Big Data And PAC Learning In The Presence Of Noise: Implications For Financial Risk Management

V L Raju Chinthalapati, Sovan Mitra* and Antoaneta Serguieva

Abstract

High accuracy forecasts are essential to financial risk management, where machine learning algorithms are frequently employed. We derive a new theoretical bound on the sample complexity for PAC learning in the presence of noise, and does not require specification of the hypothesis set $|H|$. Consequently, we demonstrate that for realistic financial applications (where $|H|$ is typically infinite) that big data is necessary, contrary to prior theoretical conclusions. Secondly, we show that noise (which is a non-trivial component of big data) has a dominating impact on the data size required for PAC learning. Consequently, contrary to current big data trends, we argue that high quality data is more important than large volumes of data. Thirdly, we demonstrate that the level of algorithmic sophistication (specifically the Vapnik–Chervonenkis dimension) needs to be traded-off against data requirements to ensure optimal algorithmic performance. Finally, our new Theorem can be applied to a wider range of machine learning algorithms, as it does not impose finite $|H|$ requirements. This paper will be of interest to researchers and industry specialists who are interested in machine learning in financial applications.

keywords: Big Data ; Risk Management ; Noisy Data ; Machine Learning ; VC dimension ; Sample Complexity.

*University of Liverpool,
Department of Mathematics,
Liverpool,
UK.
sovan.mitra@liverpool.ac.uk

1 Introduction

In financial risk management, forecasting plays a fundamental role in the ability to effectively manage risk. Financial risk management is most crucially concerned with forecasting risks (which by definition are unknown and random events) because they can have substantial impacts upon a firm. Examples of financial risks include stock market crashes [11], the Global Financial Crisis (which has led to prolonged years of financial difficulties), and exchange rate risks that can result from political factors (such as a political election).

Machine learning algorithms are being increasingly used to improve forecasts in financial risk management [12, 15]. This is because machine leaning provides a method of modelling (and therefore predicting) data that may exhibit non-trivial properties which other modelling methods would not be able to sufficiently model [20]. The ability of machine learning algorithms to create hypotheses from data, rather than from a fixed set of instructions, offers high flexibility to computational modelling. A particularly advantageous aspect of machine learning is that it can engage in iterative learning, that is learning and modelling can be adapted as new data is introduced [4, 14]. In fact, this property alone has led to a wide range of important applications being developed [3, 6], such as sophisticated fraud detection, automated automobile driving, and applications for learning human behaviour (eg consumer purchasing decisions).

PAC (probably approximately correct) [17, 18] learning provides a mathematical framework for machine learning. PAC learning determines if a potential hypothesis (arising from a classifier or oracle) is deemed to have learnt the correct function that maps inputs to their associated outputs. Additionally, Valiant [16] proved that a minimum bound exists for the (training) data's length required to obtain a hypothesis within quantified bounds of accuracy. Consequently, PAC learning is important to financial risk management as poor learning can impact the accuracy of forecasts. This has become a particularly important issue since the Global Financial Crisis because models have been cited as a key cause for the crisis.

A fundamental issue in machine learning (and PAC learning) is the issue of sample complexity. This is concerned with the total number of training samples m required to achieve sufficient learning accuracy, under the PAC learning framework (and under its respective assumptions). The sample complexity or m is of fundamental importance because PAC learning theory implies that the probability and amount of accuracy possible for a learning function is limited by m. In other words, if we wish to obtain better learning then this requires more training data m.

In addition to training data impact the quality of learning, the training data size m in itself is important due to its impact on algorithmic implementation and analysis. Firstly, a large training data m may be practically infeasible if insufficient data availability. This can occur in financial applications, particularly if a new financial product is created and has limited data available. Hence m tells us the feasibility of implementing some algorithms. Secondly, a large m value could lead to large computational complexity, that is requiring powerful computer hardware and other computational resources to enable enough data can be processed in a feasible timescale. Such issues are particularly important in many real world applications, where computational resources and timescales are limited.

Whilst the standard PAC framework offers a useful analytical concept for ma-

chine learning, current theory makes a number of restrictive assumptions that negate its usefulness to financial risk management applications. Firstly, some theorems typically require the hypothesis set $|H|$ to be finite in order to obtain informative bounds on m. However, most machine learning algorithms typically have $|H| = \infty$. This would be particularly the case in financial risk management where a wide and sophisticated range of machine learning algorithms are employed to forecast future events. Consequently bounds on sample complexity for financial risk management applications are not realistic.

Secondly, a significant amount of literature on learning theory assumes that little or no noise exists. In other words we assume that the input data that a classifier receives data is not corrupted by any noise [7]. In financial applications this would be an unrealistic assumption because many financial variables are frequently modelled with noisy components. In fact stock market prices are typically modelled with Brownian motion to incorporate noise in their prices. A natural consequence of the presence of noise is that it can impact the learning ability of any algorithm, since it will be necessarily harder to learn any relationship between input and output data. A simple analogy would be fitting a line of best fit is more challenging with noisy data than compared to noiseless data.

Given that we are likely to encounter noise in financial applications, it is important to understand the impact of noise upon sample complexity for the reasons previously outlined (such as feasibility of algorithm and impact on computational resources). Moreover, to what extent are the learning algorithm's learning ability affected by noise. Additionally, as financial applications typically employ machine learning algorithms with infinite hypotheses sets, we would like to understand m for such algorithms.

In this paper we investigate PAC learning in the presence of noise. Specifically we investigate PAC learning when a noisy oracle or classifier exists, which assigns an incorrect output to an associated input, based on some noise level. Although the issue of PAC learning in the presence of noise has been addressed in a number of papers (due to the relevance of noise in real world applications), our paper makes a new contribution. In particular, Angluin and Laird's seminal paper [1] introduces new results in PAC learning in the presence of (classification) noise. However, Angluin and Laird necesssarily require finite $|H|$ for their Theorem to be informative, hence the Theorem is not particularly applicable financial risk management applications.

In this paper, we make a number of contributions. Firstly, we derive a new bound on the sample complexity for PAC learning, specifically the minimum sample length m for a given level of learning accuracy, in the presence of noise classification. Moreover, we extend and generalise further upon the results of Angluin and Laird's classic Theorem [1] by not requiring $|H|$ to determine m in our bound on the sample size. Consequently, our bound on the required sample size is more applicable to financial applications.

Secondly, using our bound we show that, contrary to Angluin and Laird's Theorem [1] which assumes finite $|H|$, machine learning algorithms require very large values in m. We show that even for very low noise data that we require data of the order of big data sizes, in order to produce sufficiently accurate forecasts that would be applicable in financial risk maangement applications. Thirdly, we show that the noise term significantly impacts the amount of data required for forecasting and that the size of big data required can increase substantially with noise. Consequently,

we argue that data cleaning techniques, or conversely high quality (or low noise) data, is more important greater volumes of big data. This conclusion is contrary to the current 'direction of travel' in big data research trends, which emphasise greater volume of data rather than higher quality (or data cleaning techniques).

The paper is organised as follows: in the next section we introduce the preliminaries and notation of the paper, providing the background review and related literature to our work. In the next section we provide main results and contributions, showing our new derivations in the presence of classification noise and PAC learning. We then analyse and demonstrate the implications of our Theorem for financial risk management applications. Finally we end with a conclusion and give suggestions for further areas of work.

2 Preliminaries

In this section we introduce the introductory theory and notation of the paper, provide a background review and related literature to our work.

2.1 Introduction To Big Data and Machine Learning

Big data has recently received significant attention in academic literature as well as in the general media. This is because it has posited that big data will lead to a paradigm shift in data analysis and forecasting. The term 'big data' currently has no consensus definition but typically refers to data sizes that traditional data processing software normally cannot manage. Therefore big data is typically of the order of sizes of at least 1TB or higher [10]. Consequently, big data involves new challenges in research, and these are frequently related to storage and analysis issues.

The emergence of big data has occurred due to the proliferation of data in the modern world. Firstly, large amounts of data can be collected at relatively little or no cost, in volumes that were previously not economically feasible. Hence the amount of data that is available has grown. Secondly, there has been a significant development in technological products in recent years that enable high volumes of data to be captured. For example, internet and internet related software, mobile devices, digital and other internet related products means that products exist that can capture large amounts of data.

Machine learning is concerned with learning algorithms designed to learn some mapping function between pairs of data [9]. Typically the pair represent some input data and its associated output data, and the aim of the algorithm is to determine the function relating the inputs and outputs for all possible values. To achieve this, the algorithm is supplied with some training data of length m. We also assume the data is supplied with the correct output (also called classification or labels) for each input data point.

Let there exist some sample data which consists of a pair (x_i, y_i), where x_i is some input, i is an index, $x_i \in X$, X is called the instance space. We also have $y_i \in Y$ where y_i is the associated output, label or classification of x_i in (x_i, y_i), and Y is called the output set. Typically the classification is Boolean, that is $y_i \in \{0, 1\}$ $\forall i$, although it is also possible for the classification to be specified to take values in $\mathbb{R}$ (that is $y_i \in \mathbb{R}$ $\forall i$). For the remainder of this paper we assume the output set Y is always Boolean, unless stated otherwise.

The true relation between input and output data points can be expressed by the target function (or target concept), denoted $t(.)$. Therefore we have the relation

$$y_i = t(x_i), \forall i.$$

The target function is some unknown function that we aim to discover, or "learn" by employing our learning algorithm. We also have $t(.) \in C$, where C is a set of possible target functions and C is known as the concept class. Now let there exist a learning algorithm L, which produces a function or hypothesis $h(.)$, and $h(.) \in H$ where H is the hypothesis set (set of all hypotheses that can be computed by the learning algorithm L). The ultimate aim of the learning algorithm L is to produce a hypothesis $h(.)$ that is as close to $t(.)$ as possible.

Let $Z = X \times Y$, $z_i = (x_i, y_i)$ and $z_i \in Z$, then the learning algorithm L receives a sequence of training data z of length m:

$$z = (z_1, z_2, \ldots, z_m) = ((x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)),$$

where $z \in Z^m$. The sample $(x_1, x_2, \ldots, x_m)$ is drawn from X^m, and we define X to have an associated probability $P(.)$ and X^m to have a probability $P^m(.)$ (see [2] for more detail). After observing a sufficiently high number of training datapoints, the learning algorithm L must output a hypothesis h estimating the target hypothesis t. Therefore the learning algorithm can be considered as a function that maps the set of all training samples Z^m, for all m, onto the hypothesis set H:

$$L : \cup_{m=1}^{\infty} Z^m \to H.$$

An error in our hypothesis $h(.)$ is defined as a misclassification, that is

$$h(x_i) \neq t(x_i), x_i \in X.$$

We can consider an error in our hypothesis $h(.)$ as a measure of the performance of hypothesis h. In other words errors tell us how accurate $h(.)$ will be in correctly determining the outputs. We define $er_P(h)$ as the error function for $h(.)$, under the probability measure P, using the equation

$$er_P(h) = P\{h(x_i) \neq t(x_i)\}, x_i \in X.$$

Also, the sample error of hypothesis h, denoted by $er_z(h)$, is defined as follows:

$$er_z(h) = \frac{1}{m} \sum_{i=1}^{i=m} \mathbf{1}_{\{h(x_i) \neq y_i\}},$$

where $\mathbf{1}_{\{.\}}$ is the indicator function. The sample error is also a measure of performance of the hypothesis $h(.)$ in terms of error. The function $er_z(h)$ is simple to determine and can be used as an approximate estimate of $er_P(h)$, which is akin to the the 'true' error. We note that $er_z(h)$ is an error measure over the (training) data z of length m. Consequently, $er_z(h)$ as a measure of performance or error is dependent upon m, and so does not measure the full error in the sense that it does not measure the error over the entire data.

2.2 PAC Learning And Noise

In order to determine whether a potential hypothesis has learnt a function well enough to a sufficient standard, we require some criterion. Valiant in his seminal paper introduced the concept of PAC (probably approximate correct) learning [16], and defines a good hypothesis as one that has a specifically low classification error, for a specified level of probability. A key contribution that was obtained from PAC learning theory was the relation between machine learning and the computational complexity, essentially the intensiveness of the computational resources required to implement a learning algorithm. Valiant proved that a good hypothesis can exist, provided that the training data of length m is sufficiently large. Hence m and sample complexity become critical aspects in PAC learning. PAC (probably approximate correct) learning is a standard criterion for supervised learning and has been a major area of research in the past 30 years.

In PAC learning, we define a good hypothesis if there exists some chosen (and small) constant $\epsilon > 0$, a constant $0 < \delta < 1$ relating to probability, and that

$$P(er_P(h) \leq \epsilon) > 1 - \delta.$$

Essentially a good hypothesis must have a low error or $er_P(h)$. Moreover, Valiant proved that the minimum training sample length m required to obtain a good hypothesis is given by

$$m \geq \frac{1}{\epsilon} log\left(\frac{|H|}{\delta}\right).$$

We notice that the minimum sample length value is a function of ϵ and δ, hence the level and probability of accuracy required in our hypothesis is directly related to the training data used.

Our learning algorithm L can produce hypotheses from the hypothesis set H. We would therefore be interested in some measure of the capability of the set of functions in H, where we intuitively define capability as meaning the complexity, flexibility and general richness of functions. The VC (Vapnik -Chervonenkis) dimension provides such a method of measuring capability; we denote this as $VC(H)$ for H. The VC dimension is also important because a larger VC dimension number implies it is harder to learn all the correct functions possible, and so more data samples will be needed to determine the correct function.

So far we have assumed that we receive perfect labelling from the target function $t(.)$, that is the training data (provided to L) is uncorrupted. However, in many real world applications the training data is corrupted, that is we have $y_i \neq t(x_i)$ for some i values. This is known as classification noise. To make this principle clear, we say we have either a normal oracle EX or a corrupted oracle EX_η. The normal oracle means that the training data is not corrupted, that is $y_i = t(x_i)$, $\forall i$. However for a corrupted oracle $y_i \neq t(x_i)$ for some i values. The noise parameter η, where $0 < \eta < \frac{1}{2}$, determines the level of corruption. Specifically, we have the binomial probability: $P(y_i = t(x_i)) = 1 - \eta$, and $P(y_i \neq t(x_i)) = \eta$.

The noise η can be used to model any generic noise in the data or our system. Given that training data has $y_i \neq t(x_i)$ for some i, we define the *disagreement number* as the number of labelled instances (x_i, y_i) in the training sample which are such that $y_i \neq t(x_i)$. By definition a corrupted Oracle must have some outputs where

$y_i \neq t(x_i)$, for some i, hence the disagreement number will never be 0 for corrupted oracles.

In order for classification methods to be effective, it is essential that data is correctly labelled. In fact Zhu and Wu [21], as well as many other authors, have demonstrated that noise can adversely affects the performance of classifiers. However in financial risk management applications, data or measurement systems may be corrupted by noise, hence the learning algorithms receive corrupted labels. Moreover, it is typically impractical, too time consuming or uneconomical to obtain training data without noise. Hence it is therefore of great practical importance to develop learning algorithms in the presence of noise, and this has been of significant interest to the machine learning community.

A number of approaches have been proposed to deal with machine learning in the presence of noise. One method is based on designing learning algorithms that are naturally robust to noise, so that the noise itself cannot affect the learning. In other words, the learning algorithm would not be sensitive to noise. In fact there exist many practical algorithms that are resistant to noise and so do not affect performance, however such algorithms cannot always be used for financial risk management applications. Moreover, if we have the situation where $Pr(y_i \neq t(x_i)) = \eta$, then the algorithm cannot simply be made insensitive to the noise.

A second approach to dealing with noise is that one can provide better quality training data (that is less noisy data) to improve learning. This is effectively achieved by applying a filter to the data to remove or attenuate noise. Essentially, noisy data is either eliminated from the training data, or assigned a different (and more correct) value. For financial risk management applications, filters are cheap and simple to implement, however a major disadvantage is that filters typically remove too much data prior to training. Consequently, the reduction in data (in terms of quantity) can then impair the learning algorithm's performance.

A more flexible approach is to understand learning algorithms and assume that some data will be noisy (due to corruptly labelled data). In such a situation it is important to understand the sample complexity and related issues for reasons previously outlined (such as the impact on computational resources, data requirements and feasible computation times). The seminal paper in PAC learning under classification noise is the following result of Angluin and Laird [1] that relates the noise parameter η to the disagreement number, where the output labels are Boolean.

Theorem 1 (Angluin and Laird) *Let η_b be a known upper bound on η, where $\eta \leq \eta_b$, and $\eta_b < \frac{1}{2}$. If we draw a sample of size m from $EX_\eta(t, P)$, where $t \in H$, m is given by*

$$m \geq \frac{2}{\epsilon^2(1-2\eta_b)^2} \ln\left(\frac{2|H|}{\delta}\right),$$

and find any hypothesis $h \in H$ with a minimal disagreement number, then

$$P^m(er_P(h) > \epsilon) \leq \delta.$$

Essentially, Angluin and Laird's (hearafter, we use AL instead of Angluin and Laird) theorem answers the fundamental question (under its given assumptions) about the sample complexity (that is size m), for a given ϵ and δ, in the presence of a corrupted oracle $EX_\eta(t, P)$. It is also worth pointing out that more than one h can exist with a minimal disagreement number.

3 Main Results

The Angluin and Laird Theorem is an important equation for understanding learning under noise. However, we notice that it requires $|H|$ and so the Angluin and Laird Theorem requires finite $|H|$ for the bound to be informative. For financial applications this is a significant disadvantage since we can have $|H| = \infty$. In our paper, we develop an alternative derivation that provides a bound on m and does not require $|H|$. We now state our main Theorem and new contribution:

Theorem 2 *Let d be the VC dimension of the hypothesis set H. Let us also assume that the output set Y is Boolean valued, that is $Y \in \{0, 1\}$. For an oracle $EX_\eta(t, P)$, where $\eta < \frac{1}{2}$, if we draw a sample of size m, where*

$$m \geq \frac{64}{(1-2\eta)^2\epsilon^2}\left[d \ln\left(\frac{128}{(1-2\eta)^2\epsilon^2}\right) + \ln\left(\frac{8}{\delta}\right)\right],$$

and we find any hypothesis $h \in H$ with minimal disagreement number, then

$$P^m(er_P(h) > \epsilon) \leq \delta. \qquad \blacksquare$$

Theorem 2 provides a useful result, which gives us the minimum sample length m, for a given level of error, hence we have quantified the sample complexity. Moreover our Theorem involves the VC dimension d, rather than $|H|$ and so is therefore more widely applicable to financial risk management applications. In the proceeding sections we will now discuss our Theorem by elaborating on the derivation at each step.

3.1 Error Function Probability Bounds In The Presence of Noise

In this section we derive the PAC bound in the presence of noise. This will also help with deriving our main Theorem in the subsequent section. We begin by firstly deriving a relation between $er_Q(h)$ in terms of $er_P(h)$, that is the error term for the noisy (or corrupt) oracle in terms of the noiseless oracle.

Lemma 1 *Let Q in $er_Q(h)$ be the probability measure equivalent to the probabilities obtained when the inputs in X are corrupted by noise. From the construction of the probability measure Q, it is clear that $\forall\ h \in H$ that*

$$er_Q(h) = \eta + (1-2\eta)er_P(h).$$

Proof:
Under a noiseless oracle EX, we have $\forall x_i \in X$, $er_P(h) = P\{h(x_i) \neq t(x_i)\}$. Under a noisy oracle we receive noisy data, so that $t(x_i) :\mapsto t'(x_i)$. Similarly, we can write the error function for EX_η as

$$er_Q(h) = Q\{h(x_i) \neq t(x_i)\}.$$

Alternatively this can be written as $er_Q(h) = P\{h(x_i) \neq t'(x_i)\}$. We can rewrite this equation:

$$er_Q(h) = P\{h(x_i) \neq t(x_i)\}(1-\eta) + P\{h(x_i) = t(x_i)\}\eta$$

The equation can be explained as follows. Now $er_Q(h)$ can have errors (that is $h(x_i) \neq t'(x_i)$) due to 2 sources. Firstly, $t(x_i) = t'(x_i)$ when the corrupt oracle EX_η does not alter the output compared to EX, however the hypothesis h itself is wrong. Hence $h(x_i) \neq t(x_i)$ and this occurs with probability $1-\eta$. Secondly, when the corrupt oracle EX_η alters the output compared to EX, this is when $t'(x_i) \neq t(x_i)$ (or alternatively when $h(x_i) = t(x_i)$) and this occurs with probability η. (We note in passing that by observing the first and third definition of $er_Q(h)$ it can be seen that

$$Q(.) = P(.)(1-\eta) + (1-P(.))\eta).$$

We can now re-express the previous equation using

$$P\{h(x_i) \neq t(x_i)\} = er_P(h) \implies P\{h(x_i) = t(x_i)\} = 1 - er_P(h),$$

so that,

$$er_Q(h) \;=\; (1-\eta)er_P(h) + \eta(1-er_P(h)).$$

We then rearrange the equation so that we obtain the final solution

$$\begin{aligned} er_Q(h) &= er_P(h) - \eta er_P(h) + \eta - \eta er_P(h), \\ &= \eta + (1-2\eta)er_P(h). \end{aligned}$$ ■

Let us now assume that $er_P(h) \geq \epsilon$, where ϵ is some arbitrarily chosen small constant. Using our previous Lemma, we can now write

$$\begin{aligned} er_Q(h) &= \eta + (1-2\eta)er_P(h), \\ er_Q(h) &\geq \eta + (1-2\eta)\epsilon, \\ &\geq \eta + s, \end{aligned}$$

where we denote $s = (1-2\eta)\epsilon$ for convenience. Hence we can also write

$$P^m\left(er_P(h) \geq \epsilon, er_z(h) < \eta + \frac{s}{2}\right) \;=\; P^m\left(er_Q(h) \geq \eta + s, er_z(h) < \eta + \frac{s}{2}\right).$$

where $P(a,b)$ denotes the joint probability of events a and b, under P. Now given that we have $Q(.) = \eta + (1-2\eta)P(.)$, then $Q(.)$ is linear in η, given that $0 < \eta < 0.5$. Therefore, with respect to η, $Q(.)$ is a minimum at $\eta = 0$ and $Q(.) = P(.)$, and $Q(.)$ is a maximum at $\eta = 0.5$ where $Q(.) = 0.5$, $\forall$ $P(.)$. In PAC learning, we assume $P(.) \leq \upsilon$, where $0 < \upsilon < 0.5$, to ensure we have a good learning algorithm. Therefore $Q(.) \geq P(.)$. We can therefore write

$$P^m\left(er_P(h) \geq \epsilon, er_z(h) < \eta + \frac{s}{2}\right) \;\leq\; Q^m\left(er_Q(h) \geq \eta + s, er_z(h) < \eta + \frac{s}{2}\right).$$

Using a result in Angluin and Laird [1], we can express an upper bound on $P^m(er_P(h) > \epsilon)$

$$P^m(er_P(h) > \epsilon) \leq P^m\left(er_z(t') \geq \eta + \frac{s}{2}\right) + P^m\left(er_P(h) \geq \epsilon, er_z(h) < \eta + \frac{s}{2}\right).$$

The upper bound on $P^m(er_P(h) > \epsilon)$ can be explained as follows. The probability $P^m(er_P(h) > \epsilon)$ must be bounded above by (i) firstly the probability of the sample error of t' for $er_z(t') \geq \eta + \frac{s}{2}$, plus also (ii) the probability that the hypothesis h has error function $er_P(h) \geq \epsilon$, when the sample error of h is $er_z(h) \leq \eta + \frac{s}{2}$. With this upper bound, the right hand term can now be expressed in terms of Q^m using our previous expression; substituting this into the equation now gives us

$$P^m(er_P(h) > \epsilon) \leq P^m\left(er_z(t') \geq \eta + \frac{s}{2}\right) + Q^m\left(er_Q(h) \geq \eta + s, er_z(h) < \eta + \frac{s}{2}\right).$$

We can re-express the second term on the right hand side. As we have the condition $er_z(h) < \eta + \frac{s}{2}$, or alternatively,

$$\eta > er_z(h) - \frac{s}{2},$$

therefore

$$\begin{aligned} er_Q(h) &\geq \eta + s, \\ er_Q(h) &\geq er_z(h) - \frac{s}{2} + s, \\ &\geq er_z(h) + \frac{s}{2}. \end{aligned}$$

Therefore we now have

$$\begin{aligned} P^m(er_P(h) > \epsilon) &\leq P^m\left(er_z(t') \geq \eta + \frac{s}{2}\right) + Q^m\left(er_Q(h) \geq er_z(h) + \frac{s}{2}\right), \\ &\leq P^m\left(er_z(t') \geq \eta + \frac{s}{2}\right) + Q^m\left(|er_Q(h) - er_z(h)| \geq \frac{s}{2}\right). \end{aligned}$$

We now have a bound on the probability for $P^m(er_P(h) > \epsilon)$. Therefore the probability that the error function will exceed ϵ for a hypothesis h, is bounded above by the two probabilities on the right hand side. This is a useful inequality because it gives us a means to quantify the accuracy or error of our hypothesis h and with a degree of statistical confidence. Such calculations are particularly important in financial forecasting applications, where one must understand statistical calculations with a degree of confidence.

The equation is also a useful relationship because it demonstrates that the probability of the error function exceeding ϵ is not only dependent on the sample error $er_z(h)$ (for reasons mentioned before) but also on the level of noise itself. As we can see that η and η related terms (such as s) are present in the equation, the impact of noise will affect the probability of the error function increasing or decreasing. This is a reassuring result as we would expect noise to affect the algorithm's accuracy but also demonstrates the importance of noise in algorithmic performance.

3.2 Application Of The Vapnik-Chervonenkis Inequality

Although we derived a bound on the probability of the error function, and this provides a useful quantity in terms of the accuracy of our learning algorithm, it does not tell us about the sample complexity or the training data length (m) required for PAC learning. The sample complexity is fundamental to machine learning (and therefore financial forecasting applications involving machine learning) and has significant implications on various aspects of algorithms.

The Vapnik and Chervonenkis Inequality [19, 17] is an important Theorem in machine learning (also see *Theorem 4.3* from [2] for more information). The VC (Vapnik and Chervonenkis) Inequality is frequently applied and utilised in machine learning theory, and as it relates the last term in our equation to m, it can provide useful information on m using our equation. The VC inequality as given in [2] is:

Lemma 2 (Vapnik and Chervonenkis Inequality) *Suppose that H is a set of {0, 1} - valued functions defined on a set X and that P is a probability on $Z = X \times \{0, 1\}$. For $0 < \epsilon < 1$, m a positive integer, with VC dimension d, then we have for every $h \in H$*

$$P^m\{|er_P(h) - er_z(h)| \geq \epsilon\} \leq 4\left(\frac{2\kappa m}{d}\right)^d e^{\frac{-\epsilon^2 m}{8}},$$

where $\kappa = e$ (that is the exponential constant (we use a different letter for this rather than e for clarity of derivation)), and d denotes the VC dimension of the hypothesis set H (as defined earlier).

The VC inequality is important because it shows that provided our training sample is large enough, then with a sufficiently high enough probability we can conclude that for any $h \in H$, the sample error of h and the "true" error of h are extremely close. Additionally, the inequality is bounded by a negative exponential in m, implying that the boundary will rapidly approach 0 as m increases (assuming the bracketed expression grows less quickly). Hence training data m is important to reducing error in any learning algorithm. Such a result is important in ensuring that one is able to obtain good estimations in forecasting applications such as in financial risk management.

The VC Inequality is a particularly relevant Theorem to apply for financial forecasting applications because the VC inequality is independent of any probability distribution. Hence the inequality is widely applicable to a wide range of forecasting applications in finance, where a diverse range of distributions exist. Hence we would like our learning algorithms to be distribution independent, otherwise this would impose a significant constraint on forecasting applications.

We now apply the famous VC inequality to our derivation. Since $s = (1-2\eta)\epsilon \Rightarrow \epsilon > s \Rightarrow e^{-\epsilon} < e^{-s}$. Therefore, using the VC inequality Lemma 2, then

$$\begin{aligned} P^m\left(|er_P(h) - er_z(h)| \geq \epsilon\right) &\leq 4\left(\frac{2\kappa m}{d}\right)^d e^{\frac{-\epsilon^2 m}{8}} \Rightarrow \\ Q^m\left(|er_Q(h) - er_z(h)| \geq \frac{s}{2}\right) &\leq 4\left(\frac{2\kappa m}{d}\right)^d e^{-\frac{(s/2)^2 m}{8}}. \end{aligned}$$

If we now rewrite this inequality then our result becomes that $\exists\, h \in H$

$$Q^m\left(|er_Q(h) - er_z(h)| \geq \frac{s}{2}\right) \;\leq\; 4\left(\frac{2\kappa m}{d}\right)^d e^{-\frac{s^2 m}{32}}.$$

Now, let us assume that δ is bounded below by

$$4\left(\frac{2\kappa m}{d}\right)^d e^{-\frac{s^2 m}{32}} \leq \frac{\delta}{2}. \tag{1}$$

If we rearrange this then we have

$$\frac{s^2}{4} \geq \frac{8}{m} \ln\left(8\frac{\left(\frac{2\kappa m}{d}\right)^d}{\delta}\right),$$

or alternatively rearranging we have

$$m \geq \frac{32}{s^2}\left(d\ln m + d\ln\left(\frac{2\kappa}{d}\right) + \ln\left(\frac{8}{\delta}\right)\right). \tag{2}$$

Using the VC inequality we have therefore now derived a bound on m. A bound on m is typically more useful than our (previous) probability bound as m has significant implications on computation and forecasting.

The application of the VC inequality provides some interesting insights. Firstly, our equation shows that the bound on m is dependent on the VC dimension d, implying that the VC dimension is important to forecasting regardless of any distributions. Secondly, the inequality contains s and therefore we can see that as noise increases the training data length required (m) will also increase. Hence algorithms obtain PAC learning in the presence of noise only if the training data size m increases.

3.3 Sample Complexity Bound In The Presence Of Noise

Our equation (2) provides an inequality on m, however it does not provide a particularly tractable or useful boundary on m. Firstly, the equation (2) contains m on both sides of the inequality, hence we cannot easily understand how m behaves in the equation. Secondly, whilst the inequality could be rearranged so that the m terms are all grouped on one side of the inequality, we would also have to include other terms on the side of the equation. In other words we cannot separate out terms, and have m's boundary expressed in terms of other variables. Therefore we cannot easily understand the boundary on m, particularly if we wish to understand the impact on financial forecasting applications in risk management.

To obtain a more useful and tractable bound on m, particularly for the purposes of financial risk management applications and prove our main Theorem, we can improve further upon our sample length m bound. To achieve this we first prove our proceeding Lemma.

Lemma 3 (Logarithmic Bound On m) *The following logarithmic bound on m exists:*

$$\ln m \leq \frac{s^2}{64d}m - \ln\left(\frac{s^2}{64d}\right) - 1.$$

Proof: First let us consider the equation

$$f(y) = e^y - y.$$

If we differentiate this we have

$$f'(y) = e^y - 1 \implies f'(0) = 0.$$

We therefore have a minimum value at $y = 0$, with value $f(0) = 1$. Therefore

$$f(y) \geq 1,$$

and therefore by substitution,

$$e^y - y \geq 1,$$

and so by rearrangement we have $\forall y \in \mathbb{R}$

$$1 + y \leq e^y.$$

Let us now make the substitution $y = \alpha x - 1$, $\forall x$ where $x > 0$, α is a constant where $\alpha > 0$. Therefore

$$\begin{aligned} 1 + (\alpha x - 1) &\leq e^{\alpha x - 1}, \\ \alpha x &\leq e^{\alpha x - 1}. \end{aligned}$$

We now take logarithms on both sides of the expression and rearrange:

$$\begin{aligned} ln(\alpha x) &\leq \alpha x - 1, \\ ln(\alpha) + ln(x) &\leq \alpha x - 1, \\ lnx &\leq \alpha x - ln\alpha - 1. \end{aligned}$$

Let us now make the substitution $x = m$, and $\alpha = \dfrac{s^2}{64d}$. Therefore substituting into our previous equation we now have

$$\ln m \leq \frac{s^2}{64d} m - \ln\left(\frac{s^2}{64d}\right) - 1.$$

■

The logarithmic inequality in m can be applied to equation (2), so that we have

$$\begin{aligned} \frac{32}{s^2}\left(d\ln m + d\ln\left(\frac{2\kappa}{d}\right) + \ln\left(\frac{8}{\delta}\right)\right) \leq\ & \frac{m}{2} + \frac{32d}{s^2}\ln\left(\frac{64d}{s^2}\right) - \frac{32d}{s^2} \\ & + \frac{32d}{s^2}\ln\left(\frac{2\kappa}{d}\right) + \frac{32}{s^2}\ln\left(\frac{8}{\delta}\right). \end{aligned}$$

If we now simplify further and rearrange the right hand side, we have

$$\frac{32}{s^2}\left(d\ln m + d\ln\left(\frac{2\kappa}{d}\right) + \ln\left(\frac{8}{\delta}\right)\right) \leq \frac{m}{2} + \frac{32d}{s^2}\ln\left(\frac{128}{s^2}\right) + \frac{32}{s^2}\ln\left(\frac{8}{\delta}\right). \quad (3)$$

Our previous inequality on m (equation (2)) now becomes using equation (3):

$$\begin{aligned} m &\geq \frac{m}{2} + \frac{32d}{s^2} \ln\left(\frac{128}{s^2}\right) + \frac{32}{s^2} \ln\left(\frac{8}{\delta}\right), \\ \frac{m}{2} &\geq \frac{32}{s^2}\left(d \ln\left(\frac{128}{s^2}\right) + ln\left(\frac{8}{\delta}\right)\right), \\ m &\geq \frac{64}{s^2}\left(d \ln\left(\frac{128}{s^2}\right) + \ln\left(\frac{8}{\delta}\right)\right). \end{aligned}$$

Therefore we finally have

$$m \geq m_0,$$

where (recalling $s = (1-2\eta)\epsilon$), we have

$$m_0 = \frac{64}{(1-2\eta)^2\epsilon^2}\left(d \ln\left(\frac{128}{(1-2\eta)^2\epsilon^2}\right) + \ln\left(\frac{8}{\delta}\right)\right). \qquad \blacksquare$$

This proves our main Theorem. This is a useful result as it tells us the minimum sample length (m), in other words the sample complexity, that is required to achieve learning within the framework of PAC learning. This is because it is well known that the sample length significantly impacts the practicability of any computation. In particular a large m may mean large computation times, significant computational resources (that may render a method unworkable), and the data requirements. In some cases the data required (due to m) may be too large.

4 Implications For Financial Risk Management

In this section we analyse the implications of our Theorem in terms of financial risk management applications. In order to examine PAC learning for algorithms used in financial risk management applications, we must first understand the ϵ and δ specifications required. As ϵ denotes the error in our algorithm, ideally we want ϵ to be as small as possible. Similarly, since δ denotes the percentage of bad hypotheses, we want δ to be as small as possible. Consequently ϵ and δ represent a source of model risk in our algorithms. In the financial sector risk is measured in terms of VaR (Value at Risk) and this is typically set at the 99^{th} percentile. Although there is no fundamental theory for choosing the 99^{th} percentile, we apply this percentile choice to our model for ϵ and δ. Hence we set $\epsilon < 0.01$ and $\delta < 0.01$.

For δ, we notice both in AL and our Theorem that δ will have relatively little impact on m, for the purposes of our study. In AL we have the term

$$ln\left(\frac{2|H|}{\delta}\right),$$

which can be written as

$$ln(2|H| - \delta).$$

Given that in PAC learning we have the condition that $0 < \delta < 1$, and for financial risk management applications the type of algorithms we will typically employ will have $|H| >> 1$, therefore $|H| >> \delta$ and the impact of δ can be ignored. Similarly, in our Theorem in equation (2), we have the term

$$ln\left(\frac{8}{\delta}\right),$$

which can be written as

$$ln(8 - \delta).$$

Given that in PAC learning we have the condition that $0 < \delta < 1$, the previous term can vary between 1.94 and 2.08 and so has relatively little impact on m. Although we have reasoned (for our studies) that δ will have a relatively insignificant impact, for the purposes of investigation we must assign a value to it.

4.1 PAC Learning: Big Data Implications

A key advantage of Theorem 2 is that it does not require $|H|$. Angluin and Laird's Theorem requires $|H|$, hence their Theorem is only informative for finite $|H|$. However for machine learning algorithms we typically have $|H| = \infty$ (see for example [13]. Consequently, the requirement for finite $|H|$ is a highly restrictive assumption. Moreover, in financial applications where there exist high complex and non-trivial patterns and data sources, we would typically expect non-trivial algorithms to be employed [6, 4]. Hence an assumption of finite $|H|$ in financial risk maangement applications would be unrealistic.

On the other hand, Theorem 2 does not require $|H|$, hence we can determine m or data bounds for a wider range of algorithms in finance. Whilst Theorem 2 requires d (and d is related is to $|H|$), if $|H| = \infty$, this does not imply $d = \infty$. Moreover for a wider class of algorithms it is more likely that $|H| = \infty$ than $d = \infty$, hence our Theorem is more applicable to a wider range of algorithms than AL's Theorem. A consequence of relaxing the finite $|H|$ assumption (using our Theorem) is that we can calculate the data or m required for training.

Using our Theorem, we now calculate m (or the number of datapoints required) for PAC learning. For the benefit of clarity we set $\eta = 0.00001$ and d=2 and do not vary them. We will investigate d and η later on, and note that varying either parameters would not significantly affect our results in this table in any case. We calculate m for different values of ϵ and δ, noting that their typical values for financial risk management applications would be set to $\epsilon = 0.01$ and $\delta = 0.01$. We also note that our method of choice of ϵ and δ is similar to the approach taken by AL in their original paper.

Table 1 provides m according to our Theorem (m_T) for different ϵ and δ values. As we can see from the results from our Theorem, a key consequence is that m is in the order of millions. We observe that we only require m to be in the order of $100,000$ if ϵ is of the order of 0.1 or above. However, as mentioned before, in financial applications we require high accuracy and we would expect $\epsilon \leq 0.01$, therefore it is unlikely we would require data of the order of $100,000$ datapoints.

Table 1: Sample Size m_T by Theorem 2 for different δ and ϵ Values

ϵ	δ	m_T
0.01	0.01	22,278,928
0.01	0.1	20,805,215
0.01	0.3	20,102,075
0.01	0.5	19,775,133
0.01	0.7	19,559,783
0.01	0.9	19,398,935
0.02	0.01	5,126,100
0.04	0.01	1,170,617
0.05	0.01	726,344
0.08	0.01	264,927
0.1	0.01	163,841

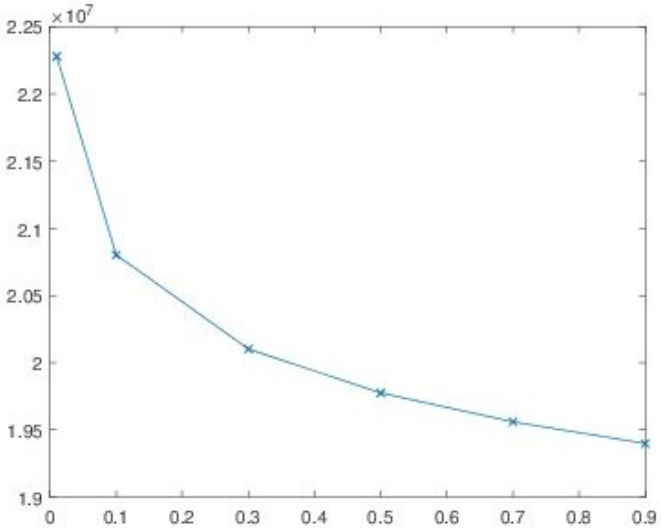

Figure 1: Graph of m_T $(y-axis)$ Against δ $(x-axis)$ for $\epsilon = 0.01$

The fact that m is of the order of millions has important implications. For financial data, given that one stock price quote will typically consist of its price to 4-6 significant figures, we would therefore require approximately 1TB of data for our learning algorithms [5, 8]. As mentioned before, there is currently no strict definition of big data but 1TB has been a proposed definition. Consequently, our Theorem implies that learning algorithms in the presence of noise require big data, for the purposes of financial risk management applications.

The fact that we require big data is a revealing conclusion since this would not be necessarily revealed using AL Theorem. If we calculate m for different $|H|$ values we obtain the following results. We set $\epsilon = 0.01$, $\delta = 0.01$, and $\eta = 0.1$ to give more realistic values.

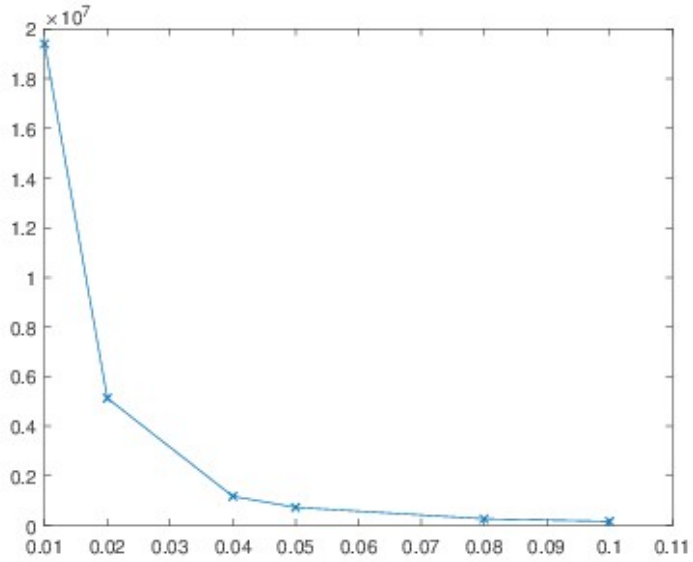

Figure 2: Graph of m_T $(y - axis)$ Against ϵ $(x - axis)$ for $\delta = 0.01$

Table 2 provides m according to Angluin and Laird's Theorem (m_A) for different ϵ and δ values. As one can see from the results, m is in the range of $10,000$ - $100,000$ and so for financial applications AL does not imply big data is required. We expect $|H|$ to be large for financial applications, due to the sophisticated algorithms required to analyse complex data in finance. However, we can see that even as $|H|$ increases exponentially, its impact on datapoints required (m) does not increase significantly and hardly reaches above 1 million. We can understand this from the AL equation whereby there is a logarithmic dependence on $|H|$ in $ln(2|H|/\delta)$. Hence an increase in $|H|$ has little impact on m as we take its logarithmic value.

If we calculated the equivalent values for m using our Theorem (noting that our Theorem does not depend on $|H|$)), for all the different values above we have $m = 35,701,927$. In this calculation we set $d = 2$ so that our m value gives a lower limit value. This is because we would expect $d > 2$ in real world applications and this would increase m (this can be understood by examining the equation). Hence we can see that while AL does not imply big data, our equation does conclude big data is required.

A reason for the difference in data requirements between AL and our Theorem is that AL assume finite $|H|$, consequently this will limit the data required for PAC learning. On the other hand, our equation does not impose this assumption and so a larger set of datapoints is required for PAC learning.

The implication that we require big data for financial risk management applications is significant. Firstly, given that risk management needs to be done frequently (for example it is common to calculate VaR at least on a daily basis and on a short time basis) implies that we not only require good learning algorithms but also fast algorithms. Big data can require significant time to process and if we do not have

Table 2: Sample Size m_A by Angluin and Laird's Theorem

$\lvert H\rvert$	m_A
5	215,867
10	237,528
100	309,484
1000	381,440
10^4	453,396
10^5	525,351
10^6	597,307
10^7	669,263
10^8	741,219
10^{10}	885,130
10^{12}	1,029,042

fast learning algorithms then the algorithms may take too long for financial risk management applications.

Secondly, fast computation time for financial risk management applications dealing with big data may imply that we would require higher end hardware to cope with learning algorithms for financial applications. For instance we may require computers with faster processors, we would also need to consider data storage issues and data curation (organisation and collation) since 1TB is required just for learning purposes in 1 round. If we had used AL equation then we would not be concerned with the computational hardware requirements as they are quite modest, essentially possible to run on a standard laptop.

Thirdly, from a financial data point of view, the fact that m is of the order of millions means that financial forecasting may not be possible with learning algorithms. This is because some financial assets and derivatives have limited history, for instance they may have been trading for only a few years. On a daily basis, this equates to approximately 750 datapoints, and so this would be far too low for learning purposes.

4.2 Impact Of Big Data Quality (Noise)

In many real world applications noise, or corruption of data, can occur for a number of reasons. In [21], they discuss that noise is unavoidable in affecting data. For example, noise can occur from noise existing in any measurement or recording system, distorting the original data. Sometimes data can be transformed (eg discretising data to convert it into a binary form), which can lead to noise in the data in various ways [22].

To understand the impact of noise on data requirements, we calculate m for different values of η, where we recall that $0 < \eta < 0.5$. As before we set $\epsilon = 0.01$, $\delta = 0.01$, and we set $d = 2$ to give a conservative value for m using our Theorem. For comparison, we also calculate m using AL's Theorem and set $|H| = 10^{12}$ to give an optimistic calculation of m.

Table 3 gives m values under AL (m_A) and our Theorem (m_T). In the final

Table 3: Sample Size m_A and m_T By Noise

η	m_T	m_A	Percentage increase (%)
10^{-5}	22,278,928	658,613	3283
10^{-4}	22,287,412	658,850	3283
0.01	23,250,422	685,742	3291
0.1	35,701,927	1,029,042	3369
0.15	47,328,722	1,344,055	3421
0.2	65,515,832	1,829,408	3481
0.25	96,209,771	2,634,347	3552
0.3	153,898,064	4,116,167	3639
0.35	281,779,514	7,317,631	3751
0.4	659,953,674	16,464,669	3908
0.45	2,817,260,376	65,858,677	4178
0.49	80,731,912,034	1,646,466,924	4803

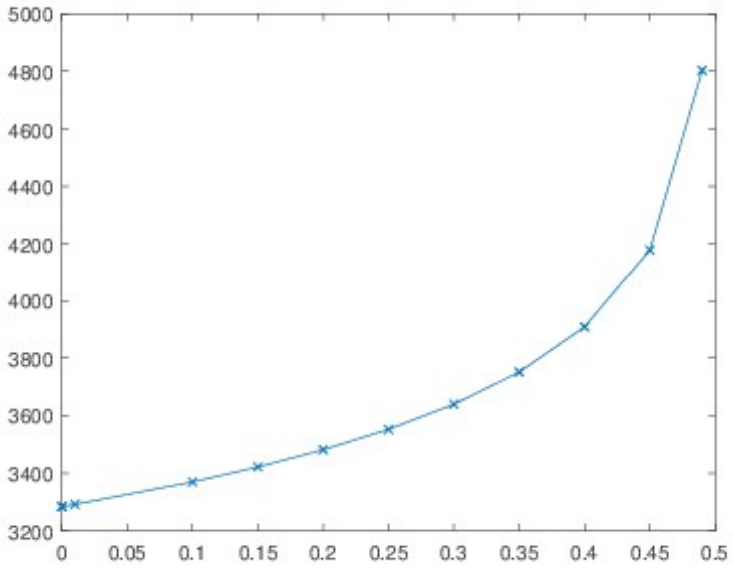

Figure 3: Graph of Percentage Difference Between m_T and m_A $(y-axis)$ Against Noise $(x-axis)$

column we calculate the percentage increase between m_T and m_A. As we can see from our results, the impact of noise is significant. As we approach $\eta = 0.4 - 0.45$ the data required reaches the the billion datapoint range, hence the demands for big data processing become even more important in for noisy data. However, we should note that it in reality, it is unlikely that financial data will approach a level where 50% of data would be noise.

A key insight from our Theorem is that the impact of noise is far more significant than what we may have assumed under AL. Under AL, whilst we have a requirement for data in the order of millions as η approaches 0.1, it still gives a bound for m

in the millions range until $\eta = 0.49$ (and such a figure is unrealistic with financial data). We also note that this is an optimistic estimate from AL as we have set $|H|$ at a high value. However AL assume finite $|H|$, which is unrealistic for financial applications.

Our Theorem on the other hand does not assume finite $|H|$ and we have provided a conservative estimate (given that d has been set to 2). The impact of noise increases the data requirements from multi-millions to billions. Additionally, we see that the incremental increase in m is far higher than what would see under AL's equation. Given that noise is practically always present in real world applications, our Theorem therefore provides an important insight into impact of noise.

We can understand the difference on noise dependence by examining the equations. In our Theorem we observe that m has a logarithmic dependence on the reciprocal of $1 - 2\eta$, consequently m can increase significantly with small changes in noise. In AL's Theorem on the other hand, m depends only on the reciprocal of $1 - 2\eta$. Consequently, the big data required in our Theorem will be more significant as η changes, compared to AL's Theorem.

The fact that noise has an impact on m is significant in itself because we are dealing with big data volumes. This is because noise is typically far more complex in big data than in conventional data sizes. This occurs by the very nature of the fact that a larger data set is more likely to contain more complex noise processes than in shorter data sets, since larger datasets give more opportunity for more complex noise processes to occur. Additionally, the quality of data captured in big data can vary far more than in traditional data, with more gaps in data and potential distortions in the data.

Consequently, the amount and type of noise that exists in big data tends to be more complex, hence eliminating noise from big data is typically harder to achieve than in smaller datasets. Moreover, it is well known that noise is financial data is highly non-trivial, in fact many commentators have stated it is the noisiness of financial data which prevents investors from profiting from stock patterns.

Another revealing insight from our Theorem is that noise (or equivalently data quality) has a far more significant impact on computational productivity than we may have assumed. In other words, for financial applications, if we have high quality data (with low noise $\eta = 0.1$) rather than low quality data (with high noise $\eta = 0.4$) then this can lead to a reduction of data by a factor of approximately 300. Consequently (with all other factors equal) this would lead a processing speed of 300 times faster, which is a very significant productivity gain.

This is an unexpected conclusion as the current 'trend' in big data analysis is that more volume is generally better, however our analysis shows that it is data quality that can be significantly more effective in providing better learning outcomes. Consequently, we suggest that big data applications should focus more on data cleaning applications, to produce high quality data, rather than focussing on higher volume data (which may be very noisy). The additional advantage of focussing on data cleaning methods is that such methods are normally cheap and easy to implement, unlike other big data techniques e.g. some require expensive high performance computers.

Table 4: VC Dimension d and Sample Size m_T

d	m_T
1	21,193,269
2	35,701,927
5	79,227,900
7	108,245,216
10	151,771,189
12	180,788,505
15	224,314,478
20	296,857,766
25	369,401,055
30	441,944,344

4.3 Impact of Algorithmic Capability

A new insight from our Theorem is the dependence of training data upon the VC dimension d. The VC dimension d captures the capabilities of the algorithms used for learning (in terms of general sophistication and flexibility). Given that financial data is highly non-trivial and that incorrect modelling can lead to significant losses to firms, we typically require highly sophisticated algorithms to ensure better financial risk management.

In AL's Theorem, the VC dimension is not included in the equation. Hence AL's Theorem does not tell us how d impacts data requirements. This in itself could be more harmful to financial risk management than using more simplistic algorithmics, since a sophisticated but badly trained (or calibrated) model can perform worse than a simple but well calibrated model. In fact such issues have been cited as a major cause of poor risk management. Moreover, it is well known in the financial sector that sometimes many parsimonious models are preferred than more sophisticated models due to the data requirements imposed by them.

To investigate the impact of d on the data requirements, we calculate m using our Theorem. For the benefit of clarity we set $\eta = 0.1, \epsilon = 0.01, \delta = 0.01$. We vary d from 1 to 30 to see the impact of VC dimension upon data requirements. We chose the upper limit at 30 similar to the range used in the analysis of VC dimension of neural networks in [14]. Given that neural networks are used in financial applications and represent the more sophisticated algorithms that can be employed in finance, the upper limit of 30 is a suitable value for our study.

For the benefit of comparison we also calculate m using AL's Theorem for the equivalent parameter values. To provide optimistic calculations under AL we set $|H| = 10^{12}$. For all the calculations above, AL data requirements would be 1,029,042 (for all values of d). Therefore,even for large $|H|$ AL does not provide an adequate estimate of data requirements.

An important insight from our Theorem is that the data required is significantly dependent on d. In AL, as AL assumes a finite $|H|$ (and a logarithmic relationship with $|H|$) the impact of d upon $|H|$ is not fully reflected in the equation, even though $|H|$ and d are (non-trivially) related. However, with our results and Theorem we can see that the data required grows proportionately with d, and d is important to (big) data requirements.

In fact, from our Theorem we can see that d directly impacts m, in that the bound on m increases linearly with d. Hence d provides a direct "proxy" on the data requirements. This is important to understand in financial risk management applications because we typically require complex learning algorithms as financial data has non-trivial relationships. However, if the more complex algorithms lead to larger data requirements (through increasing d) then such algorithms may not be preferable. Hence our Theorem tells that the linear relation between m and d implies there is a direct trade-off between algorithmic complexity and sample complexity.

5 Conclusion

In this paper we investigate PAC learning in the presence of noise. We derive a new Theorem on the theoretical bound on the training data required for PAC learning, in the presence of noise. We therefore extend the classic Theorem of Angluin and Laird by including algorithms that do not require finite $|H|$. Hence our Theorem is more widely applicable.

This paper makes a number of contributions. Firstly, contrary to prior theoretical analyses, we show that big data is necessary for training algorithms used for realistic financial applications (where $|H| = \infty$). Secondly, we demonstrate that noise has a substantial impact on the data size required for PAC learning. Consequently, contrary to current big data trends, we demonstrate that high quality data is more important than large volumes of data. Thirdly, we show that the level of algorithmic sophistication (specifically the Vapnik–Chervonenkis dimension) is not necessarily advantageous to learning algorithms, as it can impose high training data requirements. Hence a trade-off is required between the Vapnik–Chervonenkis dimension and the data required for training.

In terms of future areas of work, we would like to develop our model for specific computational applications eg. fraud detection, marketing applications, transport applications etc.. Another area we would like to investigate is data cleaning methods to reduce noise, to improve learning performance for PAC learning algorithms. Finally, we would like to develop our results further for the purposes of financial risk management theory, such as tail risk theory. Our paper provides a more realistic learning model, taking into account non-finite $|H|$ and noise. Therefore our paper will be of interest to commercial industry, where PAC based machine learning and noisy data have important applications.

References

[1] Angluin, D., Laird, P.: Learning from noisy examples. Machine Learning **2**, 343–370 (1988)

[2] Anthony, M., Bartlett, P.L.: Neural Network Learning: Theoretical Foundations. Cambridge University Press (1999)

[3] Bajari, P., Nekipelov, D., Ryan, S.P., Yang, M.: Machine learning methods for demand estimation. American Economic Review **105**(5), 481–485 (2015)

[4] Belloni, A., Chernozhukov, V., Wei, Y.: Post-selection inference for generalized linear models with many controls. ArXiv e-prints **1304.3969** (2013)

[5] Bholat, D.: Big data and central banks. Bank of England Quarterly Bulletin **55**(1), 86–93 (2015)

[6] Blanco, A., Pino-Mejias, R., Lara, J., Rayo, S.: Credit scoring models for the microfinance industry using neural networks: Evidence from peru. Expert Systems with Applications **40**(1), 356–364 (2013)

[7] Bshouty, N.H., Eiron, N., Kushilevitz, E.: Pac learning with nasty noise. Theoretical Computer Science **288**(2), 255–275 (2002)

[8] Daas, P., Bart, B., Van Den, H.P., Puts, M.: Big data as a source for official statistics. Journal of Official Statistics **31**(2), 249–262 (2015)

[9] Finlay, S.: Predictive Analytics, Data Mining and Big Data: Myths, Misconceptions and Methods. Palgrave Macmillan (2014)

[10] Gandomi, A., Haider, M.: Beyond the hype: Big data concepts, methods, and analytics. International Journal of Information Management **35**(2), 137–144 (2015)

[11] Kirilenko, A., Kyle, A., Tuzun, T., Samadi, M.: The flash crash: High frequency trading in an electronic market. Journal of Finance **72**, 967–998 (2017)

[12] Lacher, R.C., Coats, P.K., Sharma, S.C., Fant, L.F.: A neural network for classifying the financial health of a firm. European Journal of Operational Research **85**(1), 53–65 (1995)

[13] Mehryar, M., Rostamizadeh, A., Talwalkar, A.: Foundations of machine learning. MIT press (2012)

[14] Mertens, S., Engel, A.: Vapnik-chervonenkis dimension of neural networks with binary weights. Phys. Rev. E **55**, 4478 (1997)

[15] Mullainathan, S., Spiess, J.: Machine learning: An applied econometric approach. Journal of Economic Perspectives **31**(2), 87–106 (2017)

[16] Valiant, L.G.: A theory of the learnable. Communications of the ACM **27**(11), 1134–1142 (1984)

[17] Vapnik, V.: Statistical learning theory. John Willey & Sons (1998)

[18] Vapnik, V., Chervonenkis, A.: Ordered risk minimization. Automation and Remote Control **34**, 1226–1235 (1974)

[19] Vapnik, V.N., Chervonenkis, A.Y.: On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability and its Applications **16(2)**, 264–280 (1971)

[20] Varian, H.: Big data: New tricks for econometrics. Journal of Economic Perspectives **28**(2), 3–28 (2014)

[21] Zhu, X., Wu, X.: Class noise vs. attribute noise: A quantitative study. Artificial Intelligence Review **22**(3), 177–210 (2004)

[22] Zimek, A., Schubert, E., Kriegel, H.P.: A survey on unsupervised outlier detection in high-dimensional numerical data. Statistical Analysis and Data Mining **5**(5), 363–387 (2012)