Robust Selection Stability Estimation in Correlated Spaces

Victor Hamer[1]*✉ and Pierre Dupont[2]

UCLouvain - ICTEAM/INGI/Machine Learning Group, Place Sainte-Barbe 2,
B-1348 Louvain-la-Neuve, Belgium.
[1]victor.hamer@uclouvain.be [2]pierre.dupont@uclouvain.be

Abstract. The stability of feature selection refers to the variability of the selected feature sets induced by small changes of data sampling or analysis pipeline. Instability may strongly limit a sound interpretation of the selected features by domain experts. This work addresses the problem of assessing stability in the presence of correlated features. Correctly measuring selection stability in this context amounts to estimate to which extent several correlated variables contribute to predictive models, and how such contributions may change with the data sampling. We propose here a novel stability index taking into account such multivariate contributions. The shared contributions of several variables to predictive models do not only depend on the possible correlations between them. Computing this stability index therefore requires to solve a weighted bipartite matching problem to discover which variables actually share such contributions. We demonstrate that our novel approach provides more robust stability estimates than current measures, including existing ones taking into account feature correlations. The benefits of the proposed approach are demonstrated on simulated and real data, including microarray and mass spectrometry datasets. The code and datasets used in this paper are publicly available: https://github.com/hamerv/ecml21.

1 Introduction

Feature selection, *i.e.* the selection of a small subset of relevant features to be included in a predictive model, has already been studied in depth [3,8,10]. It has become compulsory for a wide variety of applications due to the appearance of very high dimensional data sets, notably in the biomedical domain [8].

Assessing feature selection has two distinct objectives: 1) a measure of the predictive performance of the models built on the selected features and 2) a measure of the stability of the selected features. Possible additional quality criteria are minimal model size or sparsity. Instability arises when the selected features drastically change after marginal modifications of the data sampling or processing pipeline. It prevents a correct and sound interpretation of the selected features and of the models built from them. One could even prefer a more stable modeling even if slightly less accurate [3,7].

* Victor Hamer is a Research Fellow of the Fonds de la Recherche Scientifique - FNRS

A typical protocol to assess selection stability amounts to run a feature selection algorithm over marginally modified training sets (*e.g.* through sub-sampling or bootstrapping) and to compare the selected feature sets across runs. Adequately evaluating the observed differences between these feature sets is the question we address here. A common measure is the Kuncheva index [4] which computes the proportion of common features across a pair of runs and reports the average over all pairs of runs, after correcting these proportions for chance. Such a measure and additional variants (briefly revisited in section 2) only focus on the identities of selected features in each run but plainly ignore the possible correlations between such features. This could lead to a pessimistic estimation of the stability when some features are selected in one run and other features, distinct from but highly correlated to the initial ones, are selected in another run. Previous works specifically address this issue by proposing stability indices taking into account feature correlations (or, more generally, feature similarity values) [9,12].

In this work, we argue that correctly assessing feature selection stability should go beyond considering correlations between features. Indeed, the selected features are the input variables of predictive models. Measuring stability should also assess to which extent the selected features jointly contribute to a multivariate model, and how such contributions vary across selection runs. In the simplest case, the importance of a specific feature in a (generalized) linear model is directly proportional to its absolute weight value in such a model. Section 3 extends to non-linear models this notion of feature importance. Section 4 describes typical situations in which current stability measures are questionable. This analysis motivates our *maximum shared importance stability measure*, formally defined in section 5. Computing this stability index requires to solve a weighted bipartite matching problem to discover which variables actually share such importance values across selection runs. Practical experiments on simulated and real data, including microarray and mass spectrometry datasets, are reported in section 6 to illustrate the benefits of the proposed approach.

2 Related Work

Let $\mathcal{F}_{1 \leq i \leq M}$ denote the subsets of features, among the d original features, produced by a feature selection algorithm run on M (*e.g.* bootstrap) samples of a training set. Let k_i denote $|\mathcal{F}_i|$, the number of selected features in run i. The Kuncheva index [4] quantifies the stability across the M selected subsets of features, whenever the number of selected features is constant across runs. Nogueira [6] generalizes the Kuncheva index to handle a varying number k_i of selected features:

$$\phi = 1 - \frac{\frac{1}{d}\sum_{f=1}^{d} s_f^2}{\frac{\bar{k}}{d} * (1 - \frac{\bar{k}}{d})} \tag{1}$$

with $\bar{k}$ the mean number of selected features over all runs, and $s_f^2 = \frac{M}{M-1}\hat{p}_f(1 - \hat{p}_f)$ the estimator of the variance of the selection of feature f over the M selected subsets, where $\hat{p}_f$ is the fraction of times f has been selected among them.

The Kuncheva index (KI) and the ϕ measure plainly ignore possible correlations between features. Inspired from Sechidis [9], we consider a scenario where a selection algorithm toggles, between pairs of correlated variables, across runs. More specifically, let us represent the selection matrix $\mathcal{Z}$ below with one row per selection run, $z_{i,f}$ indicating the selection of feature f in run i

$$\mathcal{Z} = \begin{pmatrix} 1\,0\,1\,0\,0\,0\,...\,0 \\ 0\,1\,1\,0\,0\,0\,...\,0 \\ 1\,0\,0\,1\,0\,0\,...\,0 \\ 0\,1\,0\,1\,0\,0\,...\,0 \end{pmatrix} \quad \begin{array}{l} \mathbf{z}_{1,\text{-}}, \text{ selected features in run 1} \\ \mathbf{z}_{2,\text{-}}, \text{ selected features in run 2} \\ \cdots \end{array} \tag{2}$$

In this limit case scenario, the selection algorithm toggles between features (1,2) and between features (3,4), while no other feature is ever selected. Let us assume that each pair of these features is strongly correlated. Since the selection algorithm always picks one variable in each correlated group, the information captured is essentially the same in each run and the selection should be considered perfectly stable. Yet, the Kuncheva Index and ϕ would tend to $\frac{1}{3}$ as d tends to infinity in such a scenario.

Sechidis [9] generalizes ϕ in order to accurately measure the selection stability in the presence of highly correlated variables:

$$\phi_{\mathcal{S}} = 1 - \frac{\text{tr}(\mathcal{S}K_{\mathcal{Z}\mathcal{Z}})}{\text{tr}(\mathcal{S}\Sigma^0)} \tag{3}$$

where the elements $s_{f,f'} \geq 0$ of the symmetric matrix $\mathcal{S}$ represent the correlations or, more generally, a similarity measure between features f and f', the matrix $K_{\mathcal{Z}\mathcal{Z}}$ is the variance-covariance matrix of $\mathcal{Z}$ and Σ^0 a normalization matrix. In the limit case scenario presented above and assuming perfect correlation between features 1 and 2, and, 3 and 4, respectively, $\phi_{\mathcal{S}} = 1$. This stability index $\phi_{\mathcal{S}}$ thus considers the feature correlations and only reduces to ϕ whenever $\mathcal{S}$ is the identity matrix. Yet, we show in section 4 and 6.1 that this measure is not lower bounded and can tend to $-\infty$ in seemingly stable situations.

Similarly to our proposal (detailed in section 5), Yu et al. [11] define a stability measure as the objective value of a maximum weighted bipartite matching problem. The two sets of vertices represent the selected features of two selection runs while the edge weights are the correlations between these features. The authors also propose a variant of their measure where each vertex can represent a correlated feature group as a whole. This measure is however not *fully defined*, as it requires the number of vertices (individually selected features or feature groups) to be constant across selection runs. This restriction is hardly met by all selection algorithms that could be considered in practice. In contrast, we show in section 5 that our novel measure is fully defined and actually generalizes the measure proposed in [11].

The POGR index is another existing stability measure defined to handle feature correlations [12]:

$$\text{POGR} = \frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{j=1, j\neq i}^{M} \frac{|\mathcal{F}_i \cap \mathcal{F}_j| + O_{i,j}}{k_i}, \tag{4}$$

with $|\mathcal{F}_i \cap \mathcal{F}_j|$ the number of features selected in both runs i and j, and $O_{i,j}$ the number of selected features in run i which are not selected in run j but are significantly correlated to at least one feature selected in run j. In our limit case scenario (2), POGR=1 as the algorithm always selects a feature from each of the two correlated pairs. Yet, we also illustrate its limitations in section 4 and 6.1.

We also consider a popular stability measure which estimates the stability of a *feature weighting*. It computes the average pairwise correlation between feature weights of different selection runs:

$$\phi_{\text{pears}} = \frac{2}{M(M-1)} \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} \frac{\sum_{f=1}^{d}(w_{i,f} - \mu_i)(w_{j,f} - \mu_j)}{\sqrt{\sum_{f=1}^{d}(w_{i,f} - \mu_i)^2} * \sqrt{\sum_{f=1}^{d}(w_{j,f} - \mu_j)^2}}, \quad (5)$$

with $w_{i,f}$ the weight, or score, associated to feature f in selection run i and μ_i the average feature weight in this run. Whenever these weights are either 0 or 1 (indicating the selection status of the feature) and if the number of non-zero weights is constant across selection runs, ϕ_{pears} becomes equivalent to KI and ϕ [5]. In this work, we use as feature weights the feature importance values, formally defined in the next section.

3 Feature importance

For each selection run i, the selected features $\mathcal{F}_i$ are assumed to be the input variables used to estimate a predictive model. We refer to $I_{i,f}$ as the importance value of the selected feature f in the predictive model $\mathcal{P}_i$ of run i, with $I_{i,f} = 0$ if f is not selected in run i. The binary matrix $\mathcal{Z}$ (introduced in (2)) is now replaced by a real matrix I made of these positive importance values.

We define the importance of a feature f in the predictive model $\mathcal{P}_i$, assumed here to be a classifier, as the inverse of the smallest noise applied to f necessary to flip the decision of model $\mathcal{P}_i$, averaged over the n learning examples. Formally,

$$I_{i,f} \triangleq \frac{1}{n} \sum_{l=1}^{n} \frac{k_i \times I_{i,f,\mathbf{x_l}}}{\sum_{f' \in \mathcal{F}_i}^{d} I_{i,f',\mathbf{x_l}}}, \quad I_{i,f,\mathbf{x_l}} = \frac{\sigma_f}{\delta_{\mathbf{x_l},i,f}} \quad (6)$$

where $\delta_{\mathbf{x_l},i,f}$ is the smallest additive change (in absolute value) required to feature f such that the decision of the predictive model $\mathcal{P}_i$ on example $\mathbf{x_l}$ changes, and σ_f the standard deviation of feature f. Intuitively, if one can change feature f by large amounts without perturbing the decisions of the classifier, then f is not important in its decisions. To the contrary, if a small change to f causes a lot of decision switches, then the model is highly sensitive to it. Such a definition is highly reminiscent of the permutation test introduced by Breiman to quantify the importance of a feature in a Random Forest. Yet, our formulation has been preferred due to its high interpretability for linear models.

Indeed, for a linear model with weights $\mathbf{w}_i$, built on features normalized to unit variance, this formulation can be shown to be equivalent to

$$I_{i,f} = ||\mathbf{w}_i||_0 \times \frac{|w_{i,f}|}{||\mathbf{w}_i||_1}. \quad (7)$$

The importance of a feature in a linear model is proportional to the absolute value of its weight in such a model. We further normalize feature importance such that each row of the importance matrix I sums up to $\bar{k}$, the average number of selected features.

4 Limits of existing stability measures

We describe here 2 scenarios of feature selection in correlated feature spaces. We show that the existing stability measures exhibit undesirable behaviors in these scenarios, which will further motivates our novel stability measure introduced in section 5.

Figures 1 and 2 illustrate the similarity between two selection runs. These figures represent feature importance in a more intuitive way than the importance matrix I. Each feature (or each group of highly correlated features) is identified by a unique color and the width of the rectangles correspond to the relative importance value of a feature (group) in the predictive model for this run. We assume for simplicity that the total number d of features tends to infinity and that the only non-negligible feature correlations are the ones illustrated by explicit links in the figures. This implies that the denominator of $\phi_{\mathcal{S}}$ in equation (3) simply becomes $tr(\mathcal{S}\Sigma^0) = \bar{k}$, the average number of selected features.

In both scenarios, we consider a group of q perfectly correlated features and we study what happens when q, starting from 1, gradually increases. The importance values of the features inside a correlated group can either be concentrated on a single feature in the group or, to the contrary, be divided, possibly unevenly, among several or all correlated features. No matter the case, we assume that the cumulative importance of all features within a correlated group roughly stay constant across selection runs. This assumption is actually confirmed in our later practical experiments reported in Figure 4. Our limit case scenarios are designed in such a way that the stability value should be equal to $\frac{1}{2}$ and stay constant no matter the value of q. We show that no existing stability index satisfies this property.

In the first scenario (Figure 1), a group of q perfectly correlated features (orange) is selected with a *cumulative* importance of $\frac{\bar{k}}{4}$ in both selection runs, together with another feature (green). The two additional selected features differ in each run. Arguably, whether a large group of correlated features or a single feature from this group is selected should not impact stability, as long as the global contribution of the group to the predictive models is unchanged. In such a scenario we argue that the stability should be equal to $\frac{1}{2}$, independently of q, as half of the selected information is in common between both runs. In this scenario, $\phi_{\mathcal{S}}$, ϕ, ϕ_{pears} and POGR correctly start at $\frac{1}{2}$ when $q = 1$ but $\phi_{\mathcal{S}}$, ϕ (both are equivalent in this example) and POGR increase and tend to 1 when q tends to ∞ while ϕ_{pears} decreases with q. The precise dependencies of these measures on q are summarized in Table 1.

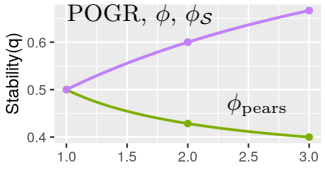
Fig. 1. Scenario 1. A group of q perfectly correlated features (orange) is selected in both selection runs. Their specific importance in predictive models may vary but their *cumulative* importance is assumed constant across runs and equal to $\frac{k}{4}$. The stability should be equal to $\frac{1}{2}$ in such a scenario, independently of the q value, because the information captured by predictive models is essentially the same in both runs.

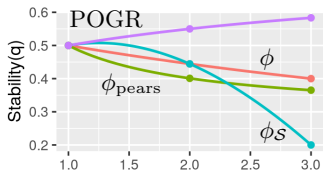
Fig. 2. Scenario 2. In the second selection run (below), a group of q perfectly correlated features is selected with a cumulative importance of $\frac{k}{4}$. In the first run (top), a single feature from this group is selected and captures the whole group importance in the predictive model. The stability should be equal to $\frac{1}{2}$, independently of the q value.

Table 1. Dependencies of $\phi_{\mathcal{S}}$, ϕ, ϕ_{pears} and POGR on the size q of the correlated group in the scenarios represented in Figures 1 (top) and 2 (bottom). In order to get a closed formula for ϕ_{pears}, importance values are assumed to be evenly distributed within a correlated group.

Measure	Value	$q = 1$	$\lim_{q\to\infty}$
$\phi_{\mathcal{S}}$	$\frac{q+1}{q+3}$	$\frac{1}{2}$	1
ϕ	$\frac{q+1}{q+3}$	$\frac{1}{2}$	1
ϕ_{pears}	$\frac{q+1}{3q+1}$	$\frac{1}{2}$	$\frac{1}{3}$
POGR	$\frac{q+1}{q+3}$	$\frac{1}{2}$	1

Measure	Value	$q = 1$	$\lim_{q\to\infty}$
$\phi_{\mathcal{S}}$	$1 - \frac{q^2-2q+5}{q+7}$	$\frac{1}{2}$	$-\infty$
ϕ	$1 - \frac{q+3}{q+7}$	$\frac{1}{2}$	0
ϕ_{pears}	$\frac{\frac{1}{4q}+\frac{1}{4}}{\sqrt{\frac{1}{4q}+\frac{3}{4}}}$	$\frac{1}{2}$	$\frac{\sqrt{3}}{6}$
POGR	$\frac{1}{2}(\frac{1}{2} + \frac{q+1}{q+3})$	$\frac{1}{2}$	$\frac{3}{4}$





The second scenario (Figure 2) is nearly identical to the first one, except that a single feature of the correlated group is selected in the first selection run. This feature captures the whole group importance. Again, stability should be equal to $\frac{1}{2}$, independently of q. In this second scenario, $\phi_{\mathcal{S}}$, ϕ, ϕ_{pears} and POGR correctly start at $\frac{1}{2}$ when $q = 1$ but $\phi_{\mathcal{S}}$, ϕ and ϕ_{pears} decrease and respectively tend to $-\infty$, 0 and $\frac{\sqrt{3}}{6}$ when $q \to \infty$, while POGR increases and tends to $\frac{3}{4}$.

5 Stability as Maximal Shared Importance

In this section, we introduce a novel stability measure and show that it behaves adequately in the two scenarios discussed in section 4. We further demonstrate that it generalizes previous work and prove several of its properties.

Unlike the limit cases presented in section 4, actual correlations between features need not be restricted to 0 or 1 values. Moreover, observing some correlation between *e.g.* a pair of features does not guarantee that they will necessarily share their importance values in models taking these features as input variables. Hence, to correctly assess stability while considering importance values, one needs to discover which features actually share importance values across predictive models built from different selection runs. This is why the evaluation of our stability index requires to solve a linear program.

Let $\mathcal{S}$ be a symmetric similarity matrix with $s_{f,g}$ the *similarity value* between feature f and feature g. We assume that such a similarity value falls in the $[0, 1]$ interval. It is supposed to be *a priori* defined or estimated over the whole training set for any pair of features. Typical choices include the absolute values of the Pearson's or Spearman's correlations, or mutual information normalized in the $[0, 1]$ interval over the whole training set.

For each pair (i, j) of feature selection runs, one looks for the matching between features that maximizes a (similarity weighted) shared importance. Formally, $S(i, j)$ is the optimal objective value of the following constrained optimization problem:

$$S(i,j) = \max_{\mathbf{x}} \frac{\sum_{f,g} s_{f,g} \times x_{f,g}}{\bar{k}} \tag{8}$$

$$\text{subject to} \quad \sum_{g \in \mathcal{F}_j} x_{f,g} \leq I_{i,f}, \quad \forall f \in \mathcal{F}_i \tag{9}$$

$$\sum_{f \in \mathcal{F}_i} x_{f,g} \leq I_{j,g}, \quad \forall g \in \mathcal{F}_j \tag{10}$$

$$x_{f,g} \geq 0, \quad \forall (f,g) \in (\mathcal{F}_i, \mathcal{F}_j) \tag{11}$$

The variables $x_{f,g}$ represent the *latent* amount of shared importance by feature f, selected in run i, and feature g, selected in run j. This shared importance is multiplied by the similarity $s_{f,g}$ between the two features. The feature f can share its importance with several features of run j, but its total shared importance cannot exceed its own importance $I_{i,f}$ according to the constraint (9)

(and its reciprocal (10)). Feature importance values are normalized such that $\sum_{f \in \mathcal{F}_i} I_{i,f} = \bar{k}$, the average number of selected features.[1] Since the objective (8) and all constraints are linear with respect to the variables $x_{f,g}$, this optimization problem can be efficiently solved by linear programming. The stability over M feature selection runs is defined as the average pairwise optimal $S(i,j)$ values:

$$\phi_{\text{msi}} = \frac{2}{M(M-1)} \sum_{i=1}^{M} \sum_{j=i+1}^{M} S(i,j) \tag{12}$$

An example of optimal solution is depicted in Figure 3. Features 1, 2, 3 and 4 are selected in run i while features 2, 5, 6 and 7 are selected in run j. Feature 1 is heavily correlated to feature 6 ($s_{1,6} = 0.8$) and relatively well correlated to feature 5 ($s_{1,5} = 0.6$). Feature 6 is also somewhat correlated to feature 3 ($s_{3,6} = 0.4$). To maximize the objective, the link $x_{1,5}$ between features 1 and 5 is set to the maximum possible value, 0.7. The remaining importance of feature 1 is shared with feature 6 ($x^*_{1,6} = 0.6$) and the link between feature 3 and 6 is set to the maximum remaining importance of feature 6 ($x^*_{3,6} = 0.8$).

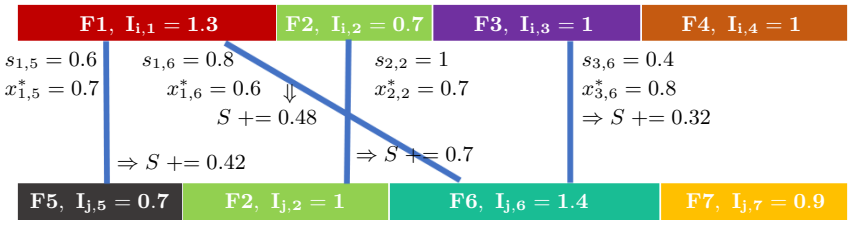


Fig. 3. Example of optimal solution with $S(i,j) = \frac{0.42+0.48+0.7+0.32}{4} = 0.48$. One can verify that $x^*_{1,5} + x^*_{1,6} \leq I_{i,1} = 1.3$ and $x^*_{1,6} + x^*_{3,6} \leq I_{j,6} = 1.4$.

The stability ϕ_{msi} behaves correctly in the two scenarios presented in section 4. Considering the first scenario (Figure 1), the constant cumulative importance of the correlated group, independently of its size q, guarantees that the stability is also constant with q. Indeed, the optimal solution verifies $\sum_{q',q'' \in [1,q]} x_{q',q''} = \frac{\bar{k}}{4}$ which implies

$$S(i,j) = \frac{\sum_{q',q'' \in [1,q]} s_{q',q''} x_{q',q''} + \frac{\bar{k}}{4}}{\bar{k}} = \frac{\frac{\bar{k}}{4} + \frac{\bar{k}}{4}}{\bar{k}} = \frac{1}{2}, \forall q \geq 1,$$

as all the similarities $s_{q,q'}$ are equal to 1. In scenario 2 (Figure 2), the importance of feature 1 in the first run is shared among the q correlated features in the second

[1] Since this normalization is undefined for a run i with $k_i = 0$ (a limit case with no feature selected in this run), we pose $S(i,j) = 0$ if $k_i = 0 \oplus k_j = 0$, and $S(i,j) = 1$ if $k_i = k_j = 0$, with $\oplus$ the XOR operator.

run: $x_{1,q'} = I_{2,q'}$, for all $1 \leq q' \leq q$, the optimal objective value is then

$$S(i,j) = \frac{\sum_{1 \leq q' \leq q} s_{1,q'} x_{1,q'} + \frac{\bar{k}}{4}}{\bar{k}} = \frac{\sum_{1 \leq q' \leq q} I_{2,q'} + \frac{\bar{k}}{4}}{\bar{k}} = \frac{\frac{\bar{k}}{4} + \frac{\bar{k}}{4}}{\bar{k}} = \frac{1}{2}, \forall q \geq 1.$$

Section 6.1 further illustrates on simulated data that our measure ϕ_{msi} does not suffer from the limitations of current measures.

We show below that ϕ_{msi} is bounded, which is necessary for a sound interpretation of the stability value, and fully defined (Property 5.1 and 5.2). We also demonstrate its maximality conditions in Property 5.3.

Property 5.1. The stability measure ϕ_{msi} is bounded in $[0, 1]$.

Proof. As every variable $x_{f,g}$ and every entry of the similarity matrix $\mathcal{S}$ are positive, the objective (8) is positive as well. The measure is thus lower-bounded by 0.

If one assumes maximally similar features ($s_{f,g} = 1, \forall f, g$) instead of their actual similarity values, the resulting optimization problem has an optimal objective value at least equal to the optimal objective value of the initial problem. The set of feasible solutions is the same as the constraints do not depend on $s_{f,g}$, and every solution has a larger or equal objective value than the corresponding solution of the initial problem. The optimal solution becomes

$$S(i,j) = \frac{\sum_{f \in \mathcal{F}_i} \sum_{g \in \mathcal{F}_j} x_{f,g}}{\bar{k}} \leq \frac{\sum_{f \in \mathcal{F}_i} I_{i,f}}{\bar{k}} = \frac{\bar{k}}{\bar{k}} = 1, \tag{13}$$

using constraint (9). The stability ϕ_{msi} is thus upper-bounded by 1.

Property 5.2. The stability measure ϕ_{msi} is fully defined.

Proof. We have posed $S(i,j) = 0$ if $k_i = 0 \oplus k_j = 0$ and 1 if $k_i = k_j = 0$, with $\oplus$ the XOR operator. It remains to show that the optimization problem always admits a feasible solution when k_i and k_j are both non-zero. Since $I_{i,f} \geq 0, \forall f \in \mathcal{F}_i, \forall i$, the trivial assignation $x_{f,g} = 0, \forall f, g$ is a solution.

Property 5.3. The stability measure ϕ_{msi} is maximal ($= 1$) iff, for all pairs of runs i and j, each feature importance in run i can be fully shared with the importance of one or several perfectly correlated features in run j. In other words, there exists no link $x_{f,g} > 0$ with $s_{f,g} < 1$, and all constraints from the set (9) are active: $\sum_{g \in \mathcal{F}_j} x_{f,g} = I_{i,f}, \forall f \in \mathcal{F}_i$.

Proof. Suppose there exists a variable $x_{f,g} > 0$ with $s_{f,g} < 1$. Then increasing $s_{f,g}$ strictly increases the objective value. The previous objective value was thus not maximal. Suppose that $s_{f,g} = 1$ for every variable $x_{f,g} > 0$. Then, similarly to equation (13),

$$S(i,j) = \frac{\sum_{f \in \mathcal{F}_i} \sum_{g \in \mathcal{F}_j} x_{f,g}}{\bar{k}} \tag{14}$$

which is maximal ($= 1$) iff $\sum_{g \in \mathcal{F}_j} x_{f,g} = I_{i,f}, \forall f \in \mathcal{F}_i$. By reciprocity, the set of constraints (9) is active iff the set of constraints (10) is active.

Theorem 5.1 shows that solving the optimization problem (8) is equivalent to solving a maximum weighted bi-partite matching problem whenever the importance of all selected features is evenly distributed between them in any given run and the number of selected features is constant across the M runs. In this specific case, ϕ_{msi} reduces to the measure proposed by Yu et al. [11].

Theorem 5.1. *Whenever the importance of all selected features is evenly distributed between them in any given run and the number of selected features is constant across the M runs, the constrained optimization problem (8) is a maximum weighted bi-partite matching problem.*

Proof. The assumptions imply $I_{i,f} = 1, \forall f \in \mathcal{F}_i, \forall i$. Problem (8) has the form of $\{\max cx | Ax \leq b,\ x \geq 0\}$, with A the matrix of constraints (9) and (10). We first show that A is totally submodular, *i.e.* every of its square submatrix has determinant $0, +1$ or -1. The following four conditions are sufficient for a matrix to be totally submodular [1]:

1. Every entry in A is 0, +1, or -1;
2. Every column of A contains at most two non-zero (i.e., +1 or -1) entries;
3. If two non-zero entries in a column of A have the same sign, then the row of one is in B and the other in C, with B and C two disjoint sets of rows of A;
4. If two non-zero entries in a column of A have opposite signs, then the rows of both are in B or both in C, with B and C two disjoint sets of rows of A.

Condition 1 is satisfied as the coefficients multiplying $x_{f,g}$ in the set of constraints (9) and (10) are 0 or 1. Condition 2 holds because each variable $x_{f,g}$ has a non-zero coefficient in two constraints, one from the set of constraints (9) and one from (10). Condition 3 holds as we let B be the rows of A representing the set of constraints (9) and C be the rows of A corresponding to the set of constraints (10). If two non-zero entries in a column of A have the same sign, then one row represents a constraint in set (9) (and is thus in B) while the other represents a constraint in set (10) (and is thus in C). Condition 4 is trivially satisfied as two non-zero entries of A never have opposite signs.

If A is totally submodular and b is integral (which is here the case as $I_{i,f} = 1, \forall f \in \mathcal{F}_i, \forall i$), then linear programs of the forms $\{\max cx | Ax \leq b,\ x \geq 0\}$ have integral optimal solutions. In our case, the variables $x_{f,g}$ can only belong to $\{0, 1\}$ which makes the original optimization problem equivalent to maximum weighted bipartite matching.

Computing $S(i, j)$ requires to solve a linear programming problem with $k_i k_j$ variables, which can currently be done in $O((k_i k_j)^{2.055})$ time [2]. The overall time complexity to compute ϕ_{msi} is then $O(M^2 \overline{(k_i k_j)^{2.055}}) \approx O(M^2 \overline{k}^{\ 4.11})$. Even though this is a somewhat high computational cost, it does not depend on the typically very high number d of features, unlike $\phi_{\mathcal{S}}$ which requires $O(d^2)$ time.

6 Experiments

In this section, we illustrate on simulated data that our proposed measure ϕ_{msi} improves the behavior of current measures in the presence of highly correlated

feature groups (section 6.1). We further show on microarray and mass spectrometry data that the (accuracy, stability) Pareto fronts change when the stability measure includes feature correlation and feature importance values (section 6.2).

Classification accuracy and stability of the feature selection are estimated through bootstrapping. Feature selection is applied on each bootstrap sample and the stability is computed across M runs. Classification accuracy is evaluated on the out-of-bag examples of each run and its average value is reported.

6.1 Simulated Data

We use an artificially generated data set with $N = 5$ groups of variables. Each group contains q features that are highly correlated to each other (average correlation of $\rho^g \geq 0.8$). In addition to these correlated groups, the data set contains $l = 1000$ variables. Feature values are sampled from two multivariate normal distributions using the `mvrnorm` R package. Positive examples ($n_+ = 100$) are sampled from a first distribution, centered on $\boldsymbol{\mu_+}$, a vector with $\mu_{+,f} = \mu_+^g$ if feature f belongs to one of the $N = 5$ correlated groups, $\mu_+^{\neg g}$ otherwise. Negative examples ($n_- = 100$) are sampled from a second distribution, centered on $\boldsymbol{\mu_-} = -\boldsymbol{\mu_+}$. Both distributions have unit variance. We consider three scenarios with different values of μ_+^g, $\mu_+^{\neg g}$ and ρ^g, specified in Table 2. In all scenarios, features inside a correlated group are very relevant to the binary prediction task, while features outside such groups are less but still marginally relevant. The group LASSO is used as feature selection method (scenarios 1 and 2), regularized such as to select all features inside a group or none of them. The standard LASSO, which tends to select only a few features inside each correlated group, is also evaluated (scenario 3). The regularization parameter λ of the LASSO and group LASSO is chosen so as to select approximately 40 features when $q = 1$ (each correlated group is reduced to a single feature). For larger q values, the $N = 5$ correlated groups are expected to be selected in most of the $M = 30$ selection runs while the selection of the additional features is likely to be unstable. This experiment is repeated 10 times using different generative seeds for the data sets and the mean stability values are reported on Figure 5 as a function of q, the size of the correlated groups.

Table 2. Experimental settings for the 3 scenarios. The relevance of features inside one of the $N = 5$ correlated groups is related to μ_+^g while the relevance of features outside any group ($\sim \mu_+^{\neg g}$) is constant across scenarios. The average intra-group correlation is ρ^g and inter-group correlation is negligible.

Scenario	μ_+^g	$\mu_+^{\neg g}$	ρ^g	method
1	0.35	0.05	0.8	group LASSO
2	0.5	0.05	0.8	group LASSO
3	0.5	0.05	0.95	LASSO

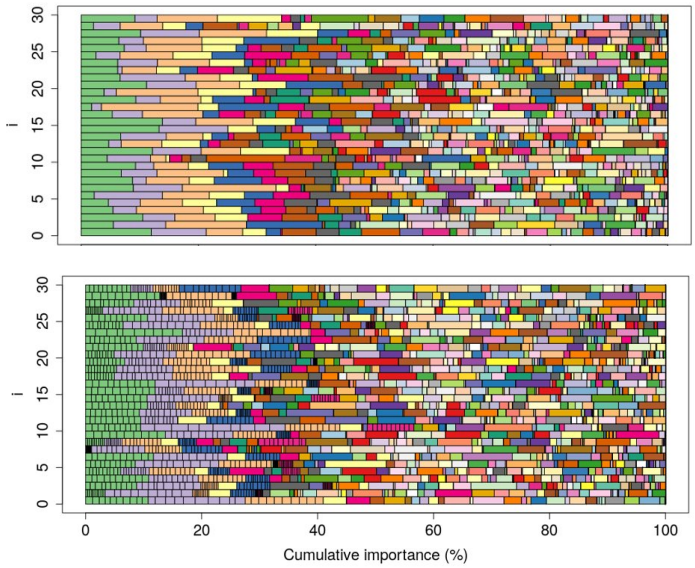
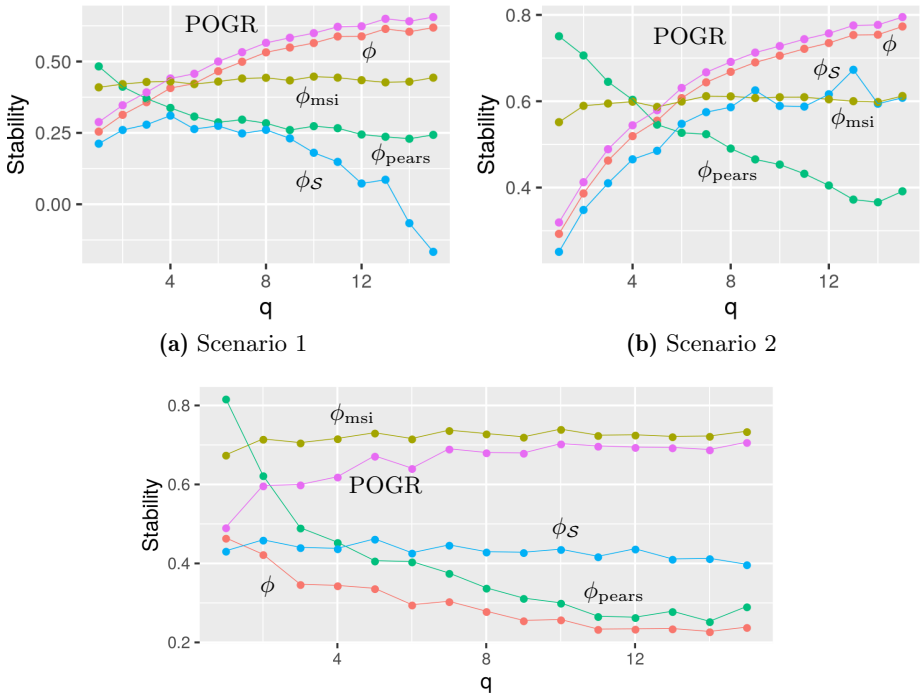


Fig. 4. Feature stability maps of the group LASSO (scenario 1) when the size of the correlated groups q is equal to 1 (top) and 10 (bottom). As the cumulative importance of each group is approximately constant in both feature stability maps, their stability should be similar.

Figure 4 represents the cumulative importance of the features that are selected by the group LASSO in scenario 1 when $q = 1$ (top) and $q = 10$ (bottom). We use here a similar representation as in Figures 1, 2 and 3, extended to $M = 30$ runs. We refer to such a representation as a *feature stability map*. Figure 4 illustrates that the group LASSO gives more importance to the features of the 5 "groups" when $q = 1$ as they are more relevant (by design) to the classification. When $q = 10$, the *cumulative* importance of each correlated group is approximately the same as in the $q = 1$ case, with the individual feature importance proportionally reduced. This result supports the assumptions made when defining the limit case represented in Figure 2.

In this controlled experiment, the similarity matrix $\mathcal{S}$ is estimated as the absolute values of the pairwise Spearman's ρ correlation: $s_{f,f'} = |\rho_{f,f'}|$. Such similarities are used when computing ϕ_{msi} and $\phi_{\mathcal{S}}$. Figure 5 compares ϕ, ϕ_{pears} (with the importance values $I_{i,f}$ as feature weights), ϕ_{msi}, $\phi_{\mathcal{S}}$ and POGR when the size q of the correlated groups increases. When the group LASSO is used (Figures 5a and 5b), ϕ and POGR increases with q while ϕ_{pears} decreases with q.

The evolution of $\phi_{\mathcal{S}}$ depends on the scenario considered. The experiments reported in Figure 5b are such that the correlated groups are sufficiently relevant for the classification so as to be selected in nearly all selection runs. In such a situation, $\phi_{\mathcal{S}}$ tends to increase with q. Whenever some correlated groups are



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

Fig. 5. Stability values of ϕ, $\phi_{\mathcal{S}}$, ϕ_{msi}, ϕ_{pears} and POGR in the presence of highly correlated feature groups, in function of q, the size of such groups. The group LASSO is used for feature selection in (a)(scenario 1) and (b)(scenario 2), the LASSO in (c)(scenario 3). Given the design of these experiments, the stability value should not depend on q, which is only the case for ϕ_{msi} in the 3 scenarios.

regularly not selected, as in scenario 1 reported in Figure 5a, $\phi_{\mathcal{S}}$ actually tends to decrease with q and decays to $-\infty$.

Figure 5c is obtained with the pure LASSO selection which selects a few features inside correlated groups instead of the whole groups. In such a scenario, ϕ and ϕ_{pears} decreases with q as the selection of a few features inside each group becomes more and more unstable and because these measures do not take feature correlations into account. POGR somewhat increases with q. This is the only scenario for which $\phi_{\mathcal{S}}$ is approximately constant, with a lower value than ϕ_{msi}. The novel stability measure ϕ_{msi} is the only one to be approximately constant with q, in all 3 scenarios.

6.2 Stability of Standard Selection Methods

In this section, we report (accuracy, stability) Pareto fronts on real data sets, including microarray and mass spectrometry data. We compare here the standard

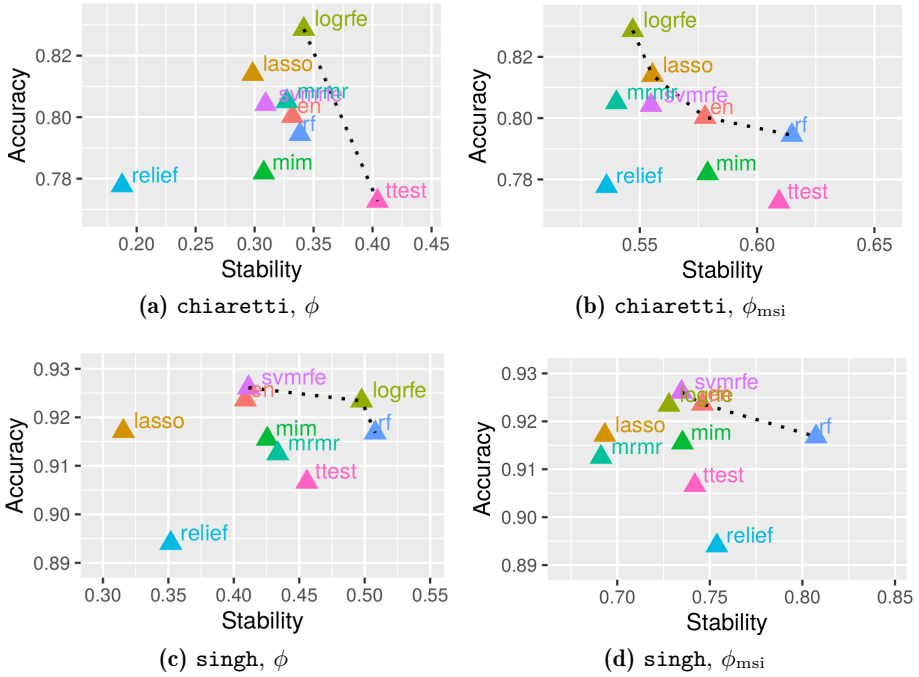
stability ϕ, which ignores feature correlations, and ϕ_{msi}. The similarity matrix $\mathcal{S}$ is estimated as the absolute values of the pairwise Spearman's ρ correlation: $s_{f,f'} = |\rho_{f,f'}|$. We consider the following feature selection methods and report performances over $M = 100$ selection runs.

- Random forests with 1000 trees. A first forest is learned on the original d features. The 20 features whose removal would cause the greatest accuracy decrease on the out-of-bag examples are selected. A second forest of 1000 trees built on those 20 features is used for prediction.
- Logistic regression with a LASSO and ELASTIC-NET penalty. For the ELASTIC-NET $\lambda_1(\lambda_2 \text{L1} + (1 - \lambda_2)\text{L2})$, the parameter λ_2, which dictates the balance between L1 and L2 norms, is set to 0.8. The parameter λ_1 is set to select, on average, $\bar{k} = 20$ features over the M runs.
- The RELIEF algorithm with 5 neighbors with equal weights to select 20 features. Predictive models are 5-NN classifiers.
- The logistic or hinge loss RFE. Predictive models are obtained by logistic regression or by fitting a linear SVM on the 20 selected features.
- The t-test, Mutual Information Maximization (MIM) (`infotheo` R package) and minimum Redundancy Maximum Relevance (mRMR) (`mRMRe` R package) information theoretic methods. For these three last approaches, the final model used for prediction is logistic regression estimated from the 20 selected features.

We report experiments on 5 microarray data sets (`alon`, `singh`, `chiaretti`, `gravier` and `borovecki`) from the `datamicroarray` R package and one mass-spectrometric data set, `arcene`, from the UCI machine learning repository. They have a high number of features (from 2,000 for `alon` to 22,283 for `borovecki`) with respect to the number of training examples (from 31 for `borovecki` to 198 for `arcene`), which generally leads to instability. In each selection run, the feature space is pre-filtered by keeping the 5,000 features with highest variance before running a specific selection algorithm.

Figure 6 represents the (accuracy, stability) Pareto front across all feature selection methods on two representative datasets (`chiaretti` (top) and `singh` (bottom)), for the two stability measures ϕ (left) and ϕ_{msi} (right). Results on all 6 datasets are summarized in Table 3. These results show that the choice of best performing feature selection methods highly depends on the stability measure used. Figure 6 and Table 3 further show that the stability ϕ_{msi} is much higher than ϕ for all selection methods. This phenomenon is the most pronounced on `borovecki` where $0.06 < \phi < 0.24$ and $0.7 < \phi_{\text{msi}} < 0.88$. This indicates that the observed instability is largely due to the high correlation between the input features. Even though the selection is unstable at the level of the input features, selected features from different selection runs tend to be highly correlated.

Correcting for the correlation between features, as done by $\phi_{\mathcal{S}}$, is not enough however. Stability results according to $\phi_{\mathcal{S}}$ are detailed in appendix, with even lower values than those of ϕ. The proposed measure ϕ_{msi} behaves better because it does not only consider feature correlations but also feature importance values, which are matched between predictive models from several selection runs.



(a) `chiaretti`, ϕ **(b)** `chiaretti`, ϕ_{msi}

(c) `singh`, ϕ **(d)** `singh`, ϕ_{msi}

Fig. 6. Pareto fronts across selection methods obtained when ϕ (a,c) or ϕ_{msi} (b,d) estimates stability, on two representative datasets: `chiaretti` and `singh`.

Table 3. Stability ranges and Pareto fronts for ϕ and ϕ_{msi} on all datasets.

Data	Range ϕ	Pareto front ϕ	Range ϕ_{msi}	Pareto front ϕ_{msi}	Range ac.
`chiar.`	0.19-0.41	`logrfe/ttest`	0.54-0.62	`logrfe/lasso/en/rf`	0.77-0.83
`singh`	0.32-0.51	`svmrfe/logrfe/rf`	0.69-0.81	`svmrfe/en/rf`	0.89-0.93
`alon`	0.21-0.47	`logrfe/svmrfe`	0.64-0.76	`logrfe/en/ relief/svmrfe`	0.77-0.82
`grav.`	0.11-0.27	`logrfe/ttest`	0.36-0.47	`logrfe/ttest/en`	0.68-0.75
`arcene`	0.11-0.3	`rf/relief/ logrfe/en`	0.41-0.7	`rf`	0.68-0.75
`borov.`	0.06-0.24	`ttest`	0.7-0.88	`ttest/rf`	0.88-0.97

7 Conclusion

Current feature selection methods, especially applied to highly dimensional data, tend to suffer from instability since marginal modifications in data sampling may

result in largely distinct selected feature sets. Such instability may strongly limits a sound interpretation of the selected variables.

In this work, we focus on estimating stability in feature spaces with strong feature correlations. We pose stability as the optimal objective value of a constrained optimization problem, which can be efficiently solved by linear programming. This objective depends on a similarity measure between features and on their relative importance values in predictive models. We demonstrate on handcrafted examples and on simulated data that our approach provides more relevant stability estimates than existing stability measures. Experimental results on microarray and mass spectrometry data also illustrate that a sound stability estimation may strongly affect the choice of selection method when picking an optimal trade-off between feature selection stability and predictive performance.

References

1. Heller, I., Tompkins, C.: An extension of a theorem of dantzig's. Linear inequalities and related systems **38**, 247–254 (1956)
2. Jiang, S., Song, Z., Weinstein, O., Zhang, H.: Faster dynamic matrix inverse for faster lps. in arxiv preprint (2020)
3. Kalousis, A., Prados, J., Hilario, M.: Stability of feature selection algorithms: a study on high-dimensional spaces. Knowledge and information systems **12**(1), 95–116 (2007)
4. Kuncheva, L.I.: A stability index for feature selection. In: Artificial intelligence and applications. pp. 421–427 (2007)
5. Nogueira, S., Brown, G.: Measuring the stability of feature selection. In: Joint European conference on machine learning and knowledge discovery in databases. pp. 442–457. Springer (2016)
6. Nogueira, S., Sechidis, K., Brown, G.: On the stability of feature selection algorithms. The Journal of Machine Learning Research **18**(1), 6345–6398 (2017)
7. Saeys, Y., Abeel, T., Van de Peer, Y.: Robust feature selection using ensemble feature selection techniques. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 313–325. Springer (2008)
8. Saeys, Y., Inza, I., Larrañaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics **23**(19), 2507–2517 (2007)
9. Sechidis, K., Papangelou, K., Nogueira, S., Weatherall, J., Brown, G.: On the stability of feature selection in the presence of feature correlations. In: Proceedings of the 2019 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (2019)
10. Tang, J., Alelyani, S., Liu, H.: Feature selection for classification: A review. Data classification: Algorithms and applications p. 37 (2014)
11. Yu, L., Ding, C., Loscalzo, S.: Stable feature selection via dense feature groups. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 803–811 (2008)
12. Zhang, M., Zhang, L., Zou, J., Yao, C., Xiao, H., Liu, Q., Wang, J., Wang, D., Wang, C., Guo, Z.: Evaluating reproducibility of differential expression discoveries in microarray studies by considering correlated molecular changes. Bioinformatics **25**(13), 1662–1668 (2009)