Université de Lille
Faculté des Sciences et Technologies

Impact of Meta Features on Supervised Classification

Selim Lakhdar

Master Informatique
Master in Computer Science

Département d'Informatique
Faculté des Sciences et Technologies

October, 2023

This report partially satisfies the requirements defined for the Project/Internship course, in the 3rd year, of the Master in Computer Science.

Candidate: Selim Lakhdar, No. 11705793,
selim.lakhdar.etu@univ-lille.fr

Scientific Guidance: Laetitia Jourdan, laetitia.jourdan@univ-lille.fr

Département d'Informatique
Faculté des Sciences et Technologies
Campus Cité Scientifique, Bât. M3 extension, 59655 Villeneuve-d'Ascq

October, 2023

Abstract

Supervised Classification is a sub-field of Supervised Learning, which is a set of algorithms that belong to Machine Learning techniques. They rely on labeled datasets, which are characterized by Meta-Features (MFs). Those MFs are a set of measurements derived from different concepts to give an insight into the representation and the complexity of the data. Those measurements are widely used in Machine Learning techniques to choose the appropriate approach that will outperform others. This paper exposes common Meta-Features used in the literature and their impact and usage for supervised classification by exposing the concept of Meta-Learning.

Keywords: Machine Learning, Supervised Learning, Classification, Meta-Features, Algorithm Selection, Feature Selection, Meta-Learning

Résumé

La classification supervisée est un sous-domaine de l'apprentissage supervisé, qui est un ensemble d'algorithmes appartenant aux techniques d'apprentissage automatique. Ils s'appuient sur des ensembles de données étiquetés, qui sont caractérisés par des méta-caractéristiques (MF). Ces MF sont un ensemble de mesures dérivées de différents concepts pour donner un aperçu de la représentation et de la complexité des données. Ces mesures sont largement utilisées dans les techniques d'apprentissage automatique pour choisir l'approche appropriée qui surpassera les autres. Cet article expose les méta-attributs courants utilisés dans la littérature ainsi que leur impact et leur utilisation pour la classification supervisée en exposant le concept de méta-learning.

Mots-clés: Machine Learning, Apprentissage Supervisé, Classification, Meta-Features, Algorithm Selection, Feature Selection, Meta-Learning

Contents

List of Figures

Chapter 1

Introduction

Machine Learning Machine Learning (ML) is a branch of artificial intelligence Artificial Intelligence (AI) that relies on different techniques to build a ML model that imitates the way that humans learn, gradually improving its accuracy. Classification is a sub-field of ML that relies on labelled datasets that are processed by algorithms to learn an underlying representation about the data that can be generalized to solve daily problems.

Labeled datasets can be characterized by Meta-Features (MFs). Those MFs are a set of measurements derived from different concepts to give an insight into the representation and the complexity of the data. This paper will present the use of those MF to guide or select the appropriate classification model.

The current work is organized as follows; Chapter 2 will introduce Machine Learning techniques (supervised, unsupervised and meta-learning) while exposing common ML limitations. In Chapter 3 we will expose different MFs used in the literature and their definition and usage. Finally, Chapter 4 will introduce some relevant articles that use MFs to select or guide classification problems.

Chapter 2

Machine Learning

Machine Learning is a field of Artificial Intelligence (IA). It relies on building models based on sample data, to make predictions or decisions without being explicitly programmed to do so. Those models are able to be improved automatically through experience [1].

Machine Learning algorithms usage is widespread. It can be found in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

Nasteski et al [2] organized machines learning algorithms into taxonomy, based on the desired outcome of the algorithm. It can be summed up by :

- Supervised learning: where the algorithm generates a function that maps inputs to desired outputs. One standard formulation of the supervised learning task is the classification problem: the learner is required to learn (to approximate the behavior of) a function which maps a vector into one of several classes by looking at several input-output examples of the function.
- Unsupervised learning: which models a set of inputs: labeled examples are not available.
- Semi-supervised learning: Combines both labeled and unlabeled examples to generate an appropriate function or classifier.

- Reinforcement learning: Algorithm learns a policy of how to act given an observation of the world. Every action has some impact in the environment, and the environment provides feedback that guides the learning algorithm.
- Transduction: Similar to supervised learning, but does not explicitly construct a function: instead, tries to predict new outputs based on training inputs, training outputs, and new inputs.
- Meta learning: where the algorithm learns its own inductive bias based on previous experience.

In a general way, Machine Learning algorithms can be divided into two general groups; supervised and unsupervised learning [2].

2.1 Unsupervised Learning

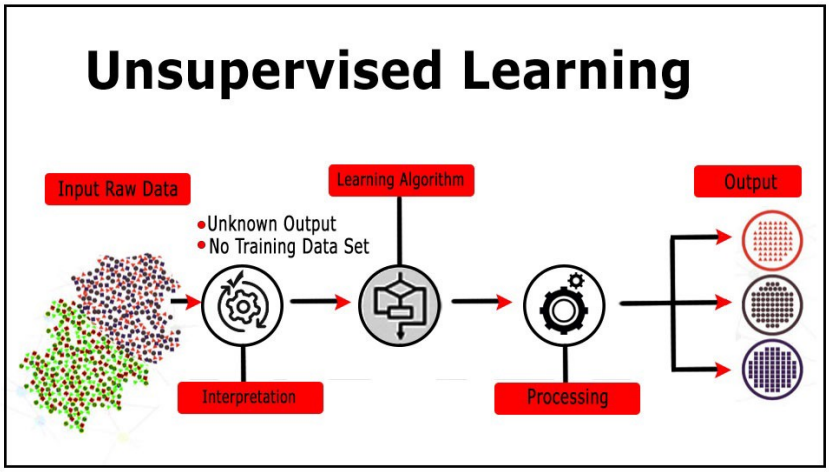

Figure 2.1: Unsupervised Learning

Unsupervised learning is used to find some underlying relations in an unlabeled dataset. It's employed to discover some common patterns to split the dataset into distinct groups [3]. It relies on searching for similarities between observations in order to determine if they can be categorized and create a group. These groups are named clusters [2]. Besides clustering usage, dimensionality reduction is another large subclass of unsupervised learning models. It's used to reduce large dataset features to a smaller one without losing information about the data.

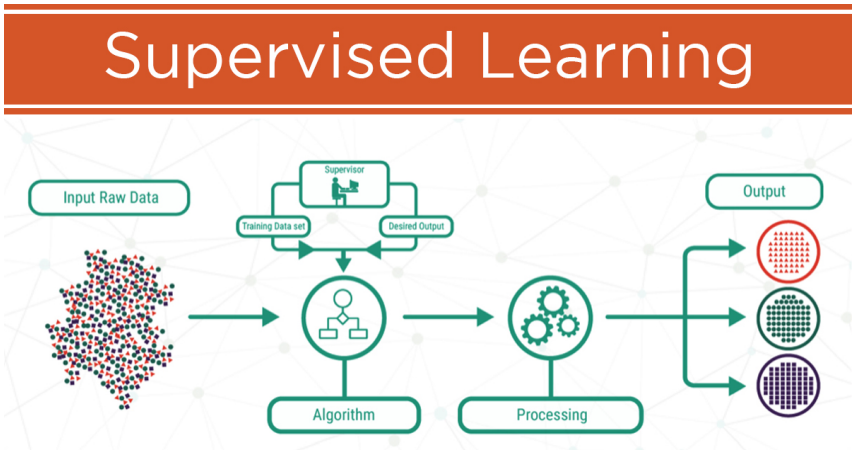

Figure 2.2: Supervised Learning

2.2 Supervised Learning

Supervised learning is a subset of algorithms used for acquiring the input-output relationship based on a given set of paired input-output training samples. An input-output training sample is called a labeled dataset. The goal of those algorithms is to learn the underlying mapping between the input and output and predict unseen observations [4].

In supervised algorithms, the classes are predetermined (ie: expected output). These classes are crafted by the user. The task of the machine learning algorithm is to find patterns and construct mathematical models. These models are then evaluated based on their predictive capacity in relation to measures of variance in the data itself [2].

Supervised learning is divided into two main supervised models; **Classification models (classifiers)** map the input space into predefined classes, **Regression models** map the input space into a real-value domain.

The potential benefits of progress in classification are immense since the technique has a great impact on other areas, both within Data Mining and in its applications [2].

2.3 Meta Learning

Meta-learning, also known as learning to learn, is the science of observing how different machine learning approaches perform on various learning tasks, and then learning from this experience to learn new tasks much faster than otherwise possible. Not only does this dramatically speed up and improve the design of machine learning pipelines, it also allows the replacement of hand-engineered algorithms with

novel approaches learned in a data-driven way [5].

Meta learning algorithms use metadata of learning algorithms as input. Then, they make predictions and provide information about the performance of these learning algorithms as their output. Those metadata are described as Landmarkers, which are exposed in section 3.5.

Meta-learning uses some common approaches like [6];

- Model Based: Model-based meta-learning models update their parameters rapidly with a few training steps.
- Metric Based: Metric-based meta-learning is similar to nearest neighbor algorithms, in which weight is generated by a kernel function. It aims to learn a metric or distance function over objects. The notion of a good metric is problem-dependent. It should represent the relationship between inputs in the task space and facilitate problem solving.
- Optimization Based: What optimization-based meta-learning algorithms intend to do is to adjust the optimization algorithm so that the model can be good at learning with a few examples.

2.4 Known Limitation

In this section we will discuss about some known limitation that machine learning faces.

2.4.1 No Free Lunch Theorem (NFL)

Historically, No Free Lunch (NFL) theorem is declined to two well-known theorems bearing the same name. One for supervised learning which was introduced by Wolpert in 1996 [7]. The second for search and optimization in 1997 introduced by Wolpert and Macready [8].

The theorem states that all optimization algorithms perform equally well when their performance is averaged across all possible problems. It implies that there is no single best machine learning algorithm for predictive modeling problems such as classification and regression (Figure 2.5.

NFL argues that, without having substantive information about the problem, like insights about the input data with strong assumptions, there is no reason to prefer one algorithm over another. The direct implication of this statement, is that

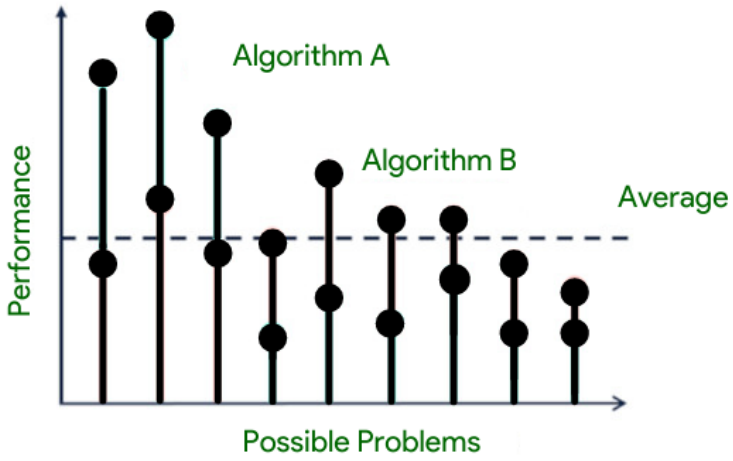

Figure 2.3: No Free Lunch Demonstration [9]

we need to explore different types of models to cover the wide variety of data that occurs in the real world [10].

2.4.2 Overfitting

Generalization of a Machine Learning Model against unseen data is what allows us to use ML algorithms for every day task. A common problem in Supervised Learning is that models could perfectly fit its training data which leads to the inability to generalize against unseen data, this phenomenon is called **overfitting** [11].

As we stated before Supervised Learning models rely on labelled data. They are trained on a large subset of the dataset (about 80%), then tested on what remains (about 20%). This approach gives us some insight about how well our models perform with some error metrics. Nevertheless, training models for a long time, or using complex models could lead to learning the "noise", or irrelevant information, within the dataset.

Contrary to Overfitting, Underfitting phenomenon is the fact that the model hasn't been trained for enough time or input features don't describe well the data which can't determine an underlying relationship between the input and output variables. As a result, underfitting also generalizes poorly to unseen data.

Low error rates and a high variance are good indicators of overfitting. In order to prevent this type of behavior, part of the training dataset is typically set aside as the "test set" to check for overfitting. If the training data has a low error rate and the test data has a high error rate, it signals overfitting. A generalization of

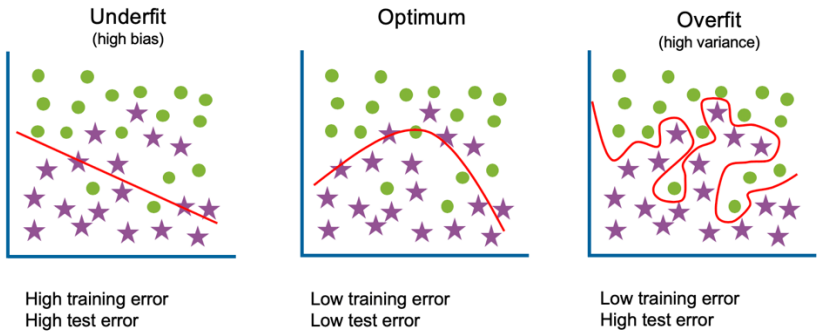

Figure 2.4: Overfitting vs Underfitting [12]

this approach is the K-fold cross-validation technique, which is the most popular to assess the accuracy of the model.

In K-folds cross-validation, data is split into k equally sized subsets, which are also called "folds." One of the k-folds will act as the test set, also known as the holdout set or validation set, and the remaining folds will train the model. This process repeats until each of the folds has acted as a holdout fold. After each evaluation, a score is retained, and when all iterations have been completed, the scores are averaged to assess the performance of the overall model.

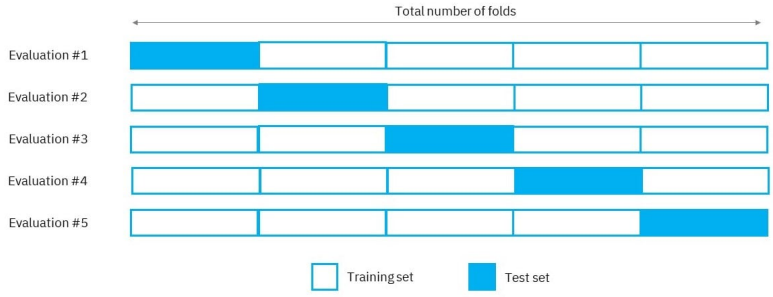

Figure 2.5: K-folds Cross–Validation (k=5) [12]

Common techniques used to mitigate this problem are;

- Early stopping: This method seeks to pause training before the model starts learning the noise within the model.
- Train with more data: Enlarge dataset with new observations.
- Data augmentation: Data augmentation makes a sample data look slightly different every time it is processed by the model. The process makes each data set appear unique to the model and prevents the model from learning the characteristics of the data sets. Another option that works in the same way as data augmentation is adding noise to the input and output data. Adding noise

to the input makes the model become stable, without affecting data quality and privacy, while adding noise to the output makes the data more diverse.

- Regularization: Regularization applies a "penalty" to the input parameters with the larger coefficients, which subsequently limits the amount of variance in the model. While there are a number of regularization methods, such as L1 regularization, Lasso regularization, and dropout, they all seek to identify and reduce the noise within the data.
- Ensemble methods: Ensemble learning methods are made up of a set of classifiers like decision trees—and their predictions are aggregated to identify the most popular result.

2.4.3 Data Imbalance

In classification, machine learning algorithms are trained on labeled datasets. The distribution of labels could cause problems when they are not well balanced. If observations are unequal, where a certain label is more present than another, we have to take into account that our dataset is imbalanced (Figure 2.6).

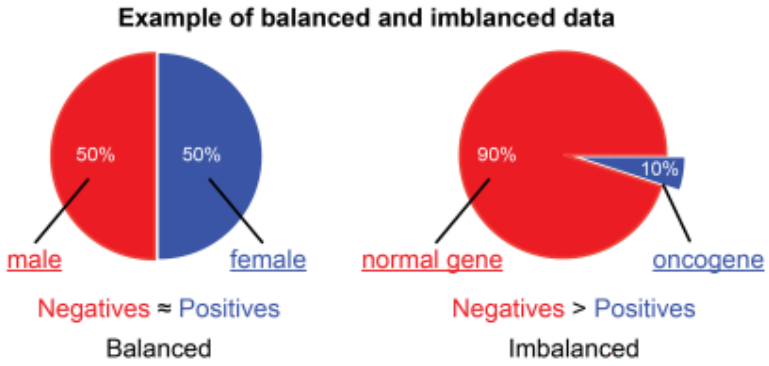

Figure 2.6: Dataset Imbalance [13]

Two known techniques are used to mitigate this problem (Figure 2.7).

Undersampling : The undersampling methods work with the majority class. In these methods, we randomly eliminate instances of the majority class. It reduces the number of observations from the majority class to make the dataset balanced. It results in a severe loss of information. This method is applicable when the dataset is huge and reducing the number of training samples makes the dataset balanced.

Oversampling : The Oversampling methods work with the minority class. In these methods, we duplicate random instances of the minority class. So, it replicates

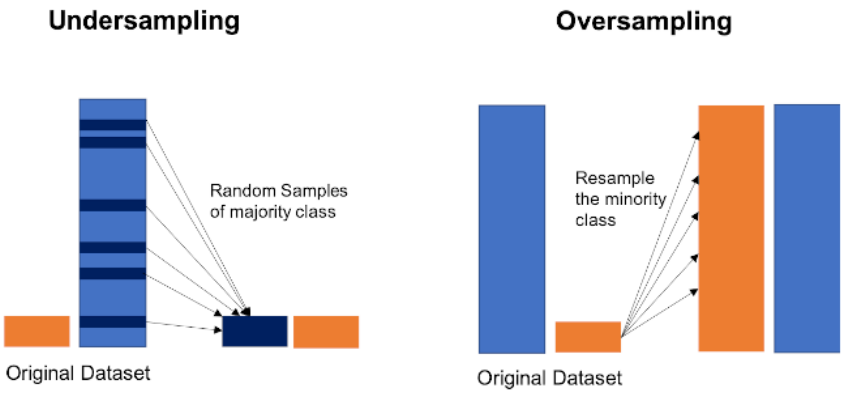

Figure 2.7: Under/Over Sampling [14]

the observations from the minority class to balance the data. It is also known as upsampling. It may result in overfitting due to duplication of data points.

Chapter 3

Dataset Meta Features

Meta Features (MFs) are a set of measurements that characterize a dataset. They are derived from different concepts to give an insight into the representation and the complexity of the data. Generally, they are categorized into five groups [15]:

- Simple/General Meta Features: Measures that are easily extracted from data. They do not require significant computational resources.
- Statistical Meta Features: Measures that capture the statistical properties of the data.
- Information Theoretic Meta Features: Measures from the information theory field.
- Model Based Meta Features: Measures extracted from a model induced using the training data.
- Landmarking Meta Features: Measures that use the performance of simple and fast learning algorithms to characterize datasets.

3.1 Simple/General Meta Features

Simple/General MFs are the most basic information about the dataset. They are directly derived from the data. They include [16];

- Number of attributes.

- Number of classes.
- Amount of observation.
- Amount of output values.
- Dataset dimensionality.

Acronym	Task	Argument	Domain	Range	Card.	Excep.
attrToInst	Any	∗P	Both	$[0, \overline{d}]$	1	No
catToNum	Any	∗P	Both	$[0, \overline{d}]$	1	Yes
classToAttr	Classif.	∗P+T	Both	$[0, q]$	1	No
freqClass	Classif.	T	Categ.	$[0, 1]$	q	No
instToAttr	Any	∗P	Both	$[0, \overline{n}]$	1	No
instToClass	Any	∗P+T	Both	$[1, \overline{n}]$	1	No
nrAttr	Any	∗P	Both	$[1, +\infty]$	1	No
nrAttrMissing	Any	∗P	Both	$[0, d]$	1	No
nrBin	Any	∗P	Both	$[0, d]$	1	No
nrCat	Any	∗P	Both	$[0, d]$	1	No
nrClass	Classif.	T	Categ.	$[2, \overline{n}]$	1	No
nrInst	Any	∗P	Both	$[q, +\infty]$	1	No
nrInstMissing	Any	∗P	Both	$[0, n]$	1	No
nrMissing	Any	∗P	Both	$[0, \overline{dn}]$	1	No
nrNum	Any	∗P	Both	$[0, d]$	1	No
numToCat	Any	∗P	Both	$[0, \overline{d}]$	1	Yes

Figure 3.1: Rivolli et al - Simple/General Meta Features Representation

Rivolli et al [17] presented an exhaustive representation of Simple MFs showed in Figure 3.1 (Associated parameters are described in Appendix A). Number of attributes (nrAttr); number of binary attributes (nrBin); number of categorical attributes (nrCat); number of numeric attributes (nrNum); proportion of categorical versus numeric attributes (catToNum) and vice-versa (numToCat) are measures related to attributes.

The measures attrToInst and instToAttr express the dimensionality and sparsity of the data. instToAttr can indicates overfitting when its value is too small.

The frequency of instances in each class (freqClass) allows the extraction of relevant measures like frequency of proportion of majority and minority class, default accuracy/error and standard deviation of the class distribution. Combined with summarization functions, it can describe imbalanced learning scenarios.

3.2 Statistical Meta Features

Statistical MFs describe the dataset by a statistical approach. They include [18]:

- Linear correlation coefficient: Shows the linear association strength between X and Y by means of a single value.
- Skewness : measures the lack of symmetry in the distribution of a random variable X.
- Kurtosis : Measures the peakedness in the distribution of a random variable X
- Standard Deviation: Estimates the dispersion of a random variable
- Variation: Evaluates the normalization of the standard deviation of a random variable X with respect to its mean value.
- Covariance: Extends the variance concept to the bidimensional case. It expresses the linear relationship between two random variables X and Y.

3.3 Information Theoretic Meta Features

Information Theoretic MFs are used to describe discrete (categorical) and continuous (numerical) attributes. It capture the amount of information in the data, they are directly computed, free of hyperparameter, deterministic and robust. Semantically, they describe the variability and redundancy of the predictive attributes to represent the classes [17]. It mainly relies on entropy which is a measure of the randomness in the variable. It includes [18];

- Normalized class entropy: Indicates how much information is necessary to specify one class.
- Normalized attribute entropy: Measures the information content related to the values that X may assume.
- Joint entropy of class and attribute: Measures the total entropy of the combined system of variables, i.e. the pair of variables *(C, X)*, which could be represented by a class variable and one of the m discretized input attributes, respectively
- Mutual information of class and attribute: measures the common information shared between two random variables C and X.
- Noise-signal ratio: measures the amount of irrelevant information contained in a dataset.

Acronym	Task	Argument	Range	Card.
attrEnt	Any	1P	$[0, \log_2(n)]$	d
classEnt	Classif.	T	$[0, \log_2(q)]$	1
eqNumAttr	Classif.	$*$P+T	$[0, \infty]$	1
jointEnt	Classif.	1P+T	$[0, \log_2(n)]$	d
mutInf	Classif.	1P+T	$[0, \log_2(n)]$	d
nsRatio	Classif.	$*$P+T	$[0, \infty]$	1

Figure 3.2: Rivolli et al - Information Theoretic MFs Representation

Rivolli et al [17] presented an exhaustive representation of Simple MFs showed in Figure 3.2.

The entropy of the predictive attributes (attrEnt) and the target values (classEnt) capture the average uncertainty present in the predictive and class attributes.

- attrEnt can provide an overview of the attributes' capacity for class discrimination.
- classEnt represents how much information, on average, is necessary to specify one class.

The joint entropy (jointEnt) and the mutual information (mutInf) compute the relationship of each attribute with the target values.

- jointEnt captures the relative importance of the predictive attributes to represent the target.
- mutInf represents the common information shared between them, indicating their degree of dependency.

Number of attributes (eqNumAttr) and the noise signal ratio (nsRatio) capture information that is related to the minimum number of attributes necessary to represent the target attribute and the proportion of data that are irrelevant to describe the problem.

3.4 Model Based Meta Features

Model Based MFs are information extracted from a predictive learning model. Decision Trees models are the most used. Indeed, they can express the complexity of the dataset by the number of leaves, the number of nodes and the shape of the tree.

Acronym	Task	Argument	Range	Card.
leaves	Sup.	∗P+T	$[q, \overline{n}]$	1
leavesBranch	Sup.	∗P+T	$[1, \overline{n}]$	$\overline{n}$
leavesCorrob	Sup.	∗P+T	$[0, 1]$	$\overline{n}$
leavesHomo	Sup.	∗P+T	$[q, +\infty]$	$\overline{n}$
leavesPerClass	Classif.	∗P+T	$[0, 1]$	q
snodes	Sup.	∗P+T	$[q, \overline{n}]$	1
nodesPerAttr	Sup.	∗P+T	$[0, \overline{n}]$	1
nodesPerInst	Sup.	∗P+T	$[0, 1]$	1
nodesPerLevel	Sup.	∗P+T	$[1, \overline{n}]$	$\overline{n}$
nodesRepeated	Sup.	∗P+T	$[0, \overline{n}]$	$\overline{d}$
treeDepth	Sup.	∗P+T	$[1, \overline{n}]$	$\overline{n}$
treeImbalance	Sup.	∗P+T	$[0, 1]$	$\overline{n}$
treeShape	Sup.	∗P+T	$[0.0, 0.5]$	$\overline{n}$
varImportance	Sup.	∗P+T	$[0, 1]$	$\overline{d}$

Figure 3.3: Rivolli et al - Model Based MFs Representation

Rivolli et al [17] presented an exhaustive representation of Simple MFs showed in Figure 3.3.

- Measures based on leaves describe, in some degree, the complexity of the orthogonal decision surface.
  - leavesBranch: measures the number of distinct paths (maximum value limited by the number of instances).
  - leavesCorrob: measures the proportion of training instances to the leaf (fixed range, dataset independant).
  - leavesHomo: measures the distribution of the leaves in the tree (no limit of values).
  - leavesPerClass: represents the classes complexity and the result is summarized per class (fixed range, dataset independant).
- Measures based on nodes extract information about the balance of the tree to describe the discriminatory power of attributes.
  - nodesPerAttr: Express the proportion of nodes per attribute.
  - nodesPerInst: Express the proportion of nodes per instance.
  - nodesPerLevel: Express the number of nodes per level.
  - nodesRepeated: Express the number of repeated nodes.
  - nodesRepeated: Express the number of repeated nodes.
- Measures based on tree size extract information about the leaves and nodes to describe the data complexity.
  - treeDepth: Represents the depth of each node and leaf.
  - treeImbalance: Express the degree of imbalance in the tree.
  - treeShape: represents the entropy of the probabilities to randomly reach a specific leaf in a tree from each one of the nodes.
  - varImportance: Represents the amount of information present in the attributes before a node split operation.

3.5 Landmarking Meta Features

Landmarking is a technique for characterizing datasets by using the performance of rapid and simple learners. It differs from model-based meta-features, which extract data from learning models [17].

Balte et al [16] presented two main criteria for meta-feature landmarking;

- Efficiency: Landmarker should be inexpensive. That's mean, if heavy computation is required to obtain landmark, brute force approach leads to a similar performance.
- Bias Diversity: The good landmarker are consisted of landmark with different prejudice and both have similar performance measure on all data sets then it would be sufficient to choose any one.

They also presented an exhaustive landmarking representation which gives the idea of what type of relations are present in landmarking. Those relations can be described as:

- Absolute (LM): Conventional strategy in landmarking. Used to directly estimate accuracy of landmark algorithms.
- Ranks (RK): It is possible that one attribute corresponds to more than one landmark. It's used as performance rank among competitors unlike accuracy score.
- Order (OR): Each possible rank contains at least one attribute and its value is the landmark that obtains that rank.
- Pair wise (RL): Pair-wise comparison between accuracy of the landmarkers. For each pair of landmarkers such relation returns +1 when first value is bigger, -1 if second value is bigger and for missing value? Otherwise.
- Ratios (RT): This is the generalization of previous one. For all pairs of landmark, the pair wise accuracy ration is encoded.

Landmarker can be resumed to [16];

- Average Node Learner: Calculates the average accuracy of single node decision tree.
- Best Node Learner: Based on information gain ratio, it shows how informative is an attribute with respect to classification task using its entropy. It chooses attribute which have highest information gain.
- Worst Node Learner: In this the information gain criteria uses the attribute which represents lowest selected value.
- One Nearest Node Learner: This landmark learner classifies how near the test point that belongs to same class are.
- Elite 1-Nearest Neighbor Learner: This subset is comprised of the most informative attributes, if information gain ratio among attribute is smaller than 0.1. Otherwise the elite subset is singleton and learner acts like a decision node learner.

- Randomly Choose Node Learner: This result is based on randomly choose attribute. This node is used to split the training set and classifies given test examples.
- Naïve Bayes Learner: Training set uses bayes theorem to classify test cases.

Chapter 4

Impact of Meta Features on Supervised Classification

In this section we will expose some relevant works that used Meta Features to enhance classification accuracy. Majority of articles are relying on Meta-Learning approaches introduced in section 2.3 and the usage of meta-features exposed in Chapter 3.

Golshanrad et al [19] proposed an efficient assembling method that employs both meta-learning and a genetic algorithm for the selection of the best classifiers. Their method is called MEGA, standing for using MEta-learning and a Genetic Algorithm for algorithm recommendation. Figure 4.1 expose an overview of MEGA and the main steps executed by its three components. The Training component generates a model from meta-features by applying a genetic algorithm on input datasets. The Model Interpretation component interprets the generated model using a multi-label decision tree and an a priori algorithm. The Testing component recommends a combination of classifiers for an unseen dataset using kNN algorithm. MEGA does not only achieve superior classification results compared to existing methods, but is also able to generate novel, human-interpretable classification rules. This "interpretability feature" makes MEGA particularly valuable for applications in domains (e.g., in the medical sector) that require explainable classification results, be it for legal or ethical reasons. Apart from this, MEGA is unique in that it additionally ensures diversity in the ensemble system and variability of the meta-features as well as the

individual classifier.

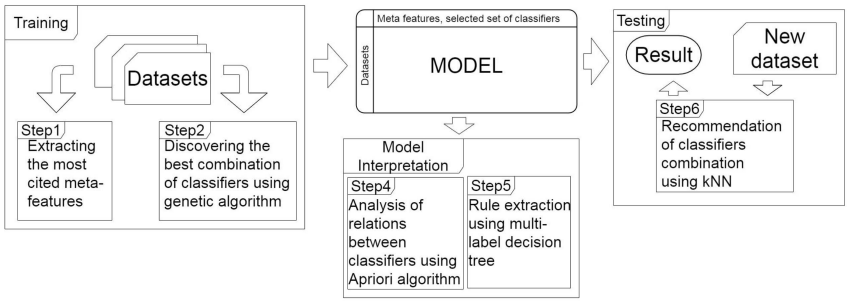

Figure 4.1: Golshanrad et al - overview of MEGA [19]

Parmezan et al [20] addressed the metalearning challenge of recommending feature selection algorithms by proposing a novel meta-feature engineering model. The model considers a broad collection of meta-features that enable the study of the relationship between the dataset properties and the feature selection algorithm performance in terms of several criteria. A set of new meta-features was also proposed in this study. Their work made use of the proposed meta-feature engineering model to recommend feature selection algorithms based on information, distance, dependency, consistency, and precision measures. An overview of the results indicated that the data characterization primarily via measures of simple, landmarking, image-based, and graph-based categories is promising for the problem of automatic choice of feature selection algorithms.

Pise et al [21] proposed an approach for solving algorithm selection problem using meta-learning. The paper uses three types of data characteristics. Simple, information theoretic, and statistical data characteristics are used. Results are generated using nine different algorithms on thirty eight benchmark datasets from UCI repository. The proposed approach uses K-nearest neighbor algorithm for suggesting the suitable algorithm. Classifier accuracy is taken as a basis for recommending the algorithm. By using meta-learning, accurate method can be recommended as per the given data, and cognitive overload for applying each method, comparing with other methods and then selecting the suitable method for use can be reduced. Thus it helps in adaptive learning methods. The experimentation shows that predicted accuracy are matching with the actual accuracy for more than 90% of the benchmark datasets used. Thus it is concluded that the number of attributes, the number of instances, the number of classes, maximum probability of class and class entropy are playing a major role in classifier accuracy and algorithm selection for thirty eight datasets used for experimentation.

Reif et al [15] empirically evaluate five different categories of state-of-the-art meta-features for their suitability in predicting classification accuracies of several

widely used classifiers (including Support Vector Machines, Neural Networks, Random Forests, Decision Trees, and Logistic Regression). Based on the evaluation results, they developed the first open source meta-learning system that is capable of accurately predicting accuracies of target classifiers. The user provides a dataset as input and gets an automatically created high-performance ready-to-use pattern recognition system in a few simple steps.

Lee et al [22] revisited 26 meta-features typically used in the context of meta-learning for model selection. Using visual analysis and computational complexity considerations, they found 4 meta-features whose values are directly relevant to certain ranges of predictive accuracy for 7 learning algorithms on 135 UCI datasets. According to the experimental results, mean correlation coefficient of attributes, median entropy of attributes, mean 2 of attributes, and mutual information between attributes and target class turn out to be meta-features that are directly relevant with high performance of our seven test-bed learning algorithms. With those meta-features, we are able to formulate converted meta-features to boost meta-learning performance.

Gopal et al [23] proposes an alternative approach, to multilabel classification, by transforming conventional representations of instances and categories into a relatively small set of link-based meta-level features, and leveraging successful learning-to-rank retrieval algorithms (e.g., SVM-MAP) over this reduced feature space.

Chapter 5

Conclusion

Meta-Features are measurements used to characterize a dataset. Generally, they are categorized into five groups; Simple/General Meta Features, Statistical Meta Features, Information Theoretic Meta Features, Model Based Meta Features, Landmarking Meta Features. The last one employs different ML techniques that are rapid and simple to give an insight into how well they perform.

In this paper, we introduced Machine Learning and its techniques like the Supervised / Unsupervised Learning, and Meta-Learning. We exposed common limitations to those techniques like the No Free Lunch Therom (NFL), which states that all ML algorithms perform equally well when their performance is averaged across all possible problems. It implies that there is no single best machine learning algorithm for predictive modeling problems unless we have strong assumptions about the data. This theorem consolidates the fact that MFs could bring new directions in Machine Learning modeling.

Finally, we show off some relevant articles that deal with the usage of MFs to guide or select an appropriate ML classification model. These papers demonstrate the impact of MFs in the process of model selection and model performance. They leverage the importance of MF in Machine Learning field in general.

References

[1] "Geeksforgeeks.org." https://www.geeksforgeeks.org/what-is-no-free-lunch-theorem/, Accessed August 2022). [Cited on pages vii and 7]

[2] "Ibm overfitting." https://www.ibm.com/cloud/learn/overfitting, Accessed August 2022. [Cited on pages vii and 8]

[3] "Data imbalance." https://medium.com/analytics-vidhya/what-is-balance-and-imbalance-dataset-89e8d7f46bc5, Accessed August 2022. [Cited on pages vii and 9]

[4] "Under/over sampling." https://www.datascience2000.in/2021/06/how-to-handle-imbalanced-dataset.html, Accessed August 2022. [Cited on pages vii and 10]

[5] P. Golshanrad, H. Rahmani, B. Karimian, F. Karimkhani, and G. Weiss, "Mega: Predicting the best classifier combination using meta-learning and a genetic algorithm," *Intelligent Data Analysis*, vol. 25, no. 6, pp. 1547–1563, 2021. [Cited on pages vii, 19, and 20]

[6] A. Rivolli, L. P. Garcia, C. Soares, J. Vanschoren, and A. C. de Carvalho, "Characterizing classification datasets: a study of meta-features for meta-learning," *arXiv preprint arXiv:1808.10406*, 2018. [Cited on pages vii, 12, 13, 14, 16, and 27]

[7] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015. [Cited on page 3]

[8] V. Nasteski, "An overview of the supervised machine learning methods," *Horizons. b*, vol. 4, pp. 51–62, 2017. [Cited on pages 3, 4, and 5]

[9] Z. Ghahramani, "Unsupervised learning," in *Summer school on machine learning*, pp. 72–112, Springer, 2003. [Cited on page 4]

[10] Q. Liu and Y. Wu, "Supervised learning," 01 2012. [Cited on page 5]

[11] J. Vanschoren, "Meta-learning," in *Automated machine learning*, pp. 35–61, Springer, Cham, 2019. [Cited on page 6]

[12] L. Weng, "Meta-learning: Learning to learn fast," *lilianweng.github.io*, 2018. [Cited on page 6]

[13] D. H. Wolpert, "The lack of a priori distinctions between learning algorithms," *Neural computation*, vol. 8, no. 7, pp. 1341–1390, 1996. [Cited on page 6]

[14] 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. [Cited on page 6]

[15] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012. [Cited on page 7]

[16] T. Dietterich, "Overfitting and undercomputing in machine learning," *ACM computing surveys (CSUR)*, vol. 27, no. 3, pp. 326–327, 1995. [Cited on page 7]

[17] 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. [Cited on pages 11 and 20]

[18] A. Balte, N. Pise, and P. Kulkarni, "Meta-learning with landmarking: A survey," *International Journal of Computer Applications*, vol. 105, no. 8, 2014. [Cited on pages 11, 16, and 17]

[19] C. Castiello, G. Castellano, and A. M. Fanelli, "Meta-data: Characterization of input features for meta-learning," in *International Conference on Modeling Decisions for Artificial Intelligence*, pp. 457–468, Springer, 2005. [Cited on page 13]

[20] A. R. S. Parmezan, H. D. Lee, N. Spolaôr, and F. C. Wu, "Automatic recommendation of feature selection algorithms based on dataset characteristics," *Expert Systems with Applications*, vol. 185, p. 115589, 2021. [Cited on page 20]

[21] N. Pise and P. Kulkarni, "Algorithm selection for classification problems," in *2016 SAI Computing Conference (SAI)*, pp. 203–211, IEEE, 2016. [Cited on page 20]

[22] J. won Lee and C. Giraud-Carrier, "Predicting algorithm accuracy with a small set of effective meta-features," in *2008 Seventh International Conference on Machine Learning and Applications*, pp. 808–812, IEEE, 2008. [Cited on page 21]

[23] S. Gopal and Y. Yang, "Multilabel classification with meta-level features," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pp. 315–322, 2010. [Cited on page 21]

Appendix A

Categories used to describe a measure or group of measures

Level	Category Name	Options
Input	Task	Classification
		Supervised
		Any
	Extraction	Direct
		Indirect
	Argument	n Predictive Attribute (nP)
		All Predictive Attributes (∗P)
		Target Attribute (T)
	Domain	Numerical
		Categorical
		Both
	Hyperparameters	Yes, No
Output	Range	[min, max]
	Cardinality	k
	Deterministic	Yes, No
	Exception	Yes, No

Figure A.1: Rivolli et al - Categories used to describe a measure or group of measures [17]