Time and the Value of Data

Ehsan Valavi
Joel Hestness
Newsha Ardalani
Marco Iansiti

Working Paper 21-016

Time and the Value of Data

Ehsan Valavi
Harvard Business School

Joel Hestness
Cerebras Systems

Newsha Ardalani
Baidu Research

Marco Iansiti
Harvard Business School

Working Paper 21-016

Funding for this research was provided in part by Harvard Business School.

Time and the Value of Data

Ehsan Valavi[1], Joel Hestness[2], Newsha Ardalani[3], and Marco Iansiti[1]
evalavi@hbs.edu, jthestness@gmail.com, newsha@baidu.com, miansiti@hbs.edu

Harvard Business School, Boston, Massachusetts.[1]
Cerebras Systems, Los Altos, California.[2]
Baidu Research, Sunnyvale, California.[3]

Abstract

This paper investigates the effectiveness of time-dependent data in improving the quality of AI-based products and services. Time-dependency means that data loses its relevance to problems over time. This loss causes deterioration in the algorithm's performance and, thereby, a decline in created business value. We model time-dependency as a shift in the probability distribution and derive several counter-intuitive results.

We, theoretically, prove that even an infinite amount of data collected over time may have limited substance for predicting the future, and an algorithm that is trained on a current dataset of bounded size can attain a similar performance. Moreover, we prove that increasing data volume by including older datasets may put a company in a disadvantageous position.

Having these results, we answer questions on how data volume creates a competitive advantage. We argue that time-dependency weakens the barrier to entry that data volume creates for a business. So much that competing firms equipped with a limited, but sufficient, amount of current data can attain better performance. This result, together with the fact that older datasets may deteriorate algorithms' performance, casts doubt on the significance of first-mover advantage in AI-based markets.

We complement our theoretical results with an experiment. In the experiment, we empirically measure the value loss in text data for the next word prediction task. The empirical measurements confirm the significance of time-dependency and value depreciation in AI-based businesses. For example, after seven years, 100MB of text data becomes as useful as 50MB of current data for the next word prediction task.

Keywords: Economics of AI, machine learning, non-stationarity, perishability, value depreciation

1. Introduction

We are witnessing a dramatic acceleration of digitization in infrastructure, products, and services. Artificial Intelligence (AI) enabled solutions are on the rise, and more than ever, data appears to be a critical strategic asset [1,5,12,17]. As a result, companies are amassing substantial volumes of user data to improve their current and future services, hoping that it gives them an advantage over their competitors.

Recent research hypothesizes that AI-enabled products' quality improves from a reinforcing feedback loop created by increasing data volume [23,27]. Gregory et al. [23] compare this data externality to network effects, where the value of a service or product increases in user-base size. In the "data network effect" [23,26,35], more data leads to a higher quality of algorithms, which means better services [24]. Better service then leads to a higher user engagement or larger user-base, which in turn creates even more data. The logic is intuitive, and generally speaking, the more the data, the more the value created and delivered.

Beyond the generality, we want to delve into the mechanisms by which data generates value, and then study whether data volume creates benefits such as barriers to entry or other competitive advantages. More specifically, we would like to understand how data characteristics influence the way created value scales in a business. Examples of those characteristics include the dataset's size, information richness, potential biases, and time dependency. From an economics perspective, factors influencing the strength of any potential "data network effect" can include whether the data is exclusive, imitable, nonrival, complementary, or perishable [12,31,38,42].

Of data characteristics, we identify time-dependency and perishability as key characteristics with a quantifiable mechanism to influence value creation. Time-dependency refers to the time since a data point was sampled or the time period over which a dataset was collected. Both definitions are the subject of research in this study. The perishability refers to the value loss as the dataset ages. With this definition, we call a dataset that loses its value over time, a perishable dataset.

In this paper, we investigate the impact of time-dependency and data perishability on the effectiveness of a dataset in creating value for a business. Any product or service can be the outcome of many tasks. In lieu of studying many tasks, we focus our attention on a generic task and study how time-dependency and perishability characteristics influence the AI performance on that task. In practice, a company must study the value that a dataset creates for all tasks and the tasks' inter-relations. It can then derive the overall impact of the dataset on the entire business.

To study the effectiveness of a dataset, our base of comparison would be a dataset of similar size that is sampled independently, without bias or any delay compared to testing time. We refer to it as the baseline dataset. Without delay means that data sampling, training, and testing happen in a relatively short time, and hence, we have stationarity. The comparison would be then between the baseline dataset and the one we have in hand, which is sampled indeed unbiasedly and independently; however, testing time is different from the sampling time. In other words, we fix all characteristics of the datasets and compare the effectiveness only in the time dimension.

In our study, we observe time-dependency as a shift in data's probability distribution over time. In this view, we relate gradual changes in the real world to increase or decrease in their appearance frequency in the dataset. Our goal is then to see how changes in the probability distribution alter an algorithm's performance. We expect a higher loss value due to the differences in distributions. A higher loss value means that if we do not have time-dependency (If we use baseline dataset for training), a dataset of smaller size can generate a similar loss value. Therefore, we conclude that a given dataset, for a given task, loses a portion of its power, and we refer to this loss as perishability.

Without loss of generality, we only investigate the problem of learning the probability distribution. Because it is the ultimate goal of statistical learning problems, and we can directly derive every statistic from the learned distribution. We also use the Maximum Likelihood Estimation (MLE) of the probability distribution for our analysis since it is a consistent and efficient estimator. Efficiency means that the MLE reaches Cramer-Rao lower bound, and hence, for any given number of data points, it has the lowest estimation variance between unbiased estimators. Furthermore, on the execution side, we assume that algorithms converge to the globally optimal points.

We present several counter-intuitive results. We prove that even a dataset of infinite size from a wrong distribution has limited predictive substance, which can be met by a dataset of bounded size sampled from the right distribution. Relating this to time-dependency, increasing training dataset size with the data sampled from the very past (Lost its relevance significantly) may not improve an algorithm's performance. Furthermore, we argue that using data from a very distant past may deteriorate the algorithm's performance. Therefore, this performance decline over time may put a firm in a disadvantageous position.

The performance decline argument requires further investigation into the functional form of algorithms' performance versus data over time. We need to know if, over time, the algorithm's performance is deteriorating monotonically. To confirm the monotonic behavior, we empirically measure the effectiveness curve of data over time. For a given time, the effectiveness curve shows the effective size of a dataset compared to a baseline dataset.

We choose the next word prediction task for our empirical measurements and use a dataset [19] from Reddit.com. After confirming the effectiveness curve's semi-monotonicity, we propose a simple framework to improve the effectiveness of a dataset sampled over a period of time. The method, called sequential offloading, deletes data from the past in the hope of increasing the relevance (Freshness) of a dataset. Upon successful deletion, a dataset loses size, which reduces the complexity of operations and gains relevance, which means it has better merit. We use this method to prove the argument that we made earlier on how data may put a firm at a disadvantage.

In this paper, in the framework section, we explain our approach to the problem and clarify why we made particular choices. Then, in section 3, we introduce the effectiveness curve and explain value depreciation over time. We show the bounded effectiveness in this section. Section 4 investigates the effectiveness of datasets curated over time. These datasets are a combination of recent and old data. We explain sequential offloading and suggest that, in businesses with high time-dependency, old data may put a firm in a disadvantaged position. Section 5, empirically, measures the value depreciation for the next word prediction task. Finally, in the conclusion section, we wrap this paper with a discussion.

1.1 Literature Review

Our work is related to several areas in machine learning, economics, and statistics. The entire statistics and machine learning literature consider data as a fundamental asset. Our work contributes to the applied machine learning literature by providing insights into evaluating the value of perishable data. Notably, our sequential offloading algorithm provides a method for data scientists to determine the value that data has for a particular learning problem. It enables them to better manage their resources toward developing more effective AI-based solutions.

In the economics literature, the impact of data and AI on economics and firm performance has been examined by several authors [e.g., 3,4,8,11,28,29,37,41,43]. As stated in [4], AI is a general-purpose technology that reduces the price of prediction. Prediction problems are fundamental in many economic sectors, and the new technology changes the way firms operate, which has implications on productivity [3], employment [8], inequality [29], and competition [41]. Despite its rapid advancement in recent years [11], the technology is still far from maturity, and its impact, compared to its potential, is minimal. [43] provides an overview of challenges that industries are facing and how these challenges affect the industrial organization of providers and adopters of AI technology. [15] studies the widespread application of machine learning in fields other than computer science, thereby attesting to its potential for innovation in different areas. As an example, in economics and specifically in market design, [32] discusses the applications. Because of AI's widespread applications, data becomes an essential asset for firms and for the economy in general, which motivates companies to curate big datasets.

Acquiring data can happen in many ways. While firms may purchase data through an intermediary, they may also organically gather datasets over time from interactions with their users. In both methods, there are privacy concerns that may prevent users from sharing data. [10], through modeling an intermediary that acquires data from users and sells the obtained data to firms, investigates the issue of data externality and privacy. [2] discusses similar issues for organic data generation and claims that more data lowers the privacy barrier and motivates more users to share their data. This argument, together with the data network effect arguments above, suggests a significant growth rate in the size of a firm's data repository. [9, 20, 21] proposed a growth model for data in firms and the economy. They answer questions on the firm's growth process.

Our paper researches the effectiveness of curated datasets, and hence, is not concerned with data solicitation and growth of the firm's data repository. We argue that curating big datasets and blindly using them may not always give a firm a significant advantage, and it may even put a firm in a disadvantageous position. Our arguments, thus, question universal assumptions about the value of data for a firm and how it may change the modes of competition. More precisely, we investigate how and if curating large datasets can create barriers to entry and deter threats from entrants.

[17,18,33,34,38] discuss the implication of AI and, more precisely, data on competition. About data, particularly, most debates are around its volume and whether it creates a competitive advantage. Some of these studies focus on the antitrust issues and the potential role that data plays in creating a winner take all (monopoly) situation. [31,39] are examples of these studies. Furthermore, there are researches on how data can improve the prediction quality of services with respect to either the degree of personalization [25,39] or between adjacent products [7]. These researches have direct strategy implications on how firms compete on growing the user-base.

We believe that data characteristics play a crucial role in the value creation cycle and the modes of competition. For example, non-rivalry and exclusivity of a dataset to a firm can prevent other players from obtaining it, which in turn puts the owner in a superior position. Under exclusivity, data becomes an asset that behaves like the supply of a physical good. Biases build a harmful environment for both the company and its users. For the algorithm's fairness and potential biases, [16] provides a discussion. Paying closer attention to dataset characteristics, we can see that time-dependency and perishability are similar since the sampling time determines them both. It is of great interest to see how dependency on time changes the strength of data externalities and influences the value creation cycle.

Perhaps the closest research to ours is [14], where the authors investigate the effect of historical search data on search results' quality. They found little empirical evidence on the effectiveness of old data in the quality of search engine results. Also, [6] raises a question on the economy of scale that data provides for specific problems. They suggest a diminishing return to scale value model for data and argue that increasing data volume in advertisement applications does not improve the service quality. The results in these papers endorse our findings on the effectiveness of perishable data. We believe that both the search engine and advertisement businesses use time-sensitive data and hence, face significant time dependency. Therefore, the data loses its effectiveness quickly, contributing to not seeing significant improvement in prediction quality.

2. Background and Framework

In this section, our goal is to introduce how we approach the problem and explain why we make particular choices. We first describe time-dependency as a shift in distribution over time and dig into its cause. We argue that time-dependency is mostly due to exceptional reasons that most often cause a monotonic decrease in the value of data over time. We then have a brief introduction to machine learning and explain why we focus on the probability distribution's maximum likelihood estimation. We mainly introduce a decomposition of the MLE's objective function to lay the groundwork for the next section's propositions. We finally formalize the notion of effectiveness and define the substitution gain curve as a proportion of two effectiveness quantities; one from past and the other from the substitute time.

2.1. Change in distribution

Time-dependency is due to many reasons, among which we can mention the change in consumers' tastes and behavior. If we look at music albums' best sellers from the 80s and compare them with the best sellers in 2020, we can see the difference in taste. Innovation is another reason for the change because, nowadays, we witness continuous innovation and considerable variation in product and services space. Telegram seems ancient these days, and so will be the hardline telephones soon. Because of the differences, it is not easy to translate the environment from different times to each other. Perhaps the way people communicate is the best way to observe such changes. Hundreds of years ago, people used letters and the post to communicate to far distances. If we use those letters to train a text auto-completion model for smartphones today, users will be disappointed. It is because, today, we use some expressions or words less frequently. From a scientific modeling perspective, it is as if data generating distribution is changed and has lower significance for those particular words or expressions. In the meantime, the language allows for the birth of new phrases and words, which is equivalent to an increase in their frequency of use. This birth and death process of probability space elements is among the very reasons we see the depreciation in data value over time.

Almost everything related to the effectiveness or relevance of a dataset, like the shift in consumer tastes or other contributing factors in perishability, can be observed in the data's histogram. In this view, a particular datapoint loses its relevance over time if the number of times it shows up becomes larger/smaller (Compared to today) over time. It is because it is not reflecting the actual frequency of the datapoint, and hence, it overestimates/underestimates its importance. For example, think of a song on Spotify and assume that the company saves subscriber's data over time. If the number of times a particular song is requested declines, its appearance frequency declines in the data as well, and hence, we should expect a smaller value in the histogram. Despite expecting a decline, its higher frequency in the past data leads to overestimation. Alternatively, if the number of times a song is played increases over time and we use the dataset with its lower appearance frequency, we underestimate its importance.

We believe that over time, more elements are either born or eliminated. Since it is unlikely for dying elements (Elements whose appearance frequency becomes smaller) to gain relevance over time, we expect some sort of monotonic behavior in the relevance or effectiveness of data. There may still be arguments about domains like fashion or periodic things like seasonal and recurring events. We conjecture that there are slight bouncing back in relevance over time, but as our experiment on Reddit data shows, the effect is negligible. Intuitively, we argue that fashion, by definition, requires exclusivity, and it is highly unlikely to witness a complete comeback of older tastes. As of recurring events and seasonal data, like purchasing behavior at Christmas or Valentine's, we should not forget the innovation in the product and services space. We may observe specific behavior on those occasions, but still each time, there is something different.

To compare histograms and distributions over time, we create a universal set of elements. Without it, we cannot compare elements from the past and future. For example, the word iPhone is created in the 2000s. In the language dataset from 1900, this element does not exist, and hence, it is not measurable. Therefore, it is natural to put this word in the element set of the 1900s' probability space and designate it a zero probability. In our research, we add nonexistent elements like iPhone to the element set of every other time. Still, instead of assigning them a zero probability, we give them an infinitesimal value. This infinitesimal probability helps us use different functional forms like the log function without being worried if they are on the functional domains.

Formalizing the assumptions we made so far, we assume that the prediction is for time 0 with a model trained on data from the past. The data is from a time that is t period prior time 0 where $t \in \{0\} \cup \mathbb{R}^+$. For the sake of simplicity, we call it sampled data at time t. We show the element set at time t by χ_t and define the probability space by $\left(\chi_t, \sigma(\chi_t), \tilde{P}_t\right)$. The universal probability space is then $(\chi, \sigma(\chi), P_t)$, where $\chi = \bigcup \chi_t$ and

$$P_t(x) = \begin{cases} \dfrac{\tilde{P}_t}{1-\delta} & x \in \chi_t \\ \delta_x & x \in \chi - \chi_t \end{cases}$$

$\delta = \sum_{x \in \chi - \chi_t} \delta_x$ and $\delta_x > 0$. As explained earlier, we prefer δ to be zero, but due to some regulatory conditions in the MLE's loss function, we assume δ is infinitesimal. With this change in the measure, it is possible to compare datasets and define the shift in distribution. A change in distribution between the time i and j means $\exists\ x \in \chi\ s.t.\ P_i(x) \neq P_j(x)$.

2.2. Learning Data Distribution

Machine learning is fundamentally dealing with finding meaningful relations between inputs and outputs of an unknown system. In this framework, the unknown system is doing a particular task, and our goal is to mimic the way the system operates as closely as possible. Putting this into mathematical semantic, given a dataset $D_{n,t} = \{(x_i, y_i)_t\}_{i=1}^{n}$, which is composed of n input-output samples collected at time t, we want to find a model or a function $m(x, y) \in \mathcal{M}$ that describes the relationship between the input vector x_i and the output y_i. $\mathcal{M}$ is the set of all candidate functions. In the case of supervised learning, y_i is observable and is given in the dataset, whereas in unsupervised learning, it is masked. In both cases, the learning goal is achievable if data is sampled independently and from an identical distribution. Dataset's elements are from the element space χ.

In most machine learning cases, the set $\mathcal{M}$ is composed of functions $m(x, \theta)$ where the goal is to make $m(x, \theta)$ as close as possible to y by learning the parameter θ. Linear, logistic, and deep neural network compositional functions are examples of $m(x, \theta)$. Table 1 provides the functional forms for these three examples.

Case	Functional form $m(x, \theta)$
Linear functional	θx
Logistic functional	$\dfrac{e^{\theta x}}{1 + e^{\theta x}}$
Simple Deep Learner with L layers and non-linear functions σ	$\theta_L \sigma_L(\theta_{L-1}\sigma_{L-1}(\ldots\ (\ \theta_2\sigma_2(\theta_1 x)\ )\ \ldots))$

Table 1) Examples of functional forms for famous ML models.

Identifying the unknown probability distribution is the fundamental problem of statistical learning theory. It does not matter whether we directly want to learn the transition probabilities in a decision tree or, indirectly, trying to fit a functional to data. Either way, the goal is to deal with the characterization of a distribution function. In the case of the decision tree or generally Markov decision processes, the ultimate goal is to deal with the transition probabilities between states. Remember that in a decision tree, upon taking action, the system has a transition to the next state. The optimal action, in this problem, is a function of transition probabilities.

In fitting a functional, we have a known function that we believe can describe the data. No matter how suitable we believe the model is for the data, it is often not entirely fit, and there exist a fitting error (Noise) ϵ. The noise can have additive, multiplicative, or other forms of contribution in the fitting task depending on its nature. Without loss of generality, we consider the additive contribution.

$$y = m(x, \theta) + \epsilon$$

The fitting job is then to identify the distribution of ϵ.

$$\epsilon \sim P\big(y - m(x, \theta)\big) = P_\epsilon$$

In general, all statistics (Models) are a function of data distributions. Consequently, under specific regulatory conditions, a sequence of distributions converging to the underlying distribution defines also a converging sequence of any statistic (Model) to its converging value. This argument attests that learning the underlying distribution is the fundamental problem in machine learning.

Consequently, we restrict our theoretical analysis to the problem of learning the underlying data distribution. Further, we choose the maximum likelihood estimator for this task since it is an efficient unbiased and consistent estimator. Due to its efficiency, it is rational to prefer it over other unbiased estimators. Note that in this research, we are not concerned about time-complexity or other computational issues. Our goal is to get the most from a limited number of data points, and hence, we care about efficiency.

2.3. Maximum Likelihood Estimation and Learning the Probability Distribution

In the problem of learning a probability distribution, the unknown system is the distribution's functional form. The unknown distribution is defined over the set χ, and inputs to the systems are elements $x \in \chi$. The goal is to introduce an estimator $m(x, \theta)$ that converges to $P(x)$ for all $x \in \chi$ as dataset size approaches infinity ($n \to \infty$).

The MLE's objective function for estimating the probability distribution, using the model $m(x, \theta)$ and the dataset $D_n = \{x_i\}_{i=1}^{n}$, has following form

$$\theta_n = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{n} \log\big(m(x_i, \theta)\big)$$

By dividing the sum by the number of samples and multiplying it by -1, we reach the following equivalent minimization problem. The objective function denotes a loss function called empirical **cross-entropy**.

$$\theta_n = \underset{\theta}{\operatorname{argmin}} \ -\frac{1}{n}\sum_{i=1}^{n} \log\big(m(x_i, \theta)\big)$$

As the size of the dataset grows, convergence to a local optimum happens. For the sake of simplicity and for not dealing with issues of local optimums, we assume our optimization reaches the global optimum and $\lim_{n\to\infty} \theta_n = \theta^*$ where $m(x, \theta^*) = P(x)\ \forall\, x$. Of course, this is true with the assumption that $P \in \mathcal{M}$ (The solution exists in the search domain). From the Central Limit Theorem, we can see the following approximation for the loss function's value.

Theorem 1) Assuming $E(\log m(x, \theta^*))^2 < \infty$, for a sufficiently large number of data points ($n >> 1$), the loss function can be approximated with

$$-\frac{1}{n}\sum_{i=1}^{n} \log\big(m(x_i, \theta)\big) = H(P) + D(\,P||m(x, \theta)\,) + O\left(\frac{C_1}{\sqrt{n}}\right)\mathcal{N}(0,1)$$

Where C_1 is a constant, and, is a function of $var(\log m(x, \theta^*))$. $H(P)$ is the Shannon entropy defined as $H(P) = -\sum_{x\in\chi} p_x \log(p_x)$ [40], and the summation is over the element set χ. $D(\,P||m(x, \theta_n)\,) = \sum_{x\in\chi} p_x \log\left(\frac{p_x}{m(x,\theta)}\right)$ is the Kullback–Leibler (KL) divergence [30] between the actual distribution P and the estimator $m(x, \theta)$.

As the size of the dataset approaches infinity, the error term is getting smaller. Immediately from theorem 1, we can see that KL-divergence is the only component of the loss function that is a function of θ (Model). Hence, minimizing the loss function is equivalent to minimizing $D(\,P\big|\big|m(x, \theta)\big)$. The property of KL-divergence is that it is always positive. Besides, it is equal to zero if and only if $P(x) = m(x, \theta)$ almost anywhere. With KL-divergence equal to zero, the only term remaining in the loss function is $H(P)$, which describes the system's entropy. The convergence speed of loss function to $H(P)$ as the function of dataset size is called the learning curve.

2.4. Learning curve

Learning curves are concerned with the expected value of the loss function with respect to randomness in sampling or algorithm's initialization versus the number of data points. Fixing the size, if we sample infinite time and take the expected error, we reach this error level. It is a function $G_t(n)$: $\mathbb{R}^+ \rightarrow \mathbb{R}$ that takes the size of a dataset as input and outputs the value we should expect for the loss function. For the problem of distribution learning, this function is related to how KL-divergence $D(\, P\|m(x,\theta))$ changes with the number of samples.

From theorem 1, with infinite sample size, we can see the loss function's convergences to the entropy of the underlying distribution. Since underlying distribution changes over time, its entropy changes as well and hence, we added the subscript t to $G_t(n)$ to capture this time-dependency. This function is monotonically decreasing and hence, is invertible. Due to its asymptotic convergence to a bounded value ($H(P_t)$), it has a convex form for large dataset sizes. We further assume that it is continuous and differentiable, meaning that $\frac{\partial G_t(n)}{\partial n} < 0$.

In practice, this function is shown to be predictable for deep learning algorithms [24] and is composed of small data, power-law, and irreducible error regions. In small data regions, the model is not scaling significantly with dataset size. The power-law region is where model performance scales with dataset size. In this region, the function $G_t(n)$ is believed [24] to have a power-law functional form. Lastly, in the irreducible error region, the model's generalization loss value does not improve significantly. Between these regions, the power-law region is the one that we can see improvement in performance as we increase the dataset size.

3. Effectiveness curve and value depreciation

Our ultimate goal is valuing a dataset. However, value is subjective and hard to measure since it depends on the context, problem definition, and implementation. Alternatively, we seek to measure how valuable a dataset is comparing to the value of a baseline dataset. The baseline dataset defines a reference point and creates a base of comparison. It is as if we know what the value of the baseline dataset is, and given this value, we want to see how it changes over time.

For prediction at time 0, we fix all other attributes and characteristics of both datasets and only compare them with respect to their sampling time. We define the baseline dataset to be the one that has been sampled independently from P_0. Our dataset is indeed sampled independently, but its sampling distribution is P_t, which is not identical to P_0.

A good starting point would be to compare a dataset of infinite size and see how well-performing it is when predicting P_0. It is particularly important since we expect the infinite size to be helpful in reaching ultimate algorithm's performance. Proposition 1 investigates it.

Proposition 1) Assuming $P_t(x) \neq P_0(x)$, a dataset of infinite size from the wrong distribution $P_t(x)$ has limited learning power at time 0, and a dataset of bounded size from the right distribution $P_0(x)$ reaches the same loss function value.

The argument in proposition 1 is that in the training phase, due to change in the probability distribution, $m(x, \theta)$ convergences to the wrong distribution $P_t(x)$. Therefore, MLE's loss function has an additional term $D(P_0||P_t)$ besides the Shannon entropy $H(P_0)$. It is as if we used a dataset of bounded size from $P_0(x)$, and due to its limited size, we did not reach the ultimate performance. Ultimate performance is reached when MLE's loss function is equal to $H(P_0)$.

This proposition is particularly important for practitioners and also in academic antitrust debates. Proposition states that curating super large datasets does not create a significant barrier to entry if the underlying distribution changes. In our interviews with practitioners, we always found them hopeful that increasing dataset size can compensate for the shortcomings in scaling. Besides, they believe that super large datasets created a barrier to entry advantage for big data companies.

In contrast to their views, this proposition suggests a bound on the achievable performance, no matter the size of a dataset. It says that even infinite dataset size has bounded performance if the underlying distribution is different. We will see that we can answer more questions with counter-intuitive solutions by building on this foundation. For example, suppose someone offers a dataset from the past to be added to our current dataset. It is essential to see how effective it is in improving the quality of the service. We will later argue through the sequential offloading algorithm that adding an old dataset may sometimes hurt performance and put a business at a disadvantage.

Something lacking from proposition 1 is that it talks about loss value, which is not very informative in making comparisons. It is not informative because we do not know how to interpret the excess loss value term $D(P_0||P_t)$. We just know that it is positive, and therefore, the loss value should be bigger than the one for the baseline dataset. To solve this issue, we use the learning curve inverse function to translate the loss function back into the dataset size. Dataset size is easy to understand and compare.

Recall that learning curve at time zero $G_0(n)$ is a monotone function and therefore has an inverse. Using the inverse of the learning curve $G_0^{-1}(.)$, we can find the expected size of a dataset from time zero with an equivalent MLE loss value. Briefly, what we do to form the equivalent size is to first train a model on data sampled from $P_t(x)$. Then, we use the trained model to find the loss value on the data that has been sampled from $P_0(x)$. Finally, we use the function $G_0^{-1}(.)$ to see what size of the data from $P_0(x)$ could have generated similar loss values. This is the basis for our definition of equivalent size.

Definition 1) Dataset $D_{n,t}$ has the **equivalent size** $\bar{n}_{D_{n,t}}$at time 0:

$$\bar{n}_{D_{n,t}} = E_{\theta_{n,t}}\left(G_0^{-1}\left(-E_{P_0}\left(\log\ m\left(x,\theta_{n,t}\right)\right)\right)\right)$$

Where $\theta_{n,t}$ is the solution to:

$$\theta_{n,t} = \underset{\theta}{\operatorname{argmin}} - \frac{1}{|D_{n,t}|}\sum_{x \in D_{n,t}} \log\left(m(x,\theta)\right)$$

In this definition, there exist two expectations. The first one is inside $G_0^{-1}(.)$ and measures the expected model's loss over the test set. The second one is the outer expectation and calculates the expectation with respect to randomness in the algorithm's initializations and steps. In practice, we can approximate the outer expectation by deriving $\theta_{n,t}$ multiple times. Using averaging limits, we can calculate the equivalence empirically in the following way

$$\lim_{k\to\infty}\frac{1}{k}\sum_{j=1}^{k}\left(G_0^{-1}\left(\lim_{\mathrm{l}\to\infty} -\frac{1}{l}\sum_{i=1}^{l}\log\left(m\left(x_i,\theta_{n,t}^{(j)}\right)\right)\right)\right)$$

Where $x_i \sim P_0(x)$ and the outer sum is over multiple runs of the algorithm. For a fairly large number of testing data points, the inner expectation converges. Using theorem 1 to simplify the definition further, we have

$$\bar{n}_{D_{n,t}} = E\left(G_0^{-1}\left(H(P_0) + D\left(P_0||m\left(x,\theta_{n,t}\right)\right)\right)\right)$$

Letting $n \to \infty$ eliminates algorithms' initialization issues as well as other types of randomness and hence, $m\left(x,\theta_{n,t}\right) \to P_t(x)$. Therefore, in the limit

$$\bar{n}_{D_{\infty,t}} = G_0^{-1}\left(H(P_0) + D(P_0||P_t)\right)$$

It is in agreement with proposition 1 where it argues that $\bar{n}_{D_{\infty,t}} < \infty$ if $P_0(x) \neq P_t(x)$.

Notice that equivalent size is a function of the algorithm as well as dataset itself. Dependence on the algorithm is recognized through the inverse function $G_0^{-1}(.)$. It means that the algorithm's power in scaling with dataset size shapes the effectiveness of a dataset. The following example makes it clear. Suppose we have a very large dataset, but we do not use it to train a model. In that case, the sampling time is not essential and, regardless of time, the dataset is as effective as not having it in the first place ($n = 0$). On the other hand, if the algorithm scales fast in the number of data points, a small dataset from $P_0(x)$ can reach $H(P_0) + D(P_0||P_t)$, which means $\bar{n}_{D_{\infty,t}}$ is indeed small.

Definition 2) **Effectiveness** of dataset $D_{n,t}$ is defined as $E_{D_{n,t}} = \frac{\bar{n}_{D_{n,t}}}{n}$.

Intuitively it should always be between zero and one, i.e., $E_{D_{n,t}} \in [0,1]$. 1 means that the given dataset's value is equal to the value of the baseline dataset. 0 means that data is worthless compared to the baseline dataset. The more perishable the data (which means it loses its relevance to the prediction problem quicker), the less the effectiveness. For example, if the effectiveness is equal to 0.8, we say that the dataset lost 20% of its effective size.

Proposition 1 argues that effectiveness of $E_{D_{\infty,t}} = 0$ if $P_0(x) \neq P_t(x)$. It is because $\bar{n}_{D_{n,t}}$ remains bounded and therefore, $\lim_{n\to\infty} \frac{\bar{n}_{D_{n,t}}}{n} = 0$

Definition 3) **Substitution curve** is a function $f_n(t_1, t_2): \mathbb{R}^2 \to \mathbb{R}$ and is defined as

$$f_n(t_1, t_2) = \frac{\bar{n}_{D_{n,t_1}}}{\bar{n}_{D_{n,t_2}}}$$

It shows how well we will be off in terms of effectiveness if we substitute a dataset of size n from time t_2 with a dataset of the same size that has been sampled at time t_1. Note that choosing $t_2 = 0$ brings us back to the definition of effectiveness. Using theorem 1, the substitution curve has following formulation

$$f_n(t_1, t_2) = \frac{\bar{n}_{D_{n,t_1}}}{\bar{n}_{D_{n,t_2}}} = \frac{E\left(G_0^{-1}\left(H(P_0) + D\left(P_0||m\left(x, \theta_{n,t_1}\right)\right)\right)\right)}{E\left(G_0^{-1}\left(H(P_0) + D\left(P_0||m\left(x, \theta_{n,t_2}\right)\right)\right)\right)}$$

Theorem 2) Substitution curve has the following properties.

a) It is non-negative and bounded.
b) It is a monotonic function of n.
c) It is converging to a substitution frontier

$$\lim_{n\to\infty} f_n(t_1, t_2) = \frac{\bar{n}_{D_{\infty,t_1}}}{\bar{n}_{D_{\infty,t_2}}} = \frac{G_0^{-1}\left(H(P_0) + D\left(P_0||P_{t_1}\right)\right)}{G_0^{-1}\left(H(P_0) + D\left(P_0||P_{t_2}\right)\right)}$$

Nonnegativity and boundedness are immediate. It is nonnegative because function G_0^{-1} is non-negative by definition. Boundedness is also immediate from proposition 1, because for $i \in \{1,2\}$ and $t_i \neq 0$, $0 < \bar{n}_{D_{n,t_i}} < \infty$ for all n.

The substation curve is an important definition in this paper. It is a building block for the argument we make in the next section on the effectiveness of datasets gathered over a long time. The concept will be used in the sequential offloading algorithm used in the next session.

Assuming a monotonic decline in the value of data over time, figure 1 depicts examples of substitution curves $f_n(t,1)$. Each curve represents the substitution gain for different dataset sizes when the substitution time is fixed at $(t_1, t_2) = (t, 1)$. $f_\infty(t,1)$ is the frontier. This is a direct result of theorem 2 on the substitution function's monotonicity on n and its convergence to the frontier. Building further on this result, in Appendix B, we empirically measure the substitution curve for our experiment in this paper and show that it increases in n for $t_1 > t_2$ and decreases for $t_1 < t_2$.

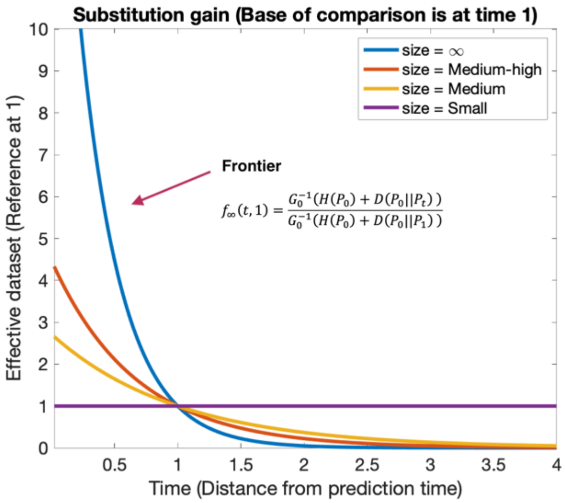


Figure 1) Substitution curves for different sizes of datasets. The frontier is marked as blue. It shows the maximum depreciation in substituting datasets of time 1 with a dataset of any other time.

As apparent in this figure, we do not gain much in substituting data from different times for very small dataset sizes. It is because small datasets do not provide significant scaling in performance, and hence, it does not matter when they were sampled. This behavior is mostly seen in the small data region of the learning curve. For medium dataset sizes, when we are in the learning curve's power-law region, we gradually see significant gains in substituting dataset sizes from different times. Increasing dataset size in the power-law region brings us to the medium-high dataset size regime. This region will be used in our experiments (In later sections) to measure perishability. Finally, the infinite dataset size speaks of the irreducible error region and higher sensitivity to substitution.

4. Datasets collected over time

So far, we studied the effectiveness of a dataset that has been sampled at a given time t. Nevertheless, most datasets are collected over time, and there is a need to study their effectiveness. Notation wise, we show these datasets with $D_{n,[t_1,t_2],\lambda_t}$, where it represents a dataset with size n that is collected during the period $[t_1, t_2]$. λ_t shows the proportion of samples that have been collected at time t.

Despite the change in notation and the nature of the problem, the question on its effectiveness still can be answered with the tools we developed so far. Still, dataset $D_{n,[t_1,t_2],\lambda_t}$ has a “Net Distribution” that can be used to measure its effectiveness. It is like the mixture of underlying distributions over time, and hence, to calculate it, we need a dataset of infinite size. Net distribution is a function of $[t_1, t_2]$, and λ_t.

In this paper, we focus on datasets of the form $D_{n,[0,t],\lambda_t}$ where the sampling time $t_1 = 0$. It is because they provide better generalization intuition. Besides, it is easy to turn a bigger period of time into smaller periods with the function λ_t. For example, the dataset $D_{n,[t_1,t_2],\hat{\lambda}_t}$ is equivalent to dataset $D_{n,[0,t_2],\lambda_t}$ with λ_t equal to

$$\lambda_t = \begin{cases} 0 & t < t_1 \\ \hat{\lambda}_t & t_1 \leq t \leq t_2 \end{cases}$$

Lemma 1) Net distribution of dataset $D_{n,[0,t],\lambda_t}$ is equal to

$$P_{[0,t],\lambda_t}(x) = \int_0^t P_s(x)\lambda_s ds$$

Lemma 1 states that Net distribution is the convex combination of all distribution from time 0 to t with weights $\lambda_t \in [0,1]$ & $\int_0^t \lambda_s ds = 1$.

As this lemma states, the net distribution is not necessarily equal to P_0. Therefore, using proposition 1, we argue that datasets curated over a period of time have limited substance. The following proposition investigates the effectiveness of these datasets.

Proposition 2) There exists an equivalent time $t^* \in [0, t]$ such that the dataset $D_{n,[0,t],\lambda_t}$ provides an equivalent loss value to the dataset D_{n,t^*} i.e. $\bar{n}_{D_{n,[0,t]}} = \bar{n}_{D_{n,t^*}}$. The solution is unique when decline in value of data is monotonic.

Proposition 2 is the key to understanding the next subsection on sequential offloading. As much as it is important to understand what it says, it is also important to realize what it does not. It does not say that the

"Net distribution" is equal to P_{t^*}. Net distribution is a combination of many distributions, including P_{t^*}, and therefore, it is not necessarily equal to P_{t^*}. Instead, proposition 2 suggests that $P_{[0,t],\lambda_t}$ and P_{t^*} are in a way that they make equal KL divergences with P_0, i.e. $D\left(P_0||P_{[0,t],\lambda_t}\right) = D(P_0||P_{t^*})$. Consequently, they produce equivalent MLE loss value, which means $\bar{n}_{D_{n,[0,t],\lambda_t}} = \bar{n}_{D_{n,t^*}}$.

Note that having t^* between zero and t is important in this proposition. The emphasis is on the fact that the period $[0, t]$ starts from time 0. Even if the dataset has been sampled from $[t_1, t]$, still, $t^* \in [0, t]$. It is because, for the dataset ($D_{n,[t_1,t],\lambda_t}$) where $0 < t_1 < t$, there might exist a sampling density λ_t such that it makes the Net distribution $P_{[0,t],\lambda_t}(x) = P_0(x)$ for all $x \in \chi$. It means that in this example $t^* = 0 \notin [t_1, t]$.

The most exciting thing about this theorem is that $t^* < t$. If we deliberately delete the portion $[t_1, t]$ from the dataset where $t_1 < t^*$, despite losing size, the remaining dataset ($D_{n_1,[0,t_1],\lambda_t}$) will have a new equivalent time t^{**} which is $t^{**} < t^*$. In other words, the dataset gained relevance.

4.1. Sequential offloading

The idea of sequential offloading is founded in increasing the value of a dataset by reducing its size. It looks to be counter-intuitive, but in a time-dependent context, data perish quickly, and it may be beneficial to discard useless information. Clearly, deleting old data means loss of dataset size, which is a bad thing. Nevertheless, gaining relevance may offset the loss of dataset size, and deletion likely improves the overall effectiveness.

The idea is centered around proposition 2 and theorem 2. Proposition 2 states that for a dataset $D_{n,[0,t],\lambda_t}$ there exist a time $t^* \in [0, t]$ such that $\bar{n}_{D_{n,[0,t],\lambda_t}} = \bar{n}_{D_{n,t^*}}$. By deleting data $[t^*, t]$ from the dataset, we end up with a smaller size n_0, but the equivalent time shifts from t^* to $t^{**} \in [0, t^*]$, which is more relevant. If the substitution gain is higher than the lost size due to deletion, it means we gained from deletion i.e.

$$f_{n-n_0}(t^{**}, t^*) > \frac{n}{n - n_0} \Rightarrow \bar{n}_{D_{n,[0,t],\lambda_t}} < \bar{n}_{D_{n-n_0,[0,t^*],\lambda_t}}$$

Where n_0 is the size that has been deleted from the dataset. Algorithm 1 Formalizes the sequential offloading.

Algorithm 1) Sequential offloading algorithm

```
Given dataset $D_{n,[0,t],\lambda_t}$, substitution gain function $f_n(t_1, t_2)$
$i = 1$
$t^{(0)} = t$
$n^{(0)} = n$
While (Gain is possible)
    Find $t^*$ as explained it theorem 2 and call $t^{(i)}$
    $n^{(i)} = n^{(i-1)} \int_{t^{(i)}}^{t^{(i-1)}} \lambda_t \, dt$
    Delete sampled data $[t^{(i)}, t^{(i-1)}]$ from $D_{n^{(i-1)},[0,t^{(i-1)}],\lambda_t}$ and call it $D_{n^{(i)},[0,t^{(i)}],\lambda_t}$
    if $\left(f_{n^{(i)}-n^{(i-1)}}\left(t^{(i)}, t^{(i-1)}\right) > \frac{n^{(i-1)}}{n^{(i)}-n^{(i-1)}}\right)$
            Gain is possible
            $i = i + 1$
    $else$
            Gain is not possible
     end
end
```

This algorithm stops when there is no gain in deleting old data. It also opens a philosophical question on what a successful iteration means for the data. A successful iteration means $\bar{n}_{D_{n,[0,t]}} < \bar{n}_{D_{n-n_0,[0,t^*]}}$ and hence, there is positive improvement upon losing a portion of data. Therefore, as the following corollary states, old data actually did put us in a disadvantageous position.

Corollary 1) A dataset collected in a long period of time may weakly create a disadvantage for a firm.

5. Experimental Design

Our goal in this section is to measure effectiveness and thereby perishability empirically. In other words, after training an algorithm with data that has been sampled on one stationary period, we measure its performance at any other time. In addition, we would like to observe a monotonic decline in the value of a dataset. The monotonic decline in the dataset's value guarantees a unique solution to theorem 2. This claim was supported in the framework section by arguing that the birth of new elements and the death of old elements decrease the dataset's relevance over time. Because of that, we should expect monotonic behavior. However, in fashion or any periodic source of data generation, we should expect the return of old elements, and hence, we may have complications in arguing monotonicity. We measure perishability and observe partial monotonicity in the natural language processing context and for the next word prediction task. Partial monotonicity means that the effectiveness curve has an overall declining form. Nevertheless, it has a small periodicity.

We chose language modeling because its datasets tend to be the largest and most easily collected in machine learning. They are easily collected because language modeling is an unsupervised task; The model tries to predict the next word or masked words in a given sentence, so each text sample does not need to be labeled. Further, the language modeling task is currently used as a common pre-training objective for many other language tasks [45]. Thus, we choose language modeling as the target task and seek a large corpus of English language data. In this section, we first explain data and how we process it for the task. Then, we explain the algorithm and model architecture, and lastly, we present the measurements.

5.1. Data Collection and Processing

Our challenge is to find a large enough dataset that has been collected over a long period of time. It is because text processing algorithms require large training set sizes to have significant improvement in quality. In addition, we need this dataset to be sampled over a long period of time to let us make an observable perishability measurement. From a technical standpoint, the dataset must be large enough to reliably measure the power-law portion of the learning curves associated with each time period. Thus, the dataset must span roughly two orders of magnitude in size larger than the smallest dataset in the power-law region. Prior results show that, for language modeling, the smallest such dataset is at most 1 million words [24]. Consequently, the dataset should contain roughly 10-100 million words per time period.

We choose the Reddit post dataset as it fits our needs. This data was collected and used in [19]. It is a collection of posts and comments from the years 2006 to 2018 and was scraped from Reddit between September 2016 and July 2018. We preprocessed the dataset to create flat text files with the following format:

```
Title (6): What was the biggest scandal in your school?
Text:
Comment (4): Vampires. This was almost 6 years ago now at my
high school, but vampires. Do a quick...
Comment (3): Not sure if I'd call it a "scandal," but when I
was in college...
Comment (2): Freshman year a friend of mine found a paper bag
at the bus stop full of money - and it...
```

‘Title’ is the title that the author specified when posting the submission, and ‘Text’ is an optional field of body text associated with the post. After the post, each line is a comment from other users designated by

‘Comment’. Comments only contain text. The values in parenthesis are submission or comment scores based on upvotes or downvotes given to each by users. We filtered out posts and comments with scores less than 2.

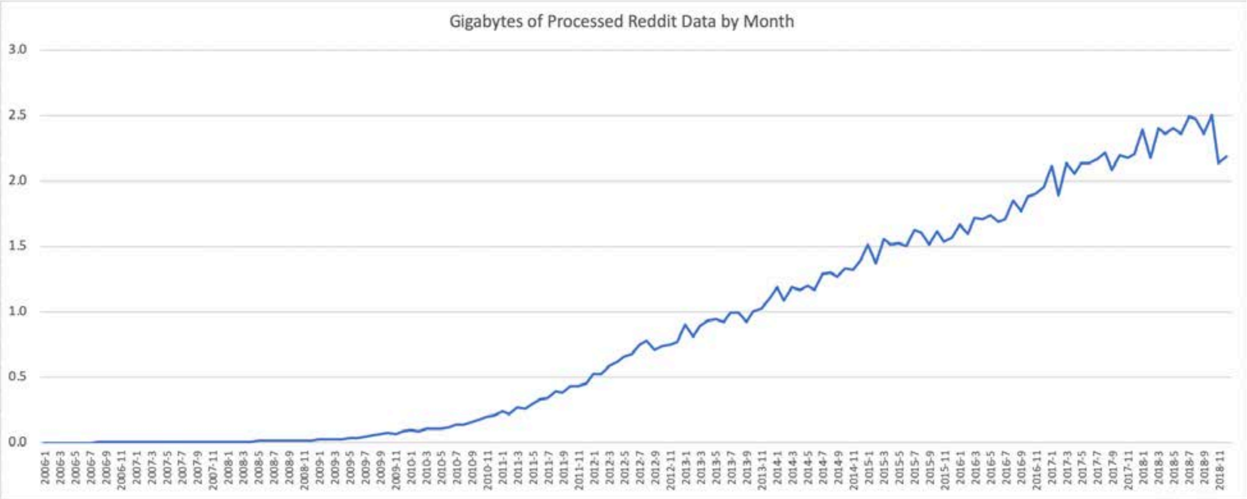


Figure 2) Size of datasets processed for each month. For example, for July 2013, 1 Gigabyte of text data is processed. This is not a cumulative dataset size. The growth in the size shows the growth in the number of topics discussed, the number of users as well as their engagement.

To evaluate how data distributions and value shifts over time, we split the dataset into chunks based on the timestamp of the submissions and comments. We aim for 100 million words per time period, so we group data until each split is at least that large. Specially, we group posts and comments into the following periods: the years 2006-2009, January-June 2010, July-December 2010, January-March 2011, April-June 2011, July-September 2011, October-December 2011, and then monthly for the years 2012-2018. Earlier years of Reddit dataset have less data because the platform was becoming established and growing, so we had to group more extended periods together. Figure 2 shows the amount of data we processed each month.

Finally, we subdivide the data from each time period to form a standard machine learning training and testing setup for collecting learning curves. First, we randomly sample and split the posts (and their comments) into training, development/validation, and test/evaluation subsets. The development and test sets are at least 2 million words each. The development set is used to validate that the model is learning to generalize during training and to early-stop training when the model performs the best on the development set. The test set is used after training to evaluate how well the training is done. We use these test sets to cross-evaluate models trained on data from other periods. The model never trains on these subsets.

After splitting out the development and test sets, we randomly shuffled the remaining data as the full training set for the time period. We subdivide this training set into chunks of exponentially increasing size by factors of 2. Empirically, we find that datasets of size 1.25 million words are large enough to be in the power-law portion of learning curves, so we break the training set into successively overlapping subsets of size 40 million, 20 million, 10 million, 5 million, 2.5 million, 1.25 million words by taking the first half of the prior subset. We train separate models on each training subset to collect how models generalize as they are allowed to train with increasing dataset size. The resulting data size-generalizability curves are learning curves for the time period.

5.2. Model Architecture and Training Process

We chose to train current state-of-the-art language models on the data to collect their generalization error and learning curves. Specifically, we train GPT-2, the Generative Pre-Training transformer-based model from OpenAI [22, 36]. Collecting learning curves can be costly due to the training time required to train large models on each of the training subsets. We chose to train a small variant of GPT-2 that was expected to be large enough (i.e., sufficient parameters) to overfit all of the training set splits and yet small enough to train in a reasonable amount of time---at most about 32 hours per training subset on a single GPU. We configure our GPT-2 model variant as follows: Vocabulary size 50257 sub-word tokens, maximum sequence length 512 tokens, depth 6 transformer blocks each with 8 self-attention heads and hidden dimension 512. The model has 44.9 million parameters total—a rule of thumb in language modeling is to use a model with as many parameters as words in the largest dataset.

We train the models using the Adam optimizer with a static learning rate of 2e-4 and with batch sizes 12 and 24. The training objective is the cross-entropy loss of the model's prediction of the probability of the target next token in the input sentence. We empirically find that changing the batch size marginally changes the final loss (<0.3% change in cross-entropy), so we do not further explore optimization hyperparameters to mitigate total training time. Finally, we validate the models using the development dataset every 50-200 training steps, depending on the size of the dataset—smaller datasets require fewer training steps for the model to converge. We early-stop training when the development set loss stops improving for more than 15 validation runs.

5.3. Evaluation Process and Effective Dataset Size

Our objective is to measure how much the data distribution has changed over time. In that cause, we evaluable how well a dataset that has been sampled from one time period, can predict values for each other time period’s data. To do so, we train a model and evaluate its test error for each time period over multiple time periods. Furthermore, we characterize the learning curves so that we can translate measured test errors back to equivalent dataset sizes. Finally, we present the effectiveness curve.

In the training phase, we first find the finest model for each time period and each dataset size. The finest model is the one that achieves a smaller development set loss. Its selection process mimics the way models are chosen for deployment in AI-enabled products. To find the finest model, at each training run, we validate the models on the given time period's development set and choose the model weights that achieve smaller development set loss. When we test with multiple different batch sizes, the finest model is the one that achieves superior performance in separate training runs for the given time period and training set size.

We collect the finest model for each training set size ranging from 1.25 to 40 million words. Doing so allows us to construct learning curves across different time periods. We cross-evaluate all finest models—one for each time period and training set size—by evaluating them on the test sets for all other time periods. We use these results to curve fit learning curves and indirectly calculate its inverse: Given finest models for the time period t_1, and their evaluation scores for the time period, t_0 (t_0 can be equal to t_1), these scores will be used to show how increasing the training set size from period t_1 might improve prediction accuracy for the time period t_0. We curve fit learning curves with power-laws.

Figure 3 shows examples of learning curves for models trained at different times. Each curve shows a model that has been trained on a specific time-period. The learning curves are different from each other and form parallel curves. The offset is due to change in the entropy $H(P)$, which is different at different times. Earlier models like those that have been trained in 2010 have lower values than the model of 2018. To answer why this is happening, we should look at figure 2. As apparent from figure 2, the dataset size per month is growing, which is a clear sign of the increase in the contribution and growth of the user base. This growth adds to the diversity in topics as well as language styles. The more diverse the dataset, the higher its entropy. It is also apparent from this graph that the learning curve is a decreasing function, and hence, more data causes lower cross-entropy value.

Figure 4 shows test evolution results for models trained on different time periods. Training size is fixed, and the algorithm is trained on data from a few time-periods. Periods are shown in the legend section of this figure. Each point in this graph is the evaluation result of a training and test pair and curves are made by joining pairs with similar training time. For example, the blue curve shows the finest model's test results that have been trained 2006-2009 and tested on every other time.

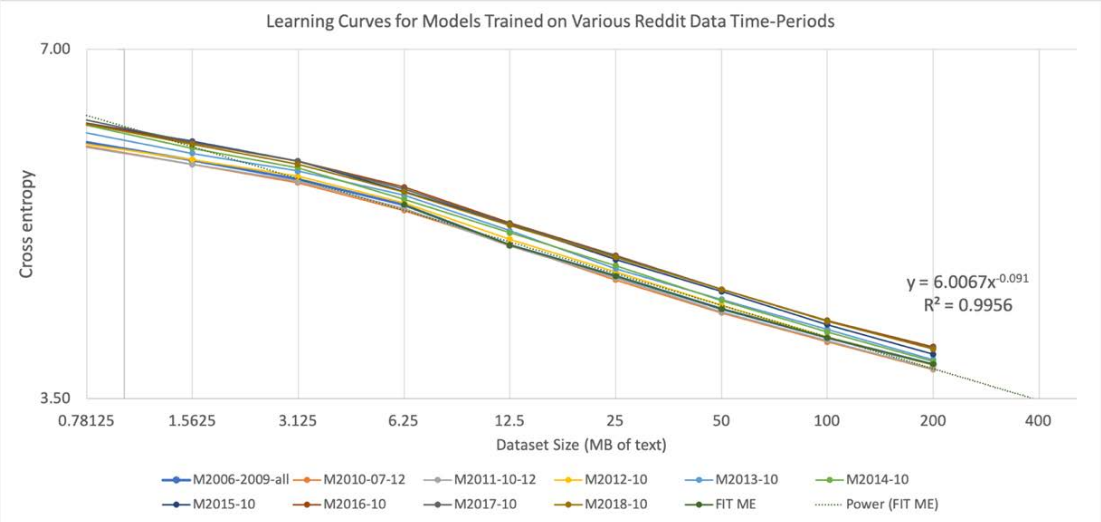


Figure 3) Measured learning curves for models that have been trained at different times. The x-axis is in the log-scale and shows the dataset size. Y-axis is the cross-entropy value. The legend describes the time we used to train these models. For example, the yellow curve shows a model that has been trained on data from October 2012.

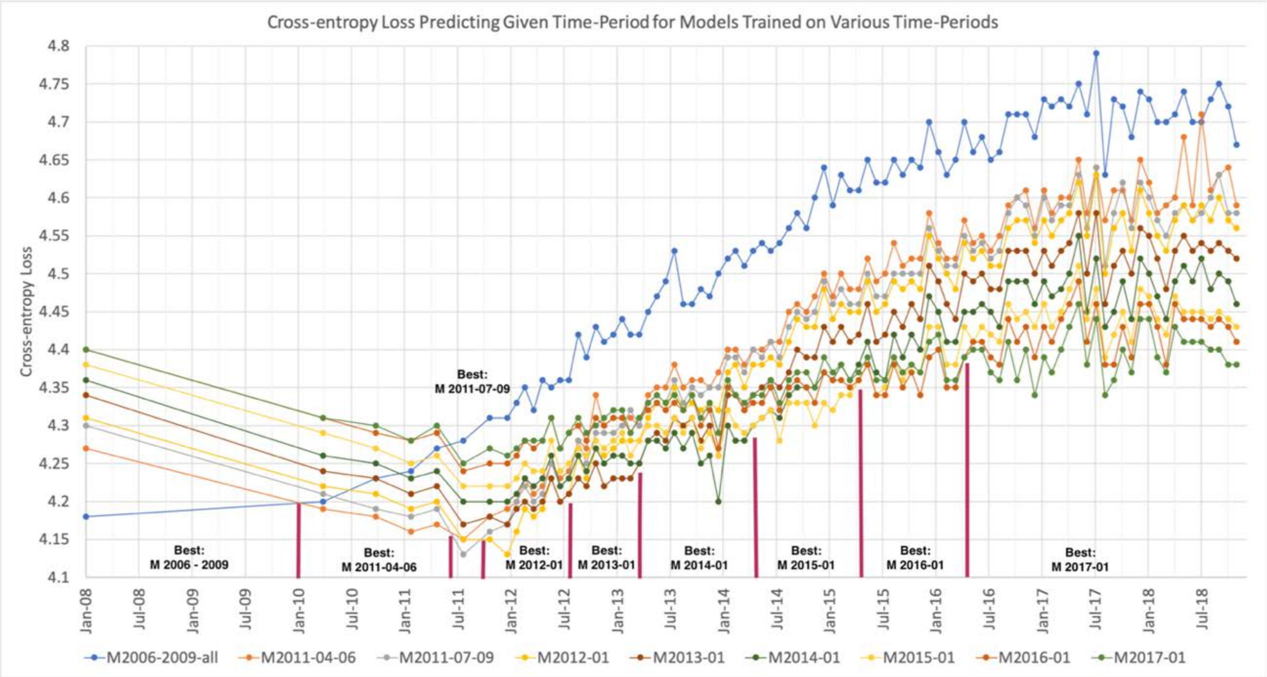


Figure 4) Cross entropy loss value when we use a model that has been trained on year z (each curve) and is tested on data from year x (x-axis). Y-axis is the cross-entropy loss. The legend describes the time we used to train these models. For example, the green curve shows a model that has been trained on data from January 2014. The best cross-entropy loss in each time period is mentioned in this graph as well.

The first observation is that the best model for prediction in t_0 is the one trained on data from t_0. As an example, before January 2010, the model that has been trained on data from 2006-2009 ($m_{2006-2009}$) has the lowest cross-entropy and hence, has the best predicting power compared to other curves. In contrast, from January 2010 to June 2011, the April 2010's model ($m_{2010-04-06}$) is the best performer replacing the blue curve. It immediately shows perishability. It is because the best performing model at one period loses its power as we move away from its sampling time. Despite apparent perishability, as time goes by, we see an increase in the cross-entropy values across all models. It is again due to the increase in the diversity of topics in Reddit data over time. In other words, the entropy function $H(P)$ is increasing.

Finally, we invert these learning curves to estimate the equivalent dataset size from time period t_1 when predicting for the time period t_0. Start with the best model, $m_{t_1,50\mathrm{M}}$, for time period t_1 trained on 50 million words, for example. Evaluate $m_{t_1,50\mathrm{M}}$ to collect cross-entropy loss for time period t_0. Now use the learning curve for models trained and tested on time period t_0 to estimate how much training data from time period t_0 is required to achieve that cross-entropy loss. Suppose the inverted learning curve yields 40 million words required in time period t_0, then the equivalent dataset size from time period t_1 is 40 million words at time t_0, or it is effectively 80% of its time t_1 size.

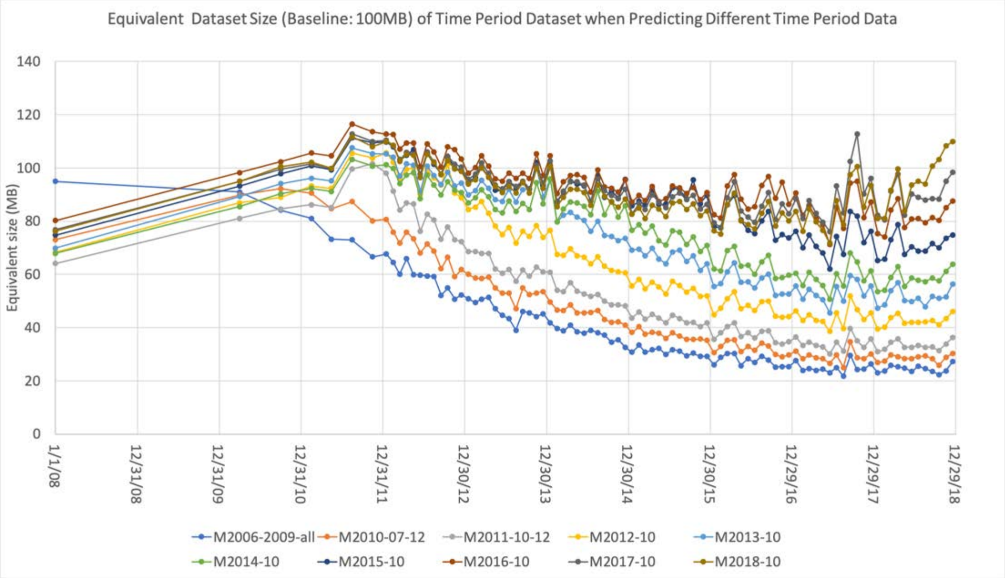


Figure 5) Equivalent sizes over time (x-axis) when we used 100MB of data in the training phase. Each curve is the trained model. The legend describes the time we used to train these models. For example, the yellow curve shows a model that has been trained on data from October 2012.

Figure 5 shows the equivalent dataset sizes for models trained on the 100MB of data sampled from different times. We chose 100MB for this graph to make it easier for readers to convert values to percentages. As seen in this figure, for periods after sampling time, the equivalent sizes are monotonically decreasing. Despite overall monotonicity, we need to answer two questions about this graph:

1. Why do we observe higher equivalence variability on curves with higher equivalence (Closer to 100MB) sizes?

2. Why do we, on some occasions, observe a sudden increase in all equivalence curves?

For the first question, we believe it happens due to numerical errors in the inversion of learning curves. As we see in figure 3, learning curves have power-law functional forms. Hence, in different regions of the learning curve, small change in measured cross-entropy translates to different magnitudes of change in equivalent sizes. For example, in figure 3, if the training size is 100MB with measured cross-entropy of 5, the equivalent size is roughly 25MB. A small change of 0.1 in the measured cross-entropy translates to an equivalent size of roughly 20MB, which is 5MB different from the previous measurement. However, a similar small change, when the cross-entropy is 4, makes the difference of roughly 50MB. Therefore, the closer the equivalent size is to the training size, the smaller the error causes a higher variability. This also explains the overshoots of later models (2017 and 2018) in equivalent sizes in August 2017.

For the second question, aside from the test set's sampling issues, model errors, and numerical error in fitting the learning curve's functional, we believe it is natural for events on those occasions to be slightly more predictable by all models. For example, for August 2017, if we look at the predictive power of $m_{2006-2009}$, we cannot find a considerable change, and sudden increase looks normal. However, due to the magnification of error and variability in later models (Models with sampling time closer to 2017), we see considerable changes in their equivalence values that sometimes lead to overshoots above 100MB.

At last, figure 6 shows the effectiveness curves. To deal with issues of the sudden increase in equivalent sizes, we made a slight alteration on the way we calculate the effectiveness curve. In this way, since theoretically $\bar{n}_{D_{n,0}} = n$, we calculate $E_{n,t} = \frac{\bar{n}_{D_{n,t}}}{n} = \frac{\bar{n}_{D_{n,t}}}{\bar{n}_{D_{n,0}}}$. In other words, instead of dividing the equivalent size of time t to 100MB, we divide it by the measured equivalent size of test time. It is as if we divide the measured value by the value of the best model predicting the test time. Doing this process over models from a few time periods creates figure 6.

In this figure, we can confirm a monotonic decrease of the effectiveness curve. It is interesting to see that the effectiveness curves of models from different times are all lined up. As this graph shows, roughly around 7 to 8 years, the value of data for the algorithm and the next word prediction task drops 50%. Furthermore, we can see small periodic behavior in the measurements. For example, looking at the values of days 365, 730, and 1095 and comparing them with the values of days 181, 550, and 915, we can see small ripples in the overall form of effectiveness functional. It suggests small periodicity in the data.

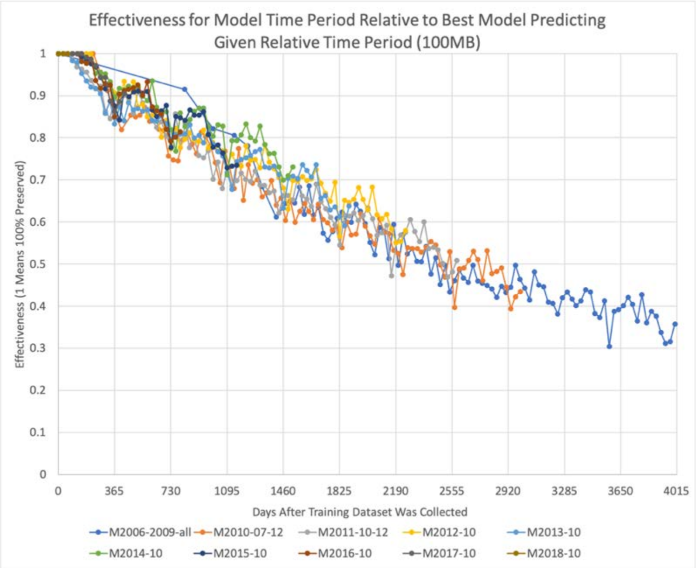


Figure 6) Effectiveness curve. The X-axis shows the number of days after the training dataset was collected. The Y-axis shows the effectiveness of the trained model. 1 means that 100% of the dataset's value has persevered. The legend describes the time we used to train these models. For example, the yellow curve shows a model that has been trained on data from October 2012.

6. Discussion and Conclusions

An increase in the size of a dataset, with independent and identically distributed samples, improves the generalizability of trained models. This improvement increases the quality of algorithms. Using this fact, economists and data scientists argued that having more data (Weakly) improves the quality of AI-based products and services.

For long, this argument triggered debates on whether data volume owned by big tech companies creates a barrier to entry and, hence, deters the entry of new firms. Those in support argue that the data network effect creates a winner take all situation. Hence, having a sufficiently massive amount of data, compared with competitors, pushes competing firms off the market. Besides, this argument suggests first-mover advantage in AI-based businesses meaning that firms who adopt the technology earlier can curate bigger datasets and have a better competitive position.

In refute, researchers cite the diminishing return to scale of dataset size in the algorithm's performance. They argue that data alone cannot contribute much to an algorithm’s performance, and there is a limit to its power. In practice, once a model reaches its ultimate performance, companies propose new and more complicated models that need even more data. For example, using deep learners, companies can always increase the number of layers as well as the number of neurons per layer to create more powerful models. This change makes the diminishing return to scale argument weak.

Seeking another direction, in our paper, we argue that time-dependency, despite having a significant effect, is neglected in the debates. We believe that it plays a crucial role in determining the importance of data in AI-based businesses. The change over time is justified with innovation in products and services' space as well as the change in consumers' taste and behavior. Because of innovation, we believe that data distribution is different in the future from any combination of distributions in the past, meaning that older datasets may not be relevant enough to problems in hand. The lack of relevance means that data loses its effectiveness in creating value. By means of an experiment, we empirically showed a semi-monotonic decline in the value of a dataset.

Having the shift in distribution over time, we theoretically proved that even an infinite size of time-dependent data has limited power in predicting the future, which means a dataset of bounded size from the right distribution can reach a similar performance level. Therefore, the bounded effective size attests to the importance of fresh data and dismisses the diminishing value of new datapoints. Further, the bounded equivalent size of data puts a limit on the importance of data in creating a barrier to entry.

Our argument supports recent research conducted by [6] and [14]. Notably, we believe that both the search engine and advertisement businesses face a high level of time-dependency, and hence, the value of their data perishes very quickly. As explained in the literature review, [6] considered the value of data in advertisement and [14] research value of online search data.

We move one step forward and, through sequential offloading, argue that mass of data may even put a firm in a disadvantageous position. For more clarification, consider a case where data perishes extremely fast. In this case, training an algorithm on the old data frequently produces irrelevant outcomes and leads to user frustration. This proposition cast doubts on arguments supporting the first-mover advantage in AI-business. It means that, in businesses with highly perishable data, being a first mover is not necessarily an advantage, and using the entire curated dataset creates a disadvantage.

We can extend our arguments and results to any data property that can be modeled by the underlying distribution. It is because all our definitions, theorems, and propositions are a function of variation in the distributions. For example, we may extend this result to measure the value loss in the user dimension. In other words, we may model the heterogeneity in preferences across users by variation in their preference distributions. Then, we measure the value of a user's data on predicting other user's preferences. In this paper, we chose to center our arguments around the change over time since it is easier to visualize. Besides, experimenting over time dimension has the benefit of having a semi-monotonic decline in the value of data.

References:

1. Abrardi, L., Cambini, C., and Rondi, L., 2019. The economics of Artificial Intelligence: A survey. *Robert Schuman Centre for Advanced Studies Research Paper No. RSCAS, 58*.
2. Acemoglu, D., Makhdoumi, A., Malekian, A., and Ozdaglar, A., 2019. *Too much data: Prices and inefficiencies in data markets* (No. w26296). National Bureau of Economic Research.
3. Aghion, P., Jones, B.F., and Jones, C.I., 2017. *Artificial intelligence and economic growth* (No. w23928). National Bureau of Economic Research.
4. Agrawal, A., Gans, J., and Goldfarb, A., 2019. Economic policy for artificial intelligence. *Innovation Policy and the Economy*, *19*(1), pp.139-159.
5. Agrawal, A., Gans, J., and Goldfarb, A., 2018. *Prediction machines: the simple economics of artificial intelligence*. Harvard Business Press.
6. Arnold, R., Marcus, J.S., Petropoulos, G., and Schneider, A., 2018. Is data the new oil? Diminishing returns to scale.
7. Bajari, P., Chernozhukov, V., Hortaçsu, A., and Suzuki, J., 2019, May. The impact of big data on firm performance: An empirical investigation. In *AEA Papers and Proceedings* (Vol. 109, pp. 33-37).
8. Baldwin, R. (2019): *The Globotics Upheaval: Globalization, Robotics, and the Future of Work*. Oxford University Press.
9. Begenau, J., Farboodi, M., and Veldkamp, L., 2018. Big data in finance and the growth of large firms. *Journal of Monetary Economics*, *97*, pp.71-87.
10. Bergemann, D., Bonatti, A. and Gan, T., 2020. The economics of social data.
11. Brynjolfsson, E., Mitchell, T., and Rock, D., 2018, May. What can machines learn and what does it mean for occupations and the economy?. In *AEA Papers and Proceedings* (Vol. 108, pp. 43-47).
12. Carriere-Swallow, M.Y. and Haksar, M.V., 2019. *The economics and implications of data: an integrated perspective*. International Monetary Fund.
13. Casella, G., and Berger, R.L., 2002. *Statistical inference* (Vol. 2, pp. 337-472). Pacific Grove, CA: Duxbury.
14. Chiou, L., and Tucker, C., 2017. *Search engines and data retention: Implications for privacy and antitrust* (No. w23815). National Bureau of Economic Research.
15. Cockburn, I.M., Henderson, R., and Stern, S., 2018. *The impact of artificial intelligence on innovation* (No. w24449). National bureau of economic research.
16. Cowgill, B., and Tucker, C.E., 2020. Algorithmic Fairness and Economics. *The Journal of Economic Perspectives*.
17. Crémer, J., de Montjoye, Y.A. and Schweitzer, H., 2019. Competition policy for the digital era. *Report for the European Commission*.
18. De Corniere, A., and Taylor, G., 2020. Data and Competition: a General Framework with Applications to Mergers, Market Structure, and Privacy Policy.
19. Fan, A., Jernite, Y., Perez, E., Grangier, D., Weston, J., and Auli, M., 2019. Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*.
20. Farboodi, M., Mihet, R., Philippon, T., and Veldkamp, L., 2019, May. Big data and firm dynamics. In *AEA papers and proceedings* (Vol. 109, pp. 38-42).
21. Farboodi, Maryam, and Laura Veldkamp. 2019. “A Growth Model of the Data Economy.” Working Paper, Columbia Business School, New York, June 20.
22. GPT-2 Source Code, OpenAI, 2018-2020 (https://github.com/openai/gpt-2)
23. Gregory, R.W., Henfridsson, O., Kaganer, E., and Kyriakou, H., 2020. The Role of Artificial Intelligence and Data Network Effects for Creating User Value. *Academy of Management Review*, (ja).
24. Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M., Ali, M., Yang, Y. and Zhou, Y., 2017. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*.

25. Holtz, D., Carterette, B., Chandar, P., Nazari, Z., Cramer, H., and Aral, S., 2020. The Engagement-Diversity Connection: Evidence from a Field Experiment on Spotify. *Available at SSRN*.
26. Hagiu, A., and Wright, J., 2020. *Data-enabled learning, network effects and competitive advantage*. Working Paper.
27. Ichihashi, S., 2020. The Economics of Data Externalities.
28. Jones, C.I., and Tonetti, C., 2019. *Nonrivalry and the Economics of Data* (No. w26260). National Bureau of Economic Research.
29. Korinek, A., and Stiglitz, J.E., 2017. *Artificial intelligence and its implications for income distribution and unemployment* (No. w24174). National Bureau of Economic Research.
30. Kullback, S., and Leibler, R.A., 1951. On information and sufficiency. *The annals of mathematical statistics*, *22*(1), pp.79-86.
31. Lambrecht, A., and Tucker, C.E., 2015. Can Big Data protect a firm from competition? *Available at SSRN 2705530*.
32. Milgrom, P.R., and Tadelis, S., 2018. *How artificial intelligence and machine learning can impact market design* (No. w24282). National Bureau of Economic Research.
33. Newman, N., 2014. Search, antitrust, and the economics of the control of user data. *Yale J. on Reg.*, *31*, p.401.
34. Petit, N., 2017. Antitrust and artificial intelligence: a research agenda. *Journal of European Competition Law & Practice*, *8*(6), pp.361-362.
35. Prufer, J. and Schottmüller, C., 2017. Competing with big data.
36. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I., 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, *1*(8), p.9.
37. Reimers, I. and Shiller, B., 2018. Welfare Implications of Proprietary Data Collection: An Application to Telematics in Auto Insurance. *Available at SSRN 3125049*.
38. Rubinfeld, D.L., and Gal, M.S., 2017. Access barriers to big data. *Ariz. L. Rev.*, *59*, p.339.
39. Schaefer, M., Sapi, G., and Lorincz, S., 2018. The effect of big data on recommendation quality: The example of internet search.
40. Shannon, C.E., 1948. A mathematical theory of communication. *The Bell system technical journal*, *27*(3), pp.379-423.
41. Tirole, J., 2020. Competition and the Industrial Challenge for the Digital Age.
42. Van Til, H., van Gorp, N. and Price, K., 2017. Big Data and Competition, *Ecorys*
43. Varian, H., 2018. *Artificial intelligence, economics, and industrial organization* (No. w24839). National Bureau of Economic Research.
44. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S.R., 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Appendix A

Proof of Theorem 1)

Define $v = -\log(m(x,\theta))$. For a given θ and IID $x_i \sim P(x)$, v_i becomes IID samples of random variable v. If $Ev_i^2 < \infty$, for a large number of data points we can use central limit theorem and hence,

$$\frac{1}{n}\sum_{i=1}^{n} v_i = E_P(v) + o\left(\frac{C_1}{\sqrt{n}}\right)\mathcal{N}(0,1)$$

Where C_1 is a function of $var(v)$. Note that $E_P(v) = -E_P\big(\log\big(m(x,\theta)\big)\big) = -E_P(\log(P)) + E_P(\log(P)) - E_P\big(\log\big(m(x,\theta)\big)\big) = -E_P\big(\log\big(P(x)\big)\big) + E_P \log\left(\frac{P(x)}{m(x,\theta)}\right) = H(P) + D(P\,||m(x,\theta)\big)$.

Therefore,

$$-\frac{1}{n}\sum_{i=1}^{n} \log\big(m(x_i,\theta)\big) = H(P) + D(\,P||m(x,\theta)\,) + O\left(\frac{C_1}{\sqrt{n}}\right)\mathcal{N}(0,1)$$

Q.E.D.

Proof of Proposition 1)

From our assumptions in the paper and the asymptotic efficiency of MLE [13], we know that $\lim_{n\to\infty} m(x,\theta_n) = P(x)$ where $\theta_n = \max_{\theta} \sum_{i=1}^{n} \log\big(m(x_i,\theta)\big)$

Hence, for $E|\log\big(m(x_i,\theta_n)\big)| < \infty$ and using the strong law of large number we have

$$\lim_{n\to\infty} -\frac{1}{n}\sum_{i=1}^{n} \log\big(m(x_i,\theta_n)\big) = H(P) + D(P||m(x,\theta_\infty)\big) = H(P) + D(P||P) = H(P)$$

Therefore, a model that has been trained on $D_{\infty,0}$ should reach the loss value $H(P_0)$. Assume $x^{(0)} \sim P_0(x)$ and $x^{(t)} \sim P_t(x)$. Consider a model that has been trained on a dataset from time t ($D_{\infty,t}$) and been tested on a dataset from time 0, $D_{\infty,0}$. In this case, $\lim_{n\to\infty} m\big(x^{(t)},\theta_n\big) = P_t(x)$ where $\theta_{n,t} = \max_{\theta} \sum_{i=1}^{n} \log\left(m\left(x_i^{(t)},\theta\right)\right)$

The test loss value for this model is

$\lim_{n\to\infty} -\frac{1}{n}\sum_{i=1}^{n} \log\left(m\left(x_i^{(0)},\theta_{\infty,t}\right)\right) = H\big(P(x)\big) + D\left(P(x)||\, m\big(x,\theta_{\infty,t}\big)\right) = H(P_0) + D(P_0||P_t)$

Since both $H(P_0)$ and $D(P_0||P_t)$ are non-negative functions of distributions [30,40], we conclude that the loss value is higher than $H(P_0)$. Therefore, a bounded size dataset should reach the loss value $H(P_0) + D(P_0||P_t)$.

Formalizing this argument, we define a neighborhood around $H(P_0)$ with the size $\delta > 0$ and prove that with probability $(1-\epsilon)$, any dataset of bounded size reaches a value in the neighborhood.

Mathematically, for large dataset samples $n \gg 1$ and $\delta > 0$, using theorem 1 we have

$$P\left(\left|-\frac{1}{n}\sum_{i=1}^{n}\log\left(m\left(x_i^{(0)},\theta_{n,0}\right)\right)-H(P_0)\right|>\delta\right)=P\left(\left|D(P_0||P_t)+O\left(\frac{1}{\sqrt{n}}\right)\mathcal{N}(0{,}1)\right|>\delta\right)$$

$$=P\left(\left|\mathcal{N}\left(D(P_0||P_t),o\left(\frac{1}{\sqrt{n}}\right)\right)\right|>\delta\right)=$$

$$=P\left(\mathcal{N}\left(D(P_0||P_t)-\delta,o\left(\frac{1}{\sqrt{n}}\right)\right)>0\right)+P\left(\mathcal{N}\left(D(P_0||P_t)+\delta,o\left(\frac{1}{\sqrt{n}}\right)\right)<0\right)$$

$$=\underbrace{\Phi\left(\frac{\delta-D(P_0||P_t)}{o\left(\frac{1}{\sqrt{n}}\right)}\right)}_{(i)}+\underbrace{\Phi\left(\frac{-\delta-D(P_0||P_t)}{o\left(\frac{1}{\sqrt{n}}\right)}\right)}_{(ii)}$$

Where $\Phi(.)$ is the cumulative distribution function of standard Normal. In above equation, since $\delta > 0$, (i) is bigger than (ii) which means

$$P\left(\left|-\frac{1}{n}\sum_{i=1}^{n}\log\left(m\left(x_i^{(0)},\theta_{n,0}\right)\right)-H(P_0)\right|>\delta\right)<2\Phi\left(\frac{\delta-D(P_0||P_t)}{o\left(\frac{1}{\sqrt{n}}\right)}\right)$$

Since for $\delta < D(P_0||P_t)$ the numerator is negative,

$$\lim_{n\to\infty}\Phi\left(\frac{\delta-D(P_0||P_t)}{o\left(\frac{1}{\sqrt{n}}\right)}\right)=\Phi(-\infty)=0$$

Therefore, For any $\epsilon,\delta > 0, \exists\, n_0 < \infty\ \ s.t. \forall\, n > n_0$

$$P\left(\left|-\frac{1}{n}\sum_{i=1}^{n}\log\left(m\left(x_i^{(0)},\theta_{n,0}\right)\right)-H(P_0)\right|>\delta\right)<2\Phi\left(\frac{\delta-D(P_0||P_t)}{o\left(\frac{1}{\sqrt{n}}\right)}\right)<\epsilon$$

Meaning that a dataset size of $n > n_0$ with probability $1-\epsilon$ surpass the performance of infinite dataset size from time t.

Q.E.D.

Proof of Theorem 2)

a, c) This is a direct result of theorem 1 and proposition1.

b) Due to monotonic decline of effectiveness over time, $D\left(P_0||P_{t_1}\right) < D(P_0||P_{t_2})$ for $t_2 > t_1$ and $D\left(P_0||P_{t_1}\right) > D(P_0||P_{t_2})$ for $t_2 < t_1$.

For sufficiently large number of datapoints, the model $m(x,\theta)$ almost converged to $P(x)$. Therefore, due to continuity and differentiability of the learning curve, we can use Taylor expansion of learning curve's inverse on the neighborhood of $P(x)$.

$$\begin{aligned}
\bar{n}_{D_{n,t}} &= EG_0^{-1}\left(H(P_0) + D\left(P_0||m\left(x,\theta_{n,t}\right)\right)\right) \\
&\sim G_0^{-1}\left(H(P_0) + D(P_0||P_t)\right) \\
&+ E\left[\left(D\left(P_0||m\left(x,\theta_{n,t}\right)\right) - D(P_0||P_t)\right)\frac{\partial G_0^{-1}(q)}{\partial q}|_{H(P_0)+D(P_0||P_t)}\right] \\
&= G_0^{-1}\left(H(P_0) + D(P_0||P_t)\right) - E\left[\left(E_{P_0}\log\frac{m\left(x,\theta_{n,t}\right)}{P_t(x)}\right)\right]\frac{\partial G_0^{-1}(q)}{\partial q}|_{H(P_0)+D(P_0||P_t)}
\end{aligned}$$

Using Taylor expansion $\log(1+x)\sim x - \frac{x^2}{2} + \frac{x^3}{3} + o(x^4)$, in the neighborhood of $x = 0$. We do this because we expect $m\left(x,\theta_{n,t}\right) \to P_t(x)$. Using Taylor expansion, we have

$$\begin{aligned}
n_{D_{n,t}} &= G_0^{-1}\left(H(P_0) + D(P_0||P_t)\right) \\
&- E_{P_0}\left(\frac{m\left(x,\theta_{n,t}\right) - P_t(x)}{P_t(x)} - \frac{1}{2}\left(\frac{m\left(x,\theta_{n,t}\right) - P_t(x)}{P_t(x)}\right)^2 + \frac{1}{3}\left(\frac{m\left(x,\theta_{n,t}\right) - P_t(x)}{P_t(x)}\right)^3\right. \\
&\left. + o\left(\frac{m\left(x,\theta_{n,t}\right) - P_t(x)}{P_t(x)}\right)^4\right)\frac{\partial G_0^{-1}(q)}{\partial q}|_{H(P_0)+D(P_0||P_t)}
\end{aligned}$$

Assuming $m(x,\theta)$ to be a continuous function of θ, we can use theorem 10.1.12 in [13] (Asymptotic efficiency of MLE) and approximate $m\left(x,\theta_{n,t}\right)$ with respect to randomization in algorithms and choice of dataset in the training phase. Therefore,

$$m\left(x,\theta_{n,t}\right) \sim P_t(x) + \frac{1}{\sqrt{n}}\mathcal{N}(0, v(\theta))$$

Where $v(\theta)$ is the Cramer-Rao lower bound.

$$\begin{aligned}
\bar{n}_{D_{n,t}} &= G_0^{-1}\left(H(P_0) + D(P_0||P_t)\right) \\
&- E[E_{P_0}\left(\frac{1}{\sqrt{n}}\mathcal{N}\left(0,\frac{v(\theta)}{P_t(x)}\right) - \frac{1}{2n}\left(\mathcal{N}\left(0,\frac{v(\theta)}{P_t(x)}\right)\right)^2 + \frac{1}{3n\sqrt{n}}\left(\mathcal{N}\left(0,\frac{v(\theta)}{P_t(x)}\right)\right)^3\right. \\
&\left. + o\left(\frac{1}{n^2}\right)\right)]\ \frac{\partial G_0^{-1}(q)}{\partial q}|_{H(P_0)+D(P_0||P_t)}
\end{aligned}$$

$$= G_0^{-1}\big(H(P_0) + D(P_0||P_t)\big) + \frac{1}{2n}\left[EE_{P_0}\left(\mathcal{N}\left(0, \frac{v(\theta)}{P_t(x)}\right)\right)^2 + o\left(\frac{1}{n^2}\right)\right] \frac{\partial G_0^{-1}(q)}{\partial q}|_{H(P_0)+D(P_0||P_t)}$$

Since the first and third moment of centered Gaussian distribution is equal to 0.

As a side note, the argument inside the brackets is positive. Since $\frac{\partial G_0^{-1}(q)}{\partial q} < 0$ *we conclude*

$$\frac{1}{2n}\left[EE_{P_0}\left(\mathcal{N}\left(0, \frac{v(\theta)}{P_t(x)}\right)\right)^2\right] \frac{\partial G_0^{-1}(q)}{\partial q}|_{H(P_0)+D(P_0||P_t)} < 0$$

Hence, $\bar{n}_{D_{n,t}}$ *is an increasing function in* n *for sufficiently large n.*

Back to the prove, we now take the derivative of $f_n(t_1, t_2)$ with respect to n. For large n we use the following approximation

$$\hat{f}_n(t_1, t_2) = \frac{G_0^{-1}\left(H(P_0) + D\big(P_0||P_{t_1}\big)\right) + \frac{1}{2n}\left[EE_{P_0}\left(\mathcal{N}\left(0, \frac{v(\theta)}{P_{t_1}(x)}\right)\right)^2\right] \frac{\partial G_0^{-1}(q)}{\partial q}|_{H(P_0)+D\left(P_0||P_{t_1}\right)}}{G_0^{-1}\left(H(P_0) + D\big(P_0||P_{t_2}\big)\right) + \frac{1}{2n}\left[EE_{P_0}\left(\mathcal{N}\left(0, \frac{v(\theta)}{P_{t_2}(x)}\right)\right)^2\right] \frac{\partial G_0^{-1}(q)}{\partial q}|_{H(P_0)+D\left(P_0||P_{t_2}\right)}}$$

$$= \frac{\bar{n}_{D_\infty, t_1} + \frac{1}{2n}\left[EE_{P_0}\left(\mathcal{N}\left(0, \frac{v(\theta)}{P_{t_1}(x)}\right)\right)^2\right] \frac{\partial G_0^{-1}(q)}{\partial q}|_{H(P_0)+D\left(P_0||P_{t_1}\right)}}{\bar{n}_{D_\infty, t_2} + \frac{1}{2n}\left[EE_{P_0}\left(\mathcal{N}\left(0, \frac{v(\theta)}{P_{t_2}(x)}\right)\right)^2\right] \frac{\partial G_0^{-1}(q)}{\partial q}|_{H(P_0)+D\left(P_0||P_{t_2}\right)}}$$

To show the derivative sign, we focus on the for large n (Omitting $o\left(\frac{1}{n^3}\right)$)

$$\Rightarrow num\left(\frac{\partial \hat{f}_n(t_1, t_2)}{\partial n}\right) \sim \frac{1}{2n^2}\Bigg[\bar{n}_{D_\infty, t_1}\left[EE_{P_0}\left(\mathcal{N}\left(0, \frac{v(\theta)}{P_{t_2}(x)}\right)\right)^2\right] \frac{\partial G_0^{-1}(q)}{\partial q}|_{H(P_0)+D\left(P_0||P_{t_2}\right)}$$

$$- \bar{n}_{D_\infty, t_2}\left[EE_{P_0}\left(\mathcal{N}\left(0, \frac{v(\theta)}{P_{t_1}(x)}\right)\right)^2\right] \frac{\partial G_0^{-1}(q)}{\partial q}|_{H(P_0)+D\left(P_0||P_{t_1}\right)}\Bigg]$$

Since the argument in the brackets are not a function of n, we can conclude that for large n, the substitution function $f_n(t_1, t_2)$ is monotonic in n.

Q.E.D.

Proof of Lemma 1)

Assume dataset $D_{n,t}$ is sampled over time with the density function $\lambda_{t=t_0} = \frac{1}{n}\sum_{i=1}^{n} 1(t_i = t_0)$ Considering each sample a random variable, number of times $1(x < a) = 1$ in the dataset is equal to $\sum_{i=1}^{n} 1_{t_i}(x < a)$. Therefore, the expected frequency of the event $\{x < a\}$ is equal to

$$P_{D_{n,t}}(x < a) = E\left(\sum_{i=1}^{n} \frac{1_{t_i}(x < a)}{n}\right) = \underbrace{\frac{1}{n}\sum_{i=1}^{n} E(1_{t_i}(x < a))}_{Fubini\ theorem} = \frac{1}{n}\sum_{i=1}^{n} P_{t_i}(x < a)$$

Integrating the density function λ_t into formulation

$$P_{n,[0,t],\lambda_t}(x < a) = \frac{1}{n}\sum_{i=1}^{n} P_{t_i}(x < a) = \frac{1}{n}\int_0^t \sum_{i=1}^{n} P_s(x < a)1(t_i = s)\ ds$$

$$= \int_0^t P_s(x < a)\frac{1}{n}\sum_{i=1}^{n} 1(t_i = s)\ ds = \int_0^t P_s(x < a)\lambda_s ds$$

Q.E.D.

Proof of Proposition 2)

Using lemma 1, we know that dataset's net distribution is

$$P_{[0,t],\lambda_t}(x < a) = \int_0^t P_s(x < a)\lambda_s ds$$

Therefore, training on the dataset of infinite size and test it at time 0, the error will be equal to

$$H(P_0) + D\big(P_0||P_{[0,t],\lambda_t}\big) = H(P_0) + D\left(P_0||\int_0^t P_s(x < a)\lambda_s ds\right)$$

Since KL-divergence is a convex function [30], we use Jensen inequality to derive an upper bound

$$D\left(P_0||\int_0^t P_s(x < a)\lambda_s ds\right) = D\left(\int_0^t P_0\lambda_s ds\ ||\int_0^t P_s(x < a)\lambda_s ds\right)$$

$$= \int_0^t \lambda_s D(P_0||P_s)ds < \max_{s\in[0,t]} D(P_0, P_s)$$

Besides, we know that KL-divergence is nonnegative which means

$$D(P_0||P_0) = 0 \le D\left(P_0||\int_0^t P_s(x < a)\lambda_s ds\right) \le \max_{s\in[0,t]} D(P_0, P_s)$$

Since we assumed in this paper that the function $h(t) = D(P_0||P_t)$ is continuous over time (The change in distribution is gradual and hence, $h(t)$ is continuous) There exist a time $t^* \in [0, t]$ such that

$$D(P_0||P_{t^*}) = D\left(P_0||\int_0^t P_s(x < a)\lambda_s ds\right)$$

Therefore

$$H(P_0) + D(P_0||P_{t^*}) = H(P_0) + D\left(P_0||\int_0^t P_s(x < a)\lambda_s ds\right)$$

This means that P_{t^*} generate the same loss value as $P_{[0,t],\lambda_t}$.

Q.E.D.

Appendix B

We ran four experiments with different dataset sizes over Reddit data. The up-left figure shows the effectiveness curve when we trained the model over 25MB of data. Up-right, down-left, and down-right show the curves for 50, 100, 200 MBs, respectively. As can be seen in these graphs, the effectiveness curve is becoming steeper as expected. Meaning that substitution gain will be monotonically increasing in the number of samples.

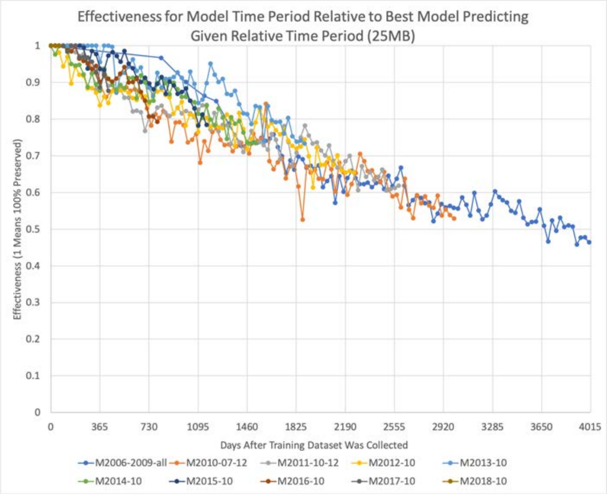
For example, looking at the effectiveness value for day 2920, we can see the effectiveness values of roughly 0.55, 0.5, 0.45, and 0.4 in the 25, 50, 100, and 200 MBs graphs, respectively.

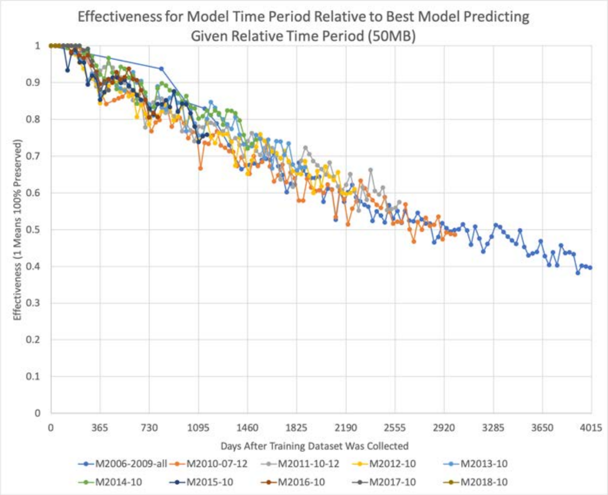
$f_{25MB}(0,2920) \sim 1.81$

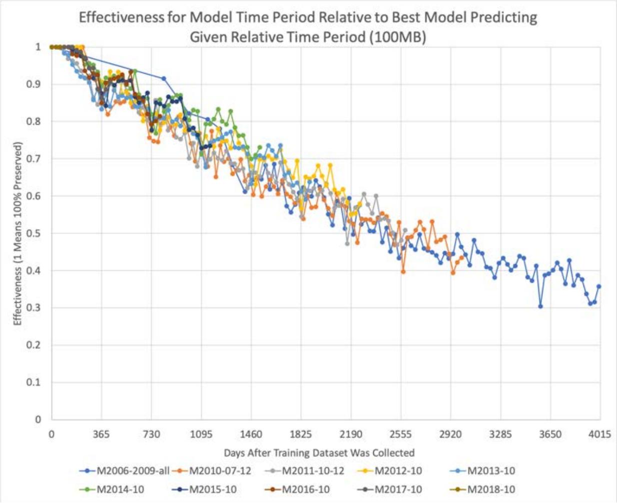
$f_{50MB}(0,2920) \sim 2.00$

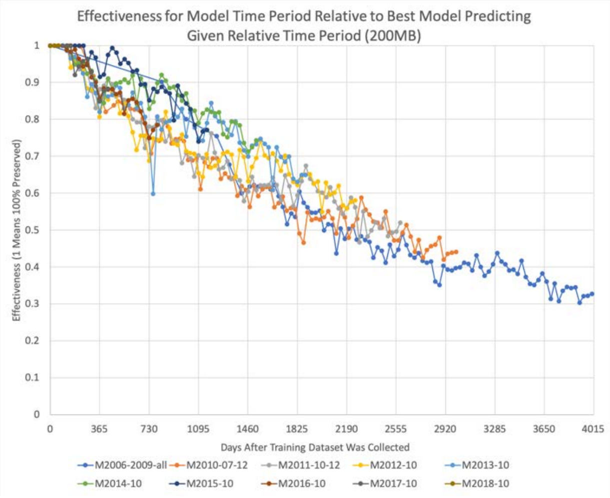
$f_{100MB}(0,2920) \sim 2.22$

$f_{200MB}(0,2920) \sim 2.50$


Effectiveness for Model Time Period Relative to Best Model Predicting Given Relative Time Period (25MB)


Effectiveness for Model Time Period Relative to Best Model Predicting Given Relative Time Period (50MB)


Effectiveness for Model Time Period Relative to Best Model Predicting Given Relative Time Period (100MB)


Effectiveness for Model Time Period Relative to Best Model Predicting Given Relative Time Period (200MB)