Machine Learning (2022) 111:3897–3923
https://doi.org/10.1007/s10994-022-06221-9

A user-guided Bayesian framework for ensemble feature selection in life science applications (UBayFS)

Anna Jenul[1] · Stefan Schrunner[1] · Jürgen Pilz[2] · Oliver Tomic[1]

Received: 15 December 2021 / Revised: 24 May 2022 / Accepted: 2 July 2022 /
Published online: 22 August 2022

Abstract

Feature selection reduces the complexity of high-dimensional datasets and helps to gain insights into systematic variation in the data. These aspects are essential in domains that rely on model interpretability, such as life sciences. We propose a (U)ser-Guided (Bay)esian Framework for (F)eature (S)election, UBayFS, an ensemble feature selection technique embedded in a Bayesian statistical framework. Our generic approach considers two sources of information: data and domain knowledge. From data, we build an ensemble of feature selectors, described by a multinomial likelihood model. Using domain knowledge, the user guides UBayFS by weighting features and penalizing feature blocks or combinations, implemented via a Dirichlet-type prior distribution. Hence, the framework combines three main aspects: ensemble feature selection, expert knowledge, and side constraints. Our experiments demonstrate that UBayFS (a) allows for a balanced trade-off between user knowledge and data observations and (b) achieves accurate and robust results.

Keywords Ensemble feature selection · Bayesian model · Dirichlet-multinomial · User constraints

Editors: Krzysztof Dembczynski and Emilie Devijver.

Anna Jenul and Stefan Schrunner have contributed equally to this work.

✉ Stefan Schrunner
stefan.schrunner@nmbu.no

Anna Jenul
anna.jenul@nmbu.no

Jürgen Pilz
juergen.pilz@aau.at

Oliver Tomic
oliver.tomic@nmbu.no

[1] Department of Data Science, Norwegian University of Life Sciences, Ås, Norway

[2] Department of Statistics, University of Klagenfurt, Klagenfurt, Austria

1 Introduction

Feature selection pursues two major goals: to improve the performance of predictive algorithms like classification, regression, or clustering models as well as to improve data understanding and interpretability. Both aspects are of significant interest in the field of life science, such as healthcare, where major decisions may be based on data analysis. Here, two sources of information are often available: large-scale collections of data from multiple sources and profound knowledge from domain experts. Previous works tend to handle these sources as opposites, see Cheng et al. (2006), or neglect expert knowledge completely, see Pozzoli (2020). However, a combination of both can be valuable to compensate for underdetermined problem setups from high-dimensional datasets, which are prevalent in healthcare data analysis. Moreover, meta-information on the feature set may leverage interpretability. Works such as Liu and Zhang (2015) consider constraints between samples but neglect constraints between features. The extension of L1 regularization to the so-called *Group Lasso* (Yuan & Lin , 2006) and its variants (Ida et al. , 2019) account for block structure but cannot handle more complex constraint types. Elementary approaches to integrating user knowledge and feature selection include Guan Guan et al. (2009), who suggest manually adding user-defined features to the feature selection output of algorithms. A more advanced model by Brahim and Limam (2014) embeds prior knowledge into three particular feature selection algorithms. Though, their work neither allows a direct generalization to other feature selectors nor the integration of more general types of prior knowledge, such as side constraints. Hence, there is a lack of general and sophisticated frameworks for feature selection that combine data-driven methods with user knowledge and deliver transparent results.

Apart from measuring predictive model performance, properties like stability and reproducibility of the feature selector are essential for transparency. A model-independent approach for improving feature selection stability is to deploy ensembles of elementary feature selectors. Recent research by Bose (2021), and Jenul (2021) pursued this idea by utilizing sub-sampling strategies to generate model ensembles as such provide feature stability measures aside from good predictive performance. Seijo-Pardo et al. (2017) conclude that meta-models composed of elementary feature selectors improve the performance and robustness of the selected feature set in many cases. However, to the best of our knowledge, probabilistic approaches that exploit both — a sound statistical framework and individual model benefits of using an ensemble elementary feature selectors — are not yet available.

A prominent framework with the capability to combine data and expert knowledge is Bayesian statistics, which has been applied for feature selection in linear models, see O'Hara and Sillanpää (2009). Intentions behind the usage of Bayesian methodology vary significantly between authors and do not necessarily involve expert knowledge. Examples include Dalton (2013), who investigates sparsity priors, and Goldstein et al. (2020), who suggest a Bayesian framework to quantify the level of uncertainty in the underlying feature selection model. Other Bayesian approaches for feature selection include Saon and Padmanabhan (2001), and Lyle et al. (2020), but these works do not investigate the usage of expert knowledge as prior. Although the availability of expert knowledge plays a role in life sciences, none of these approaches strongly emphasizes domain knowledge about features, nor do they involve specific prior constraints defined by the user.

In this work, we propose a novel Bayesian approach to feature selection that incorporates expert knowledge and maintains considerable model generality. We aim to fill the gap between data-driven feature selection on one side and purely expert-focused feature selection on the other side. Our presented probabilistic approach, UBayFS, combines a generic ensemble feature selection framework with the exploitation of domain knowledge. Hence, it supports interpretability and improves the stability of the results. For this purpose, feature importance votes from independent elementary feature selectors are merged with constraints and feature weights specified by the expert. Constraints may be of a general type, such as selecting a maximum number of features or blocks of features. Both inputs, likelihood and prior, are aggregated in a sound statistical framework, producing a posterior probability distribution over all possible feature sets. We use a Genetic Algorithm for discrete optimization to efficiently optimize the posterior feature set in high-dimensional datasets. In an extensive experimental evaluation, we analyze UBayFS in a variety of model setups involving prior knowledge and constraints. Results on open-source datasets are benchmarked against state-of-the-art feature selectors in terms of predictive performance and stability, underlining the potential of UBayFS.

Notations We will denote vectors by bold, uncapitalized, and matrices by bold, capitalized letters. Non-bold, uncapitalized letters indicate scalars or functions, and non-bold, capitalized letters indicate sets or constants. $||.||_1$ denotes the $L1$-norm. $[N]$ is an abbreviation of the set of indices $1,\dots,N$. The N-dimensional vector of ones will be written as $\mathbf{1}_N$. Furthermore, we refer to sets of features by their feature indices, such as $S \subseteq [N]$, or by a binary membership vector $\boldsymbol{\delta}^S \in \{0,1\}^N$ with components $(\boldsymbol{\delta}^S)_n = \begin{cases} 1 & \text{if } n \in S, \\ 0 & \text{otherwise.} \end{cases}$

2 User-guided ensemble feature selector

Given a finite set of N features, the goal of UBayFS is to find an optimal subset of feature indices $S^\star \subset [N]$, or, equivalently, $\boldsymbol{\delta}^\star = \boldsymbol{\delta}^{S^\star} \in \{0,1\}^N$. We assume that information is available from

1. Training data to collect evidence by conventional data-driven feature selectors—we denote this as information from data $\boldsymbol{y}$,
2. The user's domain knowledge encoded as subjective beliefs $\boldsymbol{\alpha} \in \mathbb{R}^N$ about the importance of features, where $\alpha_n > 0$ for all $n \in [N]$, and
3. Side constraints, given as inequality system $\boldsymbol{A\delta} \leq \boldsymbol{b}$, to ensure that the obtained feature set conforms with practical requirements and restrictions.

UBayFS assumes a feature importance vector $\boldsymbol{\theta} \in [0,1]^N$, $||\boldsymbol{\theta}||_1 = 1$, which is probabilistic and not directly observable, such that evidence about $\boldsymbol{\theta}$ is collected from data $\boldsymbol{y}$ and prior weights $\boldsymbol{\alpha}$. Our model aims to maximize the accumulated importances $\boldsymbol{\delta}^T\boldsymbol{\theta}$ of the selected features subject to side constraints $\boldsymbol{A\delta} \leq \boldsymbol{b}$. More specifically, we maximize the utility function

$$U(\boldsymbol{\delta},\boldsymbol{\theta}) = \boldsymbol{\delta}^T\boldsymbol{\theta} - \lambda\kappa(\boldsymbol{\delta}),\ \lambda > 0, \tag{1}$$

where $\kappa(\boldsymbol{\delta})$ is a non-negative scalar function which penalizes the degree of violation of the constraints. The precise form of $\kappa(.)$ will be given later. Clearly, we require that $\kappa(\boldsymbol{\delta}) = 0$, if $\boldsymbol{A\delta} \leq \boldsymbol{b}$ is satisfied. In Eq. 1, $\lambda > 0$ plays the role of a Lagrange parameter, $\lambda\kappa(\boldsymbol{\delta})$ increases the amount of penalization imposed on a feature set violating the constraints. In terms of statistical decision theory, a Bayes decision should maximize the posterior expected utility

$$\mathbb{E}_{\theta|y}[U(\boldsymbol{\delta}, \boldsymbol{\theta}(\boldsymbol{y}))] = \boldsymbol{\delta}^T \mathbb{E}_{\theta|y}[\boldsymbol{\theta}(\boldsymbol{y})] - \lambda\kappa(\boldsymbol{\delta}) \longrightarrow \max_{\delta \in \{0,1\}^N}. \tag{2}$$

We denote the optimal feature set according to Eq. 2 by $\boldsymbol{\delta}^{\star}$. The importance parameter $\boldsymbol{\theta}$ is inferred from data from elementary feature selectors trained on subsets of the dataset, summarized as $\boldsymbol{y}$, as well as prior feature importance scores $\boldsymbol{\alpha}$. Thus, the posterior probability distribution of $\boldsymbol{\theta}$ given observations $\boldsymbol{y}$, $p(\boldsymbol{\theta}|\boldsymbol{y})$, is decomposed using Bayes' theorem into

$$p(\boldsymbol{\theta}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\theta}) \cdot p(\boldsymbol{\theta}), \tag{3}$$

where $p(\boldsymbol{y}|\boldsymbol{\theta})$ describes the model likelihood (evidence from elementary feature selector model) and $p(\boldsymbol{\theta})$ describes the density of a prior distribution (user domain knowledge).

The remainder of this Section focuses on determining the missing model components to define the problem stated in Eq. (2), comprising (a) the feature importances $\boldsymbol{\theta}$, discussed in Sect. 2.1 and 2.2, and (b) the function κ, discussed in Sect. 2.3. Finally, Sect. 2.4 suggests the discrete optimization procedure to solve Eq. (2).

2.1 Ensemble feature selection as likelihood

To collect information about feature importances from the given dataset, we train an ensemble of M elementary feature selectors of the same model type on distinct training subsets. The selection of a feature index set $\boldsymbol{\delta}^{(m)}$ comprising a constant number of $l = \|\boldsymbol{\delta}^{(m)}\|_1$ features in each elementary model m out of a total of M models can be interpreted as a result of drawing l balls from an urn, where each ball has a distinct color representing one feature $n \in [N]$. Over all elementary models, $\boldsymbol{y}$ collects the counts of each feature being selected, resulting in a count vector in

$$\boldsymbol{y} = \sum_{m=1}^{M} \boldsymbol{\delta}^{(m)} \in \{0, \dots, M\}^N. \tag{4}$$

Each elementary feature selector delivers a proposal for an optimal feature set. Thus, we let the frequency of drawing a feature throughout $\boldsymbol{\delta}^{(1)}, \dots, \boldsymbol{\delta}^{(M)}$ represent its *importance* by defining the latent importance parameter vector $\boldsymbol{\theta} \in [0, 1]^N$, $\|\boldsymbol{\theta}\|_1 = 1$, as the success probabilities of sampling each feature in an individual urn draw. In a statistical sense, we interpret the result from each elementary feature selector as realization from a multinomial distribution with parameters $\boldsymbol{\theta}$ and l.[1] This multinomial setup delivers the likelihood $p(\boldsymbol{y}|\boldsymbol{\theta})$ as joint probability density

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \prod_{m=1}^{M} f_{\text{mult}}(\boldsymbol{\delta}^{(m)}; \boldsymbol{\theta}, l), \tag{5}$$

where $f_{\text{mult}}(\boldsymbol{\delta}^{(m)}; \boldsymbol{\theta}, l)$ denotes the density of a multinomial distribution with success probabilities $\boldsymbol{\theta}$ and a number of l urn draws. Relevant notations are summarized in Table 1.

[1] The exact way to describe this procedure is a multivariate hypergeometric distribution, since each feature occurs at most once in a set, but an approximation using the multinomial distribution facilitates computation.

Table 1 Notations for likelihood parameters

Input and elementary models	
$n \in [N]$	Feature indices
$m \in [M]$	Elementary models
$\boldsymbol{\delta} \in \{0,1\}^N$	Feature index set
$\boldsymbol{\theta} \in \Theta \subset [0,1]^N$	Feature importances
$\boldsymbol{y} \in \{0,\dots,M\}^N$	Feature counts

2.2 Expert knowledge as prior weights

To constitute the prior distribution, UBayFS uses expert knowledge as a-priori weights of features. Since the domain of the distribution of feature importances $\boldsymbol{\theta}$ is defined to be a simplex $\boldsymbol{\theta} \in \Theta \subset [0,1]^N$, $\|\boldsymbol{\theta}\|_1 = 1$, the Dirichlet distribution is a natural choice as prior distribution, which is widely used in data science problems, such as Nakajima et al. (2014). Thus, we initially assume that a-priori

$$p(\boldsymbol{\theta}) = f_{\text{Dir}}(\boldsymbol{\theta};\boldsymbol{\alpha}), \tag{6}$$

where $f_{\text{Dir}}(\boldsymbol{\theta};\boldsymbol{\alpha})$ denotes the density of the Dirichlet distribution with positive $\boldsymbol{\alpha} = (\alpha_1,\dots,\alpha_N)$. Since the Dirichlet distribution is a conjugate prior of the multinomial distribution, the posterior distribution results in a Dirichlet type, again, see DeGroot (2005). Thus, it holds for the posterior density that

$$p(\boldsymbol{\theta}|\boldsymbol{y}) \propto f_{\text{Dir}}(\boldsymbol{\theta};\boldsymbol{\alpha}^{\circ}), \tag{7}$$

where the parameter update is obtained in closed form by

$$\boldsymbol{\alpha}^{\circ} = \boldsymbol{\alpha} + \boldsymbol{y}. \tag{8}$$

In case of integer-valued prior weights $\boldsymbol{\alpha}$, they may be interpreted as pseudo-counts in the context of modelling success probabilities in an urn model—comparable to the information gained if the corresponding counts were observed in a multinomial data sample. In UBayFS, we obtain $\boldsymbol{\alpha}$ as feature weights provided by the user. If no user knowledge is available, the least informative choice is to specify uniform counts with a small positive value, such as $\boldsymbol{\alpha}_{\text{unif}} = 0.01 \cdot \mathbf{1}_N$.

2.2.1 Generalized Dirichlet model

Even though the presented Dirichlet-multinomial model is a popular choice due to its favorable statistical properties, it implicitly assumes that classes (in our case, features) are mutually independent. However, high-dimensional datasets frequently involve complex correlation structures between the features. To account for this aspect, we generalize the setup by replacing the Dirichlet prior distribution with some generalized Dirichlet distribution. The highest level of generalization is achieved by Hankin (2010), who introduced the hyperdirichlet distribution, which may take arbitrary covariance structures into account. The hyperdirichlet distribution maintains the conjugate prior property with respect to the multinomial likelihood, and thus, inference is tractable; however, the analytical expression of the expected value involves the intractable normalization constant and, as a result, requires numerical means such as Monte-Carlo Markov Chain (MCMC) methods, which may face computational challenges due to the high dimensionality of the problem.

A compromise between the complexity of the problem and the flexibility of the covariance structure is given by an earlier version of the generalized Dirichlet distribution by Wong (1998), which is a special case of the hyperdirichlet setup, but more general than the standard Dirichlet distribution. In addition to the properties of the hyperdirichlet distribution, the expected value of the generalized Dirichlet distribution can be directly evaluated from the distribution parameters. Section 3 provides an experimental evaluation of the proposed variants to account for covariance structures in the UBayFS model.[2]

2.3 Side constraints as regularization

Practical setups may require that a selected feature set fulfills certain consistency requirements. These may involve a maximum number of selected features, a low mutual correlation between features, or a block-wise selection of features. UBayFS enables the feature selection model to account for such requirements via a function κ, which incorporates a system of K inequalities restricting the feature set $\boldsymbol{\delta}$, $\boldsymbol{A}\boldsymbol{\delta}-\boldsymbol{b}\leq 0$, where $\boldsymbol{A}\in\mathbb{R}^{K\times N}$ and $\boldsymbol{b}\in\mathbb{R}^{K}$. Each single constraint $k\in[K]$ can be evaluated via an inadmissibility function $\kappa_k(.)$, such that

$$\kappa_k(\boldsymbol{\delta}) = \begin{cases} 0 \text{ if } \left(\boldsymbol{a}^{(k)}\right)^T\boldsymbol{\delta} - b^{(k)} \leq 0 \\ 1 \text{ otherwise,} \end{cases} \tag{9}$$

where $\boldsymbol{a}^{(k)}$ is the k-th row vector of $\boldsymbol{A}$ and $b^{(k)}$ the k-th element of $\boldsymbol{b}$. UBayFS generalizes the setup by relaxing the constraints: in case that a feature set $\boldsymbol{\delta}$ violates a constraint, it shall be assigned a higher penalty rather than being excluded completely. This effect is achieved by replacing $\kappa_k(.)$ with a relaxed inadmissibility function $\kappa_{k,\rho}(.)$ based on a logistic function with relaxation parameter $\rho\in\mathbb{R}^{+}\cup\{\infty\}$:

$$\kappa_{k,\rho}(\boldsymbol{\delta}) = \begin{cases} 0 & \text{if } \left(\boldsymbol{a}^{(k)}\right)^T\boldsymbol{\delta} \leq b^{(k)} \\ 1 & \text{if } \left(\boldsymbol{a}^{(k)}\right)^T\boldsymbol{\delta} > b^{(k)} \wedge \rho = \infty \\ \frac{1-\xi_{k,\rho}}{1+\xi_{k,\rho}} & \text{otherwise,} \end{cases} \tag{10}$$

with $\xi_{k,\rho} = \exp\left(-\rho\left(\left(\boldsymbol{a}^{(k)}\right)^T\boldsymbol{\delta} - b^{(k)}\right)\right)$. Fig. 1 illustrates that a large parameter $\rho\longrightarrow\infty$ lets the inadmissibility converge pointwise towards the associated hard constraint. A low ρ changes the shape of the penalization to an almost constant function in a local neighborhood around the decision boundary, such that only a minor difference is made between feature sets that fulfill and those that violate a constraint.[3]

Finally, the joint inadmissibility function $\kappa(.)$ aggregates information from all constraints

$$\kappa(\boldsymbol{\delta}) = 1 - \prod_{k=1}^{K}\left(1-\kappa_{k,\rho}(\boldsymbol{\delta})\right), \tag{11}$$

[2] Details on the generalized prior distributions are provided in Appendix A.

[3] for a proof see Appendix A

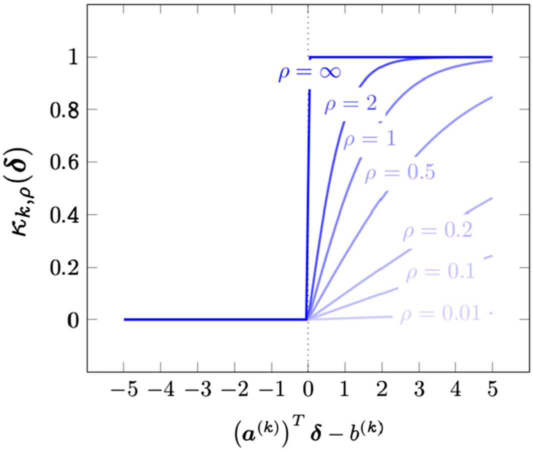


Fig. 1 The effect of ρ on $\kappa_{k,\rho}$ for soft constraints

Table 2 Notations used for prior parameters

Prior parameters	
$\boldsymbol{\alpha}, \boldsymbol{\alpha}^\circ \in \mathbb{R}^N$	Prior/posterior weights
$k \in [K]$	Constraint index
$\boldsymbol{A} \in \mathbb{R}^{K \times N}, \boldsymbol{b} \in \mathbb{R}^K$	Inequality system
$\boldsymbol{\rho} \in \mathbb{R}^K$	Relaxation parameters
$\kappa(.) : \{0,1\}^N \rightarrow [0,1]$	Joint inadmissibility

which originates from the idea that $\kappa = 1$ (maximum penalization) if at least one $\kappa_{k,\rho} = 1$, while $\kappa = 0$ (no penalization) if all $\kappa_{k,\rho} = 0$.

Note that different relaxation parameters may be used to prioritize the constraints among each other, hence κ involves a parameter vector $\boldsymbol{\rho} = (\rho_1, \dots, \rho_K)$. Notations related to prior parameters and constraints are summarized in Table 2.

2.3.1 Feature decorrelation constraints

Commonly, feature sets with low mutual correlations are preferred since they tend to contain less redundant information. A special case of prior constraints can be defined to enforce that such feature sets are selected. We will refer to such constraints as decorrelation constraints. Decorrelation constraints are pairwise cannot-link constraints between highly correlated features, i.e., features i and j with a correlation coefficient $\tau_{i,j}$ exceeding a predefined absolute threshold $|\tau_{i,j}| > \tau$. For each such pair $i,j \in [N], i \neq j$, a constraint is added to the constraint system as follows: the vector $\boldsymbol{a}$ with elements

$$a_n = \begin{cases} 1 & \text{if } n \in \{i,j\} \\ 0 & \text{else,} \end{cases} \tag{12}$$

and an element $b = 1$ are appended to $\boldsymbol{A}$ and $\boldsymbol{b}$, respectively. We set the shape parameter ρ to the odds ratio of the absolute correlation coefficient $\tau_{i,j}$, given as

$$\rho = \frac{|\tau_{i,j}|}{1 - |\tau_{i,j}|}. \tag{13}$$

Hence, features with higher absolute correlations are assigned higher penalties and vice versa. As a result, the selected feature set contains features with lower mutual correlations.[4]

2.3.2 Feature block priors

User knowledge may as well be available for *feature blocks* rather than for single features. Feature blocks are contextual groups of features, such as those extracted from the same source in a multi-source dataset. It can be desirable to select features from a few distinct blocks so that the model does not depend on all sources at once. While prior weights can be trivially assigned on block level, we transfer the concept of side constraints to feature blocks.

Feature blocks are specified via a block matrix $\boldsymbol{B} \in \{0,1\}^{W \times N}$, where 1 indicates that the feature $n \in [N]$ is part of block $w \in [W]$ and 0, else. Even though a full partition of the feature set is common, feature blocks are neither required to be mutually exclusive, nor exhaustive. Along with the block matrix $\boldsymbol{B}$, an inequality system between blocks consists of a matrix $\boldsymbol{A}^{\text{block}} \in \mathbb{R}^{K \times W}$ and a vector $\boldsymbol{b}^{\text{block}} \in \mathbb{R}^{K}$. To evaluate whether a block is selected by a feature set $\boldsymbol{\delta}$, we define the block selection vector $\boldsymbol{\delta}^{\text{block}} \in \{0,1\}^{W}$, given by

$$\boldsymbol{\delta}^{\text{block}} = \left(\boldsymbol{B}\boldsymbol{\delta} \geq \mathbf{1}_{W}\right), \tag{14}$$

where $\geq$ refers to an element-wise comparison of vectors, delivering 1 for a component, if the condition is fulfilled, and 0, otherwise. In other words, a feature block is selected, if at least one feature of the corresponding block is selected. Although block constraints introduce non-linearity into the system of side constraints, they can be used in the same way as linear constraints between features and integrated into the joint inadmissibility function κ.

2.4 Optimization

Exploiting the conjugate prior property, the posterior density of $\boldsymbol{\theta}$ can be expressed as a Dirichlet, generalized Dirichlet or hyperdirichlet distribution, respectively. The expected value $\mathbb{E}_{\boldsymbol{\theta}}[\boldsymbol{\theta}]$ can be computed either in a closed-form expression (Dirichlet or generalized Dirichlet) Wong (1998), or simulated via a sampling procedure (hyperdirichlet) Hankin (2010). It remains to solve the discrete optimization problem in Eq. (2) as a final step.

[4] We suggest to use Spearman's rho as correlation coefficient, since it is robust (in contrast to Pearson's correlation coefficient) and faster to compute than Kendall's tau.

Algorithm 1 Probabilistic sampling algorithm to initialize GA.

```
Require: $\boldsymbol{\alpha}^\circ$, $\boldsymbol{A}$, $\boldsymbol{b}$, $\boldsymbol{\rho}$, sample size $Q$
G ← {}
for $q \in [Q]$ do
    $\boldsymbol{\delta} \leftarrow (0, 0, \ldots, 0)$
    generate a permutation $\pi$ on $[N]$ by sampling $N$ times without replacement with probabilities proportional to $\boldsymbol{\alpha}^\circ$
    for $i = \pi(1), \ldots, \pi(N)$ do
        define $\boldsymbol{\delta}^\dagger$ as $\delta_n^\dagger \leftarrow \begin{cases} \delta_n & n \neq i \\ 1 & n = i \end{cases}$ for each $n \in [N]$
        sample $u \sim \mathrm{Unif}_{[0,1]}$
        if $u \leq r_{\boldsymbol{\delta}^\dagger, \boldsymbol{\delta}}$ then
            update $\boldsymbol{\delta} \leftarrow \boldsymbol{\delta}^\dagger$
        end if
    end for
    $G \leftarrow G \cup \{\boldsymbol{\delta}\}$
end for
return $G$
```

Since an analytical minimization of the resulting knapsack problem is not feasible, we determine a numerical optimum $\boldsymbol{\delta}^\star$ by using discrete optimization: we deploy the Genetic Algorithm (GA) described by Givens and Hoeting (2012). To guarantee a fast convergence towards an acceptable solution, it is beneficial to provide initial samples, which are good candidates for the final solution. For this purpose we propose a probabilistic sampling algorithm, Alg. 1: In essence, the algorithm creates a random permutation of all features, $\pi : [N] \to [N]$, by weighted and ordered sampling without replacement. The weights represent the posterior parameter vector $\boldsymbol{\alpha}^\circ$. Then, the algorithm iteratively accepts or rejects feature $\pi(n)$ with a success probability

$$r_{\boldsymbol{\delta}^\dagger, \boldsymbol{\delta}} = \begin{cases} \frac{1-\kappa(\boldsymbol{\delta}^\dagger)}{1-\kappa(\boldsymbol{\delta})} & \text{if } \kappa(\boldsymbol{\delta}) < 1 \\ 0 & \text{else,} \end{cases} \tag{15}$$

denoting the admissibility ratios of feature sets with and without feature $\pi(n)$. The generated sample accounts for high feature weights by low ranks, resulting in a higher probability to be accepted in the acceptance/rejection step.

The Genetic Algorithm (GA) for discrete optimization is initialized using Algorithm 1. Starting with an initial set of feature membership vectors $\left\{\boldsymbol{\delta}^0 \in \{0,1\}^N\right\}$, GA creates new vectors $\boldsymbol{\delta}^t \in \{0,1\}^N$ as pairwise combinations of two preceding vectors $\boldsymbol{\delta}^{t-1}$ and $\tilde{\boldsymbol{\delta}}^{t-1}$ in each iteration $t \in [T]$. A combination refers to sampling component $\boldsymbol{\delta}_n^t$ from either $\boldsymbol{\delta}_n^{t-1}$ or $\tilde{\boldsymbol{\delta}}_n^{t-1}$ in a uniform way and adding minor random mutations to single components. The posterior density serves as fitness when deciding which vectors $\boldsymbol{\delta}^{t-1}$ and $\tilde{\boldsymbol{\delta}}^{t-1}$ from iteration $t-1$ should be combined to $\boldsymbol{\delta}^t$ — the fitter, the more likely to be part of a combination.

The runtime of GA depends linearly on the population size, and the number of iterations. A good trade-off between runtime and convergence properties is important—a small population size, for example, might lead to faster convergence but might get trapped towards a local minimum. Further, the runtime is dependent on the complexity to compute the fitness function, which in turn depends on the dimensionality of the problem.

3 Experiments and results

Our numerical experiments evaluate the performance, flexibility, and applicability of UBayFS in two parts: first, a study conducted on synthetic datasets demonstrates the properties of the various model parameters, including

a. The number of elementary models M (1a),
b. The prior weights $\boldsymbol{\alpha}$ in a block-wise setup (1b),
c. The constraint types and their shapes ρ in a block-wise setup (1c), as well as
d. The type of prior distribution to account for feature dependencies (1d).

The second part of our experiment is conducted on real-world classification datasets from the life science domain. In a comparison with state-of-the-art ensemble feature selectors, we demonstrate that UBayFS delivers similar model performances. Our setups include ordinary and block feature selection without prior knowledge to ensure a fair comparison. Finally, we conduct a case study with expert knowledge available from biological investigations, and demonstrate how informative priors increase model performance in practice.

3.1 Default parameters

Six types of feature selectors are evaluated as elementary models for UBayFS:

- Minimum Redundancy Maximum Relevance (mRMR) Ding and Peng (2005),
- Fisher score Bishop (1995),
- Decision tree for classification Breiman et al. (1984),
- Recursive feature elimination (RFE) Guyon et al. (2002),
- Hilbert-Schmidt Independence Criterion Lasso (HSIC) Yamada et al. (2014),
- Lasso Tibshirani (1996).

However, the main focus of the present work is to evaluate the generic concept of UBayFS rather than to provide an in-depth analysis of these elementary feature selectors.

Our implementation of UBayFS in R (R Core Team , 2020)[5] uses the Genetic Algorithm package authored by Scrucca (2013) with $T = 100$ and $Q = 100$; in most cases, convergence is achieved after around ten iterations. By default, each UBayFS setup comprises an uninformative prior with $\alpha_n = 0.01$ for all $n \in [N]$, and a max-size constraint instructing to select b_{MS} features, which is determined individually for each dataset. Thus, by default, the constraint system is given as:

$$\boldsymbol{A} = (1\ 1\ \ldots\ 1), \boldsymbol{b} = b_{\text{MS}}, \boldsymbol{\rho} = 1.$$

[5] For implementation and experimental setups, see https://github.com/annajenul/UBayFS and https://github.com/annajenul/UBayFS_experiments; for details, see Appendix B.

No further user knowledge or side constraints are introduced unless stated explicitly in the particular setups. Each setup is executed in $I = 10$ independent runs $i \in [I]$, representing distinct random splits of the dataset $\mathcal{D}$ into train data $T^{(i)}_{\text{train}}$ and test data $T^{(i)}_{\text{test}} = \mathcal{D} \setminus T^{(i)}_{\text{train}}$ (stratified 75%/25% split).

3.2 Evaluation metrics

For the synthetic datasets, performance is measured by the F1 score of correctly / incorrectly selected features since the ground truth about the relevance of features is known from the simulation procedure. For real-world data, F1 scores refer to the predictive results obtained by training a classification model after feature selection, and judge the feature selection quality indirectly. Furthermore, all experiments evaluate the *stability* measure by Nogueira et al. (2018) across I independent feature selection runs. Stability ranges asymptotically in [0, 1], where 1 indicates that the same features are selected in every run (perfectly stable). *Runtime*[6] refers to the time the model requires to perform feature selection, including elementary model training and optimization, but excluding any predictive model trained on top of the feature selection results. Since prior parameters have a minor influence on the runtime, times will not be provided for experiments investigating these aspects.

3.3 Experiment 1: simulation study

To investigate major properties of UBayFS, we simulate four different datasets:

i. An additive model (experiment 1a) similar to *Data1* in Yamada et al. (2014), composed of a $(x_1, \dots, x_{1000}) \sim 1000 \times 1000$ data matrix simulated from a Gaussian distribution $N(\mathbf{0}_{1000}, \boldsymbol{I}_{1000})$, and a binary target variable

$$f(\boldsymbol{x}, \varepsilon) = g(-2\sin(2x_1) + x_2^2 + x_3 + \exp(-x_4) + \varepsilon),$$

where $x_1, \dots, x_4$ denote the features 1 to 4 and $\varepsilon \sim N(0, 1)$. The function g transforms z into a class variable by

$$g(z) = \begin{cases} 1 & \text{if } z \geq 0, \\ 0 & \text{otherwise;} \end{cases}$$

ii. A non-additive model (experiment 1a) similar to *Data2* in Yamada et al. (2014), equivalent to the setup of i., except for a multiplicative target variable

$$f(\boldsymbol{x}, \varepsilon) = g(x_1 \cdot \exp(2x_2) + x_3^2 + \varepsilon);$$

iii. A simulated dataset (experiment 1b, 1c) with group structure among the features, produced via *make_classification* (Pedregosa , 2011), delivering a 512×256 dataset with 8 feature blocks à 32 features—4 of these blocks contain relevant features (4 important features per block), 2 blocks contain redundant features representing arbitrary linear combinations of the relevant features (3 redundant features per block);

[6] CentOS Linux 7.9.2009, Intel Xeon(R) CPU E5-2650 @ 2.60GHz, 3 GB RAM, R v3.6.0.

iv. Another dataset simulated via *make_classification*, comprising 32 features in total (16 important, 16 redundant) without block structure. This smaller dataset (64 × 32) has a complicated correlation structure due to the high number of redundant features and is used to evaluate UBayFS variants that take feature dependence into account (experiment 1d).

The maximum number of selected features b_{MS} is set to the ground truth number of relevant features, i.e. $b_{\mathrm{MS}} = 4$ (dataset i.), $b_{\mathrm{MS}} = 3$ (dataset ii.), and $b_{\mathrm{MS}} = 16$ (datasets iii. and iv.), respectively. The default constraint shape parameters for MS is set to $\rho_{\mathrm{MS}} = 1$. Unless otherwise stated, the prior weights are set to a constant, uninformative value of $\alpha = 0.01$ for all features.

In addition to the constraint shape ρ associated with a single constraint, λ balances the overall impact of side constraints with the Dirichlet-multinomial model. A small parameter $\lambda < 1$ is not recommended since a lack of influential constraints (including the MS constraint) results in selecting all features due to an unregularized utility function U. On the other hand, a high λ has a similar effect as setting all shape parameters uniformly to $\rho = \infty$; thus, all constraints are required to be fulfilled. In this study, λ has only a minor impact on the resulting model metrics and, therefore, is set to $\lambda = 1$.

3.3.1 Experiment 1a—likelihood parameters

Figure 2 demonstrates the effect of an increasing number of elementary models M to build the feature selector. M represents the parameter to steer the likelihood. Due to their excessive runtimes, HSIC and RFE are computed only for $M \leq 10$, while all other elementary feature selectors are evaluated for up to $M = 200$.

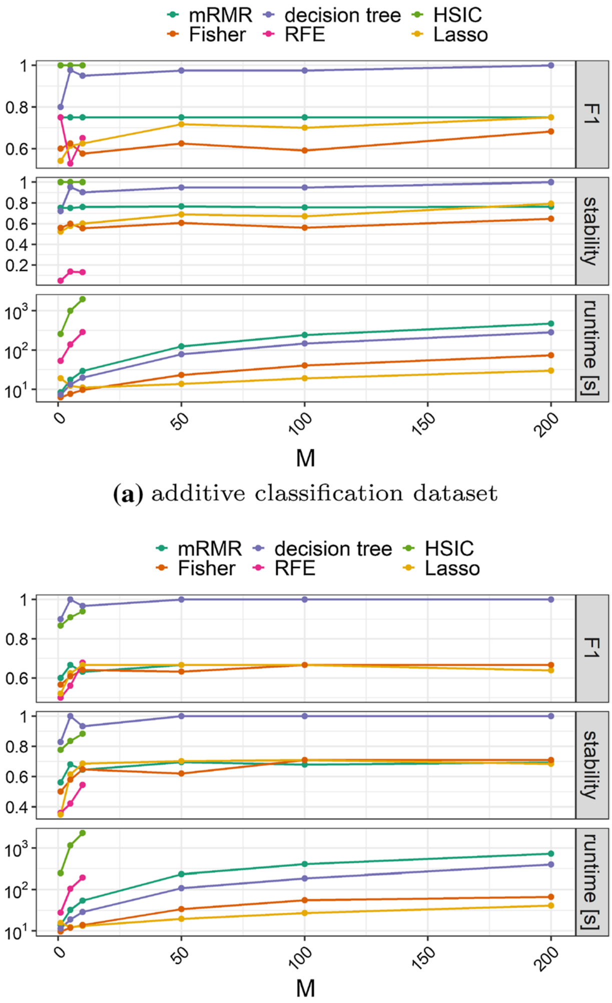
As expected, a higher M contributes largely to the runtime of the model, which increases linearly. In contrast, both F1 scores and stability values begin to saturate at around $M = 50$ to $M = 100$ models. Even though large ensembles are intractable with HSIC and RFE, small ensembles with $M = 5$ allow HSIC to retrieve almost all features, whereas simpler elementary feature selectors struggle to achieve high performances and stabilities even at higher levels of M. We conclude that large M does not necessarily improve the results but significantly impacts the runtime. Thus $M \approx 100$ appears to be a reasonable choice in the subsequent settings, except for HSIC and RFE, where $M = 5$ will be set as a default.

3.3.2 Experiment 1b—"correct" and "incorrect" prior weights

To investigate the effect of prior weights $\boldsymbol{\alpha}$, we alter the prior weights in dataset iii. by feature block. A constant prior weight α_R is assigned to all features from relevant blocks, i.e., blocks 1-4 containing informative and non-informative features. In contrast, features from blocks 5-8 (containing only non-informative features) are assigned a constant prior weight α_{-R}—thereby, we simulate that the expert has approximate, yet not exact beliefs about feature relevance. By assigning higher prior weights $\alpha_R > \alpha_{-R}$, the experiment simulates an agreement between the expert belief and the ground truth ("correct prior"), while a lower $\alpha_R < \alpha_{-R}$ represents "wrong" prior information ("incorrect prior"). To simulate correct and incorrect prior knowledge at different levels, we increase α_R while setting α_{-R} to the default value 0.01, and vice versa.

Figure 3 illustrates that, as expected, feature selection performance in terms of F1 scores (evaluated with respect to the ground truth features) increases for higher α_R and decreases

Fig. 2 Different numbers of elementary models M



(a) additive classification dataset

(b) non-additive classification dataset

for higher α_{-R}. Thus, across all elementary feature selectors, an improvement of the uninformative case $\alpha_R = \alpha_{-R} = 0.01$ can be achieved by an informative prior, if the prior represents a reasonable overlap with reality—this holds even though the relevant blocks also contain uninformative features, which are incremented by α_R as well. On the other hand, erroneous prior knowledge can impact the feature selection results negatively. In contrast to the feature-wise F1 scores, stability remains mostly unaffected from strong prior knowledge on relevant or irrelevant blocks—incorrect prior knowledge merely tends to decrease stability to a minor degree.

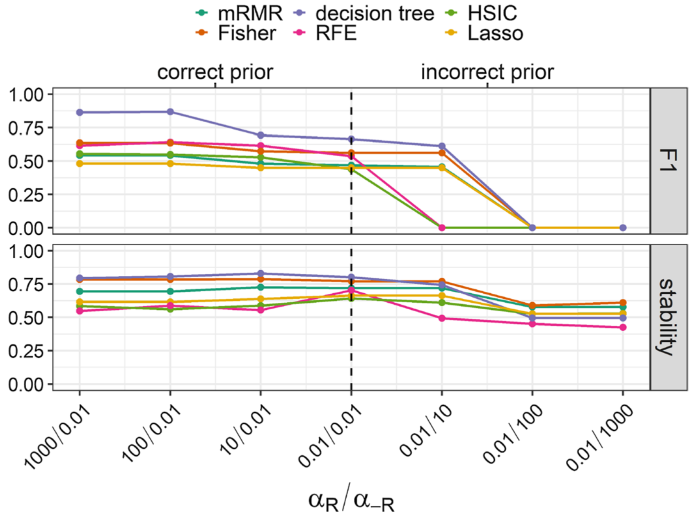


Fig. 3 Different prior weights assigned to relevant blocks, α_R, and to non-relevant blocks, α_{-R}

3.3.3 Experiment 1c—side constraints

We investigate the following opposite constraint types:

- *Block-max-size* (BMS): features are selected from at most b_{BMS} distinct blocks, and
- *Max-per-block* (MPB): at most b_{MPB} features are selected from each block.

BMS is designed to enforce a clustering behavior, where all selected features originate from a maximum number of $b_{\text{BMS}} = 4$ blocks. On the other hand, MPB aims to disperse the selection, indicating that a maximum number of $b_{\text{MPB}} = 2$ features per block is favorable. The strength of these constraints is steered via the corresponding shape parameters ρ_{BMS} and ρ_{MPB}, respectively, while $\rho = 0$ indicates that a constraint is omitted. From a default case of $\rho_{\text{BMS}} = \rho_{\text{MPB}} = 0$ (no block constraints), we investigate the behavior of UBayFS under one of the two constraints at a time at an increasing level of ρ_{BMS} or ρ_{MPB}.

Fig. 4 illustrates how the opposite side constraints BMS and MPB affect the model at different levels of relaxation parameters. Both constraint types have a slightly negative impact on the outcome in terms of F1 and stability. This is caused by the fact that the "best" feature set has to be determined under a side constraint, which is not compatible with the ground truth—the ground truth defines 16 features out of four distinct blocks to be relevant, which cannot be covered by any of the constraints. Therefore, we can observe that UBayFS can handle such scenarios and still deliver appropriate and near-optimal solutions.

3.3.4 Experiment 1d—between-feature correlations

In Sect. 2, multiple variants were discussed to account for datasets with a given correlation structure. On the one hand, the UBayFS framework permits to account for between-feature correlations via a generalization of the prior distribution; on the other hand, we may enforce that the highly correlated features should not be selected jointly via a decorrelation constraint. Both variants are different insofar as generalized priors aim to deliver a more

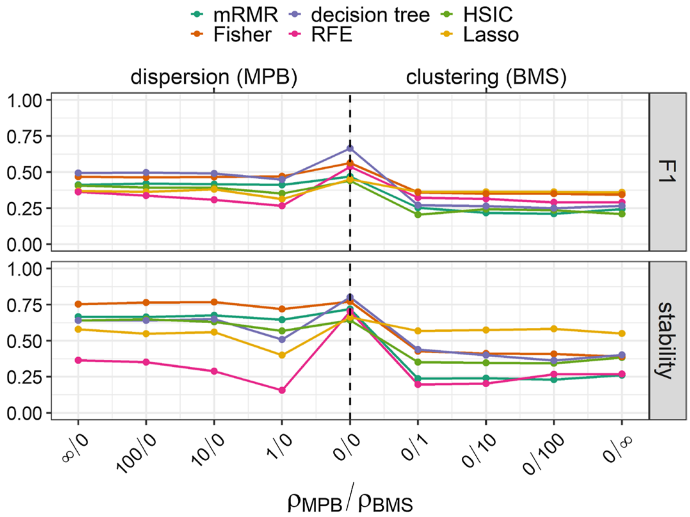


Fig. 4 Different prior constraints assigned to blocks: MPB (maximum one feature per block) and BMS (block max-size) constraint types at distinct levels of ρ. The special case $\rho = 0$ indicates that the corresponding constraint is omitted

appropriate estimation of the expected feature importances by correcting for dependencies in the observed feature sets, while decorrelation constraints directly affect the optimization procedure for $\boldsymbol{\delta}$.

In this experiment, we investigate both possibilities to account for between-feature correlations, along with combinations of both: we set a decorrelation constraint between all features with a mutual Spearman correlation $\tau > 0.4$ as described in Sect. 2.3, such that joint selection of highly correlated features is penalized. Further, we apply the following prior setups:

- Dirichlet prior distribution (default),
- Generalized Dirichlet distribution Wong (1998),
- Hyperdirichlet distribution Hankin (2010).

Our experiment involves all combinations of prior setups with and without decorrelation constraint, executed on dataset iv. To measure the effect of decorrelation, we further evaluate the redundancy rate (RED) Zhao et al. (2010), defined as the average absolute Pearson correlation among selected features. A small RED is commonly preferred in practical setups.

The results in Fig. 5 show that neither feature-wise F1 scores nor stabilities change significantly between the prior models. Thus, the default Dirichlet model seems sufficient in practice. However, introducing decorrelation constraints has a slightly negative impact on stability, while yielding a small improvement in F1 scores and RED. Nonetheless, the most significant change between the variants can be observed with respect to runtime, which reflects the high computational burden associated with the hyperdirichlet prior model—even on a small dataset, the runtimes show a significant increase on a logarithmic scale. Thus, higher-dimensional datasets can only be tackled at an enormous computational cost with the hyperdirichlet setup.

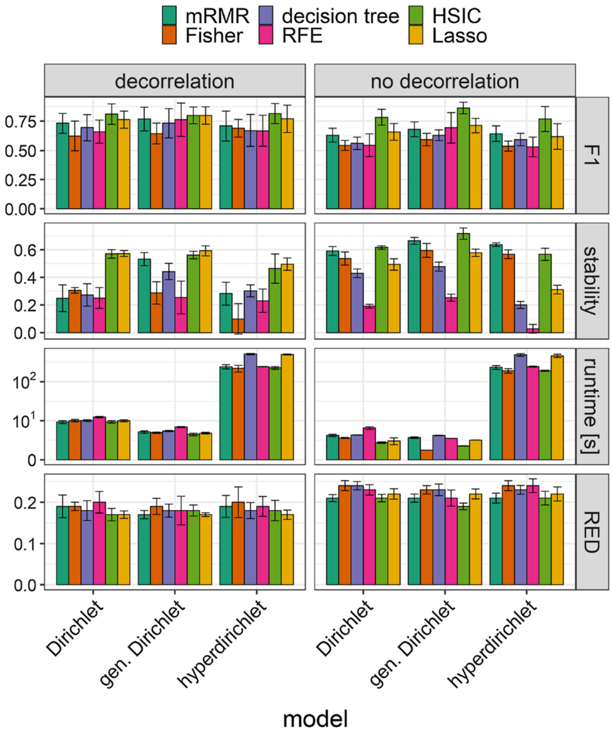


Fig. 5 Different setups to account for dependence structures between features

Table 3 Real-world binary classification datasets from the life science domain used for experimental evaluation. For p53, a stratified subset out of > 16000 rows was used from the original dataset for this experiment

Dataset / source	# Features	# Rows	b_{MS}	# Blocks	b_{BMS}
Breast cancer wisconsin (BCW) Wolberg and mangasarian (1990)	30	569	5	3	1
Heart disease (HD) Detrano (1989)	46	101	5	–	–
Mice protein expression (MPE) Higuera et al. (2015)	77	552	5	–	–
Colon gene expression (COL) Yang and Zou (2015)	100	62	5	20	2
LSVT voice rehabilitation Tsanas (2013)	310	126	10	14	2
p53 Danziger (2006)	5409	351	20	2	1
Prostate (PRO) Singh (2002)	6033	102	20	–	–
Leukaemia (LEU) Golub (1999)	7129	72	20	–	–
Lung cancer (LUNG) Gordon (2002)	12533	181	100	–	–

3.4 Experiment 2: real-world datasets

Numerical studies are conducted on eight open-source datasets presenting binary classification problems from the life science domain, see Table 3. For simplicity and due to extensive runtimes, we restrict the choice of the elementary feature selector for UBayFS to mRMR, Fisher, and decision tree with an uninformative prior, an MS constraint, and $M = 100$. The number of selected features is specified according to the size of the dataset (b_{MS} = 5 / 10 / 20 / 100 for datasets with fewer than 100 / between 100 and 1000 / between 1000 and 10000 / more than 10000 features, respectively).

In addition to conventional feature selection (scenario 1) with max-size constraint b_{MS}, specified in Table 3, we evaluate a block feature selection (scenario 2) for datasets with block-wise feature structure. For block feature selection, up to b_{MS} features should be selected from at most b_{BMS} distinct blocks.[7] Random forests (RF) Breiman (2001), and RENT Jenul (2021) (representing ensemble feature selectors that extend the concepts of decision trees and elastic net regularized models, respectively) are used as state-of-the-art benchmarks for standard feature selection, while Sparse Group Lasso (GL) Ida et al. (2019) is used as the benchmark for block feature selection. To conform with UBayFS, RENT and RF are adjusted to $M = 100$ elementary models, and all models are tuned to select approximately the same number of features, b_{MS}. Since RENT and GL cannot be instructed to select b_{MS} features directly, regularization parameters are determined via bisection, such that the number of selected features is approximately equal to b_{MS}.

The selected features cannot be evaluated directly in real-world datasets due to unknown ground truth on the feature relevance. Therefore, we train predictive models on $T^{(i)}_{\text{train}}$ after feature selection and evaluate the selected features indirectly via the predictive performance on the test instances. To reduce the influence of the predictive model type, we train two distinct classifiers on $T^{(i)}_{\text{train}}$ after feature selection, and report F1 scores for predictions on $T^{(i)}_{\text{test}}$ for both. The choice of baseline classifiers to obtain the prediction comprises:

- generalized linear model: logistic regression (GLM),
- support vector machine (SVM).

3.4.1 Results

Tables 4 and 5 present the results of the experiments on real-world data. Thereby, UBayFS achieves good predictive F1 scores throughout the different datasets, even though no expert knowledge is introduced to ensure a fair comparison. In the block feature selection setups, UBayFS benefits from block constraints and shows more flexibility than Sparse Group Lasso. Altogether, UBayFS can keep up with its competitors in terms of predictive performance in a diverse range of scenarios (low-dimensional and high-dimensional data, as well as unconstrained and constrained setups) while providing higher flexibility to introduce additional information or constraints. Overall, the results reflect that a particular strength of UBayFS lies in delivering a good trade-off between stabilities and predictive performance, compared to competitors such as RF, which deliver high F1 scores, but very low stabilities.

[7] Details on the block structure of the datasets are provided in Appendix B.

Table 4 UBayFS with three distinct elementary feature selectors (M: mRMR, F: Fisher, T: decision tree) is compared to ensemble feature selectors RF and RENT in a standard feature selection scenario. UBayFS with additional (BMS) constraint is compared to Sparse Group Lasso (GL) for block-feature selection on datasets with block structure. Average F1 scores are given for different predictive models (GLM, SVM). The best scores for each dataset and evaluation metric are marked in bold—standard feature selection and block feature selection are assessed separately

Dataset	Standard feature selection					Block feature selection			
	RF	RENT	UBayFS			GL	UBayFS		
			M	F	T		M	F	T
(a) Average F1 score per run (predictor: GLM).									
BCW	0.95	**0.97**	0.96	**0.97**	0.95	**0.96**	**0.96**	**0.96**	**0.96**
HD	0.92	0.88	0.91	0.90	**0.93**	–	–	–	–
MPE	0.86	**0.95**	0.87	0.83	0.83	–	–	–	–
COL	0.85	0.83	0.83	0.78	**0.88**	0.82	0.74	0.77	**0.89**
LSVT	0.70	0.75	0.80	**0.84**	0.68	0.77	0.67	**0.79**	0.59
p53	0.71	0.66	**0.80**	0.78	**0.80**	0.63	0.76	**0.79**	**0.79**
PRO	0.88	**0.89**	0.78	0.85	0.84	–	–	–	–
LEU	0.88	0.93	0.88	0.91	**0.95**	–	–	–	–
LUNG	0.93	**0.97**	0.91	0.90	0.92	–	–	–	–

Dataset	Standard feature selection					Block feature selection			
	RF	RENT	UBayFS			GL	UBayFS		
			M	F	T		M	F	T
(b) Average F1 score per run (predictor: SVM).									
BCW	0.95	**0.97**	0.96	0.96	0.94	**0.97**	0.96	0.96	0.95
HD	0.92	0.88	0.91	0.91	**0.95**	–	–	–	–
MPE	0.87	**0.95**	0.89	0.84	0.84	–	–	–	–
COL	0.86	0.85	0.87	0.83	**0.88**	0.81	0.82	0.79	**0.89**
LSVT	0.75	0.75	0.80	**0.84**	0.71	**0.80**	0.79	0.79	0.57
p53	0.81	**0.82**	0.81	0.80	**0.82**	**0.84**	0.77	0.82	0.80
PRO	**0.91**	0.90	0.87	0.88	0.85	–	–	–	–
LEU	**0.96**	0.94	0.88	0.95	**0.96**	–	–	–	–
LUNG	**0.98**	0.97	**0.98**	0.96	0.94	–	–	–	–

Figures 6 and 7 give additional insights into the performances of the UBayFS variants in the standard feature selection and block feature selection scenario, respectively. Differences between the F1 scores obtained by the different elementary feature selectors underline that UBayFS inherits benefits and drawbacks from its underlying elementary model type—in particular, the decision tree and HSIC achieved top results. Nevertheless, the building of ensembles allows to compensate in parts for mediocre stabilities.

3.4.2 Case study with prior knowledge

Our evaluations underlined the applicability of UBayFS in real-world scenarios. However, due to the absence of prior knowledge, these scenarios covered only parts of the capabilities of

Table 5 Mean stabilities of UBayFS with three distinct elementary feature selectors (M: mRMR, F: Fisher, T: decision tree), compared to ensemble feature selectors RF and RENT in standard feature selection, as well as to GL in block feature selection scenarios. The best scores in each row are marked in bold for each scenario

Dataset	Standard feature selection					Block feature selection			
	RF	RENT	UBayFS			GL	UBayFS		
			M	F	T		M	F	T
BCW	0.73	0.87	0.87	**1.00**	0.61	**0.90**	0.80	0.80	0.80
HD	0.45	0.87	**0.88**	0.65	0.59	–	–	–	–
MPE	0.72	**0.87**	0.92	0.85	0.77	–	–	–	–
COL	0.39	0.67	0.80	0.72	**0.81**	0.56	**0.84**	0.72	0.82
LSVT	0.31	0.59	0.72	**0.79**	0.55	0.73	0.66	**0.88**	0.31
p53	0.11	**0.56**	0.34	0.34	0.36	**0.68**	0.19	0.25	0.31
PRO	0.17	0.53	0.56	**0.61**	0.42	–	–	–	–
LEU	0.07	0.64	0.46	**0.76**	0.53	–	–	–	–
LUNG	0.18	0.78	**0.80**	0.79	0.40	–	–	–	–

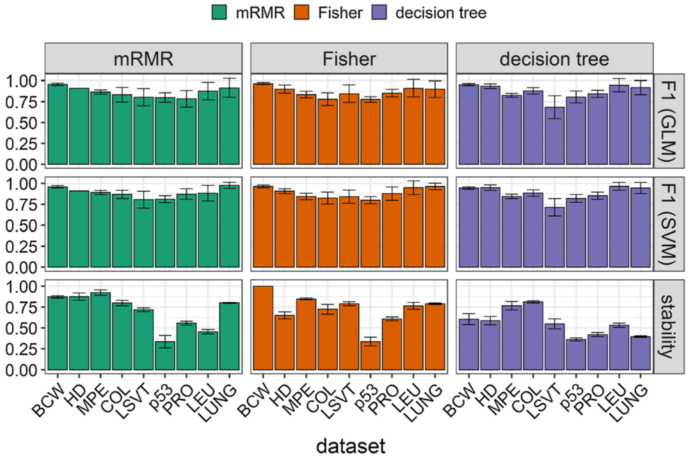


Fig. 6 Performance results of UBayFS feature selection on real-world datasets (MS constraint). F1 scores are determined after training and predicting a classifier (GLM or SVM) after feature selection. Results show mean values over $I = 10$ runs along with standard deviations

the method. To exploit prior knowledge in practice, we revisit the lung cancer genome dataset (LUNG): in the dataset, eight gene expression features were identified as relevant in biological studies by Guan Guan et al. (2009). Thus, we assign higher prior weights α_R to a-priori relevant features, while all other features get assigned the default prior weight $\alpha_{-R} = 0.01$. Our setups include one with "weak" prior ($\alpha_R = 20$), and one with "strong" prior ($\alpha_R = 100$), in addition to the setup without prior, shown in Table 4. The max-size constraint is set to $b_{\text{MS}} = 100$.

As summarized in Table 6, incorporating prior knowledge leads to an improvement of UBayFS results in most cases. Thus, the absolute performance lies in a similar top range as those reported in previous work by Brahim and Limam (2014), who evaluated averaged accuracies in a comparable setup on the same dataset (> 0.99 avg. accuracy). However, the comparability of accuracies is limited due to the unbalanced nature of the dataset. Between the UBayFS setups, results with weak prior are similar to those from no-prior results in the case of stable elementary feature selectors (mRMR and Fisher). In contrast, weak prior results resemble the strong prior in the case of a

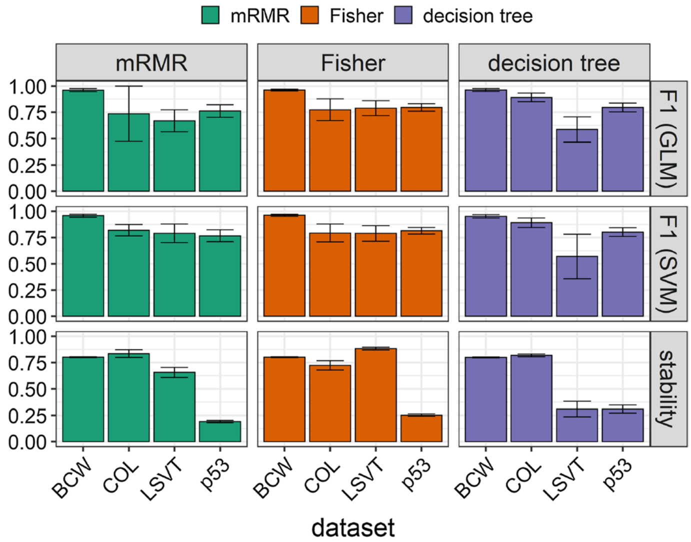


Fig. 7 Performance results of UBayFS block feature selection on real-world datasets (MS and BMS constraints). F1 scores are determined after training and predicting a classifier (GLM or SVM) after feature selection. Results show mean values over $I = 10$ runs along with standard deviations

Table 6 Average performance scores delivered by UBayFS on the LUNG dataset with and without prior knowledge

Setup	GLM			SVM			Stability		
	M	F	T	M	F	T	M	F	T
Without prior	0.91	0.90	0.92	0.98	0.96	0.94	0.80	0.79	0.40
With prior ($\alpha_{\text{imp}} = 20$)	0.91	0.90	0.91	0.98	0.96	0.96	0.80	0.79	0.45
With prior ($\alpha_{\text{imp}} = 100$)	0.91	0.94	0.91	0.98	0.96	0.96	0.82	0.81	0.45

non-stable elementary feature selector (decision tree). Thus, a weak prior has a higher impact on the final results if the elementary models are more diverse.

3.4.3 Runtime

Runtimes of all methods and datasets are provided in Table 7. Given a fixed set of model parameters, it becomes obvious that the major factor influencing the runtime of UBayFS is the number of features (columns) rather than the number of samples (rows). UBayFS runtimes refer to the MS setup—however, experiments showed only minor differences to the runtimes in the block feature selection setup. While RF and GL are more tractable in high-dimensional datasets, RENT seems to suffer from data dimensionality to a more considerable extent.

Across larger datasets, the main influencing factor on the runtime is the number and type of elementary models. For example, on the LUNG dataset (> 12000 features), the training procedure of 100 mRMR models as elementary models comprised 40 minutes

Table 7 Average runtime per run [s]

Dataset	RF	RENT	GL	UBayFS		
				M	F	T
BCW	6.7	3.4	10.9	6.2	2.2	4.3
HD	6.3	3.2	–	1.8	1.6	2.1
MPE	9.4	24.3	–	12.3	5.3	9.6
COL	6.1	3.8	4.6	3.7	2.9	3.6
LSVT	10.0	77.9	9.0	6.4	6.7	9.6
p53	80.2	2712.3	112.7	366.8	125.6	440.3
PRO	29.8	1217.2	–	370.9	232.6	708.0
LEU	41.5	980.9	–	263.0	160.8	549.5
LUNG	116.8	2834.1	–	1930.3	535.1	1885.0

(88% of UBayFS runtime), while optimization using the Genetic Algorithm comprised 5 minutes (11% of UBayFS runtime).[8]

4 Discussion and conclusion

The presented Bayesian feature selector UBayFS has its strength in combining information from a data-driven ensemble model with expert prior knowledge targeted at life science applications. The generic framework is flexible in the choice of the elementary feature selector type, allowing a broad scope of applications scenarios by deploying adequate elementary feature selectors, such as those suggested by Sechidis and Brown (2018) for semi-supervised or Elghazel and Aussem (2015) for unsupervised problems. An extension of the presented experiments to multiple classes or multi-label classification problems (one object is not uniquely assigned to one class) is straightforward as well if the elementary feature selector is capable of tackling such datasets, such as Petković et al. (2020).

In general, the choice of the elementary feature selector is a central step when deploying the concept in practice—in particular, the size and structure of a dataset need to be taken into account. This work presented a broad range of elementary models to provide user guidance in practical setups. The option to build ensembles combining different model types, as discussed by Seijo-Pardo et al. (2017), turned out to deteriorate the stability of ensemble feature selectors and hence, is not considered in this study.

UBayFS presents two ways to account for feature dependencies: a generalized prior model as well as a decorrelation constraint. The latter effectively restricts the results, such that a simultaneous selection of highly correlated features is penalized. The generalizations of the prior model correct the estimated feature importances by the dependencies—in a low-dimensional scenario, the hyperdirichlet variant is the most accurate choice. However, this variant becomes intractable, if the dimensionality exceeds a few hundred features and requires simulation to determine the expected value in almost any case, preventing from analytically exact solutions. Since our experiments depicted that feature importances obtained from each of the three prior setup types are numerically similar, a conventional Dirichlet setup seems to deliver a sufficiently accurate approximation

[8] Runtime information refers to the current version of the implementation and is subject to further code optimization.

for high-dimensional datasets. This observation is also supported by the fact that many elementary feature selectors, such as mRMR or HSIC, can account for between-feature correlations, thus reducing the need to consider correlations in the meta-model.

Prior information from experts is introduced via prior feature weights and linking constraints describing between-feature dependencies, represented in a system of side constraints. Via a relaxation parameter, the inadmissibility is transferred into a soft constraint, favoring solutions that fulfill the constraints and penalizing violations. Introducing user knowledge directly into the feature selection process opens new opportunities for data analysis in life science applications. Still, such methodology bears the potential of intentional or unintentional misuse: as demonstrated in the experiment, the integration of unreliable or incorrect user knowledge may distort predictive results. Users have to be aware that UBayFS may contain subjective inputs and thus, take precautions to ensure that prior information is sufficiently verified, e.g., by published research in the field.

Based on the results from extensive experimental evaluations on multiple open-source datasets, a clear benefit of the proposed feature selector lies in the balance between predictive performance and stability. Particularly in life sciences, where few instances are available in high-dimensional datasets, user-guided feature selection is an opportunity to guide models to achieve otherwise intractable results. UBayFS delivers more flexibility to integrate domain knowledge than established state-of-the-art approaches. A practical limitation of UBayFS is that the runtime is arguably slower than simpler feature selectors, which becomes an obstacle in very high-dimensional datasets. The use of highly optimized algorithms like the Genetic Algorithm, along with an initialization using the suggested Alg. 1 mitigates this issue. However, it cannot compensate for the computational burden of training multiple elementary models.

Appendix A theory

A.1 Convergence of inadmissibility function

The point-wise convergence $\kappa_{k,\rho} \underset{\rho\to\infty}{\longrightarrow} \kappa_k$ holds for arbitrary $\boldsymbol{A} \in \mathbb{R}^{K\times N}$ and $\boldsymbol{b} \in \mathbb{R}^K$ on the domain $\mathcal{D} = \{0,1\}^N$.

Proof From the definition of $\kappa_{k,\rho}(\boldsymbol{\delta})$, the claim is trivially fulfilled for

$$\boldsymbol{\delta} \in \left\{\boldsymbol{\delta}' \in \{0,1\}^N : \left(\boldsymbol{a}^{(k)}\right)^T \boldsymbol{\delta}' - b^{(k)} \leq 0\right\}.$$

In the opposite case, we define λ_k as $\lambda_k = \left(\boldsymbol{a}^{(k)}\right)^T \boldsymbol{\delta} - b^{(k)} > 0$. It holds that

$$\begin{aligned}\kappa_{k,\rho}(\boldsymbol{\delta}) &= \frac{1-\xi_{k,\rho}}{1+\xi_{k,\rho}} \\ &= \frac{1-\exp\left(-\rho\lambda_k\right)}{1+\exp\left(-\rho\lambda_k\right)}.\end{aligned}$$

Since $\lambda_k > 0$, we obtain $-\rho\lambda_k \underset{\rho\to\infty}{\longrightarrow} -\infty$, and thus $\xi_{k,\rho} = \exp\left(-\rho\lambda_k\right) \underset{\rho\to\infty}{\longrightarrow} 0$. It follows that $\kappa_{k,\rho}(\boldsymbol{\delta}) \underset{\rho\to\infty}{\longrightarrow} 1$. Hence, we have shown a point-wise convergence of

$$\kappa_{k,\rho}(\boldsymbol{\delta}) \xrightarrow[\rho\to\infty]{} \begin{cases} 1 & \text{if } \lambda_k \leq 0 \\ 0 & \text{if } \lambda_k > 0, \end{cases}$$

which equals to κ_k on the domain $\mathcal{D}$.

A.2 Generalizations of the Dirichlet distribution

In Sect. 2.2, we discuss the possibility to replace the Dirichlet distribution with one out of two generalized variants:

- the generalized Dirichlet distribution, and
- the hyperdirichlet distribution.

Both variants preserve the conjugate prior property with respect to the multinomial likelihood, as explained by the corresponding authors who had introduced these generalizations. In this part, we provide a short overview on the probability density functions, parameters and (posterior) expected values of these distributions, as these quantities are relevant for the UBayFS setup.

The standard Dirichlet distribution, see e.g. DeGroot (2005), is commonly defined by the probability density function

$$f_{\text{Dir}}(\boldsymbol{\theta};\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{n=1}^{N} \theta_n^{\alpha_n - 1}, \tag{16}$$

where $B(\boldsymbol{\alpha}) = \frac{\prod_{n=1}^{N} \Gamma(\alpha_n)}{\Gamma\left(\sum_{n=1}^{N} \alpha_n\right)}$ denotes the multivariate beta function. Due to the simple parameter update in the inference step, we obtain the posterior expected value

$$\mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{y}}[\boldsymbol{\theta}] = \frac{1}{\|\boldsymbol{\alpha}^{\circ}\|_1} \boldsymbol{\alpha}^{\circ},$$

where $\boldsymbol{\alpha}^{\circ} = \boldsymbol{\alpha} + \boldsymbol{y}$.

In essence, the generalized Dirichlet distribution by Wong (1998) adds an additional parameter vector $\boldsymbol{\beta} \in \mathbb{R}^{N-1}$ to the parameter vector $\boldsymbol{\alpha}$ from the Dirichlet distribution and is defined via the probability density

$$f_{\text{gDir}}(\boldsymbol{\theta}') = \prod_{n=1}^{N-1} \frac{1}{B(\alpha_n, \beta_n)} \left(\theta'_n\right)^{\alpha_n - 1} \left(1 - \sum_{i=1}^{n} \theta'_i\right)^{\gamma_n}, \tag{17}$$

where $B(\alpha_n, \beta_n) = \frac{\Gamma(\alpha_n)\Gamma(\beta_n)}{\Gamma(\alpha_n + \beta_n)}$, $\gamma_n = \beta_n - \alpha_{n+1} - \beta_{n+1}$ for $n \in [N-2]$, and $\gamma_{N-1} = \beta_{N-1} - 1$. In contrast to the standard Dirichlet setting, the distribution is defined on the $N-1$-dimensional space, relaxing the side constraint $\|\boldsymbol{\theta}\|_1 = 1$ to $\|\boldsymbol{\theta}'\|_1 \leq 1$, $\boldsymbol{\theta}' \in \mathbb{R}^{N-1}$ — both are equivalent, if $\theta_n = \theta'_n$ for $n \in [N-1]$, and $\theta_N = 1 - \sum_{n=1}^{N-1} \theta'_n$. The posterior expected value for the generalized Dirichlet distribution is given in closed-form by

$$\left(\mathbb{E}_{\theta}[\theta]\right)_n = \begin{cases} \frac{\alpha_n + y_n}{\alpha_n + \beta_n + v_n} & n = 1 \\ \frac{\alpha_n + y_n}{\alpha_n + \beta_n + v_n} \prod_{i=1}^{n-1} \frac{\beta_i + n_{i+1}}{\alpha_i + \beta_i + n_i} & n = 2, \dots, N-1 \\ \prod_{i=1}^{N-1} \frac{\beta_i + n_{i+1}}{\alpha_i + \beta_i + v_i} & n = N, \end{cases}$$

where $v_n = \sum_{i=n}^{N} y_i$, see Wong (1998).

An even more general version is the hyperdirichlet distribution by Hankin (2010), who characterizes the distribution by the probability density function

$$f_{\mathrm{hDir}}(\theta) \propto \left(\prod_{n=1}^{N} \theta_n\right)^{-1} \prod_{G \in \mathcal{P}([N])} \left(\sum_{i \in G} \theta_i\right)^{\mathcal{F}(G)}, \tag{18}$$

where $\mathcal{P}(.)$ denotes the power set and $\mathcal{F}(G)$ denotes the parameter for each possible subset of $[N]$. Since the closed-form expression of the expected value involves the normalization constant, which is intractable in practical high-dimensional setups, we deploy the Metropolis-Hastings (MH) algorithm implemented in Hankin (2017) to sample from the hyperdirichlet distribution and determine the expected value empirically from the sample mean.

Table 8 Dataset sources

Name	Link
HD	https://archive.ics.uci.edu/ml/datasets/heart+disease
BCW	https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic)
MPE	https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression
COL	https://github.com/cran/gglasso
LVST	https://archive.ics.uci.edu/ml/datasets/LSVT+Voice+Rehabilitation
p53	https://archive.ics.uci.edu/ml/datasets/p53+Mutants
LEU	see R package *spls* Chung et al. (2019)
PRO	see R package *propOverlap* Mahmoud (2014)
LUNG	https://leo.ugr.es/elvira/DBCRepository/LungCancer/LungCancer-Harvard2.html

Table 9 Block indices for datasets with block structure. Feature names indicate the column name patterns, which is used for defining blocks

Dataset	Block no	Indices	Feature names
BCW	1	1–10	Mean
	2	11–20	Error
	3	21–30	Worst
COL	1	1–5	
	2	6–10	
	⋮	⋮	
	20	96–100	
LSVT	1	97–124	Delta
	2	160–179, 200–219, 251–270, 291–310	Det
	3	129–139, 220–230	E
	4	140–159, 180–199, 231–250, 271–290	Entropy
	5	62–67	GNE
	6	52–53	HNR
	7	77–82	IMF
	8	1–30	Jitter
	9	84–96	MFCC
	10	54–55	NHR
	11	56–58	OQ
	12	31–51	Shimmer
	13	68–76	VFER
	14	59–61, 83, 125–128	Other
p53	1	14826	
	2	4827–5408	

Appendix B Experimental datasets

All real-world datasets are publicly available (status: 12/2021), see Table 8. For datasets with block structure (BCW, COL, LSVT and p53), block indices are given in Table 9.

Acknowledgements In special we thank Kristian Hovde Liland (NMBU), Cecilia Marie Futsaether (NMBU) and Eirik Malinen (University of Oslo) for their constructive discussions and valuable input for this work, as well as Michael P. Alley (Penn State University) for proof-reading the paper.

Author contributions AJ, SS and JP developed the theory part of this work. AJ, SS and OT planned and conducted the associated experiments. AJ and SS wrote the manuscript. All authors contributed to the proof-reading and editing of the paper.

Funding Open access funding provided by Norwegian University of Life Sciences. This work was partly funded by the Norwegian Cancer Society (Grant no. 182672-2016).

Availability of data and materials All real-world datasets are publicly available, see Appendix B.

Code availability Code is made publicly available on GitHub, see https://github.com/annajenul/UBayFS.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Consent to participate All authors consented to the submission of the manuscript.

Consent for publication All real-world datasets are obtained from publicly available platforms under open licenses. All figures in this manuscript are created by the authors.

Ethics approval Not applicable.

References

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

Bose, S., Das, C., Banerjee, A., Ghosh, K., Chattopadhyay, M., Chattopadhyay, S., & Barik, A. (2021). An ensemble machine learning model based on multiple filtering and supervised attribute clustering algorithm for classifying cancer samples. *Peer J Computer Science, 7,* e671.

Brahim, A. B., & Limam, M. (2014). New prior knowledge based extensions for stable feature selection. In *2014 6th international conference of soft computing and pattern recognition (SoCPaR)* (pp. 306–311).

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. Taylor & Francis.

Cheng, T.-H., Wei, C.-P. & Tseng, V.S. (2006). Feature selection for medical data mining: Comparisons of expert judgment and automatic approaches. In *19th IEEE symposium on computer-based medical systems (CBMS'06)* (p. 165-170).

Chung, D., Chun, H. & Keles, S. (2019). spls: sparse partial least squares (SPLS) regression and classification [Computer software manual]. R package version 2.2-3.

Dalton, L. A. (2013). Optimal Bayesian feature selection. In *2013 IEEE global conference on signal and information processing* (p. 65-68).

Danziger, S., Swamidass, S., Zeng, J., Dearth, L., Lu, Q., Chen, J., et al. (2006). Functional census of mutation sequence spaces: The example of p53 cancer rescue mutants. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 3*(2), 114–124.

DeGroot, M. H. (2005). *Optimal statistical decisions*. Wiley.

Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J.-J., Sandhu, S., et al. (1989). International application of a new probability algorithm for the diagnosis of coronary artery disease. *American Journal of Cardiology, 64*(5), 304–310.

Ding, C., & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology, 3*(02), 185–205.

Elghazel, H., & Aussem, A. (2015). Unsupervised feature selection with ensemble learning. *Machine Learning, 98*(1), 157–180.

Givens, G. H., & Hoeting, J. A. (2012). *Computational statistics* (Vol. 703). John Wiley & Sons.

Goldstein, O., Kachuee, M., Karkkainen, K., & Sarrafzadeh, M. (2020). Target-focused feature selection using uncertainty measurements in healthcare data. *ACM Transactions on Computing for Healthcare, 1*(3), 1–17.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., et al. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science, 286*(5439), 531–537.

Gordon, G. J., Jensen, R. V., Hsiao, L.-L., Gullans, S. R., Blumenstock, J. E., Ramaswamy, S., et al. (2002). Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Research, 62*(17), 4963–4967.

Guan, P., Huang, D., He, M., & Zhou, B. (2009). Lung cancer gene expression database analysis incorporating prior knowledge with support vector machine-based classification method. *Journal of Experimental & Clinical Cancer Research., 28*(1), 1–7.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning, 46*(1), 389–422.

Hankin, R. K. S. (2010). A generalization of the Dirichlet distribution. *Journal of Statistical Software, 33*(11), 1–18.

Hankin, R.K.S. (2017). Partial rank data with the hyper2 package: Likelihood functions for generalized Bradley-Terry models. *The R Journal*, 9.

Higuera, C., Gardiner, K. J., & Cios, K. J. (2015). Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. *PloS one, 10*(6), e0129126.

Ida, Y., Fujiwara, Y. & Kashima, H. (2019). Fast sparse group lasso. *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc.

Jenul, A., Schrunner, S., Liland, K.H., Indahl, U.G., Futsæther, C.M. & Tomic, O. (2021). RENT—repeated elastic net technique for feature selection. *IEEE Access*, 9, 152333-152346.

Liu, M., & Zhang, D. (2015). Pairwise constraint-guided sparse learning for feature selection. *IEEE Transactions on Cybernetics, 46*(1), 298–310.

Lyle, C., Schut, L., Ru, R., Gal, Y., & van der Wilk, M. (2020). A Bayesian perspective on training speed and model selection. *Advances in neural information processing systems, 33,* 10396–10408.

Mahmoud, O., Harrison, A., Perperoglou, A., Gul, A., Khan, Z. & Lausen, B. (2014). propOverlap: feature (gene) selection based on the proportional overlapping scores [Computer software manual]. R package version 1.0

Nakajima, S., Sato, I., Sugiyama, M., Watanabe, K. & Kobayashi, H. (2014). Analysis of variational Bayesian latent Dirichlet allocation: Weaker sparsity than MAP. Advances in neural information processing systems (Vol. 27). Curran Associates, Inc.

Nogueira, S., Sechidis, K., & Brown, G. (2018). On the stability of feature selection algorithms. *Journal of Machine Learning Research, 18*(174), 1–54.

O'Hara, R. B., & Sillanpää, M. J. (2009). A review of Bayesian variable selection methods: What, how and which. *Bayesian Analysis, 4*(1), 85–117.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research, 12,* 2825–2830.

Petković, M., Džeroski, S., & Kocev, D. (2020). Multi-label feature ranking with ensemble methods. *Machine Learning, 109*(11), 2141–2159.

Pozzoli, S., Soliman, A., Bahri, L., Branca, R. M., Girdzijauskas, S., & Brambilla, M. (2020). Domain expertise-agnostic feature selection for the analysis of breast cancer data. *Artificial Intelligence in Medicine, 108,* 101928.

R Core Team. (2020). *R: A language and environment for statistical computing [Computer software manual].* Austria.

Saon, G., & Padmanabhan, M. (2001). Minimum Bayes error feature selection for continuous speech recognition. *Advances in Neural Information Processing Systems, 13,* 800–806.

Scrucca, L. (2013). GA: A package for genetic algorithms in R. *Journal of Statistical Software, 53*(4), 1–37.

Sechidis, K., & Brown, G. (2018). Simple strategies for semi-supervised feature selection. *Machine Learning, 107*(2), 357–395.

Seijo-Pardo, B., Porto-Díaz, I., Bolón-Canedo, V., & Alonso-Betanzos, A. (2017). Ensemble feature selection: Homogeneous and heterogeneous approaches. *Knowledge-Based Systems, 118,* 124–139.

Singh, D., Febbo, P. G., Ross, K., Jackson, D. G., Manola, J., Ladd, C., et al. (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell, 1*(2), 203–209.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological), 73*(3), 273–282.

Tsanas, A., Little, M. A., Fox, C., & Ramig, L. O. (2013). Objective automatic assessment of rehabilitative speech treatment in Parkinson's disease. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 22*(1), 181–190.

Wolberg, W. H., & Mangasarian, O. L. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences, 87*(23), 9193–9196.

Wong, T.-T. (1998). Generalized Dirichlet distribution in Bayesian analysis. *Applied Mathematics and Computation, 97*(2), 165–181.

Yamada, M., Jitkrittum, W., Sigal, L., Xing, E. P., & Sugiyama, M. (2014). High-dimensional feature selection by feature-wise kernelized lasso. *Neural Computation, 26*(1), 185–207.

Yang, Y., & Zou, H. (2015). A fast unified algorithm for solving group-lasso penalize learning problems. *Statistics and Computing, 25*(6), 1129–1141.

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)., 68*(1), 49–67.

Zhao, Z., Wang, L., Liu, H. (2010). Efficient spectral feature selection with minimum redundancy. *In Proceedings of the AAAI conference on artificial intelligence* (Vol. 24, pp. 673–678).