Factors related to dataset that influence the shape of learning curves

Nam Thang Bui Supervisor(s): Tom Viering, Marco Loog
EEMCS, Delft University of Technology, The Netherlands

June 19, 2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

Although there are many promising applications of a learning curve in machine learning, such as model selection, we still know very little about what factors influence their behaviours. The aim is to study the impact of the inherent characteristics of the datasets on the learning shapes, which are noise, discretized input and dimensionality. We trained two classifiers with a panoply of datasets for the investigation to see how the learning curve behaves under different circumstances. Firstly, we found that the shapes of the curves varied with different levels of noise injected into the original datasets. Secondly, using the equal width interval binning technique to discretize continuous features did not make the classifiers learn exponentially but caused the learning curves to behave unpredictably; thus, it does not transform the continuous problem into the easier class of problems mentioned in [1]. Finally, the more dimension we reduced using the PCA technique, the learning curve showed strange behaviours.

1 Introduction

A learning curve is a tool for plotting a machine learning model's generalization performance against incremental subsets of training data. In practice, investigating the empirical learning curves of different machine learning models for various problems could give insights into the needed training data, thus improving model selection and reducing computation complexity. Indeed, numerous amounts of research have been conducted to generally model the shape of the learning curves, [1, 2, 3]. While some studies find the power-law models [2] often give good fit, others show evidence for exponential [1] and log models [3]. Recently, Bousquet et al. [4] claim that there are three main behaviours of learning curves for the optimal learners: exponential, power-law, and slow convergent. However, In an attempt to review the general shapes of the curves for classification problems, Viering and Loog find that no flawless study can conclude the universal shape of the learning curve.[5].

Decades of work on learning curve has revealed many interesting properties regarding the universal forms of the well-behaved learning curves [6], and perhaps more importantly, contributed to accelerating the machine learning field by the intelligent use of data. However, there are strange learning curves that we still do not fully understand. While different researchers show different views on the shapes of learning curves, several theorists agree that dimensionality and sample size cannot be decoupled in evaluating the learner's generalization performance [7, 8, 9]. Noise is another factor that can significantly impact the learning process. Real-world data usually contains noise that hinders the learner from detecting the data's underlying pattern, thus degrading its performance [10]. From the standpoint of computational complexity, there might be a difference in learners' behaviours regarding the input types, binary or continuous [1]. In this paper, we will conduct an empirical study to answer the question:

"How do the inherent factors related to the datasets such as noise, types of numerical input, and dimensionality influence the shapes of the learning curves?"

2 Related work

In this section, we will explore some of the literature that is relevant to our study. First, we will discuss two papers to investigate the use of artificial noise to test the robustness of various basic supervised machine learning models for classification problems. Second, we

will look at two papers that compared models' generalization performances under discrete and continuous input settings. Third, we will discuss the papers examining the relationship between dimensionality and the training set size. Finally, we will discuss the paper that tries to point out the curves' general forms.

Closely related works examined how sensitive the learners are to injected noise [11, 12]. Four different supervised classifiers: *Decision Tree* (DT), *Naive Bayes*, *Support Vector Machine*, and *Logistic Regression* were trained on various datasets which are injected varying level (0% to 50%) of mislabeled noise [11]. This work reported the average accuracies of 10 times experimenting with each of the 18 datasets from *WEKA*. While in the extreme (50% noise), *Naive Bayes* is the most resilient model; the *Decision Tree* performs best on average when the degree of noise is reduced quickly, followed by *Support Vector Machine* and *Logistic Regression*. [12] is the more extensive work, which evaluated 11 different models on three real-life and four artificial problems. The noise level was the same as [11], but they considered both the normally distributed feature noise and mislabelled noise. Using *RMSE* as a performance metric, they found the *Decision Table* and *Linear Regression* are less sensitive to noise.

Cohn and Tesauro [1] trained the neural network on four N-dimensional classification tasks by two linearly separable and high-order functions. This research aimed to see if average generalization performance might outperform the worst-case bounds derived from formal learning theory by using the *Vapnik-Chervonekis* dimension. In this work, the experiment curves of two problems with strict binary inputs behave exponentially. In contrast, the other real-valued inputs problems have the average curves' error close to the worst-case theoretical bound. Interestingly, They raised a hypothesis concerning the existence of the class of problems in the same *Vapnik-Chervonekis* dimension that is easier than the others. This hypothesis motivates us to examine whether we can transform a problem with continuous features into an easier one using the discretization technique so that the classifier can learn exponentially. Dougherty el at. [13] found that using multiple discretization techniques did not significantly degrade the performance of a model. *Naive Bayes* even showed improvements and outperformed *C4.5 algorithms* in discretized version.

Real-world data can have thousands or millions of features per instance, and considering all these features is unnecessarily computational expensiveness. As a result, it is harder to find optimal solutions. Dimensionality reduction techniques such as *Principal Component Analysis* (PCA) may reduce noise and extraneous features, resulting in improved performance, but in most cases, it will speed up training. This is because it leads to information loss, and there will be a trade-off between the model's performance and computational complexity [10]. In general, the higher the dimensionality, the more complex the problem is. As statistical theories show the relationship between dimensions and sample, [11] claimed that dimension is also an essential factor for studying learning curve behaviours, thus suggesting using both learning curves and feature curves to gain further insights.

The work [6] studied the shape of the curves and pointed out three properties: monotonicity, convexity and peaking. These properties are essential when considering extrapolation techniques. Conventionally, we assume that the error rates will decrease when the amount of training data increases. If that is the case, we say the learning curves are well-behaved; otherwise, we call them strange curves. While monotonicity and convexity are useful for identifying well-behaved curves, the peaking phenomenon is the indicator for strange curves. The degree of monotonicity is identified by measuring the *highest* negative step between any two anchor points of the learning curve. The convexity can be computed similarly with three anchor points. Peaking, on the other hand, showed improvement at the beginning but

then got worse later.

Although not all the works mentioned above directly study learning curves, they motivate us to come up with ideas to examine how learning curves behave in classification tasks under real-world data or modified data.

3 Methodology

In our study, we consider the learning curves plotting method that most uses data. Next, three methods used for *Noise, Discretization and Dimensionality* will be described in detail.

3.1 Learning curve generating

We decided to implement our learning curves because using the *Scikit-learn* implementation's learning curve used *Kfold* cross-validation, which results in wasting data when the size of the training set and k is small [13]. To overcome this drawback, we implemented the *Kfold* cross-validation with stratification. The splits k can be varied but preferably be high, as we want to avoid overfitting and reduce as much as bias possible. Next, for each fold, we obtain the size of the training set, also defined as anchor size, as a geometric sequence. Particularly, the anchor size at i index will be determined by the function $s_i = \lceil 2^{\frac{7+i}{2}} \rceil$ [6], the rest of the data is merged to the validation set for testing to avoid wasting data. Another property of the training subset is strictly increasing, and the training subset $S_i \subset S_{i+1}$. To simulate the real-world scenario in which learners' performance is evaluated on unseen data, we only apply data preprocessing and tuning on each anchor. Finally, the empirical learning curve is obtained by averaging out over k curves.

3.2 Noise model

As mentioned in section 2, there are several kinds of noise. However, we only consider attribute noise in our experiments as attribute noise in the dataset usually increases the model's complexity, thus probably degrading the learning process. In order to increase the dispersion level of the datasets, we created a noise model, which is added to the original datasets at hand. For all the datasets, we assume the following:

1. The variables of the dataset are either uniformly distributed or normally distributed.
2. Noise is also uniformly or normally distributed and independent of the dataset.

For the example pair (x_i, y_i) in the dataset L has N instances and D features: where x_i is the ith example in design matrix $X \in \mathrm{R}^{N \times D}$, and y_i is the label of the ith example in X. if the noise level $n > 0$ then the example $(x_i^{'}, y_i^{'})$ will replace (x_i, y_i) in the matrix X. The new example can be calculated as follows:

$$x_{ij}^{'} = x_{ij} + n\sigma_{xj} z_j \text{if } n > 0 \tag{1}$$

Where:

- x_{ij} is the entry of the feature j of the instance i in the original dataset which is substituted by $x_{ij}^{'}$,

- $n \geq 0$ is level of noise,
- σ_{xj} is the standard deviation of feature j
- z_j is random variables which $z_j \sim \mathcal{N}\ (\mu, \sigma^2)$ for normally distributed noise and $z_j \sim \mathcal{U}\ (a, b)$ for uniformly distributed noise.

3.3 Discretization

To examine the learning curves' behaviours with different input bases: discrete and continuous features. Instead of using different problems, we consider the equal width interval binning technique for discretizing continuous feature spaces. This technique is usually used for getting categorical values from continuous ones. Then for each continuous feature, x_j, the observable values will be sorted, and the minimum value and the maximum value are used to partition the space of the features into k equally sized bins, where k is the parameters provided by users. The new space of the feature is discretized and only contains k values starting from the range (min - ϵ, $min + \frac{min+max}{k}$] up to ($min + \frac{(k-1)(min+max)}{k}$, max].

3.4 Dimensionality

To study the impact of dimensionality on the learning curves, the dimension reduction technique PCA is used to form new versions with different features. With these new datasets, two classifiers are trained and produce learning curves. These curves are then compared with the original curves to investigate any shape changes.

4 Experimental setup

In this section, the details of our experimental setup will be specified. The source code is written in *Python* with the utilization of *Scikit-learn* library for training machine learning models, and *OpenML* is the platform for data retrieving.

There are more than 20 classification algorithms from the *Scikit-learn* library, which can be categorized as linear classifiers or non-linear classifiers. In this initial work, we only considered one from each category, such as *Linear Support Vector Machine* (SVC) and *Decision Tree Classifiers* (DTC). For all three experiments, we used *Kfold* cross-validation with $k = 25$ to avoid overfitting. The performance metric was the normal error as it is easy to measure and understand. The error can be calculated as the formula: $1 - \frac{True\ Positive\ +\ True\ Negative}{TruePositive+TrueNegative+FalsePositive+FalseNegative}$. Preprocessing data was applied to each anchor to simulate real-world settings. The attributes with missing values were preprocessed by using mode and mean for categorical and numerical features, respectively. Then the standard scaler was utilized, as LinearSVC tend to perform better with it [10]. The hyperparameters' distribution used for tuning LinearSVC was $C = [0.1, 0.5, 1.0, 5.0, 10.0]$ and DT was $\{max\_depth : [10, None], criterion : [\text{"}gini\text{"}, \text{"}entropy\text{"}]\}$. For the *Discretization* experiments, the number of bins k used was 10. the Table 1 below gives information on the datasets used in the experiments. For convenience, we will refer to the dataset using its name and openmlid onward. The dataset yeast (181) and diabetes (42608) were used for both *Noise* and *Discretization* experiments since they have reasonable sizes (1484 and 768 instances, respectively) and they only contain continuous features. The rest five datasets were for *Dimensionality* experiments. Particularly, the fri_c0_1000_5 (799) was used for

experimenting with PCA, and the other four represent the same problem (1000 instances) but have different numbers of features (5, 10, 25, 50 features).

OpenML ID	Name	Nominal Attributes	Numeric Attributes	Number of Instances
181	yeast	0	8	1484
799	fri_c0_1000_5	0	5	1000
866	fri_c2_1000_50	0	50	1000
903	fri_c2_1000_25	0	25	1000
912	fri_c2_1000_5	0	5	1000
913	fri_c2_1000_10	0	10	1000
42608	diabetes	0	8	768

Table 1: Datasets used for all experiments

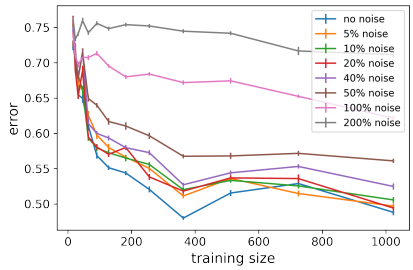
5 Results

This section shows the results of experiments with *Noise, Discretization,* and *Dimensionality*. First, the process of each experiment is described, and then we will discuss some key observations. The visualization used for displaying the results is learning curve plots. Furthermore, the standard errors of 25 curves at each anchor point are also included in the plots.

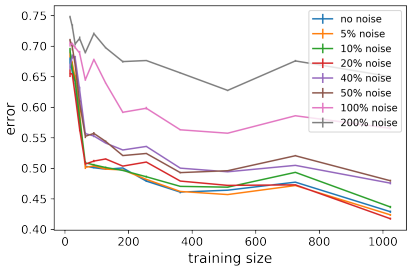
5.1 Can feature noise influence curve's behaviour?

Three experiments on two datasets were conducted to examine the impact of feature noise on learning curve behaviour. First, the normally distributed noise was injected into the original datasets. Moreover, the other two experiments are for the uniformly distributed noise. All experiments used the same levels of noise n in the formula (1) section 3.2, which is varied from 0% (no noise) to 200% (very noisy). The level of noise n multiplied by the standard deviation of original datasets, which are then injected into the original ones to generate new noisy datasets.

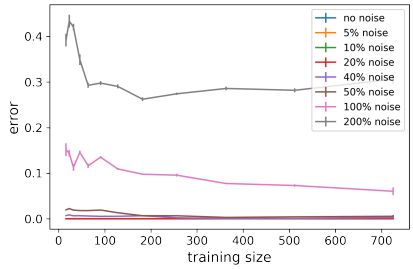
5.1.1 Normally distributed noise



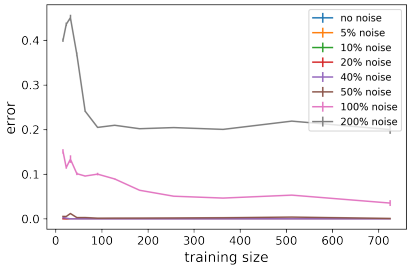
(a) Learner: DecisionTree, dataset: yeast



(b) Learner: Linear SVC, dataset: yeast



(c) Learner: DecisionTree, dataset: diabetes

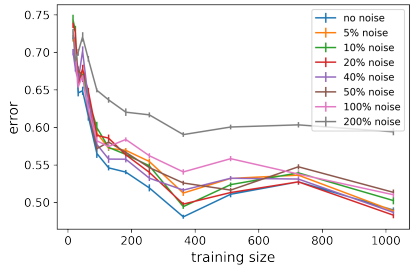


(d) Learner: LinearSVC, dataset: diabetes

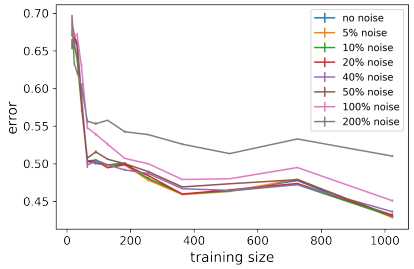
Figure 1: Learning Curve comparisons of DecisionTree and LinearSVC on 2 datasets: yeast (top 2 figures), and diabetes (bottom 2 figures). These plots compare the learning curve of the original dataset with the learning curves of normal distributed noise datasets. The degrees of noise increase from $\mathcal{N}$ $(0, (10\%\sigma_X)^2)$ to $\mathcal{N}$ $(0, (200\%\sigma_X)^2)$.

Figure 1 above shows various learning curves of 2 learners on two datasets (yeast and diabetes). As we can see, all learning curves seem not well-behaved. The first observation is that the noisy learning curves stay above the original curve for most of the anchor points in both learners. The higher the degree of noise, the higher the error difference between the noisy and zero-noise curves. Secondly, regarding the shape of the curves, they are not monotonically decreasing because there is some increase in errors at some anchor points. Additionally, the original curve learns faster from the beginning to the anchor size of 363 (top two figures) and when the anchor size increases. When the noise level is too high (above 50%), the error increases quickly, and the curve shows some strange patterns. Notably, the noisy curves of both learners have many peaks at different anchors. Some curves seem monotonically decreasing from the certain anchor point (100% curves of Figures 1a, 1c, 1d). Crossing behaviour is another observation; for example, the learners perform better at 20% noise at some anchor points than the lower noise levels.

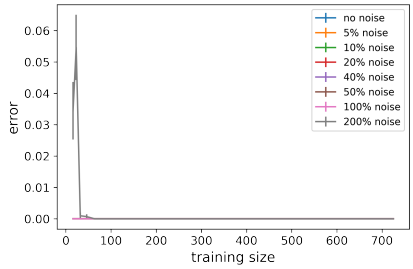
5.1.2 Uniformly distributed noise with Non-negative bounds



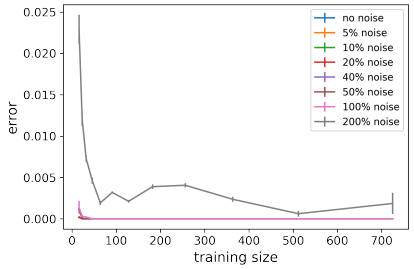
(a) Learner: DecisionTree, dataset: yeast



(b) Learner: LinearSVC, dataset: yeast



(c) Learner: DecisionTree, dataset: diabetes

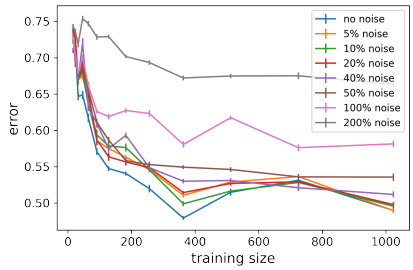


(d) Learner: LinearSVC, dataset: diabetes

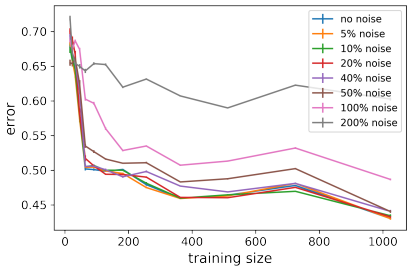
Figure 2: Learning Curve comparisons of DecisionTree and LinearSVC on 2 datasets: yeast (top 2 figures), and diabetes (bottom 2 figures). These plots compare the learning curve of the original dataset with the learning curves of uniformly distributed noise datasets. The degree of noise increase from $\mathcal{U}(0, 10\%\sigma_X)$ to $\mathcal{U}(0, 200\%\sigma_X)$

The critical observation we can make from the results in Figure 2 is that the curves are not well-behaved and monotonic, as they have peaks at some anchor points. In addition, some noisy curves cross each other and the zero-noise curves. The 20% noise curve outperformed the original curve at the highest anchor point (2a and 2b).

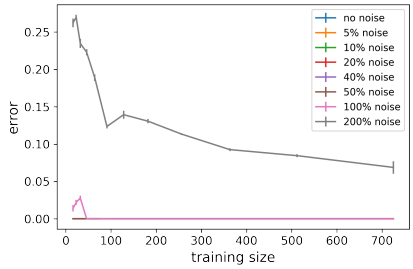
Negative min and positive max



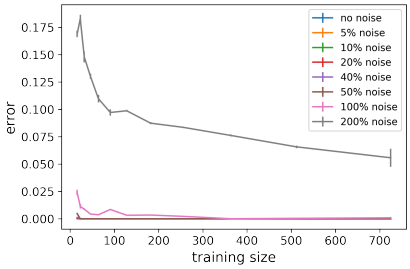
(a) Learner: DecisionTree, dataset: yeast



(b) Learner: LinearSVC, dataset: yeast



(c) Learner: DecisionTree, dataset: diabetes



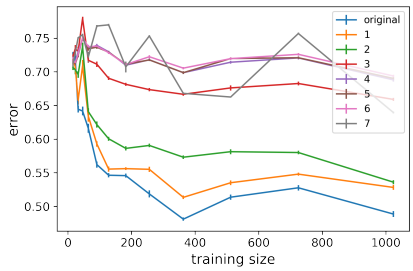
(d) Learner: LinearSVC, dataset: diabetes

Figure 3: Learning Curve comparisons of DecisionTree and LinearSVC on 2 datasets: yeast (top 2 figures), and diabetes (bottom 2 figures). These plots compare the learning curve of the original dataset with the learning curves of uniformly distributed noise datasets. The degrees of noise increase from $\mathcal{U}(-10\%\sigma_X, 10\%\sigma_X)$ to $\mathcal{U}(-200\%\sigma_X, 200\%\sigma_X)$

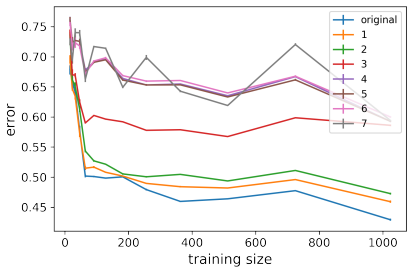
Figure 3 shows the results of the similar experiments in Figure 2, but different kinds of bounds for uniform distribution. For negative lower and positive upper bounds, both learners again show significant degradation in performance when the noise level is pretty high, and all the curves show no indicator of monotonicity.

5.2 Can discretized curves behave exponentially?

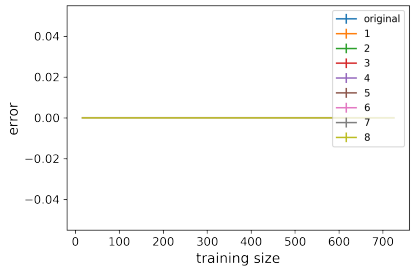
In order to check whether the learning curves of discrete value feature datasets have any differences in shapes compared with continuous value features. We trained two learners on two datasets that only contain continuous features. The number of discretized features varied and depended on different datasets.



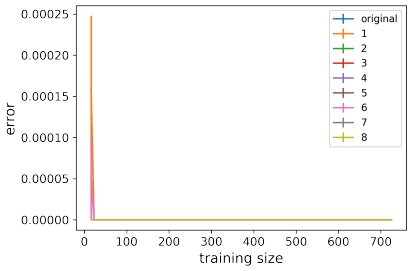
(a) Learner: DecisionTree, dataset: yeast



(b) Learner: LinearSVC, dataset: yeast

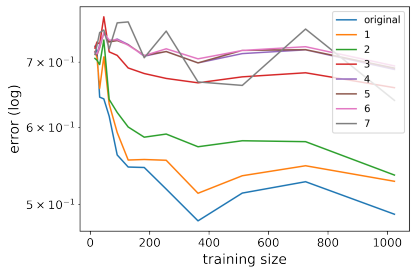


(c) Learner: DecisionTree, dataset: diabetes

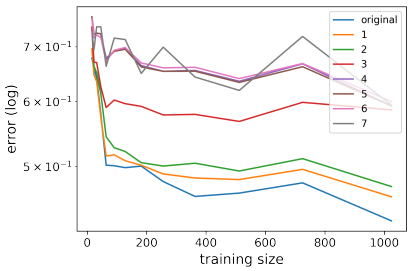


(d) Learner: LinearSVC, dataset: diabetes

Figure 4: Learning curves comparisons between datasets with different numbers of discretized features. The number of bins used in the experiments is 10, and the numbers of discretized features increases from 0 to *m-1*, with m is the numbers of continuous features



(a) Learner: DecisionTree, dataset: 181



(b) Learner: LinearSVC, dataset: 181

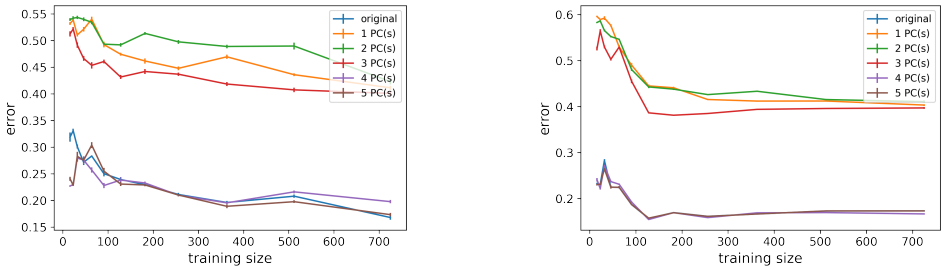
Figure 5: The log transformation of Learning curves comparisons of the dataset yeast (181) with different numbers of discretized features. This plot is used to investigate whether the curves' shapes are exponential

Figure 4 above shows some interesting results. There are two key observations we can make in these experiments. The bottom two figures represent the curves of the problem (dataset diabetes) considered easy. In this experiment, both learners show no problem with any number of discretized features. However, with the slightly more complex problem, as we can see, the more features we discretize, the more likely their shapes varied. Even with small discretized features (less than 7), the learning curves are not well-behaved, as at some anchor points, the error increases. When the number of discretized features is

high (7 discretized features), both DecisionTree and LinearSVC's curves clearly show zigzag patterns with multiple peaks. The log scale learning curves in Figure 5 also demonstrate the same behaviours as the unscaled ones.

5.3 Does low dimensionality lead to a strange curve?

To study how the dimensionality of the dataset influence the shapes of the learning curves. There are two experiments with 5 datasets: fri_c0_1000_5 (799), fri_c2_1000_50 (866), fri_c2_1000_25 (903), fri_c2_1000_5 (912), fri_c2_1000_10 (913). The experiment with dataset fri_c0_1000_5 is to check how PCA techniques affect the learning curves by progressively reducing the dimensions of the original datasets. Furthermore, the experiment with the other four datasets is from the same problem with the same number of instances but different numbers of features.

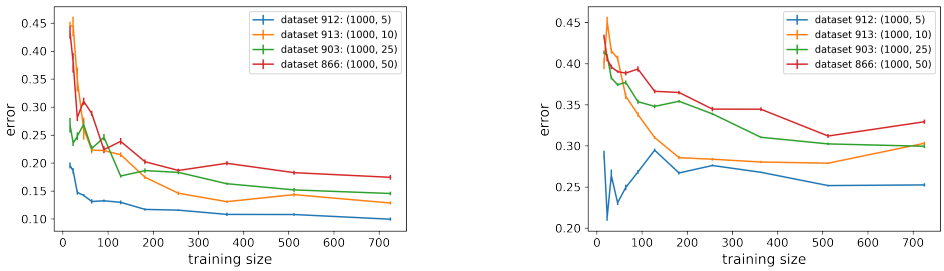


(a) Learner: DecisionTree, dataset: 799

(b) Learner: LinearSVC, dataset: 799

Figure 6: Learning curves comparisons between dataset fri_c0_1000_5 with its variants. These modified dataset are generated by using PCA dimension reduction techinque

From Figure 6 above, we can see that both learners' performance drastically drops when reducing too many dimensions. As a result, the curves of low dimensionality datasets (less than four dimensions) stay above the high dimensionality ones. Furthermore, the curves of lower dimensions cross each other; similarly, high dimensionality curves (greater than three dimensions) also have crossing behaviours. Both LinearSVC and DecisionTree's curves are not well-behaved and monotonic. However, they seem to converge to some error when the anchor increases.



(a) Learner: DecisionTree, dataset: 912, 913, 903, 866

(b) Learner: LinearSVC, dataset: 912, 913, 903, 866

Figure 7: Learning Curve's comparison between the different datasets of the same problem. These datasets are only different in numbers of features

In this experiment (Figure 7), we can see that only DecisionTree's curve of the dataset fri_c2_1000_5 (912) seems well-behaved and monotonically decreasing. The other curves are not monotonic as they have many peaks. Notably, The gradients of fri_c2_1000_5 (912) learning curves go down very quickly when the anchor size increases. Besides, crossing behaviours also happened with the curves of datasets: fri_c2_1000_5 (913), fri_c2_1000_5 (903), and fri_c2_1000_5 (866).

6 Discussion, limitation and future recommendation

In this section, we first discuss the general findings of the experiments and give a justification for the results. Following that, there will be some reflections on the study's limitations and how further research is done to help the future study.

6.1 General findings

The first important finding is that the learning curves of DecisionTree and LinearSVC have unwell-behaved shapes under different noise levels. Moreover, the more noise added to the datasets, the higher the error increase and the stranger the learning curve behaves. This could be explained by the way we design our noise experiments. Not Surprisingly, all kinds of noise distribution make the learning curves not well-behaved and monotonic. Whereas under low noise levels (below 100%), crossing behaviours happened at some anchor points because the random noise, dependent on standard deviation σ_X, might, by chance, distort the original datasets such that the instances of classes are separable easily. Under the high noise level (above 50%), the original datasets were distorted so that the feature space of many instances of different classes overlapped. Consequently, DecisionTree and LinearSVC, which utilizes feature space separation for classifying, have trouble making the decision.

The second finding is that the learning curves of discretized feature datasets created by equal width interval binning do not behave exponentially. The log transformation can verify this in Figure 5. If the discretized curves' shapes have exponential law properties, then the error log on the y-axis and the training size on the x-axis will be the straight line, as they have a linear relationship. However, this is not the case, as the log transformation learning curves still show strange shapes with a zigzag pattern. This could imply that these modified datasets do not belong to the class of problems proposed by [1]. In fact, under our

experiments, the more discretized features we have, the shape of the curve starts to behave unpredictably. This is because combining too many values associated with many different classes by using binning causes classification information loss; thus, it is difficult for learners to make predictions.

The final finding regarding dimensionality is that an ideal number of features always exist such that the curves behave exponentially. The results are in line with what is found in [5]. Additionally, dimension reduction techniques like PCA cause the learning curves to behave unpredictably, leading to strange curves. This is because it is harder to preserve the variance of the original dataset with a low number of principal components. Therefore, the lower number of principal components, the stranger the learning curves behave.

6.2 Limitations and future recommendations

The assumptions regarding the distribution of noise and datasets are not necessarily accurate. Furthermore, we only consider feature noise with specific noise degrees. These open up ideas for experiments with higher noise levels or different combinations of feature noise, outliers noise or mislabelled noise to see how significantly the noisy curves deviate from the original curves.

Discretization experiments only utilized the simplest discretizing technique, and the number of bins is quite arbitrary. Hence, experimenting with more k and figuring out the best way to choose k could lead to more insights into why such learning behaviours happened. Furthermore, to verify whether discretization can transform one class of problem (difficult one) into the class of problem (easy one) that [1] proposed, we should experiment with more advance discretization techniques described in [13]

Dimensionality experiments inherently have limitations of PCA techniques. Future work should consider these drawbacks by using different techniques to reduce the dimension of the datasets. Another thing worth noticing is applying feature selection and feature engineering techniques to investigate the influence of dimensionality on learning curve shape.

7 Conclusions

In summary, we showed an empirical comparison of the original learning curves with the learning curves on the modified datasets, which are *noisy* datasets, *discretized* datasets and datasets transformed with PCA. First, we found that with the experimented noise level, the noisy learning curves seem not well-behaved, and their shape tends to vary under different noise levels. However, the results may differ with datasets, different types of noise, and different noise levels. Second, under discretization settings, we also show that equal width interval binning could prevent the learners from achieving effective classification due to the information loss caused by this simple technique. As a result, learning curves of discretized problems generated by binning techniques do not behave exponentially; thus, these problems cannot be in the class of problems in [1] proposed. Finally, the learning curves start to strangely behave if we try to reduce as many dimensions with PCA. This indicates that dimension reduction techniques like PCA should be used with care and suggests other techniques can be utilized to study the effect of dimensionality on learning curves. Study learning curves' behaviours on the datasets, which are only different in the number of features, also indicate that there should be an ideal dimension at any anchor point such that the error rate decreases the most.

8 Responsible Research

In this section, we will reflect on the reproducibility of our experiments and the scientific integrity of our report. While computation plays a vital role in the scientific community, the credibility gap between experiments' details and verification results is getting larger. Therefore, the reproducibility of our experiments should be one of our priorities. To reflect on this aspect, we will follow Yale Law School's six recommendations [14]. The first recommendation is that the source code should be public. The source code of our experiments is hosted on Github and can be found and accessed via this link. We did not expect our code to change, so we did not follow the version recommendation. Regarding the computer environment, all experiments ran locally on HP ZBook Studio x360 G5 with an i7-8750H CPU @ 2.2GHz, which installed Windows 11 Education 64-bit. The software required for running the code can also be found in the *yaml* file in the Github repository. MIT license was used for reusing purposes; in other words, anyone can use and adapt our code to any experiment. Next, anyone can find and access our paper via TUDelft repository, which also follows the fifth recommendation. The last recommendation was resolved as the code was written in Python and used one of the most popular frameworks, which can be seen as readable code for the future. The next essential aspect of Responsible research is scientific integrity. The code and data are provided to avoid any misconduct pitfalls. Moreover, any modification of data was communicated in section 3, which helped increase the transparency. The work not originally thought by ourselves was referred to, and limitations were also reflected in the Discussion section.

Acknowledgement

We want to extend our sincere thanks to one member of our research group Dean for his valuable contribution to our experimental setup.

References

[1] D. Cohn and G. Tesauro, "Can neural networks do better than the vapnik-chervonenkis bounds?" in *Advances in Neural Information Processing Systems*, R. Lippmann, J. Moody, and D. Touretzky, Eds., vol. 3. Morgan-Kaufmann, 1990. [Online]. Available: https://proceedings.neurips.cc/paper/1990/file/816b112c6105b3ebd537828a39af4818-Paper.pdf

[2] L. J. Frey and D. H. Fisher, "Modeling decision tree performance with the power law," in *AISTATS*, 1999.

[3] S. Singh, "Modeling performance of different classification methods: deviation from the power law," *Project Report, Department of Computer Science, Vanderbilt University, USA*, 2005.

[4] O. Bousquet, S. Hanneke, S. Moran, R. van Handel, and A. Yehudayoff, "A theory of universal learning," *CoRR*, vol. abs/2011.04483, 2020. [Online]. Available: https://arxiv.org/abs/2011.04483

[5] T. J. Viering and M. Loog, "The shape of learning curves: a review," *CoRR*, vol. abs/2103.10948, 2021. [Online]. Available: https://arxiv.org/abs/2103.10948

[6] F. Mohr, T. J. Viering, M. Loog, and J. N. van Rijn, "Lcdb 1.0: An extensive learning curves database for classification tasks," unpublished.

[7] M. Opper, W. Kinzel, J. Kleinz, and R. Nehl, "On the ability of the optimal perceptron to generalise," *Journal of Physics A: Mathematical and General*, vol. 23, no. 11, p. L581, 1990.

[8] T. L. Watkin, A. Rau, and M. Biehl, "The statistical mechanics of learning a rule," *Reviews of Modern Physics*, vol. 65, no. 2, p. 499, 1993.

[9] A. Engel and C. Van den Broeck, *Statistical mechanics of learning*. Cambridge University Press, 2001.

[10] A. Gron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 1st ed. O'Reilly Media, Inc., 2017.

[11] A. Atla, R. Tada, V. Sheng, and N. Singireddy, "Sensitivity of different machine learning algorithms to noise," *Journal of Computing Sciences in Colleges*, vol. 26, no. 5, pp. 96–103, 2011.

[12] E. Kalapanidas, N. Avouris, M. Craciun, and D. Neagu, "Machine learning algorithms: a study on noise sensitivity," in *Proc. 1st Balcan Conference in Informatics*, 2003, pp. 356–365.

[13] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *Proceedings of the Twelfth International Conference on International Conference on Machine Learning*, ser. ICML'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, p. 194â202.

[14] V. C. Stodden, "Reproducible research: Addressing the need for data and code sharing in computational science," 2010.