Nonparametric Feature Selection by Random Forests and Deep Neural Networks

Xiaojun Mao
School of Mathematical Sciences,
Shanghai Jiao Tong University, Shanghai, China

Liuhua Peng
School of Mathematics and Statistics,
The University of Melbourne, Victoria, Australia
and
Zhonglei Wang*
Wang Yanan Institute for Studies in Economics,
Xiamen University, Xiamen, Fujian, China

Abstract

Random forests are a widely used machine learning algorithm, but their computational efficiency is undermined when applied to large-scale datasets with numerous instances and useless features. Herein, we propose a nonparametric feature selection algorithm that incorporates random forests and deep neural networks, and its theoretical properties are also investigated under regularity conditions. Using different synthetic models and a real-world example, we demonstrate the advantage of the proposed algorithm over other alternatives in terms of identifying useful features, avoiding useless ones, and the computation efficiency. Although the algorithm is proposed using standard random forests, it can be widely adapted to other machine learning algorithms, as long as features can be sorted accordingly.

Key words: Feature importance, Maximum mean discrepancy, Reproducing kernel Hilbert space.

---

*Correspondence to wangzl@xmu.edu.cn.

1 Introduction

Random forests (Breiman; 2001, RFs) are a widely used machine learning algorithm (Caruana and Niculescu-Mizil; 2006; Criminisi et al.; 2012; Fernández-Delgado et al.; 2014). However, their computational efficiency is compromised when they are applied to large-scale datasets with numerous useless features.

Since the landmark of Breiman (2001), research on RFs has been active in different scientific fields. Xu and Jelinek (2005) used RFs for structured language learning, and they showed that their method outperformed its competitors in terms of perplexity and error rates. Based on a synthetic dataset generated from a certain reference distribution, Shi and Horvath (2006) applied RFs to obtain dissimilarities among the original unlabeled data. Payet and Todorovic (2010) proposed an RFs algorithm to directly estimate the ratio of the proposal and posterior distributions nonparametrically for the Metropolis-Hastings algorithm, and the corresponding theoretical properties were also investigated. Using a Mondrian process, Lakshminarayanan et al. (2014) proposed a computationally efficient RFs algorithm for online learning, but splits were made independent of the response of interest. Inspired by the local polynomial regression, Li and Martin (2017) proposed a robust RFs algorithm incorporating different loss functions and showed that their method generalized the standard and quantile RFs (Meinshausen; 2006). Haghiri et al. (2018) proposed a comparison-based RFs algorithm for the case when the sample was not representative and when it was difficult to measure the distance between instances. Siblini et al. (2018) proposed an algorithm for extreme multi-label learning using a tree-based method and demonstrated that the computation was more efficient than other competitors in parallel. Scornet et al. (2015), Mentch and Hooker (2016), and Wager and Athey (2018) investigated theoretical properties of RFs algorithms. Refer to Criminisi et al. (2012) and Goel et al. (2017) for a more comprehensive review of RFs.

Albeit it is common to have numerous features in practice, only a limited portion contributes to the response of interest (Fan and Lv; 2008). Because existing RFs algorithms do not identify useful features before growing trees, the corresponding computation efficiency is undermined, especially when most features are useless. For example, it is well

known that local polynomial regression suffers from the curse of dimensionality; thus, the estimation efficiency of Li and Martin (2017) is questionable. In addition, the online RFs (Lakshminarayanan et al.; 2014) may also lead to inefficient estimation when the number of useless features is large, since splits are made based on features regardless of the response of interest. Thus, feature selection is essential for a high dimensional dataset.

Even though RFs have been actively investigated under different scientific fields, feature selection by RFs does not receive so much attention. There are two main approaches for feature selection by RFs. One approach is permutation-based, using feature importance (Breiman; 2001). Strobl et al. (2008) proposed a permutation algorithm to compute conditional feature importance, but they did not provide a general guidance for feature selection. Kursa and Rudnicki (2010) used a new set of shadow features to debias the feature importance, and a feature selection procedure was also proposed based on a "Z score"; also see Sandri and Zuccolotto (2008) for a similar approach. Altmann et al. (2010) proposed to permute the response vector to get the "null importance", and it was used for feature selection heuristically. Genuer et al. (2010) proposed a two-step procedure for feature section based on feature importance, and their method worked well for highly correlated features. However, the theoretical properties of existing permutation-based methods have not been rigorously investigated. The other approach is based on the minimum depth proposed by Ishwaran et al. (2010), and it has been widely applied in survival analysis (Ishwaran et al.; 2008; Twyman-Saint Victor et al.; 2015; Benci et al.; 2016). The corresponding theoretical properties were investigated by Ishwaran et al. (2010) under strong conditions. For example, the split for a node should always be the median of the values, but such an assumption may undermine the estimation efficiency.

Herein, we propose a nonparametric feature selection algorithm that incorporates RFs and deep neural networks (NFSRD). Specifically, we adopt nonparametric two-sample tests using deep neural networks (Liu et al.; 2020) to select *useful* features, and the corresponding theoretical properties are investigated under regularity conditions. To improve the computational efficiency of the NFSRD, subsampling is adopted. Experiments reveal that the NFSRD outperforms its alternatives in terms of detecting useful features, avoiding useless

ones, and the computation efficiency. Another advantage of subsampling is that it saves the computer memory while retaining the desired accuracy for feature selection; refer to Section 4 for details.

The NFSRD differs from existing works in the following aspects. First, we propose the use of nonparametric two-sample tests to select useful features incorporating RFs and deep neural networks, and the corresponding theoretical properties can be rigorously established under regularity conditions. In addition, subsampling is adopted for feature selection; thus, the NFSRD is more computationally efficient than existing RFs-based feature selection algorithms. Moreover, we do not make any strong assumption for splits, as in Ishwaran et al. (2010). Rather, the only crucial assumption we make is that a limited portion of the features contributes to the response of interest, and such an assumption is also widely adopted for feature selection (Fan and Lv; 2008). Besides, the NFSRD adopts a forward feature selection procedure based on the features sorted by their importance, so it can be widely adapted to existing algorithms mentioned in the preceding paragraph and other machine learning algorithms (Chen and Guestrin; 2016; Wager and Athey; 2018), as long as the features can be sorted accordingly.

The remainder of this paper is organized as follows. The model setup is introduced in Section 2. The detailed algorithm for the NFSRD is presented in Section 3, and the corresponding theoretical properties are also investigated. Simulation studies are presented in Section 4. Section 5 describes an application of the NFSRD to identify useful features based on a superconductivity dataset. The conclusions are provided in Section 6.

2　Model Setup

Let $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip}) \in \mathcal{X}$ be a p-dimensional feature vector and $y_i \in \mathbb{R}$ be the corresponding response of interest, where $i \in \mathcal{I}_n$, $\mathcal{I}_n = \{1, \ldots, n\}$ is the sample index set of size n, and $\mathcal{X} \subset \mathbb{R}^p$. Consider the following regression model:

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i \quad (i \in \mathcal{I}_n), \tag{1}$$

where $f(\boldsymbol{x}_i)$ is a smooth function involving $K_0 \ll p$ useful features (Fan and Lv; 2008), and ϵ_i is white noise. We are interested in identifying the K_0 useful features based on the sample $\mathcal{D}_n = \{(\boldsymbol{x}_i, y_i) : i \in \mathcal{I}_n\}$. Without loss of generality, lower cases denote observed data, and upper cases denote the associated random variables. The vectors are column-wise, unless explicitly explained otherwise.

Before presenting the NFSRD algorithm, we briefly introduce a bagged-tree learner and RFs. A bagged-tree learner is

$$\widehat{\theta}_{TR}(\boldsymbol{x}) = \mathbb{E}_*\{t(\boldsymbol{x}; \boldsymbol{Z}_1^*, \ldots, \boldsymbol{Z}_n^*)\},$$

where $\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i)$, and $\{\boldsymbol{Z}_i^* : i \in \mathcal{I}_n\}$ is a random sample of size n generated by an empirical distribution $\mathbb{P}^*(\boldsymbol{z}) = n^{-1} \sum_{i \in \mathcal{I}_n} \mathbb{I}(\boldsymbol{z} = \boldsymbol{z}_i)$, $\mathbb{I}(\boldsymbol{z} = \boldsymbol{z}_i) = 1$ if $\boldsymbol{z} = \boldsymbol{z}_i$ and 0 otherwise, and $\mathbb{E}_*(\cdot)$ is the expectation with respect to $\mathbb{P}^*(\boldsymbol{z})$. For ease of notation, the sample size n is omitted for $\widehat{\theta}_{TR}(\boldsymbol{x})$ and other statistics.

Notably, RFs extend bagged-tree learners by allowing additional randomness within trees to reduce the correlation among them. Specifically, an RFs learner is

$$\widehat{\theta}_{RF}(\boldsymbol{x}) = \mathbb{E}_*\{t(\boldsymbol{x}; \xi, \boldsymbol{Z}_1^*, \ldots, \boldsymbol{Z}_n^*)\}, \tag{2}$$

where $\xi \sim \Xi$, and Ξ is a pre-specified distribution; see Breiman (2001) for details. A popular choice for Ξ is the random selection of candidate features for the split of each node (Hastie et al.; 2009, §15.2). The RFs learner $\widehat{\theta}_{RF}(\boldsymbol{x})$ is a bagged-tree learner $\widehat{\theta}_{TR}(\boldsymbol{x})$ if Ξ is omitted. Bootstrapping is used to approximate $\widehat{\theta}_{RF}(\boldsymbol{x})$ in (2) by

$$\widehat{\theta}_{RF}^{(B)}(\boldsymbol{x}) = \frac{1}{B} \sum_{b=1}^{B} t(\boldsymbol{x}; \xi_b, \boldsymbol{z}_{b1}^*, \ldots, \boldsymbol{z}_{bn}^*), \tag{3}$$

where B is the number of bootstraps, $\xi_b \sim \Xi$, and $\boldsymbol{z}_{bi}^*$ is a "realization" of $\boldsymbol{Z}_i^*$ for $b = 1, \ldots, B$ and $i \in \mathcal{I}_n$. In practice, a large B is suggested to make the approximation error between (2) and (3) negligible .

3 NFSRD

The assumption $K_0 \ll p$ validates the feature selection (Guyon and Elisseeff; 2003). The NFSRD consists of two steps. The first step corresponds to obtaining bias-corrected feature

importance (BCFI) by shadow features, and the second step is a forward feature selection based on the *ordered* features by BCFI. Two-sample tests using maximum mean discrepancy and deep neural networks (MMD-D) are conducted sequentially for feature selection in the second step. To improve the computational efficiency, subsampling is applied in both steps.

Before diving into details, we briefly discuss the intuition for the NFSRD. The proposed method is a forward-stepwise selection algorithm. To avoid including useless features as much as possible, we first order the features based on their "importance" for estimating the response of interest. Due to their flexibility for nonparametric modeling, RFs are implemented to order the features by a subset of the instances. Based on the ordered features, forward-stepwise selection is conducted by sequential hypothesis tests. After fitting a full model and a reduced model only retaining the first several important features nonparametrically, the corresponding null hypothesis is that the distributions of the residuals from a full model and a reduced model are identical. If the null hypothesis holds for a certain reduced model, then we treat the involved features as useful and the remaining ones as useless. We still use RFs to train the full and reduced models, and other nonparametric algorithms can be implemented as long as certain consistency results hold; see Supplementary Material and Theorem 1 of Mentch and Hooker (2016) for details.

3.1   Bias-Corrected Feature Importance

Feature importance is a rudimentary indicator of the usefulness of features, and it serves as a building block of the NFSRD.

Existing algorithms calculate the feature importance using the entire sample $\mathcal{D}_n$. To achieve better computational efficiency, we propose using subsampling. That is, the feature importance is obtained based on a subsample $\{\boldsymbol{z}_i : i \in \mathcal{A}_0\}$, where $\mathcal{A}_0$ is a subset of $\mathcal{I}_n$, and its size is $m_0 < n$. The numerical results reveal that m_0 should be large to guarantee good performance, and we suggest $m_0 \geqslant 400$ for practical guidance; see Section 4 for details. For example, we can evaluate the feature importance by

$$F_k = \frac{1}{B} \sum_{t_b} \sum_{j:s_{bj}=k} \Delta(s_{bj}, t_b) \quad (k = 1, \ldots, p),$$ (4)

where the first summation is with respect to the B trees, the second summation is with respect the splits made for the bth tree, s_{bj} is the feature used to split the jth node of t_b, $\Delta(s_{bj}, t_b) = w_{bj} V_{bj} - w_{bj}^{(l)} V_{bj}^{(l)} - w_{bj}^{(r)} V_{bj}^{(r)}$ is the weighted decrease in variance, $w_{bj} = n_{bj}/m_0$, n_{bj} is the number of instances in the jth node of t_b, $w_{bj}^{(l)}$ and $w_{bj}^{(r)}$ are the corresponding proportions on the left and right subnodes after splitting the jth node by s_{bj}, V_{bj} is the sample variance of the response of interest in the jth node, and $V_{bj}^{(l)}$ and $V_{bj}^{(r)}$ are variances of the two subnodes. See Breiman (2001) and Sandri and Zuccolotto (2008) for details.

However, the feature importance in (4) is unfairly biased toward those with numerous distinct values (White and Liu; 1994; Louppe et al.; 2013). To debias, we incorporate shadow features (Sandri and Zuccolotto; 2008; Kursa and Rudnicki; 2010). Specifically, for $i \in \mathcal{A}_0$, a shadow feature $\boldsymbol{x}_i^\dagger$ is randomly selected from $\{\boldsymbol{x}_i : i \in \mathcal{A}_0\}$ without replacement. Thereafter, an RF is trained based on the extended data $\{\boldsymbol{z}_i^\dagger : i \in \mathcal{A}_0\}$ with $\boldsymbol{z}_i^\dagger = (\boldsymbol{x}_i, \boldsymbol{x}_i^\dagger, y_i)$; see Sandri and Zuccolotto (2008) for details. The entire procedure is repeated R times, and the BCFI of X_k is calculated as

$$I_k = \frac{1}{R} \sum_{r=1}^{R} (F_{rk} - F_{rk^\dagger}) \quad (k = 1, \ldots, p), \tag{5}$$

where F_{rk} and $F_{rk^\dagger}$ correspond to X_k and $X_k^\dagger$, respectively, and $X_k^\dagger$ is the corresponding shadow feature for X_k for the rth repetition; Sandri and Zuccolotto (2008) suggest that $R = 100$ in practice. Algorithm 1 shows the algorithm for BCFI.

Remark 1.

We should be cautious about algorithms using feature importance as the *only* criterion for feature selection, especially as no rigorous theoretical properties have been investigated. Thus, in the next subsection, we propose the use of nonparametric two-sample tests for feature selection by ordered features using BCFI.

Remark 2. *Other than the shadow features, another way to debias the feature importance is to use the out of bag sample (OOB). Li et al. (2019) argued that the OOB-based feature importance outperforms the traditional one in terms of AUC scores. In addition, we can also consider the minimum depth (Ishwaran et al.; 2010) as an importance metric; see Section 4 for details. Thus, those two metrics can be applied to obtain the feature importance instead.*

Algorithm 1 Bias-Corrected Feature Importance (BCFI)

Input: $\mathcal{D}_n$, m_0, and R.

Select $\mathcal{A}_0$ randomly from $\mathcal{I}_n$.

for $r = 1, \ldots, R$ **do**

    for $i \in \mathcal{A}_0$ **do**

        Generate $\boldsymbol{x}_i^\dagger$ from $\{\boldsymbol{x}_i : i \in \mathcal{A}_0\}$ randomly without replacement.

    end for

    Train an RF using $\{(\boldsymbol{x}_i, \boldsymbol{x}_i^\dagger, y_i) : i \in \mathcal{A}_0\}$.

    Solicit $\{(F_{rk}, F_{rk\dagger}) : k = 1, \ldots, p\}$.

    for $k = 1, \ldots, p$ **do**

        Calculate $I_{rk} = F_{rk} - F_{rk\dagger}$.

    end for

end for

for $k = 1, \ldots, p$ **do**

    Calculate $I_k = R^{-1} \sum_{r=1}^{R} I_{rk}$

end for

Output: $\{I_k : k = 1, \ldots, p\}$.

3.2 Feature Selection by Deep Neural Network

Ideally, we want to identify a set of features, say $\boldsymbol{X}^{(K_0)}$, such that

$$Y \perp \boldsymbol{X}^{(-K_0)} \mid \boldsymbol{X}^{(K_0)},$$

where $X \perp Y \mid Z$ denotes that X and Y are conditionally independent given Z, and $\boldsymbol{X}^{(-K_0)}$ contains features other than $\boldsymbol{X}^{(K_0)}$. Equivalently, we want to identify $\boldsymbol{x}^{(K_0)}$ such that

$$f(\boldsymbol{x}) = f(\boldsymbol{x}^{(K_0)}) \quad (\boldsymbol{x} \in \mathcal{X}). \tag{6}$$

Based on the ordered features by BCFI, a forward feature selection algorithm is proposed by sequentially conducting nonparametric two-sample tests using deep neural networks (FS-D). Let the ordered BCFI be $I^{(1)} \geqslant I^{(2)} \geqslant \cdots \geqslant I^{(p)}$ and $x_i^{(k)}$ be the feature corresponding to $I^{(k)}$ for $k = 1, \ldots, p$. Consider the following hypothesis testing problem:

$$H_{0K} : f(\boldsymbol{x}) = f(\boldsymbol{x}^{(K)}) \quad (\boldsymbol{x} \in \mathcal{X}), \tag{7}$$

where $\boldsymbol{x}^{(K)} = (x^{(1)}, \ldots, x^{(K)})$ contains the first K features with respect to the ordered BCFI, and $K = 1, \ldots, p$.

Although Scornet et al. (2015) investigated theoretical properties of an RFs estimator, the convergence rate of $g(\boldsymbol{x}) - f(\boldsymbol{x})$ is still an open question, where $g(\boldsymbol{x}) = \mathbb{E}\{\widehat{\theta}_{RF}(\boldsymbol{x})\}$. Thus, it is difficult to work with (7) directly. Instead, we consider $\eta = y - g(\boldsymbol{x})$ and $\eta^{(K)} = y - g(\boldsymbol{x}^{(K)})$, where $g(\boldsymbol{x}^{(K)}) = \mathbb{E}\{\widehat{\theta}_{RF}(\boldsymbol{x}^{(K)})\}$. Thus, we have

$$
\begin{aligned}
\eta^{(K)} &= y - g(\boldsymbol{x}) + [g(\boldsymbol{x}) - g(\boldsymbol{x}^{(K)})] \\
&= \eta + [g(\boldsymbol{x}) - g(\boldsymbol{x}^{(K)})].
\end{aligned}
$$

Instead of H_{0K} in (7), we consider

$$
H'_{0K} : \mathbb{P} = \mathbb{P}^{(K)}, \tag{8}
$$

where $\mathbb{P}$ and $\mathbb{P}^{(K)}$ are the distributions of η and $\eta^{(K)}$, respectively. If a feature is useful, it also contributes to $g(\boldsymbol{x})$, and vice versa. Thus, it is valid to use (8) for feature selection.

Remark 3. *Because the standard RFs (Breiman; 2001) is widely implemented in practice, we apply it to obtain BCFI and approximate $g(\boldsymbol{x})$ and $g(\boldsymbol{x}^{(K)})$ using (3). If we consider an honest tree (Wager and Athey; 2018), we can directly use $g(\boldsymbol{x})$ to approximate $f(\boldsymbol{x})$ according to the consistency result in Theorem 3 of Wager and Athey (2018).*

To test H'_{0K} in (8), we adopt a nonparametric two-sample test using the MMD-D (Liu et al.; 2020). The maximum mean discrepancy between $\mathbb{P}$ and $\mathbb{P}^{(K)}$ is

$$
\begin{aligned}
\mathrm{MMD}(\mathbb{P}, \mathbb{P}^{(K)}; \mathcal{H}_\kappa) &= \sup_{f \in \mathcal{H}_\kappa; \|f\|_{\mathcal{H}_\kappa} \leqslant 1} |E\{f(\boldsymbol{X})\} - E\{f(\boldsymbol{X}^{(K)})\}| \\
&= \{E[\kappa(\eta, \eta)] + E[\kappa(\eta^{(K)}, \eta^{(K)})] - 2E[\kappa(\eta, \eta^{(K)})]\}^{1/2},
\end{aligned}
$$

$$
\tag{9}
$$

where $\kappa : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is the kernel for a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_\kappa$, $\|f\|_{\mathcal{H}_\kappa}$ is the corresponding norm; see Supplementary Material for a brief introduction on RKHS. For a characteristic kernel κ (Fukumizu et al.; 2007; Gretton et al.; 2012), $\mathrm{MMD}(\mathbb{P}, \mathbb{P}^{(K)}; \mathcal{H}_\kappa) = 0$ is equivalent to $\mathbb{P} = \mathbb{P}^{(K)}$. Recall that a kernel κ is characteristic if the map $\mathbb{Q} \to m_{\mathbb{Q}}$ is one-to-one, where $m_{\mathbb{Q}} = \mathbb{E}_{\zeta \sim \mathbb{Q}} \kappa(\cdot, \zeta) \in \mathcal{H}_\kappa$ for $\mathbb{Q} \in \mathcal{P}_{\mathbb{R}}$, $\mathcal{P}_{\mathbb{R}}$ is the set of probability measures on the measurable space $(\mathbb{R}, \mathcal{B})$, $\mathcal{B}$ is the Borel σ-algebra on $\mathbb{R}$, and $\mathbb{E}_{\zeta \sim \mathbb{Q}} \kappa(\cdot, \zeta)$ is the expectation of $\kappa(\cdot, \zeta)$ with respect to $\zeta \sim \mathbb{Q}$.

Remark 4. *The test statistic is based on (9), and its intuition is briefly discussed. Evidently, the value* $MMD(\mathbb{P}, \mathbb{P}^{(K)}; \mathcal{H})$ *is determined by the functional space* $\mathcal{H}$. *On the one hand, the functional space* $\mathcal{H}$ *should be sufficiently large to distinguish two different distributions* $\mathbb{P}$ *and* $\mathbb{P}^{(K)}$ *by the supremum of* $|E\{f(\boldsymbol{X})\} - E\{f(\boldsymbol{X}^{(K)})\}|$ *for* $f \in \mathcal{H}$. *On the other hand, the functional space* $\mathcal{H}$ *should also be restricted, such that the estimator of* $MMD(\mathbb{P}, \mathbb{P}^{(K)}; \mathcal{H})$ *converges duly to guarantee good statistical properties. Thus, a unit ball of an RKHS,* $\{f \in \mathcal{H}_\kappa; \|f\|_{\mathcal{H}_\kappa} \leqslant 1\}$, *is a good choice for* $\mathcal{H}$. *See Gretton et al. (2012) for details on (9).*

If $\{\eta_i : i \in \mathcal{A}_1\}$ and $\{\eta_i^{(K)} : i \in \mathcal{A}_2\}$ were observed, an estimator of $MMD(\mathbb{P}, \mathbb{P}^{(K)}; \mathcal{H}_\kappa)$ could be obtained by a U-statistic based on two subsamples $\mathcal{A}_1$ and $\mathcal{A}_2$:

$$\widehat{MMD}_u^2(\widehat{\mathbb{P}}, \widehat{\mathbb{P}}^{(K)}; \kappa) = \frac{1}{m_1(m_1 - 1)} \sum_{i \neq j} H_{ij}^{(K)}, \tag{10}$$

where $\widehat{\mathbb{P}}(\eta) = m_1^{-1} \sum_{i \in \mathcal{A}_1} \mathbb{I}(\eta = \eta_j)$ is the empirical distribution of $\{\eta_i : i \in \mathcal{A}_1\}$, $\widehat{\mathbb{P}}^{(K)}$ is the one of $\{\eta_i^{(K)} : i \in \mathcal{A}_2\}$, m_1 is the size of both $\mathcal{A}_1$ and $\mathcal{A}_2$, $\mathcal{A}_1 \cap \mathcal{A}_2 = \varnothing$, and $H_{ij}^{(K)} = \kappa(\eta_i, \eta_j) + \kappa(\eta_i^{(K)}, \eta_j^{(K)}) - \kappa(\eta_i, \eta_j^{(K)}) - \kappa(\eta_i^{(K)}, \eta_j)$. The disjoint condition between $\mathcal{A}_1$ and $\mathcal{A}_2$ guarantees independence between the two error sets. The numerical results reveal that the sample sizes of $\mathcal{A}_1$ and $\mathcal{A}_2$ should be large, and we suggest that these sizes be larger than 400 for practical guidance ; see Section 4 for details.

However, η_i and $\eta_i^{(K)}$ are unavailable; thus, we use $\widehat{\eta}_{ni} = y_i - \widehat{\theta}_{RF}^{(B)}(\boldsymbol{x}_i)$ for $i \in \mathcal{A}_1$ and $\widehat{\eta}_{ni}^{(K)} = y_i - \widehat{\theta}_{RF}^{(B)}(\boldsymbol{x}_i^{(K)})$ $i \in \mathcal{A}_2$ instead. The following theorem validates this choice:

Theorem 1. *Under mild conditions, in Supplementary Material,*

$$\widetilde{MMD}_u^2(\widetilde{\mathbb{P}}, \widetilde{\mathbb{P}}^{(K)}; \kappa) = \frac{1}{m_1(m_1 - 1)} \sum_{i \neq j} \widehat{H}_{ij}^{(K)}$$

has the same limiting distribution as $\widehat{MMD}_u^2(\mathbb{P}, \mathbb{P}^{(K)}; k)$ *in (10), where* $\widetilde{\mathbb{P}}$ *and* $\widetilde{\mathbb{P}}^{(K)}$ *are the empirical distributions of* $\{\widehat{\eta}_{ni} : i \in \mathcal{A}_1\}$ *and* $\{\widehat{\eta}_{ni}^{(K)} : i \in \mathcal{A}_2\}$, $\{\widehat{\eta}_{ni} : i \in \mathcal{A}_1\}$ *is independent of* $\{\widehat{\eta}_{ni}^{(K)} : i \in \mathcal{A}_2\}$, *and* $\widehat{H}_{ij}^{(K)} = \kappa(\widehat{\eta}_i, \widehat{\eta}_j) + \kappa(\widehat{\eta}_i^{(K)}, \widehat{\eta}_j^{(K)}) - \kappa(\widehat{\eta}_i, \widehat{\eta}_j^{(K)}) - \kappa(\widehat{\eta}_i^{(K)}, \widehat{\eta}_j)$.

The proof of Theorem 1 is relegated to Supplementary Material. The difference between (10) and the one in Theorem 1 is that the estimated residuals are considered instead.

Theorem 1 validates that instead of the true residuals, we can use the estimated values, $\{\widehat{\eta}_{ni} : i \in \mathcal{A}_1\}$ and $\{\widehat{\eta}_{ni}^{(K)} : i \in \mathcal{A}_2\}$, for analysis. Thus, $\widetilde{\text{MMD}}_u^2(\widetilde{\mathbb{P}}, \widetilde{\mathbb{P}}^{(K)}; \kappa)$ serves as the test statistic for the hypothesis testing problem (8).

To guarantee independence between $\{\widehat{\eta}_{ni} : i \in \mathcal{A}_1\}$ and $\{\widehat{\eta}_{ni}^{(K)} : i \in \mathcal{A}_2\}$, we train $\widehat{\theta}_{RF}^{(B)}(\boldsymbol{x})$ and $\widehat{\theta}_{RF}^{(B)}(\boldsymbol{x}^{(K)})$ by $\{(\boldsymbol{x}_i, y_i) : i \in \mathcal{A}_3\}$ and $\{(\boldsymbol{x}_i^{(K)}, y_i) : i \in \mathcal{A}_4\}$ separately, where $\mathcal{A}_3$ and $\mathcal{A}_4$ are two subsamples with the same sample size m_2, such that $\mathcal{A}_1, \ldots, \mathcal{A}_4$ are mutually disjoint. Moreover, to avoid correlation, we suggest that $\mathcal{A}_1, \ldots, \mathcal{A}_4$ are generated from $mathcalI_n \backslash \mathcal{A}_0$. For practical guidance, we suggest that $m_2 \geqslant 400$; see Section 4 for details.

It is hard for simple kernels to distinguish two distributions with complex structures. For example, a translation-invariant Gaussian kernel requires a large sample to distinguish two distributions, since it cannot identify "direction" information around each mode in a multivariate setup; see Figure 1 and its discussion of Liu et al. (2020) for details. To avoid the drawbacks of traditional parametric kernels, Liu et al. (2020) proposed obtaining one using deep neural networks:

$$\widehat{\kappa}_\omega = \arg\max_{\kappa_\omega} \frac{\widetilde{\text{MMD}}_u^2(\widehat{\mathbb{P}}, \widehat{\mathbb{P}}^{(K)}; \kappa_\omega)}{\widetilde{\sigma}_{1,\lambda}(\widehat{\mathbb{P}}, \widehat{\mathbb{P}}^{(K)}; \kappa_\omega)},$$

where $\kappa_\omega = \{(1-\eta)\kappa[\phi_\omega(x), \phi_\omega(y)] + \eta\}q(x, y)$, $\eta > 0$ is predefined, $\phi_\omega(x)$ is a deep neural network with parameter ω that extracts features, $\kappa(x, y)$ and $q(x, y)$ are Gaussian kernels with lengthscales σ_ϕ and σ_q, respectively, and $\widetilde{\sigma}_{1,\lambda}^2(\widehat{\mathbb{P}}, \widehat{\mathbb{P}}^{(K)}; \kappa_\omega) = 4m_1^{-3}\sum_{i=1}^{m_1}(\sum_{j=1}^{m_1} H_{ij}^{(K)})^2 - 4m_1^{-4}(\sum_{i=1}^{m_1}\sum_{j=1}^{m_1} H_{ij}^{(K)})^2 + \lambda$.

The limiting distributions of the U-statistic $\widetilde{\text{MMD}}_u^2(\widehat{\mathbb{P}}, \widehat{\mathbb{P}}^{(K)}; \mathcal{H}_\kappa)$ are established below for the null and alternative hypotheses.

Lemma 1. *Under the null hypothesis $H'_{0K} : \mathbb{P} = \mathbb{P}^{(K)}$ and regularity conditions in Supplementary Material, we have*

$$m_1 \widetilde{MMD}_u^2(\widehat{\mathbb{P}}, \widehat{\mathbb{P}}^{(K)}; \kappa_\omega) \to \sum_{i=1}^\infty \zeta_i(Z_i^2 - 2) \tag{11}$$

in distribution, where ζ_i are the eigenvalues satisfying

$$\int \kappa_\omega(\boldsymbol{x}, \boldsymbol{z})\Psi_i(\boldsymbol{x})\mathbb{P}(\mathrm{d}\boldsymbol{x}) = \zeta_i \Psi_i(\boldsymbol{z})$$

11

for $i = 1, 2, \ldots,$, $\{\Psi_i(\boldsymbol{x}) : i = 1, 2, \ldots\}$ are the eigenfunctions, $Z_i \sim N(0, 2)$, and $\mathcal{N}(\mu, \sigma^2)$ is a normal distribution with mean μ and variance σ^2.

Under the alternative hypothesis $H'_{aK} : \mathbb{P} \neq \mathbb{P}^{(K)}$, we have

$$\sqrt{m_1}[\widehat{MMD}_u^2 - MMD^2] \to \mathcal{N}(0, \sigma_1^2)$$

in the distribution, where $\sigma_1^2 = 4[\mathbb{E}(H_{12}H_{13}) - \mathbb{E}(H_{12})^2]$.

By Theorem 1, we can prove Lemma 1 in a manner similar to Theorem 12 of Gretton et al. (2012), so its proof is omitted. Instead of deriving the asymptotic distributions in Lemma 1, Liu et al. (2020) suggested permutation for hypothesis testing. Algorithm 2 shows the routine of FS-D, and the detailed algorithm for the MMD-D step is relegated to Supplementary Material.

Remark 5. *The basic idea of the NFSRD is to apply nonparametric two-sample tests sequentially based on ordered features. Thus, the NFSRD is also applicable to other machine learning algorithms, such as XGBoost (Chen and Guestrin; 2016) and the causal tree (Wager and Athey; 2018), as long as the features can be sorted accordingly.*

4   Simulation

In this section, we conduct Monte Carlo simulations to compare the performance of the NFSRD with its alternatives under different model setups. As mentioned in Section 1, there are two main approaches for feature selection by random forests. Thus, we compare the NFSRD with a feature-importance-based algorithm and a minimum-depth-based one.

We use 19 Xeon Cascade Lake (2.5 GHz) CPUs to train RFs in parallel, and an NVIDIA Tesla T4 GPU is used for FS-D. Five-fold cross validation is conducted to tune the model parameters for RFs; see Supplementary Material for more computational details.

4.1   Independent Features

Table 1 shows the setups for six synthetic models. Model 1 is linear and is widely used for feature selection (Tibshirani; 1996). Model 2 represents a nonlinear model and only

Algorithm 2 Feature Selection by Deep Neural Network (FS-D)

Input: $\mathcal{D}_n$, m_2, $\{I_k : k = 1, \ldots, p\}$, and α.

Generates mutually disjoint sets $\mathcal{A}_1, \ldots, \mathcal{A}_4$ from $\mathcal{I}_n \backslash \mathcal{A}_0$.

Train $\widehat{\theta}_{RF}^{(B)}(\boldsymbol{x})$ by $\{(\boldsymbol{x}_i, y_i) : i \in \mathcal{A}_3\}$.

Obtain $\{\widehat{\eta}_{ni} : i \in \mathcal{A}_1\}$.

for $k = 1, \ldots, p$ **do**

    if $K = 1$ **then**

        Train KRR by $\{(\boldsymbol{x}_i^{(1)}, y_i) : i \in \mathcal{A}_4\}$.

        Obtain $\{\widehat{\eta}_{ni}^{(K)} : i \in \mathcal{A}_2\}$.

    else

        Train $\widehat{\theta}_{RF}^{(B)}(\boldsymbol{x}^{(K)})$ by $\{(\boldsymbol{x}_i^{(K)}, y_i) : i \in \mathcal{A}_3\}$.

        Obtain $\{\widehat{\eta}_{ni}^{(K)} : i \in \mathcal{A}_2\}$.

    end if

    Conduct MMD-D with significant level α.

    if H'_{0K} is rejected **then**

        Continue.

    else

        Denote $\widehat{K}_0 = K$.

        Break the for loop.

    end if

end for

Output: Selected feature set $\mathcal{P}$.

---

involves one useful feature. Both Model 1 and Model 2 are additive, and Model 3 is more complex and is non-additive. Besides, interaction is involved in Model 3. The difference between Models 1–3 and Models 4–6 is the distribution to generate features. Specifically, features are generated independently from a uniform distribution over $[1, 10]$ for Models 1–3, and a *skewed* beta distribution with shape parameter (2,4) is used for Models 4–6. The distribution parameters are chosen such that the features have approximately the same variance in different synthetic models, and the regression parameters are selected such that the signal-to-noise ratio (SNR) ranges from 1 to 3 approximately, where SNR is obtained by $\{\text{var}[f(\boldsymbol{X})]/\text{var}(\epsilon)\}^{1/2}$, and $\text{var}(X)$ is the variance of a random variable X. For each model, $\epsilon \sim \mathcal{N}(0, 1)$. Furthermore, we consider two feature sizes, including $p = 200$ and

$p = 400$. Although the number of features is large, the useful ones are limited. The sample size is $n = 10\,000$. For each synthetic model, we conduct 200 independent Monte Carlo simulations.

Table 1: Setups for synthetic models. "MI" stands for the model index, and "Dist" represents the distribution to generate features. "Uniform" corresponds to $X_k \sim \mathrm{U}(1, 10)$, and "Beta" relates to $X_k \sim 14.5\mathrm{Beta}(2, 4) + 1$, where $k = 1, \ldots, p$. The error term is $\epsilon \sim \mathcal{N}(0, 1)$.

MI	Dist	Model Setup	SNR
1	Uniform	$Y = 0.3X_1 + 0.3X_2 + \epsilon$	1.1
2	Uniform	$Y = 3\sin(X_1) + \epsilon$	2.1
3	Uniform	$Y = 5\sin(X_1/10)\sqrt{X_2} + \epsilon$	3.0
4	Beta	$Y = 0.3X_1 + 0.3X_2 + \epsilon$	1.1
5	Beta	$Y = 3\sin(X_1) + \epsilon$	2.1
6	Beta	$Y = 5\sin(X_1/10)\sqrt{X_2} + \epsilon$	2.8

We compare the following methods in terms of feature selection under the significance level $\alpha = 0.05$:

1. Boruta (Kursa and Rudnicki; 2010, BRT). BRT calculates the feature importance by a set of shadow features, and a "Z score" is used for feature selection; see Supplementary Material for a brief description of BRT. BRT is implemented using the R package `Boruta`.

2. BRT-N. NFSRD is conducted based on the features sorted by BCFI, as obtained from BRT.

3. Minimal depth variable selection (Ishwaran et al.; 2010, MVS). MVS selects features using the minimum depth (Ishwaran et al.; 2010); see Supplementary Material for a brief description of MVS. This method is implemented using the R package `randomForestSRC`.

4. MVS-N. NFSRD is conducted based on the features sorted by the minimum depth obtained from MVS.

Although `VSURF` (Genuer et al.; 2015) also conducts feature selection by RFs, it is not considered because of its heavy computational burden. For both BRT-N and MVS-N, we consider two scenarios: $m_0 = m_1 = m_2 = 200$ and $m_0 = m_1 = m_2 = 400$. For a fair comparison, $5m_0$ instances are used for feature selection by BRT and MVS.

The four methods are compared in terms of the accuracy of identifying useful features, μ_c, and the average number of useless features included, n_w, where $\mu_c = (200K_0)^{-1} \sum_{l=1}^{200} \sum_{k=1}^{K_0} \mathbb{I}_{lk}$, $n_w = 200^{-1} \sum_{l=1}^{200} n_{lw}$, $\mathbb{I}_{lk} = 1$ is the kth useful feature that is correctly identified in the lth Monte Carlo simulation, and 0 otherwise, n_{lw} is the number of useless features identified in the lth Monte Carlo simulation. The simulation results are summarized in Table 2. When the subsample sizes are small, μ_c is less than 1 for Model 1, indicating that it is difficult for the two NFSRD algorithms to identify some useful features even for a linear model when the SNR is small. This is because $\widehat{\theta}_{RF}^{(B)}(\boldsymbol{x}) - g(\boldsymbol{x})$ or $\widehat{\theta}_{RF}^{(B)}(\boldsymbol{x}^{(K)}) - g(\boldsymbol{x}^{(K)})$ is not negligible compared with the residual; thus, we cannot test (8) correctly. However, as the SNR increases, both BRT-N and MVS-N identify more useful features on average, even for small subsamples and more complex models. On the one hand, even when m_0 is small, BRT and MVS, on the other hand, can correctly select useful features with $5m_0$ instances for different models because $5m_0$ instances are used for feature selection. As m_1 and m_2 increase, useful features can be correctly identified by BRT-N and MVS-N as well as BRT and MVS for different models. However, BRT-N and MVS-N identify far less useless features compared with BRT and MVS, especially when m_1 is large. For example, when $m_1 = 200$ and $p = 400$, BRT and MVS identify 2.17 and 34.88 useless features on average for a linear model, respectively, but the average numbers are only 0.4 and 0.3 for BRT-N and MVS-N. The same conclusion applies to the other setups. As m_1 and m_2 increase to 400, BRT-N and MVS-N still outperform their alternatives in terms of avoiding useless features.

We also compare the four methods in terms of computational efficiency, and Figure 1 shows boxplots of the computation time based on the 200 Monte Carlo simulations under

Table 2: Summary of feature selection the four methods, including BRT, BRT-N, MVS, and MVS-N, based on 200 Monte Carlo simulations, where the features are generated independently. "MI" stands for the model index. m_0, m_1 and m_2 are the sizes of subsamples for NFSRD algorithms. "BRT" represent Boruta, "MVS" stands for the minimum-depth variable selection algorithm, and "BRT-N" and "MVS-N" are NFSRD algorithms with features sorted by BRT and MVS, respectively. μ_c is proportion that useful features are correctly identified, and n_w is the average length of useless features are included. The best result is highlighted by underline.

p	MI		$m_0 = m_1 = m_2 = 200$				$m_0 = m_1 = m_2 = 400$			
			BRT	BRT-N	MVS	MVS-N	BRT	BRT-N	MVS	MVS-N
200	1	μ_c	1.00	0.90	1.00	0.90	1.00	0.98	1.00	0.98
		n_w	1.75	0.38	4.91	0.34	1.35	0.40	0.02	0.46
	2	μ_c	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
		n_w	2.16	0.60	5.17	0.59	1.42	0.54	0.12	0.49
	3	μ_c	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
		n_w	1.52	0.83	1.21	0.95	0.98	0.82	0.00	0.88
	4	μ_c	1.00	0.78	1.00	0.81	1.00	0.90	1.00	0.88
		n_w	2.21	0.26	6.83	0.32	1.43	0.32	0.04	0.43
	5	μ_c	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
		n_w	1.98	0.66	4.42	0.71	1.30	0.54	0.08	0.40
	6	μ_c	1.00	0.94	1.00	0.93	1.00	1.00	1.00	1.00
		n_w	1.57	0.78	1.61	0.75	1.13	0.66	0.00	0.74
400	1	μ_c	1.00	0.86	1.00	0.85	1.00	0.98	1.00	0.98
		n_w	2.19	0.37	23.72	0.24	1.44	0.42	5.50	0.46
	2	μ_c	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
		n_w	2.35	0.64	19.60	0.56	1.34	0.58	5.43	0.44
	3	μ_c	1.00	0.99	1.00	0.99	1.00	1.00	1.00	1.00
		n_w	1.79	1.00	3.98	1.16	1.12	0.66	1.21	0.68
	4	μ_c	1.00	0.90	1.00	0.90	1.00	0.99	1.00	0.99
		n_w	2.17	0.40	34.88	0.30	1.48	0.52	7.03	0.61
	5	μ_c	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
		n_w	2.30	0.73	28.79	0.47	1.38	0.47	4.75	0.58
	6	μ_c	1.00	0.87	1.00	0.88	1.00	1.00	1.00	1.00
		n_w	1.58	0.61	17.16	0.66	0.99	0.81	1.28	0.76

different setups. When both m_0 and p are small, the computational efficiencies of the four methods are comparable, but, as shown in Table 2, BRT and MVS select more useless

Figure 1: Boxplots for the computation time of different methods based on 200 Monte Carlo simulations, where the features are independently generated. The horizontal segments within the boxplots represent the median computation time. "BRT" represent Boruta, "MVS" stands for the minimum-depth variable selection algorithm, and "BRT-N" and "MVS-N" are NFSRD algorithms with features sorted by BRT and MVS, respectively.

features than BRT-N and MVS-N. Thus, BRT and MVS are still less favored. As m_0 or p increases, the computational efficiency of BRT-N and MVS-N is generally much better, especially when both m_0 and p are large. For example, when $m_0 = 400$ and $p = 400$, the computation time required by BRT-N and MVS-N is less than 50 seconds in general, but it is more than 125 seconds and approximately 75 seconds for BRT and MVS, respectively. In addition, it is far less efficient to use BRT and MVS to select features for large datasets with numerous useless features. However, BRT-N and MVS-N can be used to solve this problem. Compared with MVS-N, BRT-N is slightly more computationally efficient in general, but the difference between the two methods is minor.

Remark 6. *It is common to have correlated features in practice, and we conduct a simulation study for this case as well. We still consider the same setups in Table 1, but the features are generated differently. First, generate $X'_1, \ldots, X'_p$ independently by a uniform or skewed*

17

beta distribution, as shown in Table 1. Then, let $X_1 = X'_1$ and $X_k = 0.7X'_k + 0.3X'_{k-1}$ for $k = 2, \ldots, p$. The features consist of $X_1, \ldots, X_p$, and we still use the six models in Table 1 to generate the responses of interest. The simulation results are similar to the aforementioned outcomes, and we relegate them to Supplementary Material.

Ishwaran et al. (2010) compared the MVS with some commonly used feature selection algorithms, including the adaptive lasso (Zou; 2006) and the l_1-regularized regression model (Park and Hastie; 2007). Their simulation results demonstrated that the MVS outperforms those two in terms of the false discovery rate and the false nondiscovery rate, regardless of the correlation among features. Since the MVS-N is generally more preferable than MVS, we do not consider those two feature selection algorithms in the simulation study.

5 Application

The superconductivity dataset Hamidieh (2018) is used to test the performance of the NFSRD. Superconducting materials have wide applications in practice, such as magnetic resonance imaging systems in hospitals and superconducting coils in the Large Hadron Collider at CERN. The accurate prediction of the superconducting critical temperature is important because the corresponding superconductor can only conduct current without resistance at or below this temperature. There are 21 263 instances in the superconductivity dataset with 81 features extracted for each instance, and the goal is select useful features for the critical temperature; see Hamidieh (2018) for details about the technical information of the features.

For BRT-N, we consider two significance levels: $\alpha = 0.05$ and $\alpha = 0.01$, and the size of $\mathcal{A}_0, \ldots, \mathcal{A}_4$ is 1 000. Table 3 lists the selected useful features. As the significance level decreases from 0.05 to 0.01, three more features are selected. In addition, all the selected features are among the top 20 most important features by XGBoost (Chen and Guestrin; 2016); see Table 5 of Hamidieh (2018) for details. In contrast, BRT blindly identifies all the features to be useful, even for $\alpha = 0.05$. Although BRT works reasonably for synthetic models in Section 4, it fails to identify useful features for superconductivity data. A similar

conclusion holds for the two minimum-depth-based methods, and, thus, we omit them.

Table 3: The selected features by BRT-N under different significance levels for the superconductivity data. Selected features are indicated by "✓", and "Feature" shows the corresponding names. "α" is the significant level.

Feature	α	
	0.01	0.05
range_ThermalConductivity	✓	✓
wtd_std_ThermalConductivity	✓	✓
range_atomic_radius	✓	✓
wtd_entropy_atomic_mass	✓	✓
wtd_entropy_Valence	✓	✓
wtd_mean_Valence	✓	✓
wtd_mean_ThermalConductivity	✓	✓
std_ThermalConductivity	✓	✓
wtd_gmean_Valence	✓	
wtd_gmean_ThermalConductivity	✓	
wtd_std_ElectronAffinity	✓	

6 Concluding Remarks

In this study, we propose the NFSRD to identify useful features. Feature selection is conducted by nonparametric two-sample tests using deep neural networks, and the theoretical properties are also investigated. Experiments show that the NFSRD outperforms its alternatives in terms of identifying useful features, avoiding useless ones and the computation efficiency.

Although the NFSRD is proposed using the standard RFs (Breiman; 2001), the same idea can be easily adapted to other machine learning methods. In addition, other nonparametric tests can also be used; these include other nonparametric kernel-based methods (Scholkopf and Smola; 2018) and traditional statistical tests. However, we should pay attention to some widely used tests, such as the t-test and the Kolmogorov–Smirnov test, because they may suffer from model misspecification.

7　Acknowledgements

References

Altmann, A., Toloşi, L., Sander, O. and Lengauer, T. (2010). Permutation importance: a corrected feature importance measure, *Bioinformatics* **26**(10): 1340–1347.

Benci, J. L., Xu, B., Qiu, Y., Wu, T. J., Dada, H., Twyman-Saint Victor, C., Cucolo, L., Lee, D. S., Pauken, K. E., Huang, A. C. et al. (2016). Tumor interferon signaling regulates a multigenic resistance program to immune checkpoint blockade, *Cell* **167**(6): 1540–1554.

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

Caruana, R. and Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms, *Proceedings of the 23rd International Conference on Machine Learning*, pp. 161–168.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794.

Criminisi, A., Shotton, J. and Konukoglu, E. (2012). Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning, *Foundations and Trends® in Computer Graphics and Vision* **7**(2–3): 81–227.

Fan, J. and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **70**(5): 849–911.

Fernández-Delgado, M., Cernadas, E., Barro, S. and Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems?, *Journal of Machine Learning Research* **15**(1): 3133–3181.

Fukumizu, K., Gretton, A., Sun, X. and Schölkopf, B. (2007). Kernel measures of conditional dependence., *Advances in Neural Information Processing Systems*, Vol. 20, pp. 489–496.

Genuer, R., Poggi, J.-M. and Tuleau-Malot, C. (2010). Variable selection using random forests, *Pattern Recognition Letters* **31**(14): 2225–2236.

Genuer, R., Poggi, J.-M. and Tuleau-Malot, C. (2015). VSURF: an R package for variable selection using random forests, *The R journal* **7**(2): 19–33.

Goel, E., Abhilasha, E., Goel, E. and Abhilasha, E. (2017). Random forest: A review, *International Journal of Advanced Research in Computer Science and Software Engineering* **7**(1): 251–257.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. and Smola, A. (2012). A kernel two-sample test, *The Journal of Machine Learning Research* **13**(1): 723–773.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection, *Journal of Machine Learning Research* **3**(Mar): 1157–1182.

Haghiri, S., Garreau, D. and von Luxburg, U. (2018). Comparison-based random forests, *Proceedings of the 35th International Conference on Machine Learning*, pp. 1871–1880.

Hamidieh, K. (2018). A data-driven statistical model for predicting the critical temperature of a superconductor, *Computational Materials Science* **154**: 346–354.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd edn, Springer, New York.

Ishwaran, H., Kogalur, U. B., Blackstone, E. H. and Lauer, M. S. (2008). Random survival forests, *Annals of Applied Statistics* **2**(3): 841 – 860.

Ishwaran, H., Kogalur, U. B., Gorodeski, E. Z., Minn, A. J. and Lauer, M. S. (2010). High-dimensional variable selection for survival data, *Journal of the American Statistical Association* **105**(489): 205–217.

Kursa, M. B. and Rudnicki, W. R. (2010). Feature selection with the Boruta package, *Journal of Statistical Software* **36**(1): 1–13.

Lakshminarayanan, B., Roy, D. M. and Teh, Y. W. (2014). Mondrian forests: Efficient online random forests, *Advances in Neural Information Processing Systems*, pp. 3140–3148.

Li, A. H. and Martin, A. (2017). Forest-type regression with general losses and robust forest, *Proceedings of the 34th International Conference on Machine Learning*, pp. 2091–2100.

Li, X., Wang, R., Basu, S., Kumbier, K. and Yu, B. (2019). A debiased MDI feature importance measure for random forests, *Advances in Neural Information Processing Systems*, Vol. 32, pp. 1–22.

Liu, F., Xu, W., Lu, J., Zhang, G., Gretton, A. and Sutherland, D. J. (2020). Learning deep kernels for non-parametric two-sample tests, *Proceeding of the 37th International Conference on Machine Learning*, pp. 6316–6326.

Louppe, G., Wehenkel, L., Sutera, A. and Geurts, P. (2013). Understanding variable importances in forests of randomized trees, *Advances in Neural Information Processing Systems*, pp. 431–439.

Meinshausen, N. (2006). Quantile regression forests, *Journal of Machine Learning Research* **7**(Jun): 983–999.

Mentch, L. and Hooker, G. (2016). Quantifying uncertainty in random forests via confidence intervals and hypothesis tests, *Journal of Machine Learning Research* **17**(1): 841–881.

Park, M. Y. and Hastie, T. (2007). L1-regularization path algorithm for generalized linear models, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **69**(4): 659–677.

Payet, N. and Todorovic, S. (2010). (RF)^2 – random forest random field, *Advances in Neural Information Processing Systems*, pp. 1885–1893.

Sandri, M. and Zuccolotto, P. (2008). A bias correction algorithm for the Gini variable importance measure in classification trees, *Journal of Computational and Graphical Statistics* **17**(3): 611–628.

Scholkopf, B. and Smola, A. J. (2018). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, London.

Scornet, E., Biau, G. and Vert, J.-P. (2015). Consistency of random forests, *Annals of Statistics* **43**(4): 1716–1741.

Shi, T. and Horvath, S. (2006). Unsupervised learning with random forest predictors, *Journal of Computational and Graphical Statistics* **15**(1): 118–138.

Siblini, W., Kuntz, P. and Meyer, F. (2018). CRAFTML, an efficient clustering-based random forest for extreme multi-label learning, *Proceedings of the 35th International Conference on Machine Learning*, pp. 4671–4680.

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T. and Zeileis, A. (2008). Conditional variable importance for random forests, *BMC Bioinformatics* **9**(1): 307.

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

Twyman-Saint Victor, C., Rech, A. J., Maity, A., Rengan, R., Pauken, K. E., Stelekati, E., Benci, J. L., Xu, B., Dada, H., Odorizzi, P. M. et al. (2015). Radiation and dual checkpoint blockade activate non-redundant immune mechanisms in cancer, *Nature* **520**(7547): 373–377.

Wager, S. and Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests, *Journal of the American Statistical Association* **113**(523): 1228–1242.

White, A. P. and Liu, W. Z. (1994). Bias in information-based measures in decision tree induction, *Machine Learning* **15**(3): 321–329.

Xu, P. and Jelinek, F. (2005). Using random forests in the structured language model, *Advances in Neural Information Processing Systems*, pp. 1545–1552.

Zou, H. (2006). The adaptive lasso and its oracle properties, *Journal of the American Statistical Association* **101**(476): 1418–1429.