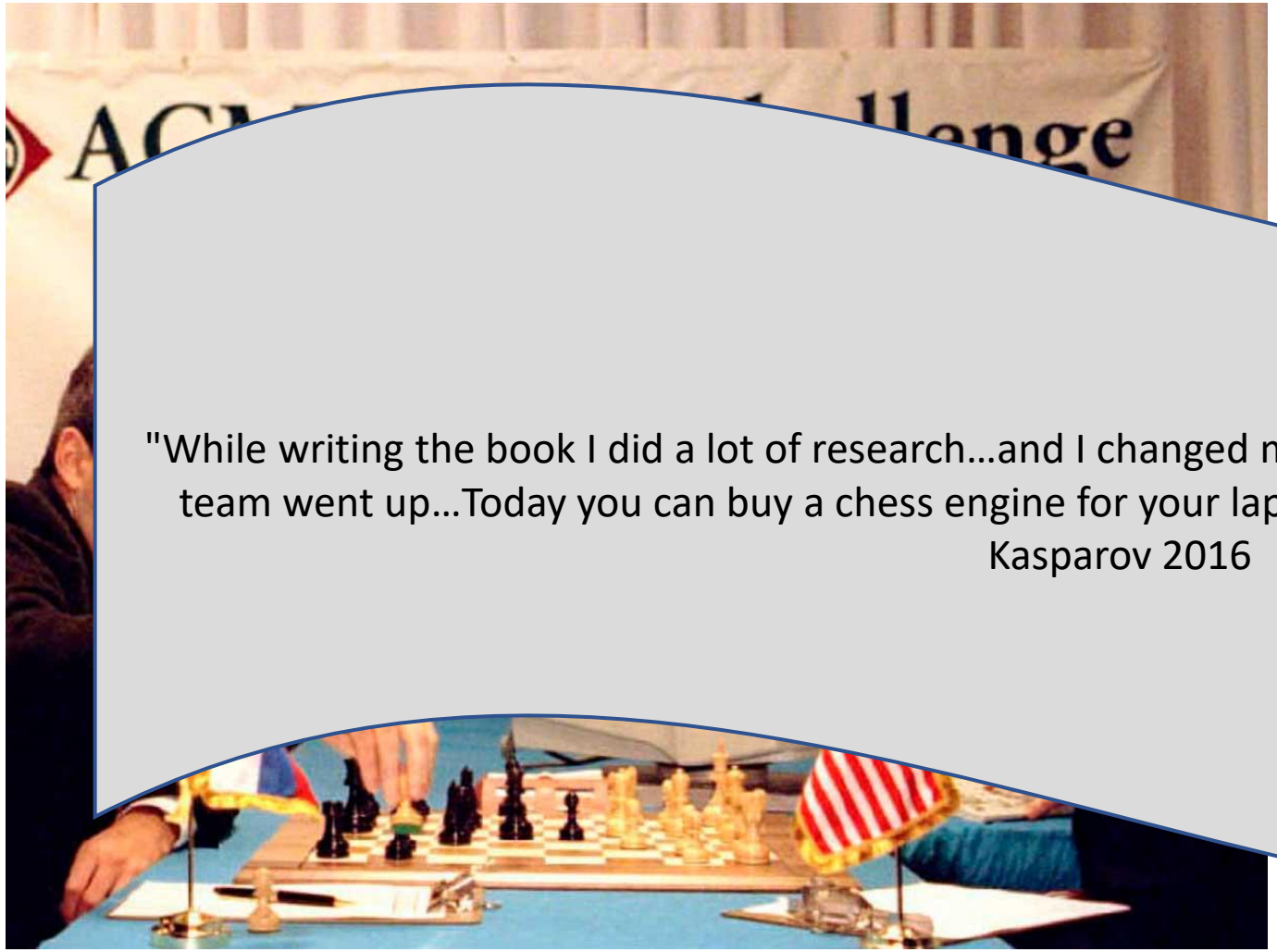
The host and promoter of the match was the MSN Gaming Zone, with sponsorship from First USA bank.[2] After 62 moves played over four months, Kasparov won the game

The 1996 match

Game #	White	Black	Result	Method of conclusion
1	**Deep Blue**	Kasparov	1–0	Resignation
2	**Kasparov**	Deep Blue	1–0	Resignation
3	Deep Blue	Kasparov	½–½	Draw by mutual agreement
4	Kasparov	Deep Blue	½–½	Draw by mutual agreement
5	Deep Blue	**Kasparov**	0–1	Resignation
6	**Kasparov**	Deep Blue	1–0	Resignation
***Result:* Kasparov–Deep Blue: 4–2**				

The 1997 rematch

Game #	White	Black	Result	Method of conclusion
1	**Kasparov**	Deep Blue	1–0	Resignation
2	**Deep Blue**	Kasparov	1–0	Resignation
3	Kasparov	Deep Blue	½–½	Draw by mutual agreement
4	Deep Blue	Kasparov	½–½	Draw by mutual agreement
5	Kasparov	Deep Blue	½–½	Draw by mutual agreement
6	**Deep Blue**	Kasparov	1–0	Resignation
***Result:* Deep Blue–Kasparov: 3½–2½**				



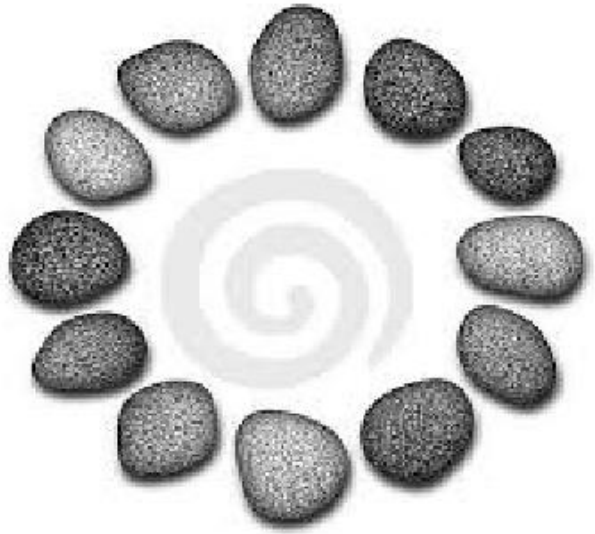
"While writing the book I did a lot of research…and I changed my conclusions…My respect for the Deep Blue team went up…Today you can buy a chess engine for your laptop that will beat Deep Blue quite easily.“ – Kasparov 2016

The 1996 match

Game #	White	Black	Result	Method of conclusion
1	**Deep Blue**	Kasparov	1–0	Resignation
2	**Kasparov**	Deep Blue	1–0	Resignation
3	Deep Blue	Kasparov	½–½	Draw by mutual agreement
4	Kasparov	Deep Blue	½–½	Draw by mutual agreement
5	Deep Blue	**Kasparov**	0–1	Resignati
				clusion
				greement
				agreement
5				Draw by mutual agreement
6	**Deep Blue**	Kasparov	1–0	Resignation
Result: **Deep Blue–Kasparov: 3½–2½**				

iMHFs

❖ Calculating an iMHF can be modeled as graph pebbling [AS15]



Background

- Users tend to pick weak passwords
- Server attacks are inevitable

TOP 20 MOST COMMON PASSWORDS

(as a percentage of all passwords)

1.	123456	4.1%	11.	login	0.2%
2.	password	1.3%	12.	welcome	0.2%
3.	12345	0.8%	13.	loveme	0.2%
4.	1234	0.6%	14.	hottie	0.2%
5.	football	0.3%	15.	abc123	0.2%
6.	qwerty	0.3%	16.	121212	0.2%
7.	1234567890	0.3%	17.	123654789	0.2%
8.	1234567	0.3%	18.	flower	0.2%
9.	princess	0.3%	19.	passw0rd	0.2%
10.	solo	0.2%	20.	dragon	0.1%

Source: SkyHigh


YAHOO!
UBER
Gmail
by Google
Adobe
BLIZZARD
ENTERTAINMENT
eHarmony
AMERICA
Online
ASHLEY
MADISON.COM
LastPass ****
ebay
Dropbox
LinkedIn
PlayStation


YAHOO!
OHIO STATE
Gmail
by Google
Adobe
AMERICA Online
U.S. DEPARTMENT OF HOMELAND SECURITY
BLIZZARD ENTERTAINMENT
eHarmony
LastPass ****
ebay
Dropbox
LinkedIn
PlayStation

UBER

U.S. DEPARTMENT OF HOMELAND SECURITY

DNC
DEMOCRATIC NATIONAL COMMITTEE

COX COMMUNICATIONS

Dropbox

PlayStation™

User	Password
Stephen	auhsoJ
Lisa	hsifdrowS
James	1010NO1Z
Harry	sinocarD tupaC
Sarah	auhsoJ

User	Password Hash
Stephen	39e717cd3f5c4be78d97090c69f4e655
Lisa	f567c40623df407ba980bfad6dff5982
James	711f1f88006a48859616c3a5cbcc0377
Harry	fb74376102a049b9a7c5529784763c53
Sarah	39e717cd3f5c4be78d97090c69f4e655

User	Random Salt	Password Hash
Stephen	06917d7ed65c466fa180a6fb62313ab9	b65578786e544b6da70c3a9856cdb750
Lisa	51f2e43105164729bb46e7f20091adf8	2964e639aa7d457c8ec0358756cbffd9
James	fea659115b7541479c1f956a59f7ad2f	dd9e4cd20f134dda87f6ac771c48616f
Harry	30ebf72072134f1bb40faa8949db6e85	204767673a8d4fa9a7542ebc3eceb3a2
Sarah	711f51082ea84d949f6e3efecf29f270	e3afb27d59a34782b6b4baa0c37e2958

Motivation

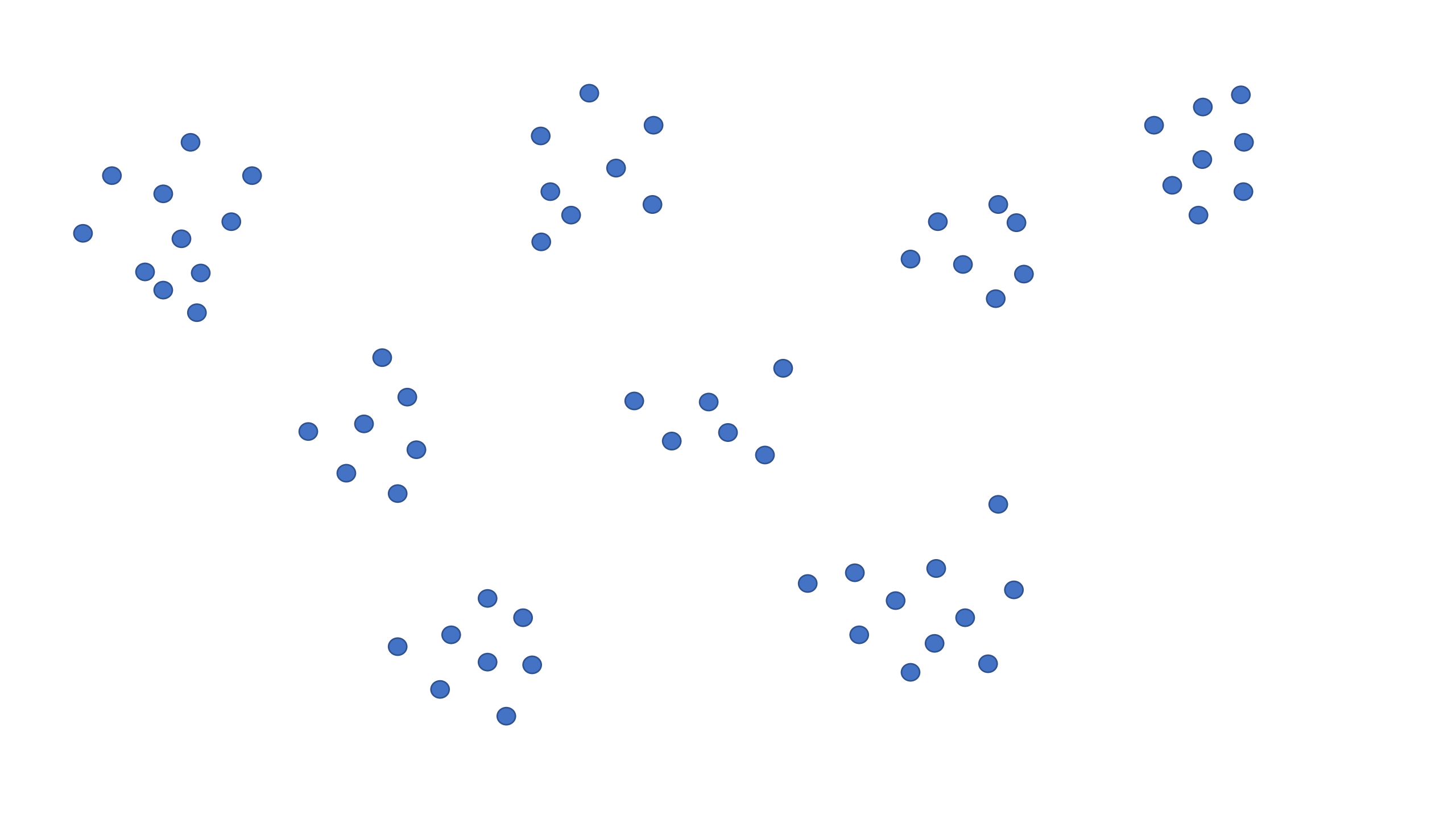
- Users tend to pick weak passwords
- Server attacks are inevitable
- Try to mitigate offline attacks
- Specialized hardware (ASIC) can compute 10^{12} hashes per second.

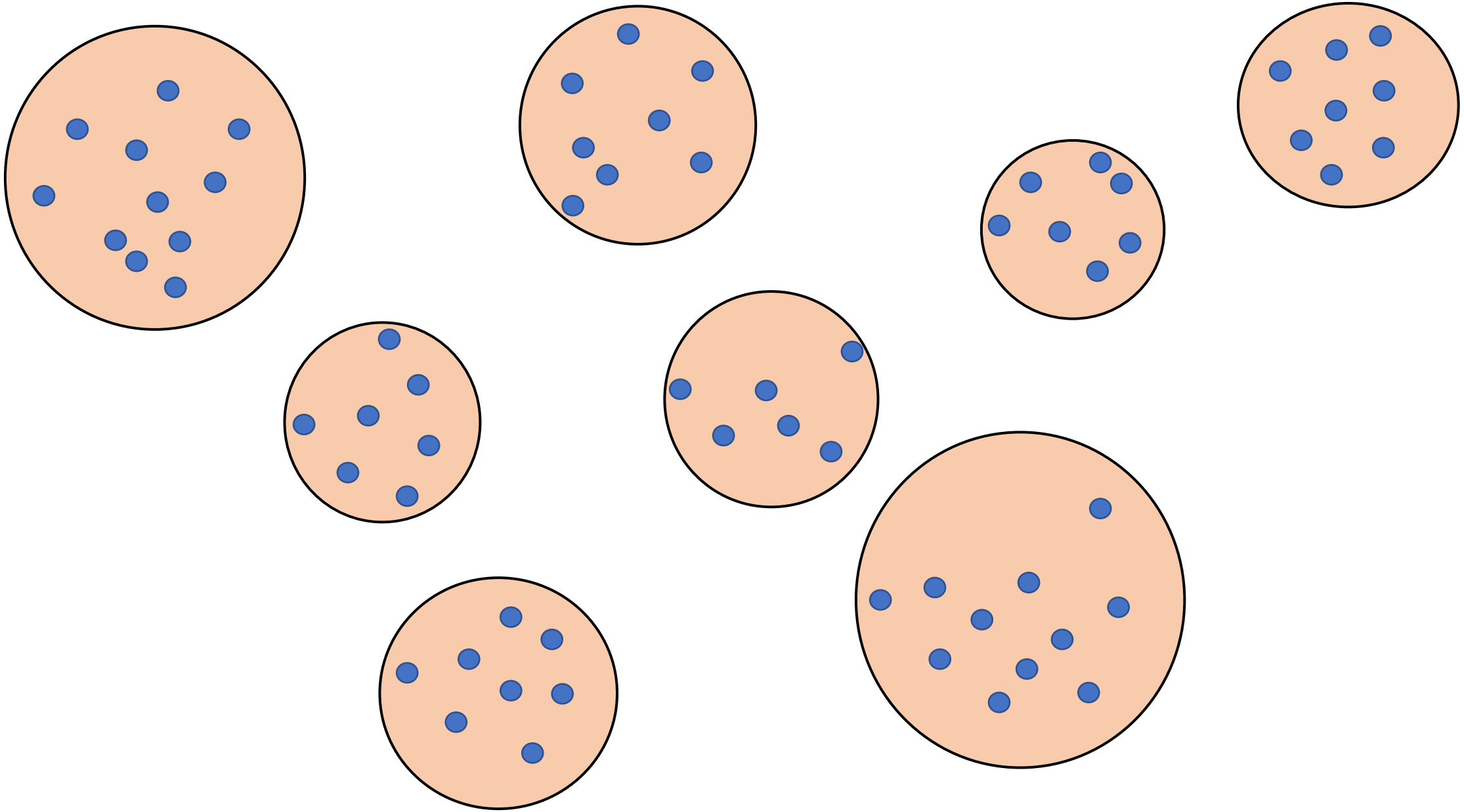
Memory Hard Functions

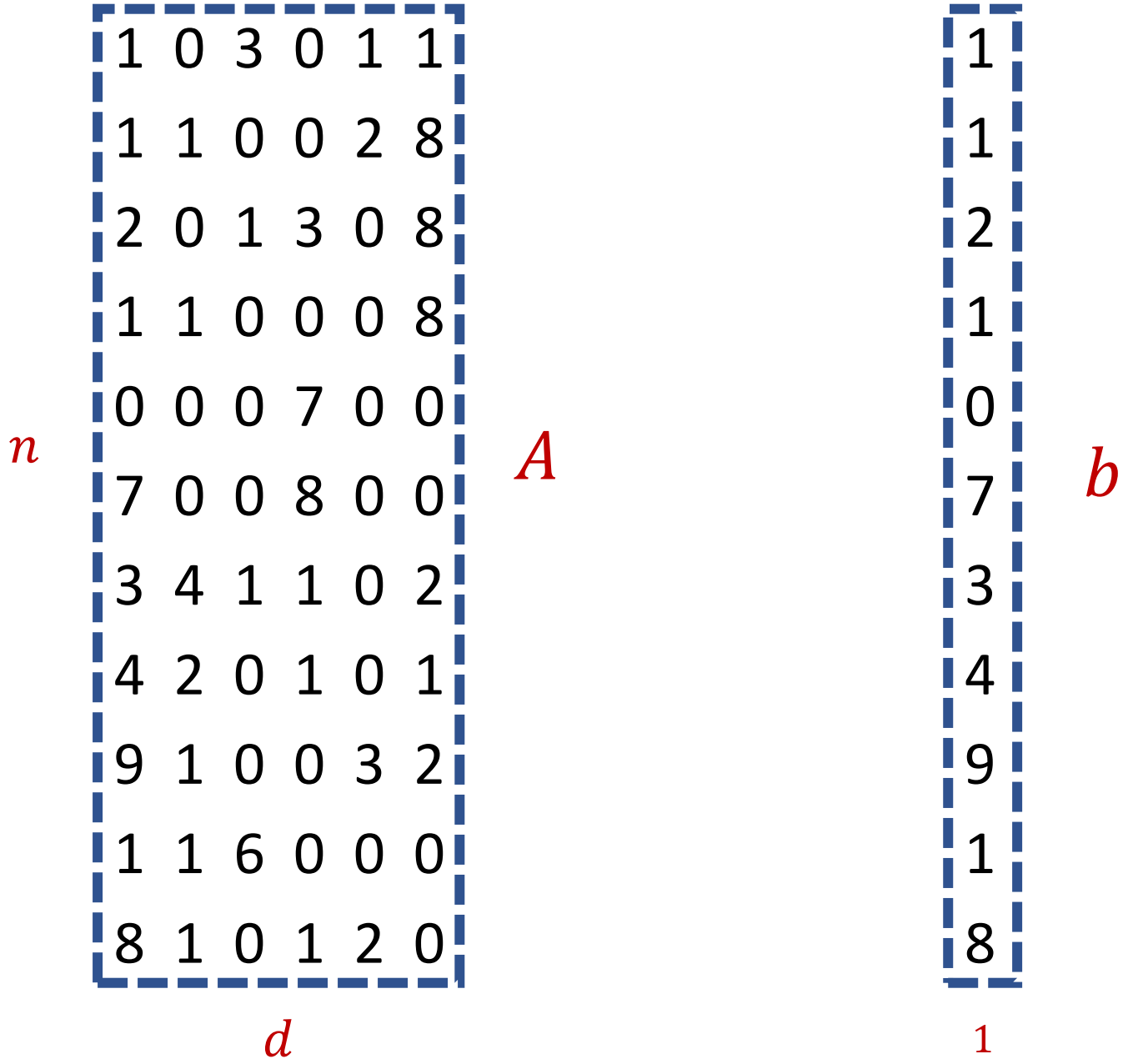
- Memory hard functions require comparatively more resources for adversaries to compute
- Data-dependent memory hard functions are susceptible to side-channel attacks
- Data-independent memory hard functions (iMHFs)

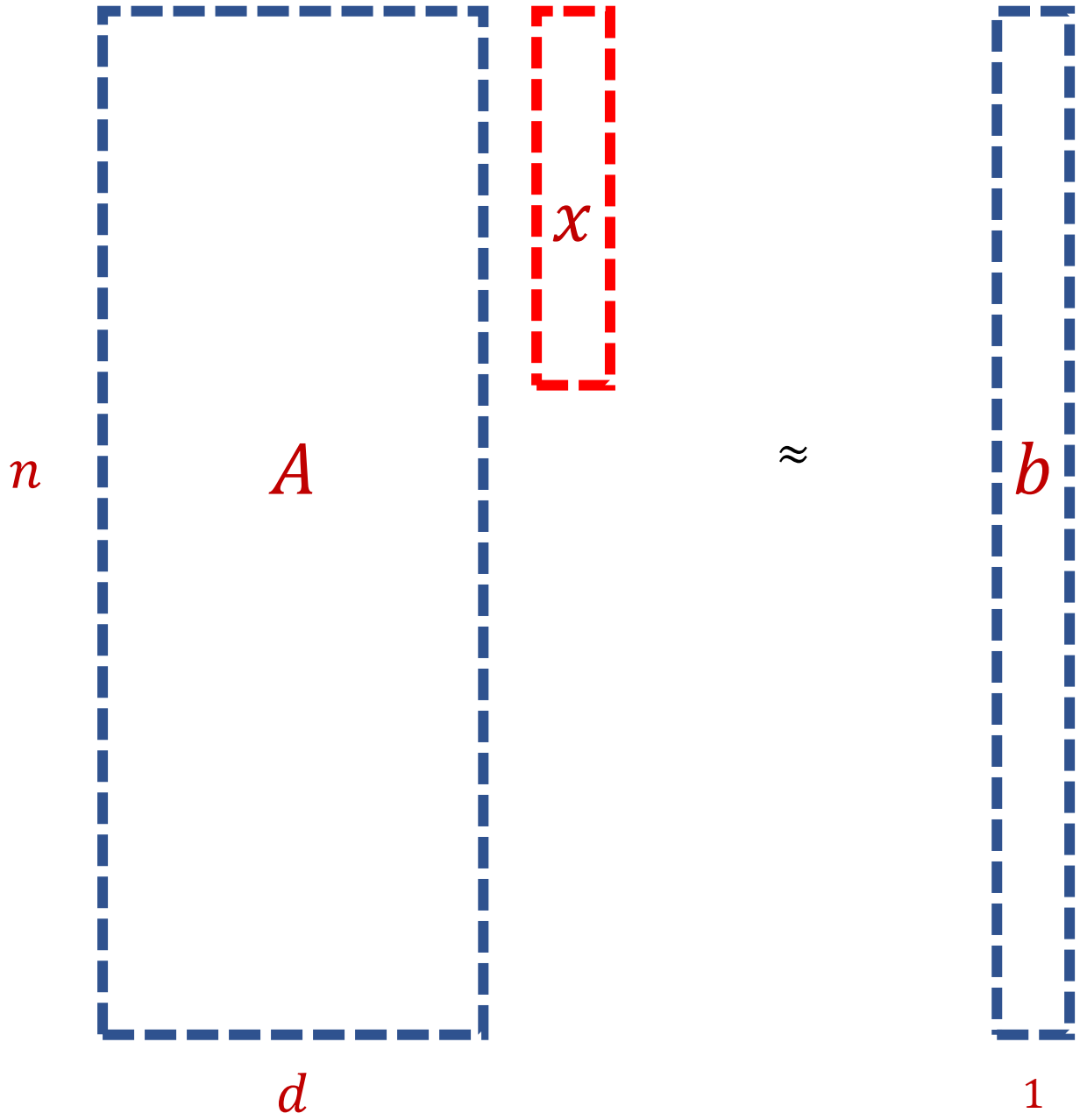
Graph Pebbling and Password Hashing Results

- Computing $cc(G)$ is NP-hard [BZ18] and UGC-hard to approximate within any constant factor [BLZ20]
- Implication: Cryptanalysis of memory-hard functions is provably hard
- Existing iMHF Argon2i is weaker than previous thought [BZ17]
- Any economical adversary is inclined to crack all passwords under hashing but many can be saved by iMHFs [BHZ18]



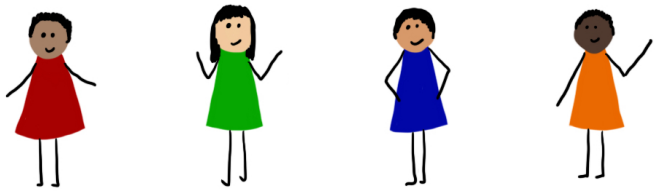
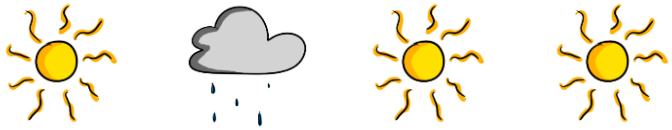
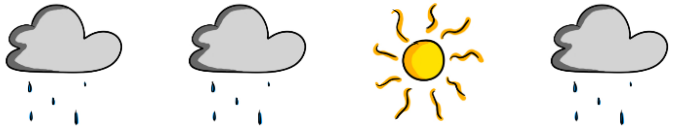
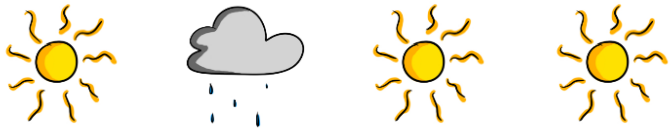
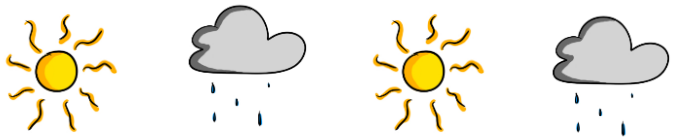



1 0 3 0 1 1
1 1 0 0 2 8
2 0 1 3 0 8
1 1 0 0 0 8
0 0 0 7 0 0
7 0 0 8 0 0
3 4 1 1 0 2
4 2 0 1 0 1
9 1 0 0 3 2
1 1 6 0 0 0
8 1 0 1 2 0
n
A
d
1
1
2
1
0
7
3
4
9
1
8
b
1


n
A
d
x
≈
b
1

Prediction with Expert Advice

a fundamental problem of **sequential prediction**

Day		You	Actual outcome
1		?	
2		?	
3		?	
4		?	

Quantifying Performance

Make no distributional assumptions

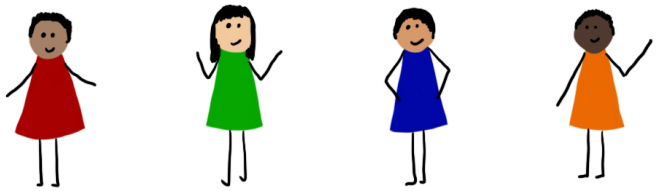
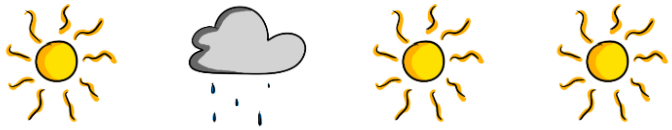
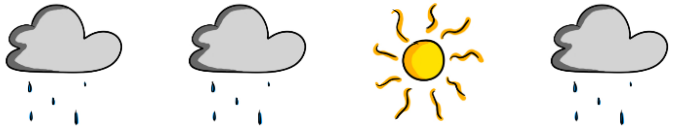
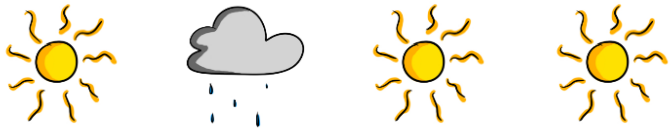
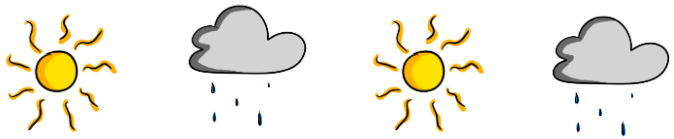
We judge our algorithm based on **regret**.

Definition (Regret)

$$\frac{\text{\# of mistakes algorithm makes more than the best expert}}{\text{\# of days}}$$

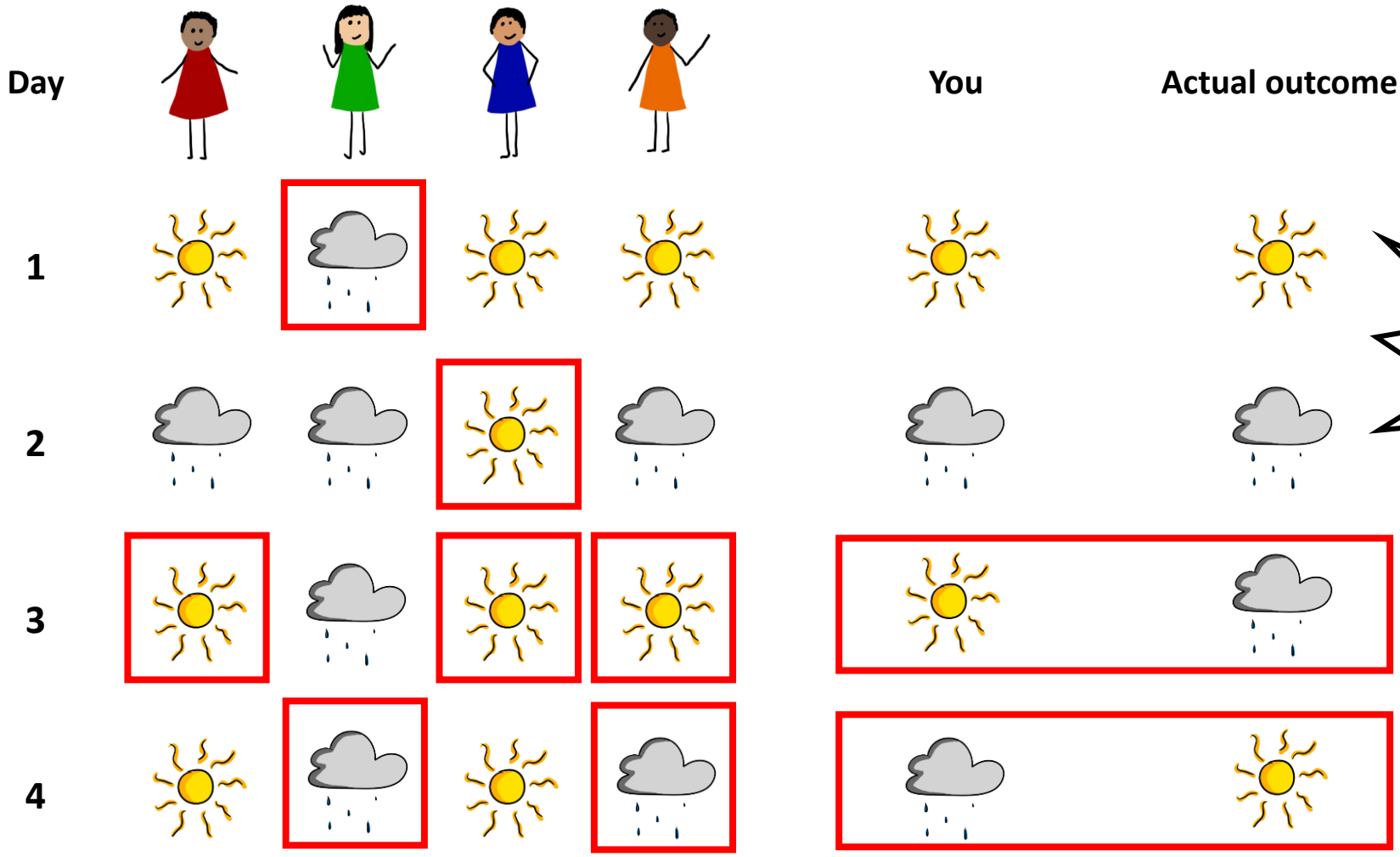
Prediction with Expert Advice

a fundamental problem of **sequential prediction**

Day		You	Actual outcome
1		?	
2		?	
3		?	
4		?	

Prediction with Expert Advice

a fundamental problem of **sequential prediction**



The Online Learning with Experts Problem

- n experts who decide either $\{0,1\}$ on each of T days ($n \gg T$)
- Algorithm takes advice from experts and predict either $\{0,1\}$ on each day
- Algorithm sees the outcome, which is either $\{0,1\}$, of each day and can use this information on future days
- The cost of the algorithm is the number of incorrect predictions

- Regret is (# of mistakes we make - M)/T, i.e., the amortized additional cost of the algorithm compared to the cost M of the best expert

Applications of the Experts Problem

- Ensemble learning, e.g., AdaBoost
- Forecast and portfolio optimization
- Special case of online convex optimization

State-of-the-Art for Unbounded Memory

- Weighted majority algorithm down-weights each expert that is incorrect on each day and selects the weighted majority as the output
- Weighted majority algorithm gets $(2+\varepsilon)M + \frac{O(\log n)}{\varepsilon}$ total mistakes
- Randomized weighted majority algorithm randomly follows each expert with probability proportional to the weight of the expert
- Randomized weighted majority algorithm achieves regret $O\left(\sqrt{\frac{\log n}{T}}\right)$

Memory Bounds for the Expert Problem

- These algorithms require $\Omega(n)$ memory to maintain weights for each expert – but what if n is very large and we want sublinear space?

- Can use no memory and just randomly guess each day – still good if the best expert makes a lot of mistakes but bad if the best expert makes very few mistakes

- What are the space/accuracy tradeoffs for the online learning with experts problem?

Online Learning with Experts in the Streaming Model (I) [SWXZ22]

- Any algorithm that achieves $\delta < \frac{1}{2}$ regret with probability at least $\frac{3}{4}$ must use $\Omega\left(\frac{n}{\delta^2 T}\right)$ space

- Lower bound holds for arbitrary-order, random-order, and i.i.d. streams

Online Learning with Experts in the Streaming Model (II) [SWXZ22]

- There exists an algorithm that uses $O\left(\frac{n}{\delta^2 T}\log^2 n \log\frac{1}{\delta}\right)$ space and achieves expected regret $\delta > \sqrt{\frac{8\log n}{T}}$ in the random-order model

- The algorithm is almost-tight with the space lower bounds and oblivious to M, the number of mistakes made by the best-expert

- Can achieve regret almost matching randomized weighted majority

- Result extends to general costs in $[0, \rho]$ with expected regret $\rho\delta$

Online Learning with Experts in the Streaming Model (III) [SWXZ22]

- For $M < \frac{\delta^2 T}{1280 \log^2 n}$ and $\delta > \sqrt{\frac{128 \log^2 n}{T}}$, there exists an algorithm that uses $\tilde{O}\left(\frac{n}{\delta T}\right)$ space and achieves regret δ with probability $\frac{4}{5}$

- The algorithm *beats* the lower bounds, showing that the hardness comes from the best expert making a "lot" of mistakes
- Can achieve regret almost matching randomized weighted majority
- The algorithm oblivious to M, the number of mistakes made by the best expert

Subsequent Work

- [PZ23] achieved the first sublinear space algorithm for arbitrary-order streams without any assumptions on the best expert
- Subsequently improved by [PR23] to upper bounds tight with our lower bounds
- [WZZ23] showed that the “natural” deterministic algorithm is the best possible for deterministic algorithms

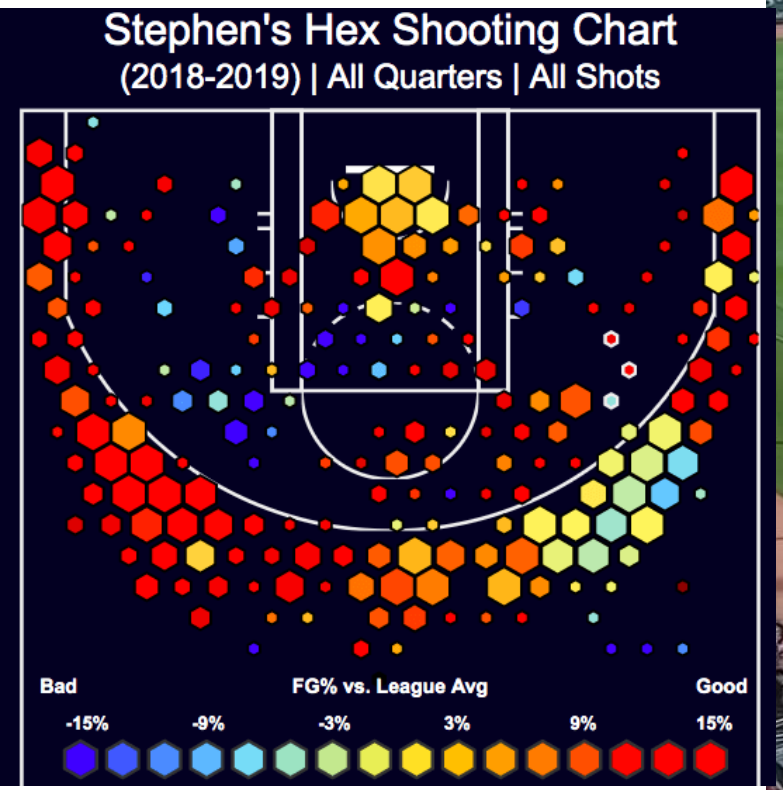
Additional Work

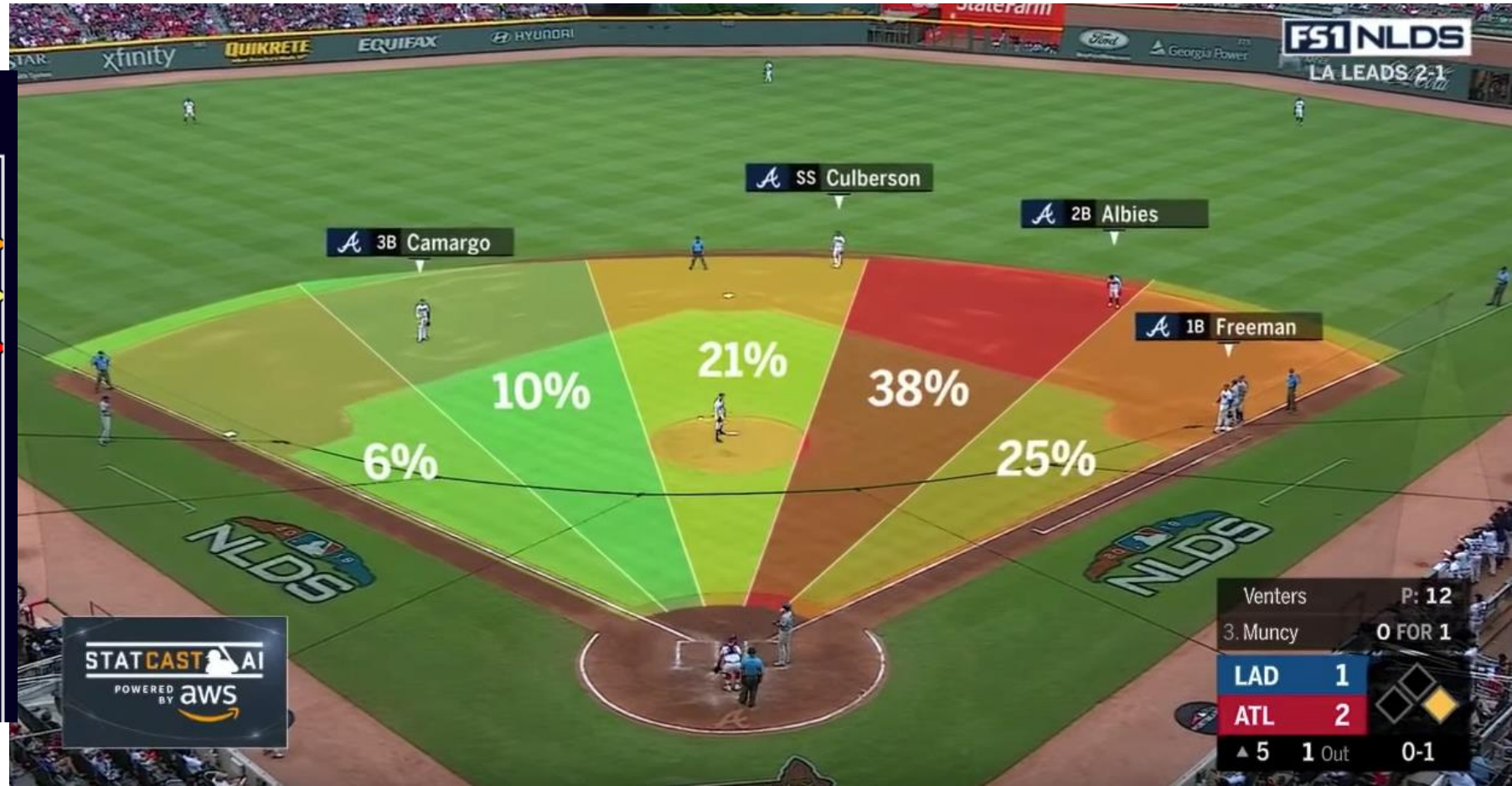
- Learning-augmented algorithms for online linear and semidefinite programming [GLSSZ17],
- Algorithms robust to noisy input for detection of repetitive structure, such as alignments [GSZ17], palindromic structure [GSZ17], periodicity [EGSZ17], [EGSZ18]
- Non-adaptive adaptive sampling on data streams [MRWZ20] for data summarization problems such as volume maximization, projective clustering, column/row subset selection, subspace approximation
- Network monitoring algorithms to detect high packet loss, high round trip time, high retransmissions, high out-of-order packets [LZRBR20]

UCLA Football: Chip Kelly's Sport Science Program is Helping the Bruins

The sport science program combined with Frank Wintrich's strength and conditioning program looks to pay off for UCLA football.

By Joe Piechowski | @JPiechowski | Aug 19, 2019, 7:55am PDT



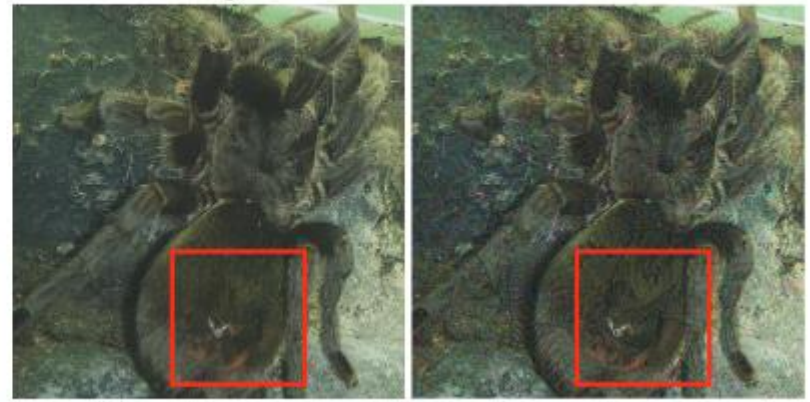


texture modification

image adv

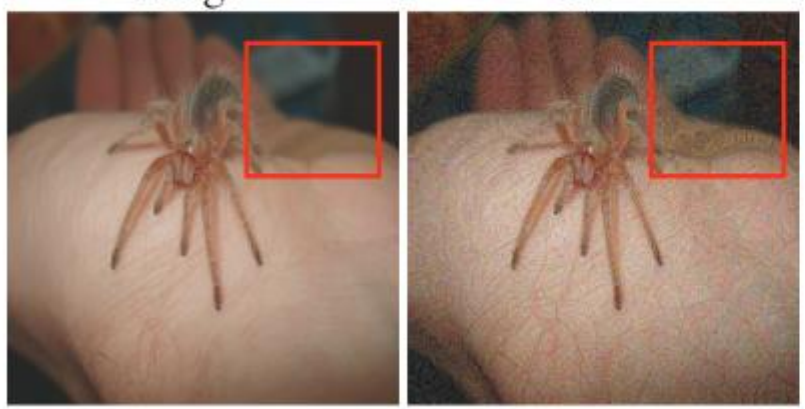
dark parts modification

image adv



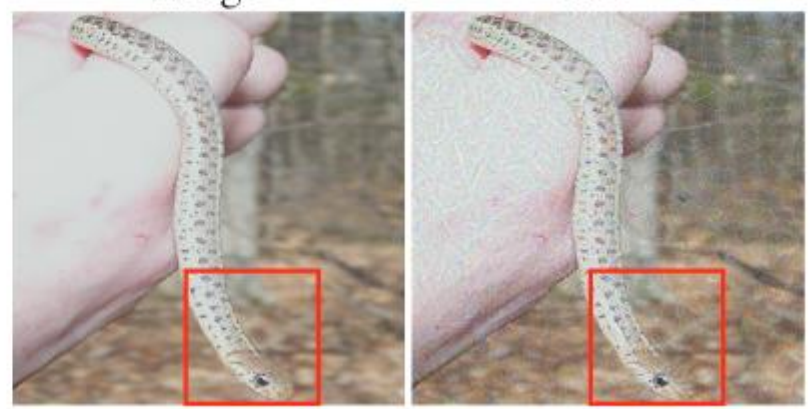
edge enhancement

image adv



edge destruction

image adv



$(1+\varepsilon)$-Approximation Streaming Algorithms

- Space $O\left(\frac{1}{\varepsilon^2}+\log n\right)$ algorithm for F_0 [KNW10], [Blasiok20]
- Space $O\left(\frac{1}{\varepsilon^2}\log n\right)$ algorithm for F_p with $p \in (0,2]$ [BDN17]
- Space $O\left(\frac{1}{\varepsilon^2} n^{1-2/p} \log^2 n\right)$ algorithm for F_p with $p > 2$ [Ganguly11, GW18]
- Space $O\left(\frac{1}{\varepsilon^2}\log n\right)$ algorithm for L_2-heavy hitters [BCINWW17]

$(1+\varepsilon)$-Robust Algorithms [BJWY20]

- Space $\tilde{O}\left(\frac{1}{\varepsilon^3}\log n\right)$ algorithm for F_0
- Space $\tilde{O}\left(\frac{1}{\varepsilon^3}\log n\right)$ algorithm for F_p with $p \in (0,2]$
- Space $\tilde{O}\left(\frac{1}{\varepsilon^3}n^{1-2/p}\right)$ algorithm for F_p with $p > 2$
- Space $\tilde{O}\left(\frac{1}{\varepsilon^3}\log n\right)$ algorithm for L_2-heavy hitters

"A general framework that loses* nothing in n and only $\frac{1}{\varepsilon}$ "

“What’s an epsilon between friends?

- Statista: $\sim 350B$ e-mails sent per day
- Unsigned integer range: $n = 2^{32} \sim 4B$
- Accuracy: $\varepsilon = 0.01$
- Since $\frac{1}{\varepsilon} > \log n$, we should care about $\frac{1}{\varepsilon}$ factors!

$(1+\varepsilon)$-Robust Algorithms [HKMMS20]

- Space $\tilde{O}\left(\frac{1}{\varepsilon^{2.5}}\log^4 n\right)$ algorithm for F_0
- Space $\tilde{O}\left(\frac{1}{\varepsilon^{2.5}}\log^4 n\right)$ algorithm for F_p with $p \in (0,2]$
- Space $\tilde{O}\left(\frac{1}{\varepsilon^{2.5}}n^{1-2/p}\right)$ algorithm for F_p with $p > 2$
- Space $\tilde{O}\left(\frac{1}{\varepsilon^{2.5}}\log^4 n\right)$ algorithm for L_2-heavy hitters

"$\frac{1}{\varepsilon}$ losses are not necessary"

$(1+\varepsilon)$-Robust Algorithms [WZ21]

- Space $\tilde{O}\left(\frac{1}{\varepsilon^2}\log n\right)$ algorithm for F_0
- Space $\tilde{O}\left(\frac{1}{\varepsilon^2}\log n\right)$ algorithm for F_p with $p \in (0,2]$
- Space $\tilde{O}\left(\frac{1}{\varepsilon^2}n^{1-2/p}\right)$ algorithm for F_p with integer $p > 2$
- Space $\tilde{O}\left(\frac{1}{\varepsilon^2}\log n\right)$ algorithm for L_2-heavy hitters

"No losses* are necessary!"

$(1+\varepsilon)$-Robust Algorithms Summary [WZ21]

Problem	[BJWY20] Space	[HKM+20] Space	Our Result
Distinct Elements	$\tilde{\mathcal{O}}\left(\frac{\log n}{\varepsilon^3}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^4 n}{\varepsilon^{2.5}}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log n}{\varepsilon^2}\right)$
F_p Estimation, $p \in (0,2]$	$\tilde{\mathcal{O}}\left(\frac{\log n}{\varepsilon^3}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^4 n}{\varepsilon^{2.5}}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log n}{\varepsilon^2}\right)$
Shannon Entropy	$\tilde{\mathcal{O}}\left(\frac{\log^6 n}{\varepsilon^5}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^4 n}{\varepsilon^{3.5}}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^3 n}{\varepsilon^2}\right)$
L_2-Heavy Hitters	$\tilde{\mathcal{O}}\left(\frac{\log n}{\varepsilon^3}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^4 n}{\varepsilon^{2.5}}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log n}{\varepsilon^2}\right)$
F_p Estimation, integer $p > 2$	$\tilde{\mathcal{O}}\left(\frac{n^{1-2/p}}{\varepsilon^3}\right)$	$\tilde{\mathcal{O}}\left(\frac{n^{1-2/p}}{\varepsilon^{2.5}}\right)$	$\tilde{\mathcal{O}}\left(\frac{n^{1-2/p}}{\varepsilon^2}\right)$
F_p Estimation, $p \in (0,2]$, flip number λ	$\tilde{\mathcal{O}}\left(\frac{\lambda \log^2 n}{\varepsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^3 n\sqrt{\lambda \log n}}{\varepsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{\lambda \log^2 n}{\varepsilon}\right)$

$(1+\varepsilon)$-Approximation Sliding Window Algorithms [WZ21]

- Space $\tilde{O}\left(\frac{1}{\varepsilon^2}\log^3 n\right)$ algorithm for F_p with $p \in (0,2]$ [WZ21]

Problem	[BO07] Space	Our Result
L_p Estimation, $p \in (0,1)$	$\tilde{\mathcal{O}}\left(\frac{\log^3 n}{\varepsilon^3}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^3 n}{\varepsilon^2}\right)$
L_p Estimation, $p \in (1,2]$	$\tilde{\mathcal{O}}\left(\frac{\log^3 n}{\varepsilon^{2+p}}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^3 n}{\varepsilon^2}\right)$
L_p Estimation, integer $p > 2$	$\tilde{\mathcal{O}}\left(\frac{n^{1-2/p}}{\varepsilon^{2+p}}\right)$	$\tilde{\mathcal{O}}\left(\frac{n^{1-2/p}}{\varepsilon^2}\right)$
Entropy Estimation	$\tilde{\mathcal{O}}\left(\frac{\log^5 n}{\varepsilon^4}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^5 n}{\varepsilon^2}\right)$

"$\frac{1}{\varepsilon}$ losses are not necessary"

Differential Privacy

- [DMNS06] Given $\varepsilon > 0$ and $\delta \in (0,1)$, a randomized algorithm $A: U^* \to Y$ is (ε, δ)-differentially private if, for every neighboring frequency vectors f and f' and for all $E \subseteq Y$,

$$\Pr[A(f) \in E] \leq e^{\varepsilon} \Pr[A(f') \in E] + \delta$$

$A($  $) \approx A($  $)$

$(1+\varepsilon)$-Robust Algorithms [BJWY20]

- $\tilde{O}\left(\frac{1}{\varepsilon^3}\right)$ space-accuracy tradeoff for F_0
- $\tilde{O}\left(\frac{1}{\varepsilon^3}\right)$ space-accuracy tradeoff for F_p with $p \in (0,2]$
- $\tilde{O}\left(\frac{1}{\varepsilon^3}\right)$ space-accuracy tradeoff for F_p with $p > 2$
- $\tilde{O}\left(\frac{1}{\varepsilon^3}\right)$ space-accuracy tradeoff for L_2-heavy hitters

"A general framework that loses* nothing in n and only $\frac{1}{\varepsilon}$ "

“What’s an epsilon between friends?

- Statista: $\sim 350B$ e-mails sent per day
- Unsigned integer range: $n = 2^{32} \sim 4B$
- Accuracy: $\varepsilon = 0.01$
- Since $\frac{1}{\varepsilon} > \log n$, we should care about $\frac{1}{\varepsilon}$ factors!

$(1+\varepsilon)$-Robust Algorithms [HKMMS20]

- $\tilde{O}\left(\frac{1}{\varepsilon^{2.5}}\right)$ space-accuracy tradeoff for F_0
- $\tilde{O}\left(\frac{1}{\varepsilon^{2.5}}\right)$ space-accuracy tradeoff for F_p with $p \in (0,2]$
- $\tilde{O}\left(\frac{1}{\varepsilon^{2.5}}\right)$ space-accuracy tradeoff for F_p with $p > 2$
- $\tilde{O}\left(\frac{1}{\varepsilon^{2.5}}\right)$ space-accuracy tradeoff for L_2-heavy hitters

"$\frac{1}{\varepsilon}$ losses are not necessary"