The Label Recorder Method
Testing the Memorization Capacity of Machine Learning Models

Kyle Rong*[0000−0002−1083−0708], Aditya Khant*[0000−0001−5437−7485], David Flores[0000−0001−5986−2690], and George D. Montañez[0000−0002−1333−4611]

AMISTAD Lab
Harvey Mudd College, Claremont CA, 91711, USA
{krong, akhant, deflores, gmontanez}@hmc.edu
* These authors contributed equally

Abstract. Highly-parameterized deep neural networks are known to have strong data-memorization capability, but does this ability to memorize random data also extend to simple standard learning methods with few parameters? Following recent work exploring memorization in deep learning, we investigate memorization in standard non-neural learning models through the label recorder method, which uses a model's training accuracy on randomized data to estimate its memorization ability, giving a distribution- and regularization-dependent label recording score. Label recording scores can be used to measure how capacity changes in response to regularization and other hyperparameter choices. This method is fully empirical, easy to implement, and works for all black-box classification methods. The label recording score supplements existing theoretical measures of model capacity such as Rademacher complexity and Vapnik-Chervonenkis (VC) dimension, while agreeing with conventional intuitions regarding statistical learning processes. We find that memorization ability is not limited to only over-parameterized models, but instead exists as a continuum, being present (to some degree) even in simple learning models with few parameters.

Keywords: Representational Capacity · Label Recorder · Label Recording Score

1 INTRODUCTION

Representational capacity of a model captures the ability of a model to fit various distributions [3, 6, 10]. A low-capacity model is incapable of fitting a complex training dataset, leading to underfitting and poor generalization. A high-capacity model can memorize a training dataset and overfit [1, 2]. Thus, representational capacity is central to understanding whether a model will generalize well given a training set, and is of general importance in machine learning research.

VC (Vapnik-Chervonenkis) dimension and Rademacher complexity are well-known theoretical measures of representational capacity. The VC dimension upper bounds the capacity of a model by considering the maximum size of a dataset for which the model can correctly draw decision boundaries that differentiate all possible label combinations [13]. In contrast, Rademacher complexity measures a model's capacity through its ability to fit random noise, gauging the maximum expected correlation that a model can create between its predicted labels and the true (random) labels. Rademacher complexity has been used to create algorithms for learning kernels [4] and to improve generalization performance of models trained with adversarial learning [15].

We take inspiration from these theoretical tools to present an empirical method for estimating representational capacity, called the *label recorder method*. In the spirit of Rademacher complexity, the label recorder method evaluates a learning algorithm's ability to fit noise and provides an empirical method for the same. It does this by training and evaluating a model on the same set of randomized data, hence the name *label recorder*, as this method trains the model to record labels of random data. We seek to measure the ability of a model to store and recall information from a random dataset, in which there is no exploitable regularity between features and labels to compress. In supervised learning, correct label prediction is the result of memorization, generalization, and luck, in some combination. By using data with no dependence between features and labels we remove the contribution of generalization, which allows us to probabilistically bound a method's information storage capacity for its models, a quantity that affects its generalization behavior [1, 2, 14]. We refer to the accuracy of the model in reproducing labels as its *label recording score*.

The label recorder method was first introduced by Sandoval Segura et al. [11]. They demonstrated how to directly estimate model capacity by proposing the label distribution matrix (LDM). For a model and some pre-selected datasets, the matrix computes the probabilities that the model predicts each possible labeling for each dataset and stores them in simplex vectors. It determines a probability distribution that can generate these simplex vectors and uses the entropy of the recovered distribution to infer the flexibility of a model. The more uniformly a model spreads its corresponding probability mass over the support (i.e., over the possible labelings of test examples), the more flexible the model is. To verify their method, Sandoval Segura et al. ran experiments to determine whether the LDM was a good empirical measure, but the results were inconclusive. In the paper, Sandoval Segura et al. also introduced the label recorder method. They showed that the method was promising because initial results of the method aligned with intuition about the models tested. In our paper, we provide theoretical background, continue the empirical exploration of the label recorder method, and explore its implications on model capacity with more experiments and a wider range of models.

Zhang et al. recently investigated the relationship (or lack thereof) between memorization and generalization for deep neural networks [16]. By training and testing state-of-the-art neural network models on randomized data, Zhang et al. determined that models which memorize training data perfectly can still generalize well. Smith and Le followed up on the work of Zhang et al., considering the memorization ability of logistic regression models [12]. They showed that a weakly regularized logistic regression model generalizes well, despite perfectly memorizing training data for the MNIST dataset, similar to the behavior observed by Zhang et al. Others have looked at the ability of human beings to create concepts from randomly generated data as a way of estimating human Rademacher complexity [17]. These experiments serve as inspiration for the label recorder method, and our study extends this mode of investigation to the memorization behavior of several standard non-neural learning methods.

2 Experimental Setup

We now describe our experimental setup, namely, the models, datasets, and data randomization methods used by our label recorder experiments. The standard classification methods we investigate are listed in Table 1. For Decision Trees and Random Forest, we generated groups of models by varying the tree depth regularization parameter from 1 to 21, in increments of 5. For the k-Nearest Neighbors algorithm, we do the same by varying the k regularization parameter from 6 to 21, in increments of 5. We used the default parameters in sklearn toolkit for rest of the models [8].

Table 1. List of models used, their abbreviations and hyperparameters varied.

Model Name	Abbreviation	Hyperparameters
Decision Trees	DT	Depth: 1-21, Criterion: Gini Impurity
k-Nearest Neighbors	KNN	k:1-21, Weights: Uniform
Random Forest	RF	Max-Depth: 1-21, Criterion: Gini
Adaboost	AB	Learning Rate: 1.0, Algorithm: SAMME.R
Quadratic Discriminant	QD	Priors: None, Regularization Parameter: 0
Gaussian Process Classifier	GPC	Optimizer:fmin_l_bfgs_b, Max Iters: 100
Gaussian Naive Bayes	NB	Priors: None, Smoothing: 10^{-9}
Linear Support Vector Machine	LinearSVC	Penalty: l2, Loss: Square-hinged
Logistic Regression	LogReg	Penalty: l2. Tolerance: 10^{-4}

We use two variants of data for our experiments. The first variant, as in Smith and Le [12], is a randomized digits dataset. For computational efficiency, we used an abridged version of randomized digits dataset, called Digits, provided by the scikit-learn library [9]. To create a balanced binary classification problem (which facilitates our later analysis), we select the 150 samples with label 0 and 150 samples with label 1 from that dataset. To randomize this digits dataset, we shuffle the labels of the samples. Then, for each model, we train it on this shuffled dataset and measure its training accuracy (namely, testing it on the same training set). To ensure the statistical significance of the results, for each algorithm and hyperparameter configuration, we repeat the above process for 500 independent trials, and compute the average and 95% confidence intervals of the resulting accuracy values. The average of those values is the label recording score of that model for the randomized Digits dataset.

However, the Digits dataset is limited in that we cannot vary the number of features nor the number of labels without compromising the distribution (digits data) that the dataset represents. Furthermore, being a real categorical dataset, the samples of the Digits dataset cluster in Euclidean space and do not represent a uniform random sampling of that space. We therefore create an additional dataset variant through uniform random sampling. This scheme generates random features and labels using integers drawn from a uniform random distribution. To create different datasets, we vary the parameters of the randomly sampled datasets as follows. The number of features for each sample ranges between 2 and 12 dimensions. The sizes of the datasets vary between

100 and 5000 samples. Finally, to compute the label recording score for the models on each dataset, we apply the label recorder method as described in the Digits experiments.

In the following section, we organize our analysis of the results in the following ways. First, we explore how label recording scores differ between the dataset variants, demonstrating the distribution dependency of label recording scores. Then, we inspect the label recording scores for all models on the shuffled Digits dataset; this demonstrates how label recording scores differ across models. After this, we use the results of the full synthetic data experiments to investigate how dataset parameters and model hyperparameters affect the label recording score of the models. Lastly, we conduct a brief theoretical analysis of the label recorder method, showing how scores can be probabilistically mapped to memorization capacities for rote learners and other supervised classification methods. Throughout all sections, we explore how the label recording score aligns with intuition on model capacity through the lens of decision boundaries.

3 Results

3.1 Differences Between Data Distributions

First, we investigate how the distribution of our datasets affects the label recording score for our models. We observe that the label recording score for models on the shuffled Digits dataset (shown in Table 3) and those of the fully synthetic data (Table 4) are clearly different. Namely, the label recording score of each model is generally lower for the shuffled Digits dataset than for the fully synthetic dataset. We can explain this phenomenon through the consideration of decision boundaries. As embodied in the notion of VC dimension, the more difficult it is to shatter a dataset, the harder it is to memorize that dataset. We hypothesize that it is more difficult to shatter the shuffled Digits dataset, i.e., to draw decision boundaries that separate the samples of that dataset by label within the feature space, than it is for the fully synthetic data, explaining the lower label recording scores for the former dataset variant. We also hypothesize that the shuffled Digits dataset is more difficult to shatter than the real Digits dataset.

To verify this, we use t-SNE [5] to visualize the spatial distribution of the samples of the real Digits dataset, the shuffled Digits dataset, and the uniform random (fully synthetic) dataset, as shown in Figure 1, projecting the samples onto a two-dimensional plane. In the resulting plots, we first notice the real digits dataset clusters samples of similar labels (as expected) while shuffled digits dataset clusters samples of different labels (due to randomization). In other words, shuffling the labels of the real Digits dataset preserves the clustering of samples but makes it harder to assign the same label to samples that are close to each other in feature space. Thus, it is more difficult for a model to draw a good decision boundary on the shuffled Digits dataset than on the real Digits dataset, which is expected as the former dataset has no correlation between labels and features.

We can also compare the t-SNE visualizations of the shuffled Digits and fully synthetic datasets. Note that the visualized synthetic dataset has the same number of samples and features as those of the shuffled Digits dataset. As shown in the Figure 1, while both datasets have differently-labeled samples in close proximity, those of the shuffled

Digits dataset are tightly clustered in groups while those of the fully synthetic dataset are more spread out in a spherical shape, due to the uniform random sampling. Thus, by the same logic as before, the shuffled Digits dataset is more difficult to shatter than the fully synthetic dataset, matching our hypothesis and explaining the higher label recording scores on the fully synthentic dataset.

We confirm our observations from the t-SNE visualization by quantifying the spread of differently labeled samples of the examined datasets. For each dataset, we calculate the average Euclidean distance from samples to their closest neighbors (ignoring labels), as well as to their closest differently-labeled and closest same-labeled neighbors. The results are in Table 2. As shown in the table, in both the full feature space and projected t-SNE space, the shuffled Digits dataset has smaller spread for differently labeled data points than both the real Digits dataset and the fully synthetic dataset.

While the shuffled Digits dataset has lower label recording scores than the fully synthetic dataset, we observe that their relative scores are similar. Specifically, when we rank the models by their scores in Table 3 and Table 4 respectively, all but two models (namely NB and QD) have the same rank. Also, preliminary experiments demonstrated that the label recording score trends with respect to dataset (i.e., the shape of the visualized line charts) of the two variants resembled each other. Combined with the fact that it easier to try different numbers of features and samples using the fully synthetic variant, we use the fully synthetic data to investigate the effects of dataset and model parameters on label recording score. We present these results in Sections 3.3-3.5.

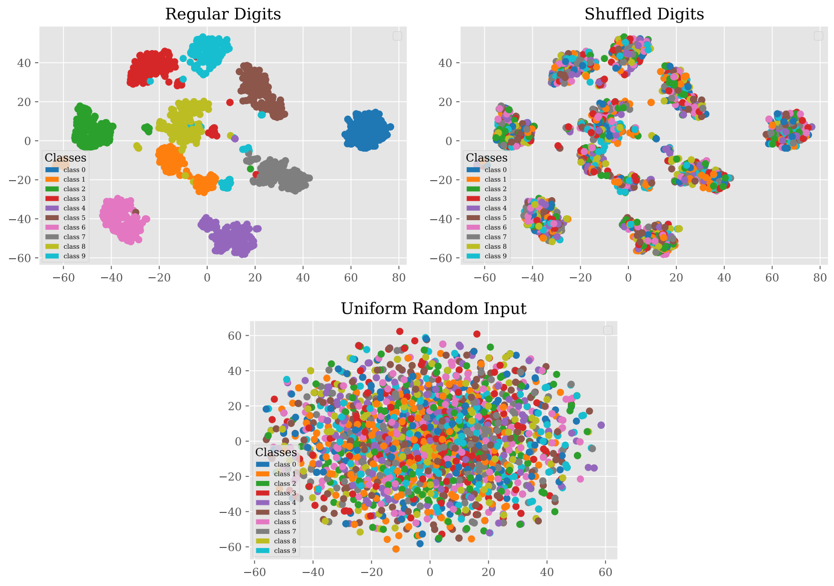

Fig. 1. The results of t-SNE for regular/true Digits dataset and Digits dataset with shuffled labels, as well as the fully synthetic uniform random dataset.

Table 2. Average Euclidean distance between a point and its nearest neighbor with specified label type in specified dataset.

Label Type	**Dataset**	**Avg. Feature Space Distance**	**Avg. t-SNE Space Distance**
Same Label	Real Digits	16.47	0.728
Same Label	Shuffled Digits	22.05	2.261
Same Label	Fully Synthetic	41.98	3.847
Different Label	Real Digits	29.50	11.54
Different Label	Shuffled Digits	16.62	0.638
Different Label	Fully Synthetic	39.42	1.071
Any Label	Real Digits	16.44	0.593
Any Label	Shuffled Digits	16.44	0.593
Any Label	Fully Synthetic	39.32	0.977

Table 3. Label recording scores for the shuffled Digits dataset. Maximum label recording scores (corresponding to perfect memorization) are bolded.

Model	**Digits Input Avg. Accuracy**	**Digits Input 95% Conf. Int.**
NB	0.52	(0.52,0.52)
GPC	**1.00**	(1.00,1.00)
AB	0.81	(0.80,0.81)
QD	0.63	(0.61,0.65)
KNN$_1$	**1.00**	(1.00,1.00)
KNN$_6$	0.65	(0.65,0.66)
KNN$_{11}$	0.62	(0.62,0.63)
KNN$_{16}$	0.60	(0.59,0.60)
KNN$_{21}$	0.59	(0.58,0.59)
DT$_1$	0.56	(0.56,0.57)
DT$_6$	0.79	(0.78,0.81)
DT$_{11}$	0.97	(0.97,0.98)
DT$_{16}$	**1.00**	(1.00,1.00)
DT$_{21}$	**1.00**	(1.00,1.00)
RF$_1$	0.62	(0.61,0.63)
RF$_6$	0.98	(0.97,0.98)
RF$_{11}$	**1.00**	(1.00,1.00)
RF$_{16}$	**1.00**	(1.00,1.00)
RF$_{21}$	**1.00**	(1.00,1.00)
LinearSVC	0.56	(0.55,0.57)
LogReg	0.66	(0.66,0.67)

Table 4. Label recording scores for fully synthetic data.

Model	**Rand. Input Avg. Acc.**	**Random Input 95% Conf. Int.**	**Random Average Minus Digits Avg.**
NB	0.73	(0.72,0.73)	0.21
GPC	**1.00**	(1.00,1.00)	0.00
AB	0.87	(0.87,0.88)	0.06
QD	**1.00**	(1.00,1.00)	0.37
KNN$_1$	**1.00**	(1.00,1.00)	0.00
KNN$_6$	0.66	(0.65,0.66)	0.01
KNN$_{11}$	0.62	(0.61,0.63)	0.00
KNN$_{16}$	0.60	(0.59,0.60)	0.00
KNN$_{21}$	0.59	(0.58,0.59)	0.00
DT$_1$	0.58	(0.58,0.58)	0.02
DT$_6$	0.87	(0.86,0.88)	0.08
DT$_{11}$	**1.00**	(1.00,1.00)	0.03
DT$_{16}$	**1.00**	(1.00,1.00)	0.00
DT$_{21}$	**1.00**	(1.00,1.00)	0.00
RF$_1$	0.74	(0.74,0.75)	0.12
RF$_6$	**1.00**	(1.00,1.00)	0.02
RF$_{11}$	**1.00**	(1.00,1.00)	0.00
RF$_{16}$	**1.00**	(1.00,1.00)	0.00
RF$_{21}$	**1.00**	(1.00,1.00)	0.00
LinearSVC	0.59	(0.58,0.60)	0.03
LogReg	0.70	(0.70,0.71)	0.04

3.2 Examining Labeled Recorder Scores

We begin by inspecting the label recording scores for the models trained on the shuffled Digits dataset. Given the randomization of labels and the balance in the distribution of the labels in the shuffled Digits dataset, one might guess that the result of training would not be much better than random guessing. However, to the contrary, we observe that many models perform significantly better than a uniform random guesser, with some achieving perfect memorization. The baseline expected accuracy for a uniform random guesser is $1/\ell$ where ℓ is the number of class labels. Table 3 gives the label recording scores for the various methods, with an average standard deviation of 0.01.

All methods perform better than 0.5, which is the baseline for a random guesser over two evenly distributed classes, and several methods achieve perfect accuracy. This demonstrates that above average labeling recording capacity is not exclusive to neural networks [16] or even limited to over-parameterized models [12]. Simple models such as depth-limited decision trees and KNN also exhibit nontrivial memorization capacity. While the dataset sizes are small for computational efficiency, our results provide evidence that standard learning methods have above uniform average memorization capacity.

One might be curious about why some methods perform better than others. This happens because of the some methods are able to carve out more complex decision boundaries than others. For example, as its name suggests, a LinearSVC model has a simple linear decision boundary. Thus it is a rigid model and no matter how the decision boundary line is drawn, on a binary classification problem on uniform random data it will perform close to a random guesser. But for tree-like classifiers (such as DT and RF) each branch allows you to encode some decision, thus making the boundary which can exclude or include points. This effectively operates as a way of storing information (namely, label mappings) from a dataset. Zhang et al.'s experiments show that neural networks can perfectly memorize random data [16]. Given many empirical instances where neural networks perform better than standard machine learning models, a difference in label recording capacity serves as one possible explanation for this performance difference.

3.3 Varying Dataset Size

To investigate the effect of dataset size on the label recording score, we conduct an additional set of uniform random data experiments, holding the number of classes, features, and feature values constant, while varying the number of samples.

As seen in Figure 2, increasing dataset size reduces label recording score for all but the KNN models. It makes intuitive sense that most methods will have lower label recording scores as they try to memorize more random samples, since models with limited storage capacity cannot retain an unlimited number of arbitrary label assignments. Note that the feature hyperspace is finite as it consists of a finite number of dimensions and finite number of possible values for each dimension. As we increase the size of the uniform random dataset, we uniformly sample this feature space more, which makes it such that datasets of larger size are more dense in the feature space. However, the labels of these samples are random; therefore, the samples with conflicting labels of larger datasets are more likely to be in close proximity (or overlap) in the feature space. This makes it more difficult to draw decision boundaries that separate and thus correctly label the uniform random samples. Thus, for most standard machine learning models, especially those whose decision boundaries are calculated using the entire dataset, the label recording score decreases with more samples, and models with these characteristics tend to approach random guessing asymptotically.

As the size of the datasets increase, Figure 2 shows that the average accuracies approach steady rates of decline, which is expected once models saturate their memorization capacity. Indeed, by computing the first and second pseudo-derivatives of the average accuracy using the differences in consecutive terms, we find that for all models the rates of change approach zero (results not shown).

In contrast to most models, the label recording scores of KNN models do not asymptotically decrease nor increase as the number of samples increases. Consider how KNN creates its decision boundaries as it trains on more samples: the predictions of a KNN model are only a function of the input sample's k nearest points. While other methods typically use the entire training set to generate a decision boundary, KNN draws its decision boundaries locally by labeling a sample with the most common label among the input sample's k nearest points. Thus, the decision boundary of a KNN model is a hyperdimensional polytope defined by the k closest samples to each point in the feature space. As more samples fill the feature space, the polytopes of the decision boundaries shrink.

Given that KNN predicts using local data points and the random nature of our data, we can calculate the expected accuracy or label recording score for a KNN model of a given k hyperparameter, and compare for agreement with our experimental results. For example, for KNN_2 testing on its own training data, the querying point will consider itself and its closet neighbor. Given that data points are randomly labeled, the neighboring point has equal probability of matching the querying point's label, and thus of being correct, assuming ties are broken randomly. For KNN_6, we can find the probability that two or more randomly labeled neighbors share the same label as the querying point to produce a correct classification, again assuming randomly broken ties and binary labels. There are 2^5 ways assigning binary labels to the 5 neighbors. We compute the probability of a correct classification as follows: there is 1 way to get all 5 other neighbors to have the specified label; there are $\binom{5}{4}$ ways to have 4 neighbors with the specified label; and there are $\binom{5}{3}$ ways to have 3 neighbors with the specified label. Assuming that KNN breaks ties randomly, there is a $\frac{1}{2}\binom{5}{2}/2^5$ probability of having exactly 2 neighbors with the specified label to get a plurality. The probability of getting a correct label is

$$\frac{\left[1+\binom{5}{4}+\binom{5}{3}+\frac{1}{2}\binom{5}{2}\right]}{2^5} \approx 0.656.$$

This agrees precisely with our observed results.

Besides the geometric interpretation, we can also think about the KNN results analytically. For each new data point, KNN determines its labeling by collecting the votes from its k-nearest neighbors. From another perspective, the model takes in the k-nearest neighbors (the rest of the dataset actually does not matter) and forms a classifier; we will call this a KNN-subclassifier. The larger a dataset, the more KNN-subclassifiers there are. Hence KNN is an ensemble of *n* KNN-subclassifers, and this ensemble without size limitation explains why KNN has a consistent capacity across different dataset size. KNN$_1$ is not displayed here as it has a 100% accuracy because it uses only one data point, which is itself, and training and testing data is identical.

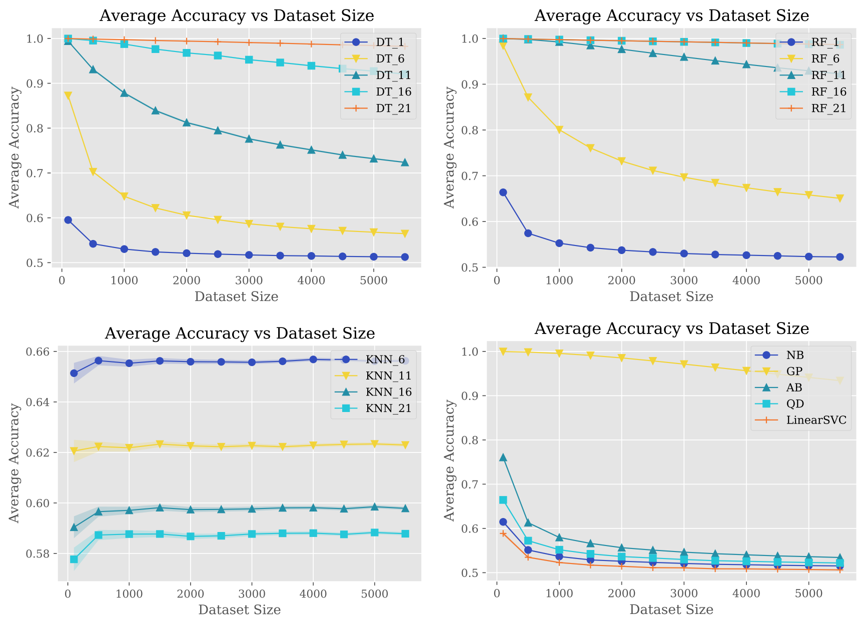

Fig. 2. Average Accuracy with 95% confidence intervals for varying dataset size, label dimension=2, and feature dimension=5.

3.4 Varying the Number of Features

We next vary the number of feature dimensions. Here the dataset consists of data with 10 possible labels and 750 data points, drawn from a uniform distribution.

As shown in Figure 3, in most cases the label recording score increases as the feature dimension increases. With fewer features, the feature space is smaller and we are more likely to experience collisions, namely, the same or similar samples having different labels. By the same logic, with few features each model (which maps features to labels) will have less information to work with. Conversely, increasing the number of features reduces these collisions. Note that weakly regularized tree-based models, e.g., Decision Trees and Random Forests with large tree depths, have high label recording scores irrespective of increasing feature dimension, because they are able to take advantage of these features in a better way through encoding each individual feature as a level of the tree. KNNs remain unaffected by changing feature dimensions because, as explained in Section 3.3, they perform inference locally and not globally.

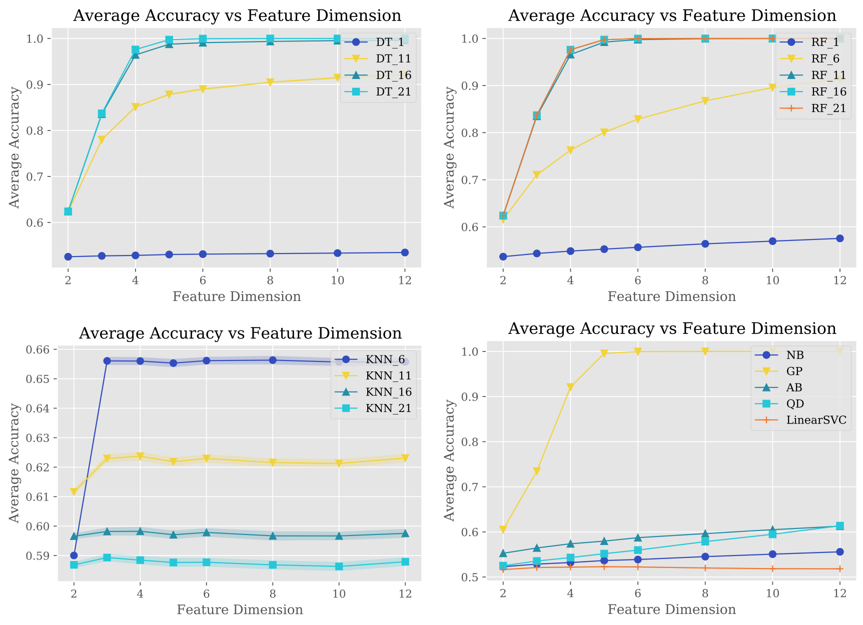

Fig. 3. Average Accuracy with 95% confidence intervals for varying feature dimensions, label dimension=2, and dataset size=1000.

3.5 The Effect of Regularization

To investigate the regularization effect on label recording scores, we compare the results of the KNN, decision trees, and random forest models with varying regularization parameters. In particular, note that having a larger number of neighbors increases regularization for the KNN model while a lower maximum tree depth increases regularization for decision tree and random forest models.

In agreement with expectation, Figure 4 shows that the more regularized a model is, the less it memorizes, highlighting how the label recording score is an intuitive measure of memorization capability. Because the data is random, this suggests the label recording score can be a useful empirical proxy for representational capacity.

3.6 Using the Label Recording Score to Gain Insights on New Models

From our analysis, we see that the label recorder score intuitively responds to changes in the parameters of the machine learning problem and model. For example, we've demonstrated a correspondence between the increase of data complexity and size and regularization of the model with the decrease of the label recording score—reflecting the model's memorization ability. It is a natural extension of this observation to use the label recorder score to gain insight into new models. For example, if adjusting a parameter of a model decreases the label recording score, we might speculate that the parameter regularizes the model. As another example, if a model's label recording score

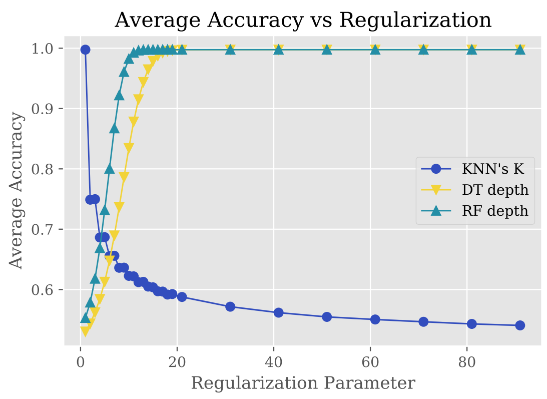

Fig. 4. Average Accuracy with 95% confidence intervals for varying regularization parameters, label dimension=2, feature dimension=5, and dataset size=1000.

does not change significantly as we increase the dataset size, we might surmise that the model creates local decision boundaries similar to those of KNN. This could be an area of further investigation.

As observed in the differences in label recording score between the two dataset variants (keeping dataset size and number of features fixed), the label recording score is distribution dependent. This might indicate that a model's memorization ability depends on the distribution from which the training data is sampled. From the perspective of representational capacity, this tells us that models are better at representing some types of data rather than others.

4 Theoretical Analysis

In this section we perform a preliminary theoretical analysis of some properties of the label recorder method, showing how to probabilistically relate label recording scores to model capacities, and work through an example using simple memorization models for which exact capacities are known. A fuller theoretical treatment is the subject of future work.

4.1 Rote Memorizer

As the name suggests, a *Rote Memorizer* memorizes some finite number of examples and will uniformly randomly guess the label of any example not memorized. It is implemented via a simple look-up table, with the features as keys and the labels as values. Figure 5 shows the label recording performance for rote memorizers of varying capacities, as measured in the number of examples memorized. As expected, a rote memorizer has perfect accuracy when the number of examples is fewer than its memorization capacity, and accuracy decreases once the dataset size exceeds its capacity.

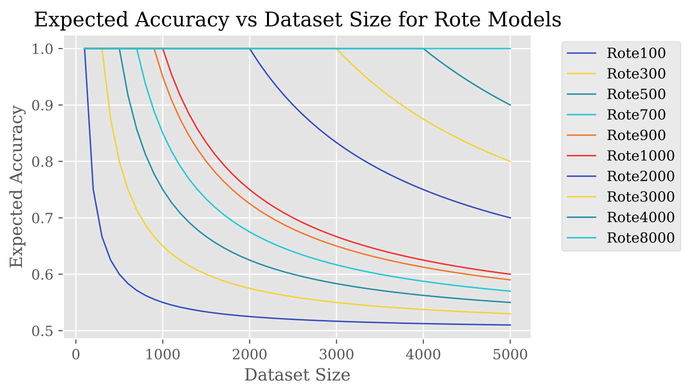

Fig. 5. Analytical plot for varying dataset size for Rote models, with varying number of examples memorized (i.e., Rote100 refers to a rote memorizer capable of storing 100 examples). The label dimension of the dataset is 2, and feature dimension is 5. These analytical plots agree with subsequent independent empirical validation (figure not shown).

Comparing Figures 5 and 2, which demonstrate the effects of varying dataset size on the Rote Memorizer and other models, we see that both have similar asymptotic behavior of approaching 0.5, that is, they both turn into random guessers for large dataset sizes. There is a corresponding decrease in accuracy after the models are saturated, which for most models occurs with the smallest dataset tested ($n = 100$), suggesting that their capacity is smaller than needed to perfectly memorize all examples. It may additionally demonstrate a strong reliance on dependence between features and labels, or between data points themselves, suggesting a strategy of generalization rather than memorization [7]. Removing all dependence would hinder these methods accordingly. Additionally, while Rote memorizers can store feature-label pairs independently of one another, generalizing methods like Logistic Regression and Decision Trees store their information in interacting ways, such that changing the classifier to correctly label one point often affects the classification of other points. This further explains why Rote Memorizers demonstrate a sharp "phase transition" after saturation, compared to the more gradual reductions in capacity observed in the other methods.

One can qualitatively compare the curves from real models to those of Rote Memorizers of different capacities. For example, seeing that Adaboost responds to increasing dataset size in a way that closely mimics the Rote100 learner suggests that its memorization capacity is close to 100 examples. Figure 5 can thus serve as a reference guide.

4.2 Expected Label Recording Scores

Let us consider expected accuracy of a Rote Memorizer. Let m be the number of entries that can be memorized by a Rote Memorizer, d be the size of the data set and ℓ be the label dimension. Furthermore, let $m' = \min(m, d)$. The number of correct examples becomes the number of examples memorized plus the number correctly guessed. Thus,

$$\mathbb{E}\left[\text{Accuracy}\right] = \frac{m' + (d - m')\frac{1}{\ell}}{d} = \left(1 - \frac{1}{\ell}\right)\frac{m'}{d} + \frac{1}{\ell} = 1 - \left(1 - \frac{1}{\ell}\right)\left(1 - \frac{m'}{d}\right).$$

For binary classification case with $\ell = 2$, this simplifies to

$$\mathbb{E}\left[\text{Accuracy}\right] = 1 - \left(1 - \frac{1}{2}\right)\left(1 - \frac{m'}{d}\right) = 0.5 + 0.5\left(\frac{m'}{d}\right).$$

When $m' \ll d$, $\mathbb{E}\left[\text{Accuracy}\right] \approx 0.5$, explaining the observed asymptotic behavior.

Similarly, for a general algorithm $\mathcal{A}$ and a dataset D of size d, we can decompose $\mathbb{E}\left[\text{Accuracy}\right]$ into its two components: the accuracy on the memorized portion of the dataset, m'/d, and the accuracy on the remaining, unmemorized portion, $\frac{d-m'}{d}u_{\mathcal{A}}(D)$, where m' is the number of data points that $\mathcal{A}$ can memorize from the dataset D, and $u_{\mathcal{A}}(D)$ is the expected accuracy of the algorithm $\mathcal{A}$ on unmemorized data. This yields

$$\mathbb{E}\left[\text{Accuracy}\right] = \frac{m'}{d} + \frac{d - m'}{d}u_{\mathcal{A}}(D) = 1 - (1 - u_{\mathcal{A}}(D))\left(1 - \frac{m'}{d}\right).$$

For the specific case of a Rote Memorizer on random data D with ℓ labels, we have $u_{\text{rote}}(D) = 1/\ell$, since the Rote Memorizer performs uniform random guessing for all unmemorized parts of the dataset. Because completely random data has no correlation between features and responses, we conjecture $u_{\mathcal{A}}(D)$ will be approximately $1/\ell$ for all methods trained on such datasets, as no generalization is possible, leaving luck and memorization as the only paths to correct label prediction.

4.3 Relating Label Recorder Score to Memorization Capacity

Let $LR(M,D)$ denote the label recording score of model M trained on dataset D, which is the observed accuracy of the model trained and tested on that same dataset. When M is stochastic or D is random, this becomes a random variable $Z = LR(M,D)$, and considering a collection of n random datasets, the label recording score for each dataset D_i is $Z_i = LR(M,D_i)$. Because our datasets are independently and identically distributed, we have $\mathbb{E}\left[\text{Accuracy}\right] = \mathbb{E}\left[Z_i\right]$.

We can reverse the relationship between $\mathbb{E}\left[\text{Accuracy}\right]$ and m' to obtain

$$m' = d\frac{\mathbb{E}\left[\text{Accuracy}\right] - u_{\mathcal{A}}(D)}{1 - u_{\mathcal{A}}(D)}.$$

Therefore, knowing $u_{\mathcal{A}}(D)$ and having a bound on $\mathbb{E}\left[\text{Accuracy}\right]$ would allow us to bound the memorization capacity of a learner.

Consider the average label recording score over n independent trials, $\bar{Z} = \frac{1}{n}\sum_{i=1}^{n} Z_i$. By linearity of expectation, $\mathbb{E}[\bar{Z}] = \mathbb{E}[Z_i] = \mathbb{E}\left[\text{Accuracy}\right]$. Since $0 \leq Z_i \leq 1$, we apply Hoeffding's inequality to obtain

$$\mathbb{P}(|\bar{Z} - \mathbb{E}\left[\text{Accuracy}\right]| \geq \varepsilon) = \mathbb{P}(|\bar{Z} - \mathbb{E}[\bar{Z}]| \geq \varepsilon) \leq 2e^{-2n\varepsilon^2}.$$

With this inequality and the average of n independent observations of the labeling recorder score, we can obtain a probabilistic bound on $\mathbb{E}\left[\text{Accuracy}\right]$, and by extension, on the memorization capacity of a learning method.

Example For a Rote Memorizer, assume we have computed the label recording score $n = 1000$ times, and found that average of the scores is $\bar{Z} = 0.8$. As such,

$$\mathbb{P}(|0.8 - \mathbb{E}[\text{Accuracy}]| \geq \varepsilon) \leq 2e^{-2000\varepsilon^2} := \delta.$$

For example, with probability at least $1-\delta \approx 0.9$, we have $|0.8 - \mathbb{E}[\text{Accuracy}]| < 0.04$. For our Rote Memorizer, this implies that with probability greater than $1-\delta = 0.89$, we have $0.76 < \mathbb{E}[\text{Accuracy}] < 0.84$, and thus,

$$d\frac{0.76 - \frac{1}{\ell}}{1-\frac{1}{\ell}} < m' < d\frac{0.84 - \frac{1}{\ell}}{1-\frac{1}{\ell}}.$$

More generally, we can say that for a learning method $\mathcal{A}$ with average label recorder score of $\bar{Z}$ computed from n independent trials, we have

$$\mathbb{P}\left(d\frac{(\bar{Z}-\varepsilon) - u_{\mathcal{A}}(D)}{1-u_{\mathcal{A}}(D)} < m' < d\frac{(\bar{Z}+\varepsilon) - u_{\mathcal{A}}(D)}{1-u_{\mathcal{A}}(D)}\right) > 1 - 2e^{-2n\varepsilon^2}. \tag{1}$$

5 Conclusion

We investigate the label recorder method, a way of empirically assessing the (data-dependent) memorization capacity of a model. To use the label recorder method, we train and evaluate a model on the same set of random data, with the evaluation accuracy (i.e., training accuracy) serving as the label recording score. We investigate how the label recording score is influenced by: the distribution of a dataset, the mechanics of a model, the size of a dataset, the number of feature of a dataset, and the regularization parameters of a model. We demonstrate that the label recording score can be explained by reasoning about the decision boundary of each model, and we observe that the relative magnitude of the label recording score matches our intuitions regarding representational capacity. The method is fully empirical and can be applied to black-box classification methods, allowing one to reason about representational and information storage capacity independently from theoretical concerns.

We further suggest that running label recording experiments on black-box models can help us to gain insights into their inner workings and hyperparameter effects on that model. Future research directions include expanding the family of models considered, using label recording scores to estimate the information storage capacity of concrete models, and using estimated capacity to derive probabilistic generalization guarantees. Estimating label recording scores directly from model hyperparameters and parameters of datasets would be useful and desirable, but cannot be done perfectly for arbitrary learning algorithms and datasets, since doing this would allow us to determine whether an arbitrary algorithm will overfit a dataset, a problem recently shown to be formally undecidable [1].

Running label recording on machine learning problems during hyperparameter optimization to compare different variations of models is another promising future application. For example, one could compare the label recording score of stopping a model at 20 epochs vs. 10 epochs. More work remains to investigate the correlations between label recording scores, regularization, and algorithm information storage capacity.

References

1. Bashir, D., Montañez, G.D., Sehra, S., Sandoval Segura, P., Lauw, J.: An Information-Theoretic Perspective on Overfitting and Underfitting. In: 33rd Australasian Joint Conference on Artificial Intelligence (AJCAI) (2020)
2. Bassily, R., Moran, S., Nachum, I., Shafer, J., Yehudayoff, A.: Learners that Use Little Information. In: Janoos, F., Mohri, M., Sridharan, K. (eds.) Proceedings of Algorithmic Learning Theory. Proceedings of Machine Learning Research, vol. 83, pp. 25–55. PMLR (07–09 Apr 2018), http://proceedings.mlr.press/v83/bassily18a.html
3. Castellini, J., Oliehoek, F.A., Savani, R., Whiteson, S.: The Representational Capacity of Action-Value Networks for Multi-Agent Reinforcement Learning. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. pp. 1862–1864 (2019)
4. Cortes, C., Kloft, M., Mohri, M.: Learning kernels using local rademacher complexity (2013)
5. Maaten, L.v.d., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research **9**(Nov), 2579–2605 (2008)
6. Mahalunkar, A., Kelleher, J.D.: Using Regular Languages to Explore the Representational Capacity of Recurrent Neural Architectures. In: International Conference on Artificial Neural Networks. pp. 189–198. Springer (2018)
7. Montanẽz, G.D.: Why Machine Learning Works. In: Dissertation. Carnegie Mellon University (2017)
8. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
9. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in Python. the Journal of machine Learning research **12**, 2825–2830 (2011)
10. Samengo, I., Treves, A.: Representational Capacity of a Set of Independent Neurons. Physical Review E **63**(1), 011910 (2000)
11. Sandoval Segura, P., Lauw, J., Bashir, D., Shah, K., Sehra, S., Macias, D., Montañez, G.D.: The Labeling Distribution Matrix (LDM): A Tool for Estimating Machine Learning Algorithm Capacity. In: Rocha, A.P., Steels, L., van den Herik, H.J. (eds.) Proceedings of the 12th International Conference on Agents and Artificial Intelligence, ICAART 2020, Volume 2, Valletta, Malta, February 22-24, 2020. pp. 980–986. SCITEPRESS (2020), https://doi.org/10.5220/0009178209800986
12. Smith, S.L., Le, Q.V.: A Bayesian Perspective on Generalization and Stochastic Gradient Descent. In: International Conference on Learning Representations (ICLR) (2018)
13. Vapnik, V., Levin, E., Cun, Y.L.: Measuring the VC-dimension of a Learning Machine. Neural computation **6**(5), 851–876 (1994)
14. Xu, A., Raginsky, M.: Information-Theoretic Analysis of Generalization Capability of Learning Algorithms. In: Proceedings of the 31st Conference on Neural Information Processing Systems (2017)
15. Yin, D., Kannan, R., Bartlett, P.: Rademacher complexity for adversarially robust generalization. In: International Conference on Machine Learning. pp. 7085–7094. PMLR (2019)
16. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding Deep Learning Requires Rethinking Generalization. CoRR **abs/1611.03530** (2016), http://arxiv.org/abs/1611.03530
17. Zhu, J., Gibson, B., Rogers, T.T.: Human Rademacher Complexity. Advances in Neural Information Processing Systems **22**, 2322–2330 (2009)