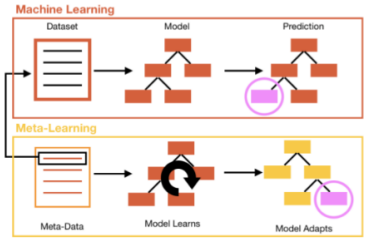
Poster

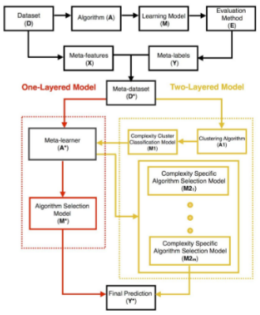
Approximating Dataset Complexity Based on Learning Curves

By Japnit Kaur Ahuja

Introduction



- Machine Learning produces **dynamic solutions.** However, most machine learning is still done **manually**.
- In this research, we provide an **Automated Machine Learning (AutoML)** framework for the **Algorithm Selection Problem.**



One-layered Framework

- Most Meta-learners available today follow this prototypical **one-layered framework**.
- Built over the **whole meta-dataset (D*).**
- Only vary in terms of the **Meta-features (X), Meta-labels (Y)** and/or **Meta-learner (A*).**

Two-layered Framework

- We hypothesise that the **complexity of a dataset (D)** is key in determining the performance of a corresponding model(**M**).
- The complexity based clusters will give more **specific algorithm selection models (M2i)** for each cluster.

Overview of Obtaining Complexity-based Clusters

Generation of complexity based Artificial Datasets

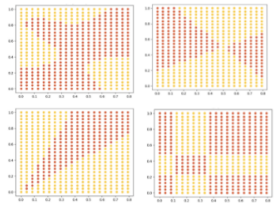
↓

Cluster based on Learning Curves

↓

Meta-features for cluster membership

Artificial Dataset Generation



Group A:

- Based on weighted **all neighbours kNN**.
- **m** random pivot points.

Group B and C:

- Common intersection in **corner** or **centre**.
- **m** linear separators.

Group D:

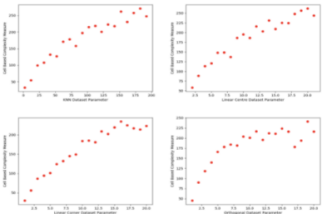
- Separators are **orthogonal** to the axes.
- **m** linear separators.

We generated 200 **binary** datasets (only 2 classes) of each group. These datasets include 25 x 25 **evenly distributed** points in the 2D Euclidean Space. The complexity for all of these datasets vary based on the parameter **m.**

Cell-based Measure

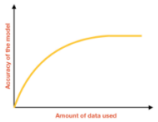
- We introduce a new measure to calculate the complexity of the dataset based on the **size of the decision boundary.**
- After dividing the dataset into a grid, count the number of **shared edges between two opposite classes.**

Cell-based vs m

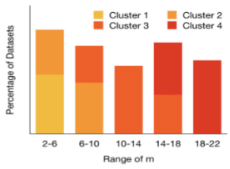


Thus, **m** is correlated with the cell-based measure and is **indicative of the complexity** of the dataset.

Clustering with Learning Curves



- Learning Curves allow us to measure complexity of a dataset **relative to an algorithm.**
- Datasets are **clustered** based on 20 points of the learning curves using **kmeans.**



We observed that each complexity based cluster corresponds to a certain range of **m.**

Meta-feature Selection

As the generation of learning curves is **computationally expensive**, meta-features are chosen for determining cluster membership.

Algorithm Description

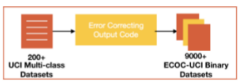
- Start with no meta-features.
- In each iteration, choose a new meta-feature to be checked.
- If the new meta-feature improves the **mutual information** between the meta-features and cluster labels then add it to the set or else stop.

Metafeature Set	Mutual Info.
Joint Entropy, Mutual Info Max, N1, N2, T1, LSCAvg, L2, L3, Density, ClsCoef, Hubs	0.637

Now, we can cluster through **kmeans** using these **11 meta-features.**

Final Experimental Setup

Problem Space



Algorithm Space

- Decision Tree (Dt)
- Random Forest (Rf)
- k-Nearest Neighbours (kNN)
- Naive Bayes (Nb)
- Linear Discriminant Analysis (Lda)
- Quadratic Discriminant Analysis (Qda)

Algorithm Selection Problem

- **15 algorithm selection problems** obtained by pairing all the algorithms.
- **For each problem, there are 3 possible Meta-labels:**
  - algorithm 1 is superior
  - algorithm 2 is superior
  - draw
- Computed through **two-tailed t-test.**

Two-layered model vs One-layered model Analysis

- In an overarching **stratified 10x10 Cross Validation evaluation**, for each fold, the **best cluster-specific meta-learner** (over kNN models of different k and meta-features set) is chosen.
- The proposed two layered model is then compared to the best model of the prototypical one-layered model.

Result and Analysis

Algorithm Pair		Baseline	Cluster
Dt	Knn	81.625	82.862
Dt	Lda	82.76	84.004
Dt	Nb	88.137	88.909
Dt	Qda	89.487	90.225
Dt	Rf	88.958	88.95
Knn	Lda	84.656	85.62
Knn	Nb	91.776	92.161
Knn	Qda	91.122	91.307
Knn	Rf	81.269	82.58
Lda	Nb	85.471	85.886
Lda	Qda	84.108	84.425
Lda	Rf	90.672	90.928
Nb	Qda	84.508	84.963
Nb	Rf	96.036	95.965
Qda	Rf	97.682	97.536

Final Experimentation Results

- Accuracy of the two-layered model is either **comparable or an improvement** to the baseline one-layered model.
- Our proposed two-layered model is a **practical substitute** to the prototypical method.

Conclusion and Future Work

Algorithm Pair		Baseline Accuracy	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Dt	Knn	81.625	89.356	86.928	79.245	73.705
Dt	Lda	82.76	94.117	81.528	80.341	76.518
Dt	Nb	88.137	92.178	91.111	86.387	77.894
Dt	Qda	89.487	96.218	91.273	87.037	82.608
Dt	Rf	88.958	97.297	95.205	88.113	81.111
Knn	Lda	84.656	92.118	90.74	88.172	77.865
Knn	Nb	91.776	94.805	94.267	93.827	89.256
Knn	Qda	91.122	96.398	95.041	91.304	83.516
Knn	Rf	81.269	93.627	85.84	71.428	67.283
Lda	Nb	85.471	95.731	92.561	83.471	81.912
Lda	Qda	84.108	95.031	92.436	84.955	74.742
Lda	Rf	90.672	96.666	94.805	91.411	82.52
Nb	Qda	84.508	88.378	86.784	83.798	67.741
Nb	Rf	96.036	98.138	97.457	96.774	91.279
Qda	Rf	97.682	99.163	97.686	97.546	95.121

Conclusion

- **Two or more** clusters performed better than the baseline for all problems.
- We may infer that these clusters contain key representative datasets, which have **distinct complexities.**

Future Work

- Acquire insights about the **cluster composition**.
- Derive a model to **optimally cluster** datasets

References

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, *1*(1), 67-82.

Smith-Miles, K. A. (2009). Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys (CSUR)*, *41*(1), 6.

Guyon, I., Chaabane, I., Escalante, H. J., Escalera, S., Jajetic, D., Lloyd, J. R., ... & Statnikov, A. (2016). A brief review of the ChaLearn AutoML challenge: any-time any-dataset learning without human intervention. In *Workshop on Automatic Machine Learning* (pp. 21-30).

Lorena, A. C., Garcia, L. P., Lehmann, J., Souto, M. C., & Ho, T. K. (2018). How Complex is your classification problem? A survey on measuring classification complexity. *arXiv preprint arXiv:1808.03591*.

Vilalta, R., & Drissi, Y. (2002). A perspective view and survey of meta-learning. *Artificial intelligence review*, *18*(2), 77-95.

Perlich, C. (2011). Learning curves in machine learning. In *Encyclopedia of machine learning* (pp. 577-580). Springer, Boston, MA.

All Images and graphs were self-drawn.

Abstract

To maximise the accuracy of a learning model for a problem there is a need to select an appropriate algorithm. Thus, the ability to predict the performance of an algorithm is imperative in the algorithm selection problem. This paper proposes a novel two-layered approach in which the complexity of a dataset plays a key role. In the first layer a category of complexity is assigned to the dataset and then in the second layer the superior algorithm is determined. In order to do this, we attempt to define the complexity of the dataset by introducing a new cell-based complexity measure. Then, we evaluate the applicability of that measure by comparing it to widely used complexity measures found in the literature review. Using our definition of complexity, we then validate the idea of using learning curves to capture the complexity of a dataset and generate complexity specific categories. Finally, we evaluate the whole model on UCI datasets to prove that our proposed two-layered model is a practical substitute to the meta-learners available today.

Research Plan

(a) Rationale

Machine learning is the study of models in which the computer uses training data to get better at a problem. Meta-learning, a widely researched branch of machine learning is concerned with learning about the learning process itself. The No Free Lunch Theorems necessitate the need of algorithm selection and given the number of algorithms available it can be a time-consuming task. Moreover, there is a growing need for methods to works as "black boxes", where no human intervention is required like WEKA in Java or scikit in Python. Thus, a lot of resources have been devoted to finding an effective way to select the best performing algorithm. The currently available one-layered metalearners predict the performance of an algorithm based on a performance metric. In this work, we propose a novel approach and suggest a two-layered framework for the algorithm selection meta-learner.

(b) Research Question

We hypothesise that the complexity of the dataset is a key aspect in determining the performance of an algorithm, thus approximating complexity before the prediction of the algorithm will help produce an accurate model for algorithm selection. In our proposed two layered framework: In the first layer the category of complexity will be determined for the dataset and then in the second layer the algorithm will be predicted. To approximate the complexity of the dataset, we use learning curves. Thus, we aim to check if the categorisation of datasets based on complexity using learning curves gives an accurate algorithm selection model.

(c) Procedure

To test the hypothesis, we generate 800 artificial datasets of varying complexity. We validate their complexity by defining a cell-based complexity measure. After which, based on the learning curve points of the datasets we generate clusters to investigate if they are indicative of the complexity of the dataset. Once validated, we create the two-layered framework which first determines the complexity category a dataset belongs to and then predicts the superior

algorithm. This model is then tested against a baseline one-layered meta-learner on real datasets from the UCI repository.

(d) Bibliography

1. Wolpert, D., & Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation,1*(1), 67-82. doi:10.1109/4235.585893

2. Guyon, I., Chaabane, I., Escalante I, H. J., Escalera, S., Jajetic, D., Lloyd, J. R., . . . Viegas, E. (n.d.). A brief Review of the ChaLearn AutoML Challenge. Retrieved from http://proceedings.mlr.press/v64/guyon_review_2016.pdf

3. Dietterich, T G., Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation, , (1998) 10(7):1895-1924.

4. Soares, C., Zoomed Ranking: Selection of Classification Algorithms based on Relevant Performance Information. Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, 126-135. Springer. . (2000b)

5. Lagoudakis, M.G. and Littman, M. L., Algorithm selection using reinforcement learning. In Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000), 511-518, Stanford, CA. (2000)

6. Kalousis, A. and Hilario, M.: Feature Selection for Meta-Learning. In Proceedings of the 5th Pacific Asia Conference on Knowledge Discovery and Data Mining. Springer. (2001)

7. Munoz, M. A., L. V., Miles, K. S., & Baatar, D. (n.d.). (PDF) Instance Spaces for Machine Learning Classification. Retrieved from https://www.researchgate.net/publication/315835025_Instance_Spaces_for_Machine_Learning_Classification

8. Lorena, A., Garcia, L. P., Souto, M. C., Lehmann, J., & Ho, T. K. (2018, August 10). How Complex is your classification problem? A survey on measuring classification complexity. Retrieved from https://arxiv.org/abs/1808.03591

9. Ho, T. K., & Basu, M. (n.d.). Retrieved from https://ieeexplore.ieee.org/document/990132

10. Perlich, C. (2009). *Learning Curves in Machine Learning*. [online] Available at: https://www.researchgate.net/publication/247934703_Learning_Curves_in_Machine_Learning [Accessed 7 Jan. 2019].

Project Report

1. Introduction

According to the No Free Lunch Theorems, no one algorithm can give the best performance over every classification problem[1], which necessitates algorithm selection. Since the seminal work by Rice (1976), a lot of research has been done in the field of metalearning, especially algorithm selection. Given the plethora of new algorithms, the task of choosing an algorithm has become complicated and time-consuming. Thus, there is a growing need for fully automated models that do not require human intervention, shifting the focus to Automatic Machine Learning (AutoML).[2] Although the machine learning literature has proposed many algorithm selection techniques[3][4], most of those are single layered frameworks which directly predict the algorithm based on certain meta-features. This paper proposes a novel approach, a two-layered mechanism: In the first layer, the complexity of the dataset is determined; then, in the second layer, a superior algorithm is predicted.

In our first experiment, we define the complexity of a dataset by introducing our own cell-based complexity measure. Next, we will check the applicability of this measure by deriving a relationship with other complexity measures found in literature review. Then, using our definition of complexity we will validate the learning curves based complexity categories on artificially generated datasets. Once validated, we will empirically prove the practicality of our two layered framework by testing it on real datasets from the UCI repository.

2. Algorithm Selection: Related Work and a New Perspective

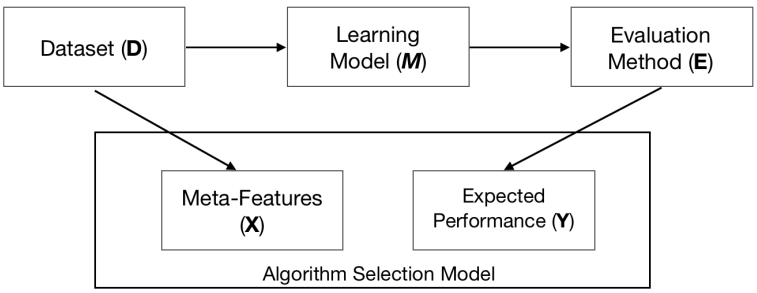


Figure 1. Algorithm selection framework

Adapted from Rice (1976), Fig 1 depicts the algorithm selection framework in which there is a problem space (**P**) which contains datasets $d_i \in D$; a performance measure (**Y**) of d_i is

calculated for a learning model which is then mapped to relevant dataset characteristics, or meta-features (**X**) by a meta-learner. Various kinds of meta-learners can be found in literature[5][6], and typically, all these meta-learners are one layered i.e. these predict the best algorithm based on a performance metric (accuracy, execution time, etc). The meta-learner's predictive model utilises meta-features, which are quantities derived from 6 main categories: simple (e.g. number of attributes or classes), statistical (e.g. skewness, kurtosis of any continuous attributes), information-theoretic (e.g. class entropy, mutual information between attributes and class - for discrete attributes), landmarking (e.g. Naive Bayes), model-based (e.g. height of the decision tree), and complexity (e.g.approximating the decision boundary).[7]

We hypothesise that the complexity of a dataset is key in determining the accuracy of a model for an algorithm. Moreover, both share an inverse relationship for a specific algorithm: as the complexity of the dataset increases, the accuracy of a model decreases. A recent work[8] surveys all the complexity meta-features based on different structures like clusters and graphs but still only vaguely define the type of complexities being measured. Building on the factors of the complexity of classification problems given by Ho and Basu (2002)[9], we define specific categories of complexity:

1. Complexity of Decision Boundary: A boundary estimates the difficulty in separating the classes and thus assigning a class to the new data points
2. Imbalance: Representation of each class in the dataset
3. Spatial Distribution of the data
    a. Class-wise coverage: measures the distribution of the data within each class
    b. Overall coverage of the space: measures the distribution of the whole dataset e.g. sparse datasets are more complex

For the purposes of this paper, we define data complexity based on the model of the algorithm itself, as it will be more practical for its application in the algorithm selection problem. To obtain this we use the points on the learning curve which will capture the key aspects of the complexity of the dataset relative to an algorithm.

3. Clustering Datasets Based on Complexity

We want to obtain clusters of datasets which are indicative of their member dataset's complexity. Complexity is an important determinant in the performance of an algorithm, thus complexity specific clusters will allow datasets which have similar algorithm performance to

be in the same group. To verify the existence of such categories of complexity based on learning curves we experiment on 800 artificially generated binary datasets. These datasets include 25 x 25 data points evenly distributed in the 2D euclidean space. For each of the following groups, 200 datasets of each type were generated.

- Group A: The instances are categorised basedon weighted all neighbours kNN.The model is trained on m randomly generated pivot instances which are classified beforehand.
- Group B and Group C: m linear separators with a common intersection at the centre or the corner are used to classify the instances.
- Group D: m linear separators which are orthogonal to the axes are used to classify the instances

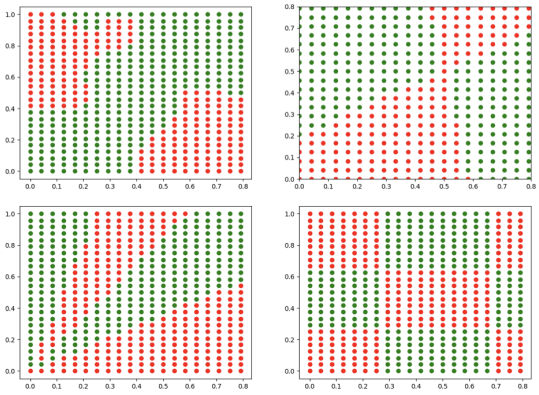


Figure 2. Group A, B, C and D of the artificial datasets

All these datasets vary in complexity based on a certain parameter m, for example, the complexity increases as the number of separators increases. To verify this relation between the parameter m and the complexity of the dataset we define dataset complexity in terms of a decision boundary. Intuitively speaking, greater size of this boundary would mean that the dataset is more complex. We calculate this boundary by dividing the instance space into a grid, where each cell contains only one instance, and counting the number of edges which are shared between cell of opposite classes. The cell-based complexity was found to be correlated with the following complexity measures associated with measuring the complexity of decision boundary[8] (Appendix A).

Metafeature	Relation	Metafeature	Relation
Fraction of Borderline Points	Direct	Fraction of Hyperspeheres	Direct
Ratio of Intra/Extra Class Nearest Neighbour Distance	Direct	Local set average cardinality	Inverse
Error rate fo the nearest neighbour classifier	Direct	Average Density of Network	Inverse
Non linearity of NN Classifier	Direct		

Table 1. Relation between cell-based complexity measures and other complexity measures

The parameter m was found to be indicative of the complexity of the dataset as it was correlated with the cell based complexity measure.

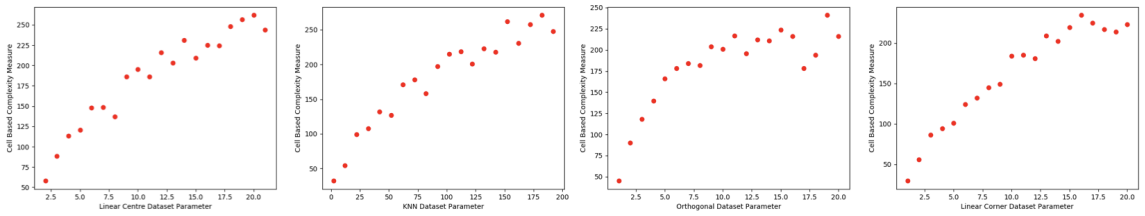


Figure 3. Cell based complexity measure vs m

To obtain the complexity specific categories, learning curves (LC) [10] (plots the rate at which the model learns - accuracy vs sample size) are generated for all artificial datasets. To lower the computational cost of the generation only 20 points are plotted. To investigate whether categories based on learning curves capture the complexity of a dataset, artificial datasets are clustered based on the 20 points of LC for Decision Tree, Random Forest and k-Nearest Neighbour using k-means from the scikit python library. The parameter k for the algorithm is selected so as to minimise the mean squared error of the cluster points to their centroid. After observing the prevalence of certain ranges of $m$for each cluster (lower values of $m$were seen for less complex clusters), it is established that these clusters are indicative of the complexity of the dataset, and thus clustering based on learning curves is a viable method to get complexity-specific categories.

	Decision Tree					Random Forest					KNN						Decision Tree					Random Forest					KNN				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Linear Corner Seperators																Linear Centre Seperators															
2-6	0	65	32.5	2.5	0	0.0	72.5	25.0	2.5	0.0	72.5	27	0.0	0.0	0.0	1-5	0.0	42.5	55.0	2.5	0.0	0.0	42.5	52.5	5.0	0.0	42.5	55	2.5	0.0	0.0
6-10	0	0	82.5	17.5	0	0.0	5.0	85.0	10.0	0.0	15.0	72.5	12.5	0.0	0.0	5-9	0.0	2.5	40.0	52.5	5.0	0.0	5.0	40.0	50.0	5.0	12.5	32.5	47.5	0	0
10-14	0	0	25	60	15	0.0	0.0	40.0	50.0	10.0	0	27	55	12.5	0.0	9-13	0.0	0.0	7.5	72.5	20.0	0.0	0.0	10.0	70.0	20.0	0.0	7.5	60.0	25.0	7.5
14-18	0	0	10	42.5	47.5	0.0	0.0	17.5	47.5	35.0	0	12.5	37.5	37.5	12.5	13-17	0.0	0.0	5.0	47.5	47.5	0.0	2.5	2.5	52.5	42.5	0	5.0	45.0	25.0	22.5
18-22	0	0	7.5	40	52.5	0.0	0.0	15.0	47.5	37.5	0	0	35.0	45.0	7.5	17-21	0.0	0.0	5.0	12.5	82.5	0.0	0.0	5.0	22.5	72.5	0.0	5.0	10.0	50.0	35.0
Orthogonal Seperators																KNN pivot instances															
1-5	25.0	75.0	0.0	0.0	0	32.5	65.0	2.5	0.0	0.0	25.0	40.0	30.0	0	0.0	2-42	0	45	52.5	2.5	0	0	45	50	0	0	42.5	50	0	0	0
5-9	87.5	12.5	0.0	0.0	0	72.5	2.5	5.0	15.0	5.0	0.0	7.5	32.5	52.5	7.5	42-82	0	0	37.5	62.5	0	0	0	32.5	57.5	0	0	27.5	65	0	0
9-13	85.0	10.0	0.0	5.0	0	50.0	0.0	2.5	15.0	32.5	0.0	0.0	20.0	55	25.0	82-122	0	0	0	80	20	0	0	0	77.5	22.5	0	0	32.5	67.5	0
13-17	82.5	12.5	0.0	5.0	0	32.5	2.5	2.5	17.5	45.0	0.0	2.5	17.5	30.0	50.0	122-162	0	0	0	32.5	67.5	0	0	0	42.5	57.5	0	0	10	75	15
17-21	82.5	10.0	0.0	7.5	0	42.5	0.0	5.0	30.0	22.5	0.0	0.0	30.0	37.5	32.5	162-202	0	0	0	5	95	0	0	0	5	95	0	0	0	47.5	52.5

Table 2. Clusters and their corresponding values of m

After literature review, we chose 6 widely used learning algorithms to diversify our algorithm

portfolio. The algorithm space consists of Decision Tree (DT), Random Forest (RF), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Naive Bayes (NB), and k-Nearest Neighbours (kNN). Default parameters were used for all the following R packages: e1071,MASS, Rweka and kknn[7].

The clusters based on learning curves were generated for all algorithms. These clusters represented the algorithm-specific complexity categories of the dataset, for example, the lowest complexity cluster in decision tree only had orthogonal datasets since decision trees work best on those, but so was not the case for the Random Forest clusters. Thus, to eliminate this algorithm bias, we used the cluster membership labels of each algorithm as meta-features of the dataset and clustered again using k-means, thus obtaining a general set of 7 clusters. To validate that the information of the key aspects of complexity captured in the algorithm specific cluster labels is not lost in the second clustering process we calculated the mutual information between each algorithm specific cluster label and the final label of the general set. The high mutual information score shows that the general set was suggestive of the complexity of its member datasets.

Algorithm	Mutual Info.	Algorithm	Mutual Info.
KNN	0.606	LDA	0.678
RF	0.601	DT	0.623
NB	0.613	QDA	0.561

Table 3. Mutual Information results for each algorithm specific cluster label

Determining the complexity based cluster that the dataset belongs to might be computationally expensive through the generation of learning curves, thus we can find the meta-features which are correlated to the membership of the dataset to the cluster. To conduct experiments, we chose 55 meta-features (Appendix B) from literature review which included statistical meta-features and information theoretic meta-features (19), decision tree based meta-features (14) and complexity meta-features (22).

For the meta-feature selection, a greedy approach was used rather than a brute force variant as its computationally expensive to go through 2^{55} possible sets. The datasets are first clustered using k-means (k=7) for each meta-feature. The mutual information between these cluster labels and the general set labels is calculated. Then, the meta-feature with the highest mutual information is selected. This meta-feature is paired with all the other meta-features

and the highest one is selected again. This process is repeated till a meta-feature set with the highest mutual information is obtained.

Metafeature Set	Mutual Info.
Joint Entropy, Mutual Info Max, N1, N2, T1, LSCAvg, L2, L3, Density, ClsCoef, Hubs	0.637

Table 4. set of meta-features and their corresponding mutual information score

Now, these 11 chosen meta-features (9 complexity measures and 2 statistical and information theoretics) can be used to allocate a cluster to the dataset in the first layer, after which in the second layer the superior algorithm can be predicted.

4. Algorithm Selection via Complexity Classification

To study the utility of the double-layered framework, we conducted an experiment on UCI datasets. To expand our problem space, we utilised the Error-Correcting Output Code ensemble[11] to convert multiclass datasets into many binary datasets, thus obtaining 9139 UCI datasets to experiment on. For the expertise space, there were three possible outcomes: algorithm 1 being superior, algorithm 2 being superior or a case of draw. These three outcomes were used as labels and were computed for all pairs of the 6 algorithms (15 expertise spaces) through a two-tailed t-test by random sampling 100 times for all UCI datasets. Then, the 11 chosen meta-features were computed to classify the UCI datasets into their respective complexity cluster through k-means. An overarching 10 x 10 fold cross validation model was built, and in each iteration the folds were divided into training (9 folds) and testing (1 fold) sets. Then, each training set was further divided into a new training and validation set to select the best performing algorithm selection model by choosing a value of k for kNN and a meta-feature set out of the following 3 options:

1. Set A: Classical (statistical and information theoretic) meta-features
2. Set B: Decision tree based meta-features
3. Set C: Complexity based meta-features

Each complexity specific algorithm selection model for a certain meta-feature set is trained in a wrapper like kNN model where the value of k varies from 2 till 10. The best performing model out of all the meta-features and the corresponding k value is chosen by comparing the accuracies of all the models when tested on the validation set. The best performing complexity specific algorithm selection model for each cluster (2 layered) is then compared

to the baseline accuracy (1 layered), which is the best algorithm selection model over all the datasets for each algorithm pair by calculating the accuracy on the test set.

Algorithm Pair		Baseline	Cluster	Algorithm Pair		Baseline	Cluster
Dt	Knn	81.625	82.862	Knn	Rf	81.269	82.58
Dt	Lda	82.76	84.004	Lda	Nb	85.471	85.886
Dt	Nb	88.137	88.909	Lda	Qda	84.108	84.425
Dt	Qda	89.487	90.225	Lda	Rf	90.672	90.928
Dt	Rf	88.958	88.95	Nb	Qda	84.508	84.963
Knn	Lda	84.656	85.62	Nb	Rf	96.036	95.965
Knn	Nb	91.776	92.161	Qda	Rf	97.682	97.536
Knn	Qda	91.122	91.307				

Table 5. The two-layered vs one-layered framework

Based on our results we can conclude that the accuracy of the complexity specific two-layered model is either comparable or an improvement to the baseline one layered model. Hence, our proposed two-layered framework is a practical substitute for the algorithm selection problem.

Algorithm Pair		Baseline Accuracy	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Dt	Knn	81.625	89.356	86.928	79.245	73.705
Dt	Lda	82.76	94.117	81.528	80.341	76.518
Dt	Nb	88.137	92.178	91.111	86.387	77.894
Dt	Qda	89.487	96.218	91.273	87.037	82.608
Dt	Rf	88.958	97.297	95.205	88.113	81.111
Knn	Lda	84.656	92.118	90.74	88.172	77.865
Knn	Nb	91.776	94.805	94.267	93.827	89.256
Knn	Qda	91.122	96.398	95.041	91.304	83.516
Knn	Rf	81.269	93.627	85.84	71.428	67.283
Lda	Nb	85.471	95.731	92.561	83.471	81.912
Lda	Qda	84.108	95.031	92.436	84.955	74.742
Lda	Rf	90.672	96.666	94.805	91.411	82.52
Nb	Qda	84.508	88.378	86.784	83.798	67.741
Nb	Rf	96.036	98.138	97.457	96.774	91.279
Qda	Rf	97.682	99.163	97.686	97.546	95.121

Table 6. Cluster wise accuracy (green: equal or more than the baseline, red: less than the baseline)

That said, in the cluster sets, although for all problems, two or more clusters performed significantly better than the baseline accuracy, the other clusters did not. From this, we may infer that the better performing clusters contain key representative datasets which have distinct complexities, and consequently, further hypothesise that the clusters with poorer performance inadequately characterise specific categories of dataset complexity.

5. Future Work

The two-layered framework has yielded results that are slightly better than baseline. However, as noted certain cluster specific models perform worse. Thus, our future work includes examining these clusters to analyse their characteristics and acquire an insight about the key datasets which represent complexity.

6. References

1. Wolpert, D., & Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation,1*(1), 67-82. doi:10.1109/4235.585893

2. Guyon, I., Chaabane, I., Escalante I, H. J., Escalera, S., Jajetic, D., Lloyd, J. R., . . . Viegas, E. (n.d.). A brief Review of the ChaLearn AutoML Challenge. Retrieved from http://proceedings.mlr.press/v64/guyon_review_2016.pdf

3. Dietterich, T G., Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation, , (1998) 10(7):1895-1924.

4. Soares, C., Zoomed Ranking: Selection of Classification Algorithms based on Relevant Performance Information. Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, 126-135. Springer. . (2000b)

5. Lagoudakis, M.G. and Littman, M. L., Algorithm selection using reinforcement learning. In Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000), 511-518, Stanford, CA. (2000)

6. Kalousis, A. and Hilario, M.: Feature Selection for Meta-Learning. In Proceedings of the 5th Pacific Asia Conference on Knowledge Discovery and Data Mining. Springer. (2001)

7. Munoz, M. A., L. V., Miles, K. S., & Baatar, D. (n.d.). (PDF) Instance Spaces for Machine Learning Classification. Retrieved from https://www.researchgate.net/publication/315835025_Instance_Spaces_for_Machine_Learning_Classification

8. Lorena, A., Garcia, L. P., Souto, M. C., Lehmann, J., & Ho, T. K. (2018, August 10). How Complex is your classification problem? A survey on measuring classification complexity. Retrieved from https://arxiv.org/abs/1808.03591

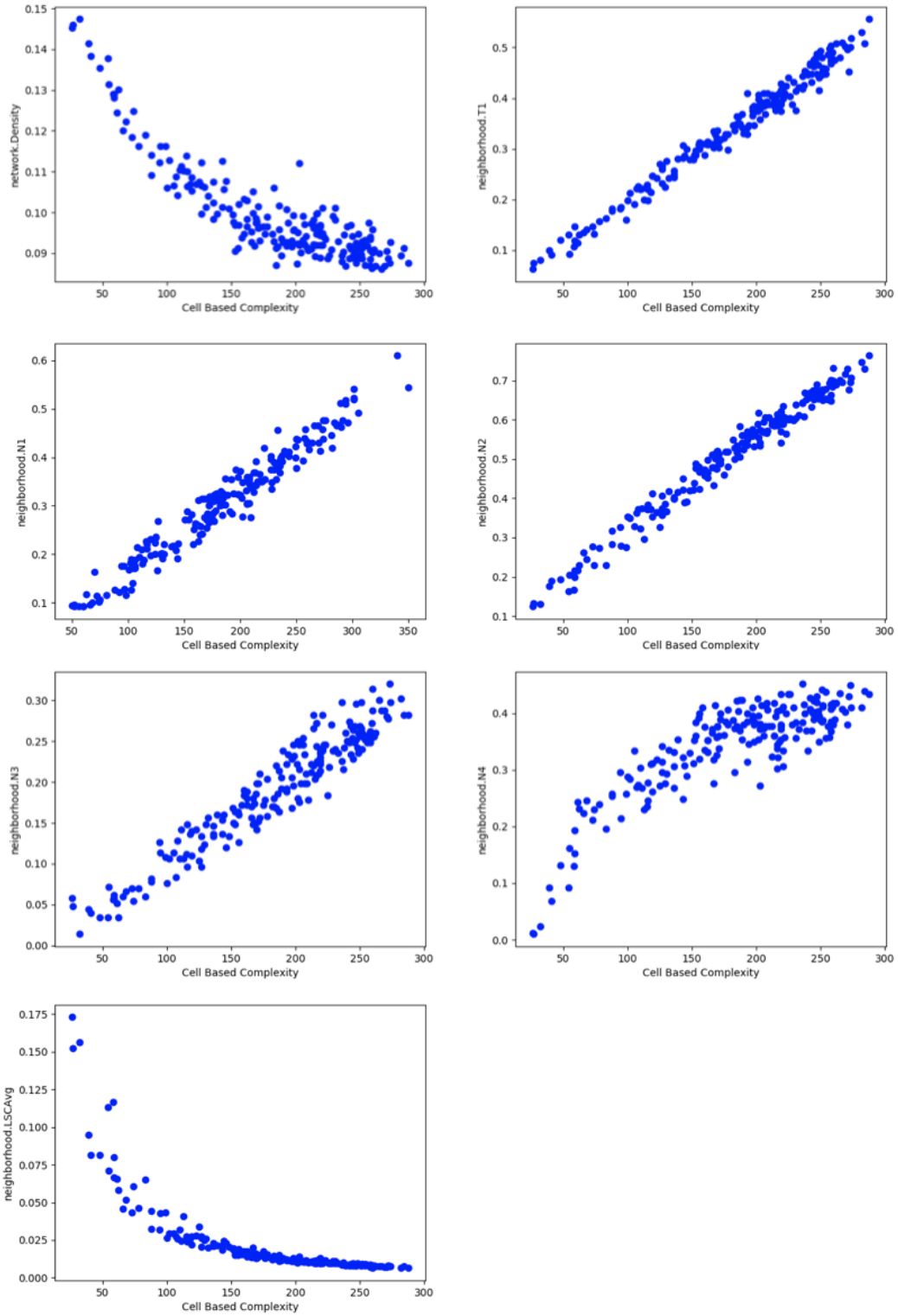
9. Ho, T. K., & Basu, M. (n.d.). Retrieved from https://ieeexplore.ieee.org/document/990132

10. Perlich, C. (2009). *Learning Curves in Machine Learning*. [online] Available at: https://www.researchgate.net/publication/247934703_Learning_Curves_in_Machine_Learning [Accessed 7 Jan. 2019].

11. Dietterich, T. G., & Bakiri, G. (n.d.). Solving Multiclass Learning Problems via Error-Correcting Output Codes. Retrieved from https://arxiv.org/pdf/cs/9501101.pdf

7. Appendices

Appendix A



Appendix B

Classical meta-features	Decision tree meta-features	Complexity meta-features
ClassEnt	treewidth	overlapping.F1
AttrEntMin	treeheight	overlapping.F1v
AttrEntMean	NumNode	overlapping.F2
AttrEntMax	NumLeave	overlapping.F3
JointEnt	maxLevel	overlapping.F4
MutInfoMin	meanLevel	neighborhood.N1
MutInfoMean	devLevel	neighborhood.N2
MutInfoMax	ShortBranch	neighborhood.N3
EquiAttr	meanBranch	neighborhood.N4
NoiseRatio	devBranch	neighborhood.T1
StandardDevMin	maxAtt	neighborhood.LSCAvg
StandardDevMean	minAtt	linearity.L1
StandardDevMax	meanAtt	linearity.L2
SkewnessMin	devAtt	linearity.L3
SkewnessMean	-	dimensionality.T2
SkewnessMax	-	dimensionality.T3
KurtosisMin	-	dimensionality.T4
KurtosisMean	-	balance.C1
KurtosisMax	-	balance.C2
-	-	network.Density
-	-	network.ClsCoef
-	-	network.Hubs