SRF: A Framework for the Study of Classifier Behavior under Training Set Mislabeling Noise

Katsiaryna Mirylenka[1], George Giannakopoulos[2], and Themis Palpanas[1]

[1] University of Trento
[2] Institute of Informatics and Telecommunications of NCSR Demokritos,
kmirylenka@disi.unitn.eu, ggianna@iit.demokritos.gr, themis@disi.unitn.eu

Abstract. Machine learning algorithms perform differently in settings with varying levels of training set mislabeling noise. Therefore, the choice of a good algorithm for a particular learning problem is crucial. In this paper, we introduce the "Sigmoid Rule" Framework focusing on the description of classifier behavior in noisy settings. The framework uses an existing model of the expected performance of learning algorithms as a sigmoid function of the signal-to-noise ratio in the training instances. We study the parameters of the above sigmoid function using five different classifiers, namely, Naive Bayes, kNN, SVM, a decision tree classifier, and a rule-based classifier. Our study leads to the definition of intuitive criteria based on the sigmoid parameters that can be used to compare the behavior of learning algorithms in the presence of varying levels of noise. Furthermore, we show that there exists a connection between these parameters and the characteristics of the underlying dataset, hinting at how the inherent properties of a dataset affect learning. The framework is applicable to concept drift scenaria, including modeling user behavior over time, and mining of noisy data series, as in sensor networks.

Keywords: classification, classifier evaluation, handling noise, concept drift

1 Introduction

Transforming vast amounts of collected — possibly noisy — data into useful information, through such processes as clustering and classification, is a really interesting research topic. The machine learning and data mining communities have extensively studied the behavior of classifiers — which is the focus of this work — in different settings (e.g., [13,20,8,9]), however the effect of noise on the classification task is still an interesting and open problem. The importance of studying noisy data settings is augmented by the fact that noise is very common in a variety of large scale data sources, such as sensor networks and the Web. Thus, there rises a need for a unified framework studying the behavior of learning algorithms in the presence of noise, regardless of the specifics of each algorithm.

In this work, we study the effect of training set mislabeling noise[3] on a classification task. This type of noise is common in cases of *concept drift*, where a

[3] for the rest of this paper we will use the term *noise* to refer to this type of noise, unless otherwise indicated.

target concept shifts over time, rendering previous training instances obsolete. Essentially, in the case of concept drift, feature noise causes the labels of previous training instances to be obsolete and, thus, equivalent to mislabeling noise. Drifting concepts appear in a variety of settings in the real world, such as the state of a free market or the traits of the most viewed movie. Giannakopoulos and Palpanas [10] have shown that the performance[4] of a classifier in the presence of noise can be effectively approximated by a *sigmoid* function, which relates the signal-to-noise ratio in the training set to the expected performance of the classifier. We term this approach the "Sigmoid Rule".

In our work, we examine how much added benefit we can get out of the sigmoid rule model, by studying and analyzing the parameters of the sigmoid in order to detect the influence of each parameter on the learner's behavior. Based on the most prominent parameters, we define the dimensions characterizing the algorithm behavior, which can be used to construct criteria for the comparison of different learning algorithms. We term this set of dimensions the *"Sigmoid Rule" Framework (SRF)*. We also study, using SRF, how dataset attributes (i.e., the number of classes, features and instances and the fractal dimensionality [6]) correlate to the expected performance of classifiers in varying noise settings.

In summary, we make the following contributions. We define a set of intuitive criteria based on the SRF that can be used to compare the behavior of learning algorithms in the presence of noise. This set of criteria provides both quantitative and qualitative support for learner selection in different settings. We demonstrate that there exists a connection between the SRF dimensions and the characteristics of the underlying dataset, using both a correlation study and regression modeling. In both cases we discovered statistically significant relations between SRF dimensions and dataset characteristics. Our results are based on an extensive experimental evaluation, using 10 synthetic and 14 real datasets originating from diverse domains. The heterogeneity of the dataset collection validates the general applicability of the SRF.

2 Background and Related Work

Given the variety of existing learning algorithms, researchers are often interested in obtaining the best algorithm for their particular tasks. This algorithm-selection is considered part of the meta-learning domain [11]. According to the No-Free-Lunch theorems (NFL) described in [22] and proven in [23], [21], there is no overall best classification algorithm. Nevertheless, NFL theorems, which compare the learning algorithms over diverse datasets, do not limit us when we focus on a particular dataset. As mentioned in [1], the results of NFL theorems hint at comparing different classification algorithms on the basis of dataset characteristics. Concerning the measures of performance that help distinguish among learners, in [1] the authors compared algorithms on a large number of datasets (100), using measures of performance that take into consideration the distribution of the classes within the dataset, thus using the characteristics of datasets.

[4] In this paper, by *performance* of an algorithm, we mean classification accuracy.

The Area Under the receiver operating Curve (AUC) is another measure used to assess machine learning algorithms and to divide them into groups of classifiers which have statistically significant difference in performance [2]. In all the above studies, the analysis of performance has been applied on datasets without noise, while we study the behavior of classification algorithms in noisy settings. Our present study is based on the work of G. Giannakopoulos and T. Palpanas [10] on concept drift, which illustrated that a sigmoid function can efficiently describe performance in the presence of varying levels of training set mislabeling noise. In this work, we analytically study the sigmoid function to determine a set of parameters that can be used to support learner selection in different noisy classification settings.

The behavior of machine learning classifiers in the presence of noise was also considered in [14]. The artificial datasets used for classification were created on the basis of predefined linear and nonlinear regression models, and noise was injected in the features, instead of the class labels as in our case. Noisy models of non-markovian processes using reinforcement learning algorithms and Temporal Difference methods are analyzed in [18]. In [4], the authors examine multiple-instance induction of rules for different noise models. There are also theoretical studies on regression algorithms for noisy data [19] and works on denoising, like [17], where a wavelet-based noise removal technique was shown to increase the efficiency of four considered machine learners. In both noise-related studies [19], [17] attribute noise was considered. However, we study class-related noise and do not consider specific noise models, which is a different problem. Class-related noise is mostly related to concept drift, as was also discussed in the introduction. In an early influential work, the problem of *concept attainment* in the presence of noise was indicated and studied in the STAGGER system [16]. To the best of our knowledge, there has been no work related to the selection of a classifier in a concept drift setting, based on the level of noise and other qualitative criteria, which will be reported below.

3 The Sigmoid Rule Framework

In order to describe the performance of a classifier, the "sigmoid rule" of [10] considers a function which relates signal-to-noise ratio of the training set to the expected performance. This function is called the *characteristic transfer function* (CTF) of a learning algorithm. In this work we will call it also the *sigmoid function* of an algorithm. The function is of the form

$$f(Z) = m + (M - m)\frac{1}{1 + b \cdot \exp(-c(Z - d))}$$

where $m \leq M$; $b, c > 0$; $Z = log(1+S) - log(1+N)$; S is the amount of "signal" or true data, while N is the amount of "noisy" or distorted data; hence, Z is the signal-to-noise ratio. As was shown in [10] the sigmoid function effectively approximates the performance of a classifier in noisy settings.

The behavior of different machine learning algorithms in the presence of noise can be compared on several *axes of comparison*, based on the sigmoid function parameters. Related to *performance* we can use (a) the minimal performance m; (b) the maximal performance M; (c) the width of the performance range $r_{alg} = M - m$, that defines the width of the interval in which the algorithm performance varies. Related to the *sensitivity* of performance to the change of the signal-to-noise ratio we want to know (a) within which range of noise levels there is a significant change in performance when changing the noise; (b) how we can tell apart algorithms that improve their performance even when the signal-to-noise levels are low over those which only improve in high ranges of signal-to-noise ratio; (c) how we can measure the stability of performance of an algorithm against varying noise; (d) at what noise level an algorithm reaches its average performance. To address these requirements we perform an analytic study of the sigmoid CTF of an algorithm. This analysis helps devise measurable dimensions that can answer our questions.

The domain of the sigmoid is in the general case $Z \in (-\infty, +\infty)$. The range of values is (m, M). Based on the first three derivatives, we determine the point $Z_{inf} = d + \frac{1}{c} \log b$, which is *the point of inflection* (curvature sign change point). In the case of the sigmoid function, this point is also the centre of symmetry. Furthermore, Z_{inf} indicates the shift of the sigmoid with respect to the origin of the axes. The zeros of the third order derivative are $Z^{(3)}_{1,2} = d - \frac{1}{c} \log \frac{2 \pm \sqrt{3}}{b}$, which can be used to estimate the slope of the sigmoid curve. Figure 1 illustrates the sigmoid curve and its points of interest.

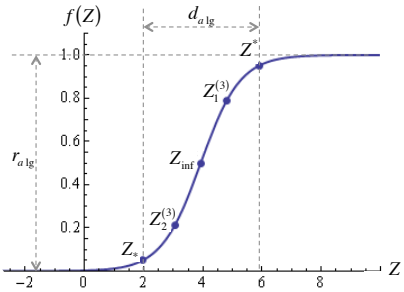

Fig. 1. Sigmoid function and points of interest.

In the following section, we formulate and discuss dimensions that describe the behavior of algorithms, based on our axes of comparison.

3.1 Sigmoid Rule Framework (SRF) Dimensions

We define several SRF dimensions based on the sigmoid properties, in addition to m, M, r_{alg} defined in Section 3. We define as *active noise range* a range $[Z_*, Z^*]$ where the change of noise induces a measurable change in the performance. To calculate $[Z_*, Z^*]$, let us assume that there is a good-enough performance for a given task, approaching M for a given algorithm. We know that $f(Z) \in (m, M)$ and we say that the performance is good enough if $f(Z) = M - (M - m) *$

$p, p = 0.05$[5]. We define the size of the signal-to-noise interval in which $f(Z) \in [m + (M - m) * p, M - (M - m) * p]$ to be the *learning improvement* of the algorithm. Then, using the inverse function $f^{-1}(y)$ we calculate the points Z^* (corr. Z_*) which is the bottom (corr. top) point in Figure 1 for a given p. We term the distance $d_{alg} = Z^* - Z_*$ as the *width of the active area* of the machine learning classifier (see Figure 1). Then, $\frac{r_{alg}}{d_{alg}}$ describes the learning performance improvement over signal-to-noise ratio change; we term this measure the *slope indicator*, as it is indicative of the slope of the CTF.

In the following paragraphs we describe how the analysis of the CTF allows to compare learning algorithm performance in the presence of noise.

3.2 Comparing Algorithms

Given the performance dimensions described above, we can compare algorithms as follows. For *performance* we can use: m, M, r_{alg}. Algorithms not affected by the presence or absence of noise will have a minimal r_{alg} value. In a setting with a randomly changing level of noise this parameter is related to the possible variance in performance. Related to the *sensitivity* of performance to the change of the signal-to-noise ratio we can use: (a) the *active noise range* $[Z_*, Z^*]$. The width of the active area of the algorithm $d_{alg} = Z^* - Z_*$, which is related to the speed of changing performance for a given r_{alg} in the domain of noise. A high d_{alg} value indicates that an algorithm varies its performance in a broad range of signal-to-noise ratios, implying less stability of performance in an environment with heavily varying degrees of noise. We say that the algorithm *operates* when the level of noise in the data is within the active noise range of the algorithm; (b) the lower bound Z_* of the active noise range, which suggests which algorithm operates earlier in noisy environment and which can reach its maximal performance fast; (c) the point of inflection Z_{inf}, that shows the signal-to-noise ratio for which an algorithm gives the average performance. Z_{inf} can be used to choose the algorithm that reaches its average performance under more noise.

A parameter related to both *performance* and *sensitivity* is the slope indicator $\frac{r_{alg}}{d_{alg}}$. It can be used to determine whether reducing the noise in a dataset is expected to have a significant impact on the performance. An algorithm with a high value of $\frac{r_{alg}}{d_{alg}}$, implies that reducing noise would be very beneficial. Furthermore, using the same dimension one can choose more stable algorithms, when the variance of noise is known. In this case, one may choose the algorithm with the lowest value of $\frac{r_{alg}}{d_{alg}}$, in order to limit the corresponding variance in performance.

Based on the above discussion, we consider the algorithms with higher maximal performance M, larger width of performance range r_{alg}, higher slope indicator $\frac{r_{alg}}{d_{alg}}$ and shorter width of the active area of the algorithm d_{alg} to behave better: we expect to get high performance from an algorithm if the level of noise in the dataset is very low, and low performance if the level of noise in the dataset

[5] The value 0.05 can be any value close to 0, describing a normalized measure of distance from optimal performance.

is very high. Decision makers can easily formulate different criteria, based on the proposed dimensions and particular settings.

4 Experimental Evaluation

In the following paragraphs, we describe the experimental setup, the datasets and the results of our experiments.

In our study, we used the following machine learning algorithms, implemented in Weka 3.6.3 [12]: (a) IBk — K-nearest neighbor classifier; (b) Naive Bayes classifier; (c) SMO — support vector classifier (cf. [15]); (d) NbTree — a decision tree with naive Bayes classifiers at the leaves; (e) JRip — a RIPPER [5] rule learner implementation. We have chosen representative algorithms from different families of classification approaches, covering very popular classification schemes [24].

We used a total of 24 datasets for our experiments.[6] Fourteen of them are real, and ten are synthetic. All the datasets were divided into groups according to the number of classes, attributes (features) and instances in the dataset as is shown on Figure 2. There are 12 possible groups that include all combinations of the parameters. Two datasets from each group were employed for the experiments.

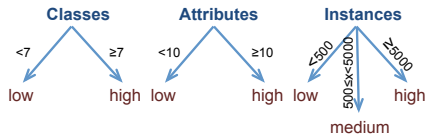

Fig. 2. Datasets grouping labels.

We created artificial datasets in the cases were real datasets with a certain set of characteristics were not available. We produced datasets with known intrinsic dimensionality. The distribution of dataset characteristics is illustrated in Figure 3. The traits of the datasets illustrated are the number of classes, the number of attributes, the number of instances and the estimated intrinsic (fractal) dimension.

The ten artificial datasets we used were built using the following procedure. Having randomly sampled the number of classes, features and instances, we sample the parameters of each feature distribution. We assume that the features follow the Gaussian distribution with mean value (μ) from the interval $[-100, 100]$ and standard deviation(σ) from the interval $[0.1, 30]$. The μ and σ intervals allow overlapping features across classes.

Noise was induced as follows. We created stratified training sets, equally sized to the stratified test sets. To induce noise, we created noisy versions of the training sets by mislabeling instances. Using different levels l_n of noise, $l_n = 0, 0.05, ..., 0.95$ [7], a training set with l_n noise is a set where there is a l_n probability

[6] Most of the real datasets come from the UCI Machine learning repository [7], and one from [10]. For a detailed list with references check the following anonymous online resource: http://tinyurl.com/3g4fmsf.

[7] We note that high levels of noise such as 95% are often observed in the presence of *concept drift*, e.g., when learning computer-user browsing habits in a network environment with a single IP, and several different users sharing it.

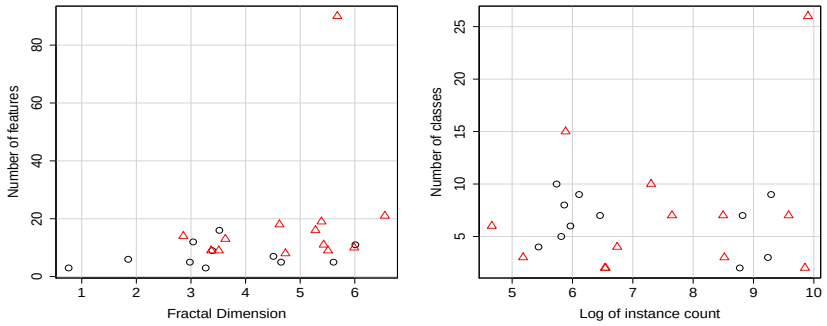

Fig. 3. Distribution of real (triangles) and artificial (circles) dataset characteristics.

that a training instance will be assigned a different label than their true one. Hence, we obtained 20 dataset versions with varying noise levels.

4.1 Using SRF

We performed experiments of "noisy" classification using the generated datasets, performing 10-fold cross validation per algorithm, and calculated the average performance for varying noise levels. Given the 20 levels of signal-to-noise ratio and the corresponding algorithm performance, (i.e., classification accuracy) we estimated the parameters of the sigmoid. The search in the parameter space is performed by a genetic algorithm, estimating an approximate good set of parameters as was proposed in [10]. The quality of estimation is checked using the Kolmogorov-Smirnov test. The results obtained are statistically significant.

A sample of true and sigmoid-estimated performance graphs for varying levels of noise can be seen in Figure 4. In our experiments, the parameters of the sigmoid were estimated offline, but SRF can be applied in an online scenario, as well, using a training period.

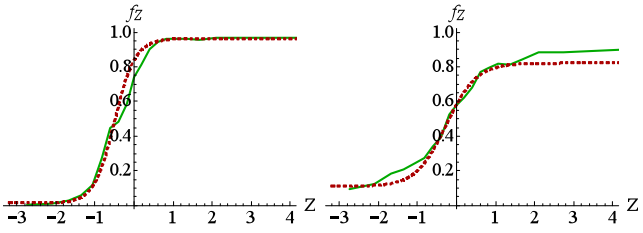

Fig. 4. Sigmoid CTF of SMO (left) and IBk (right) for "Wine" dataset. Green solid line: True Measurements, Dashed red line: estimated sigmoid.

Figure 5 illustrates the means of the SRF parameters per algorithm, over all 24 datasets. As an example of interpretation of the figure using SRF, the plots indicate that (for the studied range of datasets) SMO is expected to improve its performance faster than all other algorithms, when the signal-to-noise ratio increases. This conclusion is based on the slope indicator ($\frac{r_{alg}}{d_{alg}}$) values. Also, IBk has a smaller potential for improvement of performance (but also smaller potential for loss) than SMO when noise levels change, given that the width of the performance range r_{alg} is higher for SMO. This difference can also be seen in Figure 4, where the distance between minimum and maximum performance values is bigger for the SMO case (see Figure 4(left)).

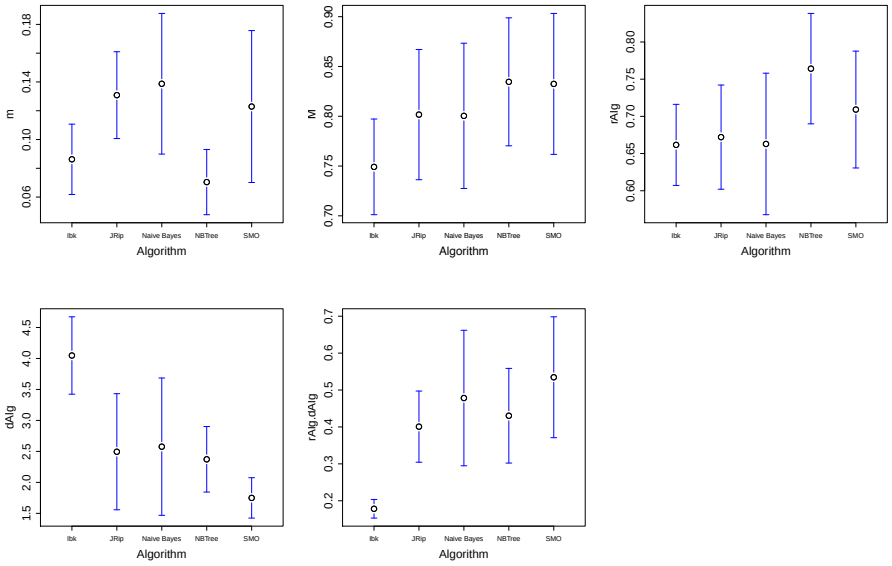

Fig. 5. SRF parameters per algorithm. X axis labels (left-to-right): IBk, JRip, NB, NBTree, SMO.

We stress that parameter estimation does not require previous knowledge of the noise levels, but it is dataset dependent. In the special case of a classifier selection process, having an estimate of the noise level in the dataset helps to reach a decision through the use of SRF.

4.2 Statistical Analysis

We now study the connection between the dataset characteristics and the sigmoid parameters (using the same 24 datasets), irrespective of the choice of the algorithm. We consider the results obtained from all the algorithms as different samples of SRF parameters for a particular dataset. We use regression analysis to observe the cumulative effect of the dataset characteristics on a single parameter, and we use correlation analysis to detect any connection between

each (dataset characteristic, sigmoid parameter) pair. We examine the connections between dataset characteristics and the sigmoid parameters both individually, and all together, in order to draw the complete picture.

Regression Analysis We wanted to examine how the number of classes (x_1), number of features (x_2), number of instances (x_3), and intrinsic dimensionality[8] (as fractal correlation dimension [3]) (x_4) of a dataset influence the CTF parameters.

We applied a leave-one-out process, where one dataset is left out from training and used for testing on every run. We used in turn m, M, r_{alg}, d_{alg}, and $\frac{r_{alg}}{d_{alg}}$ as dependent variables. The results of model fitting and prediction of SRF dimensions are reported in Table 1, where average errors between observed and predicted SRF dimensions are shown. For each SRF dimensions chosen, we have observed 5 values (since 5 machine learning algorithms were used), and having estimated them for 24 datasets, we end up with 120 predictions for a single SRF dimension. We calculated four types of errors: (1) MSE — mean square error; (2) MAE — mean absolute error; (3) RMSE — relative mean square error and (4) RMAE — relative mean absolute error. The last column of Table 1 shows the average of the adjusted R^2 statistic for models that where estimated for all the SRF dimensions (average on the 24 datasets). Figure 6 illustrates how our models fit the test data, showing that in most cases the true values of the sigmoid parameters for each dataset (illustrated by circles that correspond to 5 algorithms for each test dataset i, $i = 1, 2, ..., 24$) are within the 95% confidence level zone around the estimated values. This finding further supports the connection between the dataset parameters and SRF dimensions. According to the results, the chosen parameters of the datasets can be used to predict the parameters of the sigmoid of the algorithms.

Parameters	**Error measures**				**average**(R_a^2)
	MSE	MAE	RMSE	RMAE	
m	0.11	0.09	148.92	29.17	0.54
M	0.35	0.30	0.51	0.41	0.88
r_{alg}	0.32	0.27	0.71	0.46	0.85
d_{alg}	1.98	1.41	0.97	0.68	0.67
$\frac{r_{alg}}{d_{alg}}$	0.37	0.27	4.83	1.46	0.55

Table 1. Prediction error of linear regression models.

Correlation analysis We used three different correlation coefficients — Pearson correlation for linear correlation, Spearman's rho and Kendall's tau for monotonic correlation — to analyze the connection between the parameters of the datasets and the CTF parameters (cf. Table 2). We qualitatively interpret the strength of the correlation as follows: $[0.0; 0.1)$ →No Correlation, $[0.1; 0.3)$ →Low Correlation, $[0.3; 0.5)$ →Medium Correlation, $[0.5; 1]$ →Strong Correlation.

[8] The authors would like to thank Christos Faloutsos for kindly providing the code for the fractal dimensionality estimation.

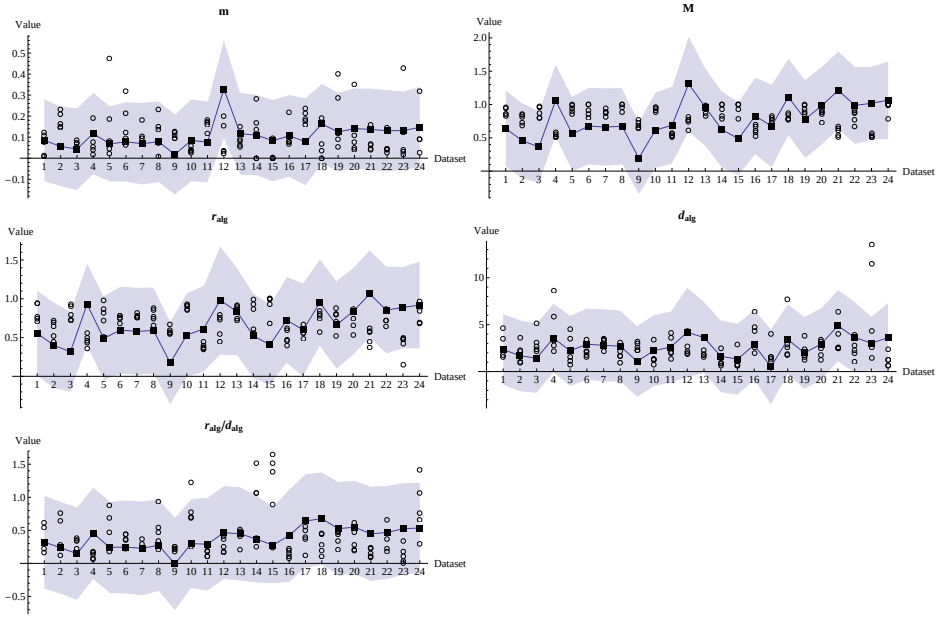

Fig. 6. Real and estimated values of the sigmoid parameters. Real values: Black rectangles, Estimated values: circles, Gray zone: 95% prediction conf. interval.

		Pearson's corr.				Spearman's rank corr.					Kendall's τ rank corr.				
Parameter	m	M	r_{alg}	d_{alg}	$\frac{r_{alg}}{d_{alg}}$	m	M	r_{alg}	d_{alg}	$\frac{r_{alg}}{d_{alg}}$	m	M	r_{alg}	d_{alg}	$\frac{r_{alg}}{d_{alg}}$
x1	-0.03	**-0.26**	**-0.21**	0.13	**-0.29**	0.02	**-0.26**	**-0.25**	**0.31**	**-0.34**	0.01	**-0.17**	**-0.18**	**0.22**	**-0.24**
x2	-0.07	**-0.31**	- **0.23**	0.13	**-0.21**	0.03	**-0.26**	**-0.24**	0.14	**-0.20**	0.02	**-0.18**	**-0.16**	0.10	**-0.14**
x3	0.14	0.08	-0.01	-0.12	0.12	-0.05	0.03	0.02	**-0.21**	***0.18***	-0.05	0.01	0.01	**-0.14**	***0.12***
x4	0.04	***-0.16***	***-0.16***	0.13	-0.09	-0.03	**-0.20**	***-0.18***	0.06	-0.11	-0.03	- ***0.12***	***-0.11***	0.04	-0.07

Table 2. Correlation between dataset parameters and SRF parameters. Colored cells: statistically significant correlation ($p - value < 0.05$: **underlined bold**, $p - value < 0.1$: ***italics-bold***). Green (dark) : medium correlation, gray (light) : low correlation.

Summarizing the results from all the correlation coefficients (refer to Table 2), some interesting conclusions can be drawn. First, the number of classes (x_1) is inversely correlated to $\frac{r_{alg}}{d_{alg}}$, r_{alg} and M. Thus, the higher the number of classes is, the lower the sensitivity to noise variation (check on $\frac{r_{alg}}{d_{alg}}$); the lower the number of classes, the higher the impact of reducing noise on performance (check r_{alg} and M). These conclusions are also supported by the direct correlation between the number of classes and the width of the active area of the algorithm d_{alg}. We also note the complete lack of significant correlation between the minimum performance m and all of the SRF dimensions: given enough noise an algorithm always performs badly. Thus, the number of classes significantly influences the behavior of an algorithm, regardless of the family of the algorithm. Second, the number of features (x_2) provides a minor reduction of sensitivity

to noise variation (resulting from low correlation to d_{alg}). This conclusion is also supported by the negative influence on $\frac{r_{alg}}{d_{alg}}$, r_{alg}. We also note that the number of features affects the maximal performance M, which shows (rather contrary to intuition) that more features may negatively affect performance in a noise-free scenario. This is most probably related to features that are not essentially related to the labeling process, thus inducing feature noise. Third, there is a correlation between the number of instances (x_3) and $\frac{r_{alg}}{d_{alg}}$. This shows that larger datasets (providing more instances) reduce sensitivity to noise variation. Last, fractal dimensionality (x_4) of a dataset has low, but statistically significant negative influence on M and on r_{alg}. Fractal dimensionality is indicative of the "complexity" of the dataset. Thus, if the dataset is complex (high x_4) machine learning is difficult even at low noise levels. We note that low r_{alg} may be preferable in cases where the algorithm should be stable even for low signal-to-noise ratios.

The correlation analysis demonstrates the connection between dataset characteristics and SRF dimensions. Consequently, the SRF can be used to reveal a-priori the properties of an algorithm with respect to a dataset of certain characteristics. This allows an expert to select a good algorithm for a given setting, based on the requirements of that settings. Such requirements may, e.g., relate to the stability of an algorithm in varying levels of noise and the expected maximum performance in non-noisy datasets.

5 Conclusions

Machine learning algorithms are often used in noisy environments. Therefore, it is important to know a-priori the properties of an algorithm with respect to a dataset of certain characteristics. In this work, we investigate whether some simple dataset properties (namely, number of classes, number of features, number of instances and fractal dimensionality) can help in the above direction.

We propose the "Sigmoid Rule" Framework, which describes a set of dimensions that may be used by a decision maker to choose a good classifier, or to estimate SRF dimensions, based on a range of dataset characteristics. Our approach is applicable to user modeling tasks, when the user changes behavior over time, and to any concept drift problems for data series mining. We showed that the parameters related to the behavior of learners correlate with dataset characteristics, and the range of their variation may be predicted using regression models. Therefore, SRF is a useful meta-learning framework, applicable to a wide range of settings that include noise. However, using these SRF models for parameter prediction does not provide enough precision to be used for performance estimation.

As part of our ongoing work, we examine whether the "Sigmoid Rule" also stands in the case of sequential classification. Preliminary experimental results on the "Climate" UCI dataset (taking into account its temporal aspect) indicate that, indeed, the "Sigmoid Rule" and therefore SRF are directly applicable, and can be used as a means to represent the behavior of an HMM-based classifier

in the presence of noise. This finding may open the way to a broader use of the SRF, including sequential learners.

6 Acknowledgement

This research was partially supported by FP7 EU IP project KAP (grant agreement no. 260111).

References

1. S. Ali and K. Smith. On learning algorithm selection for classification. *Applied Soft Computing*, 6(2):119–138, 2006.
2. A. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.
3. F. Camastra and A. Vinciarelli. Estimating the intrinsic dimension of data with a fractal-based method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(10):14041407, 2002.
4. Y. Chevaleyre and J.-D. Zucker. Noise-tolerant rule induction from multi-instance data. In *In Proceedings of the ICML-2000 workshop on Attribute-Value and Relational Learning: Crossing the Boundaries, L. De Raedt*, 2000.
5. W. W. Cohen. Fast effective rule induction. In *ICML*, 1995.
6. E. de Sousa, A. Traina, C. Traina Jr, and C. Faloutsos. Evaluating the intrinsic dimension of evolving data streams. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 643–648. ACM, 2006.
7. A. Frank and A. Asuncion. UCI machine learning repository, 2010.
8. G. Giannakopoulos and T. Palpanas. Adaptivity in entity subscription services. In *ADAPTIVE*, 2009.
9. G. Giannakopoulos and T. Palpanas. Content and type as orthogonal modeling features: a study on user interest awareness in entity subscription services. *International Journal of Advances on Networks and Services*, 3(2), 2010.
10. G. Giannakopoulos and T. Palpanas. The effect of history on modeling systems' performance: The problem of the demanding lord. In *ICDM*, 2010.
11. C. Giraud-Carrier, R. Vilalta, and P. Brazdil. Introduction to the special issue on meta-learning. *Machine Learning*, 54(3):187–193, 2004.
12. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
13. J. Han and M. Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann, 2006.
14. E. Kalapanidas, N. Avouris, M. Craciun, and D. Neagu. Machine learning algorithms: a study on noise sensitivity. In *Proc. 1st Balcan Conference in Informatics*, pages 356–365, 2003.
15. S. Keerthi, S. Shevade, C. Bhattacharyya, and K. Murthy. Improvements to platt's smo algorithm for svm classifier design. *Neural Computation*, 13(3):637–649, 2001.
16. A. Kuh, T. Petsche, and R. L. Rivest. Learning time-varying concepts. In *NIPS*, pages 183–189, 1990.
17. Q. Li, T. Li, S. Zhu, and C. Kambhamettu. Improving medical/biological data classification performance by wavelet preprocessing. *Proceedings ICDM Conference*, 2002.
18. M. Pendrith and C. Sammut. On reinforcement learning of control actions in noisy and non-markovian domains. Technical report, School of Computer Science and Engineering, the University of New South Wales, Sydney, Australia, 1994.
19. O. Teytaud. Learning with noise. Extension to regression. In *Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on*, volume 3, pages 1787–1792. IEEE, 2002.
20. S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 2003.
21. D. Wolpert. The existence of a priori distinctions between learning algorithms. *Neural Computation*, 8:1391–1421, 1996.
22. D. Wolpert. The supervised learning no-free-lunch theorems. In *Proc. 6th Online World Conference on Soft Computing in Industrial Applications*. Citeseer, 2001.
23. D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8:1341–1390, October 1996.
24. X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. F. M. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowl. Inf. Syst.*, 14(1):1–37, 2008.