Big Data and Probably Approximately Correct Learning In The Presence of Noise: Implications For Financial Risk Management

VL Raju Chinthalapati[1], Sovan Mitra[2], and Antoaneta Serguieva[4]

[1]*Business School, University of Greenwich, London, UK*
[2]*Department of Mathematics, University of Liverpool, Liverpool, UK*
[4]*nChain and LSE Systemic Risk, London, UK*

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 Probably Approximately Correct (PAC) learning in the presence of noise, and does not require specification of the hypothesis set $|H|$. We demonstrate that for realistic financial applications where $|H|$ is typically infinite. This is contrary to prior theoretical conclusions. We further 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. This paper additionally demonstrates that the level of algorithmic sophistication, specifically the Vapnik–Chervonenkis (VC) 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 contributes to theoretical and applied research in the domain of machine learning for financial applications.

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

Mathematics Subject Classification: 68Q32, 60J20, 68T05.

1 Introduction

Forecasting plays a fundamental role for financial risk management and is most crucially concerned with random and unknown events that may have substantial impacts upon a firm or a financial system. Examples include stock market crashes [16], the recent Global Financial Crisis with its prolonged effect, and exchange rate risks resulting from political elections, etc [13]. Machine learning algorithms are increasingly used to improve forecasts in financial risk management [17, 20, 9, 28]. Machine leaning provides a modelling and predicting methodology for data exhibiting non-trivial properties that other modelling approaches are not be able to cope with [27]. 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, where learning and modelling are adapted to newly introduced data [4, 19]. This property alone has led to the development of a wide range of important applications [3, 6], such as sophisticated fraud detection and learning human behaviour in investing or purchasing decisions.

PAC learning [24, 25] 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. Valiant [23] also proved that a minimum bound exists for the (training) data required to obtain a hypothesis within quantified bounds of accuracy. PAC learning is important to financial risk management as poor learning impacts the accuracy of forecasts. Incapable modeling has been cited as one of the key causes for the Global Financial Crisis. The issue of sample complexity is fundamental in machine and PAC learning. It is still not completely known how many examples (size of the training data) are needed for learning successfully in PAC learning [22]. This is concerned with the total number of training samples m required to achieve sufficient learning accuracy, under the PAC learning framework and its respective assumptions. The fundamental importance of sample complexity or m is due to PAC learning theory implying that the probability and amount of accuracy possible for a learning function is limited by m. If we wish to obtain better learning, then this requires more training data m.

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

The majority of the work in machine learning is empirical research where the performance of the algorithms are evaluated by their performance on the sample data sets. Even though this is a useful simple approach for evaluating the individual algorithm's performance, it is difficult to compare different algorithms rigorously. The standard PAC framework offers a useful analytical concept for machine learning.

Earlier theorems in the literature consider learning bounds for finite hypothesis set with noise free training data sets. However the theory associated currently with this framework makes a number of restrictive assumptions that negate 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. Most machine learning algorithms, however, typically have $|H| = \infty$, particularly in the case in financial risk management, where a wide and sophisticated range of machine learning algorithms are employed to forecast future events. Therefore, in the case of such applications, bounds on sample complexity are not realistic. Secondly, a significant volume of literature on learning theory assumes that little or no noise exists [15, 7]. It is assumed that the input data to a classifier is not corrupted by any noise [8]. This is not a realistic assumption in financial applications, as variables are frequently modelled with noisy components. Stock market prices are typically modelled with a Brownian motion to incorporating noise.

The presence of noise impacts the learning ability of any algorithm, since it is necessarily harder to learn any relationship between input and output data. A simple analogy is identifying a line of best fit, which is more challenging with noisy data compared to noiseless data. For reasons such as feasibility of algorithms and impact on computational resources, it is important to understand the impact of noise upon sample complexity. A key question concerns the extent to which algorithms' learning ability is affected by noise, particularly for machine learning algorithms with infinite hypotheses sets. We would like to understand m for such algorithms.

This paper investigates PAC learning in the presence of noise. We focus on PAC learning when a noisy oracle or classifier exists and assigns, based on some noise level, an incorrect output to an associated input. 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. In particular, Angluin and Laird's seminal paper [1] introduces new results in the presence of (classification) noise, however requires a finite $|H|$ for their Theorem to be informative. Hence, their Theorem is not applicable in financial risk management. Our paper makes the following contributions to the body of research on PAC learning. Firstly, we derive a new bound on the sample complexity, specifically the minimum sample length m, for a given level of learning accuracy in the presence of noise classification. We further extend and generalize the results of Angluin and Laird's classic Theorem, because we do not require $|H|$ in order to determine m as sample-size bound. Secondly, using our bound we show that, contrary to the classic Theorem [1] that assumes finite $|H|$, machine learning algorithms require very large values for m. Our results show that even for very low levels of noise data, very large data sizes are required in order to produce sufficiently accurate forecasts. Thirdly, this paper shows that the noise term significantly impacts the amount of data required for forecasting, and that the required big data increases substantially with noise. We argue that data cleaning techniques, or conversely high-quality or low-noise data, can be more important than greater volumes of big data. This conclusion does not align entirely with current trends in big data research, which emphasize greater volume of data rather than higher-quality data (and cleaning techniques).

The paper is organized as follows. The next section provides the background of the problem in focus, and introduces preliminaries and notations for the paper. In Section 3, we provide the main results and our contributions to PAC learning in the presence of classification noise. The implications of our Theorem for financial risk management applications are demonstrated in Section 4. Finally, Section 5 provides conclusions and directions for further research.

2 Preliminaries

2.1 Introduction to Big Data and Machine Learning

Big data is currently receiving significant attention, due to the proliferation of data in the modern world and due to the technological advances in capturing large volumes of data. It is posited that big data will lead to a paradigm shift in data analysis and forecasting. There is no consensus definition for the term ’big data’, but it typically refers to sizes beyond the capabilities of traditional data-processing software, at least 1TB or higher [12]. Big data brings new challenges to storage, analysis, and research.

Machine learning concerns the design of algorithms for learning mapping functions among data domains [11]. Typically, this involves some input data and its associated output data, and the aim of a learning algorithm is identifying the function that relates inputs to outputs for all possible values. The algorithm is supplied with some training data of length m and it is typically assumed that the data is supplied with the correct output, also called classification or label, for each input data point. Let there exists some sample data consisting of a pair (x_i, y_i), where x_i is an input, i is an index, $x_i \in X$, and X is the instance space. Also, y_i is the associated output, label, or classification of x_i in (x_i, y_i), where $y_i \in Y$ and Y is the output set. Typically, the classification is Boolean, $y_i \in \{0, 1\}$ $\forall i$, though it is possible to specify the classification as taking values in $\mathbb{R}$, $y_i \in \mathbb{R}$ $\forall i$. Since in this work we concentrate on learning Boolean functions, in the remainder of this paper, it is assume that the output set Y is Boolean unless stated otherwise.

The true relation between input and output data points is expressed with the target function or target concept, denoted with $t(.)$:

$$y_i = t(x_i), \forall i. \tag{1}$$

The target function is unknown and the aim is to discover or "learn" this function, by employing a learning algorithm. Here, $t(.) \in C$, and C is a set of possible target functions, where C is the concept class. Let there exists a learning algorithm L that produces a function or hypothesis $h(.)$, where $h(.) \in H$ and H is the hypothesis set: the set of all hypotheses that can be computed by the algorithm L). The ultimate aim of 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 $z_i \in Z$, L receives a sequence of training data z of length m:

$$z = (z_1, z_2, \ldots, z_m) = ((x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)), \tag{2}$$

where $z \in Z^m$. The sample $(x_1, x_2, \ldots, x_m)$ is drawn from X^m, and X is an associated probability $P(.)$, and so X^m is defined with 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 algorithm can be considered as a function mapping the set of all training samples Z^m, for all m, onto the hypothesis set H:

$$L : \cup_{m=1}^{\infty} Z^m \rightarrow H. \tag{3}$$

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

$$h(x_i) \neq t(x_i), x_i \in X. \tag{4}$$

We can consider an error in our hypothesis $h(.)$ as a measure of the performance of h. 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:

$$er_P(h) = P\{h(x_i) \neq t(x_i)\}, x_i \in X. \tag{5}$$

The sample error of hypothesis h, denoted with $er_z(h)$, is defined as follows:

$$er_z(h) = \frac{1}{m} \sum_{i=1}^{i=m} \mathbf{1}_{\{h(x_i) \neq y_i\}}, \tag{6}$$

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 'true' error. 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 to a sufficient standard, we require some criterion. Valiant introduced in his seminal paper the concept of PAC learning [23], and defined a good hypothesis as having a specifically low classification error, for a specified level of probability. A key contribution of the PAC learning theory was the relation between machine learning and 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 learning is a standard criterion for supervised learning, and has been a major area of research in the past 30 years.

In PAC learning, a good hypothesis is defined as follows: there exists some chosen (small) constant $\epsilon > 0$, and a constant $0 < \delta < 1$ relating to probability, so that

$$P(er_P(h) \leq \epsilon) > 1 - \delta. \tag{7}$$

Valiant also proved that the minimum training sample length m required to obtain a good hypothesis is given with

$$m \geq \frac{1}{\epsilon} log \left(\frac{|H|}{\delta} \right). \tag{8}$$

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. The learning algorithm L can produce hypotheses from the hypothesis set H, and it is of interest to have some measure of the capability of the set of functions in H. The capability is intuitively defined as meaning the complexity, flexibility and general richness of functions. The Vapnik-Chervonenkis (VC) dimension provides such a method for measuring capability and is denoted here with $VC(H)$. The VC dimension is also important because a larger VC dimension implies that it is harder to learn all the correct functions possible. So more data samples will be needed to identify the correct function.

So far we have assumed that a perfect labelling is received from the target function $t(.)$, and so the training data provided to L is uncorrupted. In many real world applications, however, the training data is corrupted and $y_i \neq t(x_i)$ for some i values. This is known as classification noise. To make this principle clear, we say that there is either a normal oracle EX or a corrupted oracle EX_η. The normal oracle means that the training data is not corrupted: $y_i = t(x_i), \forall i$. 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. In the case of 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, the *disagreement number* is defined 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 a disagreement number larger than 0.

In order for classification methods to be effective, it is essential that data is correctly labelled. Zhu and Wu [29], as well as many other authors, have demonstrated that noise can adversely affects the performance of classifiers. In financial risk management applications, data or measurement systems may be corrupted by noise and so the learning algorithms receive corrupted labels. It is also typically impractical, too time consuming or uneconomical to obtain training data without noise. Hence, it is 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 such approaches have been proposed [15, 30]. Some of the methods are based on designing learning algorithms that are naturally robust to noise, so that the noise itself cannot affect the learning [7]. There exist practical algorithms that are resistant to noise, however such algorithms are not always applicable to financial forecasting and risk management. Furthermore, in cases where $Pr(y_i \neq t(x_i)) = \eta$, the algorithm cannot simply be made insensitive to the noise. A second type of approaches to deal with noise and improve learning is to provide better-quality training data. This is effectively achieved by applying a filter to the data to remove or attenuate noise. 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 [21, 14]. Thus, the reduction in data-size can 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 on PAC learning under classification noise [1] provides the following result that relates the noise parameter η to the disagreement number, when 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*

$$m \geq \frac{2}{\epsilon^2(1-2\eta_b)^2} \ln\left(\frac{2|H|}{\delta}\right), \tag{9}$$

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

$$P^m(er_P(h) > \epsilon) \leq \delta. \tag{10}$$

Angluin and Laird's (AL) Theorem answers the fundamental question, under its given assumptions about the sample complexity (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 AL Theorem is important equation understanding learning under noise. However, it requires finite $|H|$ for the bound to be informative. For financial applications, this is a significant disadvantage, since there typically $|H| = \infty$. We alternatively derive here a Theorem that provides a bound on m and does not require $|H|$. Our main contribution is stated next, in terms of Theorem 2:

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*

$$m \geq \frac{64}{(1-2\eta)^2\epsilon^2}\left[d \ln\left(\frac{128}{(1-2\eta)^2\epsilon^2}\right) + \ln\left(\frac{8}{\delta}\right)\right], \tag{11}$$

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

$$P^m(er_P(h) > \epsilon) \leq \delta. \tag{12}$$

Theorem 2 provides a useful result, which gives the minimum sample length m, for a given level of error, hence we have quantified the sample complexity. Our Theorem further involves the VC dimension d, rather than $|H|$, and so is more widely applicable to financial risk applications. The following Sub-sections will discuss and elaborate on each step in deriving our Theorem.

3.1 Probability Bounds for the Error Function in the Presence of Noise

In this Sub-section, we derive in detail the PAC bound in the presence of noise, which also helps with deriving the main new Theorem in the subsequent Sub-section. We begin with deriving a relation between $er_Q(h)$ and $er_P(h)$, 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 $\forall\ h \in H$,*

$$er_Q(h) = \eta + (1-2\eta)er_P(h). \tag{13}$$

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, the error function for EX_η can be written as

$$er_Q(h) = Q\{h(x_i) \neq t(x_i)\}. \tag{14}$$

Alternatively, this can be written as

$$er_Q(h) = P\{h(x_i) \neq t'(x_i)\}, \tag{15}$$

and w can rewrite this equation as:

$$er_Q(h) = P\{h(x_i) \neq t(x_i)\}(1-\eta) + P\{h(x_i) = t(x_i)\}\eta \tag{16}$$

The last equation can be explained as follows. Here, $er_Q(h)$ can have errors, $h(x_i) \neq t'(x_i)$, 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. A minor note here is that observing the first and third definition of $er_Q(h)$, it is apparent that

$$Q(.) = P(.)(1-\eta) + (1-P(.))\eta). \tag{17}$$

We can now re-write the equation for $er_Q(h)$ using

$$P\{h(x_i) \neq t(x_i)\} = er_P(h) \implies P\{h(x_i) = t(x_i)\} = 1 - er_P(h), \tag{18}$$

so that

$$er_Q(h) \;=\; (1-\eta)er_P(h) + \eta(1 - er_P(h)). \tag{19}$$

and rearrange to obtain the final solution expressed as

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

Let us assume that $er_P(h) \geq \epsilon$, where ϵ is some arbitrarily chosen small constant, using our Lemma 1 to obtain

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

where s denotes $s = (1 - 2\eta)\epsilon$. Hence,

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

where $P(a, b)$ denotes the joint probability of events a and b, under P. Now, given that $Q(.) = \eta + (1 - 2\eta)P(.)$, then $Q(.)$ is linear in η, $0 < \eta < 0.5$. 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 a good learning algorithm. Therefore, $Q(.) \geq P(.)$, and we can substitute with Q^m in the last equation to obtain

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

We next express an upper bound on $P^m(er_P(h) > \epsilon)$ as

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

Using a result in [1], this upper bound 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 $er_z(t') \geq \eta + \frac{s}{2}$, as well as by (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 is now expressed in terms of Q^m, producing

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

The second term on the right hand side is next re-written, considering the condition $er_z(h) < \eta + \frac{s}{2}$ or alternatively

$$\eta > er_z(h) - \frac{s}{2}, \tag{20}$$

Therefore,

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

and

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

which gives us a bound on the probability for $P^m(er_P(h) > \epsilon)$. Thus, 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 useful inequality gives us a means to quantify the accuracy or error of our hypothesis h with a degree of statistical confidence. Such quantification is important in financial forecasting applications.

The inequality also demonstrates that the probability of the error function exceeding ϵ is not only dependent on the sample error $er_z(h)$ but also on the level of noise itself. Both $+\eta$ and $-\eta$ related terms (such as s) are present in the inequality, and so noise impacs on the probability of the error function in both directions. This is a reassuring result, as one would expect noise to affect algorithms' accuracy, demonstrating the importance of noise for algorithmic performance.

3.2 Application of the Vapnik-Chervonenkis Inequality

In the previous Sub-section, we have derived a bound on the probability of the error function, providing a useful quantity in terms of the accuracy of a learning algorithm. However, this quantity does not tell us about sample complexity or training data length (m) required for PAC learning. Sample complexity is fundamental to machine learning, and to financial forecasting applications involving machine learning, and has significant implications on various aspects of algorithms. The Vapnik and Chervonenkis inequality [26, 24] is an important Theorem in machine learning theory (see *Theorem 4.3* in [2] for more information), and helps quantify sample complexity. The VC inequality relates to the last term in our equation for m, and provides useful information anout m. The VC inequality, as given in [2], is as follows:

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 $h \in H$*

$$P^m\{|er_P(h) - er_z(h)| \geq \epsilon\} \leq 4\left(\frac{2\kappa m}{d}\right)^d e^{\frac{-\epsilon^2 m}{8}}, \tag{21}$$

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

The VC inequality importantly shows that provided the training sample is large enough, then with a sufficiently high 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 at a slower pace. Hence, training data m is important to reducing error in any learning algorithm. This is a key result for ensuring that one is able to obtain good estimations in forecasting applications. The VC Inequality is a particularly relevant Theorem to financial forecasting applications, because this inequality is independent of any probability distribution. Hence, the inequality is pertinent to a wide range of forecasting applications in finance, where a diverse range of distributions exists. We would like our learning algorithms to be distribution independent, as dependency would impose a significant constraint on forecasting applications.

Next, we apply the VC inequality in derivating our new Theorem. Since $s =$

$(1-2\eta)\epsilon \Rightarrow \epsilon > s \Rightarrow e^{-\epsilon} < e^{-s}$. Therefore, thenusing Lemma 2 produces

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

If we rewrite the last inequality then then the result is that $\exists\, h \in H$, such that

$$Q^m \left(|er_Q(h) - er_z(h)| \geq \frac{s}{2}\right) \leq 4\left(\frac{2\kappa m}{d}\right)^d e^{-\frac{s^2 m}{32}}. \tag{22}$$

Let us assume that δ is bounded below by

$$4\left(\frac{2\kappa m}{d}\right)^d e^{-\frac{s^2 m}{32}} \leq \frac{\delta}{2}, \tag{23}$$

which in a rearranged version is

$$\frac{s^2}{4} \geq \frac{8}{m} \ln\left(8\frac{\left(\frac{2\kappa m}{d}\right)^d}{\delta}\right), \tag{24}$$

or rearranging alternatively gives us

$$m \geq \frac{32}{s^2}\left(d\ln m + d\ln\left(\frac{2\kappa}{d}\right) + \ln\left(\frac{8}{\delta}\right)\right). \tag{25}$$

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

The application of the VC inequality provides 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 in such a way that as noise increases, the required training-data length (m) also increases. Therefore, algorithms achieve 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

Though our Eq. (25) provides an inequality for m, it does not provide a particularly tractable boundary on m. Eq. (25) contains m on both sides of the inequality, and one cannot easily understand the behaviour and impact of m. Further, it is not possible to separate out terms, so that m's boundary is expressed in terms of other variables. Therefore, the boundary on m is not easily tractable or understandable, particularly if we wish to understand the impact on financial forecasting applications in risk management. In order to obtain a more useful bound on m and to prove our main Theorem, we first introduce and prove our Lemma 3.

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

$$\ln m \leq \frac{s^2}{64d} m - \ln\left(\frac{s^2}{64d}\right) - 1. \tag{26}$$

Proof: First, let us consider the function

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

and differentiate it to produce

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

Therefore, the minimum value of f is at $y = 0$, with $f(0) = 1$, and for any other y,

$$f(y) \;\; \geq \;\; 1.$$

Next, by substitution,

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

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

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

Let us substitute y with $y = \alpha x - 1$, $\forall x$ where $x > 0$ and α is a positive constant $\alpha > 0$. Therefore,

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

Now take logarithms on both sides of the expression, and rearrange

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

Next, we make the substitutions $x = m$ and $\alpha = \frac{s^2}{64d}$, and produce

$$\ln m \leq \frac{s^2}{64d} m - \ln\left(\frac{s^2}{64d}\right) - 1. \tag{27}$$

This proves Lemma 3. ■

Now we will prove our Theorem 2. The logarithmic inequality in m is applied to Eq. (25), so that

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

If we now simplify further the right hand side, then

$$\frac{32}{s^2}\left(d \ln m + d \ln\left(\frac{2\kappa}{d}\right) + \ln\left(\frac{8}{\delta}\right)\right) \;\; \leq \;\; \frac{m}{2} + \frac{32d}{s^2}\ln\left(\frac{128}{s^2}\right) + \frac{32}{s^2}\ln\left(\frac{8}{\delta},\right) \tag{28}$$

and using Eq. (3), the inequality on m from Eq.(25)) now becomes:

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

Finally, we derive

$$m \geq m_0,$$

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

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

This proves our main Theorem and tells us the minimum sample length m required to achieve learning within the PAC framework. The sample length significantly impacts the practicability of any computation. A large m may mean large computation times, significant computational resources, and may render a method impractical or unworkable.

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 such applications, we must first consider the requirements for ϵ and δ. They both should be as small as possible, as ϵ denotes the error in the algorithm, and δ denotes the percentage of bad hypotheses. Consequently, ϵ and δ represent a source of model risk in our algorithms. In the financial sector, risk is typically measured in terms of Value at Risk (VaR) and usually set at the 99^{th} percentile. By analogy, although there is no fundamental theory for choosing the 99^{th} percentile, we apply this percentile choice to our model for ϵ and δ and set $\epsilon < 0.01$, $\delta < 0.01$.

4.1 PAC Learning: Big Data Implications

Theorem 2 does not require $|H|$, which is a key advantage. The AL Theorem requires $|H|$, and can only be informative for finite $|H|$. However, machine learning algorithms typically work with $|H| = \infty$ (see [18], for example). Consequently, a the requirement for a finite $|H|$ is highly restrictive. In financial applications, there further exist highly complex and non-trivial patterns and data sources, and typically non-trivial algorithms are employed [6, 4]. An assumption of finite $|H|$ is unrealistic for financial risk maangement applications. Furthermore, Theorem 2 does not require $|H|$ but rather involves d. Although d is related is to $|H|$, a condition $|H| = \infty$ does not imply $d = \infty$. 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 the AL Theorem, and provides us with the flexibility to determine m or data bounds for a wider range of algorithms in finance.

Table 1: Sample size m_T according to our Theorem, 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

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 the results in Table 1. The calculation of m is perfomred 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 $\delta = 0.01$.(Note that AL also assumed values for ϵ and δ in [1].) Table 1 provides the values of m, according Theorem (m_T), for different vales of ϵ and δ, and it is observed that m_T is in the order of millions. We also observe that m_T is in the order of $100,000$ only if ϵ is of the order of 0.1 or above. However, financial applications require high accuracy and expect $\epsilon \leq 0.01$.

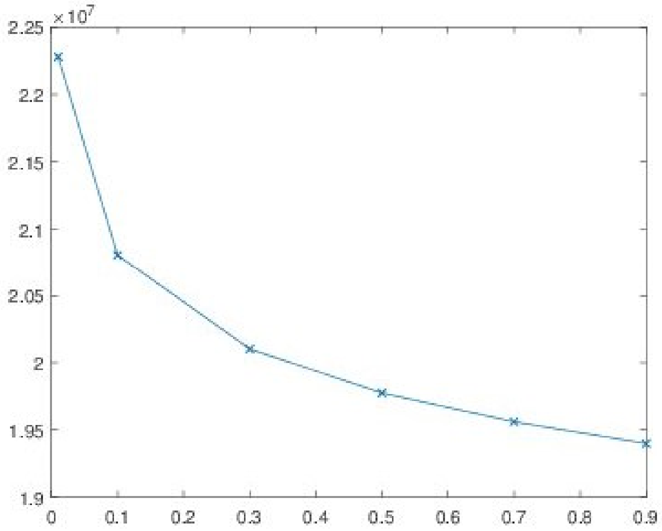

Figure 1: Graph of m_T $(y - axis)$ against δ $(x - axis)$ for $\epsilon = 0.01$

The fact that m_T is in the order of ˙millions has important implications. For financial data for example, given that a stock price is typically quoted to 4-6 signifi-

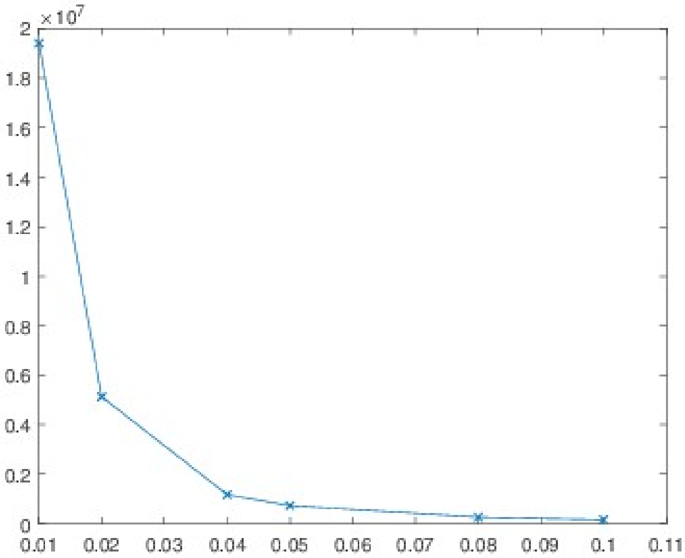

Figure 2: Graph of m_T $(y - axis)$ against ϵ $(x - axis)$ for $\delta = 0.01$

.

cant figures, the learning algorithms will require approximately 1TB of data [5, 10]. As mentioned earlier, there is currently no strict definition of big data but 1TB has been a proposed definition. Therefore, our Theorem implies that learning algorithms in the presence of noise require big data, for the purposes of financial risk management applications. This big data requirement is a revealing conclusion, since such conclusion is not necessarily revealed using the AL Theorem. Table 2 provides m according to the AL Theorem (m_A), for different ϵ and δ values. We set $\epsilon = 0.01$, $\delta = 0.01$, and $\eta = 0.1$, to give more realistic values. The results show that m_A is in the range of $10,000 - 100,000$, and so AL does not imply that big data is required for financial applications. We expect $|H|$ to be large for financial applications, due to the sophisticated algorithms required to analyse complex data. However, Table 2 shows that even when $|H|$ increases exponentially, its impact on datapoints required (m_A), according to AL, does not increase significantly and hardly reaches 1 million. This is due to the logarithmic dependence on $|H|$ in $ln(2|H|/\delta)$ in the AL inequality, which leads to $|H|$ having little impact on m_A.

If m is calculated using our Theorem, which does not depend on $|H|$, then the equivalent value of m_T for all different values of m_A in Table 2 is $m_T = 35,701,927$. In this calculation, we set $d = 2$ so that m_T gets a lower limit value. In real world applications, d>2 and m_T increases further (this can be understood by examining the equation). In summary, AL does not imply big data and our Theorem does conclude big data is required. The implication that big data is required for financial risk management applications is significant. In such applications, risk may be re-evaluated frequently, i.e. calculating VaR on a daily basis, which implies that not only good-learning algorithms but also fast algorithms are required. Big data can require significant processing time and so fast learning algorithms are a must in finance. A second implication is that fast computation time will require higher-end hardware to cope with learning algorithms for big data on financial risks. Data storage issues and data curation (organisation and collation) also need consideration

Table 2: Sample size m_A according to Angluin and Laird's Theorem

$\|H\|$	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

when 1TB is required just for learning purposes in 1 round. A third implication of our Theorem is that some types of financial forecasting may not be possible with learning algorithms, due to limited historic data-points.

4.2 Impact of Big Data Quality (Noise)

Noise or corruption of data can occur for a number of reasons in real world applications. A conclusion in [29] is that noise is unavoidable in affecting data. Noise exists in any measurement or recording system, distorting the original data. Sometimes, data can be transformed, e.g. discretising data or converting it into a binary form, which can lead to noise in the data in various ways [31]. To understand the impact of noise on data requirements, we calculate m_T for different values of η, where we recall that $0 < \eta < 0.5$. As before, the other parameters are set to $\epsilon = 0.01$, $\delta = 0.01$, and $d = 2$. For comparison, we also calculate m_A and set $|H| = 10^{12}$ to give an optimistic calculation of m_A. Table 3 gives the m values under AL (m_A) and under our Theorem (m_T), with the final column giving the percentage increase. The results show that the impact of noise is significant. Approaching $\eta = 0.4 - 0.45$ leads to the required data reaching the billion datapoint range, hence the demands for big data processing become even more important for noisy data. However, it should be noted that it is unlikely financial data will approach a level of 50% noise.

An insight from our Theorem is that the impact of noise is far more significant than what may be assumed under AL. Under AL, though the requirement for data is in the order of millions when η approaches 0.1, the bound for m_A is still 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. On the other hand, our Theorem does not assume finite $|H|$ and we have provided a conservative estimate by setting d to 2. The impact of noise increases the data requirements from multi-millions to billions, and the incremental increase in m_T is far higher than in m_T. The difference on noise dependence by examining the equations. This is due to m_T having a logarithmic dependence on the reciprocal of $1 - 2\eta$ in our Theorem, and m_A depending only on the reciprocal of $1 - 2\eta$ in the

Table 3: Sample sizes m_A and m_T for different levels of 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

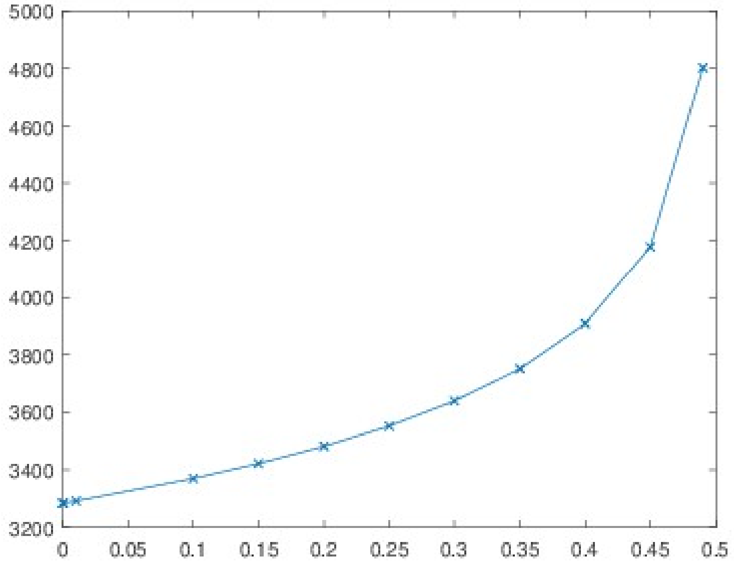

Figure 3: Graph of percentage difference between m_T and m_A $(y - axis)$ against noise $(x - axis)$

AL Theorem.

Given that noise is practically always present in real world applications, our Theorem provides an insight into impact of noise. This is particularly important when dealing with big data volumes, because noise is typically far more complex in big data than in conventional data sizes. A larger dataset is more likely to contain more complex noise processes than shorter datasets. Additionally, the quality of data captured in big data can vary far more than in traditional data, with more gaps and potential distortions in the data. Hence, eliminating noise from big data is typically harder to achieve than in smaller datasets. It is further well known that noise in financial data is highly non-trivial; in fact, many analysts have stated it is the noisiness of financial data that prevents identifying their patterns, and so prevents managing risk better.

Another insight from our Theorem is that noise, or equivalently data quality, can have a more significant impact on computational productivity than assumed before by other authors. In other words, having high-quality data ($\eta = 0.1$) rather than low-quality data ($\eta = 0.4$) leads to a reduction of data by a factor of approximately 20. With all other factors equal, this would lead to a processing speed of 20 times faster and a significant productivity gain. This is an important conclusion as the current trend in big data analysis is that higher volume is generally better. Our analysis shows that data quality is rather significantly effective in providing for better learning outcomes. Consequently, we suggest that big data applications should focus more on data cleaning applications and producing high-quality data, rather than focussing on higher-volume data that can be very noisy. The additional advantage of focussing on data cleaning methods is that they are normally cheap and easy to implement, unlike other big data techniques.

4.3 Impact of Algorithmic Capability

The dependence of training data upon the VC dimension d, is also a new insight from our Theorem. 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 non-trivial and that incorrect modelling can lead to significant financial losses, highly sophisticated algorithms are typically required to ensure better financial risk management. on the other hand, the AL Theorem does not include the VC dimension and 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. It is also well known in the financial sector that many parsimonious models are preferred to more sophisticated models, due to the data requirements imposed by them.

To investigate the impact of d on data requirements, we calculate m_T next using our Theorem, for different values of d. For the benefit of clarity, we set $\eta = 0.1, \epsilon = 0.01, \delta = 0.01$, and then vary d from 1 to 30 to observe 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 [19]. Given that neural networks are used in financial applications and represent more sophisticated algorithms, the upper limit of 30 is a suitable value for our study. The results are presented in Table 4, and show that the data required grows proportionately with d. Therefore, d is important in big data. For the benefit of comparison, we also calculate m_A using the AL Theorem for the equivalent parameter values, and set $|H| = 10^{12}$ to provide optimistic calculations. For all values of d in Table 4, the AL data requirement is $m_A = 1,029,042$. Therefore, even for large $|H|$, the 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.

Furthermore, our Theorem reveals that d provides a direct "proxy" on the data requirements, as m_Tm increases linearly with d. This is important in financial risk applications, because non-trivial relationships in financial data call for complex learning algorithms but the more complex algorithms lead to larger data require-

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

ments (through increasing d) and may not be preferable. Hence, our Theorem tells that the linear relation between m_T and d implies there is a direct trade-off between algorithmic complexity and sample complexity.

5 Conclusion

In this paper we investigate Probably Approximate Correct learning in the presence of noise. We derive new theoretical results in relation to big data Probably Approximate Correct learning. In particular, we derive a new a Theorem on the theoretical bounds on the training data required for Probably Approximate learning, in the presence of noise. A direct consequence of this derivation is that we 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 the following 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 more substantial impact on the data size required for PAC learning. Consequently, contrary to current big data trends, we demonstrate that higher quality data can be more important than larger 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, transportation applications etc.. Whilst computational methods have applications for a range of areas, many computational methods can be optimised, in terms of processing speed and quality of results, by adapting their methods for specific tasks. This can potentially produce a new line of research with high impact.

Another potential area of future research that we would like to investigate is methods with respect to data filtering, to reduce noise in any given set of data. As mentioned and analysed in our paper, the issue of noise (especially in the context of big data) is a key topic, and the nature of the noise itself can fundamentally differ compared to small sample datasets. Moreover, the removal of noise from data is an important factor in improving the learning performance for Probably Approximate Learning algorithms.

Finally, we would like to develop our results further for the purposes of financial risk management. For example, we would like to apply our results to Extreme Value Theory, and examine the impact of Probably Approximate Correct learning theory upon modelling extreme value events. Given that extreme events, such as the Global Financial Crisis, have a significant impact on economic, political and social issues, this would also be a productive research area. 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] Blum, A., Kalai, A., Wasserman, H.:

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

[9] Chandrinos, S.K., Sakkas, G., Lagaros, N.D.: Airms: A risk management tool using machine learning. Expert Systems with Applications **105**, 34–48 (2018). DOI 10.1016/j.eswa.2018.03.044

[10] 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)

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

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

[13] García, F.J.P.: Exchange rate risk. Financial Risk Management p. 135–153 (2017). DOI 10.1007/978-3-319-41366-2_6

[14] JENSEN, H.W., CHRISTENSEN, N.J.: Optimizing path tracing using noise reduction filters p. 134–142 (1995)

[15] Kearns, M.: Efficient noise-tolerant learning from statistical queries **45**(6), 983–1006 (1998)

[16] 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)

[17] 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)

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

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

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

[21] SEN, P., DARABI, S.: Implementation of random parameter filtering (2011)

[22] Simon, H.U.: An almost optimal pac algorithm p. 1552–1563 (2015)

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

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

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

[26] 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)

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

[28] Xu, W., Chen, Y., Coleman, C., Coleman, T.F.: Moment matching machine learning methods for risk management of large variable annuity portfolios. Journal of Economic Dynamics and Control **87**, 1–20 (2018). DOI 10.1016/j.jedc.2017.11.002

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

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

[31] 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)