On the influence of dataset characteristics on classifier performance

T. van Gemert

Bachelor scriptie Kunstmatige Intelligentie, 7.5 ECTS

Abstract—The field of Machine Learning has been rapidly gaining attention from both academic and commercial parties. To promote fast deployement of analytical solutions, several tools have been developed to aid the novice user. Concurrently, fields like meta-learning have been making great progress in developing models of algorithm performance on different datasets. One of the central issues in Machine Learning, for both novices and experts, is what learning algorithm to use on a given dataset. Although many solutions have been proposed, a definitive solution has yet to be found. We will argue that a possible solution lies in a deeper understanding of the data we are dealing with. By characterizing datasets in terms of meta-features such as the size of the dataset, we can compare and discuss different datasets and relate them to algorithm performance. A better empirical and analytical understanding of the data may also improve algorithm development, cause significant time-savings and present new insights. Focussing on classification algorithms, we present a number of ways in which meta-features can contribute to machine learning research. We will discuss several challenges and guidelines that have been proposed in the relevant literature and lastly we present what little is known about several meta-features and their relation to a classifier's performance.

Index Terms—Dataset characterization, meta-learning, meta-features, machine learning, classification, data mining

1 INTRODUCTION

In todays world, data is a cheap and easy to acquire commodity. A steadily increasing number of companies have extensive databases with data about their business or customers, while at the same time researchers have started initiatives to collect and publish datasets on a wide range of topics such as wildlife movement or credit scores. The broader field of Artificial Intelligence has contributed greatly to the use and development of smart algorithms that can 'learn' from data. AI and Machine Learning techniques such as classification, clustering or regression are used to, for example, diagnose a patient or predict equipment failure. Data mining is the name of the overall process of gathering, pre-processing, transforming, and analyzing data with the goal of discovering patterns and transforming information into actionable knowledge. Machine Learning concerns itself with the design and application of algorithms for analyzing data and producing patterns, predicting future data points or classifying unseen examples. Machine Learning is a largely independent sub field of Artificial Intelligence which has been widely used in areas such as computer-aided diagnosis, natural language processing, image classification, object recognition, etc. Subsequently, tools have become available for use by laymen and professionals to aid them in analyzing their data (e.g. RapidMiner.com, Prediction.io, DataRobot.com, Microsofts Azure Machine Learning, Googles Prediction API, and Amazon Machine Learning, R, WEKA, OpenML).

A particular concern of this paper is the use of classification algorithms and their relation to the data they are trained on, specifically the influence of dataset characteristics on the perfor-

- *T. (Thomas) van Gemert is a student (3995224) in the bachelor programme Artificial Intelligence, Faculty of Humanities at Utrecht University, Utrecht, The Netherlands.*
  E-mail: t.vangemert@students.uu.nl
- *dr. A. J. (Ad) Feelders is the supervisor and first examinator of this Bachelor Thesis. He is with the Department of Information and Computing Sciences at Utrecht University.*
  E-mail: A.J.Feelders@uu.nl
- *dr. T. B. (Tomas) Klos is the second examinator of this Bachelor Thesis. He is an assistant professor at Utrecht University in the Artificial Intelligence bachelor programme.*
  E-mail: t.b.klos@uu.nl

mance of supervised classification algorithms. In a broad sense classification is any situation in which a decision about a piece of information is made. In a more formal way, following Michie et al.'s definition, classification is the construction of a procedure that is applied to a series of cases, wherein each case is assigned one of a set of predefined labels/classes based on the observed features in the dataset [1]. In this paper we will refer to classification as meaning supervised classification; in which a classifier is trained on a set of examples with associated labels. In light of the large number of different classification algorithms and datasets available, some categorization has been proposed to group classifiers by their 'family'. For example, classification algorithms can be categorized as Statistical, Machine Learning (e.g. Decision Trees or Rule-based classifiers) and Neural Networks [1], [2], while datasets may be databases, ordered sets, streaming resources, etc.

One of the most challenging problems in Machine Learning is the determination of performance indicators in algorithms and the prediction of the best performing algorithm on a given dataset. This problem remains an issue to this day and has been studied for a long time, albeit sparsely (e.g. [1], [3]–[7]). When considering the availability of hundreds of machine learning algorithms and thousands of datasets with possibly millions of data points, there is a surprising lack of guidelines and standardization of what algorithm to use on a dataset, what parameters and features to select or how to pre-process the data [8]. Datasets themselves offer little information on the type of analysis to use. Picking the right algorithm for a dataset, as well as configuring and using an algorithm to its maximal potential so as to maximize its usefulness, proves to be an obstacle for both laymen and experts [9]–[11]. This is largely due to the large number of different algorithms available and their difficulty in deployment, fine-tuning and interpretation. At the same time more data than ever before has become available to users, but the quality of the data has not necessarily improved. These challenges require novel solutions based on not only smart AI techniques (such as meta-learning), but also on a proper empirical foundation of data analysis. Discovering relationships between classifiers and datasets requires the data to be comparable to other datasets: using meta-features to characterize datasets and make crude predictions of algorithm performance without the use of models will be the central theme in this paper.

1.1 Opportunities for meta-features

This paper aims to address the lack of a deeper understanding of the role of meta-features in the classification process. Most work concerning meta-features and data characterization has been done in the context of meta-learning, the field of research that aims to improve algorithm performance and selection through learning from past experiences (See section 4 for a more thorough introduction). It would seem that algorithm performance is closely related to the characteristics of the data, so it follows that to obtain the best results from a classification procedure an algorithm should be used that is known to perform well[1] on data with these characteristics.

There's a number of cases where a deeper understanding of meta-features would be beneficial. First of all, there is empirical evidence to support the No Free Lunch theorem [1], [10], [12], which states that no algorithm will perform best on all problems [13]. It would thus make sense to use an algorithm that is tailored to the problem at hand. Second, testing and fine-tuning several algorithms to find the best one is tedious and very time-consuming. A guideline provided by the data would not only help the user, but could also improve automatic selection procedures. Third, a deeper understanding of the data and differences between datasets would be helpful in designing (novel) classification algorithms, especially considering that tailored algorithms should perform best. Fourth, the superiority of a classifier may be

[1] In theory it is only possible to reliably compare two different classifiers when they are used on the exact same dataset. However, as Michie et al. note [1], the sometimes radically different nature of two algorithms makes it impossible to correctly compare them. The measure of accuracy is quite relative as well: In some cases an error of 15% is considered very good, while 5% may be bad in others.

restricted to a given domain characterized by some complexity measures [14]. Knowledge of the influence of these complexity measures may help us understand classifier limitations. Fifth, in most of the empirical literature classifiers are used in their baseline configuration, to make fair comparisons, while they usually perform much better when properly fine-tuned. Fine-tuning requires knowledge of the implicit requirements of the data. Sixth, it has been noted throughout the Machine Learning community that a standardized format for datasets for learning is lacking [15]. Knowledge of meta-features could guide the process of developing such a standardization. A last couple of challenges that would deserve attention on a meta-feature level are that meta-features appear to be largely inter-dependent, that performance data is missing for many algorithm/dataset combinations, and the use of artificial data which is likely not very representative of real-life applications, although it may be useful in meta-feature research and establishing some general relations [8].

1.2 Aim of this paper

Most approaches on selecting the best performing algorithm for a dataset are focused on automated procedures (e.g. meta-learning, feature-selection). We notice a lack of empirical foundation for the many meta-features found in the literature, and for the interpretation of the results obtained by developing meta-algorithms using these measures (e.g. [16]). The goal of this paper is thus to provide an overview of the challenges and guidelines concerning the influence of dataset characteristics, meta-features, on the performance of classification algorithms. Our central question is as follows:

"What do we know about the influence of dataset characteristics on classification algorithm performance?

To answer this question we will aim to answer related questions such as: "What meta-features are available?", "What meta-features contain the most information in any given dataset?", "What are the sensitivities of types of classifiers to different datasets?" and "What ways are available to describe a dataset?". Our questions differ from those answered by the field of meta-learning in the following way: whereas in meta-learning a model is constructed to describe the influence of meta-features on classifier performance, we focus on a more basic approach, a more *a priori* knowledge of this relationship without the need for meta-learning techniques. In **Section 2** we will discuss some related work; Landmarking in particular. We will then define and discuss meta-features, some challenges, and guidelines in **Section 3**. In **Section 4** we will discuss meta-learning, which is the most active area of research concerning the exploitation (and to some extent, understanding) of dataset characteristics. We will discuss some popular dataset characteristics and their relation to classifier performance in **Section 5**, in the categories Simple, Statistical and Information Theoretic. We conclude in **Section 6**. Our goal is to provide a broad overview of the current state-of-the-art of meta-features and their influence on classifier performance, as well as show a need for further research on this topic. We hope that addressing the issues mentioned here will spur further research to create a better understanding our data, and ultimately enhance our results in using Artificial Intelligence to learn from data.

2 Related Research Topics

Several different, but related research areas have concerned themselves with the topic of meta-features and data characterization. We will briefly discuss the most notable here, especially the topic of landmarking, while meta-learning is discussed in Section 4.

An example of an early use of meta-features to link algorithm performance and data characteristics is found in Average Case Analysis, which constructs analytical formulas to predict the performance of an algorithm on a dataset with certain characteristics (e.g. [6]). This approach was preceded however by the STATLOG project, which discussed meta-features at large and used them in an early form of meta-learning [1]. Aha (1992) proposed a method for generalizing from case-studies and produced rules to indicate what algorithm would perform best on a dataset [3]. See [16] for an elaboration on these related works.

The availability of good data is closely related to the issue of building a 'perfect' classification model. Most, if not all, datasets are rather specific to a certain topic and may contain noise, missing values or irrelevant data, especially if the data has been acquired from a real-life application. Given the increase of available datasets and massive data streams it has become increasingly hard to obtain and create 'clean' datasets. Data pre-processing is the procedure of removing detrimental data points from the data and transforming the data to a format that is easily learned from by classifiers. Cases where data is used for learning without pre-processing are rare, or even non-existent. Data pre-processing influences data in three directions: Data cleansing (noise, extremes, redundancy), altering dimensionality (attribute generation, filtering, transformation), and altering data quantity (sub/oversampling, balancing) [17].

A couple of tools have been developed to automate the process of selecting the best algorithm and fine-tuning it. Thornton et al. developed 'Auto-WEKA' as a tool to automate this in learners that can be used in the WEKA package. They achieved some promising results, although their focus was mostly on parameter and algorithm selection optimization [18]. Following this approach and extending it, were Feurer et al.: They developed a similar tool called 'auto-sklearn' to automate algorithm selection and tuning for algorithms in the *scikit-learn* Python package [10]. Compared to Thornton et al.'s approach they added a meta-learning step to "warm-start" the Bayesian optimization by using meta-features of a large number of datasets and they integrated automatic ensemble construction. Their method seemed to best Thornton et al.'s, but the most interesting result was probably that while testing hundreds of classifiers, no classifier was the best on more than half of all datasets [10]. Although tools like these show promising results in predicting a (very) good classifier for a given dataset, they require previous knowledge (for 'auto-sklearn') and a lot of time to train, plus there is not much interpretation of the results, and no interpretation of the factors that led to the results in the literature.

2.1 Landmarking

Landmarking is a method for selecting machine learning algorithms using very simple learners, as proposed by [19]. Landmarking quickly determines an estimate of an algorithm's performance on a dataset by first using a simple learner on the dataset. The results of this simple learner, that roughly resembles the full-fledged learner, are then assumed to give some kind of indication of the performance of the full-fledged algorithm on that dataset [11]. When using this strategy over multiple datasets and with multiple different simple learners, the results of the simple learners can be used to characterize the datasets and consequently relate these to algorithm performance. Landmarking is one of the ways to characterize datasets by looking at the performance or morphology of a classifier that has been trained on the data: A simple learner's performance should be very similar on datasets that are similar. Another approach to this would be to look at the number of nodes in a decision tree that has been trained on the data, for example. In Section 3 we will discuss some different ways of characterizing datasets. In this paper we will mostly concern ourselves with meta-features like 'size', but Landmarking [19], Yardsticks [1], and sampling-based landmarking [20], [21] show some interesting new perspectives on dataset characterization.

Yardstick is a early form of Landmarking proposed by Michie et al. to compensate for the difficulties in using some of their meta-features [1], which inspired the work by Pfahringer et al. In their original paper, Pfahringer et al. first use three landmarkers (linear discriminant, naive Bayes kernel, and C5.0 trees) and a number of meta-features to show that landmarkers can aid in improving the predictions of meta-learning algorithms. In the second part of their work they used 10 meta-learners to classify a problem as one for a specific classifier or one where none of the learners would excel. In this meta-learning task, only landmarkers were used as meta-features. They used a best decision tree node (according to the best information gain), a random node, a worst node, and an elite nearest-neighbor learner as landmarkers. These landmarkers give an indication of the linear separability of the problem, and the number of irrelevant attributes. Their results show that landmarking in meta-learning can be successfully used to predict the suitability of a type of classifier (e.g. Decision Tree, Neural Network) [19].

Landmarking has produced some interesting and promising results in the prediction of the best performing algorithm on a new dataset. A couple of issues remain, however. It is not clear what the relation is between a landmarker and the dataset at hand. There have also been several cases in which the performance of a landmarker was not significantly predictive of the performance of the full-fledged learner. The work by Phafringer et al. discussed above used mostly artificial data and reports that their choice of landmarkers and meta-learners was quite ad-hoc. More work on this topic would be needed to see if landmarking can be successful in meta-learning and dataset characterization. Although using landmarkers is significantly faster than trying different learners, it is still a time-consuming process, which may be one of the reasons that its use in meta-learning so far is limited [10]. For more on the topic of landmarking see [11], [19], [21], [22].

3 Introduction to meta-features

A dataset characteristic is some way of describing a property of a dataset. Several ways of doing so are possible: while we focus on 'classes' of dataset characteristics (e.g. the number of examples, the class proportion, the entropy of classes, or the modality of the data), another way would be to use algorithm performance on a dataset, as we saw in Landmarking, or the morphology of the learned model [1]. In Section 5 we will present and discuss some popular meta-features. In a learning task an algorithm learns from several features of a dataset, which, in the case of meta-learning, are meta-properties of the dataset. Hence, these dataset characteristics are called meta-features in the field of meta-learning. We will use both 'dataset characteristics' and 'meta-features' interchangeably in this paper. Dataset characteristics and meta-features represent the type of information in a dataset, how the information is distributed, and in what form it is. They describe the nature of attributes, the attributes themselves on a meta-level, associations between attributes, the association between an attribute and a target value, and more [22].

Michie et al. were one of the first ones to propose a categorization of types of meta-features: They used meta-features in the categories 'simple', 'statistical' and 'information theoretic', based on the historical origin of these measures [1]. Simple measures are, for example, the number of examples, the class proportions or the modality of the data (nominal vs. numerical). Statistical and information theoretic measures both relate to a specific family of classifiers: Statistical measures are especially useful in predicting the performance of numerical-based classifiers. Information theoretic measures are mostly used for nominal values or combinations of numerical and nominal values, and are closely related to the performance of decision trees and rule-based learners, for example [1], [17]. The measures proposed by Michie et al. have been widely used in related research (e.g. [2], [18], [22], [23], although there has been some criticism on the choice of measures as well [1], [16], [24].

One of the biggest challenges is what meta-features contain the most information, in meta-learning [8] but in general classification as well [1]. Michie et al. have tried to use multidimensional scaling to extract the essential features, but they were unable to find a proper conclusion due to the large differences between datasets and algorithms used. In a case analysis by Aha (1992) the meta-features were varied and it was found that this caused significant performance differences between the algorithms tested [3]. In a different approach to defining dataset complexity, Ho et al. found that certain complexity measures can be used to describe how hard a problem is, and that this can be linked to algorithm performance (especially class overlap seemed to be of significant influence) [25]. Kalousis et al. also analyzed the usefulness of certain meta-features and concluded that the number of examples, the ratio of examples to attributes, the class entropy and the information gain were especially informative [16]. A meta-analysis of classification algorithms concludes that homogeneity of covariance between classes increases prediction performance, that canonical correlation is related to the performance of neural networks and that a higher value of the skewness of the data may improve statistical classifier performance [2].

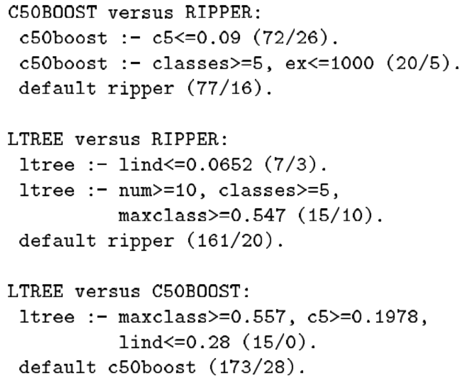
One of the most important qualities of a good dataset characteristic is its discriminative power: it should contain enough information to distinguish between datasets and algorithms [8]. Also, they should not be too computationally complex, or more time would be spent calculating meta-features than would be spent trying out different algorithms [1], [8], [19]. Because the characteristics of a dataset are directly representative of the data in the dataset, some caution should be used in transforming or pre-processing the data [22]. This has also been noted when converting nominal values to binary values, to make them compatible with numerical algorithms [1]. There is some discussion and experimentation on the representation of meta-features: most researchers use the mean of values for the meta-feature, or a ratio of that mean as a measure of said meta-feature, but this may cause severe loss of information and it does not scale well with datasets of different sizes [22]. Other methods have also been successfully applied, like logic programming [26] and histograms [22].

As mentioned earlier, both the meta-data and the data itself are likely not clean before usage and may contain missing values, extreme outliers and skewed proportions. Some have suggested to counteract these issues, to obtain more meta-data and to enhance research on meta-features, by artificially generating data (e.g. adding/removing noise, over-/undersampling attributes). This approach has been moderately successful in drawing conclusions on the role of meta-features [8], [27], but it seems unlikely that artificial data is very representative of real-world problems [1], [28]. An interesting development is the use of large data streams and extreme data mining, where the huge amount of available information compensates for some issues with for example class proportion and noise [29], [30].

4 META-LEARNING

The characterization of datasets has attracted the most attention in meta-learning research [1]. A number of slightly different definitions of meta-learning exist, but Vilalta et al. provide a concise one: ”Meta-learning studies how learning systems can increase in efficiency through experience; the goal is to understand how learning itself can become flexible according to the domain or task under study.” [31]. Basically, meta-learning concerns itself with the use of learning systems to make other learning systems flexible based on the problem at hand. Machine Learning algorithms are used to map dataset characteristics to the relative performance of algorithms. A simple meta-learning procedure would be: Characterize a number of datasets by their meta-features, benchmark a number of classifiers on these datasets and use a learning algorithm to relate the performance of the classifiers to the meta-features. Meta-features are especially important here because they are used to define a dataset as a specific type and consequently differentiate it from other datasets. After training the meta-learning algorithm, these measurable differences, or similarities, between datasets can be used to select the algorithm that would be likely to perform best on the new dataset, given that dataset's similarity to a previously seen dataset, and the different classifier performances on different datasets.

To provide an example of such a meta-learning model, we will discuss the meta-learning model constructed by Pfahringer et al. [19]. They used over 200 artificially generated datasets with different dataset characteristics to predict pair-wise winners for all pairs of C50BOOST (Boosted Trees), RIPPER (Rule-based) and LTREE (Discriminant Tree). The learners got 7 meta-features and the landmarkers as input to predict which algorithm



```
C50BOOST versus RIPPER:
 c50boost :- c5<=0.09 (72/26).
 c50boost :- classes>=5, ex<=1000 (20/5).
 default ripper (77/16).

LTREE versus RIPPER:
 ltree :- lind<=0.0652 (7/3).
 ltree :- num>=10, classes>=5,
          maxclass>=0.547 (15/10).
 default ripper (161/20).

LTREE versus C50BOOST:
 ltree :- maxclass>=0.557, c5>=0.1978,
          lind<=0.28 (15/0).
 default c50boost (173/28).
```

Fig. 1. Meta-learning model produced by the RIPPER learner [19].

would be better on a given dataset. The advantage of the RIPPER learner is that it is rule-based, and thus provides a very easy-to-interpret model, which we can see in Figure 1. In the top part of the rules in the figure, C50BOOST vs. RIPPER, we can see that RIPPER generally wins over C50BOOST (it is the default choice), but C50BOOST outperforms RIPPER if the C5-estimate (error estimate from the landmarker) is already lower than 9% or if the dataset contains more than 5 classes and the number of examples is less than 1000. Similarly, in LTREE vs. C50BOOST, we can see that C50BOOST almost always outperforms LTREE on any dataset, except in the rare cases when there is a relatively equal class distribution, the c5-estimate is quite high and the linear discriminant estimate is less than 28%. For a full explanation of the model and the results, see [19]. We have merely used this as an example of a meta-learning model, which quite clearly shows how a choice of classifier is based on meta-features.

Several challenges concerning meta-features have been noted in the meta-learning community. For example, meta-features are only useful if they contain information that can be used to discriminate between learners [1]. It is not yet clear, however, which meta-features are useful and which are not. Because most of the work done in meta-learning focuses on prediction and the construction of robust meta-learning systems, it suffers from the same lack of interpretation of meta-features as Machine Learning in general. Michie et al. also noted that one of their meta-features 'SD_ratio' was so computationally intensive that it would be much easier to just use some form of Landmarking with the algorithms that depend on 'SD_ratio' [1]. Brazdil et al. note that because the choice of parameters hugely affects the quality of the results, meta-learning methods should also provide guidelines for the selection of meta-features and parameters [8]. This is related to the challenge of deciding what meta-features to use: Just like with regular features, meta-features can suffer from noise, missing values or irrelevancy. Some meta-features may only be useful for a certain family of learners. Given the huge variety in real-world datasets an answer to this challenge is not easily formulated [1], [8].

The field of meta-learning has made some great progress in the characterization of datasets for predicting algorithm applicability. Although classes of meta-features (as opposed to landmarking, for example) seem to be the most widely researched category of characterization, some interesting results have been found in other categories as well (e.g. see [32], [33] for model-based meta-learning). It is generally accepted that meta-features are in some way or another crucial to the field of meta-learning. However, there is still a lack of interpretation and empirical evaluation of the influence of meta-features, and a lack of guidelines for usage and selection. To learn more about meta-learning, see [8]: A great book on the use of meta-learning in data mining that discusses practical applications, challenges and guidelines.

5 Discussion of meta-features

In this section we will discuss several meta-features as proposed earlier in the three categories: Simple, Statistical and Information Theoretic. We will discuss what, as far as can be concluded, their role/influence is on classifier performance and present the challenges that remain. Not all possible meta-features are covered. For some it is not clear enough what their influence might be, and often a slightly different set of meta-features is used in other literature. We have picked the ones that have been most common in the literature and those that have been most extensively covered.

5.1 Simple meta-features

Simple meta-features are dataset characteristics that are easily observable and cater to any family of classifiers. Some of the most widely used meta-features in this category are:

- Number of examples
- Number of attributes
- Number of classes / class proportions
- Ratio of nominal vs. numerical attributes
- Modality of the data
- Amount of missing values

The number of examples in a dataset is possibly the most obvious meta-feature of any dataset: it simply is a measure of how much data is in the dataset. The size of the dataset is at the same time possibly the most influential on the performance of classifiers. The size is directly linked to the training time of an algorithm, and specific issues arise when dealing with very little or huge amounts of data. A popular way of dealing with high training times is to use only a sample of the data, although this may cause underrepresentation or overfitting. While the number of examples is often regarded as an important measure in the literature, it is reportedly rarely used in data mining [8]. A reason for that may be that in an average dataset the size is not an interesting measure compared to innate algorithm performance or other measures. But when dealing with especially small datasets the number of examples is rather meaningful, because there is relatively little information in the dataset. Salperwyck et al. looked at the performance of algorithms on small datasets and found that generative algorithms (i.e. Naive Bayes) generally perform better on small datasets than discriminative ones (i.e. Decision Tree, Rule-based). Although the reason for this remains unclear, they argue that generative algorithms like Naive Bayes are able to very quickly construct a good model from little data [34]. An interesting question Salperwyck et al. pose is what the minimum number of examples would be to find an 'interesting' solution. Unfortunately this question remains unanswered. Nowadays (very) large datasets have become more easily available than ever before. These datasets offer huge potential, but large datasets also have their own share of challenges. Banko et al. for example have shown that very large datasets offer significant performance improvements for classification algorithms [29]. Large datasets offer a number of benefits due to their sheer size: Issues with noise, redundancy of examples and class underrepresentation may be overcome by the fact that there is simply more information available. Large datasets, however, may also be more complex, which calls for well-scaling algorithms that can deal with higher data complexity [1], [18]. The dimensionality of the dataset (the number of classes or number of attributes) may also pose a challenge: Most literature on data characteristics and algorithm performance uses only two classes for their empirical evaluation, but report that dimensionality may be a problem, for example for Neural Networks [12]. A higher dimensionality of attributes is likely to be counterproductive, according to [36], and large datasets are likely to contain a lot more features. A selection of features would be crucial in these cases. If computing power and time are not an issue, the advantages of big data can overcome issues like underrepresentation of classes, but most researchers argue that better data is still worth more than more data, although this view is getting challenged [30].

Another popular problem in classification is that of the bias-variance trade-off in classifiers. The bias and variance of a classifier may have a significant impact on their performance compared to other classifiers. The bias in a classification task refers to the error caused by modeling a real-life problem by a much simpler model. Variance measures the degree to which the predictions of the classifiers developed by a learning algorithm differ from training sample to training sample [27]. Brain et al. have found that the variance will decrease significantly with increasing dataset size. However, the bias seemed to remain constant, and thus bias management in classifiers may be critical for a good generalization performance [27]. Geman et al.

have also shown that bias and variance are influential in learning [35], but the question remains of what trade-off between variance and bias will render the best results. Classifiers with a low bias tend to have high variances and vice versa, thus finding the best trade-off between bias and variance would require some experimentation. Very different learning algorithms may be needed for applications on small and large datasets, and it is expected that classifiers learned from small samples will differ more significantly than classifiers learned from larger samples [27], [34].

The number of classes in a dataset and/or the class proportion (i.e. are all classes represented by the same number of attributes and examples?) is another widely used meta-feature, although the influence of this characteristic remains largely undiscussed. A classic opinion is that class imbalance (i.e. one class has much fewer examples/attributes than another class) can cause significant performance issues, especially when the minority class is the one of interest. Batista et al. tested this hypothesis and found that class imbalance is likely not an issue with most classifiers. However, class minority in combination with overlapping classes is very likely to be detrimental to classifier performance [37]. They also produced good results in using oversampling on the minority class and note that a learner's accuracy should be carefully interpreted: A 99% accuracy may seem to be very good, but if the 1% error represents the left out minority class of interest, the result is useless. Furthermore, Michie et al. report that some algorithms, like the Kohonen network (a Neural Network type), benefit greatly from equal class distributions.

The proportion of nominal attributes (e.g. names, places; symbolic or binary) to numerical attributes (i.e. continuous numbers) gives a measure of what kind of data is the majority in a dataset. It is widely known that algorithms show big differences in their ability to handle types of attributes (e.g. [1], [22]). For example, discriminant algorithms like Quadratic Discriminants are unable to deal with nominal data. This means that when presented with a largely nominal dataset, another algorithm has to be used or the nominal data has to be discarded or converted. Converting nominal attributes to numerical or binary values may cause loss of information, or greatly increase the number of attributes to learn [22]. On the other hand partitioning algorithms like decision trees are at an advantage when dealing with nominal data [1], [22], [38]. Depending on the proportion of a type of attributes some families of algorithms may perform much better on a dataset, or require complicated transformations on the data.

A related issue is that of modality: the modality of the data is the form it comes in or is transformed to. This is more a property than a measure but it nonetheless presents some interesting issues that come with certain types of data. Michie et al. report that a particular concern is that of hierarchical data. For example: the attribute 'Pregnant' is second to 'Sex', as it only applies when a value of 'Sex' is 'female'. Most algorithms perform badly with this kind of data, as they fail to abstract the underlying relations and are thus stuck with a lot of irrelevant data. Decision Trees, however, are especially good at mapping this kind of data to a model [1]. Another possible issue is that most datasets come in some sort of tabular format. However, some data, for example time-series data, may come in a completely different format and require a classifier that is able to deal with this kind of data [8]. This has become especially important with the onset of on-line learning and streaming data.

The last measure concerns that of missing data. It is widely known that missing data is detrimental to classifier performance, although some classifiers are better able to handle missing values than others [1], [23], [34]. Especially the distribution of the missing values may be crucial to classifier performance [28]. Fortunately a lot of good methods exist nowadays to deal with missing values, ranging from substitution by artificial data to using the mean value of the attribute. Unfortunately there is little consensus yet over what method is best, and it may depend on the data at hand.

5.2 Statistical meta-features

Statistical meta-features are measures of properties of a dataset that relate to the field of Statistics. They are generally only usable on predominantly numerical datasets and are particularly useful in relating the dataset characteristics to the performance of statistical/numerical classifiers. It has been suggested that these measures may be more generally applicable, but more research on that is needed [1]. The meta-features discussed here are:

- Standard Deviation ratio (SD_ratio)
- Correlation of attributes
- Skewness & Kurtosis
- Normality of the data distribution

The ratio of Standard Deviations as used in Michie et al. is a measure of how much the covariance matrices of classes differ, which is a crucial measure for predicting the performance and usability of discriminant algorithms. The SD_ratio is ″the geometric mean ratio of standard deviations of the populations of individual classes to the standard deviations of the sample″ [1]. Although this measure is very helpful in deciding whether to use a Linear or Quadratic Discriminant Analysis, its usefulness with other algorithms is unclear, if there is any use at all. Furthermore Michie et al. note that it is very computationally expensive to calculate the SD_ratio for even a moderately sized dataset, and that just applying and tuning the algorithm without a priori knowledge is faster. The regular standard deviation may be more useful in this case, but this is generally made equal to 1 in the data pre-processing step.

The correlation between attributes and between attributes and classes is another widely used, but poorly interpreted measure. It is known that some algorithms such as Naive Bayes that assume independence perform poorly on highly correlated datasets [1]. Correlation of attributes may also be affected by forms of pre-processing. There is some discussion on the usefulness of this measure in this context, as a proper feature selection or data pre-processing algorithm should be able to filter out any irrelevant or redundant data. However, datasets are rarely perfect, so it may very well be possible that the correlation of attributes can be used as a meta-feature. Kalousis et al. question its use as well, but use it nonetheless as a indication of whether some attribute influences another one, which would be important to statistical algorithms that makes assumptions about (in)dependence [22]. Unfortunately very little work on the influence of both correlation and standard deviation ratio has been done in the literature in the context of classifier performance. It is generally used as a measure for feature selection or meta-learning, where interpretation of its influence is lacking.

A more interesting meta-feature in numerical datasets may be the skewness and kurtosis of the data. Skewness is a measure of how asymmetrical a probability distribution is for a certain variable. Kurtosis is another measure of the shape of the probability distribution. These meta-features are often used in analytical discussion and meta-learning research, but very little is known about their relation to algorithm performance. It has been noted that a certain measure of skewness and kurtosis can influence the performance of a statistical learning algorithm [39], but it is unclear to what extent and for what values this influence is negative or positive [1], [39].

5.3 Information theoretic meta-features

Information Theoretic meta-features are derived from the field of Information Theory and are most appropriate for nominal attributes [1]. However, they are also able to deal with continuous attributes and are widely used throughout the Machine Learning community. Just like with the statistical measures, the measures discussed below often lack solid interpretation in the context of algorithm performance, but can be easily used to characterize datasets. Meta-features like Noise-Signal ratio have also been used as a meta-feature, but are not covered here. Noise is often dealt with in a data pre-processing step and it is widely known that noisy data degrades classifier performance. The information theoretic meta-features we will mention are:

- Class & attribute entropy
- Mutual information

The class entropy is closely related to the class proportions in the sense that the entropy of a class is a measure of how ’random’ that class is, or what the probability of that class occurring is. The entropy is maximal when all classes have an equal likelihood of occurring, and minimal when one class has a probability of 1 while the rest has a probability of 0. When some class has a much better representation (through more data point and/or attributes) than other classes, the overall entropy will thus be low. In line with what was previously mentioned, class entropy is deemed to be a useful predictor of algorithm performance and it is thus widely used in work on dataset characterization. However, Sohn (1999) noted that although many algorithms are sensitive to changes in the class entropy, the change in performance was not significant. It is thus not clear what exactly the influence of class entropy is [2]. It seems to be closely linked with the issue of class imbalance, which may be influential to some algorithms. It is also mentioned that the entropy of attributes may be more useful, as it is closely related to the relevancy of attributes [1], [2].

One of the most important characteristics of a dataset is the amount of information that attributes add to a class [22]. The mutual information between a class and an attribute is a measure of the shared information between the two [1]. The more shared information, the better the performance of an algorithm. The mutual information can also be used to determine irrelevant attributes, those that add no information to a class. Although widely used in the literature, the meta-feature of mutual information (sometimes: information gain) has not been thoroughly analyzed in the context of algorithm performance: Apart from ”more information enhances algorithm performance”, little is known about what values of mutual information are needed to achieve top performance.

6 Conclusions

In this paper we have initially stated the role of meta-features in the broad field of Artificial Intelligence and consequently the field of Machine Learning. Although meta-features are widely used in meta-learning, their use in other applications of machine learning may also be beneficial, for example in algorithm design. We have subsequently discussed a number of issues concerning meta-features, which mostly focused on the lack of interpretation of the relationship between meta-features of a dataset and the performance of a classification algorithm on that dataset. We have also discussed some meta-features themselves and have found that simple meta-features enjoy the best interpretation at the moment, although meta-features from statistical and information theoretic backgrounds are widely used as well. Unfortunately the literature is not clear as to what meta-features are the most important overall, but research on this issue is ongoing. It seems that it would not be wise to attempt to answer our central question at this point, as many answers are still unknown: It is clear that some meta-features, like the size of the dataset or the mutual information are influential to many algorithms, and while for some meta-features we know what values result in better classifier performance, many questions about this issue remain for meta-features like skewness and kurtosis, for example. However, we have mentioned several works that underline the importance of meta-features in classification. And furthermore, we have seen that meta-learning provides very promising results in relating meta-features to classifier performance.

Overall, to provide the machine learning community with an answer to the multiple questions regarding meta-features, algorithm selection and design, more research is needed. A good starting point, we believe, would be to assess what meta-features are crucially important in any dataset and why. One of the advantages of Artificial Intelligence is that it can develop algorithms to solve problems, like what algorithm to use, without needing an understanding of the data. We have seen successful approaches like this in meta-learning or automatic selection tools like ’auto-sklearn’. However, we believe that a deeper understanding of dataset characteristics would still be beneficial in algorithm design, speeding up classifier deployment and dataset pre-processing..

Acknowledgments

The author would like to thank Inergy Analytical Solutions B.V., and especially Michel Steenhuis, for the opportunity to do the orientation part of this work in partnership with them.

The author would like to thank the supervisor, dr. A. J. Feelders, of Utrecht University, for his patience, guidance and helpful information.

Lastly, the author would like to thank the anonymous contributors for their help in checking this work.

REFERENCES

[1] E. D. Michie, D. J. Spiegelhalter, and C. C. Taylor, *Machine Learning , Neural and Statistical Classification*. Taylor & Francis, Ltd., 1994. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.27.355

[2] Y. S. Sohn, "Meta Analysis of Classification Algorithms for Pattern Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1137–1144, 1999. [Online]. Available: http://ieeexplore.ieee.org/document/809107/

[3] D. Aha, "Generalizing from case studies: A case study," in *Proceedings of the Ninth International Conference on Machine Learning*, no. Section 2. Aberdeen, Scotland: Morgan Kaufmann, 1992, pp. 1–10. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.44.4156

[4] J. Gama and P. Brazdil, "Characterization of classification algorithms," *Progress in Artificial Intelligence*, pp. 189–200, 1995.

[5] A. Kalousis and T. Theoharis, "Noemon: Design, implementation and performance results of an intelligent assistant for classifier selection," *Intelligent Data Analysis*, vol. 3, no. 5, pp. 319–337, 1999.

[6] A. Langley, "Strategies for theorizing from process data," *Academy of Management review*, vol. 24, no. 4, pp. 691–710, 1999.

[7] P. Langley, J. N. Sanchez, L. Todorovski, and S. Dzeroski, "Inducing process models from continuous data," 2002.

[8] P. Brazdil, C. G. Carrier, C. Soares, and R. Vilalta, *Metalearning: Applications to Data Mining*, 1st ed., A. Bundy, J. G. Carbonell, M. Pinkal, H. Uszkoreit, M. Veloso, W. Wahlster, and M. J. Wooldridge, Eds. Springer-Verlag Berlin Heidelberg, 2009. [Online]. Available: http://www.springer.com/br/book/9783540732624

[9] E. R. Sparks, A. Talwalkar, D. Haas, M. J. Franklin, M. I. Jordan, and T. Kraska, "Automating model search for large scale machine learning," in *Proceedings of the Sixth ACM Symposium on Cloud Computing*. ACM, 2015, pp. 368–380.

[10] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and Robust Automated Machine Learning," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Montréal, Canada: NIPS, 2015, pp. 2944–2952. [Online]. Available: http://papers.nips.cc/paper/5872-efficient-and-robust-automated-machine-learning.pdf

[11] G. Luo, "A Review of Automatic Selection Methods for Machine Learning Algorithms and Hyper-parameter Values," Department of Biomedical Informatics, University of Utah, Salt Lake CIty, Tech. Rep., 2015.

[12] M. Fernández-Delgado, E. Cernadas, S. Barro, D. Amorim, and D. Amorim Fernández-Delgado, "Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?" *Journal of Machine Learning Research*, vol. 15, pp. 3133–3181, 2014. [Online]. Available: http://jmlr.org/papers/v15/delgado14a.html

[13] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

[14] N. Macià, E. Bernadó-Mansilla, A. Orriols-Puig, and T. K. Ho, "Learner excellence biased by data set selection: A case for data characterisation and artificial data sets," *Pattern Recognition*, vol. 46, no. 3, pp. 1054–1066, 2013.

[15] J. Vanschoren, H. Blockeel, B. Pfahringer, and G. Holmes, "Experiment databases," *Machine Learning*, vol. 87, no. 2, pp. 127–158, 2012.

[16] A. Kalousis, J. Gama, and M. Hilario, "On data and algorithms: Understanding inductive performance," *Machine Learning*, vol. 54, no. 3, pp. 275–312, 2004. [Online]. Available: https://link.springer.com/article/10.1023/b:mach.0000015882.38031.85

[17] R. Engels and C. Theusinger, "Using a Data Metric for Preprocessing Advice for Data Mining Applications," in *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, H. Prade, Ed. Chichester/London/New York, 1998.: John Wiley & Sons, Ltd, 1998, pp. 430–434. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.56.7414

[18] C. Thornton, F. Hutter, H. H. Hoos, K. Leyton-Brown, and K. L.-B. Chris Thornton, Frank Hutter, Holger H. Hoos, "Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, D. Inderjit S., Y. Koren, R. Ghani, T. E. Senator, P. Bradley, R. Parekh, J. He, U. Ramasamy, and R. L. Grossman, Eds. Chicago, Illinois, USA: ACM, NEw York, NY, USA, 2013, pp. 847–855. [Online]. Available: http://dl.acm.org/citation.cfm?id=2487629

[19] B. Pfahringer, H. Bensusan, and C. Giraud-Carrier, "Tell me who can learn you and i can tell you who you are: Landmarking various learning algorithms," in *Proceedings of the 17th international conference on machine learning*, 2000, pp. 743–750.

[20] J. Petrak, "Fast subsampling performance estimates for classification algorithm selection," in *Proceedings of the ECML-00 Workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*, 2000, pp. 3–14.

[21] C. Soares, J. Petrak, and P. Brazdil, "Sampling-based relative landmarks: Systematically test-driving algorithms before choosing," in *Portuguese Conference on Artificial Intelligence*. Springer, 2001, pp. 88–95.

[22] A. Kalousis, "Algorithm Selection via Meta-Learning," PhD Thesis, Université de Genève, 2002. [Online]. Available: http://cui.unige.ch/{~}kalousis/papers/metalearning/PhdThesisKalousis.pdf

[23] P. Brazdil and J. Gama, "Characterization of Classification Algorithms," *Progress in Artificial Intelligence Lecture Notes in Computer Science*, vol. 990, pp. 189–200, 1995.

[24] G. Lindner and R. Studer, "Ast: Support for algorithm selection with a cbr approach," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 1999, pp. 418–423.

[25] 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. [Online]. Available: http://ieeexplore.ieee.org/document/990132/

[26] L. Todorovski and S. Džeroski, "Experiments in meta-level learning with ilp," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 1999, pp. 98–106.

[27] D. Brain and G. I. Webb, "On The Effect of Data Set Size on Bias And Variance in Classification Learning," in *Proceedings of the Fourth Australian Knowledge Acquisition Workshop (AKAW '99)*, D. Richards, G. Beydoun, A. Hoffmann, and P. Compton, Eds. Syndey, Australia: The University of New South Wales, 1999, pp. 117–128. [Online]. Available: https://www.bibsonomy.org/bibtex/2eb55c4bdfb45c25cad6b1c613e9ef74f/giwebb

[28] A. Kalousis and M. Hilario, "Supervised knowledge discovery from incomplete data," *WIT Transactions on Information and Communication Technologies*, vol. 25, 2000.

[29] M. Banko and E. Brill, "Scaling to Very Very Large Corpora for Natural Language Disambiguation," in *ACL '01 Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, Toulouse, France, 2001, pp. 26–33. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1073012.1073017

[30] A. Halevy, P. Norvig, and F. Pereira, "The unreasonable effectiveness of data," *IEEE Intelligent Systems*, vol. 24, no. 2, pp. 8–12, 2009.

[31] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artificial Intelligence Review*, vol. 18, no. 2, pp. 77–95, 2002.

[32] H. Bensusan, C. G. Giraud-Carrier, and C. J. Kennedy, "A higher-order approach to meta-learning." *ILP Work-in-progress reports*, vol. 35, 2000.

[33] P. B. Brazdil, C. Soares, and J. P. Da Costa, "Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results," *Machine Learning*, vol. 50, no. 3, pp. 251–277, 2003.

[34] C. Salperwyck and V. Lemaire, "Learning with few examples: An empirical study on leading classifiers," in *Proceedings of the International Joint Conference on Neural Networks*. San Jose, CA, USA: IEEE, 2011, pp. 1010–1019. [Online]. Available: http://ieeexplore.ieee.org/document/6033333/

[35] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural computation*, vol. 4, no. 1, pp. 1–58, 1992.

[36] R. O. Duda, P. E. Hart, D. G. Stork *et al.*, *Pattern classification*. Wiley New York, 1973, vol. 2.

[37] G. E. a. P. a. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[38] N. Macia and E. Bernadó-Mansilla, "Towards uci+: A mindful repository design," *Information Sciences*, vol. 261, pp. 237–262, 2014.

[39] P. Flach, *Machine learning: the art and science of algorithms that make sense of data*. Cambridge University Press, 2012.