6 The Necessity of Memorizing Irrelevant Data

In this section, we will discuss cases where any accurate model must memorize *both features and labels*. The presentation is inspired by the work of Brown, Bun, Feldman, Smith, and Talwar [BBFST21] and the subsequent work of Feldman, Kornowski, and Lyu [FKL25].

6.1 Information Theory 101

Despite our long discussion of what should be the notion of label memorization, one part we didn't debate much is that we should be defining memorization in terms of how easy it is to guess the label. But asking someone to guess a feature vector is a really tall order! If our feature vectors $x \in \{0,1\}^d$ consist of d uniformly random bits, then the probability of guessing one without any hint is 2^{-d}. If d is 1000, then 2^{-1000} is unthinkably small. Now suppose the model memorizes and allows you to extract the first $19d/20$ bits of the vector, and the remainder are completely random. Intuitively we would say that the model memorized almost all of the feature vector. But the probability of guessing the remaining bits exactly would still be $2^{-d/20}$. For $d = 1000$, this probability is still 2^{-50}, which is also incredibly small.

What we need is a more flexible way to measure the amount of information that a model memorizes. We will do so using the language of *information theory*.

Definition 6.1 (Entropy and mutual information)**.** For discrete random variables U, V, W, write $p_U(u) = \mathbb{P}[U = u]$. The *entropy* of U is

$$H(U) = \sum_u p_U(u) \log_2(1/p_U(u)),$$

where $0 \log_2 0 = 0$ by convention. Its *conditional entropy* given V is

$$H(U \mid V) = \mathop{\mathbb{E}}_{v \sim V}[H(U \mid V = v)],$$

where $H(U \mid V = v)$ is the entropy of the conditional distribution of U given $V = v$. The *mutual information* and *conditional mutual information* are

$$I(U;V) = H(U) - H(U \mid V),$$
$$I(U;V \mid W) = H(U \mid W) - H(U \mid V, W).$$

All these quantities are measured in bits.

Entropy measures uncertainty, and mutual information measures how much observing one variable reduces uncertainty about another.

Example 6.2 (Revealing 950 of 1000 bits)**.** Let $U = (U_1, \ldots, U_{1000})$ be uniformly random in $\{0,1\}^{1000}$, and let $V = (U_1, \ldots, U_{950})$. There are 2^{1000} equally likely values of U, so $H(U) = 1000$. After observing V, the remaining 50 bits are still independent uniform bits. Thus

$$H(U \mid V) = 50, \qquad I(U;V) = 1000 - 50 = 950.$$

The output reveals 95% of the entropy of U.

For memorization, we measure

$$I(S; \hat{\theta} \mid \mathcal{D}) = H(S \mid \mathcal{D}) - H(S \mid \hat{\theta}, \mathcal{D}),$$

the information the model reveals about the sample even to someone who already knows $\mathcal{D}$.

Note that the concept of mutual information doesn't imply anything about how easy it is to extract information from the output. Just becasue $H(U|V)$ is small, doesn't say anything about how you should learn about U given V. In our story about LLMs, this might correspond to the situation where there exists some prompt that will convince the model to output my phone number but you the user have no idea what prompt to give.

Separating information about the distribution from information about the sample. A model can contain information about its training data simply because it has learned the underlying distribution. We will separate out this possibility in two steps: first prove that the model reveals a lot about the sample, then show that the distribution itself has too little entropy to account for most of that information. The following general inequality makes this argument precise.

Fact 6.3 (Subtracting shared information)**.** *For discrete random variables U, V, W with finite entropies,*

$$I(U; V \mid W) \geq I(U; V) - I(U; W) \geq I(U; V) - H(W).$$

Proof. The chain rule expands the information that the pair (V, W) reveals about U in either order:

$$I(U; V) + I(U; W \mid V) = I(U; (V, W)) = I(U; W) + I(U; V \mid W).$$

Rearrange and use $I(U; W \mid V) \geq 0$, followed by $I(U; W) \leq H(W)$. □

Apply this with $U = S$ the training sample, $V = \mathcal{A}(S)$ the trained model, and $W = \mathcal{D}$ the randomly chosen data distribution. If

$$I(S; \mathcal{A}(S)) \geq B \qquad \text{and} \qquad H(\mathcal{D}) \leq b,$$

then

$$I(S; \mathcal{A}(S) \mid \mathcal{D}) \geq B - b.$$

For example, if the model retains at least 900 bits about the sample, but the distribution itself has at most 100 bits of entropy, then at least 800 bits remain even after the distribution is revealed. No observation can reveal more information about the distribution than its entropy. The remaining information must therefore describe the particular sample beyond what the distribution tells us.

6.2 Two hard learning problems

Draw a distribution $\mathcal{D}$ from a prior Q, then draw $S \sim \mathcal{D}^n$ and output $\hat{\theta} = \mathcal{A}(S)$. The learner knows Q, but not $\mathcal{D}$. Define

$$\ell_n(\mathcal{A}) = \mathop{\mathbb{E}}_{\mathcal{D} \sim Q,\, S \sim \mathcal{D}^n,\, \hat{\theta} \sim \mathcal{A}(S)} [\ell(\hat{\theta}; \mathcal{D})], \qquad \text{opt}_n = \inf_{\mathcal{A}} \ell_n(\mathcal{A}).$$

The benchmark uses the same n samples and prior, without knowing $\mathcal{D}$.

There are two different claims we could make about $\mathcal{A}(S)$:

1. *Retaining training information:* $I(S; \mathcal{A}(S)) = \Omega(nd)$. This counts everything the model learns from the sample, including information about the underlying distribution.

2. *Retaining sample-specific information:* $I(S; \mathcal{A}(S) \mid \mathcal{D}) = \Omega(nd)$. This counts information about the particular sample that remains even after the distribution is revealed.

In both claims, the learner receives only S. Conditioning on $\mathcal{D}$ gives the distribution to the observer measuring memorization, *not to the learner.* If the learner knew $\mathcal{D}$, it could ignore S and output an optimal classifier determined by $\mathcal{D}$, giving $I(S; \mathcal{A}(S) \mid \mathcal{D}) = 0$. The theorem establishes the second, stronger claim [BBFST21, Section 1.1].

Theorem 6.4 (Necessity of feature memorization, informal [BBFST21, Theorem 1.1])**.** *There are families of learning problems, with arbitrarily large sample size n and dimension d, for which*

$$H(S \mid \mathcal{D}) = \Theta(nd) \qquad \text{and} \qquad H(S) = \Theta(nd),$$

and constants $C, \alpha_0 > 0$ such that every possibly randomized learner $\mathcal{A}$ satisfies

$$\ell_n(\mathcal{A}) \le \mathrm{opt}_n + \alpha_0 \quad \Longrightarrow \quad I(S; \mathcal{A}(S) \mid \mathcal{D}) \ge Cnd.$$

6.2.1 Hypercube cluster labeling

We start with a construction using n equally likely clusters in $\{0,1\}^d$ [BBFST21, Section 4]. For each cluster j, independently draw a hidden string $s_j \in \{0,1\}^d$ uniformly and a hidden set $T_j \subseteq \{1, \dots, d\}$, including each coordinate independently with probability ρ. These choices determine $\mathcal{D}$. To sample an example, choose J uniformly from $\{1, \dots, n\}$ and set

$$Y = J, \qquad X_i = \begin{cases} s_{J,i}, & i \in T_J, \\ b_i, \quad b_i \sim \mathrm{Ber}(1/2), & i \notin T_J. \end{cases}$$

The noise bits are independent and freshly sampled for each example. The learner receives labeled examples and must predict a fresh example's cluster.

The coordinates in T_j define cluster j and the remaining coordinates are random noise (Figure 5). With only one example from a cluster, the learner cannot tell which coordinates are fixed. Yet matching a test point against that example can help identify its cluster. There are about n/e singleton examples in expectation.

Why must the learner retain noise? A single observed bit is equally likely to be 0 or 1 whether its coordinate is fixed or random. Indeed, after observing one example (x, j),

$$\mathbb{P}[i \in T_j \mid X = x, Y = j] = \rho.$$

So the observation does not identify the useful coordinates. Keeping a coordinate might preserve a fixed bit that helps recognize the cluster, or a noise bit that tells us nothing about future examples, but the learner does not know which one is which.

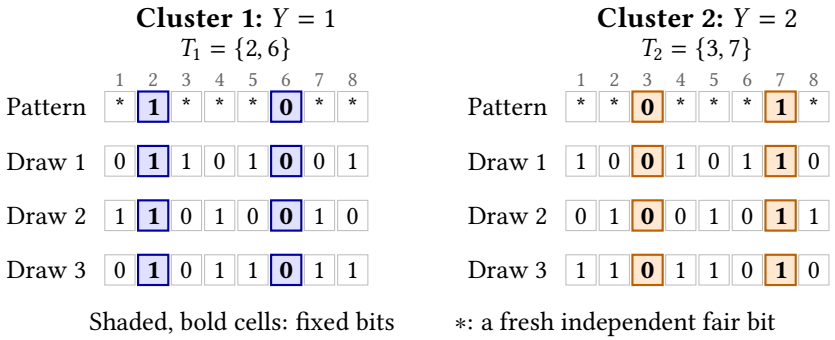


Figure 5: Three possible draws from each of two clusters with $d = 8$. The highlighted coordinates stay fixed within a cluster; the other bits are sampled afresh and may agree by chance. The learner sees the bits and training labels, but not the hidden patterns or highlighting. A single example therefore does not reveal which coordinates define its cluster.

For $d \geq n$ and a suitable choice

$$\rho = \Theta\left(\sqrt{\frac{\log n}{d}}\right),$$

the paper proves

$$\ell_n(\mathcal{A}) \leq \mathrm{opt}_n + \alpha_0 \quad \Longrightarrow \quad I(S; \mathcal{A}(S) \mid \mathcal{D}) \geq \Omega(nd)$$

for some constant $\alpha_0 > 0$ [BBFST21, Corollary 4.1].

We can explain this conclusion in two steps. First, the difficult part of the proof shows that accurate prediction requires the model $\mathcal{A}(S)$ to retain a lot of information about the sample:

$$I(S; \mathcal{A}(S)) = \Omega(nd).$$

The singleton examples are what drive this lower bound.

Second, the distribution itself has little entropy. To specify a cluster, we need to say which coordinates are fixed and give their values. The set T_j has entropy $dh_2(\rho)$, and the fixed values have conditional entropy $\mathbb{E}[|T_j|] = \rho d$, where

$$h_2(\rho) = -\rho \log_2 \rho - (1 - \rho) \log_2 (1 - \rho)$$

is binary entropy. The values of s_j outside T_j do not affect the distribution. Since the n clusters are independent and their mixture weights are fixed,

$$H(\mathcal{D}) = nd\big(h_2(\rho) + \rho\big) = o(nd),$$

because $\rho \to 0$ in our parameter regime. Applying Fact 6.3,

$$I(S; \mathcal{A}(S) \mid \mathcal{D}) \geq I(S; \mathcal{A}(S)) - H(\mathcal{D}) = \Omega(nd) - o(nd) = \Omega(nd).$$

Thus learning the distribution can account for only a vanishing fraction of the required information. Most of it must concern the particular training sample, including its random noise.

6.2.2 The noisy next-symbol prediction problem

A second hard problem comes from [BBFST21, Section 3]. Take $d \geq n$. Independently draw hidden reference strings $s_1, \ldots, s_n$ uniformly from $\{0,1\}^d$; these determine $\mathcal{D}$. To sample $(X, Y) \sim \mathcal{D}$, draw

$$J \sim \text{Uniform}(\{1, \ldots, n\}), \qquad L \sim \text{Uniform}(\{0, \ldots, d-1\}),$$

and independent noise bits $b_1, \ldots, b_{L+1} \sim \text{Ber}(1/4)$. Set

$$X = \big(J, (s_{J,1} \oplus b_1, \ldots, s_{J,L} \oplus b_L)\big), \qquad Y = s_{J,L+1} \oplus b_{L+1},$$

where $\oplus$ denotes addition modulo two. Each example reveals a noisy prefix and asks for the next noisy bit. Training and test examples use fresh independent noise, but the same reference strings.

The varying prefix lengths make training features useful for predicting future labels. A training prefix of length 6 contains noisy observations of bits $1, \ldots, 6$, and its label provides bit 7. A test prefix of length 2 asks for bit 3, while a test prefix of length 5 asks for bit 6. Both answers were previously observed as features. Since the test length varies, remembering only the training label is insufficient.

Why must the learner retain noise? Asingleton example does not reveal which bits were flipped so you do not know which bits to keep and which to discard. Memorizing a singleton record means writing down a purely random d bit string but only some of that information is relevant to the actual distribution.

For sufficiently small constant $\alpha_0 > 0$, the paper proves

$$\ell_n(\mathcal{A}) \leq \text{opt}_n + \alpha_0 \quad \Longrightarrow \quad I(S; \mathcal{A}(S) \mid \mathcal{D}) = \Omega(nd)$$

[BBFST21, Theorem 3.2]. We can again explain the conclusion in two steps, using the stronger guarantee when $\ell_n(\mathcal{A}) - \text{opt}_n = o(1)$.

First, accurate prediction forces the model to retain almost all the information in the singleton records S_{sing}:

$$I(S_{\text{sing}}; \mathcal{A}(S)) \geq H(S_{\text{sing}}) - o(nd), \qquad H(S_{\text{sing}}) = \Theta(nd).$$

The random test length is what forces the model to preserve information throughout each observed prefix [BBFST21, Sections 3.4–3.5].

Second, we bound how much of this information concerns the distribution. Here the n independent reference strings give $H(\mathcal{D}) = nd$, so the low-entropy shortcut used for clustering does not apply. Instead, each noisy bit reveals only

$$I(s_{j,i}; s_{j,i} \oplus b_i) = 1 - h_2(1/4) \approx 0.189$$

bits about its reference bit: the observed bit has entropy 1, and still has entropy $h_2(1/4)$ when the reference bit is known.