When is Memorization Necessary for Machine Learning?

BOSTON UNIVERSITY

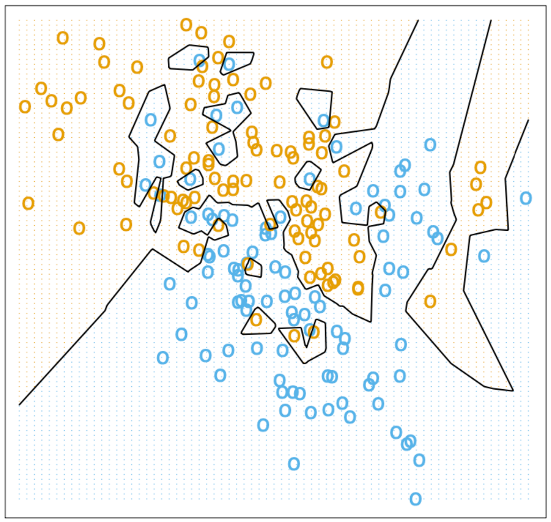
CS 591 S1 Spring 2025
Lecture 23

Prelude: Membership Inference

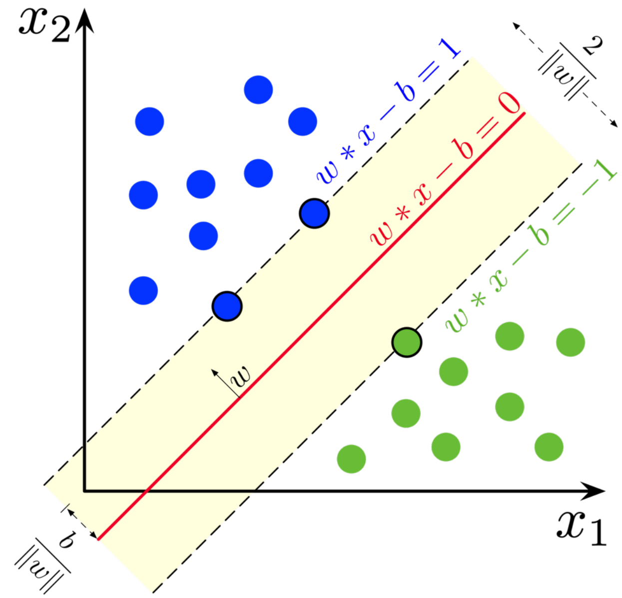
- Last time: robust membership inference in mean estimation
- More general setup:
  - $X_1, \dots, X_n \sim_{i.i.d.} P$
  - $\widehat{W} \leftarrow A(X_1, \dots, X_n)$
  - Goal is to approximately minimize population loss
    $$L_P(w) = \mathbb{E}_{X \sim P}\big(\ell(w; X)\big)$$
    (or empirical loss)
  - Test is given $(\widehat{W}, Y)$
    where Y is either X_I for $I \sim [n]$ or $X_0 \sim P$.
- Some test statistics $T(\widehat{w}, y)$ …
  - Loss: $\ell(\widehat{w}; y)$
  - Score: $\langle \nabla \ell(w^*; y), \widehat{w} - w^* \rangle$ where $w^* = \arg\min_w L_P(w)$
  - Gradient: $\langle \nabla \ell(\widehat{w}; y), \widehat{w} - w^* \rangle$
- Plots for regularized linear regression:
  https://colab.research.google.com/drive/1UTOIfHvL3qKSp8UrJMpwqsTjSBuEzF7b#scrollTo=197a8284-63a3-405a-af75-2dbf942723cf

Machine learning models contain
unnecessary, irrelevant
information about their training data.

Memorization can be explicit…

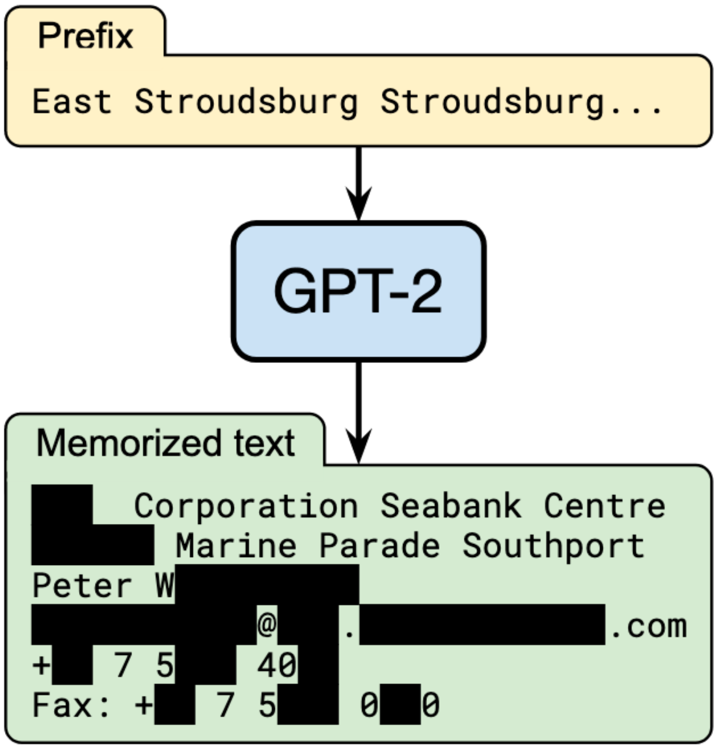


Hastie, Tibshirani, and Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

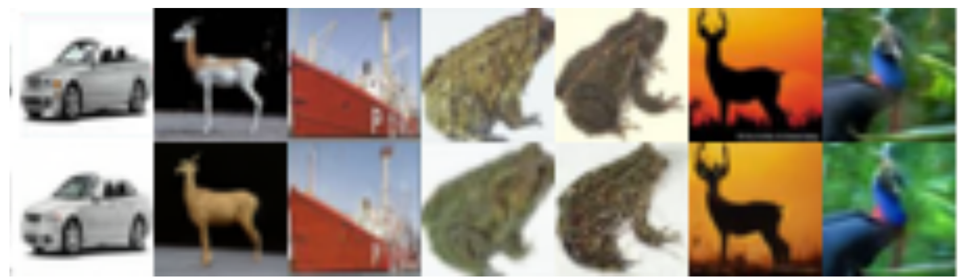


Wikipedia, *Support vector machine* (20 August 2020)

… but commonly an unintended side effect



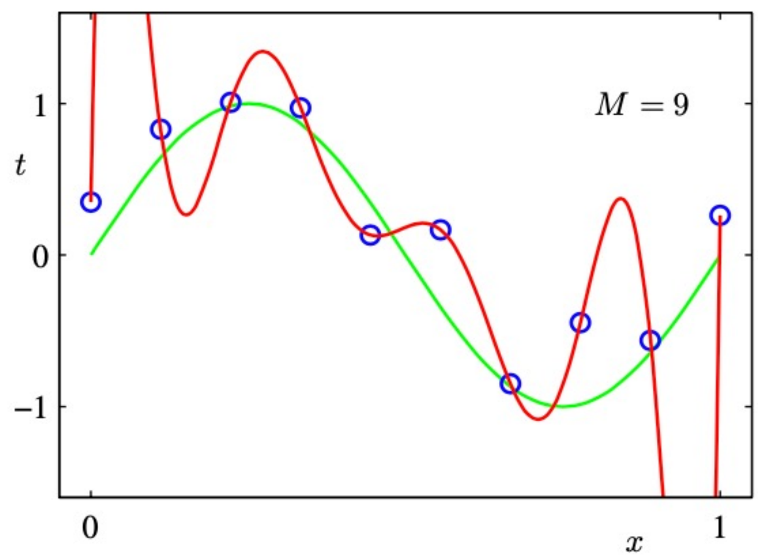
[Carlini et al. 20]
Current language models memorize irrelevant information.



Extraction from
Stable Diffusion v1.4

N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramèr, B. Balle, D. Ippolito, E. Wallace
Extracting Training Data from Diffusion Models (2023)

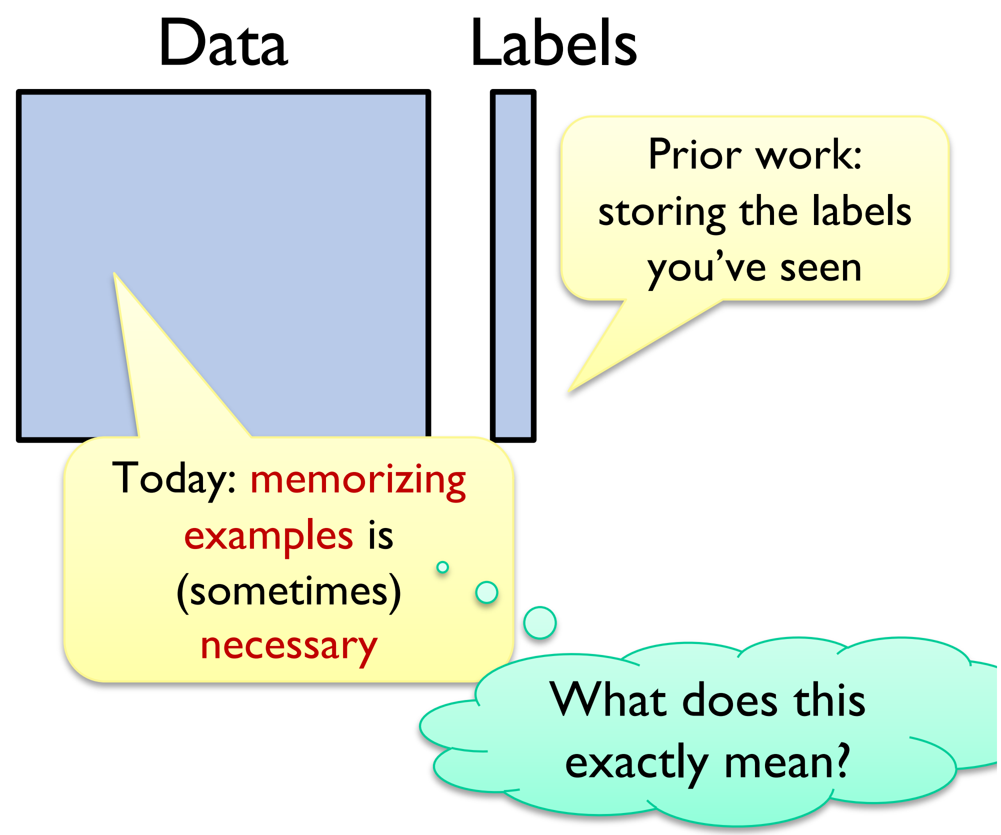
Memorization ≠ fitting or interpolation



Christopher Bishop, "Pattern Recognition and Machine Learning," 2006

[Feldman 20]

In natural problems, memorization of training **labels** is necessary no matter the learning algorithm.



Why Does Memorization Matter?

- Privacy
  - ➢ Adam D. Smith's SSN is 123-45-6789
  - ➢ Models are trained on
    - Your phone's photos
    - Your email and text messages
    - Your web browsing habits
    - Your social media posts

WHEN YOU TRAIN PREDICTIVE MODELS ON INPUT FROM YOUR USERS, IT CAN LEAK INFORMATION IN UNEXPECTED WAYS.

- Copyright
  - ➢ Huge issue currently: LLMs are trained on web data
  - ➢ Several law suits in progress
    - GitHub's Copilot leaked code from textbooks (link)
    - Some of Samsung's code was leaked after engineers fed it to ChatGPT in prompts

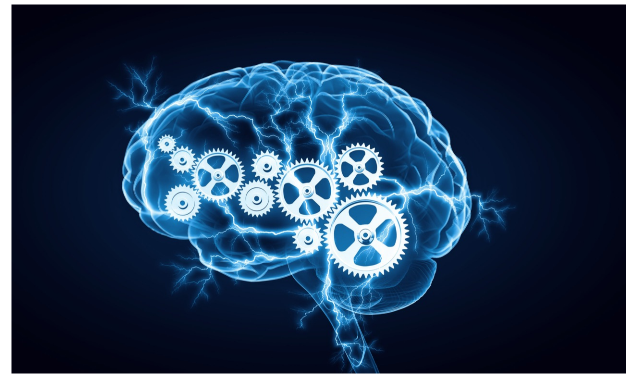
Why Does Memorization Matter?

- Models are getting bigger!
  - ➢ More opportunities for memorization

[Villalobos et al, July 2022]

Why Does Memorization Matter?

- The role of memory,
  and how we generalize from it,
  is central to cognition



- Kids (often) memorize first, then understand later
  - E.g. "here you go" is learned as "heego"

https://myheartland.net/classroom-applications-of-cognitive-psychology/
https://theweeklychallenger.com/three-key-benefits-black-children-find-in-early-childhood-education/

Machine learning models contain information about their training data.

Today: Two Results

- In natural settings,
  memorization of nearly entire examples is
  necessary for every learning algorithm,
  even when many details are irrelevant.

- In natural settings,
  every one-pass learning algorithm requires
  large memory because of memorization

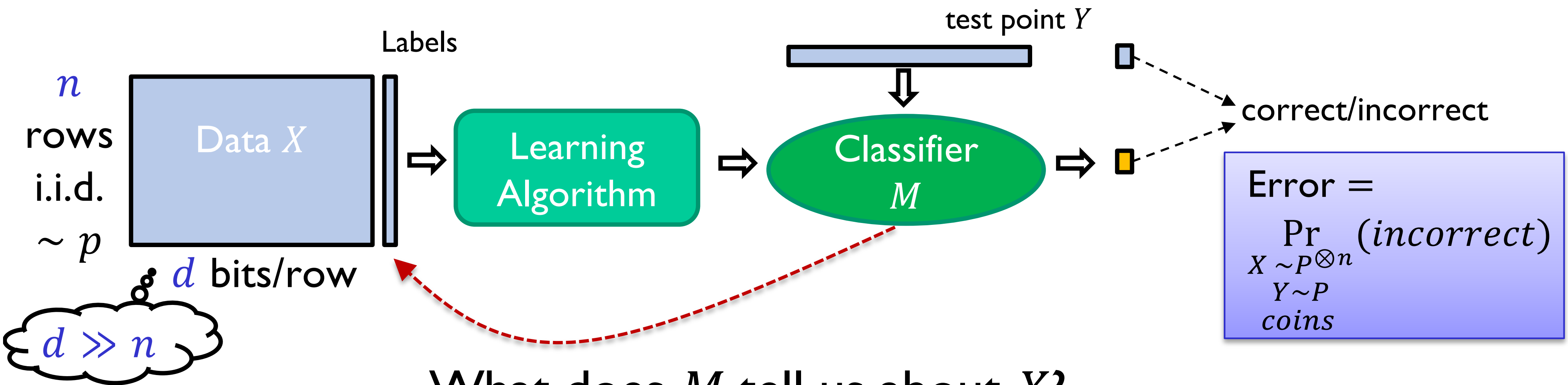
Machine learning models contain information about their training data.

Today: Two Results

- In natural settings,
  memorization of nearly entire examples is necessary for every learning algorithm,
  even when many details are irrelevant.

- In natural settings,
  every one-pass learning algorithm requires large memory because of memorization

Main Result: First Pass



What does M tell us about X?

Main result (roughly): There are natural tasks such that for every learning algorithm with error within 0.1 of optimal,

$$I(X; M) \geq c \cdot n \cdot d.$$

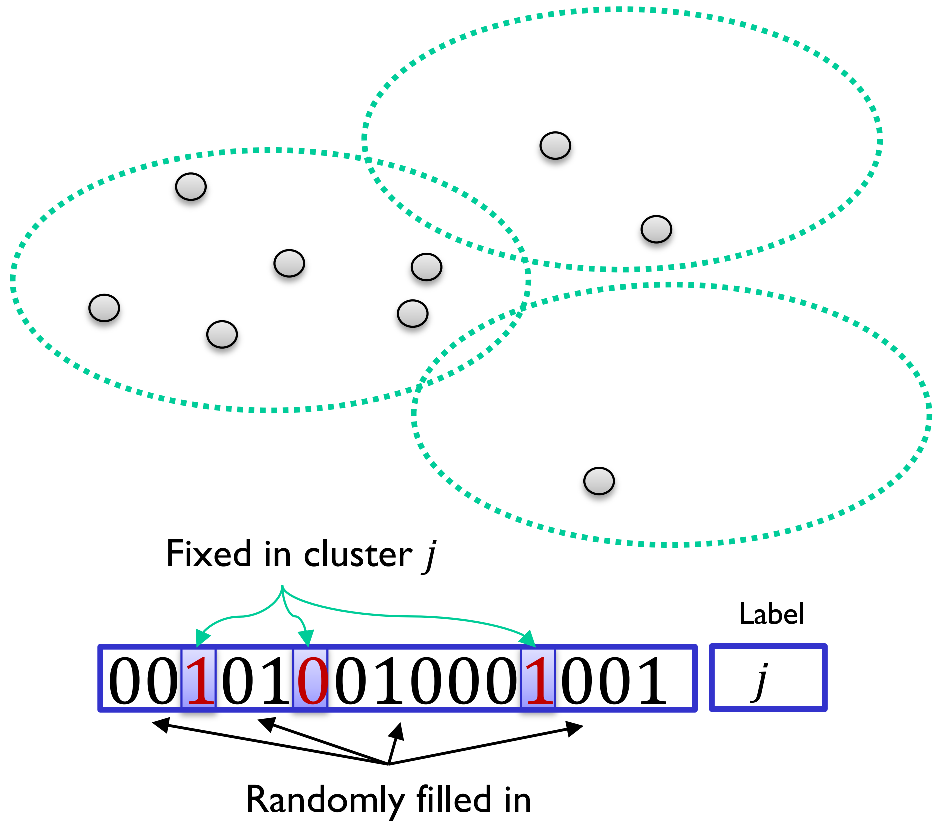
$n \cdot d$ bits

Mutual information $I(A; B)$ of two random variables A, B measures how much info B has about A (in bits)

Example task: hypercube cluster labeling

Data come from "clusters"

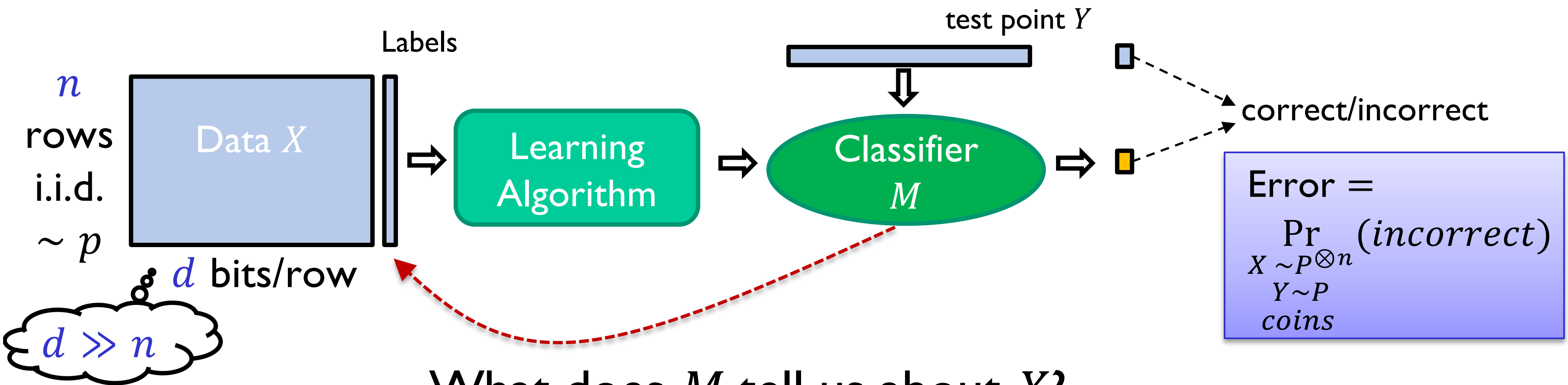
- Hypercube cluster:
  - Sparse set of fixed bits
  - Other bits filled in at random
- Points from the same cluster are closer than random



Learning task: identify which cluster generated input point

- Lots of examples from cluster ⇒ learn fixed bits
- Few examples ⇒ cannot discern which bits matter

Main Result: First Pass (Again)

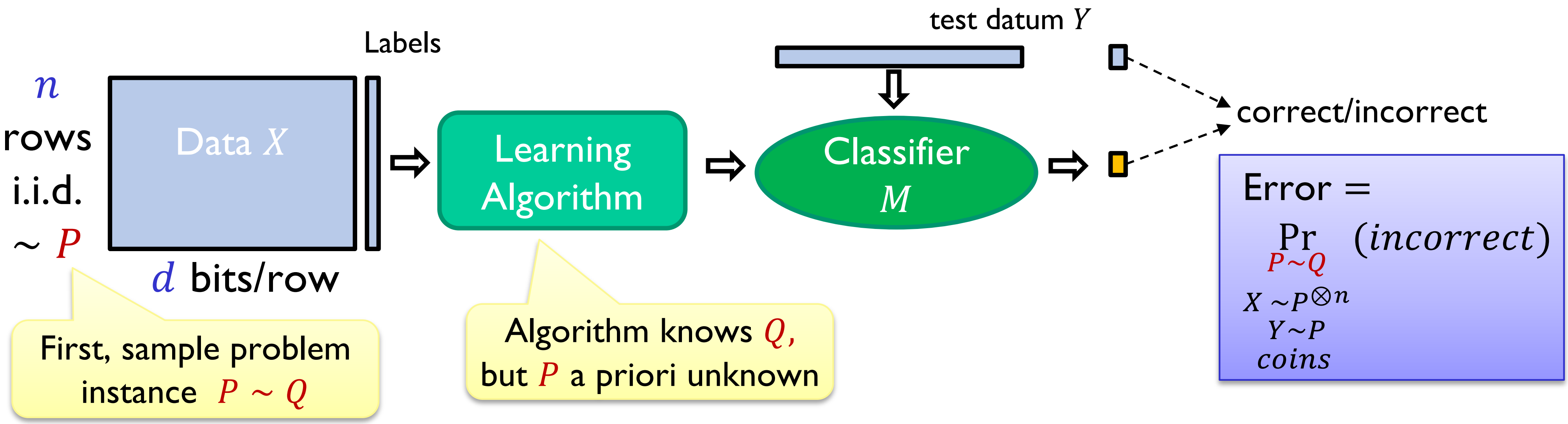


What does M tell us about X?

Main result (roughly): There are natural tasks such that for every learning algorithm with error within 0.1 of optimal,

$$I(X; M) \geq c \cdot n \cdot d.$$

Main Result: Irrelevant Information



Theorem: There is a natural task $Q_{n,d}$ such that for every learning algorithm with error $\leq OPT_{n,d} + 0.1$,

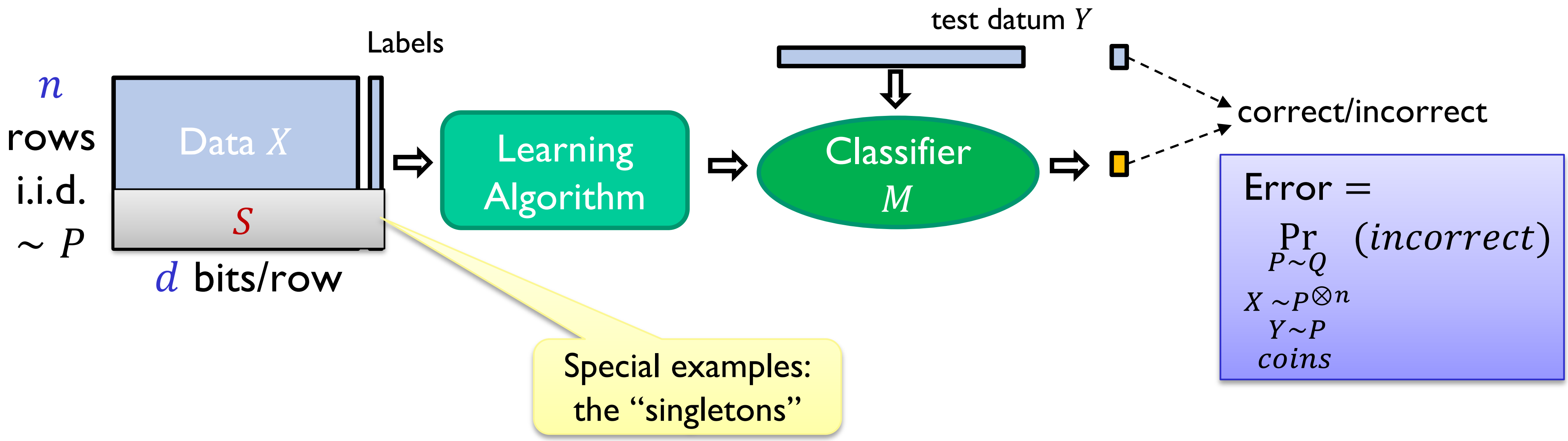
$$I(X; M \mid P) \geq c \cdot n \cdot d,$$

M stores irrelevant info

where

- p, a problem instance, is a distribution on labeled examples
- P, a random variable, is chosen from meta-distribution $Q_{n,d}$
- $OPT_{n,d}$ is the best possible average error for $Q_{n,d}$

Main Result: Memorizing Whole Examples



Theorem: There is a natural problem $Q_{n,d}$ for which every data set X has a subset of rows $S \subseteq X$ such that

Lots of "singletons"

- $|S| \geq c \cdot n$ with high probability
- For every learning algorithm with error $OPT_{n,d} + \epsilon$,

$$I(S; M|P) \geq d \cdot |S| \cdot \left(1 - f(\epsilon)\right)$$

where $f(\epsilon) \to 0$ as $\epsilon \to 0$.

M stores everything about the examples in S. Even though irrelevant.

Related Work

- Models must memorize labels [Feldman 20]
- Learners that leak little information
  [Bassily, Moran, Nachum, Shafer, Yehudayoff 18, Nachum, Shafer, Yehudayoff 18]
  - Lower bounds for PAC-Bayes framework [Livni, Moran 20]

Lower bound is $\Omega(n \log \log d)$ and $O(n \log d)$

- Representation complexity
  [Beimel, Nissim, Stemmer 13, Feldman, Xiao 14]
  - Lower bound on models' size for a class C
  - Reflects information about P, not just X

Lower bounds on $I(P; M)$, not $I(X; M|P)$

- Time-space tradeoffs for learning
  - [Shamir 14, …, Raz 18, …] Lower bounds for streaming model
  - Our results are on size of final model
    - To compete with best learner on n points, need large model
- More:
  - Information bottlenecks,
  - Minimum description length
  - …

Understanding good generalization

Common explanations for large models

1. Expressivity
2. Optimization is easier
3. Implicit regularization leads to good generalization

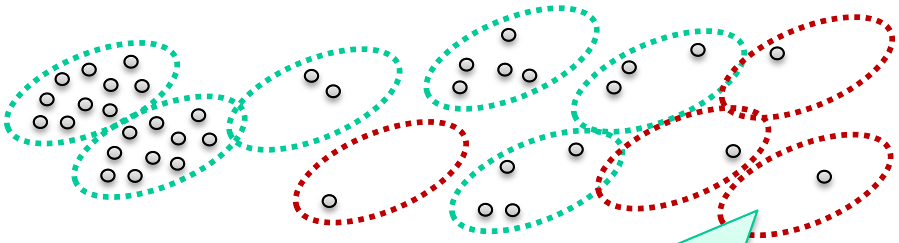
Our results suggest an additional factor:

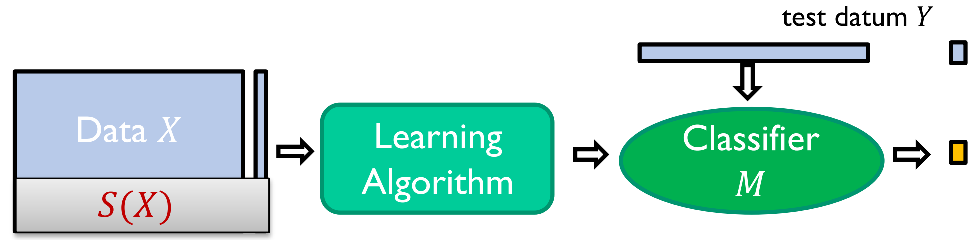
4. Large models store information whose usefulness isn't yet "understood"
   - Small subpopulations
   - Adapting to new domains

This work

Lower bounds by looking at singletons

Singletons setup from [Feldman 20]





Special examples $S(X)$, the "singletons"

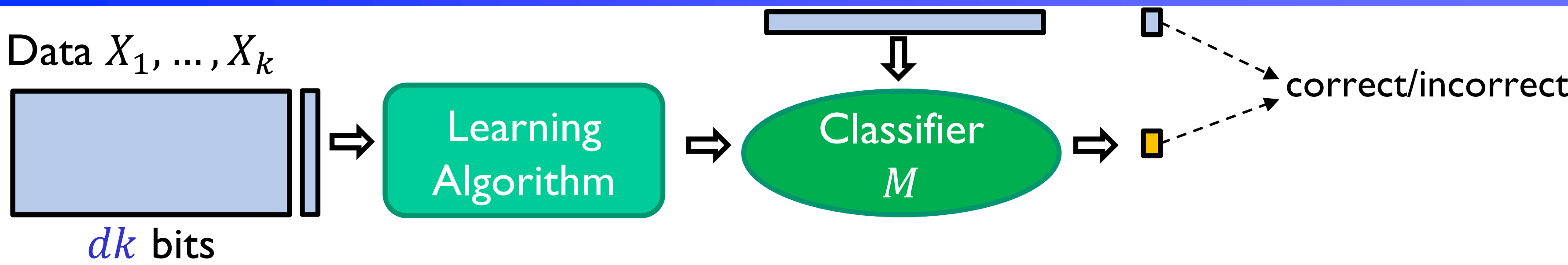
Typical data set has $\Omega(n)$ singletons. Proof strategy:

1. Accurate overall $\Rightarrow$ Accurate when test sample comes from singleton subpopulation

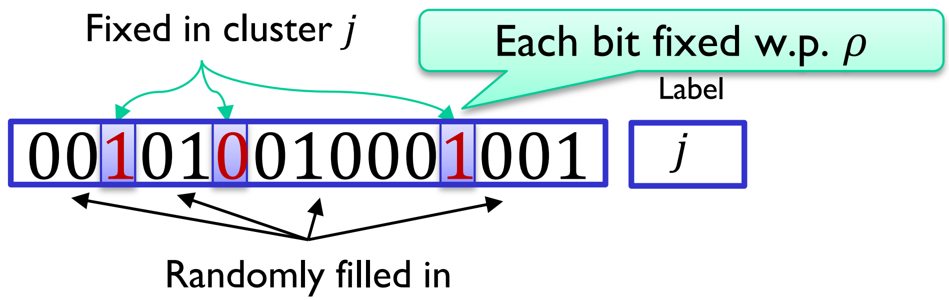
2. Accurate on singletons $\Rightarrow$ Reveal near-complete information about singletons

Via (new) information complexity lower bounds for "one-shot" learning

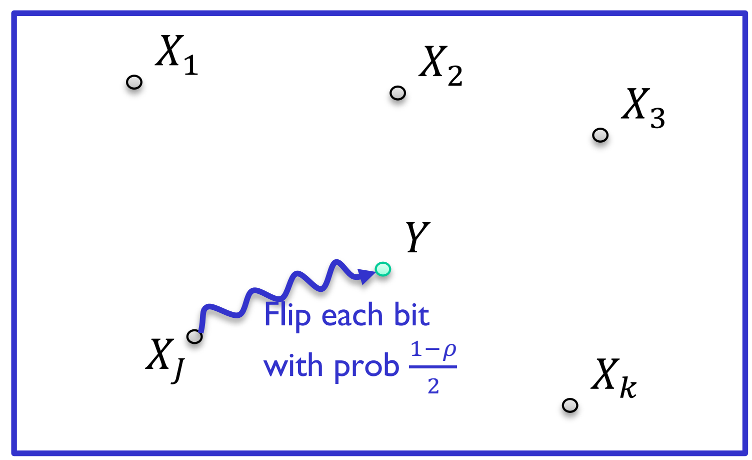
From hypercube clusters to 1-out-of-k NN



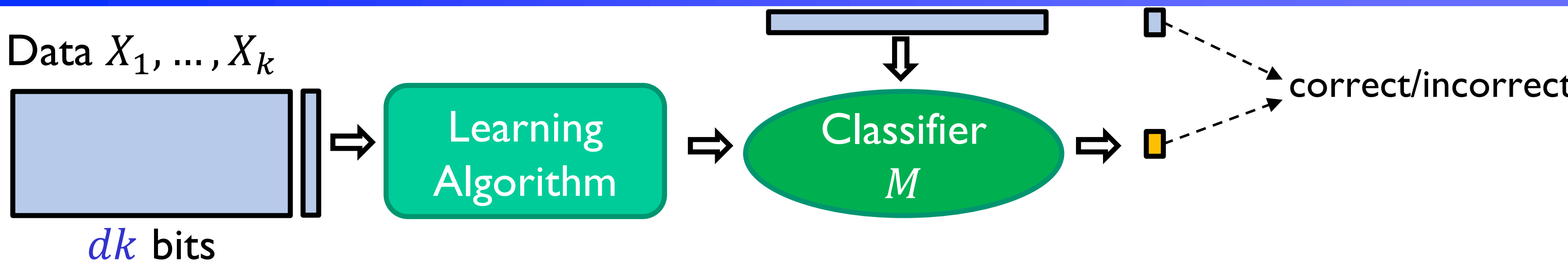
Per-cluster distribution:



- **Game:** Alice gets only k singletons $X_1, \ldots, X_k$
  - $X_j \sim P_j$
  - Bob gets a sample from cluster $J \in_R [k]$
- **Equivalently,**
  - Alice gets uniform in $\left(\{0,1\}^d\right)^k$
  - Bob data distributed as $Y \sim BSC_{\frac{1-\rho}{2}}(X_J)$ for uniform $J \in [k]$.
  - Bob must guess J
- **Corollary:** Alice and Bob's best strategy solves nearest neighbor.



From hypercube clusters to 1-out-of-k NN



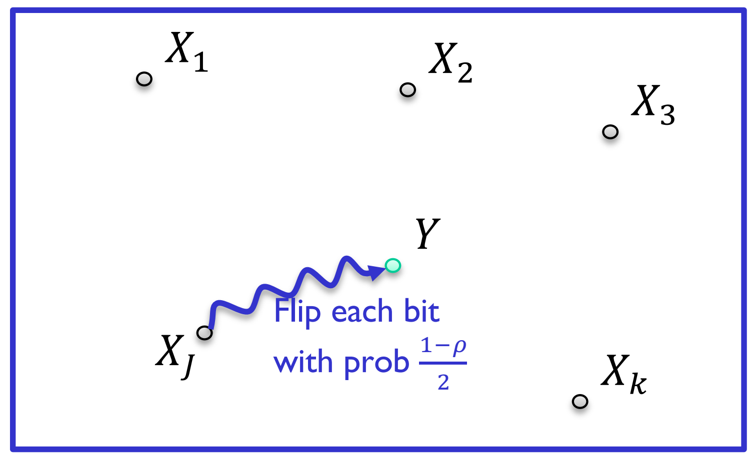
Lower bounds for nearest neighbor

Theorem: If Bob succeeds w.p. $OPT - o(1)$,

$$I(X; M) = \left(\frac{1}{2 \ln 2} - o(1)\right) kd$$

for appropriate ρ.

- Use Strong Data Prcoessing Inequality as in [Hadar, Liu, Polyanskiy, Shayevitz 19]

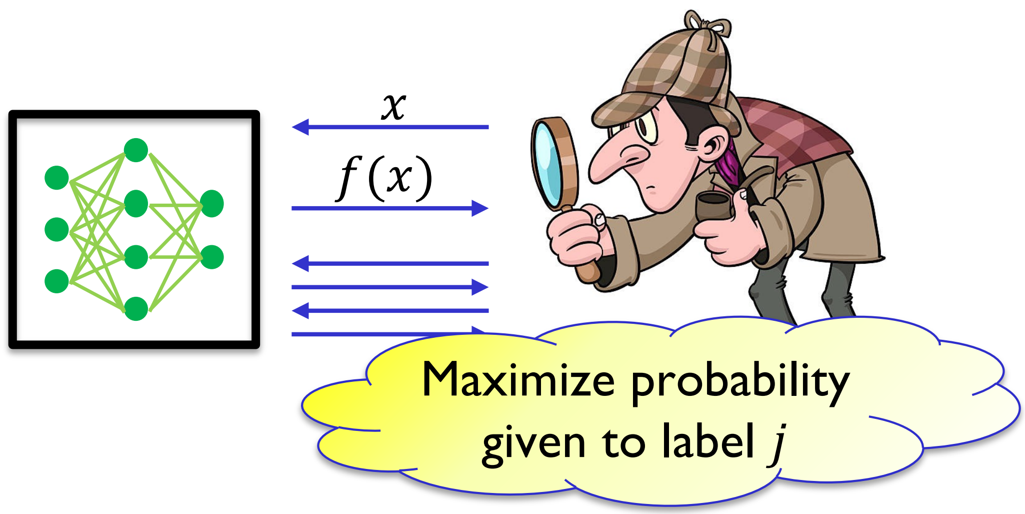


Conjecture: If Bob succeeds w.p. $OPT - o(1)$,

$$I(X; M) = \big(1 - o(1)\big) kd.$$

Trying out an attack

- Generate data from hypercube clusters learning task
  - Recall: each subpopulation has one label
- Learners:
  - Multiclass logistic regression
  - Multilayer perceptron



- Adversary gets:
  - Query access to model
  - "Label j was a singleton"
- Adversary strategy: maximize probability of label j
- Look at error vs training time

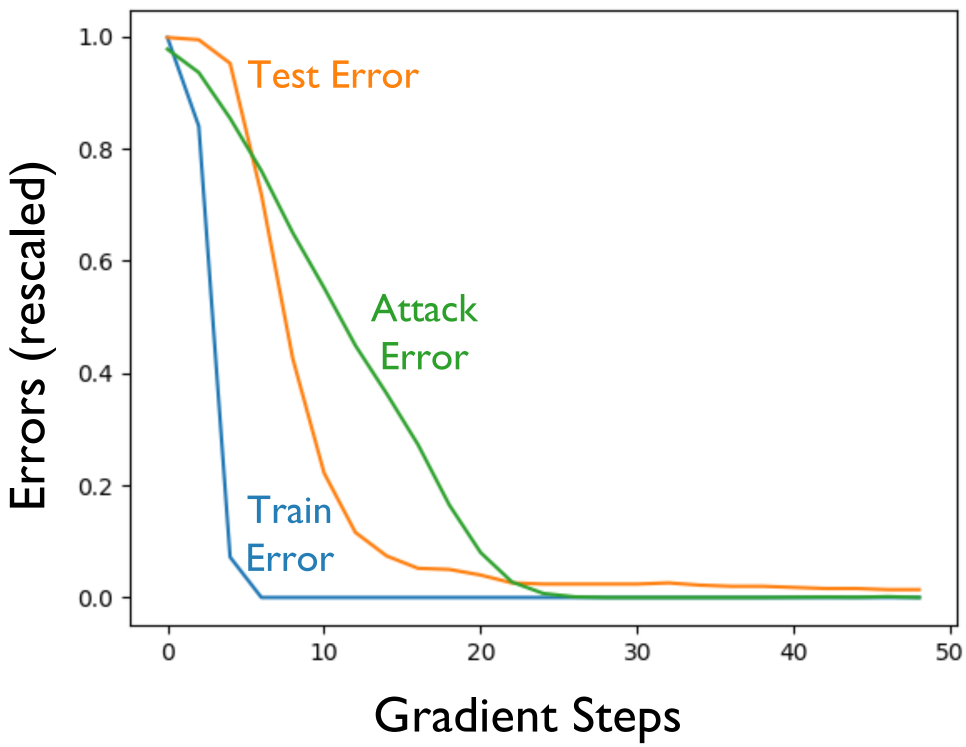
Logistic Regression and Neural Network

Experimental Setup:

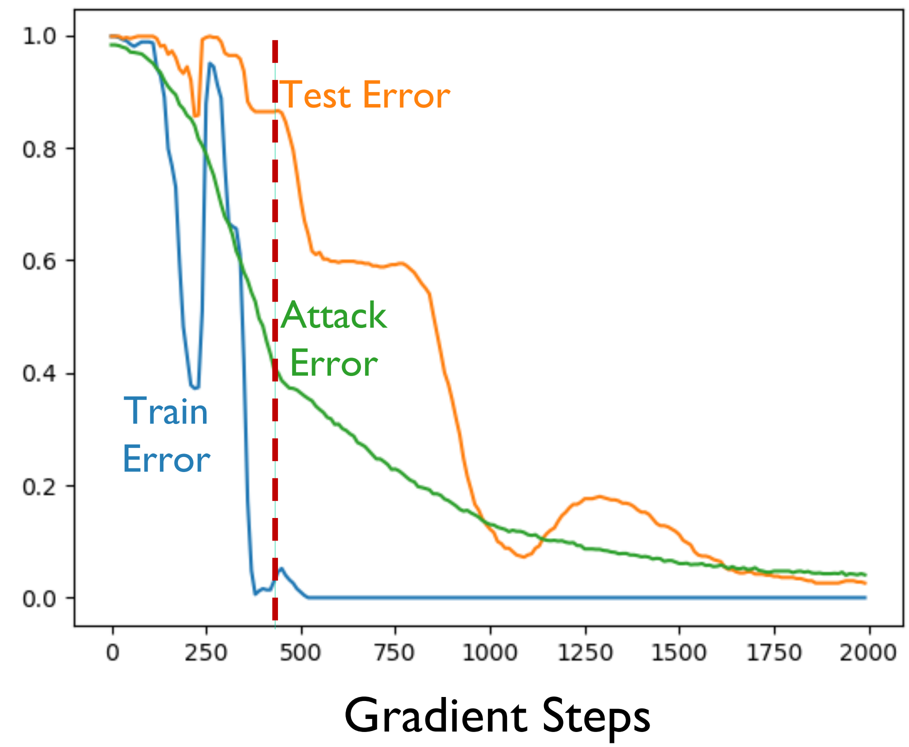
- $n = 500$ examples
- $d = 1000$ bits/example
- $\rho \approx 25\%$ of bits fixed
- Train with gradient descent

In both cases, recover
$\geq$ 98% of the singletons' bits

Multiclass
Logistic Regression



Multilayer Perceptron
(Single Hidden Layer)



Machine learning models contain information about their training data.

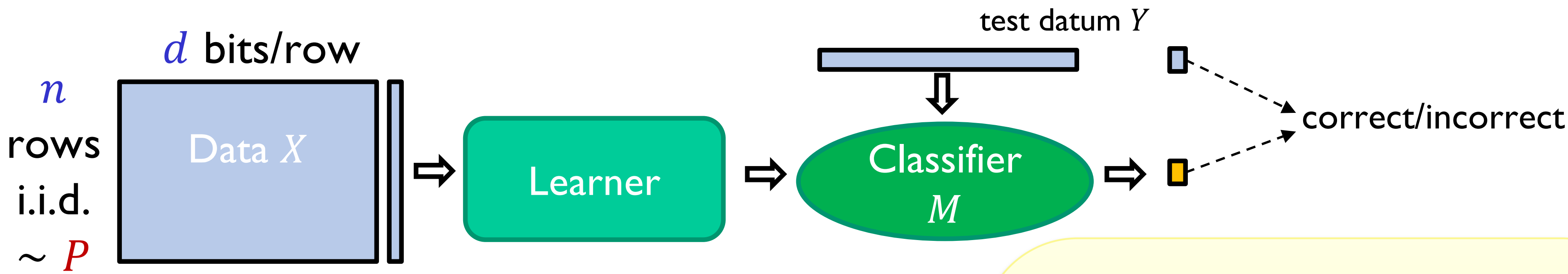
Today

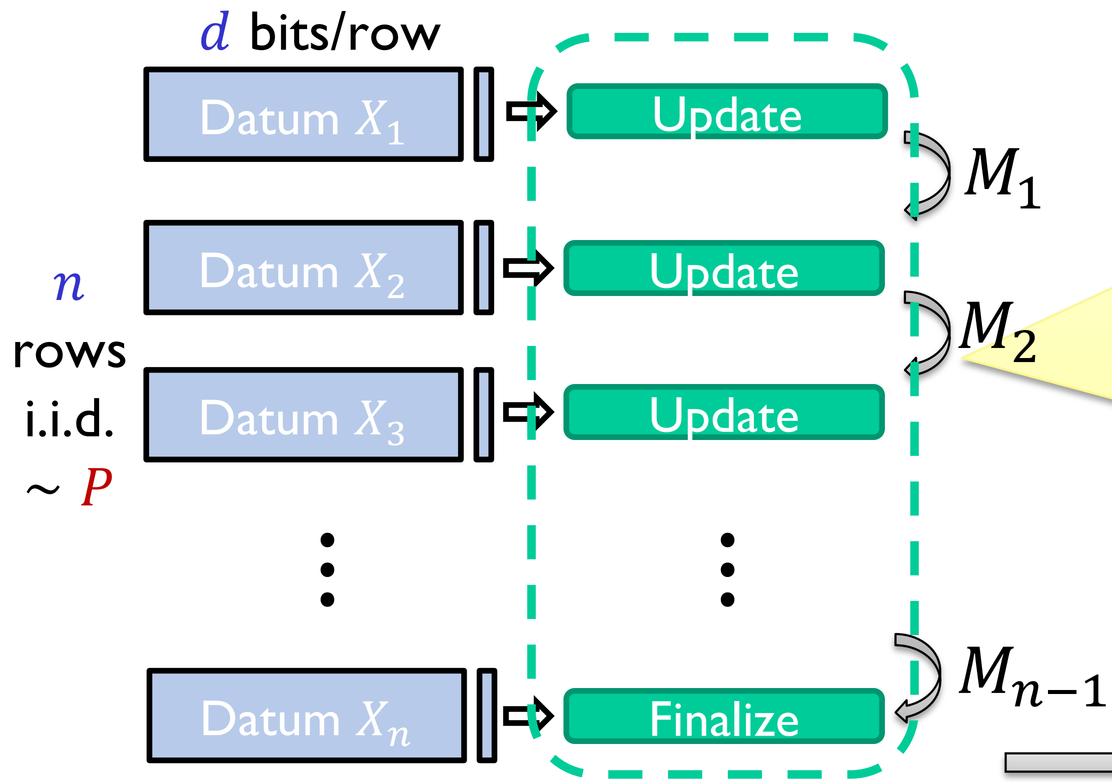
- In natural settings,
  memorization of nearly entire examples is
  necessary for every learning algorithm,
  even when many details are irrelevant.

- In natural settings,
  every one-pass learning algorithm requires
  large memory because of memorization

(The remaining material was not covered in class)

Streaming Model of Learning [1990's]

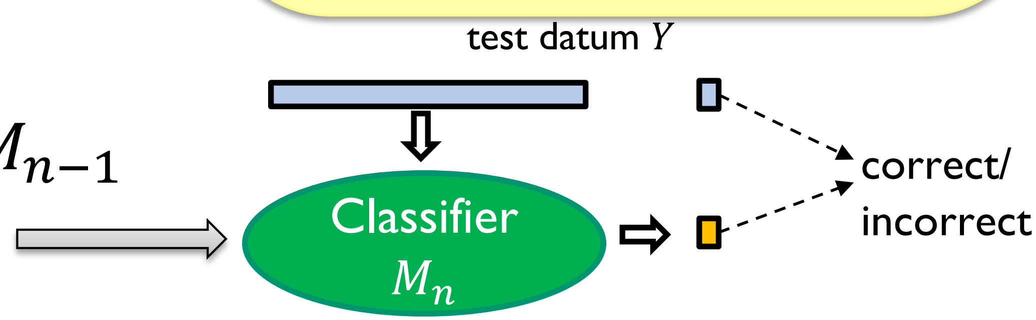




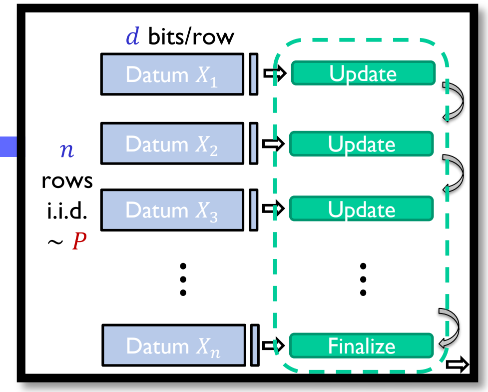
Cost driven by

- Time to update
- Size (in bits) of intermediate states $M_1, M_2, \ldots, M_n$

Intermediate states arbitrary (not classifiers)



How Much Space?

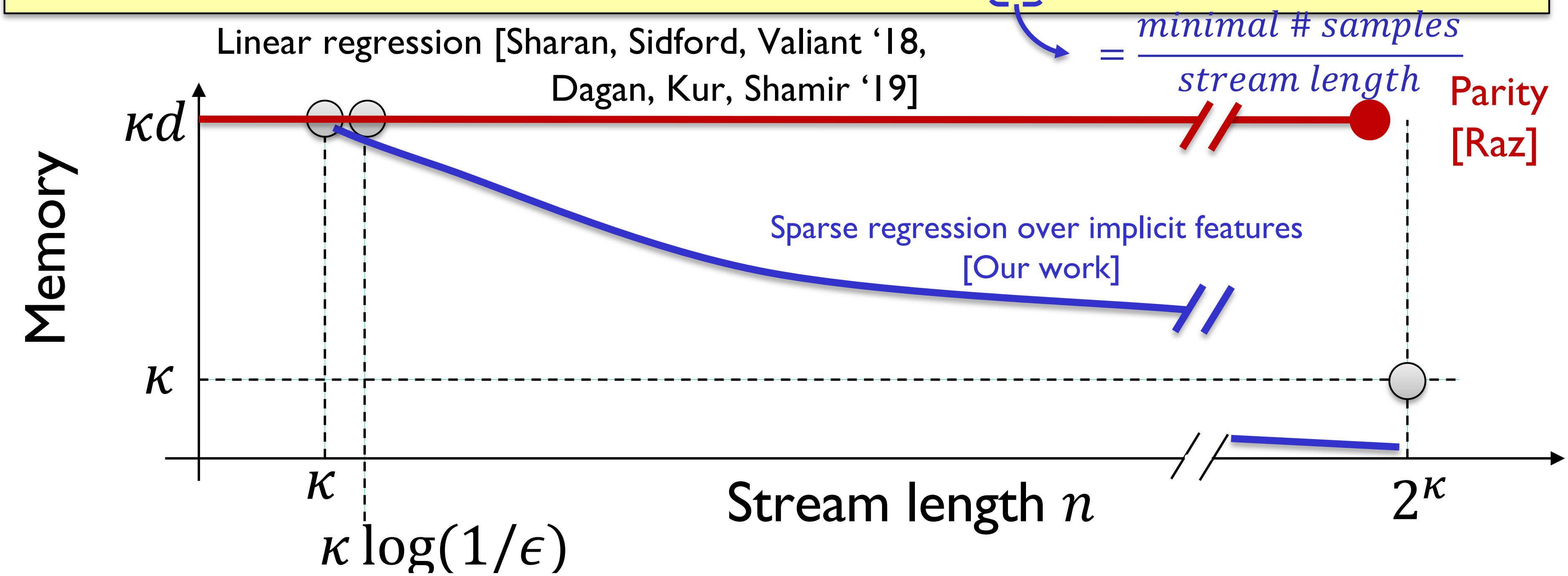


Goal: Understand space as a function of…

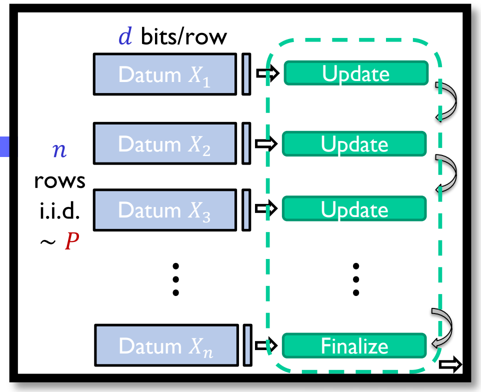
- d: data dimension (examples in $\{0,1\}^d$)
- κ: size of a "good" model
- n: stream length

Our work:

- Natural sparse regression problems for which nontrivial prediction requires memory $\tilde{\Omega}\left(d\kappa \cdot \boxed{\frac{\kappa}{n}}\right)$ for arbitrary d, κ.



How Much Space?



Goal: Understand space as a function of…

- d: data dimension (examples in $\{0,1\}^d$)
- κ: size of a "good" model
- n: stream length

Our work:

- Natural sparse regression problems for which nontrivial prediction requires memory $\Omega\left(d\kappa \cdot \frac{\kappa}{n}\right)$ for arbitrary d, κ.

How "natural"? Examples include

- Multi-class sparse logistic regression
- Binary $\frac{\kappa}{\log d}$-sparse logistic regression over degree-2 polynomial features
  - ➢ Space usage scales with ambient dimension, not model or example size

Machine learning models contain information about their training data.

Today

- In natural settings,
  memorization of nearly entire examples is necessary for every learning algorithm,
  even when many details are irrelevant.

- In natural settings,
  every one-pass learning algorithm requires large memory because of memorization

Future Work

- Models reflecting complex real world
  - In our case, “true” model is sparse multiclass logistic regression
- Algorithms that provably extract memorized info
- Connections between memorization and
  - Further data/memory tradeoffs
  - Limited-access data models
  - PAC-Bayes obstacles
- Does faster optimization “require” more memorization?