Analyzing Data Complexity Using Metafeatures for Classification Algorithm Selection

Rushit N. Shah, Varun Khemani, Michael H. Azarian, Michael G. Pecht
Center for Advanced Life Cycle Engineering (CALCE)
University of Maryland
College Park, MD 20740, USA

Yan Su
Civil Aviation College,
Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

Abstract— **Supervised machine learning is defined as the task of seeking algorithms which, based on reasoning gained from historical data, produce a general hypothesis, can then be used to infer knowledge about new data in the future. Classification, an instance of supervised learning, is statistically defined as the problem of identifying to which particular subpopulation a new observation of data belongs. Given the large number of available classification algorithms, their combinations (e.g. ensembles) and possible parameter settings, finding the best method to analyze a new data set has become an ever more challenging task. Typically, finding the best classifier for a given data set involves empirically iterating through all candidate classifiers and choosing the one which provides the best classification accuracies. Clearly, this task is computationally very expensive and the computation cost increases with the addition of each new candidate algorithm. This problem is compounded by the fact that there is not adequate generalizability of these classification methods over data sets from different domains. For example, classifier performance obtained with medical imaging data may not hold for financial data when trying to achieve a similar classification task using the same classifier. How, then, does one efficiently choose a classifier which will provide better classification performance than others? In this context, this study aims to streamline the task of algorithm selection for classification using the meta-learning framework.**

We propose a methodology to analyze empirically a set of measures of data complexity, known as metafeatures, and investigate their influence on the classification performance of several widely used classifiers. Doing so allows a map of a performance metric to be generated over the metafeature space of data sets. This map is partitioned into regions where some classifiers perform better than others. Once implemented, a new data set can be located in the metafeature continuum and the appropriate classifier for the new data set can be chosen as the one that performs best in that region of the map. The problem of algorithm selection, then, involves merely calculating the metafeatures for a new data set.

Keywords—metafeatures, algorithm selection, data complexity

I. INTRODUCTION

A classifier is an algorithm that implements the task of classification. Classifiers are used in a broad variety of fields [1] [2]. With the development of faster computational systems, and with the increasing ease of data collection, a growing array of classifiers have been employed in applications as diverse as spam e-mail detection, fraud detection, market segmentation, fault detection in mechanical systems, and image processing, amongst others. Despite theoretical studies, the empirical behavior of individual classifiers strongly depends on the nature of the training data. Theoretical studies analyze classifier behavior for different types of problems but result in weak performance bounds [3]. Empirical studies, on a small selection of data, try to characterize the aforementioned dependence of a preferred classifier on the nature of the training data. However, with the advent of the 'Big Data' paradigm, the variety of data available is vast and, in most cases, it is not possible to generalize the results of an empirical study on one type of data set to other data sets. In [4], an attempt was made to study this problem from the classification-difficulty point-of-view. The use of certain metafeatures was proposed, which characterize the geometrical complexity of the class distributions and the separability of the different classes of data. These metafeatures allow any classification problem to be viewed in the feature space comprised of these metafeatures, regardless of the domain from which the data was collected. The difficulty of a classification problem is described by its position in this continuum.

In general, *metaknowledge* is defined as knowledge about knowledge [5]. Given a theory, knowledge is what can be expressed in the theory. For example, given a language, a deductive apparatus, and a set of axioms, the axioms and the theorems derivable from the axioms constitute *knowledge* of the theory. Facts that can be proven *about* the theory constitute *metaknowledge* of that theory [6]. In the context of computational science, metaknowledge refers to the relationship between algorithm performance and the characteristics of the task at hand; for example, characteristics of input data to a mathematical model. The most common form of meta-knowledge involves extraction of morphological features from the training data. These metafeatures can be classified into five main categories – simple, statistical, information-theoretic, geometrical, and model-based [7] [8]. Multiple algorithm-recommendation frameworks have been proposed recently, which rank a finite set of candidate classification algorithms based on the computed metafeatures of the data set being analyzed [4] [7] [8]. Table 1 lists a subset of metafeatures, some of which were proposed in [4] and others that were used in [3] in an algorithm-recommendation

2166-5656/18/$31.00 ©2018 IEEE
DOI 10.1109/PHM-Chongqing.2018.00224

framework.

Although algorithm-recommendation frameworks exist, it is not known how some of these metafeatures individually or collectively influence the performance of a given classifier. For example, it is known that although linear discriminant analysis (LDA) carries an underlying assumption that the class-conditional probability densities must be normally-distributed, if the classes in the data are adequately separated, LDA provides near-optimal classification results regardless of the actual distribution within the classes. This separability is captured by Fisher's discriminant ratio, and consequently it is known that the performance of LDA and Fisher's discriminant ratio are positively correlated. Thus, using this as a motivating example, we propose a methodology to capture the influence of metafeatures on the performance of classifiers. The rest of the paper is organized as follows: the approach is outlined in section 2; in section 3 we describe our experimentation and results with artificial datasets; and conclusions follow in section 4.

II. Proposed Methodology

The methodology we propose requires the collection of several real-world data sets. Once the data sets have been collected, the set of metafeatures in Table 1 must be computed for each data set. The value of metafeatures computed for each data set, acting as coordinates, allow a data set to be located in the multidimensional metafeature space. For instance, if 18 metafeatures are computed, the data sets can be placed in an 18-dimensional metafeature space. The core idea behind this methodology is that the performance of every classifier depends on the characteristics of the data that it is applied to. For instance, LDA yields good classification accuracies only when the classes in the data are linearly separable; i.e., produce a high value of Fisher's discriminant ratio (FDR). However, for data sets containing nonlinearly separable classes, LDA yields poor results. Thus, at least the metafeatures FDR and nonlinearity dictate the performance of LDA.

Having computed the metafeatures for all data sets, the performance of candidate classifiers is measured on all data sets. This allows the metafeature map to be partitioned into regions where some classifiers perform better than others. For instance, data sets in the region of the map defined by low FDR ratios, and high nonlinearities are expected to have poor classification performance using LDA, as compared to more sophisticated classifiers such as support vector machine (SVM) or k-nearest neighbors (kNN). Figure 1 shows an illustration of such a map. $M_1, M_2, \ldots, M_N$ denote the metafeatures. The demarcated regions illustrate regions over the metafeature space where a particular classifier yields better classification performance than others. After computing metafeatures for a new data set, a user can determine the classifier best suited to their data set, based on where the data set is located on the map. It is worth noting that regions in the metafeature space are associated with only the candidate algorithms being considered in the study. As more classification algorithms are added to the analysis, a change in the topology of the map would be expected, as the algorithms yield better/worse performance than the previously considered algorithms.

TABLE I COMMONLY USED METAFEATURES

Category	Measure
Simple	1. Number of attributes 2. Number of examples 3. Number of classes
Statistical	1. Average asymmetry of attributes 2. Average kurtosis of attributes 3. Average correlation between attributes 4. Average coefficient of variation of attributes
Information-theoretic	1. Class entropy 2. Average entropy of attributes 3. Average mutual information between classes and attributes 4. Signal-to-noise ratio
Complexity	1. Maximum Fisher's discriminant ratio 2. Volume of overlap region between classes 3. Dispersion of the data set
Model-based	1. Non-linearity of linear classifier by linear programming (LP) procedure 2. Error rate of LP classifier 3. Non-linearity of 1NN classifier 4. Error rate of 1NN classifier

Apart from aiding selection of the best classification algorithm for classification, an analysis like this allows establishing limits on the performance of the candidate classifiers in terms of the metafeatures. Such information could potentially be valuable in understanding drawbacks of the candidate classification algorithms.

The proposed map provides a holistic view of different data sets, regardless of the domain from which they were collected. This representation allows comparing characteristics of data sets collected from domains as varied as rotating machinery and census data. It allows comparison of different data sets based on their proximity in the metafeature space. Further, it is anticipated that such an analysis would aide in characterizing the generalizability of a classification model. This idea is illustrated in Figure 1. For example, consider that data is collected from a fleet of five aircraft of the same make and model. In Figure 1, each star is used to denote one of these five data sets in the metafeature space. It is conceivable that owing to natural statistical variations between aircraft, the five data sets have different values of the computed metafeatures. If a model were to be trained on the data set that falls in the region where *classifier 3* performs best (*blue star* in Figure 1), the model will not be generalizable to the remainder of the aircraft in the fleet (*green stars* in Figure 1). This is an important practical consideration while addressing real-world problems. The proposed methodology allows ascertaining the generalizability of a model over the fleet *a priori*, which can be a valuable tool for any organization that maintains a fleet of systems. The value provided by this

methodology is not restricted to any one application domain. Also, the boundaries of the regions on the map are not anticipated that to be as distinctly defined as shown in the illustration. The decline in performance of a classifier is expected to be gradual along the boundaries of the regions. Thus, the boundaries are expected to be fuzzy, and the choice of a classifier can be more flexible. In other words, if a data set lies at the boundary of two regions, a user may choose either classifier for their application with similar performance. This leads to another consideration: *how far away from a boundary of two regions must a data set lie to state with certainty that a classifier performs better on the data than others?* This depends on the decline in performance acceptable to a user. For example, in the case of the fleet of five aircraft, a user may continue using *Classifier 1* for the data set marked with a *blue star* if it is determines that the error rates are within limits acceptable to the user/organization.

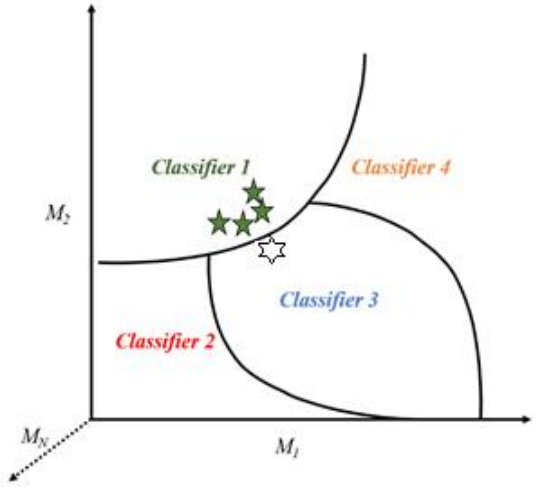

Figure 1: Illustration of a proposed map over the metafeature space showing regions where performance of a given classifier is better than others. The stars represent different representative data sets collected from different instances of a system from a single

Since this work intends to address issues faced in the real-world while solving classification problems with real-world data sets, the nature of this work is empirical. While adopting the proposed methodology, the number of data sets over which the metafeature map is built must be large so as to judge the performance of candidate classifiers on a large variety of complexities in the data, and obtain an algorithm recommendation system which can make better recommendations for any data set.

Figure 2 shows the workflow which is to be adopted to implement the proposed methodology. As a pre-processing step, all data sets must be scaled using z-score normalization, since they have been collected from a variety of different domains. In Figure 2, D_i denotes the different data sets collected and used; M_i denotes the metafeatures being considered; and A_i are the candidate classifiers being considered. In order to evaluate the performance of the candidate classifiers, a performance metric must be chosen which captures not only the classification accuracy of a particular classifier but also the training time.

$$AS = \frac{Classification\ Accuracy}{Training\ time\ (s)} \quad (1)$$

We propose using the metric called Algorithm Score (AS) shown in equation 1. The metric is a ratio of the ten-fold cross-validated classification accuracy of a given classifier, and the time (in seconds) that it takes to train that classifier on the given data set. The benefit of this is twofold. Firstly, training time is a useful practical consideration in selection of a classification algorithm. If two algorithms yield equivalent classification results, the algorithm requiring a shorter training time is preferred. Secondly, although the classification accuracy has an upper bound of 1, the chosen metric, due to the training time in the denominator, does not have an upper bound. This will be valuable in updating the metafeature map in the future as faster algorithms are developed. The selection of the winning algorithm (A_{winner} in Figure 2) is made based on the algorithm yielding the highest *Algorithm Score*. Performing Principal Component Analysis (PCA) on the metafeatures can also provide a good indication of how much each metafeature contributes in explaining the variations across data sets, and can aid in choosing which metafeatures to use.

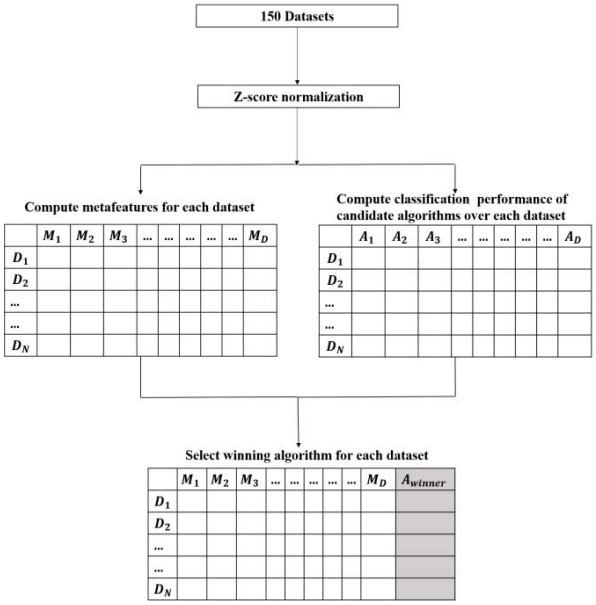

Figure 2 Workflow to be adopted for execution of research plan

III. Experimental Results

We make use of artificially-generated data sets to demonstrate the implementation of the proposed methodology, and to demonstrate the concept of dependence of classifiers on metafeatures. The classifiers whose performance we evaluated are listed in Table II. These classifiers are simple in their formulation and are used widely in machine learning applications.

TABLE II LIST OF CLASSIFIERS USED TO EVALUATE CLASSIFICATION PERFORMANCE

Classifier	Reference
Linear Discriminant Analysis (LDA)	Fisher (1936)
Quadratic Discriminant Analysis(QDA)	Fisher (1936)
1-Nearest Neighbors	Fixes & Hodges (1951)
5-Nearest Neighbors	Fixes & Hodges (1951)

Sixteen one-dimensional data sets containing two classes of data were generated. For the first class of data, points were sampled randomly from a normal distribution, $N(0,5)$. This distribution was fixed. For the other class of data, points were sampled from a normal distribution $N(\mu,\sigma)$ where μ was varied from 0 to 15 and σ was fixed to be 1. For each of the sixteen data sets, Fisher's discriminant ratio (FDR) was computed as the metafeature being examined. The four classifiers were trained on each data set and ten-fold cross-validation was performed to obtain classification accuracies. The training time for each classifier was recorded. The classifiers used were LDA, QDA, 1NN, and 5NN. The algorithm score (AS) for each algorithm was calculated as shown in equation 1. Figure 3 shows the ten-fold cross-validated classification accuracy over each data set. By examining merely the classification accuracy, it can be inferred that QDA and 5NN classifiers perform best over the entire metafeature space in this plot. However, the interesting features of this plot are the intersections between performance curves of various classifiers. Up to FDR value of ~2.5, LDA outperforms 1NN, beyond which 1NN is the better classifier. Similarly, up to an FDR value of ~5, QDA outperforms 5NN, after which the performance of QDA, 1NN and 5NN classifiers is almost comparable. The region of the metafeature space above an FDR value of ~8 would represent the fuzzy regions between boundaries of three of the four classifiers on the map of classification accuracy over the metafeature space.

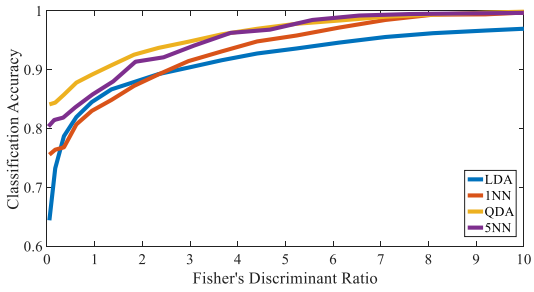

Figure 3 Ten-fold cross-validated classification accuracy for four classifiers over the metafeature FDR

Figure 4 shows the *Algorithm Score* of the four classifiers over the range of data sets generated. Here the training time is a factor in judging the algorithm performance. It is immediately apparent that 1NN, despite yielding lower classification accuracies, may be better suited for users with lower availability of computational resources as it requires shorter training times. The intersections between the curves corresponding to LDA and 5NN are also interesting, in that they alternate as the better choice of algorithm over the metafeature space being considered.

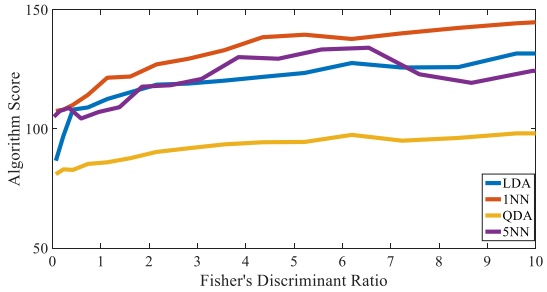

Figure 4 Algorithm Score for four classifiers over the metafeature FDR

Figures 3 and 4 are both equivalent to the proposed map over the metafeature space. While the metafeature map is expected to be multidimensional, the simple experimental study shows the one-dimensional equivalent of the proposed map.

IV. CONCLUSIONS

A methodology that recommends the most appropriate classifier to use, for a given data set, was proposed. The methodology involves initially collecting a large variety of training data sets and computing measures of data complexity, known as metafeatures, for each data set. This allows each data set to be placed in a multidimensional metafeature space. The performance of candidate classifiers is then determined on each data set, taking into account both the classification accuracy and training time in a proposed Algorithm Score. The inclusion of training time in the performance metric not only allows users to tailor the algorithm selection based on their accuracy, but also their computational efficiency; i.e., it is not sufficient that a classifier provide good results, it is also important that its execution be fast and inexpensive. Computing the metric for all data sets naturally partitions the metafeature space into regions where some classifiers perform better than others. Then, determining the most appropriate classifier for a given data set merely involves computing the metafeatures of that data set and determining which classifier yielded the best performance using the original data sets.

This methodology specifically takes into account the characteristics of a data set while recommending an algorithm, rather than the conventional approach of classifier selection which is based upon trial-and-error. Another key advantage of this method is its generalizability to any domain from which data may have been collected. This approach is as relevant to farmers trying to predict calving patterns in their cattle, as it is to aircraft manufacturers providing condition based maintenance (CBM) services to their customers.

REFERENCES

[1] A. R. S. Parmezan, H. D. Lee and F. C. I, "Metalearning for feature selection algortihms in data minind: Proposal of a new framework," Expert Systems with Applications, vol. 75, pp. 1-24, 2017.

[2] S. N. das Dores, L. Alves, D. D. Ruiz and R. C. Barros, "A meta-learning framework for algorithm recommendation in software fault prediction," in Proceedings of the 31st Annual ACM Symposium on Applied Computing, 2016.

[3] T. K. Ho and M. Basu, "Complexity measures of supervised classification problems," IEEE Transactions on Pattern Analysis and

Machine Intelligence, vol. 24, no. 3, pp. 289-300, 2002.

[4] M. Reif, F. Shafait, M. Goldstein, T. Breuel and A. Dengel, "Automatic classifier selection for non-experts," Pattern Analysis and Applications, vol. 17, no. 1, pp. 83-96, 2014.

[5] J. A. Evans and J. G. Foster, "Metaknowledge," Science, vol. 331, no. 6018, pp. 721-725, 2011.

[6] A. Luigi, C. Cecchi and D. Sartini, "Representation and use of metaknowledge," Proceedings of the IEEE, vol. 74, no. 10, pp. 1304-1321, 1986.

[7] S. B. Kotsiantis, I. Zaharakis and P. Pintelas, "Supervised machine learning: A review of classification techniques," in Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies, 2007.

[8] A. Ethem, Introduction to Machine Learning, MIT Press, 2010, p. 9.

[9] "Fleet DNA: Commercial fleet vehicle operating data," National Renewable Energy Laboratory, [Online]. Available: https://www.nrel.gov/transportation/fleettest-fleet-dna.html.

[10] R. A. Fisher, "The use of multiple measurements in taxonomic problems," Annals of Eugenics, vol. 7, no. 2, pp. 179-188, 1936.

[11] E. Fixes and J. L. Hodges, "Discriminatory analysis, nonparametric discrimination: Consistency properties.," USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

[12] L. Clemmensen, T. Hastie, D. Witten and B. Ersboll, "Sparse discriminant analysis," Technometrics, vol. 53, no. 4, pp. 406-413, 2011.

[13] D. Harris, D. I and V. N. Vapnik, "Support vector machines for spam categorization," IEEE Transactions on Neural Networks, vol. 10, no. 5, pp. 1048-1054, 1999.