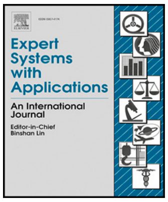
Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Review

Feature weighting methods: A review

Iratxe Niño-Adan [a,*], Diana Manjarres [a], Itziar Landa-Torres [b], Eva Portillo [c]

[a] TECNALIA, Basque Research and Technology Alliance (BRTA), 48160 Derio, Spain
[b] Petronor Innovación S.L., 48550 Muskiz, Spain
[c] Department of Automatic Control and Systems Engineering, Faculty of Engineering, University of the Basque Country, UPV/EHU, 48013 Bilbao, Spain

ARTICLE INFO

Keywords:
Feature weighting
Feature importance
Feature relevance
Review

ABSTRACT

In the last decades, a wide portfolio of Feature Weighting (FW) methods have been proposed in the literature. Their main potential is the capability to transform the features in order to contribute to the Machine Learning (ML) algorithm metric proportionally to their estimated relevance for inferring the output pattern. Nevertheless, the extensive number of FW related works makes difficult to do a scientific study in this field of knowledge. Therefore, in this paper a global taxonomy for FW methods is proposed by focusing on: (1) the learning approach (supervised or unsupervised), (2) the methodology used to calculate the weights (global or local), and (3) the feedback obtained from the ML algorithm when estimating the weights (filter or wrapper). Among the different taxonomy levels, an extensive review of the state-of-the-art is presented, followed by some considerations and guide points for the FW strategies selection regarding significant aspects of real-world data analysis problems. Finally, a summary of conclusions and challenges in the FW field is briefly outlined.

1. Introduction

Machine Learning (ML) algorithms are widely employed to successfully extract patterns and valuable information from data (Bishop et al., 1995). Nevertheless, their performance is highly dependent on the quality of the given dataset. If it contains irrelevant or noisy information, reliable knowledge cannot be easily extracted (García, Luengo, & Herrera, 2015). Consequently, data preprocessing, which transforms the raw data into a useful and understandable format, is a relevant stage in ML algorithms (García, Luengo, & Herrera, 2016).

In the same context, the selection of the representative features that best define the output behaviour is an important task, which is frequently done with the aid of expert knowledge on the application field and/or by Feature Selection methods (Jović, Brkić, & Bogunović, 2015; Li et al., 2018; Saeys, Inza, & Larrañaga, 2007; Venkatesh & Anuradha, 2019). Traditionally, it has been assumed that all the selected features are equally important when estimating the output. However, if some features present higher scale than others, the results can be over-influenced by them, affecting the performance and the accuracy of the overall algorithm (Daszykowski, Kaczmarek, Vander Heyden, & Walczak, 2007). In order to minimise such dominance, normalisation methods (Jain, Nandakumar, & Ross, 2005; Milligan & Cooper, 1988; Panday, de Amorim, & Lane, 2018) are usually employed in order to equalise the contribution of each feature on the algorithm metric (Aksoy & Haralick, 2001).

Nevertheless, it is widely known that all the features are usually not equally representative of the hidden pattern, especially in real-world problems. In this context, in the last decades a wide portfolio of Feature Weighting (FW) methods have been proposed with the aim of estimating the degree of relevance that each feature has for extracting the output pattern (Wei, Lu, & Song, 2015). Their main goal is to transform or weigh the features to contribute to the ML algorithm metric proportionally to their estimated relevance.

In this paper an extensive review of the state-of-the-art on FW methods is presented. Moreover, this paper depicts a classification of them focusing on: (1) the learning strategy followed (i.e. supervised or unsupervised), (2) the methodology used to calculate the weights (i.e. global or local), and (3) the feedback obtained from the ML algorithm when estimating the weights (i.e. filter or wrapper). In Section 2 the proposed taxonomy for their classification is shown. Following this taxonomy, Section 3 collects the FW methods proposed in the literature in the last years. Section 4 provides some considerations and guide points for the selection of optimal FW methods based on significant aspects to consider in real-world data analysis problems. To conclude this review, Section 5 gathers a summary of conclusions and challenges that remain unsolved in the FW field.

* Corresponding author.
E-mail addresses: iratxe.nino@tecnalia.com (I. Niño-Adan), diana.manjarres@tecnalia.com (D. Manjarres), itziar.landa@repsol.com (I. Landa-Torres), eva.portillo@ehu.eus (E. Portillo).

https://doi.org/10.1016/j.eswa.2021.115424
Received 11 May 2020; Received in revised form 15 September 2020; Accepted 10 June 2021
Available online 17 June 2021

2. Taxonomy

FW methods are techniques that, given a dataset $X \in \mathbb{R}^{n \times m}$ composed by n samples described by m features, obtain a set of weights W^* representing the relative relevance of the features of X. The obtained weights w^*_{ij} for $i = \{1, \ldots, n\}$ and $j = \{1, \ldots, m\}$, generally ranging from 0 to 1 in such a way that $\sum_{i,j} w_{ij} = 1$, multiply each value x_{ij} of X in order to create a weighted dataset $\widetilde{X}$ representative of the relative importance each feature has for the given system. In this sense, the higher the weight value, the higher the relative importance of the corresponding feature. Besides, when $w_{ij} = 0$ the feature weight acts as a Feature Selection factor, discarding non-relevant information from the dataset. The weighted dataset $\widetilde{X}$ is ultimately used by the ML algorithm to model the system.

From this basis, this section describes the taxonomy regarding the FW methods. At a first stage, the FW methods are classified based on the learning approach employed to estimate the weights, namely supervised or unsupervised. The supervised FW strategy refers to the methods that employ the information of the real labels Y to calculate the features weights. By contrast, unsupervised FW is considered when there is no information about the real labels. Instead, the feature weights are calculated considering other intrinsic characteristics of the dataset, such as distance between samples of the features or respect to a given point. In general, unsupervised FW methods employ clustering algorithms to extract group structures from the dataset and utilise this information to compute the weights.

At a second stage, the FW methods are classified based on the way the weights are applied: global (i.e. over the entire instance space) or local (i.e. over different parts of the instance space). Global FW approaches consider that the feature has the same relevance for calculating the output for the whole target population. For each feature X_j, a single global weight w_j is calculated, i.e. $\forall i \; w_{ij} = w_j$. Local weights are employed when it is assumed that a given feature presents different degrees of relevance, depending on its samples or subsets of them, for estimating the output. Thus, more than one weight are assigned to the same feature. In this case, the FW method obtains weights w_{gj}, with $g \in \{1, \ldots, G\}$ and $1 < G \leq n$, where G corresponds to the number of weights assigned to each feature and which is generally equal to the number of classes (supervised) or clusters (unsupervised).

Finally, the proposed taxonomy delves into the way the estimation strategy is made. In particular, filter and wrapper methods are distinguished. As Fig. 1 depicts, filter FW methods calculate the (global or local) feature weights as the relationship between the features and a given reference which corresponds to, based on the learning approach selected, the real labels Y in the supervised case or intrinsic characteristics of the data in the unsupervised one.

In contrast, as shown in Fig. 2, wrapper methods employ feedback from a given ML algorithm to estimate the feature weights (global or local) in a black-box iterative fashion. Thus, based on the performance obtained in the previous iteration calculated by supervised or unsupervised evaluation metrics, the method decides whether to adjust the weights or not in order to improve the model performance in the next iteration.

In order to provide a comprehensive overview of the paper, Fig. 3 depicts the proposed taxonomy of the FW methods, achieved from the combination of the different approaches per level, which is the basis of the next section schema.

3. Feature weighting methods

In this section an extensive review of FW methods is presented following the taxonomy proposed in Section 2. Besides, pseudo-algorithms are included to describe the main process of each group of FW methods. Note that, in the pseudo-algorithms, the learning approach (supervised/unsupervised) is remarked with underline text, the way the weights are applied (global/local) with dotted box and the estimation strategy (filter/wrapper) with dashed box. For the sake of our knowledge, there is no other review work in the literature that covers such a wide classification of FW methods, i.e. the encountered reviews are focused on a particular taxonomy level or on a subset of it, such as: FW for K-NN algorithms (Wettschereck & Aha, 1995; Wettschereck, Aha, & Mohri, 1997), FW for the K-means algorithm (de Amorim, 2016) or FW for different clustering methods but employing the wrapper methodology (Deng, Choi, Jiang, Wang, & Wang, 2016).

In this work special attention will be paid to particular characteristics and objectives of the problem at hand: (1) Labels' availability when is possible to obtain the labels. (2) High-dimensional dataset characterised by a high number of features compared to the number of observations. (3) Dimensionality reduction for those problems that require the reduction of the number of features due to high computational cost or other similar reasons. (4) Dataset understanding for those problems in which the main interest is to provide the domain expert with information about the influence of each feature on the output from the system/application's perspective. (5) Features contribution on the model for those problems in which the main objective is to infer the influence of every feature on the performance of the ML based model. (6) Missing values commonly caused by improper data collection or data acquisition fails. (7) Imbalanced dataset when the dataset has an unequal distribution of classes. (8) Outliers when the dataset contains observations that significantly deviate from most observations. (9) Noise when meaningless or corrupted features are introduced in the dataset. (10) Interpretability for those problems in which the main objective is to extract knowledge from the ML based model from the system/application's perspective. (11) Condition-based problems in which the relevance of the features varies depending on the operating condition of the system (for instance, type of material or material thickness). (12) Temporal dependency commonly found when time series are required to deal with the modelling task of interest. (13) Algorithm performance maximisation for those problems in which the main objective is to optimise the ML algorithm performance regardless the interpretability of the model. (14) Semi-supervised learning if the dataset comprises both labelled and unlabelled samples. (15) Online learning for those systems whose properties change along the time.

3.1. *Supervised feature weighting*

This section presents the works in the literature focused on supervised feature weighting. As described previously, it takes advantage of the real labels Y to estimate the feature weights. Depending on the number of weights calculated per feature, global or local approaches can be distinguished.

3.1.1. *Global supervised feature weighting*

Global feature weighting methods look for the optimal weight w^*_j to be assigned to each feature X_j based on its relevance for estimating the output pattern. Since supervised learning strategy is considered, this relevance is computed as the relationship between the feature and the labels. The calculation of the global weights can be done in a filter or a wrapper approach.

Filter global supervised feature weighting. In the case of filter global supervised FW methods, the weights estimation is done by the employment of techniques on the field of Variable Importance Analysis (V.I.A.), also called Sensitivity analysis (Christopher Frey & Patil, 2002; Wei et al., 2015). As presented in Algorithm 1, the global weights w^*_j are calculated in line 3 utilising V.I.A methods to represent the importance of each feature X_j in terms of its relationship with the label Y. Once calculated the weights, these multiply each feature of the dataset (line 5) creating the weighted dataset $\widetilde{X}$ which is ultimately passed to the ML algorithm (line 6). These works Christopher Frey and Patil (2002), Wei et al. (2015) present an excellent introduction and description of the basis and terminology associated to Variable Importance Analysis.

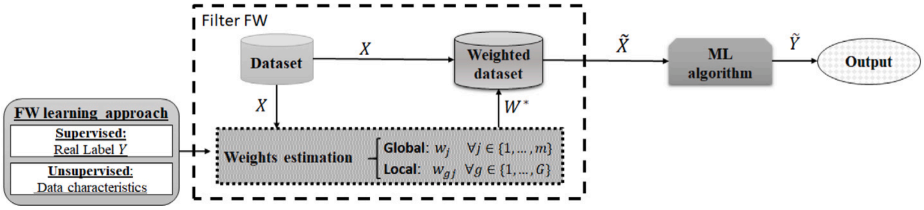


Fig. 1. Flow chart of filter FW approaches.

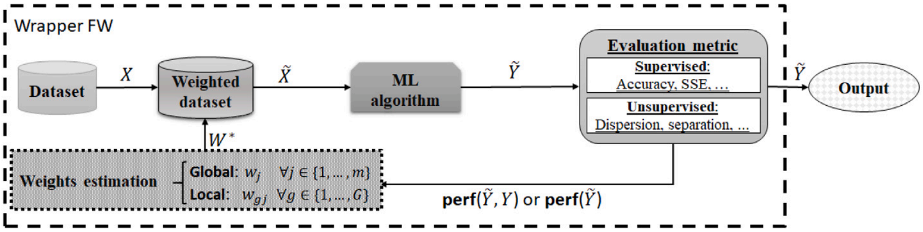


Fig. 2. Flow chart of wrapper FW approaches.

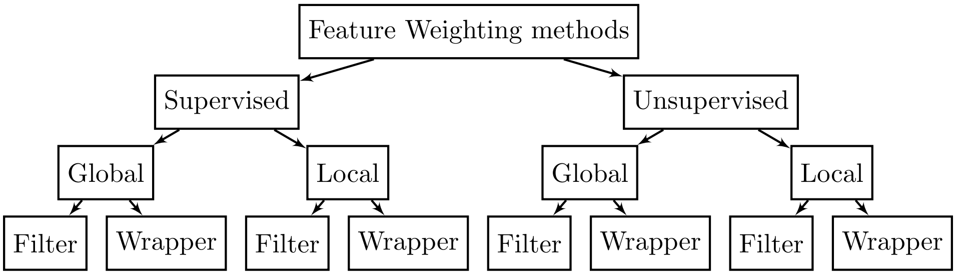


Fig. 3. Proposed taxonomy of the FW methods.

Algorithm 1 Filter Global Supervised Feature Weighting methods

```
1: Given a dataset $X \subset \mathbb{R}^{n \times m}$ and the labels $Y$
2: for $j = 1 : m$ do
3:     $w_j^* \leftarrow V.I.A.(X_j, Y)$
4: for j=1:m do
5:     $\widetilde{X}_j \leftarrow w_j \cdot X_j$
6: $\widetilde{Y} \leftarrow$ ML algorithm to $\widetilde{X}$
```

In this context, two main approaches can be found to perform the Variable Importance Analysis (V.I.A.) in line 3 of Algorithm 1: one based on Information theory, and another one related to Statistical-based methods.

Regarding the former, a commonly used V.I.A. measure employed for weight estimation is Mutual Information (MI) defined as

$$I(X_j, Y) = \sum_{x_{ij} \in X_j} \sum_{y_i \in Y} P(x_{ij}, y_i) \cdot \log\left(\frac{p(x_{ij}, y_i)}{p(x_{ij})p(y_i)}\right) \tag{1}$$

where $p(\cdot)$ is the probability distribution function and $p(X_j, Y)$ the conditional probability distribution function of X_j and Y. Several works apply Eq. (1) to estimate the weights in line 3 of Algorithm 1 (García-Laencina, Sancho-Gómez, Figueiras-Vidal, & Verleysen, 2009; Wettschereck & Dietterich, 1995). For instance, the Linear Feature Weighted SVM (LFWSVM) algorithm (Xing, Ha, Hu, & Tian, 2009), the MI-MCS-FWSVM method (Giveki, Salimi, Bahmanyar, & Khademian, 2012), the Correlation-based Feature Weighting (CFW) filter method (Jiang, Zhang, Li, & Wu, 2018) and the proposal Hussain (2019) estimate the features relevance respect to the label Y and calculate the weights employing the MI measure in line 3 of Algorithm 1. Similarly to the MI measure, the Information Gain (IG) and the Regularised Entropy (re) measures are employed in Chen and Hao (2017), Wu, Gu, and Gu (2017), respectively.

From the variable importance analysis field, the employment of Statistical-based methods to estimate the feature relevance is another extended approach. In Sahin, Ipbuker, and Kavzoglu (2015) χ^2 and Fisher (F-score) algorithms are employed to measure in line 3 of Algorithm 1 the quality of the features respect to the label. Likewise, Granger causality (Granger, 1988) and AHP (Saaty, 2014) are applied in Bhattacharya, Ghosh, and Chowdhury (2017) to improve the performance of the K-NN algorithm (line 6 of Algorithm 1).

Heretofore, all the presented FW methods estimate the weights for each feature in an independent manner. However, as shown in Elbasiony, Sallam, Eltobely, and Fahmy (2013), Niño-Adan, Landa-Torres, Portillo, and Manjarres (2019), methods that estimate the weights in a group manner, i.e. using the Random Forest (RF) algorithm, are also utilised as filter FW methods.

Although in most cases the main goal of the filter global supervised FW methods is to increase the accuracy of the model, some authors make use of the filter FW approaches for other purposes. This is the case of the proposed Fast Feature Weight algorithm for Data Gravitation Classification model (FFW-DGC) (Peng, Zhang, Zhang, & Yang, 2017) which aims at *reducing the computational complexity* of the FW process in the DGC model. In this work the feature weights (line 3 of Algorithm 1) are calculated combining two fuzzy sets: (1) one relative to the feature discrimination capability, computed by means of the MI (Eq. (1)) between the discretised features and the labels, and (2) the second one related to the redundancy between features, calculated by Pearson correlation as the covariance of X_j and Y between their standard deviations:

$$\rho(X_j, Y) = \frac{COV(X_j, Y)}{\sigma_{X_j}\sigma_Y} \tag{2}$$

Finally, due to the presence of non-independent features in real problems, another common use of filter FW methods is to estimate the feature weights in order to alleviate the conditional independence assumption of the Naive Bayes (NB) algorithm (line 6 of Algorithm 1). In this context, the approach (Zhang, Jiang, Li, & Kong, 2016) adapts two filter FW approaches for NB classifier in the field of text classification. The first approach employs in line 3 of Algorithm 1 Gain Ratio (GR) (Zhang & Sheng, 2004) assuming that the features take zero or nonzero values, while the second one obtains the feature weights (line 3 of Algorithm 1) from a Decision Tree (DT) classifier (Hall, 2007).

Wrapper global supervised feature weighting. Regarding the wrapper FW approach, as described in Algorithm 2 the weights w_j^* are first initialised with random values (line 2 of Algorithm 2) and then, adjusted (line 10 of Algorithm 2) in an iterative fashion (line 4 of Algorithm 2) considering the relationship between the estimated output $\widetilde{Y}$ and the real label Y in terms of a given supervised evaluation metric that measures the performance of the ML algorithm respect to the given label, i.e. $\textbf{perf}(\widetilde{Y}, Y)$. The iterative process ends after a predefined number of iterations *n_iter*, when the performance outperforms a given threshold θ or when the wrapper method converges, i.e. it does not present significant improvement respect to the previous iteration in terms of the performance measure (line 8 of Algorithm 2).

Algorithm 2 Wrapper Global Supervised Feature Weighting methods

```
1:  Given a dataset $X \subset \mathbb{R}^{n \times m}$ and $\underline{\textbf{the labels } Y}$
2:  Initialise $\boxed{w_j^{(0)} \in \mathbb{R}}$
3:  $s \leftarrow 0$
4:  while $s < n\_iter$ do
5:      for j=1:m do
6:          $\widetilde{X}_j^{(s)} \leftarrow w_j^{(s)} \cdot X_j$
7:      $\widetilde{Y}^{(s)} \leftarrow$ ML algorithm to $\widetilde{X}^{(s)}$
8:      if $\underline{\textbf{perf}(\widetilde{Y}^{(s)}, Y)} > \theta$ or $=\underline{\textbf{perf}(\widetilde{Y}^{(s-1)}, Y)}+\epsilon$ then
9:          Break
10:     $w_j^{(s+1)} \leftarrow$Adjust $w_j^{(s)}$
11:     $s \leftarrow s + 1$
12: $\boxed{w_j^* \leftarrow w_j^{(s)}}$
```

The most extended practice to configure the wrapper strategy is the employment of a Genetic Algorithm (GA) for the weights calculation in order to improve the performance of the model in terms of accuracy (Komosiński & Krawiec, 2000; Phan, Le Nguyen, & Bui, 2017).

The flexibility for encoding the individuals in the GA algorithm offers multiple possibilities. For instance, the work Ahn and Kim (2009) proposes a Global Optimisation of Case-Based Reasoning (CBR), a hybrid model that employs a GA algorithm to simultaneously optimise the feature weights (line 10 of Algorithm 2), the instance selection and the K value for the K-NN algorithm based on the classification accuracy (line 8 of Algorithm 2). The authors employ the Wisconsin breast cancer image dataset from UCI repository in order to validate their method and conclude that the simultaneous optimisation of the feature weights, the instance selection and the K value obtains the highest accuracy (line 8 of Algorithm 2) for this problem.

Apart from the GA algorithm, other Evolutionary Algorithms (EA) are also employed for Feature Weighting (lines 2–11 of Algorithm 2). For instance, Mateos-García, García-Gutiérrez, and Riquelme-Santos (2017) present the Simultaneous Weighting of Attributes and Neighbours (SWAN) that employs an EA for adjusting the contribution of the neighbours and the significance of the features that minimise the cross-validation error (line 8 of Algorithm 2). Similarly, the work Triguero, Derrac, GarcíA, and Herrera (2012) proposes a self-adaptive Differential Evolution (DE) algorithm in order to optimise the feature weights (line 12 of Algorithm 2) that maximise the performance of the K-NN for prototype generation (line 8 of Algorithm 2). In Sotoodeh, Moosavi, and Boostani (2019) a Particle Swarm Optimisation (PSO) algorithm is employed to generate the optimum feature weight vector (line 12 of Algorithm 2) in terms of image retrieval system performance. In the same context, the research Serrano-Silva, Villuendas-Rey, and Yáñez-Márquez (2018) presents a performance comparison of DE, GA and Novel Bath Algorithm (NBA) for FW in terms of AUC and execution time over several financial datasets.

Nevertheless, other approaches, such as the Dynamic Representation and the Neighbour Sparse Reconstruction-based Relief (DRNSR-Relief) presented in Huang, Zhang, et al. (2018), decompose the nonlinear problem into locally linear problems. Specifically, the proposed algorithm represents the dynamic relationship between the margin and the weight vectors. In this proposal, the Gradient Ascent method is employed to calculate the expected margin vector and to update the feature weights in line 10 of Algorithm 2 in a wrapper fashion. Similarly, Yang, Wang, and Zuo (2012) employ the Gradient Ascent update method to estimate the feature weights that maximises the expected leave-one-out classification accuracy (line 8 of Algorithm 2). The same approach is applied by Raghu and Sriraam (2018) for the classification of EEG signals and by Romeo et al. (2020) for the prediction of heterogeneous machine parameters in Industry 4.0. Finally, Ouyed and Allili (2020) present a Multinomial Kernel Logistic Regression with Group of Features Relevance (GFR-MKLR) approach for human interaction recognition. The authors include into the kernel and the loss function the gestures' weights, estimated during the training phase utilising as wrapper FW approach the Newton–Raphson optimisation method.

3.1.2. Local supervised feature weighting

Local FW methods aim at obtaining more than one optimal weight per feature, since it is assumed that the degree of relevance of each feature depends on the sample (or subsets of samples). Similarly to global feature weighting methods, the weights can be estimated in a filter or a wrapper approach.

Filter local supervised feature weighting. In contrast to global FW methods, very few works introduce a new filter approach for estimating local weights in a supervised fashion. In fact, most of them adapt the global methods to the local environment. In this case, as Algorithm 3 shows, the filter local supervised FW methods estimate per feature as many weights as classes C recorded in Y, and the weights in line 4 of Algorithm 3 are computed according to the distribution of the samples x_{ij} into the different classes. After the weights calculation, in line 7 of Algorithm 3 each sample of the dataset is multiplied by the corresponding weight. The weighted dataset $\widetilde{X}$ is then employed by the ML algorithm (line 8) to ultimately estimate the output $\widetilde{Y}$.

Algorithm 3 Filter Local Supervised Feature Weighting methods

```
1: Given a dataset $X \subset \mathbb{R}^{n \times m}$ and $\underline{\text{the labels } Y \in \{1, \dots, C\}^n}$
2: for $\boxed{c = 1 : C}$ do
3:     for $j = 1 : m$ do
4:         $\boxed{w_{cj}^*} \leftarrow \boxed{\text{Distribution}(\{x_{ij} | y_i = c\})}$
5: for j=1:m do
6:     for $\boxed{c = 1 : C}$ do
7:         $\widetilde{X}_{i'j} \leftarrow \boxed{w_{cj}^*} \cdot X_{i'j}$ $\forall i' \in \underline{\{i | y_i = c\}}$
8: $\widetilde{Y} \leftarrow$ ML algorithm to $\widetilde{X}$
```

For instance, Marchiori (2013) proposes a decomposition of the well-known RELIEF online method, which processes the samples in a serial way one-by-one, into class dependent feature weights vectors (line 2 of Algorithm 3), in which each vector describes the relevance of features conditioned to each class (line 4 of Algorithm 3). Each class-dependent term generates a weighted distance by enlarging the sample margin of the corresponding class. Experiments conducted over two breast cancer datasets from UCI repository show that the accuracy obtained by the proposed decomposition is similar or superior than the reached one by the traditional RELIEF method. A similar work can be found in Yilmaz, Yazici, and Kitsuregawa (2014) in which a RELIEF-based modality weighting approach, named RELIEF-MM, is proposed for fusing multimodal information in multimedia data. Here, the authors convert the original RELIEF-f algorithm into a class-specific representation. The final values of the modality weights are obtained by grouping the training examples according to the classes and processing samples of each class separately (line 4 of Algorithm 3).

In regards to the adaptations of commonly known filter methods, Chen and Guo (2015) reformulate the Simple Matching Coefficient (SMC) (line 8 of Algorithm 3) and propose the Weighted SMC distance (WSMD) (line 4 of Algorithm 3) aiming at including a supervised measurement of the contribution of the categorical features (Entropy or Gini diversity index) for a global and a local approach. Experiments over real-world datasets support that the local weighting approach outperforms the accuracies obtained by the global proposal in *high dimensional datasets*.

Wrapper local supervised feature weighting. Similar to the global approaches, as Algorithm 4 presents, the local wrapper supervised FW methods adjust the randomly initialised (line 1 of Algorithm 4) weights $w_{cj}^{(s)}$ in an iterative fashion (line 11 of Algorithm 4) in order to maximise the performance of a given ML algorithm. In the local case, there are as many weights per feature as classes C and each sample of the dataset x_{ij} is multiplied by the local weight associated to the corresponding sample label y_i (line 7 of Algorithm 4). Once the iterative process ends (line 4 of Algorithm 4), the performance improves a given threshold θ or the performance converges (line 9 of Algorithm 4), the lastly adjusted weights are selected as the optimal ones (line 13 of Algorithm 4).

Algorithm 4 Wrapper Local Supervised Feature Weighting methods

```
 1: Given a dataset $X \subset \mathbb{R}^{n\times m}$ and $\underline{\textbf{the labels } Y \in \{1,\dots,C\}^n}$
 2: Initialise $\boxed{w_{cj}^{(0)} \in \mathbb{R} \ \ \forall c = 1 : C}$, $\forall j = 1 : m$
 3: $s \leftarrow 0$
 4: while $s < n\_iter$ do
 5:     for j=1:m do
 6:         for $\boxed{c = 1 : C}$ do
 7:             $\widetilde{X}_{i'j}^{(s)} \leftarrow \boxed{w_{cj}^{(s)}} \cdot X_{i'j} \ \forall i' \in \underline{\{i|y_i = c\}}$
 8:     $\widetilde{Y}^{(s)} \leftarrow$ ML algorithm to $\widetilde{X}^{(s)}$
 9:     if $\underline{\textbf{perf}(\widetilde{Y}^{(s)}, Y)} > \theta$ or $= \underline{\textbf{perf}(\widetilde{Y}^{(s-1)}, Y)} + \epsilon$ then
10:         Break
11:     $w_{cj}^{(s+1)} \leftarrow$Adjust $w_{cj}^{(s)}$
12:     $s \leftarrow s + 1$
13: $\boxed{w_{cj}^{*} \leftarrow w_{cj}^{(s)}}$
```

Regarding wrapper local supervised approaches, Paredes and Vidal (2000) introduce the Class-Dependent Weighted (CDW) dissimilarity measure for NN classification (line 9 of Algorithm 4). The weights (line 11 of Algorithm 4) are obtained through Fractional-Programming Gradient Descent (FPGD)-based minimisation of the ratio between intra-class and inter-class distances. Similarly, Paredes and Vidal (2006) reformulate the objective function as the sigmoid function of the ratios between intra-class and inter-class distances to emphasise the importance of the prototypes that are close to the boundaries. In line with this research, authors in Ren, Wu, Sun, and Wu (2019) propose the Learning of Reduced Prototypes and Local Metric (LRPLM) that simultaneously learns a set of prototypes and optimal local feature-wise metric to minimise in line 9 of Algorithm 4 the image set classification error probability.

However, Jiao, Pan, and Feng (2015) claim that the distance metric proposed by Paredes and Vidal (2000); Paredes and Vidal (2006) is not sufficient for characterising the particularities of the different classes in the feature space. Thus, the Evidential K-Nearest Neighbour classification method with Weighted attributes (WEK-NN) method (Jiao, Pan, Feng, & Yang, 2013) is extended to propose a more general distance metric, named Class-Conditional Weighted (CCW), which is related to both the class labels of the prototypes and the query patterns. The idea of the CCW metric is employed few years later by the same authors in Jiao, Geng, and Pan (2019). In this case, a new KNN-based classifier, called BPkNN, is developed based on pairwise distance metrics and Belief function theory. Instead of learning a global distance metric, it is decomposed into a group of pairwise distance metrics and K-NN (PkNN) sub-classifiers are adaptively designed to separate the classes.

Another related approach can be found in Taheri, Yearwood, Mammadov, and Seifollahi (2014) where the authors propose a novel Attribute Weighting method for a feature weighted NB classifier (AWNB), in which for each feature, a different weight per class is calculated. An objective function based on the structure of the NB for binary classification problems (line 8 of Algorithm 4) is modelled to optimise the feature weights by means of the quasisecant method. Similarly, Jiang, Zhang, Yu, and Wang (2019), based on Weighting attributes to Alleviate Naive Bayes' Independence Assumption (WANBIA) (Zaidi, Cerquides, Carman, & Webb, 2013), introduce the Class-specific Attribute Weighted Naive Bayes (CAWNB) model, in which local weights are included into the conditional probability calculations of the Naive Bayes to consider the relative importance of the jth feature for the given class. Then, based on the objective function, the authors present two CAWNB-based proposals to improve the prediction performance: CAWNB$^{\text{CLL}}$ that maximises in line 9 of Algorithm 4 the Conditional Log Likelihood (CLL) and CAWNB$^{\text{MSE}}$ which minimises (line 9 of Algorithm 4) the Mean Square Error (MSE).

Evolutionary algorithms are also employed to estimate the local weights in the wrapper supervised approach. For instance, Mohemmed and Zhang (2008) analyse the performance of the Particle Swarm Optimisation (PSO) employing different distance measures: the Euclidean, the class-dependent Mahalanobis and the CDW proposed in Paredes and Vidal (2006) for reducing the classification error of the Nearest Centroid Classifier (NCC) (lines 8 and 9 of Algorithm 4, respectively). The authors conclude that the PSO based NCC (PSO-NCC) approach performs well in this particular classification task. Furthermore, the k-Labels Dependent Evolutionary Distance Weighting (kLDEDW) presented in Mateos-García, García-Gutiérrez, and Riquelme-Santos (2012) employs the Differential Evolution (DE) algorithm to optimise the local weights (line 11 of Algorithm 4) and the optimal value of K for the K-NN algorithm. The analysis conducted by the authors confirms that the proposed approach outperforms, in terms of accuracy, five classification algorithms including two above-mentioned local weighting methods: CDW based on Gradient Descent (AlSukker, Khushaba, & Al-Ani, 2010; Paredes & Vidal, 2006) which employs DE to estimate the weights. More recently, the work of Sinciya and Celin (2017) improves the performance of the Data Gravitation Classification (DGC) algorithm (line 8 of Algorithm 4) for high dimensional data by the employment of a PSO-based local feature weighting optimisation.

3.1.3. *Concluding remarks about supervised approaches*

Regarding supervised FW methods, it can be highlighted that filter global approaches are the most commonly employed ones in the literature. In this context, two approaches, namely Information theory and Statistical-based, can be further distinguished. Most of the Information-based FW approaches need to know in advance the probability distribution of the features, which in real world continuous problems is hard to obtain. To circumvent this, a normal distribution of the continuous variables can be assumed, but this is not always realistic. Therefore, discretisation techniques are commonly applied with the consequent loss of information. On the other hand, the Statistical-based FW methods compute the relation between the features and the label by means of statistical measures. Consequently, the selection of the statistical tests is of vital importance for these approaches.

In contrast, wrapper methods do not have such above-mentioned limitations and the flexibility in the problem configuration enables its application for local FW strategies. In this regard, the selection of the ML algorithm, its configuration and encoding is crucial for the proper performance of the method and for not getting stuck in a local optimum. However, there is a lack of an extensive empirical evaluation of wrapper methods and, in particular, a comparison between filter and wrapper methods for the same problem. As wrapper methods employ feedback from a ML algorithm to evaluate alternatives based on an external validation measure, they are widely recognised to obtain *better results than filter approaches*. Nevertheless, due to the iterative search process, wrapper methods require a moderate complexity which results in a *high computational cost*, especially with more exhaustive search strategies or in *high-dimensional datasets*. Besides, the weights resultant from wrapper approaches are high dependent on the algorithm configuration and their contribution to the ML algorithm metric can be higher than their real relevance for estimating the output. In this sense, the filter approaches, which calculate the weights separately for each feature without the influence of the specific configuration of the algorithm, enable an *interpretation of the feature relevance* in terms of the employed Information theory or Statistical-based technique.

3.2. Unsupervised feature weighting

In this section, a review about FW methods framed within the unsupervised approach is presented. As there is no information about the real labels, the weight estimation is performed in most cases by means of certain characteristics and relations of the features or by the obtained cluster structure. These characteristics are generally calculated by means of unsupervised evaluation metrics. The works of Palacio-Niño and Berzal (2019) and Rendón, Abundez, Arizmendi, and Quiroz (2011) present an excellent introduction and description of the basis and terminology associated to clustering algorithms.

3.2.1. Global unsupervised feature weighting

Global FW methods calculate a weight w_j per feature X_j, $j = \{1, \dots, m\}$. However, since the unsupervised learning strategy does not consider Y to estimate the features relevance, in these cases the relative importance of each feature is estimated according to certain clustering structure obtained from the dataset.

Filter global unsupervised feature weighting. After analysing in detail the filter global unsupervised related literature, it must be highlighted that very few works follow a filter approach for estimating the global weights in an unsupervised fashion. Algorithm 5 describes the general procedure of this approach. Generally, first a clustering algorithm is applied to group the data samples into different partitions (line 2 of Algorithm 5), and then, as shown in line 4 of Algorithm 5, based on the obtained cluster structure the feature weights are calculated as the relation between the features and the extracted structures. Then, each weight multiplies the corresponding feature of the dataset (line 6) and the obtained $\widetilde{X}$ is passed in line 6 of Algorithm 5 to the ML algorithm.

Algorithm 5 Filter Global Unsupervised Feature Weighting methods

```
1: Given a dataset $X \subset \mathbb{R}^{n \times m}$ and $\underline{\textbf{a clustering algorithm}}$
2: $\underline{centroids}$ ← Clustering algorithm to $X$
3: for $j = 1 : m$ do
4:     $\boxed{w_j^*} \leftarrow \boxed{\text{Relation}(X_j, \underline{centroids_j})}$
5: for j=1:m do
6:     $\widetilde{X}_j \leftarrow \boxed{w_j^*} \cdot X_j$
7: $\widetilde{Y}$ ← ML algorithm to $\widetilde{X}$
```

An example can be found in Gürüler (2017) where the authors propose a novel hybrid algorithm, entitled KMCFW-CVANN, that combines the K-Means Clustering algorithm (line 2 of Algorithm 5) for Feature Weighting and a Complex-Valued Artificial Neural Network (line 7 of Algorithm 5) for Parkinson disease diagnosis. The feature weights in line 4 of Algorithm 5 are calculated as the ratios of the means of the features to the centroids. For the classification phase the CVANN (line 7 of Algorithm 5) is applied. The results of the experiments show that the KMCFW-CVANN reaches the highest diagnosis accuracy in comparison with the other diagnosis approaches from the literature (Polat, 2012; Sakar & Kursun, 2010) over the Parkinson dataset (Little, McSharry, Roberts, Costello, & Moroz, 2007).

Similarly, Güneş, Polat, and Yosunkaya (2010) present the K-Means Clustering based Feature Weighting (KMCFW) method. Their proposal first extracts frequency domain features for which the mean, minimum, maximum and standard deviation are computed as statistical features. In the second stage, (line 2 of Algorithm 5) the features are clustered by the K-means algorithm and the ratios of means of the features with respect to the obtained centroids are employed as feature weights (line 4 of Algorithm 5).

Wrapper global unsupervised feature weighting. The wrapper approach is the most applied procedure when considering unsupervised global FW problems. In this context, the weights are commonly embedded into the clustering objective function and both the cluster structure and the feature weights are iteratively obtained. Thus, as presented in Algorithm 6, for each feature X_j a weight w_j is associated. The weights are included into the ML algorithm (line 7 of Algorithm 6) by multiplying the features of the dataset (line 6 of Algorithm 6) and depending on the performance obtained by the ML algorithm (line 9 of Algorithm 6) the weights are adjusted.

Algorithm 6 Wrapper Global Unsupervised Feature Weighting methods

```
1: Given a dataset $X \subset \mathbb{R}^{n \times m}$
2: Initialise $\boxed{w_j^{(0)} \in \mathbb{R}}$
3: $s \leftarrow 0$
4: while $s < n\_iter$ do
5:     for j=1:m do
6:         $\widetilde{X}_j^{(s)} \leftarrow \boxed{w_j^{(s)}} \cdot X_j$
7:     $\widetilde{Y}^{(s)}$ ← ML algorithm to $\widetilde{X}^{(s)}$
8:     $w_j^{(s+1)}$ ←Adjust $w_j^{(s)}$
9:     if $\underline{\textbf{perf}(\widetilde{Y}^{(s)})}> \theta$ or $=\underline{\textbf{perf}(\widetilde{Y}^{(s-1)})}+\epsilon$ then
10:        Break
11:    $s \leftarrow s + 1$
```

One of the first works that follows a wrapper global unsupervised FW approach is Huang, Ng, Rong, and Li (2005), in which the authors adapt the K-means algorithm by means of including the feature weights into the formula giving place to the Weighted K-Means (W-k-means). In this work, partial optimisation is employed to iteratively obtain the samples membership for the different clusters (line 7 of Algorithm 6), to estimate the new centroids and finally, to recalculate the weights(line 8 of Algorithm 6) based on the current partition of the data. In order to obtain compact clusters, it considers the intra-cluster dispersion of the different clusters for the calculation of the weights in line 8 of Algorithm 6 as:

$$w_j = \frac{1}{\sum_{t \in H} \left[D_j / D_t \right]^{\frac{1}{\beta - 1}}} \tag{3}$$

where D_j is the sum of the intra-cluster dispersion of the different K clusters in the feature j and H the set of features with $D_j \neq 0$. As it can be observed, w_j is dependent on a user-defined parameter β. Thus, the authors analyse the range of values that can be assigned to β, concluding that it is recommended to take values of $\beta < 0$ or $\beta > 1$. Experiments over synthetic and real datasets validate the proposal and conclude that the W-k-means can effectively distinguish between noisy and normal features.

Since the W-k-means, several authors have proposed a new version of it for different applications. Thus, in Chen, Yin, Tu, and Zhang (2009) the W-k-means is adapted to text clustering, while Hung, Chang, and Lee (2011) employ a variant of it for colour image segmentation. Similarly, the proposal presented in Saha and Das (2015) adapts the W-k-means to the fuzzy K-modes algorithm (line 7 of Algorithm 6) in order to handle categorical data. In this case, the W-k-means distance and the degree of membership of the samples in the different clusters take values in $\{0, 1\}$. In the case of the Minkowski W-k-means (MW-k-means) approach proposed by De Amorim and Mirkin (2012), the Minkowski metric is employed as distance function in the K-means algorithm (line 7 of Algorithm 6) and the β parameter is fixed equal to the one utilised in the Minkowski distance and learned in a *semi-supervised* manner. Additionally, the authors propose the anomalous cluster step to intelligently set K and initialise the centroids. The proposal of De Amorim and Mirkin (2012) is employed later in Panday et al. (2018) where the authors utilise feature weighting as an unsupervised *feature selection* tool.

Another adaptation is presented in Chakraborty and Das (2018) where the authors propose a version of Huang et al. (2005) for the Gaussian Means algorithm which also estimates the optimal number of clusters in the iterative process. The proposed algorithm starts with a single cluster containing all the samples. At the beginning of each iteration the objective function of the W-k-means is calculated (line 7 of Algorithm 6) and the Anderson–Darling normality test is computed for the current centroids. If a cluster is not Gaussian, the centroid is replaced by two new centroids as described in Hamerly and Elkan (2004). This iterative process ends when the objective function improvement is lower than a predefined threshold (line 9 of Algorithm 6).

Following with the same line of research, the Feature Weight Self-Adjustment (FWSA) mechanism presented in Tsai and Chiu (2008), removes the β parameter in the objective function. The optimisation process is similar to Huang et al. (2005), but the weights are updated at each iteration (s) in line 8 of Algorithm 6 by adding an adjustment margin in the following manner:

$$w_j^{(s+1)} = w_j^{(s)} + \frac{b_j^{(s)}/a_j^{(s)}}{\sum_{j=1}^{m} b_j^{(s)}/a_j^{(s)}} \tag{4}$$

where, for the jth feature, a_j, b_j are the sum of separations inter and intra clusters, respectively. Therefore, this proposal not only searches for compactness but also considers the separation between the clusters. The authors analyse the performance of their proposal over synthetic and real-world datasets. They also compare the FWSA against the W-k-means in terms of SSE, Entropy, ARI and number of iterations until convergence. Although FWSA is *computationally more expensive*, it outperforms the W-k-means. In addition, the authors remark that the proposed algorithm do not need user-defined parameters as β.

Unsupervised wrapper FW methods have been also proposed to handle different kind of data. In Benkabou, Benabdeslem, and Canitia (2018) the Detection of Outlier Time Series (DOTS) adapts the K-medians for time series clustering by minimising an Entropy and Dynamic Time Warping (DTW) based objective function (line 7 of Algorithm 6). In this proposal, the weights are assigned to each time series and the DTW between the time series and the medians of the clusters is employed to compute the distance in line 7 of Algorithm 6. In addition, the penalised entropy of the weights is added to the objective function in order to control the distribution of the weights. The optimisation process is done in a partial manner, such as in Huang et al. (2005), but the weights in line 8 of Algorithm 6 are estimated as:

$$w_j = \frac{e^{-D_j/\lambda}}{\sum_{t=1}^{m} e^{-D_t/\lambda}} \tag{5}$$

where λ is the regularisation term for the weight entropy. The idea is to assign small weights to *time series* that increase the intra-cluster distances and therefore, are considered as outliers. Several experimental comparisons over temporal data from datasets collected from the UCR repository are conducted to evaluate the performance of the proposed approach in contrast to *outlier* detection algorithms, such as: DTW+KM (Budalakoti, Srivastava, & Otey, 2008), DTW+HC (Portnoy, 2000), DTW+Spectral (Ng, Jordan, & Weiss, 2002), DL-OCSVM (Bevilacqua & Tsaftaris, 2015), FD-OCSVM (Schölkopf, Williamson, Smola, Shawe-Taylor, & Platt, 2000) DTW+LOF (Breunig, Kriegel, Ng, & Sander, 2000) and Parzen-Window (Yeung & Chow, 2002).

Finally, unsupervised global FW methods for multi-view clustering can be found in the literature. Authors Wang and Chen (2017) propose a multi-view fuzzy clustering approach in which the weights are updated in line 8 of Algorithm 6 based on the weighted distance of the data points of each view and the corresponding centroid. Similarly, Zhang, Wang, Huang, Zheng, and Zhou (2018) introduce the Two-level Weighted Collaborative k-means (TW-CO-k-means) approach for analysing data from multiple sources or views, while satisfying the consistency across different views and the diversity within each view. The views and the features in each view are assigned with weights that reflect their importance (line 6 of Algorithm 6). The two-level methodology employs an objective function in line 7 of Algorithm 6 that considers the feature weighted intra-cluster dispersion at each weighted view, a penalty term that measures the disagreement across multiple views and two weight-entropy-based terms that adjust the distribution of the weights.

3.2.2. *Local unsupervised feature weighting*

Local unsupervised feature weighting methods estimate more than one weight w_{ij} per feature X_j. Filter approaches take into consideration the samples relation with a given reference obtaining for each sample a different weight, while wrapper approaches utilise clusters structures to adjust the weights and obtain as many weights per feature as number of extracted clusters.

Filter local unsupervised feature weighting. Similar to global unsupervised FW, the filter approach is not a commonly employed technique for estimating the weights and few works are encountered in the literature. Most of works are related to *time series* data in order to include information about their temporal behaviour. In these cases, as Algorithm 7 shows, generally for each sample $\mathbf{x}_i$ with $i = \{1, \ldots, n\}$ a local weight is estimated in line 3 of Algorithm 7, aiming at measuring common information shared with a given sample of reference $\mathbf{z}$. Once the weights have been calculated, these in line 5 of Algorithm 7 multiply each sample of the dataset, creating the weighted dataset $\widetilde{X}$ which is finally utilised by the ML algorithm for the problem modelling (line 6).

Algorithm 7 Filter Local Unsupervised Feature Weighting methods

```
1: Given $X \subset \mathbb{R}^{n\times m}$ and a reference point z
2: for $\boxed{i = 1 : n}$ do
3:     $\boxed{\mathbf{w}_i^*} \leftarrow \boxed{commonality(\mathbf{x_i}, \mathbf{z})}$
4: for $\boxed{i = 1 : n}$ do
5:     $\widetilde{X}_i \leftarrow \boxed{w_i^*} \cdot X_i$
6: $\widetilde{Y} \leftarrow$ ML algorithm to $\widetilde{X}$
```

In this context, the exponential fading function, a monotonic decreasing function that decays with the time, estimates the weights of the samples in line 3 of Algorithm 7 as a function of time respect to the moment of interest ($\mathbf{z}$ in line 3 of Algorithm 7), assigning higher weight values to recent samples. This weighting function includes a decay rate λ which controls the importance of the historical information. The lower the value of λ, the higher the importance of the past data compared to more recent information. This filter local unsupervised weighting method is widely employed in temporal applications aiming at reducing the influence of the past. For instance, authors in Aggarwal, Han, Wang, and Yu (2004) include the mentioned fading function in their proposed High-dimensional projected data stream clustering method (HPStream) (line 6 of Algorithm 7) and authors in Ding and Li (2005), aiming at assigning higher importance to recent data to the recommendation process of their collaborative filtering algorithm (line 6 of Algorithm 7), include the estimated weights in the preference prediction phase.

Conversely, Jeong, Jeong, and Omitaomu (2011) propose the Weighted Dynamic Time Warping (WDTW), a variant of the traditional DTW (line 6 of Algorithm 7). This variant includes weights into the formulation to consider, when creating the path matrix, the phase difference between two samples $\mathbf{x}_i$, $\mathbf{x}_\mathbf{i}'$ from two distinct sequences of different length. In order to measure such phase difference between the samples, the authors introduce the Modified Logistic Weight Function (MLWF) for the weight calculations in line 3 of Algorithm 7. This way, the authors aim at penalising large phase differences in order to prevent minimum distance distortion caused by *outliers*. In this work, the WDTW idea is also extended to other variants of the DTW such as DDTW (line 6 of Algorithm 7). Later, the authors in Jeong and Jayaraman (2015) employ the proposed WDTW as a kernel function into a multiclass Support Vector Machine (SVM) (line 6 of Algorithm 7) for *time series* classification and validate the proposed model over datasets from the UCR repository.

The Weighted Permutation Entropy (WPE) is proposed by Fadlallah, Chen, Keil, and Príncipe (2013), in which the motif counts from signal patterns are weighted aiming at retaining the amplitude information of nonlinear time series. Given the time-delay embedding representation $X_j^{m,\tau}$ of a time series $\{x_t\}_{t=1}^T$ being $j = 1,\dots,T-(m-1)\tau$, m the embedding dimension and τ the time delay, for each j (line 2 of Algorithm 7) the authors calculate the weights in line 3 of Algorithm 7 as the variance of each neighbours vector $X_j^{m,\tau}$. This way the proposed approach assigns different weights to neighbouring vectors with same ordinal patterns but with different amplitude variations. The weighted relative frequencies for each motif are employed to compute the WPE (line 6 of Algorithm 7). WPE can be utilised to detect abrupt changes in noisy or multi-component signals and it is successfully employed by Zhou, Qian, Chang, Xiao, and Cheng (2018) in combination with Ensemble Empirical Mode Decomposition (EEMD) and SVM ensemble classifier (line 6 of Algorithm 7) for fault diagnosis of rolling bearing.

Filter approaches for local feature weights are also employed for static data with *missing values*. For instance, author in Datta, Misra, and Das (2016) propose the Feature Weighted Penalty based Dissimilarity (FWPD), a measure that considers the number of *missing features*. More concretely, such dissimilarity measure includes a parameter α to control the relative importance between two terms and a Feature Weighted Penalty that, for each pair of samples $\mathbf{x}_i, \mathbf{x}_{i'}$ with $i, i' \in \{1,\dots,n\}$, calculates the penalty weight (line 3 of Algorithm 7) as the proportion of shared features without *missing values* respect to the total number of observed features. This FWPD is included into the K-NN classifier mechanism (line 6 of Algorithm 7) to estimate the similarity between neighbours in order to handle *missing features*.

Wrapper local unsupervised feature weighting. Wrapper FW is the most common approach for searching the optimal local weights. In this case, as can be observed in Algorithm 8, there are as many weights as clusters K per feature. These randomly initialised local weights 2 of Algorithm 8 are updated in each iteration of the algorithm (line 9 of Algorithm 8) based on the results obtained by the ML algorithm (line 8 of Algorithm 8). This process ends when (line 10 of Algorithm 8) the performance overcomes a certain threshold θ, the algorithm converges, i.e., the performance do not present a significant improvement respect to the previous iteration, or until a number of iterations *n_iter* is reached (line 4 of Algorithm 8). The works related to wrapper local unsupervised FW methods found in the literature are mainly based on partitioning, hierarchical and fuzzy clustering. Other works based on classification and evolutionary algorithms can also be found.

Algorithm 8 Wrapper Local Unsupervised Feature Weighting methods

```
1: Given a dataset $X \subset \mathbb{R}^{n\times m}$, $\underline{\textbf{a clustering algorithm and } K}$
2: Initialise $\boxed{w_{kj}^{(0)} \in \mathbb{R} \;\; \forall k = 1 : K}$, $\forall j = 1 : m$
3: $s \leftarrow 0$
4: while $s < n\_iter$ do
5:     for j=1:m do
6:         for $\boxed{k = 1 : K}$ do
7:             $\widetilde{X}_{i'j}^{(s)} \leftarrow \boxed{w_{kj}^{(s)}} \cdot X_{i'j}$ $\forall i' \in \underline{\{i|\mathbf{x}_i = k\}}$
8:     $\widetilde{Y}^{(s)} \leftarrow$ ML algorithm to $\widetilde{X}^{(s)}$
9:     $w_{kj}^{(s+1)} \leftarrow$Adjust $w_{kj}^{(s)}$
10:    if $\underline{\textbf{perf}(\widetilde{Y}^{(s)})}> \theta$ or $=\underline{\textbf{perf}(\widetilde{Y}^{(s-1)})}+\epsilon$ then
11:        Break
12:    $s \leftarrow s + 1$
13: $\boxed{w_{kj}^{*} \leftarrow w_{kj}^{(s)}}$
```

Specifically, the Local Attribute Weighting K-Means (LKM) algorithm proposed in Chan, Ching, Ng, and Huang (2004) is akin to the one employed in Huang et al. (2005), but the weights, w_{kj} with $k = \{1,\dots,K\}$ and $j = \{1,\dots,m\}$, are related to a specific cluster k and feature j. The method for updating the weights in line 9 of Algorithm 8 relies on the cluster dispersion at each dimension. Similarly, Shen, Yang, Wang, and Liu (2006) propose the same idea adapted to the fuzzy C-means algorithm. Another related proposal can be found in Jing, Ng, and Huang (2007) where the Entropy Weighted K-Means algorithm (EWKM) for clustering high-dimensional objects in subspaces is presented. In this proposal the weight values are employed to identify the subsets of important dimensions that categorise different clusters. This is achieved by including the weight entropy in the objective function (line 8 of Algorithm 8) that is minimised in the clustering process in line 10 of Algorithm 8. The experiments on both synthetic and real data have shown that the new algorithm can generate better clustering results than other subspace clustering algorithms, such as: FWKM (Jing, Ng, Xu, & Huang, 2005), Bisecting K-means (Karypis, Kumar, & Steinbach, 2000), PROCLUS (Aggarwal, Wolf, Yu, Procopiuc, & Park, 1999), HARP (Yip, Cheung, & Ng, 2004), LAC (Domeniconi, Papadopoulos, Gunopulos, & Ma, 2004) and COSA (Friedman & Meulman, 2004) and that is scalable to large high-dimensional data. Following with the same line of research, in Chen, Ye, Xu, and Huang (2012) an extension of the EWKM algorithm (Jing et al., 2007) is introduced, called FG-k-means. It automatically calculates two types of local weights in line 9 of Algorithm 8, one for groups of features and the other one for the individual features in each cluster. The experimental results show the goodness of the FG-k-means over the other considered algorithms and its robustness to noise and missing values. Another work related to high-dimensional data is introduced by Gan and Ng (2015). The referred AFG-k-means algorithm, extends the idea of Chen et al. (2012) by incorporating an Automatic Feature Group selection algorithm which iteratively creates groups of features. Recently, Huang, Yang, et al. (2018) extend the weighted K-means algorithm presented in Chan et al. (2004) with a l^2 regularisation of the local features, obtaining the l^2-WKmeans. The authors introduce two versions of the algorithm to handle both numerical and categorical data clustering. Experimental results show that for both numerical and categorical data the proposal outperforms the state-of-the-art algorithms in terms of accuracy, Rand index, F-score and NMI.

In addition to the weighted K-means based algorithms, other clustering algorithms are adapted to include local weights into their formulation. In this sense, a new version of the Ward Hierarchical algorithm can be found in de Amorim (2015). It employs local cluster dependent weights calculated with the L_p norm in line 9 of Algorithm 8 over the normalised datasets. As the Ward algorithm, it does not need to know the number of clusters in advance. In addition, although the algorithm's performance depends on the term p of the L_p norm, the value of p that maximises the Silhouette index is proposed. The proposal of Chen, Wang, Wang, and Zhu (2016) presents a novel algorithm for Subspace Clustering of Categories (SCC) for clustering categorical data. The algorithm utilises a probabilistic distance function based on Kernel Density Estimation (KDE) to measure the dissimilarity between categorical objects. The local weights are calculated at each iteration of Algorithm 8 in line 9 as the smoothed dispersion of the features in each cluster. The authors also propose a new categorical cluster validity index which evaluates the average intra-cluster scatter and inter-cluster separation. In the context of consensus clustering, authors in Alguliyev, Aliguliyev, and Sukhostat (2020) develop a weighted consensus clustering for Big data applications that assigns weights to single clustering methods based on purity utility function.

Finally, to conclude with the partitioning clustering FW methods, authors in Hashemzadeh, Oskouei, and Farajzadeh (2019) present a Fuzzy C-means based method that automatically calculates local weights for the features and also includes in line 9 of Algorithm 8 a cluster weighting process to mitigate the initialisation sensitivity of the algorithm. During the optimisation process the feature weights are calculated in line 9 of Algorithm 8 based on the intra-cluster feature-weighted distance, while the cluster weights are recalculated by employing the intra-cluster distances. The authors compare their proposal with some state-of-the art algorithms (Frigui & Nasraoui, 2004; Tzortzis & Likas, 2014; Zhi, Fan, & Zhao, 2014; Zhou, Chen, Chen, Zhang, & Li, 2016) utilising large real world and synthetic datasets. The results confirmed that the proposed method is not sensitive to the cluster centroids initialisation and also that outperforms the clustering results obtained by its competitors. Furthermore, the authors conclude that the algorithm can be effectively employed on big data clustering and co-clustering applications used as *online* feature and clustering weighting method.

The local feature weighting strategy is also employed in conjunction with classification algorithms (line 8 of Algorithm 8). The Subspace Weighting Naive Bayes algorithm (SWNB) in Chen and Wang (2012) introduces a locally weighted probability model, in which the weights are optimised in line 9 of Algorithm 8 to fit a Logitnormal priori distribution and the Maximum a Posteriori principle for Bayesian modelling in high dimensional spaces. The weights are then calculated by the Newton–Raphson method. The experiments conducted on document corpora and gene microarray datasets and the comparisons with other weighting algorithms (Frank, Hall, & Pfahringer, 2002; John & Langley, 1995; Lee, Gutierrez, & Dou, 2011) demonstrate that the SWNB method is comparable or superior to its competitors.

Evolutionary algorithms are also applied to iteratively seek for the optimal local feature weights. In the research conducted by Gançarski and Blansché (2008) new FW methods based on EAs are proposed: Darwinian, Lamackian and Baldwinian Evolutionary algorithms and their co-evolutionary approach (DE-LKM, LE-LKM, BE-LKM and DC-LKM, LC-LKM, BC-LKM). The algorithms utilise in line 10 of Algorithm 8 the cost function defined in LKM (Chan et al., 2004) as fitness function and estimate the internal quality of the unsupervised classification by means of the Wemmert–Gançarski cluster quality index (Wemmert, Gançarski, & Korczak, 2000). Finally, the NSGA algorithm (Deb, Pratap, Agarwal, & Meyarivan, 2002) is employed as optimisation tool for the multi-objective optimisation approach proposed by Zhou and Zhu (2018) with the aim of simultaneously estimating the cluster centres and, in line 9 of Algorithm 8, the local weights that minimise the intra-cluster dispersion and maximise the inter-cluster separation. Similarly, the work presented in Liu, Xie, Zhao, Xie, and Liu (2019) employs the Adaptive Shorting-based Evolutionary Algorithm (ASEA) to optimise a new clustering validity index for credit risk assessment based on four terms: (1) the weighted intra-cluster compactness, (2) the inter-cluster separation, (3) penalties for the feature, (4) penalties for the cluster weights. More concretely, the cluster weights and the term intra-cluster compactness are incorporated to address the issue of class imbalance.

3.2.3. *Concluding remarks about unsupervised approaches*

Regarding unsupervised FW approaches, filter methods are not commonly employed in the literature and these obtain weights per sample, obviating the differences in relevance among dimensions. The reviewed works usually calculate the features relevance based on previous clustering results or, in the case of *time series*, estimate the relevance of the samples considering the temporal distance from the target. In these cases, the weights quality are dependent on the suitability of the clusters configuration or on the availability of expert knowledge about the temporal dependency of the system, respectively. Wrapper approaches, instead, are the most widely applied and include the feature weights into the objective function of the ML algorithm. Nevertheless, the application of evolutionary techniques is not so popular in the unsupervised environment. The weights estimation is usually done as a function of the current partition, so the results are generally highly dependent on the initialisation of the algorithm.

Table 1 summarises the reviewed research works in the field of FW in the supervised and the unsupervised environment. It includes information about (a) the specific technique applied for FW, (b) the type of data of the features, (c) the application field and/or the datasets used in the research, (d) the research works and/or techniques used to compare the authors' proposal, and (e) the ML performance measure.

4. Considerations for the application of feature weighting methods

In this section some guide points for the application of FW methods are proposed regarding the different taxonomy levels presented in Fig. 3.

4.1. *Supervised versus unsupervised learning*

At the first level of the proposed taxonomy, the FW methods are divided into supervised and unsupervised learning approaches. In our opinion, supervised FW approach is always advisable if *labels* are available (Ahn & Kim, 2009). By this way, the relevance can be computed as the informative ability of each feature for estimating the label (Chen & Hao, 2017; Jiang et al., 2018), increasing the output accuracy of the ML model (Sinciya & Celin, 2017). However, in many problems labels are not always available, thus, unsupervised FW methods can be applied as follows: (1) expert-knowledge of the problem field can be employed to approximate the influence of each feature on the physical process (Sahin et al., 2015); (2) temporal information, such as the distance of the samples to a reference temporal moment, or the phase difference (Benkabou et al., 2018; Jeong et al., 2011) can be considered to weigh the samples and (3) the estimation of the feature relevance can be done by means of the metric employed for the ML clustering algorithm, such as the cohesion or separation of the former clusters (De Amorim & Mirkin, 2012; Jing et al., 2007). In the latter case, the final result is highly dependent on the initialisation of the ML clustering algorithm (Gan & Ng, 2015). In order to circumvent this problem, some FW methods integrate an evolutionary algorithm for the optimisation process to be able to efficiently explore the solution space and not converge to a local optimum (Gançarski & Blansché, 2008; Kuo & Nguyen, 2019).

4.2. *Global versus local approach*

The second level of the proposed taxonomy discriminates between the way the weights are calculated, i.e., globally or locally. In the local supervised approach, the samples are merged according to their label and for each group of samples a local weight is commonly calculated (Chen & Guo, 2015). However, the challenge is to assign, among all the estimated weights, the corresponding one to a new unlabelled sample. In this context, probabilistic methods are usually applied for estimating the class to which the new sample belongs, and therefore,

Table 1
FW research works classified following the proposed taxonomy in Fig. 3.

			Ref.	FW technique	Features	Application	Comparison	ML performance measure
Supervised	Global	Filter	Xing et al. (2009)	MI	Real	UCI (Dua & Graff, 2017), Microarray, Ripley's synthetic dataset	Sun (2007), Wang, Wang, and Wang (2004)	Accuracy
			Giveki et al. (2012)	MI	Integer, Real	Diabetes	SVM-based classifiers	Sensitivity, Specificity, Accuracy
			Jiang et al. (2018)	MI	Categorical, Integer, Real	UCI (Dua & Graff, 2017)	Jiang, Li, Wang, and Zhang (2016), Lee et al. (2011), Wu and Cai (2011), Zaidi et al. (2013), Zhang and Sheng (2004)	CLL, AUC
			Chen and Hao (2017)	MI	Real	Stock market indices prediction	SVM-KNN	MAPE, RSME
			Wu et al. (2017)	Entropy	Text	Text categorisation	Jones, Walker, and Robertson (2000), Lan, Tan, Su, and Lu (2008), Martineau and Finin (2009), Paltoglou and Thelwall (2010), Sparck Jones (1972), Wu and Salton (1981)	Accuracy
			Sahin et al. (2015)	χ^2, F-score	Image	Landslide susceptibility mapping	AHP model	Accuracy, Success rate curve
			Bhattacharya et al. (2017)	Mean, Standard deviation	Integer ,Real	UCI (Dua & Graff, 2017), Face, Hand-writing recognition	Metrics, FS	Accuracy
			Niño-Adan et al. (2019)	Pearson/RF	Integer, Real	UCI (Dua & Graff, 2017)	normalisation methods	Accuracy
			Elbasiony et al. (2013)	RF	Categorical, Real	NID (Cup, 1999)	Other NDI method	ROC, Detection rate, FP rate
			Peng et al. (2017)	MI/Pearson	(Im)Balanced	DGC	Peng, Zhang, Yang, and Chen (2014)	Accuracy, Computational time, AUC, Cohen's kappa coefficient
			Zhang et al. (2016)	GR/DT	Binary	Text classification	Li, Luo, and Chung (2012), Wang, Jiang, and Li (2014)	Accuracy
		Wrapper	Phan et al. (2017)	GA/Accuracy, F-measure, MCC	Real	UCI (Dua & Graff, 2017)	Jiang et al. (2016), Sáez, Derrac, Luengo, and Herrera (2014), Wu et al. (2015), Xiang, Yu, and Kang (2016)	Accuracy, Computational cost
			Komosiński and Krawiec (2000)	GA	Images	Neuropathology	Feature selection	Accuracy
			Ahn and Kim (2009)	GA/Accuracy	Real	Cancer	GA for CBR	Accuracy
			Mateos-García et al. (2017)	EA/CV error	Binary, Real	UCI (Dua & Graff, 2017)	Dudani (1976), García-Gutiérrez, Mateos-García, and Riquelme-Santos (2014), Gou, Du, Zhang, Xiong, et al. (2012), Gou, Xiong, and Kuang (2011)	Accuracy
			Triguero et al. (2012)	DE/Accuracy	Real	KEEL (Alcalá-Fdez et al., 2011)	Kononenko (1994), Tahir, Bouridane, and Kurugollu (2007)	Accuracy, Reduction
			Serrano-Silva et al. (2018)	DE, GA, NBA/AUC	Categorical, Integer, Real	Financial risk	Between them	AUC, Execution time
			Huang, Zhang, et al. (2018)	Gradient Ascent/Margin vector	Categorical, Real	UCI (Dua & Graff, 2017)	Cai, Ruan, Ng, and Akutsu (2014), Kononenko (1994), Sun (2007)	Accuracy, Precision, Recall, F-measure
	Local	Filter	Marchiori (2013)	Local Relief	Categorical, Integer, Real	Cancer	Relief	Accuracy
			Yilmaz et al. (2014)	Local Relief-f	Multimedia	Multimodal fusion	Relief-F, Exhaustive search	Accuracy, Time execution
			Chen and Guo (2015)	Entropy, Gini	Categorical	UCI (Dua & Graff, 2017)	Global version	Micro / Macro F1-measure
		Wrapper	Paredes and Vidal (2000)	FPGD/Intra, inter-class distances	Categorical, Integer, Real	UCI (Dua & Graff, 2017)	L_2, MD, Class-dependent MD	Accuracy
			Paredes and Vidal (2006)	GD/Misclassification probability	Categorical, Integer, Real	Synthetic, UCI (Dua & Graff, 2017)	Dissimilarity measures, Class and/or prototype dependent	Error rate
			Jiao et al. (2015)	MHNN/Centroids distance, feature distribution	Categorical, Integer, Real	Synthetic, UCI (Dua & Graff, 2017)	L_2,Paredes and Vidal (2000), Paredes and Vidal (2006)	Run time, Error rate, Accuracy
			Jiao et al. (2019)	Maximum Likelihood/Pairwise distances	Categorical, Integer, Real	Synthetic, UCI (Dua & Graff, 2017)	Global, Local Pairwise L_2, MD	Accuracy, Run time
			Taheri et al. (2014)	Quasisecant method/NB binary classification accuracy	Categorical, Integer, Real	UCI (Dua & Graff, 2017)	NB variants	Accuracy
			Jiang et al. (2019)	GD/CLL, MSE	Categorical, Integer, Real	UCI (Dua & Graff, 2017)	Hall (2006), Jiang et al. (2016), Lee et al. (2011), Wu and Cai (2011), Zaidi et al. (2013), Zhang and Sheng (2004)	Accuracy
			Mohemmed and Zhang (2008)	PSO/NCC classification error	Categorical, Integer, Real	UCI (Dua & Graff, 2017)	L_2, Class-dependent MD, Paredes and Vidal (2006)	Accuracy
			Mateos-García et al. (2012)	DE/K-NN classification error	Categorical, Integer, Real	UCI (Dua & Graff, 2017)	AlSukker et al. (2010), Paredes and Vidal (2006)	Accuracy
			Sinciya and Celin (2017)	PSO/DGC accuracy	Real	UCI (Dua & Graff, 2017)	DGC, K-NN	Precision, Recall, F-measure

(*continued on next page*)

Table 1 (*continued*).

			Ref.	FW technique	Features	Application	Comparison	ML performance measure
Unsupervised	Global	Filter	Güriiler (2017)	Distance from centroids	Real	Parkinson (Little et al., 2007)	Polat (2012), Sakar and Kursun (2010)	Sensitivity, Specificity, Precision, Recall, F-measure, Kappa statistic
			Güneş et al. (2010)	Distance from centroids	Statistical features	Sleep stage recognition	Unweighted features	Success rate (Accuracy)
		Wrapper	Huang et al. (2005)	Partial optimisation / Intra-cluster dispersion	Categorical, Integer, Real	Synthetic, UCI (Dua & Graff, 2017)	K-means, Different β values, Feature selection	Accuracy
			Saha and Das (2015)	Alternative Optimisation/Cluster membership	Categorical	Synthetic, UCI (Dua & Graff, 2017)	Fuzzy K-modes	Rand index, Partition Coefficient, Partition Entropy
			De Amorim and Mirkin (2012)	Partial optimisation/Intra-cluster dispersion	Categorical, Integer, Real	Synthetic, UCI (Dua & Graff, 2017)	K-means Huang et al. (2005)	Accuracy
			Chakraborty and Das (2018)	Partial optimisation/Intra-cluster dispersion	Integer, Real	UCI (Dua & Graff, 2017), KEEL (Alcalá-Fdez et al., 2011)	K-means, G-means, Huang et al. (2005)	NMI
			Tsai and Chiu (2008)	Partial optimisation/Inter and intra cluster separation	Integer, Real	UCI (Dua & Graff, 2017)	K-means, Huang et al. (2005)	SSE, Entropy, ARI, Convergence
			Benkabou et al. (2018)	Partial optimisation/Intra-cluster dispersion, Weight entropy	Time series	Outlier detection	Bevilacqua and Tsaftaris (2015), Breunig et al. (2000), Budalakoti et al. (2008), Ng et al. (2002), Portnoy (2000), Schölkopf et al. (2000), Yeung and Chow (2002)	AUC, Time complexity
			Zhang et al. (2018)	Partial optimisation/Intra-cluster dispersion, Weight entropy	Multiple view	UCI (Dua & Graff, 2017)	Jing et al. (2007) and Multi-view clustering algorithms	NMI, Precision, F-score, Classification rate
	Local	Filter	Jeong et al. (2011)	Phase difference	Time series	UCR (Dau et al., 2018)	L_2, DTW	Error rate, Entropy, F-measure, Average inter/intra cluster distance
			Jeong and Jayaraman (2015)	Phase difference	Time series	UCR (Dau et al., 2018)	L_2, DTW	Accuracy, Sensitivity, Specificity
			Fadlallah et al. (2013)	WPE	Time series	Electro-encephalogram	Entropy-based measurements	Entropy
			Zhou et al. (2018)	WPE	Time series	Fault diagnosis	SVM, K-NN, Extreme Learning	Accuracy
		Wrapper	Chan et al. (2004)	Partial optimisation/Intra-cluster dispersion	Categorical, Integer, Real	Synthetic, UCI (Dua & Graff, 2017)	K-means, β	Accuracy
			Jing et al. (2007)	Partial optimisation/Intra-cluster dispersion, Weight entropy	Text	Synthetic, UCI (Dua & Graff, 2017)	Aggarwal et al. (1999), Domeniconi et al. (2004), Friedman and Meulman (2004), Jing et al. (2005), Karypis et al. (2000), Yip et al. (2004)	Entropy, F-score, NMI
			Chen et al. (2012)	Partial optimisation/Intra-cluster distance , Weights entropy	Real	Synthetic, UCI (Dua & Graff, 2017)	K-means, Domeniconi et al. (2007), Huang et al. (2005), Jing et al. (2007)	Precision, Recall, F-measure, Accuracy
			Gan and Ng (2015)	Partial optimisation/Intra-cluster distance, Feature and group weight entropy	Real	Synthetic, Gene expression	Chen et al. (2012), Domeniconi et al. (2007), Gan and Wu (2008), Jing et al. (2007)	Corrected Rand index, Execution run time
			Huang, Yang, et al. (2018)	Partial optimisation/L_2 norm regularisation, Weights entropy	Categorical	Synthetic, UCI (Dua & Graff, 2017)	Chan et al. (2004), Huang et al. (2005), Jing et al. (2007)	Accuracy, Rand index, F-score, NMI
			de Amorim (2015)	Partial optimisation/L_p distance	Categorical, Integer, Real	Synthetic, UCI (Dua & Graff, 2017)	Ward	Adjusted Rand Index, Noise features contribution
			Chen et al. (2016)	Bandwidth optimisation/Total error	Categorical	Synthetic, UCI (Dua & Graff, 2017)	Bai, Liang, Dang, and Cao (2011), Cao, Liang, Li, and Zhao (2013), Chan et al. (2004), Domeniconi et al. (2007), San, Huynh, and Nakamori (2004)	Accuracy, NMI, Categorical cluster validity index, Run time
			Hashemzadeh et al. (2019)	Partial optimisation/Intra-cluster dispersion	Categorical, Integer, Real	Synthetic, UCI (Dua & Graff, 2017)	Frigui and Nasraoui (2004), Tzortzis and Likas (2014), Zhi et al. (2014), Zhou et al. (2016)	Accuracy, NMI
			Chen and Wang (2012)	Numerical methods/Logitnormal distribution	Real	Gene expression	Frank et al. (2002), John and Langley (1995), Lee et al. (2011)	Micro/Macro F-measure, Scalability
			Gançarski and Blansché (2008)	Darwinian, Lamackian and Baldwinian (co)evolutionary algorithms/Intra-cluster distance	Categorical, Integer, Real	UCI (Dua & Graff, 2017)	K-means, Between them	Falks-Mallow index, Rand index, Jaccard coefficient
			Zhou and Zhu (2018)	Multi-objective optimisation/Inter–intra cluster measures	Categorical, Integer, Real	UCI (Dua & Graff, 2017)	FW clustering	Accuracy, Rand index, NMI
			Liu et al. (2019)	Multi-objective optimisation/intra-cluster compactness inter-cluster separation	Categorical, Integer, Real	UCI (Dua & Graff, 2017)	Dunn (1973), Gan and Wu (2008), Parvin and Minaei-Bidgoli (2013), Xia, Zhuang, and Yu (2013)	Rand Index, F-measure

assigning its weight. Another common strategy is to employ confidence-based methods for assigning a local weight to the new sample (Jiang et al., 2019). Nevertheless, depending on the number of local weights, the *computational cost* may increase and global FW methods might be a more advisable approach.

Regarding unsupervised environments, expert-knowledge based local weights can also be employed. Furthermore, in *condition-based systems*, in which some feature represents a specific condition of the physical system, makes advisable to assign different weights depending on the particular system condition. Similarly, if the *temporal evolution* of the analysed system is known, the local weight may consider the temporal distance of each sample to the referenced one.

Generally, in most cases a hybrid (global and local) approach can be applied. However, for *high dimensional datasets or when a high number of local weights* is presented, a global FW strategy may be recommended for reducing the computational cost.

4.3. Filter versus wrapper method

In the last level of the proposed taxonomy, the estimation of the weights are divided into filter and wrapper methods.

In regards to unsupervised FW methods, the wrapper methods are the most employed ones in the literature. Moreover, in supervised approaches, they have also proven to *increase the algorithm performance* respect to filter approaches (Marchiori, 2013; Yilmaz et al., 2014). However, the calculated weights are highly dependent on the ML algorithm configuration (Phan et al., 2017), its hyper-parameters, or even the training dataset used (Peng et al., 2017). Therefore, in order to ensure *explainability*, i.e. for a decision support system (Chen & Guo, 2015) or *high dimensional datasets* (Yilmaz et al., 2014), filter methods are a preferable approach.

Nevertheless, some encoding considerations must be taken to handle categorical features with continuous ones in the same classification problem. Similarly, the estimation of the probability distribution in some real scenarios is not always possible. Then, instead of Information theory-based approaches, Statistical-based FW methods may be more suitable in these cases.

Finally, it must be taken into account that the differences among the features' magnitudes can influence the results more than the calculated weights by the FW methods. Consequently, prior to the weights calculation, additional preprocessing steps, like normalisation methods (Milligan & Cooper, 1988), and a rigorous analysis about the influence of such rescaling transformation (Niño-Adan et al., 2019) should be conducted.

4.4. FW approach selection

Based on the conclusions drawn from the analysis of the literature presented in Section 3 as well as the considerations above, this Section provides recommendations for the optimal selection of the FW approach depending on the characteristics and objectives of the problem at hand.

1. **Labels**: If the dataset is labelled, **supervised** feature weighting approaches should be applied.
2. **High-dimensional dataset** (Peng et al., 2017): If the dataset comprises a large number of features, computationally cheaper strategies – like **global weights or filter methods** – should be considered.
3. **Dimensionality reduction** (Panday et al., 2018): If the reduction of the number of features is needed due to computational complexity problems, in order to select the most informative ones, **global filter** approaches should be applied as Feature Selection method.
4. **Dataset understanding**: If there is no prior information about the problem, the weights calculated by **filter methods** enhance the understanding of the relevance of each feature of the dataset.
5. **Features contribution estimation**: If the aim is to extract knowledge about the influence of each feature on the performance of the model scoring, **wrapper methods** are recommended, since the weights obtained are representative values of the contribution each feature has on the outputs estimated by the model.
6. **Missing values** (Datta et al., 2016): If the dataset contains missing values and the application of some imputation method is required, the approaches are: (1) to assign a lower **global feature weight** to those features with missing values; (2) to employ **local weights** by means of assigning lower weight to the samples with missing values, and representative values – of the relevance between the sample and the label – to the remaining samples.
7. **Imbalanced dataset** (Liu et al., 2019): If the dataset is imbalanced, **local feature weighting** strategies aim at increasing the relevance of the minority class samples.
8. **Outliers** (Jeong et al., 2011): If the dataset contains outliers, **local feature weighting** methods allow assigning lower weight to such samples for minimising their influence.
9. **Noise**: If noisy features are expected in a dataset, using **global filter weighting approaches** the weight of an artificially-generated noisy feature can be estimated w^*. Then, the features of the dataset that present similar weights w_j (calculated by the same technique) to such estimated reference weight, $w_j \leq w^* + \epsilon$ can be identified as noisy features.
10. **Interpretability** (Chen & Guo, 2015): If the interpretability of the model is imperative, as stated in Section 3.1.3, **filter approaches** based on Information theory or Statistical approaches should be considered.
11. **Condition-based problems**: If based on expert knowledge it is known that the analysed system presents different operational conditions (determined by the value of certain representative features can take), and thus, the relevance of the rest of features vary depending on the condition under the system is working, then **local feature weights** are recommended.
12. **Temporal dependency** (Fadlallah et al., 2013; Jeong et al., 2011): If the label of a sample depends not only on the values of such sample but also on the values of the previous ones, it is recommended to assign **local weights** to the samples according to their temporal distance respect to the analysed one.
13. **Algorithm performance maximisation** (Ahn & Kim, 2009; Mateos-García et al., 2017; Triguero et al., 2012; Yang et al., 2012): If a maximisation of the algorithm performance – in terms of metric optimisation – is sought regardless the interpretability of the model, **wrapper weighting** methods that adjust the weights in order to maximise such metric are suggested.
14. **Semi-supervised learning** (De Amorim & Mirkin, 2012): If the dataset comprises both labelled and unlabelled samples, **local weights** to increase the impact of the labelled samples respect to the unlabelled ones are advisable.
15. **Online learning** (Marchiori, 2013; Yilmaz et al., 2014) : If an online fashion is faced and extreme label latency is expected, then **unsupervised weighting methods** are recommended. Besides, if concept drift is foreseen, for both quick discovering of such drift and rapid adjustment of the weights **filter approaches** are advised.

Fig. 4 depicts the above mentioned recommendations into the different levels of the proposed taxonomy (Fig. 3).

5. Conclusions and future challenges

This work presents an extensive review of FW methods based on a proposed taxonomy or classification scheme, i.e: (1) At a first level, supervised and unsupervised methods are differentiated; (2) Then,

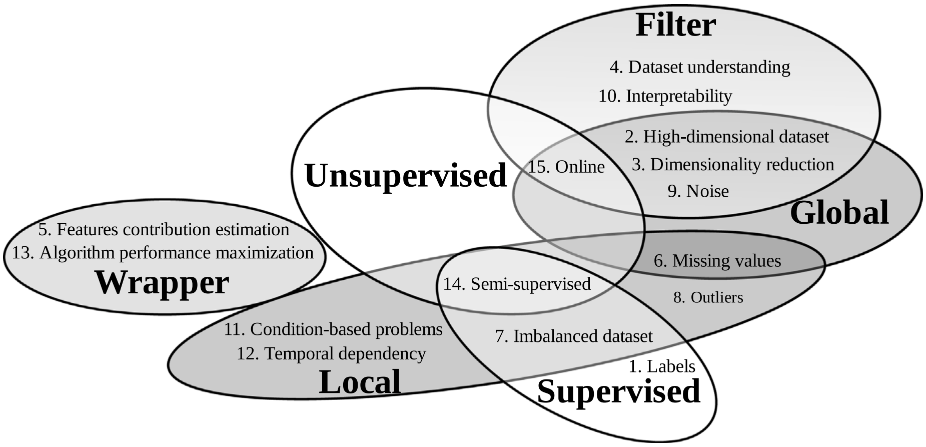


Fig. 4. Recommendations for the optimal selection of the Feature Weighting approach.

regarding whether the application of the weights is over the entire or over a subset of the instance space, global and local approaches are presented, respectively; and (3) finally, the evaluation criteria and the interaction with the ML algorithm give rise to a filter/wrapper classification at a lower level. Moreover, some recommendations and guide points for the optimal selection of the FW approach are shown, regarding the characteristics and objectives of the problem at hand.

In light of the great interest that attracted the FW field in the last years, it is expected that the number of related works continues growing. As observed in the conducted review, the majority of works are focused on integer, real or categorical input data, being few the FW works applied to time series data. Since nowadays many real applications rely on time series data, future work may be focused on delving FW methods for time series data. Moreover, in order to be able to create a methodology for the application of a FW method based on the characteristics of the input data and the problem at hand, a complete comparison in theoretical and experimental terms of the different FW approaches has to be addressed. Similarly, an exhaustive study of the degree of susceptibility of the ML algorithms to the feature weighting transformation will be interesting and help the selection of the FW strategy.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has been supported in part by the ELKARTEK Research Programme of the Basque Government (Argia KK-2019/00068), the HAZITEK program (DATALYSE ZL-2018/00765), the University of the Basque Country (GIU19/045), the DATAinc program (48-AF-W1-2019-00002), and a TECNALIA Research and Innovation PhD Scholarship.

References

Aggarwal, C. C., Han, J., Wang, J., & Yu, P. S. (2004). A framework for projected clustering of high dimensional data streams. In *Proceedings of the thirtieth international conference on very large data bases-Volume 30* (pp. 852–863).

Aggarwal, C. C., Wolf, J. L., Yu, P. S., Procopiuc, C., & Park, J. S. (1999). Fast algorithms for projected clustering. *ACM SIGMoD Record*, *28*(2), 61–72.

Ahn, H., & Kim, K.-j. (2009). Global optimization of case-based reasoning for breast cytology diagnosis. *Expert Systems with Applications*, *36*(1), 724–734.

Aksoy, S., & Haralick, R. M. (2001). Feature normalization and likelihood-based similarity measures for image retrieval. *Pattern Recognition Letters*, *22*(5), 563–582.

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., et al. (2011). Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, *17*.

Alguliyev, R. M., Aliguliyev, R. M., & Sukhostat, L. V. (2020). Weighted consensus clustering and its application to Big data. *Expert Systems with Applications*, *150*, Article 113294.

AlSukker, A., Khushaba, R., & Al-Ani, A. (2010). Optimizing the k-nn metric weights using differential evolution. In *2010 international conference on multimedia computing and information technology* (pp. 89–92). IEEE.

de Amorim, R. C. (2015). Feature relevance in ward's hierarchical clustering using the L p norm. *Journal of Classification*, *32*(1), 46–62.

de Amorim, R. C. (2016). A survey on feature weighting based K-means algorithms. *Journal of Classification*, *33*(2), 210–242.

Bai, L., Liang, J., Dang, C., & Cao, F. (2011). A novel attribute weighting algorithm for clustering high-dimensional categorical data. *Pattern Recognition*, *44*(12), 2843–2861.

Benkabou, S.-E., Benabdeslem, K., & Canitia, B. (2018). Unsupervised outlier detection for time series by entropy and dynamic time warping. *Knowledge and Information Systems*, *54*(2), 463–486.

Bevilacqua, M., & Tsaftaris, S. (2015). Dictionary-decomposition-based one-class svm for unsupervised detection of anomalous time series. In *Proceedings of 23rd European signal processing conference* (pp. 1776–1780).

Bhattacharya, G., Ghosh, K., & Chowdhury, A. S. (2017). Granger causality driven AHP for feature weighted kNN. *Pattern Recognition*, *66*, 425–436.

Bishop, C. M., et al. (1995). *Neural networks for pattern recognition*. Oxford University Press.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on management of data* (pp. 93–104).

Budalakoti, S., Srivastava, A. N., & Otey, M. E. (2008). Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *39*(1), 101–113.

Cai, H., Ruan, P., Ng, M., & Akutsu, T. (2014). Feature weight estimation for gene selection: a local hyperlinear learning approach. *BMC Bioinformatics*, *15*(1), 70.

Cao, F., Liang, J., Li, D., & Zhao, X. (2013). A weighting k-modes algorithm for subspace clustering of categorical data. *Neurocomputing*, *108*, 23–30.

Chakraborty, S., & Das, S. (2018). Simultaneous variable weighting and determining the number of clusters- A weighted Gaussian means algorithm. *Statistics & Probability Letters*, *137*, 148–156.

Chan, E. Y., Ching, W. K., Ng, M. K., & Huang, J. Z. (2004). An optimization algorithm for clustering using weighted dissimilarity measures. *Pattern Recognition*, *37*(5), 943–952.

Chen, L., & Guo, G. (2015). Nearest neighbor classification of categorical data by attributes weighting. *Expert Systems with Applications*, *42*(6), 3142–3149.

Chen, Y., & Hao, Y. (2017). A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction. *Expert Systems with Applications*, *80*, 340–355.

Chen, L., & Wang, S. (2012). Automated feature weighting in naive bayes for high-dimensional data classification. In *Proceedings of the 21st ACM international conference on information and knowledge management* (pp. 1243–1252). ACM.

Chen, L., Wang, S., Wang, K., & Zhu, J. (2016). Soft subspace clustering of categorical data with probabilistic distance. *Pattern Recognition*, *51*, 322–332.

Chen, X., Ye, Y., Xu, X., & Huang, J. Z. (2012). A feature group weighting method for subspace clustering of high-dimensional data. *Pattern Recognition*, *45*(1), 434–446.

Chen, X., Yin, W., Tu, P., & Zhang, H. (2009). Weighted k-means algorithm based text clustering. In *2009 international symposium on information engineering and electronic commerce* (pp. 51–55). IEEE.

Christopher Frey, H., & Patil, S. R. (2002). Identification and review of sensitivity analysis methods. *Risk Analysis*, *22*(3), 553–578.

Cup, K. (1999). The UCI KDD Archive. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

Daszykowski, M., Kaczmarek, K., Vander Heyden, Y., & Walczak, B. (2007). Robust statistics in data analysis - a review: basic concepts. *Chemometrics and Intelligent Laboratory Systems*, *85*(2), 203–219.

Datta, S., Misra, D., & Das, S. (2016). A feature weighted penalty based dissimilarity measure for k-nearest neighbor classification with missing features. *Pattern Recognition Letters*, *80*, 231–237.

Dau, H. A., Keogh, E., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., et al. (2018). The UCR time series classification archive. URL https://www.cs.ucr.edu/∼eamonn/time_series_data_2018/.

De Amorim, R. C., & Mirkin, B. (2012). Minkowski metric, feature weighting and anomalous cluster initializing in K-means clustering. *Pattern Recognition*, *45*(3), 1061–1075.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197.

Deng, Z., Choi, K.-S., Jiang, Y., Wang, J., & Wang, S. (2016). A survey on soft subspace clustering. *Information Sciences*, *348*, 84–106.

Ding, Y., & Li, X. (2005). Time weight collaborative filtering. In *Proceedings of the 14th ACM international conference on information and knowledge management* (pp. 485–492).

Domeniconi, C., Gunopulos, D., Ma, S., Yan, B., Al-Razgan, M., & Papadopoulos, D. (2007). Locally adaptive metrics for clustering high dimensional data. *Data Mining and Knowledge Discovery*, *14*(1), 63–97.

Domeniconi, C., Papadopoulos, D., Gunopulos, D., & Ma, S. (2004). Subspace clustering of high dimensional data. In *Proceedings of the 2004 SIAM international conference on data mining* (pp. 517–521). SIAM.

Dua, D., & Graff, C. (2017). UCI machine learning repository. URL http://archive.ics.uci.edu/ml.

Dudani, S. A. (1976). The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, (4), 325–327.

Dunn, J. C. (1973). *A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters*. Taylor & Francis.

Elbasiony, R. M., Sallam, E. A., Eltobely, T. E., & Fahmy, M. M. (2013). A hybrid network intrusion detection framework based on random forests and weighted k-means. *Ain Shams Engineering Journal*, *4*(4), 753–762.

Fadlallah, B., Chen, B., Keil, A., & Príncipe, J. (2013). Weighted-permutation entropy: A complexity measure for time series incorporating amplitude information. *Physical Review E*, *87*(2), Article 022911.

Frank, E., Hall, M., & Pfahringer, B. (2002). Locally weighted naive bayes. In *Proceedings of the nineteenth conference on uncertainty in artificial intelligence* (pp. 249–256). Morgan Kaufmann Publishers Inc.

Friedman, J. H., & Meulman, J. J. (2004). Clustering objects on subsets of attributes (with discussion). *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *66*(4), 815–849.

Frigui, H., & Nasraoui, O. (2004). Unsupervised learning of prototypes and attribute weights. *Pattern Recognition*, *37*(3), 567–581.

Gan, G., & Ng, M. K.-P. (2015). Subspace clustering with automatic feature grouping. *Pattern Recognition*, *48*(11), 3703–3713.

Gan, G., & Wu, J. (2008). A convergence theorem for the fuzzy subspace clustering (FSC) algorithm. *Pattern Recognition*, *41*(6), 1939–1947.

Gançarski, P., & Blansché, A. (2008). Darwinian, lamarckian, and baldwinian (co) evolutionary approaches for feature weighting in K-means-based algorithms. *IEEE Transactions on Evolutionary Computation*, *12*(5), 617–629.

García, S., Luengo, J., & Herrera, F. (2015). *Data preprocessing in data mining*. Springer.

García, S., Luengo, J., & Herrera, F. (2016). Tutorial on practical tips of the most influential data preprocessing algorithms in data mining. *Knowledge-Based Systems*, *98*, 1–29.

García-Gutiérrez, J., Mateos-García, D., & Riquelme-Santos, J. C. (2014). Improving the k-nearest neighbour rule by an evolutionary voting approach. In *International conference on hybrid artificial intelligence systems* (pp. 296–305). Springer.

García-Laencina, P. J., Sancho-Gómez, J.-L., Figueiras-Vidal, A. R., & Verleysen, M. (2009). K nearest neighbours with mutual information for simultaneous classification and missing data imputation. *Neurocomputing*, *72*(7–9), 1483–1493.

Giveki, D., Salimi, H., Bahmanyar, G., & Khademian, Y. (2012). Automatic detection of diabetes diagnosis using feature weighted support vector machines based on mutual information and modified cuckoo search. arXiv preprint arXiv:1201.2173.

Gou, J., Du, L., Zhang, Y., Xiong, T., et al. (2012). A new distance-weighted k-nearest neighbor classifier. *Journal of Information and Computer Sciences*, *9*(6), 1429–1436.

Gou, J., Xiong, T., & Kuang, Y. (2011). A novel weighted voting for K-nearest neighbor rule. *Journal of Computational Physics*, *6*(5), 833–840.

Granger, C. W. (1988). Causality, cointegration, and control. *Journal of Economic Dynamics and Control*, *12*(2–3), 551–559.

Güneş, S., Polat, K., & Yosunkaya, Ş. (2010). Efficient sleep stage recognition system based on EEG signal using k-means clustering based feature weighting. *Expert Systems with Applications*, *37*(12), 7922–7928.

Gürüler, H. (2017). A novel diagnosis system for Parkinson's disease using complex-valued artificial neural network with k-means clustering feature weighting method. *Neural Computing and Applications*, *28*(7), 1657–1666.

Hall, M. (2006). A decision tree-based attribute weighting filter for naive Bayes. In *International conference on innovative techniques and applications of artificial intelligence* (pp. 59–70). Springer.

Hall, M. (2007). A decision tree-based attribute weighting filter for naive Bayes. *Knowledge-Based Systems*, *20*(2), 120–126. http://dx.doi.org/10.1016/j.knosys.2006.11.008, AI 2006.

Hamerly, G., & Elkan, C. (2004). Learning the k in k-means. In *Advances in neural information processing systems* (pp. 281–288).

Hashemzadeh, M., Oskouei, A. G., & Farajzadeh, N. (2019). New fuzzy C-means clustering method based on feature-weight and cluster-weight learning. *Applied Soft Computing*, *78*, 324–345.

Huang, J. Z., Ng, M. K., Rong, H., & Li, Z. (2005). Automated variable weighting in k-means type clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5), 657–668.

Huang, X., Yang, X., Zhao, J., Xiong, L., & Ye, Y. (2018). A new weighting k-means type clustering framework with an l2-norm regularization. *Knowledge-Based Systems*, *151*, 165–179.

Huang, X., Zhang, L., Wang, B., Zhang, Z., & Li, F. (2018). Feature weight estimation based on dynamic representation and neighbor sparse reconstruction. *Pattern Recognition*, *81*, 388–403.

Hung, W.-L., Chang, Y.-C., & Lee, E. S. (2011). Weight selection in WK-means algorithm with an application in color image segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *62*(2), 668–676.

Hussain, S. F. (2019). A novel robust kernel for classifying high-dimensional data using support vector machines. *Expert Systems with Applications*, *131*, 116–131.

Jain, A., Nandakumar, K., & Ross, A. (2005). Score normalization in multimodal biometric systems. *Pattern Recognition*, *38*(12), 2270–2285.

Jeong, Y.-S., & Jayaraman, R. (2015). Support vector-based algorithms with weighted dynamic time warping kernel function for time series classification. *Knowledge-Based Systems*, *75*, 184–191.

Jeong, Y.-S., Jeong, M. K., & Omitaomu, O. A. (2011). Weighted dynamic time warping for time series classification. *Pattern Recognition*, *44*(9), 2231–2240.

Jiang, L., Li, C., Wang, S., & Zhang, L. (2016). Deep feature weighting for naive Bayes and its application to text classification. *Engineering Applications of Artificial Intelligence*, *52*, 26–39.

Jiang, L., Zhang, L., Li, C., & Wu, J. (2018). A correlation-based feature weighting filter for naive Bayes. *IEEE Transactions on Knowledge and Data Engineering*, *31*(2), 201–213.

Jiang, L., Zhang, L., Yu, L., & Wang, D. (2019). Class-specific attribute weighted naive Bayes. *Pattern Recognition*, *88*, 321–330.

Jiao, L., Geng, X., & Pan, Q. (2019). BP k NN: k-nearest neighbor classifier with pairwise distance metrics and belief function theory. *IEEE Access*, *7*, 48935–48947.

Jiao, L., Pan, Q., & Feng, X. (2015). Multi-hypothesis nearest-neighbor classifier based on class-conditional weighted distance metric. *Neurocomputing*, *151*, 1468–1476.

Jiao, L., Pan, Q., Feng, X., & Yang, F. (2013). An evidential k-nearest neighbor classification method with weighted attributes. In *Proceedings of the 16th international conference on information fusion* (pp. 145–150). IEEE.

Jing, L., Ng, M. K., & Huang, J. Z. (2007). An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Transactions on Knowledge & Data Engineering*, (8), 1026–1041.

Jing, L., Ng, M. K., Xu, J., & Huang, J. Z. (2005). Subspace clustering of text documents with feature weighting k-means algorithm. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 802–812). Springer.

John, G. H., & Langley, P. (1995). Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the eleventh conference on uncertainty in artificial intelligence* (pp. 338–345). Morgan Kaufmann Publishers Inc.

Jones, K. S., Walker, S., & Robertson, S. E. (2000). A probabilistic model of information retrieval: development and comparative experiments: Part 2. *Information Processing & Management*, *36*(6), 809–840.

Jović, A., Brkić, K., & Bogunović, N. (2015). A review of feature selection methods with applications. In *2015 38th international convention on information and communication technology, electronics and microelectronics* (pp. 1200–1205). IEEE.

Karypis, M. S. G., Kumar, V., & Steinbach, M. (2000). A comparison of document clustering techniques. In *TextMining Workshop At KDD2000 (May 2000)*.

Komosiński, M., & Krawiec, K. (2000). Evolutionary weighting of image features for diagnosing of CNS tumors. *Artificial Intelligence in Medicine*, *19*(1), 25–38.

Kononenko, I. (1994). Estimating attributes: analysis and extensions of RELIEF. In *European conference on machine learning* (pp. 171–182). Springer.

Kuo, R., & Nguyen, T. P. Q. (2019). Genetic intuitionistic weighted fuzzy k-modes algorithm for categorical data. *Neurocomputing*, *330*, 116–126.

Lan, M., Tan, C. L., Su, J., & Lu, Y. (2008). Supervised and traditional term weighting methods for automatic text categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(4), 721–735.

Lee, C.-H., Gutierrez, F., & Dou, D. (2011). Calculating feature weights in naive bayes with Kullback-Leibler measure. In *2011 IEEE 11th international conference on data mining* (pp. 1146–1151). IEEE.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., et al. (2018). Feature selection: A data perspective. *ACM Computing Surveys*, *50*(6), 94.

Li, Y., Luo, C., & Chung, S. M. (2012). Weighted naive Bayes for text classification using positive term-class dependency. *International Journal on Artificial Intelligence Tools*, *21*(01), Article 1250008.

Little, M. A., McSharry, P. E., Roberts, S. J., Costello, D. A., & Moroz, I. M. (2007). Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *Biomedical Engineering Online*, *6*(1), 23.

Liu, C., Xie, J., Zhao, Q., Xie, Q., & Liu, C. (2019). Novel evolutionary multi-objective soft subspace clustering algorithm for credit risk assessment. *Expert Systems with Applications*, *138*, Article 112827.

Marchiori, E. (2013). Class dependent feature weighting and k-nearest neighbor classification. In *IAPR international conference on pattern recognition in bioinformatics* (pp. 69–78). Springer.

Martineau, J. C., & Finin, T. (2009). Delta tfidf: An improved feature space for sentiment analysis. In *Third international AAAI conference on weblogs and social media*.

Mateos-García, D., García-Gutiérrez, J., & Riquelme-Santos, J. C. (2012). On the evolutionary optimization of k-NN by label-dependent feature weighting. *Pattern Recognition Letters*, *33*(16), 2232–2238.

Mateos-García, D., García-Gutiérrez, J., & Riquelme-Santos, J. C. (2017). On the evolutionary weighting of neighbours and features in the k-nearest neighbour rule. *Neurocomputing*.

Milligan, G. W., & Cooper, M. C. (1988). A study of standardization of variables in cluster analysis. *Journal of Classification*, *5*(2), 181–204.

Mohemmed, A. W., & Zhang, M. (2008). Evaluation of particle swarm optimization based centroid classifier with different distance metrics. In *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)* (pp. 2929–2932). IEEE.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems* (pp. 849–856).

Niño-Adan, I., Landa-Torres, I., Portillo, E., & Manjarres, D. (2019). Analysis and application of normalization methods with supervised feature weighting to improve K-means accuracy. In *International workshop on soft computing models in industrial and environmental applications* (pp. 14–24). Springer.

Ouyed, O., & Allili, M. S. (2020). Group-of-features relevance in multinomial kernel logistic regression and application to human interaction recognition. *Expert Systems with Applications*, *148*, Article 113247.

Palacio-Niño, J.-O., & Berzal, F. (2019). Evaluation metrics for unsupervised learning algorithms. arXiv preprint arXiv:1905.05667.

Paltoglou, G., & Thelwall, M. (2010). A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 1386–1395). Association for Computational Linguistics.

Panday, D., de Amorim, R. C., & Lane, P. (2018). Feature weighting as a tool for unsupervised feature selection. *Information Processing Letters*, *129*, 44–52.

Paredes, R., & Vidal, E. (2000). A class-dependent weighted dissimilarity measure for nearest neighbor classification problems. *Pattern Recognition Letters*, *21*(12), 1027–1036.

Paredes, R., & Vidal, E. (2006). Learning weighted metrics to minimize nearest-neighbor classification error. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (7), 1100–1110.

Parvin, H., & Minaei-Bidgoli, B. (2013). A clustering ensemble framework based on elite selection of weighted clusters. *Advances in Data Analysis and Classification*, *7*(2), 181–208.

Peng, L., Zhang, H., Yang, B., & Chen, Y. (2014). A new approach for imbalanced data classification based on data gravitation. *Information Sciences*, *288*, 347–373. http://dx.doi.org/10.1016/j.ins.2014.04.046.

Peng, L., Zhang, H., Zhang, H., & Yang, B. (2017). A fast feature weighting algorithm of data gravitation classification. *Information Sciences*, *375*, 54–78.

Phan, A. V., Le Nguyen, M., & Bui, L. T. (2017). Feature weighting and SVM parameters optimization based on genetic algorithms for classification problems. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *46*(2), 455–469.

Polat, K. (2012). Classification of Parkinson's disease using feature weighting method on the basis of fuzzy C-means clustering. *International Journal of Systems Science*, *43*(4), 597–609.

Portnoy, L. (2000). *Intrusion detection with unlabeled data using clustering* [Ph.D. thesis], Columbia University.

Raghu, S., & Sriraam, N. (2018). Classification of focal and non-focal EEG signals using neighborhood component analysis and machine learning algorithms. *Expert Systems with Applications*, *113*, 18–32.

Ren, Z., Wu, B., Sun, Q., & Wu, M. (2019). Simultaneous learning of reduced prototypes and local metric for image set classification. *Expert Systems with Applications*, *134*, 102–111.

Rendón, E., Abundez, I., Arizmendi, A., & Quiroz, E. M. (2011). Internal versus external cluster validation indexes. *International Journal of Computers and Communications*, *5*(1), 27–34.

Romeo, L., Loncarski, J., Paolanti, M., Bocchini, G., Mancini, A., & Frontoni, E. (2020). Machine learning-based design support system for the prediction of heterogeneous machine parameters in industry 4.0. *Expert Systems with Applications*, *140*, Article 112869.

Saaty, T. L. (2014). Analytic heirarchy process. *Wiley StatsRef: Statistics Reference Online*.

Saeys, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, *23*(19), 2507–2517.

Sáez, J. A., Derrac, J., Luengo, J., & Herrera, F. (2014). Statistical computation of feature weighting schemes through data estimation for nearest neighbor classifiers. *Pattern Recognition*, *47*(12), 3941–3948.

Saha, A., & Das, S. (2015). Categorical fuzzy k-modes clustering with automated feature weight learning. *Neurocomputing*, *166*, 422–435.

Sahin, E. K., Ipbuker, C., & Kavzoglu, T. (2015). A comparison of feature and expert-based weighting algorithms in landslide susceptibility mapping. *Procedia Earth and Planetary Science*, *15*, 462–467.

Sakar, C. O., & Kursun, O. (2010). Telediagnosis of Parkinson's disease using measurements of dysphonia. *Journal of Medical Systems*, *34*(4), 591–599.

San, O. M., Huynh, V.-N., & Nakamori, Y. (2004). An alternative extension of the k-means algorithm for clustering categorical data. *International Journal of Applied Mathematics and Computer Science*, *14*, 241–247.

Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., & Platt, J. C. (2000). Support vector method for novelty detection. In *Advances in neural information processing systems* (pp. 582–588).

Serrano-Silva, Y. O., Villuendas-Rey, Y., & Yáñez-Márquez, C. (2018). Automatic feature weighting for improving financial Decision Support Systems. *Decision Support Systems*, *107*, 78–87.

Shen, H., Yang, J., Wang, S., & Liu, X. (2006). Attribute weighted mercer kernel based fuzzy clustering algorithm for general non-spherical datasets. *Soft Computing*, *10*(11), 1061–1073.

Sinciya, P., & Celin, J. J. A. (2017). Weight optimized gravitational classifier for high dimensional numerical data classification. *International Journal of Pure and Applied Mathematics*, *116*(22), 251–263.

Sotoodeh, M., Moosavi, M. R., & Boostani, R. (2019). A novel adaptive LBP-based descriptor for color image retrieval. *Expert Systems with Applications*, *127*, 342–352.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, *28*(1), 11–21.

Sun, Y. (2007). Iterative RELIEF for feature weighting: algorithms, theories, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(6), 1035–1051.

Taheri, S., Yearwood, J., Mammadov, M., & Seifollahi, S. (2014). Attribute weighted naive Bayes classifier using a local optimization. *Neural Computing and Applications*, *24*(5), 995–1002.

Tahir, M. A., Bouridane, A., & Kurugollu, F. (2007). Simultaneous feature selection and feature weighting using Hybrid Tabu Search/K-nearest neighbor classifier. *Pattern Recognition Letters*, *28*(4), 438–446.

Triguero, I., Derrac, J., GarcíA, S., & Herrera, F. (2012). Integrating a differential evolution feature weighting scheme into prototype generation. *Neurocomputing*, *97*, 332–343.

Tsai, C.-Y., & Chiu, C.-C. (2008). Developing a feature weight self-adjustment mechanism for a K-means clustering algorithm. *Computational Statistics & Data Analysis*, *52*(10), 4658–4672.

Tzortzis, G., & Likas, A. (2014). The MinMax k-means clustering algorithm. *Pattern Recognition*, *47*(7), 2505–2516.

Venkatesh, B., & Anuradha, J. (2019). A review of feature selection and its methods. *Cybernetics and Information Technologies*, *19*(1), 3–26.

Wang, Y., & Chen, L. (2017). Multi-view fuzzy clustering with minimax optimization for effective clustering of data from multiple sources. *Expert Systems with Applications*, *72*, 457–466.

Wang, S., Jiang, L., & Li, C. (2014). A cfs-based feature weighting approach to naive Bayes text classifiers. In *International conference on artificial neural networks* (pp. 555–562). Springer.

Wang, X., Wang, Y., & Wang, L. (2004). Improving fuzzy c-means clustering based on feature-weight learning. *Pattern Recognition Letters*, *25*(10), 1123–1132.

Wei, P., Lu, Z., & Song, J. (2015). Variable importance analysis: a comprehensive review. *Reliability Engineering & System Safety*, *142*, 399–432.

Wemmert, C., Gançarski, P., & Korczak, J. J. (2000). A collaborative approach to combine multiple learning methods. *International Journal on Artificial Intelligence Tools*, *9*(01), 59–78.

Wettschereck, D., & Aha, D. W. (1995). Weighting features. In *International conference on case-based reasoning* (pp. 347–358). Springer.

Wettschereck, D., Aha, D. W., & Mohri, T. (1997). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, *11*(1–5), 273–314.

Wettschereck, D., & Dietterich, T. G. (1995). An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms. *Machine Learning*, *19*(1), 5–27.

Wu, J., & Cai, Z. (2011). Attribute weighting via differential evolution algorithm for attribute weighted naive bayes (wnb). *Journal of Computational Information Systems*, *7*(5), 1672–1679.

Wu, H., Gu, X., & Gu, Y. (2017). Balancing between over-weighting and under-weighting in supervised term weighting. *Information Processing & Management*, *53*(2), 547–557.

Wu, J., Pan, S., Zhu, X., Cai, Z., Zhang, P., & Zhang, C. (2015). Self-adaptive attribute weighting for naive Bayes classification. *Expert Systems with Applications*, *42*(3), 1487–1502.

Wu, H., & Salton, G. (1981). A comparison of search term weighting: term relevance vs. inverse document frequency. In *Proceedings of the 4th annual international ACM SIGIR conference on information storage and retrieval: Theoretical issues in information retrieval* (pp. 30–39).

Xia, H., Zhuang, J., & Yu, D. (2013). Novel soft subspace clustering with multi-objective evolutionary approach for high-dimensional data. *Pattern Recognition*, *46*(9), 2562–2575.

Xiang, Z.-L., Yu, X.-R., & Kang, D.-K. (2016). Experimental analysis of naïve Bayes classifier based on an attribute weighting framework with smooth kernel density estimations. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *44*(3), 611–620.

Xing, H.-j., Ha, M.-h., Hu, B.-g., & Tian, D.-z. (2009). Linear feature-weighted support vector machine. *Fuzzy Information and Engineering*, *1*(3), 289–305.

Yang, W., Wang, K., & Zuo, W. (2012). Neighborhood component feature selection for high-dimensional data. *Journal of Computational Physics*, *7*(1), 161–168.

Yeung, D.-Y., & Chow, C. (2002). Parzen-window network intrusion detectors. In *Object recognition supported by user interaction for service robots, vol. 4* (pp. 385–388). IEEE.

Yilmaz, T., Yazici, A., & Kitsuregawa, M. (2014). RELIEF-MM: effective modality weighting for multimedia information retrieval. *Multimedia Systems*, *20*(4), 389–413.

Yip, K. Y., Cheung, D. W., & Ng, M. K. (2004). Harp: A practical projected clustering algorithm. *IEEE Transactions on Knowledge and Data Engineering*, *16*(11), 1387–1397.

Zaidi, N. A., Cerquides, J., Carman, M. J., & Webb, G. I. (2013). Alleviating naive Bayes attribute independence assumption by attribute weighting. *Journal of Machine Learning Research*, *14*(1), 1947–1988.

Zhang, L., Jiang, L., Li, C., & Kong, G. (2016). Two feature weighting approaches for naive Bayes text classifiers. *Knowledge-Based Systems*, *100*, 137–144.

Zhang, H., & Sheng, S. (2004). Learning weighted naive Bayes with accurate ranking. In *Fourth IEEE international conference on data mining* (pp. 567–570). IEEE.

Zhang, G.-Y., Wang, C.-D., Huang, D., Zheng, W.-S., & Zhou, Y.-R. (2018). TW-Co-k-means: two-level weighted collaborative k-means for multi-view clustering. *Knowledge-Based Systems*, *150*, 127–138.

Zhi, X.-b., Fan, J.-l., & Zhao, F. (2014). Robust local feature weighting hard c-means clustering algorithm. *Neurocomputing*, *134*, 20–29.

Zhou, J., Chen, L., Chen, C. P., Zhang, Y., & Li, H.-X. (2016). Fuzzy clustering with the entropy of attribute weights. *Neurocomputing*, *198*, 125–134.

Zhou, S., Qian, S., Chang, W., Xiao, Y., & Cheng, Y. (2018). A novel bearing multi-fault diagnosis approach based on weighted permutation entropy and an improved SVM ensemble classifier. *Sensors*, *18*(6), 1934.

Zhou, Z., & Zhu, S. (2018). Kernel-based multiobjective clustering algorithm with automatic attribute weighting. *Soft Computing*, *22*(11), 3685–3709.