Consensus in sequential wrapper feature selection: a unifying approach

L. Martino[⋆], G. Villacrés[◇], S. Arcidiacono[⋆]

[⋆] Università degli studi di Catania, Catania (Italy).

[◇] Universidad Rey Juan Carlos, Madrid (Spain).

Abstract

Feature selection is a crucial task in statistics and machine learning, with direct implications for model interpretability and computational efficiency. This study introduces a unifying approach that combines the four possible sequential wrapper methods employed for variable selection, aiming to exploit their complementary strengths. The proposed procedure computes feature relevance scores and, subsequently, integrates the outputs from each sequential wrapper method. The underlying idea is simple and efficient. We test it in a controlled experiment with a known ground truth. The results indicate that the ranking obtained by consensus clearly outperform the individual rankings obtained by the wrapper methods.

Keywords: Variable selection; wrapper methods; sequential model selection; feature importance

1 Introduction

In modern statistical modeling and machine learning, feature selection, i.e., the process of identifying relevant variables, is a fundamental task [1, 2, 3]. Isolating informative features enhances model interpretability, mitigates overfitting, reduces computational burden, and uncovers meaningful patterns within the data. These advantages are particularly valuable across diverse domains such as bioinformatics, finance, and environmental modeling, to name a few.

Feature selection theoretically involves two distinct (and possibly separate) steps: the ranking of candidate features based on a defined criterion, and determining an appropriate effective number of variables to retain (i.e., the actual size of subset of features) [4, 5, 6, 7]. In this work, we focus on the first part, specifically ranking the variables. The ranking methods are commonly classified into three main categories: (a) filter, (b) embedded (or intrinsic), and (c) wrapper techniques [8, 9, 10]. Filter methods rank or exclude features using statistical metrics (e.g., correlation, mutual information, or univariate tests) independently of the learning algorithm or the specific task to solve [11]. Embedded methods are internal selection schemes within a specific algorithm: for example, through penalization techniques like the LASSO [12], the use of Gini impurity index

in decision tree-based algorithms [13], or the use of the automatic relevance determination (ARD) kernel in a Gaussian process [14] (more specifically, see [15, Sections 5.4 and 5.5]). Clearly, embedded (intrinsic) methods require the use of the specific learning algorithm in which they are integrated. Wrapper methods, on the other hand, assess subsets of features by training and validating a model, selecting those combinations that optimize a predefined performance metric [6, 16, 17, 18]. Wrapper methods can be employed for different tasks and different algorithms. A famous example of non-sequential wrapper technique is the so-called "leave one covariate out" (LOCO) [19]. Ensemble feature selection techniques have been proposed to combine the outputs of multiple methods, improving robustness and stability in variable selection [20, 1, 21]. Closely related to the feature rankings is the concept of variable importance [22], as the well-known Shapley values, based on a game-theoretic approach [23]. Other ideas have also been explored in the literature [24, 25, 26].

In this work, we focus on the four possible sequential wrapper methods for variable selection [6, 17]. Sequential wrapper schemes are forward and backward procedures such that features are incrementally added or removed based on their contribution to model performance, as evaluated by a chosen criterion. These methods are compatible with any regression or classification model and allow for different performance evaluation criteria to be employed. Even if the four procedures (two forward and two backward) are theoretically similar, the resulting rankings are generally different. In this work, we introduce a procedure aimed at reaching consensus among the differing results. Hence, the proposed procedure can also be interpreted as a unifying approach of the four sequential wrapper methods. The procedure relies on constructing an importance measure within the sequential framework, determined by variations (increase or decrease) in the chosen performance metric. Furthermore, the novel method is able to provide a measure of uncertainty of each aggregated importance value. The resulting consensus outperforms the individual rankings provided by the four sequential wrapper schemes, according to two different scores (exact match and Kendall correlation). These results indicate that the proposed strategy leverages the complementary advantages of each scheme in an effective and balanced way.

2 Framework and notation

Given a set of R input variables/features $\mathbf{x} = [x_1, ..., x_R]^\top$ (forming an input vector) and a related output variable y, in many applications, we observe a set of N data pairs $\{\mathbf{x}_n, y_n\}_{n=1}^N$ where $\mathbf{x}_n = [x_{n,1}, ..., x_{n,R}]^\top$, and $R \leq N$. Hence we have a $N \times R$ matrix of inputs,

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^\top = \begin{bmatrix} x_{1,1} & x_{1,2} & ... & x_{1,R} \\ x_{2,1} & x_{2,2} & ... & x_{2,R} \\ ... & & & \\ x_{N,1} & x_{N,2} & ... & x_{N,R} \end{bmatrix} = [\mathbf{z}_1, \dots, \mathbf{z}_R], \tag{1}$$

where we set

$$\mathbf{z}_r = [x_{1,r}, ..., x_{N,r}]^\top,$$

that represents the $N \times 1$ column vector with all the components of r-the variable/feature, with $r = 1, ..., R$. The dataset is then generally formed by $\mathbf{X}$ and the vector of outputs $\mathbf{y} = [y_1, ..., y_N]^\top$.

We assume that the relationship between inputs and outputs can be approximated by a parametric model, i.e., $\widehat{y}_n = g_n(\mathbf{x}_n; \widehat{\boldsymbol{\beta}})$, and in vectorial form

$$\widehat{\mathbf{y}} = \mathbf{g}(\mathbf{X}; \widehat{\boldsymbol{\beta}}) = \mathbf{g}([\mathbf{z}_1, ..., \mathbf{z}_R]; \widehat{\boldsymbol{\beta}}). \tag{2}$$

The vector of the estimated parameters is $\widehat{\boldsymbol{\beta}} = [\widehat{\beta}_0, \widehat{\beta}_1, ..., \widehat{\beta}_R]^\top$, and $\mathbf{g}(\mathbf{X}; \widehat{\boldsymbol{\beta}}) = [g_1(\mathbf{x}_1; \widehat{\boldsymbol{\beta}}), ..., g_N(\mathbf{x}_N; \widehat{\boldsymbol{\beta}})]^\top$. Considering all data for training and all the features, according to some internal procedure and/or some internal cost function $C_{\texttt{int}}(\mathbf{y}, \mathbf{z}) : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$, we can optimize the vector of parameters,

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} C_{\texttt{int}}(\mathbf{y}, \mathbf{g}(\mathbf{X}; \boldsymbol{\beta})). \tag{3}$$

The internal cost function $C_{\texttt{int}}(\mathbf{y}, \mathbf{z})$ determines the type of inference method employed. In some parts of the text below, we also use the notation $\widehat{\mathbf{y}}_0 = \mathbf{g}(\mathbf{X}; \widehat{\boldsymbol{\beta}}_0)$.

2.1 External evaluation of the model performance

For our purpose of building feature ranking, we have an additional degree of freedom given by an external evaluation measure of the model performance, which is independent of the internal procedure used to learn the optimal parameter vector $\widehat{\boldsymbol{\beta}}$. Considering the vector of observed outputs $\mathbf{y}$ and the predicted outputs $\widehat{\mathbf{y}} = \mathbf{g}(\mathbf{X}; \widehat{\boldsymbol{\beta}})$, we define a general external cost function $C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}) : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$. For instance, in regression, we can consider

$$C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}) = \frac{1}{N} \sum_{n=1}^{N} |y_n - \widehat{y}_n|^\alpha, \tag{4}$$

or in classification, we can consider the cross-entropy, i.e.,

$$C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}) = -\frac{1}{N} \sum_{n=1}^{N} \left[y_n \log \widehat{y}_n + (1 - y_n) \log(1 - \widehat{y}_n) \right]. \tag{5}$$

An interesting point is that $C_{\texttt{ext}}$ can differ from $C_{\texttt{int}}$. In certain circumstances, this flexibility can be useful in increasing the robustness of the obtained results.

3 Sequential wrapper methods for variable selection

We need to define a trained model considering only a subset of $m < R$ features, i.e.,

$$\widehat{\mathbf{y}}_m = \mathbf{g}([\mathbf{z}_{k_1}, \mathbf{z}_{k_2} ..., \mathbf{z}_{k_M}]; \widehat{\boldsymbol{\beta}}_m), \tag{6}$$

where $k_i \in \{1, ..., R\}$, $k_j \neq k_i$ (for $i \neq j$), and $\widehat{\boldsymbol{\beta}}_m = [\beta_0, \beta_{k_1}, ..., \beta_{k_m}]^\top$. Below, we describe techniques that sequentially add or remove a variable/feature at each iteration, then consider a subset of inputs (let's say $m < R$) in the input matrix. Note that where $\widehat{\mathbf{y}}_m$ is the $N \times 1$ prediction vector when m features are included in the input matrix.

3.1 Wrapper ranking methods

In this section, we describe the sequential wrapper methods employed and described in the literature, where one variable is added or removed at each iteration. There are four different ranking methods denoted as RM1, RM2, RM3, and RM4. In the description of the methods, we always consider a final ranking in a decreasing order of importance.

3.1.1 RM1: *Forward selection adding the best variable, minimizing the external cost*

This method sequentially adds one feature at a time to the sub-models and in the final ranking. At each step, the feature that achieves the best performance in terms of minimizing the cost function $C_{\texttt{ext}}$ is selected as the best one at that step. Thus, RM1 adds a most significant feature at each step. At the beginning, the method considers all the possible sub-models with only one feature. The feature that obtains the best prediction performance (in terms of external cost) will be the most significant, and will be denoted as $\mathbf{z}_{k_1}$. Namely, $\mathbf{z}_{k_1}$ is the variable that obtains the smallest $C_{\texttt{ext}}$ when is included. Fixing $\mathbf{z}_{k_1}$ and considering a model with bi-dimensional input (i.e., including 2 features), at the second step, RM1 selects another relevant feature, denoted as $\mathbf{z}_{k_2}$. At the m-th step, fixing the previous $m-1$ selected features, RM1 selects the most significant feature as $\mathbf{z}_{k_m}$, and so on. The sequence of models and predictions would be

$$\begin{aligned} &\widehat{\mathbf{y}}_1 = \mathbf{g}(\mathbf{z}_{k_1}; \widehat{\boldsymbol{\beta}}_1), \quad \widehat{\mathbf{y}}_2 = \mathbf{g}([\mathbf{z}_{k_1}, \mathbf{z}_{k_2}]; \widehat{\boldsymbol{\beta}}_2), ..., \\ &\widehat{\mathbf{y}}_m = \mathbf{g}([\mathbf{z}_{k_1}, \mathbf{z}_{k_2}, ..., \mathbf{z}_{k_m}]; \widehat{\boldsymbol{\beta}}_m), \text{ etc.} \end{aligned} \tag{7}$$

The final ranking (in a decreasing order of importance) would be $\mathbf{z}_{k_1}, \mathbf{z}_{k_2}, \ldots, \mathbf{z}_{k_R}$ (where $\mathbf{z}_{k_1}$ is the best and $\mathbf{z}_{k_R}$ is the worst).

3.1.2 RM2: *Backward elimination removing the worst variable, minimizing the external cost*

This method removes one feature at each step in a sequential way. RM2 removes the least significant feature at each step. Namely, at the beginning, RM2 includes all R features in the model and removes the least significant feature, denoted as $\mathbf{z}_{k_1}$, that is the variable that produces the smallest decrease in $C_{\texttt{ext}}$ when it is removed (this would be the worst variable). Fixing the remaining $R-1$ features, RM2 removes the next feature denoted as $\mathbf{z}_{k_2}$, reducing the model to $R-2$ features. At the m-th step, RM2 removes the feature that contributes the least to the minimization of the external cost, among the $R-m$ remaining features, denoted as $\mathbf{z}_{k_m}$, and so on. The last remaining variable will be the most important one. Namely, the dimension of the input space decreases at each step of the process,

$$\begin{aligned} &\widehat{\mathbf{y}}_0 = \mathbf{g}(\mathbf{X}; \widehat{\boldsymbol{\beta}}_0), \quad \widehat{\mathbf{y}}_1 = \mathbf{g}(\mathbf{X}_{-k_1}; \widehat{\boldsymbol{\beta}}_1), \\ &\widehat{\mathbf{y}}_2 = \mathbf{g}(\mathbf{X}_{-k_1,-k_2}; \widehat{\boldsymbol{\beta}}_2), \text{ etc.} \end{aligned} \tag{8}$$

Note that we have use the notation

$$\mathbf{X}_{-k_1,-k_2} = [\mathbf{z}_1, ..., \mathbf{z}_{k_1-1}, \mathbf{z}_{k_1+1}, ..., \mathbf{z}_{k_2-1}, \mathbf{z}_{k_2+1}, ..., \mathbf{z}_R].$$

The final ranking (in decreasing order of importance) would be $\mathbf{z}_{k_R}, \mathbf{z}_{k_{R-1}}, \ldots, \mathbf{z}_{k_1}$ (where $\mathbf{z}_{k_R}$ is the most relevant feature, and $\mathbf{z}_{k_1}$ represents the least relevant feature).

3.1.3 RM3: *Backward elimination removing the best variable, maximizing the external cost*

As RM2, this method also removes one feature at a time, but now it removes the variable that maximizes the cost function $C_{\texttt{ext}}$. The method starts again, considering the complete model as RM2. Then, RM3 removes the variable that produces the highest increase in the cost $C_{\texttt{ext}}$, i.e., the prediction error. This variable would be the most important. RM3 continues sequentially removing variables that maximize the cost function $C_{\texttt{ext}}$. We denote the first variable removed as $\mathbf{z}_{k_1}$ (the most relevant variable), the second removed variable $\mathbf{z}_{k_2}$, and so on. The final ranking (in decreasing order of importance) is $\mathbf{z}_{k_1}, \mathbf{z}_{k_2}, \ldots, \mathbf{z}_{k_R}$.

3.1.4 RM4: *Forward selection adding the worst variable, maximizing the external cost.*

This method is similar to RM1, but instead of adding the most significant feature at each step, RM4 adds the feature that achieves the worst performance, i.e., the feature that maximizes the external cost function $C_{\texttt{ext}}$ when it is included. Namely, at each step, RM4 selects the feature that causes the largest increase in the cost function (that is, the worst variable). At the initial step, RM4 selects the feature, denoted as $\mathbf{z}_{k_1}$, that maximizes $C_{\texttt{ext}}$ (then $\mathbf{z}_{k_1}$ is the worst variable). Fixing $\mathbf{z}_{k_1}$, RM4 then selects the next feature with a model of dimension 2, denoted as $\mathbf{z}_{k_2}$, which again maximizes $C_{\texttt{ext}}$. Note that RM4 creates a sequence of variables from the worst to the best variable (i.e., increasing their relevance). The final ranking (in decreasing order of importance) would be $\mathbf{z}_{k_R}, \mathbf{z}_{k_{R-1}}, \ldots, \mathbf{z}_{k_1}$.

4 Agreement on a aggregated ranking

4.1 A measure of importance associated with the features

Employing these wrapper methods, we can also compute a scalar quantity that measures the importance of each variable/feature [24]. For the sake of simplicity, we consider a regression problem where we use $C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}) = \frac{1}{N}\sum_{n=1}^{N}(y_n - \widehat{y}_n)^2$. Note that if we are employing a proper regressor, $\widehat{\mathbf{y}} = \mathbf{g}(\mathbf{X}; \widehat{\boldsymbol{\beta}})$, we have $C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}) \leq \text{var}[\mathbf{y}]$, since the simplest possible predictive method consists on setting $\widehat{y}_n = \frac{1}{N}\sum_{n=1}^{N} y_n$ for all $n = 1, ..., N$. Note that in a binary classification problem, we have $C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}) \leq 0.5$.

Forward schemes. Using RM1 and RM4, we can associate a measure of importance to the j-th selected variable, computing a weight w_{k_j} that is equal to the reduction to prediction error obtained by adding the j-th selected variable, i.e.,

$$\begin{aligned}
\rho_{k_1} &= \text{var}[\mathbf{y}] - C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}_1), \\
&\vdots \\
\rho_{k_j} &= C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}_{j-1}) - C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}_j), \\
&\vdots \\
\rho_{k_R} &= C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}_{R-1}) - C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}_R),
\end{aligned}$$

where the predictions $\widehat{\mathbf{y}}_j$ are obtained as in Eq.(7). Note that all the weights are non-negative, i.e., $\rho_{k_j} \geq 0$, by construction (except for numerical issues). In case of a binary classification problem, the first value ρ_{k_1} would be $\rho_{k_1} = 0.5 - C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}_j)$, where 0.5 represents the performance of a random classifier.

Backward schemes. Using RM2 and RM3, the weights can computed as

$$\begin{aligned}
\rho_{k_1} &= C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}_1) - C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}_0), \\
&\vdots \\
\rho_{k_j} &= C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}_j) - C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}_{j-1}), \\
&\vdots \\
\rho_{k_R} &= C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}_R) - C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}_{R-1}),
\end{aligned}$$

where the predictions $\widehat{\mathbf{y}}_j$ are obtained as in Eq.(8). Note that, again, all the weights are non-negative, i.e., $\rho_{k_j} \geq 0$, by construction (except for numerical issues).

Remark. Note that in both scenarios, forward and backward schemes, the computation of the importance values requires the implementation of the sequential procedure.

4.2 Finding a suitable consensus

In this section, we describe how to obtain a unique ranking aggregating (i.e., combining) the results provided by the different RMs. After applying the $M = 4$ different RMs described above, from each of them we obtain the importance measures, $\rho_{m,1}, \rho_{m,2}, ..., \rho_{m,R}$, with $m = 1, ..., M = 4$. Hence for each variable, we can define the vector that contains all the corresponding importance values, $[\rho_{1,j}, ..., \rho_{M,j}]^\top$. Then, the simplest way to combine these importance values is to compute the aggregated importance:

$$I_j = \frac{1}{M} \sum_{m=1}^{M} \rho_{j,m}. \tag{9}$$

for each feature, i.e., $j = 1, ..., R$. Thus, we can order the final importance values in decreasing order,

$$I_{k_1} \geq I_{k_2} \geq .... \geq I_{k_R}, \tag{10}$$

where $k_j \in \{1, ..., R\}$ is an index representing the final position of the j-the feature in the final ranking. Namely, the final aggregated ranking is (in decreasing order of importance), $\mathbf{z}_{k_1}, \mathbf{z}_{k_2}, \ldots, \mathbf{z}_{k_R}$. A measure of uncertainty can also be computed as

$$\text{var}(j) = \frac{1}{M} \sum_{m=1}^{M} (\rho_{j,m} - I_j)^2. \tag{11}$$

Finally, note that we have considered four wrapper methods (i.e., $M = 4$), but the procedure above is valid for any positive integer value, $M \geq 1$.

5 Numerical experiments

We generate a synthetic dataset to evaluate the performance of the different RMs described in Section 3.1 under controlled conditions, i.e., knowing the ground-truth. The data structure follows a linear model with specific variables included and excluded based on predefined criteria. Specifically, the dataset consists of $N = 5000$ observations and $R = 20$ variables/features, $\mathbf{x} = [x_1, \ldots, x_{20}]$, and the variables are defined as shown in Table 1.

Table 1: Feature generation: sampling from a distribution

Variables	Generation / Distribution
x_1, x_2, x_5 x_7, x_{15}, x_{16}, x_{18}, x_{19}	$\mathcal{N}(0,1)$
x_3, x_4, x_8 x_9, x_{10}, x_{13}, x_{20}	$\mathcal{U}\left(\left[-\frac{\sqrt{12}}{2}, \frac{\sqrt{12}}{2}\right]\right)$
x_6	x_2^2
x_{11}	$z = 0.5x_8 + \epsilon, \quad \epsilon \sim \mathcal{N}(0,1), \quad \frac{z - \text{mean}(z)}{\text{std}(z)}$
x_{12}	$z = 0.5x_{10} + \epsilon, \quad \epsilon \sim \mathcal{N}(0,1), \quad \frac{z - \text{mean}(z)}{\text{std}(z)}$
x_{14}	$z = x_5 + \epsilon, \quad \epsilon \sim \mathcal{N}(0,1), \quad \frac{z - \text{mean}(z)}{\text{std}(z)}$
x_{17}	$z = 0.2x_2 + u, \quad u \sim \mathcal{U}([0,1]), \quad \frac{z - \text{mean}(z)}{\text{std}(z)}$

Note that for all the input variables we have $\mathbb{E}[x_i] = 0$ and $\text{var}[x_i] = 1$, so that they are all normalized in terms of the power of the signal.

True model. The corresponding observations were generated as follows

$$\begin{aligned} y_n =& 0.6x_2 + 0.6x_3 - 0.2x_4 + 0.1x_5 - 0.3x_7 + 0.1x_8 \\ &+ 0.8x_9 - 0.3x_{11} + 0.3x_{12} + 0.3x_{14} + 0.5x_{15} + 0.9x_{16} \\ &+ 0.2x_{17} - 0.3x_{18} - 0.5x_{19} + 0.6x_{20} + \epsilon_n. \end{aligned} \tag{12}$$

ϵ_n is a Gaussian noise perturbation with zero mean and variance $\sigma^2 = 0.1$. Note that the model above in Eq. (12) excludes explicitly the following features:

$$x_1, \ x_6, \ x_{10}, \text{ and } x_{13}.$$

However, x_6 is included as a transformation of x_2, i.e., $x_6 = x_2^2$. Moreover, some variables present linear correlation: x_8 and x_{11}, x_{10} and x_{12}, x_5 and x_{14}, x_2 and x_{17}. Indeed, x_{11}, x_{12}, x_{14}, and x_{17} are obtained with a linear transformation of another variable plus noise as shown in Table

1. Furthermore, some variables have identical coefficients in the true model but have different distributions: x_2 and x_3 share the same regression coefficient but follow different distributions. Similarly, x_7 and x_9 have identical coefficients but have been drawn from different distributions.

Regression model. As internal model $\mathbf{g}(\mathbf{X};\boldsymbol{\beta})$, we use the true generative model as a regression model, i.e.

$$\mathbf{y} = \mathbf{g}(\mathbf{X};\boldsymbol{\beta}) + \boldsymbol{\epsilon} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{13}$$

where $\boldsymbol{\epsilon}$ is a Gaussian noise perturbation with zero mean and diagonal covariance matrix with elements $\sigma^2 = 0.1$ on the diagonal. Considering $C_{\texttt{int}}(\mathbf{y}, \mathbf{g}(\mathbf{X};\boldsymbol{\beta})) = \sum_{n=1}^{N}(y_n - g_n(\mathbf{X};\boldsymbol{\beta}))^2$, the optimal solution $\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} C_{\texttt{int}}(\mathbf{y}, \mathbf{g}(\mathbf{X};\boldsymbol{\beta}))$ is given by

$$\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{y}, \quad \text{and} \quad \widehat{\mathbf{y}} = \mathbf{X}\widehat{\boldsymbol{\beta}} = \mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{y}. \tag{14}$$

To evaluate the model performance, we employ $C_{\texttt{ext}}(\mathbf{y}, \widehat{\mathbf{y}}) = \frac{1}{N}\sum_{n=1}^{N}(y_n - \widehat{y}_n)^2$. Note that the predictive model is linear, as is the true model. Then, we remove the issue of model mismatch, and we can focus on the RMs and their built consensus.

Ground-truth. We assume that the true importance of a feature is given by the absolute value of its coefficient in the true linear model presented in Eq. (12). Based on this definition, the variables (identified by their sub-indices) are ranked in descending order of importance, which is determined by sorting the absolute values of the coefficients from largest to smallest. The most important is x_{16}, the second most important is x_9, and so on. Note that there are several ties.

	Ground-truth												
Pos	1^{th}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}	11^{th}	12^{th}	
	16	9	(2, 3, 20)			(15, 19)		(7, 11, 12, 14, 18)					
Pos	13^{th}	14^{th}	15^{th}	16^{th}	17^{th}	18^{th}	19^{th}	20^{th}	——				
	(4, 17)		(5, 8)		(1, 6, 10, 13)				——				

The indices within the parentheses $(\cdot, \cdots, \cdot)$ indicate ties in the ranking, meaning the variables inside the parentheses have the same importance in the model. Any permutation of these variables is considered a correct ranking.

Scoring the ranking methods. To evaluate the RMs, we employ the exact match scoring and Kendall correlation. Specifically, regarding the exact match, let us define S as the score for a given ranking method. Each time that a variable is placed in the ranking in a correct position, the score of the ranking is increased by 1, i.e., $S \leftarrow S+1$. Otherwise, the score remains invariant. Note that as a correct position, we also consider the draw positions (i.e., ties): for instance, variable 2 is considered well placed in the third, fourth, or fifth position. In Table 2, we consider the total score based on the exact match over all 20 variables. The Kendall correlation among the positions of the ground-truth and obtained rankings is also applied [27]. The Kendall correlation has the advantage that it takes into account if wrong (erroneous) positions in the analyzed RM are close to the right position in the ground-truth, or if they are far away.

Results. We present the results obtained from the different RMs and their consensus obtained as in Section 4.2. Table 2 summarizes the rankings across RMs and their consensus. By examining these rankings, we can observe how similar or different the ranking methods are in prioritizing features. The table shows only the indices of the corresponding variable, e.g., 16 corresponds to x_{16}. The total and partial scores for the exact match scoring are numerical values derived from the procedure described above. The maximum value is 20 for the score match, achieved when all features are ranked correctly, and the minimum possible score is 0. For the Kendall correlation, the maximum possible value is 1, reached if all features are ranked correctly.

Table 2: Rankings by RMs and the consensus, in decreasing order of importance. For instance, the feature x_{16} is the most relevant for all the rankings. The errors in the consensus ranking are highlighted in boxes, whereas the best scores are highlighted in bold.

RMs	**Ranking**	**Score**	**Kendall**
RM1	16 9 2 3 20 15 18 14 12 7 17 11 4 8 5 10 1 6 13 19	15	0.88
RM2	16 9 2 3 20 15 19 14 12 7 17 11 4 8 5 10 1 6 13 18	**16**	0.92
RM3	16 9 2 20 3 15 17 12 7 14 5 11 4 10 8 6 13 1 18 19	14	0.81
RM4	16 9 2 20 3 15 19 18 14 5 17 12 7 11 4 10 8 6 13 1	13	0.94
Consensus	16 9 2 3 20 15 $\boxed{14}$ $\boxed{19}$ 18 12 7 $\boxed{17}$ $\boxed{11}$ 4 8 5 10 1 13 6	**16**	**0.98**
	Maximum possible score values:	20	1.00

In Table 2, we observe that all RMs correctly rank the first six variables (considering the ties). There is a clear agreement among all the RMs on at least the first three most important variables, 16, 9, and 2. For the fourth and fifth positions, they agree to choose 3 and 20, and in the sixth position, the variable 15. RM2 ranks the top 10 variables correctly. RM1 also provides a high match score. RM4 is the best wrapper method in terms of the Kendall correlation.
The consensus obtained as in Section 4.2 commits 4 errors (score match of 16) but the errors are very "close": switching the position of x_{14}, x_{19} and x_{17} and x_{11} the consensus would obtain the maximum score match of 20. However, the score match of 16 is the maximum value obtained by the RM2 and the consensus ranking. The Kendall correlation easily detects the benefits of RM4 and also of the consensus ranking (due to the proximity of the errors). The maximum value of Kendall's correlation, very close to 1, is achieved by the consensus ranking.
RM2 and RM4 are the unique methods able to properly rank x_{19}, but the consensus also gets close (only for one position). The feature x_{18} is well-ranked only by RM4 and consensus. Note that both x_{18} and x_{19} have negative coefficients in the model. Hence, the consensus ranking provides the best scores and is able to combine properly the behaviors of the 4 different RMs.

6 Conclusions

In this work, we have proposed a procedure to achieve a consensus among the four possible sequential wrapper methods for feature selection. This provides a unified perspective on the four sequential wrapper techniques, all of which operate by iteratively including or excluding one

variable at a time. The introduced approach involves developing an importance metric within the sequential process, based on variations (increases or decreases) in the chosen performance metric. The method is simple yet effective, as it combines the strengths and characteristics of the individual wrapper approaches. Furthermore, the novel method provides an uncertainty quantification of each aggregated importance value. Numerical experiments have demonstrated that the resulting consensus consistently outperforms each individual method.

References

[1] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 08 2007.

[2] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Miguez, and P. M. Djuric. Adaptive importance sampling: the past, the present, and the future. *IEEE Signal Processing Magazine*, 34(4):60–79, 2017.

[3] Thinzar Saw and Si Si Mar Win. Feature ranking based dimensional reduction algorithm in high dimensional data classification. In *2023 IEEE Conference on Computer Applications (ICCA)*, pages 300–305, 2023.

[4] Huanjing Wang, Qianxin Liang, John T Hancock, and Taghi M Khoshgoftaar. Feature selection strategies: a comparative analysis of shap-value and importance-based methods. *Journal of Big Data*, 11(1):44, 2024.

[5] R. L. Thorndike. Who belongs in the family? *Psychometrika*, 3:267–276, 1953.

[6] R. San Millán-Castillo, L. Martino, and E. Morgado. A variable selection analysis for soundscape emotion modelling using decision tree regression and modern information criteria. *IEEE Access*, 2024.

[7] E. Morgado, L. Martino, and R. San Millán-Castillo. Universal and automatic elbow detection for learning the effective number of components in model selection problems. *Digital Signal Processing*, 140:104103, 2023.

[8] I. M. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, 2003.

[9] Xueyi Cheng. A comprehensive study of feature selection techniques in machine learning models. *Artificial Intelligence and Digital Technology*, 1(1):65–78, Nov. 2024.

[10] Huan Liu and Hiroshi Motoda. *Computational Methods of Feature Selection*. CRC Press, October 2007. Publisher Copyright: © 2007 by Taylor & Francis Group, LLC. All rights reserved.

[11] Andrea Bommert, Xudong Sun, Bernd Bischl, Jörg Rahnenführer, and Michel Lang. Benchmark for filter methods for feature selection in high-dimensional classification data. *Computational Statistics & Data Analysis*, 143:106839, 2020.

[12] R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[13] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and Regression Trees*. Taylor & Francis, 1984.

[14] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[15] L. Martino, V. Elvira, and G. Camps-Valls. The recycling Gibbs sampler for efficient learning. *Digital Signal Processing*, 74:1–13, 2018.

[16] Josef Kittler. Feature set search algorithms. *Pattern recognition and signal processing*, 1978.

[17] Xiaojuan Huang, Li Zhang, Bangjun Wang, Fanzhang Li, and Zhao Zhang. Feature clustering based support vector machine recursive feature elimination for gene selection. *Applied Intelligence*, 48:594–607, 2018.

[18] Pengyi Yang, Wei Liu, Bing B. Zhou, Sanjay Chawla, and Albert Y. Zomaya. Ensemble-based wrapper methods for feature selection and class imbalance learning. In Jian Pei, Vincent S. Tseng, Longbing Cao, Hiroshi Motoda, and Guandong Xu, editors, *Advances in Knowledge Discovery and Data Mining*, pages 544–555, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[19] J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

[20] Nazrul Hoque, Mihir Singh, and Dhruba K Bhattacharyya. Efs-mi: an ensemble feature selection method for classification: An ensemble feature selection method. *Complex & Intelligent Systems*, 4:105–118, 2018.

[21] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos. An ensemble of filters and classifiers for microarray data classification. *Pattern Recognition*, 45(1):531–539, 2012.

[22] Fiona Katharina Ewald, Ludwig Bothmann, Marvin N Wright, Bernd Bischl, Giuseppe Casalicchio, and Gunnar König. A guide to feature importance methods for scientific inference. In *World Conference on Explainable Artificial Intelligence*, pages 440–464. Springer, 2024.

[23] S. M. Lundberg and S. I. Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.

[24] L. Martino, E. Morgado, and R. San Millán-Castillo. An index of effective number of variables for uncertainty and reliability analysis in model selection problems. *Signal Processing*, 227:109735, 2025.

[25] Isabella Verdinelli and Larry Wasserman. Decorrelated variable importance. *Journal of Machine Learning Research*, 25(7):1–27, 2024.

[26] Christina Heinze, Brian McWilliams, Nicolai Meinshausen, and Gabriel Krummenacher. Loco: Distributing ridge regression with random projections. *arXiv preprint arXiv:1406.3469*, 2014.

[27] M. G. Kendall. *Rank correlation methods*. Griffin, 4th edition, 1970.