Gene expression

Knockoff boosted tree for model-free variable selection

Tao Jiang¹, Yuanyuan Li² and Alison A. Motsinger-Reif ²,*

¹Department of Statistics, Bioinformatics Research Center, North Carolina State University, Raleigh, NC 27695, USA and ²Biostatistics & Computational Biology Branch, National Institute of Environmental Health Sciences, Durham, NC 27709, USA

*To whom correspondence should be addressed.

Received on April 4, 2020; revised on August 17, 2020; editorial decision on August 24, 2020; accepted on September 9, 2020

Abstract

Motivation: The recently proposed knockoff filter is a general framework for controlling the false discovery rate (FDR) when performing variable selection. This powerful new approach generates a ‘knockoff’ of each variable tested for exact FDR control. Imitation variables that mimic the correlation structure found within the original variables serve as negative controls for statistical inference. Current applications of knockoff methods use linear regression models and conduct variable selection only for variables existing in model functions. Here, we extend the use of knockoffs for machine learning with boosted trees, which are successful and widely used in problems where no prior knowledge of model function is required. However, currently available importance scores in tree models are insufficient for variable selection with FDR control.

Results: We propose a novel strategy for conducting variable selection without prior model topology knowledge using the knockoff method with boosted tree models. We extend the current knockoff method to model-free variable selection through the use of tree-based models. Additionally, we propose and evaluate two new sampling methods for generating knockoffs, namely the sparse covariance and principal component knockoff methods. We test and compare these methods with the original knockoff method regarding their ability to control type I errors and power. In simulation tests, we compare the properties and performance of importance test statistics of tree models. The results include different combinations of knockoffs and importance test statistics. We consider scenarios that include main-effect, interaction, exponential and second-order models while assuming the true model structures are unknown. We apply our algorithm for tumor purity estimation and tumor classification using Cancer Genome Atlas (TCGA) gene expression data. Our results show improved discrimination between difficult-to-discriminate cancer types.

Availability and implementation: The proposed algorithm is included in the KOBT package, which is available at https://cran.r-project.org/web/packages/KOBT/index.html.

Contact: motsingerreifaa@nih.gov

Supplementary information: Supplementary data are available at Bioinformatics online.

1 Introduction

Variable selection is arguably one of the most important challenges in ‘big data’ analysis. Across many scientific disciplines, including a broad range of bioinformatics and statistical genetics applications, variable selection is a primary experimental focus. For example, gene expression data is often collected to narrow down genes across the genome to a subset involved in a specified biological process or condition. Predictors in the final selected model are necessarily a subset of the candidate set (Li et al., 2005). This selection process is not straightforward given the complexity of biological data, especially when considering the order (e.g. quadratic, cubic) of main effects and interactions with a large number of variables. Model-free variable selection methods have accordingly become important data mining tools that help avoid the need for jointly performing model selection and variable selection. Li et al. (2005) stated that the difference between model selection and variable selection is whether to assume a model for comparison. Model-free methods, such as random Forest (Breiman, 2001), AdaBoost (Ratsch et al., 2001) and convolutional neural networks (LeCun et al., 1995), are widely used in research and industry for regression and classification purposes.

These methods recognize data patterns but do not necessitate imposing specific model structures on regression functions. This independence allows model-free methods more flexibility, making them highly appropriate in many cases.

Among the widely used model-free methods, tree-based models have demonstrated greater effectiveness and interpretability than most other learning algorithms (Song and Ying, 2015). For example, in a standard regression model, an interaction term is not estimated if it is not specified in the model. In contrast, tree-based methods automatically include interactions during tree growing, without the requirement of an a priori formula preconception (Su et al., 2011).
Further, tree-based models are more naturally associated with non-linear interactions (Schizl et al., 2018), and the branches in a tree model are original interaction variables. Friedman (2001) stated that the number of terminal nodes of a boosted tree model should depend on the highest order of the dominant interactions among the variables. This implies that (i) terminal (leaf) nodes in tree models cannot be interpreted simply as main effects of original variables and (ii) growth of a tree model is selective.

Variable selection with tree-based methods is an active research area. For example, variable selection is affected by the number of feature categories (Kim and Loh, 2001), with variables with more categories preferred. To eliminate this bias in variable selection, Loh (2002) proposed the GUIDE algorithm, which is used to conduct $\chi^2$ analysis of residuals and bootstrap calibration of significance probabilities. In Random Forest methods (Breiman, 2001), permutation testing has been proposed to generate an unbiased importance index for variable selection (Breiman, 2001). Similar efforts are also ongoing for boosted trees. Miller et al. (2016) extended gradient tree boosting to multivariate tree boosting and proposed non-parametric regression to identify important variables. Several other heuristics have been proposed, with commonly used importance ranking measures such as gain, weight and cover, in the highly successful XGBoost (Chen and Guestrin, 2016). Further, SHapley Additive exPlanation (SHAP) (Lundberg and Lee, 2017b) has been proposed to interpret model predictions, and its performance has been highly consistent when applied in tree models (Lundberg et al., 2018).

While there are many goals of data mining, preventing overfitting is important in all contexts. Originally described for Random Forests (Breiman, 2001), permutation methods are widely used to generate a null distribution of test statistics to detect significance in tree models. Unfortunately, some statistical inference, such as accuracies if all instances have a weight of 1. Similar to elastic net (Zou and Hastie, 2005), XGBoost introduces $\ell_2$ regularization and the learning rate $\eta$ is a loss function based on the given response $y$ is a loss function based on the given response $y$. The $p$-th tree structure is

$$f_{\mathcal{B}}(\mathcal{X}, \mathcal{Y}; f_{\mathcal{B}}^{-1}) = \frac{1}{m} \sum_{i=1}^{m} I(\mathcal{X}_i \in \mathcal{R}_G^k)w_i,$$

where $I(\cdot)$ is an indicator function of $\mathcal{X}_i$, $\mathcal{X}_i \in \mathcal{R}_G^k$, the $k$-th disjoint partition region, and otherwise $I(\mathcal{X}_i \in \mathcal{R}_G^k) = 0$. $\mathcal{R}_G^k$ is a split region or a so-called leaf, $k = 1, \ldots, m$. The estimated tree function $\hat{f}(\cdot)$ is flexible depending on the data splitting results. When a single tree is not strong enough or when there is high bias or a high variance problem, ensemble tree models such as bagging or boosting structures are preferred. Defining a boosted tree model as a sum of $B$ trees, $f_B(\mathcal{X}, \mathcal{Y}; f_{\mathcal{B}}^{-1}) = \frac{1}{m} \sum_{i=1}^{m} I(\mathcal{X}_i \in \mathcal{R}_G^k)w_i$, the FDR is controlled using the stagewise algorithm (Hastie et al., 2005), the $B$th tree structure is found by solving $\Theta_B = \arg\min_{\Theta_B} L(\mathcal{Y}; f_B^{-1}(\mathcal{X}, \Theta_B))$, where $L(\Theta_B; \gamma) = f_B^{-1}(\mathcal{Y}) - \mathcal{X}, \Theta_B$ is a loss function based on the $B$th tree structure, given response $\gamma$ and previously fitted (B−1) tree models $f_B^{-1}(\mathcal{X}, \Theta_{B-1})$.

In boosted tree model fitting, the objective function generally contains a loss function and penalty function for tree complexity. The complexity depends on the number of ensemble trees, the depth of each tree and the number of terminal nodes in each tree. One way to control the complexity of tree model fitting is to set a minimum number of instances in a single terminal node. For example, the min_child_weight parameter in XGBoost (Chen and Guestrin, 2016) controls the minimum sum of instance weight (Hessian) in a terminal node, which is equivalent to a minimum number of instances if all instances have a weight of 1. Similar to elastic net (Zou and Hastie, 2005), XGBoost introduces $L_1$- and $L_2$-norm penalty terms in objective functions. Therefore, the objective function for the $B$th tree can be described as

$$\text{Obj}(\Theta_B; \gamma, f_B^{-1}) = \frac{1}{m} \sum_{i=1}^{m} I(\mathcal{X}_i \in \mathcal{R}_G^k)w_i + \gamma \| \Theta_B \|_1 + \frac{1}{2} \| \Theta_B \|_2^2,$$

where $w_i$ is a vector of leaf weights in the $B$th tree, and $\gamma \geq 0$, $\lambda > 0$ and $\alpha = 0$ are tuning parameters. Besides setting the minimum number of instances in terminal nodes and applying penalization on node weights, regularization can be achieved by limiting the maximum depth of a tree, applying the maximum number of trees in a boosting sequence, or defining an early stopping criterion for boosting (Zhang and Yu, 2005). Given the numerous parameters and hyperparameters of regularization and tree structure, it is tedious to find the optimal group of parameters by grid search. Accordingly, other methods such as Bayesian optimization (Snook et al., 2012) are preferred for parameter tuning.

As reviewed by Nielsen (2016), there are three kinds of regularization parameters: boosting parameters, randomization parameters and tree parameters. Boosting parameters include the number of trees $B$ and the learning rate $\eta$. A small step-size of the learning rate has been found to play an important role in the convergence of boosting procedures (Zhang and Yu, 2005). In simulation studies,
decreasing the learning rate increased the performance of boosted models (Friedman et al., 2000). Using a smaller learning rate and thus a relatively larger number of trees has been suggested, given the relationship between the learning rate and the number of boosted models. The randomization parameters refer to the row subsampling parameter, which controls the ratio for a subset of the data, and the column subsampling parameter, which determines the ratio for a subgroup of features in each tree fitting. All of the tree parameters implicate a tradeoff between bias and variance. In each tree, besides the regularization parameters $\gamma$, $d$, and $x$, the tree parameters include structure parameters, the maximum depth of the tree and the minimum sum of observation weights required for each leaf. The highest order of interaction that a tree model can reach is limited by the maximum depth of the tree, $D$. The minimum sum of observation weights in a leaf determines the variance of $w$, the estimated weights of leaves in a tree. If the minimum sum is large, the growth of a tree will be conservative, which means fewer leaves will be grown and thus a smaller variance of $w$.

Finally, we discuss how Bayesian optimization is applied for tuning $\gamma$, $d$, and $x$ in Equation (1). We define a configuration of tuning hyperparameters, $\theta = (\gamma, d, x)$, for the function we want to minimize, CVTE$[\gamma, d, x, X] = \frac{1}{2} \sum (y_i - f^\theta(x_i))^2$, where CVTE$\theta$, $x_i$ is a row vector containing $p$ features of an observation and $f^\theta()$ is the fitted boosted tree model with $B$ individual trees. An early stopping criterion states that if adding new trees does not decrease cross-validation error within five trees, the boosting iteration should be stopped before the maximum number of trees is reached. The value of $B$ is equal to the number of trees in the sequence, where the combination of trees provides the lowest cross-validation error score. This error score is the output value from CVTE$\theta$, and the support sets of $\gamma$, $d$, and $x$ are located within a tuning region, $[0, 20]$. The optimal combination is used in the final model for comparison.

2.2 Knockoff variables in boosted tree models

Barber and Candès (2015) proposed the knockoff filter, a new variable selection method that controls the FDR. The knockoff filter generates knockoff variables that mimic the correlation structure of original variables but are not associated with the response. These knockoff variables are used as controls for the original variables, so only the original variables with a significantly stronger association with the response than their respective knockoffs are selected. The knockoff filter has been shown to provide accurate FDR control, which cannot be realistically achieved using permutation methods. Candès et al. (2018) extended the knockoff framework to model-X and high-dimensional knockoffs. Following the definition of model-X knockoffs in Candès et al. (2018), we restate the definition for boosted tree models below.

**Definition 2.1. Boosted Tree Models**

For a row vector of $p$ random variables $x_{1:p} = (x_1, x_2, \ldots, x_p)$, where each $x_i$ is a random variable that represents a feature, the corresponding model-X knockoff variables $z_{1:p} = (z_1, z_2, \ldots, z_p)$ are constructed such that:

1. For a combined random vector $(x, z)_{1:2p}$, if its $j$th random variable is switched with its $(j + p)$th random variable ($j = 1, \ldots, p$) (i.e. an original variable is switched with its knockoff counterpart), the distribution of the new random vector is invariant to $(x, z)_{1:2p}$ and $z_i$ is sampled from the conditional distribution $L(x_j | x_{-j}, z_{1:j-1})$, where $j = 1, \ldots, p$, and $x_j = (x_{j-1}, x_{j+p})$. Approximate construction focuses on whether $(x, z)_{1:2p}$ retains its first two moments of a distribution after swapping. Given this summary of the two methods, it is apparent that the approximate construction method requires less complex computations than the exact construction method.

In this study, we compare three algorithms for knockoff generation. The first algorithm, approximate construction (AC), is available in the knockoff R package (Candes et al., 2018), where the covariance matrix of original variables, $\Sigma(X_{1:p})$, is estimated directly. The estimated covariance matrix is shrunk to the identity matrix if it is not positive definite (Oppegren and Strimmer, 2007). In addition, we propose two other algorithms: one that is similar to the AC algorithm described above and one without Gaussian assumption. The second algorithm (i.e. the one that is similar to the AC algorithm) is a sparse construction (SC) algorithm. Instead of estimating the covariance matrix directly, we conduct sparse estimation of the covariance matrix (Bien and Tibshirani, 2011), which generates a sparse matrix with a simpler structure. The rest of the algorithm is the same as the AC algorithm. The third algorithm (i.e. the one without Gaussian assumption) is a principal component construction (PCC) algorithm and was inspired by Algorithm A.1 in Shen et al. (2019).

In accordance with Definition 2.1, we describe our proposed PCC algorithm in Algorithm 2.1 below. Unlike the AC and SC algorithms, the PCC algorithm does not require data with a Gaussian assumption.

**Algorithm 2.1 Principal Component Construction (PCC)**

For each column vector of original variable $x_j$, where $j = 1, \ldots, p$,

1. Conduct principal component analysis on matrix $(X_{-j}, Z_{1:j-1})_{k\times(p+j-1)}$, where $X_{-j}$ is matrix $X$ without the $j$th column and $Z_{1:j-1}$ is the first $(j-1)$ columns in matrix $Z$. When $j = 1$, $Z_{1:j-1}$ is empty.
2. Denote $K$ as the number of principal components chosen for a regression model, $K = 1, \ldots, n-1$. For a fixed $K$, fit $x_j$ on $K$ PCs. There is a tradeoff in that the larger the $K$, the more akin the knockoff will be to the original variables. This results in a smaller type 1 error but weaker power of the test.
3. Compute a residual vector $e_j = (x_j - \hat{x}_j)$. Permute $e_j$ randomly. Denote the permuted vector as $e'_j$.
5. Set $z_j = x_j + e'_j$ and combine it with the current knockoff matrix $Z_{1:j-1}$.

This algorithm was designed in accordance with our knockoff definition. Using linear regression models, the empirical conditional distribution of $x_j$. $L(x_j | X_{-j}, Z_{1:j-1})$ can be estimated. Using permutation, we eliminate the Gaussian assumption. Further, the generated $z_j$ is independent of the response $y$, since $y$ is ignored in our algorithm. Note that this is a sequential algorithm and thus the computational cost could be high. In the study by Shen et al. (2019), the residuals are assumed to be approximately independently and identically distributed, and we retain this assumption for our algorithm. Note that in scenarios with rare events and/or strong class imbalance, the top principal components may not appropriately approximate the variation. In such scenarios, methods like generative adversarial networks (GANs) can be used for data augmentation before generating knockoffs (Frid-Adar et al., 2018).

To evaluate knockoffs generated by these methods, we propose the mean absolute angle of columns (MAAC) metric for checking vector independence and use the kernel maximum mean discrepancy (KMMD) metric to test distribution similarity (Gretton et al., 2007).
For two column vectors with the same length, x and z, we define MAAC(x, z) = arccos[\frac{x^Tz}{\|x\|\|z\|}]. For two matrices with the same dimensions, X and Z, with p columns, we define MAAC(X, Z) = \frac{1}{p} \sum_{j=1}^{p} MAAC(x_j, z_j).

MAAC indicates the correlation between corresponding columns in two matrices. For knockoff variable selection, we know that the weaker the correlation, the more powerful the test. The KMMD metric is used to perform a non-parametric distribution test. The null hypothesis test, H_0, is that the row vectors x_i in X and z_i in Z come from the same distribution. In summary, MAAC is for type 2 error control, and KMMD is for type 1 error control. Section 3.1 describes simulation tests and outlines how permutation tests are used for comparison.

2.3 Variable importance test statistics in tree models

Barber and Candès (2015) suggested the Lasso signed max statistic as a test statistic for variable selection for each original and knockoff variable pair. Candès et al. (2018) later proposed the Lasso coefficient difference statistic for the same purpose. Among the importance of statistics in tree models, SHAP, which is based on game theory and local explanations, has demonstrated the advantage of controlling the FDR. We use the SHAP and Candes\(\text{et al.}\) (2018), we define a test statistic for knockoff variable selection, FDR control, and model-free variable selection with FDR control. We propose a robust multiple-stage KOBT algorithm. Figure 1 is a flowchart that outlines the procedure. The algorithm contains four main steps and one optional step (circled in red): (1) sample knockoff matrix Z according to original matrix X; (2) grow boosted trees using combined predictors (X, Z); (3) calculate the test statistic, \(\phi_j\), for each variable from the fitted boosted tree model, \(f_B(X, Z)\) and (4) conduct steps (1)-(3) \(q\) times and get \(T_j = \frac{1}{q} \sum_{m=1}^{q} \phi_{jm}^2\) for the \(j\)th original variable. A larger positive \(T_j\) indicates a variable is more closely associated with response. The optional step is taken in a scenario in which some variables (shown as W in Fig. 1) need to be retained in the final model. The first part of the algorithm is optional, and a regression model is added before the boosted tree model. Given that we want to retain some covariate variables in our final model, we conduct a regression, \(y_g = Wx + y\), where \(y_g\) is the original responses, \(W\) is the covariate matrix, and \(y\) represents residuals from this optional regression. This \(y\) is treated as new responses for the subsequent boosted tree fitting.

For the first required step of KOBT, we generate knockoff variables Z conditional on original variables X. With the assumption that the data in X follow a Gaussian distribution, the column mean and covariance matrix of X are estimated. As discussed in Section 2.2, there are two choices for covariance matrix estimation: shrinking the estimation to the identity matrix (Openen-Rhein and Strimmer, 2007) and sparse estimation (Bien and Tibshirani, 2011). Without a Gaussian assumption, we propose a strategy based on principal components and permutation (see Section 2.2 for details). Section 3.1 compares the properties of knockoff variables.

During the process of model fitting, hyperparameters \(\gamma\), \(\lambda\) and \(\alpha\) are tuned through Bayesian optimization. Once a boosted model is fitted, test statistics showing importance are calculated. Multiple statistics are considered, such as gain, cover, weight, SHAP value, and Saabas value. Supplementary Section S2 discusses the values of these statistics. The above steps, from generating knockoff variables to calculating test statistics, are repeated \(q\) times to achieve the mean absolute value for each variable, where \(q\) should be sufficiently large. According to the strategy of applying knockoff variables (Barber and Candès, 2015; Candès et al., 2018), for each pair of original and knockoff variables, a test statistic is calculated,

\[
\phi_j^2(\gamma, \lambda, \alpha) + \phi_j^2(\gamma, \lambda, \alpha)
\]

\[
\text{Calculate mean absolute value for each variable}
\]

\[
\frac{1}{q} \sum_{m=1}^{q} |\phi_{jm}|, \quad \frac{1}{q} \sum_{m=1}^{q} |\phi_{jm}^2|
\]

\[
\text{Sample knockoffs } q \text{ times;}
\]

\[
\text{Variable selection}
\]

\[
\text{FDR control}
\]

\[
T_j = \frac{1}{q} \sum_{m=1}^{q} |\phi_{jm}| - \frac{1}{q} \sum_{m=1}^{q} |\phi_{jm}^2|
\]

Fig. 1. Knockoff boosted tree flowchart. An optional step is circled in red.
which is the difference between the mean absolute test statistic values of each variable pair. It is intuitive that a larger test statistic indicates that the original variable is more important than its corresponding knockoff. Finally, the FDR of variable selection is controlled through values of all generated \( T_i \).

We use simulation tests to address the following topics: (i) control of type I and type II errors for knockoffs generated by different methods; (ii) the properties of different variable importance statistics in boosted tree models and (iii) power and false discovery control using various combinations of knockoffs and statistics. Further, we compare the performance of boosted tree fitting for different model structures.

### 3 Simulation studies

**Supplementary Section S1** provides details about boosted tree model fitting using incorrect but related variables. Main-effect, interaction effect, exponential effect and quadratic effect models are defined. **Supplementary Section S2** compares variable importance ranking statistics, namely gain, cover, frequency, SHAP and Saabas.

#### 3.1 Power and type I error control of knockoff variables

As in Section 2.2, we discuss three strategies to generate knockoff variables: Gaussian assumption with shrunk covariance matrix, Gaussian assumption with sparse covariance matrix and permuted residuals from principal component regression. We are interested in evaluating the power and type I error performance of each strategy using different knockoff variables for testing. We apply the MAAC metric, which is defined in Section 2.2, to evaluate the power of each strategy. A higher positive MAAC value shows that two tested variables are more related column space, so the difference between a real signal and its knockoff is more significant. As for type I errors, we use KMMD to test if row vectors in the original design matrix \( X \) and knockoff matrix \( Z \) are drawn from the same distribution. A larger positive test statistic implies that it is more likely that the null hypothesis can be rejected, with the assumption that the two row vectors are from the same distribution. If the knockoff variables are highly different from the original variables, this leads to large type I errors. In other words, we want knockoff variables to be drawn from the same distribution as the original variables but with sufficiently different values. As always, there is a tradeoff between power and type I error control.

We first assume the row vector in design matrix \( X \) is from a normal distribution. We simulate \( n = 100 \) samples and \( p = 500 \) variables. Each row vector in design matrix \( X_{n \times p} \) is generated as \( x_i \sim N_p(0, \Sigma) \) with block dependence structure matrix \( \Sigma = \text{diag}(\Sigma_1, \Sigma_2, \Sigma_3, \ldots) \), where each \( \Sigma_i \) is a \( np \times np \) matrix with matrix element \( \sigma_{ik} = (p^{i-1}) \). We set \( \pi = 0.01 \) and \( \rho = 0.1 \). Figure 2 shows a simulation of 50 original design matrices and the 1000 corresponding knockoff matrices that were generated for each original design matrix. Four kinds of knockoffs were created for comparison: shrunk Gaussian, sparse Gaussian, permuted principal component 10 and permuted principal component 30. The y-axis is calculated as the mean value of each 1000 knockoff samples from one original matrix. The x-axis is the seed used to sample the original matrix. Therefore, y values of different methods at the same x location are comparable. It is clear from Figure 2a and c that all methods are consistent under a normal assumption. The MAAC values in Figure 2a show that the sparse method provides the most similar and thus the least powerful knockoffs. On the other hand, the shrunk Gaussian method has more different knockoffs. This is reasonable since the shrunk Gaussian method directly estimates the covariance matrix of \( X \) and shrinks it to the identity matrix if it is not definitively positive while the sparse Gaussian method has zeros in the estimated matrix given its sparse assumption. Principal component methods provide knockoffs built with multiple principal components from a design matrix. The properties of the knockoffs are thus tunable. If more principal components are used, the generated knockoffs are more similar to the original matrix, with consequences for balancing MAAC and type I errors. Here, 10 and 30 principal components are used as examples in our simulation tests, but any number of principal components can be used given the original design matrix. Figure 2c shows the MAAC plot for each original matrix. Since both the shrunk Gaussian and sparse Gaussian methods assume \( X \) follows a normal distribution, it is interesting to observe what happens when a design matrix does not follow a normal distribution. Figure 2b and d shows that the sparse Gaussian method is less consistent for a Poisson distribution than for a normal distribution. Note that large spikes occur for the sparse Gaussian method at two fixed design matrix seeds for Poisson distribution. This is because of the trade-off between accuracy and complexity in the estimation process for the covariance matrix of the design matrix. Unlike the direct estimation in the shrunk Gaussian method, the trade-off step in the sparse Gaussian method would generate more biased estimated values when the normality assumption is invalid. Biased estimation would improve the power and type I errors shown in Figure 2b and d as fluctuation. However, the principal component method is consistent because it does not depend on a normal assumption. With some exceptions, for most design matrices, the y-axis order of these four curves is the same as for a normal distribution. In summary, the sparse Gaussian method is the most conservative, the shrunk Gaussian method is the most powerful and the principal component method lies between these. When the number of principal components is increased, the performance of the generated knockoff is similar to the performance of the shrunk Gaussian method.

#### 3.2 SEAD, other knockoff test statistics and their false discovery control

This section combines all of the previous steps and demonstrates the whole framework of the KOBT algorithm. We are interested in the power and FDR of various combinations of ranking statistics and knockoff types. We simulate \( n = 100 \) samples and \( p = 500 \) variables. Each row vector in design matrix \( X_{n \times p} \) is generated as \( x_i \sim N_p(0, \Sigma) \) or \( x_i \sim \text{Poisson}_p(5, \Sigma) \) with block dependence structure matrix \( \Sigma = \text{diag}(\Sigma_1, \Sigma_2, \Sigma_3, \ldots) \), where each \( \Sigma_i \) is a \( np \times np \) matrix with matrix element \( \sigma_{ik} = (p^{i-1}) \). We set \( \pi = 0.01 \) and \( \rho = 0.1 \). The maximum depth of each tree is fixed at 6. For simplicity, we choose GBRT as the booster and the main-effect model as the true model. **Supplementary Tables S6–S9** list the power and FDR for different combinations of knockoff types and ranking statistics for comparison. We choose a condition in which signals are sparse and weak so that the performance of each method is considerably distinct.

\[
T_i = \frac{1}{q} \sum_{m=1}^{q} |\phi_{i,m}^y - \frac{1}{q} \sum_{m=1}^{q} |\phi_{i,m}^x|, \tag{2}
\]
Supplementary Table S6 shows the simulation of 50 normally distributed design matrices, each with 1000 shrunk Gaussian knockoffs, 1000 sparse Gaussian knockoffs, 1000 10-principal component knockoffs and 1000 30-principal component knockoffs. We set the targeted FDR as 0.1. The means and corresponding standard errors of power for each combination are listed. Among the four types of knockoffs, the shrunk Gaussian knockoff has the highest power. The same conclusion can be reached from Figure 2a. This is followed by 10- and 30-principal component knockoffs and, finally, the sparse Gaussian knockoff, which has the lowest power. Among the five tested importance statistics, frequency is always the most powerful statistic for all knockoffs while gain is the least powerful. The other three statistics are moderately powerful and fall between frequency and gain. Supplementary Table S7 shows the means and corresponding standard errors of the FDR for each combination. The targeted FDR is 0.1, indicating that the sparse Gaussian knockoff is the most conservative method and is the only one that can ensure the FDR stays under the targeted level. At the same order of power, the shrunk Gaussian knockoff has the highest FDR, followed by the two principal component knockoffs.

Supplementary Table S8 shows the power of the statistics when the normality assumption is invalid. Fifty Poisson-distributed design matrices are simulated, each with 1000 shrunk Gaussian knockoffs, 1000 sparse Gaussian knockoffs, 1000 10-principal component knockoffs and 1000 30-principal component knockoffs. Again, the targeted FDR is set as 0.1. The power of the knockoffs is the same as for a normal distribution. For the importance statistics, the order for Poisson-distributed matrices differs from the order for normally distributed matrices, where cover has the highest power, followed by frequency and then the other three statistics. Both the sparse Gaussian and 30-principal component knockoffs can control the FDR to stay close to the targeted level. The shrunk Gaussian and 10-principal component knockoffs have higher FDRs. The order of importance statistics for FDR is the same as their power ranking. In summary, the power ranking of knockoffs is shrunk Gaussian > 10-principal component > 30-principal component > sparse Gaussian.

4 Real data applications

4.1 Tumor sample purity estimation

Tumor sample purity refers to the percentage of cancer cells in a tumor tissue sample and is used to discover the roles of cancerous and non-cancerous cells in the tumor microenvironment, which mainly comprises immune cells (Aran et al., 2015; Turley et al., 2015). To estimate tumor purity, Carter et al. (2012) proposed ABSOLUTE, a method used to perform analysis of somatic DNA alterations. In addition, it has been reported that DNA methylation data and expression data from selected stromal genes (Houseman et al., 2012; Yoshihara et al., 2013) have been successfully used for estimation. Recently, Aran et al. (2015) and Li et al. (2019) used RNA-seq gene expression data to determine tumor purity and found several gene signatures of individual cancer types. As prior work has shown that it is reasonable to estimate tumor sample purity using gene expression data, we apply our KOBT algorithm so that we can compare our results with those of previous reports. Our interest here is not the accuracy of estimation but rather the selection of genes that are important in estimation with no topology assumptions. If a gene’s expression level is positively correlated with tumor purity, it is highly likely that this gene is expressed primarily by cancer cells in tumor samples.

Processed TCGA RNA-seq gene expression data were downloaded from the Pan-Cancer Atlas Publication website (Hoadley et al., 2018). Among the 33 available tumor types, breast invasive carcinoma (BRCA) and skin cutaneous melanoma (SKCM) were chosen for tumor purity estimation. There are 1017 BRCA and 459 SKCM samples with 17 176 expressed genes for each sample. Genes with missing values or zero variance were filtered out and excluded. We used the tumor purity estimates in the study by Hoadley et al. (2018) as the responses, which were obtained using ABSOLUTE. The responses are thus bounded between 0 and 1.

Supplementary Table S10 reports the selected genes whose expression is related to BRCA tumor purity, where the targeted FDR is 0.1. Supplementary Figure S11 displays the results of hierarchical clustering analysis and a heatmap of the results of gene expression. Supplementary Table S11 lists genes selected by the 10-principal
component knockoff. Among the detected genes, CSF2RB (colony stimulating factor 2 receptor beta common subunit), C15 (complement C1s), CCDC69 (coiled-coil domain containing 69) and FGR (FGR proto-oncogene) are also reported among the top 10-ranked important genes for pan-cancer tumor purity prediction by Li et al. (2019). CSF2RB is an immune-related gene. It has been reported that in BRCA, high expression of CSF2RB is positively correlated with patient survival (Liu et al., 2019). ESTIMATE (Yoshihara et al., 2013) uses stromal and immune gene expression to predict tumor purity. Our detected immune genes CCDC69, FGR and IL7R are included by Yoshihara et al. (2013). Supplementary Figure S2 presents the box plots of gene expression levels for genes that are detected by both types of knockoffs. Other genes are plotted in Supplementary Figure S3. All samples are grouped according to their purity, with samples in the top 1/3 labeled as high purity (blue) and samples in the bottom 1/3 labeled as low purity (yellow). A non-parametric Wilcoxon signed-rank test was conducted for each low–high pair, and the corresponding P-values are shown at the top of the plot. This figure indicates that gene expression levels of almost all selected genes are highly correlated with tumor purity. Samples are also grouped according to their gene expression levels. The top 1/3 of samples is grouped into a high-level group, and the bottom 1/3 is grouped into a low-level group. Survival analysis was conducted for these genes, and CCDC69, CXORF65 and FGR had significant P-values for coefficients in Cox regression. Supplementary Figure S4 includes plots of the Kaplan–Meier estimators and P-values. Supplementary Table S12 reports additional genes associated with tumor purity. Supplementary Figure S12 presents the results of a heatmap and hierarchical clustering analysis of these genes. Li et al. (2019) reported CSF2RB, RHOB, CIS, CCDC69 and CCL22 as among the 10 most important genes for pan-cancer tumor purity prediction. Most of these detected genes are from stromal cells and not cancer cells. For example, the expression level of CCDC69 is negatively correlated with tumor purity, which would not be possible if it mainly comes from cancer cells. Immune genes CCDC69, FGR and RHOB, and stromal gene TXNDC3 are included in ESTIMATE models. Supplementary Figure S5 presents the box plots of gene expression levels for selected genes, with other genes plotted in Supplementary Figure S6. As with BRCA, samples are grouped according to their purity, with the top 1/3 labeled as high purity (blue) and the bottom 1/3 labeled as low purity (yellow). Related P-values from Wilcoxon signed-rank tests are attached to the plots. Gene expression levels of almost all the selected genes are highly correlated with tumor purity. Again, the samples are grouped according to their gene expression levels. Supplementary Figures S7 and S8 present plots of their Kaplan–Meier estimators and P-values. In summary, for tumor purity estimation, we propose that our KOBT algorithm can detect some genes that are largely expressed in stromal cells. Our results reproduce the genes shown to be important in estimating tumor purity using previous methods in pan-cancer analysis, while also detecting new genes important in estimating purity in two of the most difficult to classify tumor types.

4.2 Tumor-type classification

For the application of KOBT for classification, we focus on its performance when assigning tumors to known classes and identifying genes whose expression levels are related to classification. Golub et al. (1999) used gene expression monitored by DNA microarrays to conduct cancer classification. Their test classification of acute myeloid leukemia and acute lymphoblastic leukemia showed that it is feasible to distinguish cancer subtypes based only on gene expression data. Li et al. (2017) performed pan-cancer classification with RNA-seq data from TCGA using the boosted tree and k-nearest neighbours methods. Since the KOBT algorithm can detect signals with FDR control, it is interesting to observe how it works on gene selection for tumor-type classification.

The TCGA RNA-seq gene expression data are the same as that described in Section 4.1. However, instead of pan-cancer classification, we focus on two challenging binary classifications for (i) esophageal carcinoma (ESCA) versus stomach adenocarcinoma (STAD) and (ii) rectum adenocarcinoma (READ) versus colon adenocarcinoma (COAD). We chose these two pairs of cancers because of their similarities. Esophageal carcinoma (ESCA) and stomach adenocarcinoma (STAD) are both malignant tumors in the digestive tract. Li et al. (2017) reported that almost all READ samples were mis-assigned to COAD. We used 499 samples for the ESCA versus STAD classification test and 529 samples for the READ versus COAD classification test. The steps are the same as those described in Section 4.1 with the exception that the responses are binary: 0 for one cancer and 1 for the other cancer. Supplementary Table S13 presents the selected genes that can distinguish ESCA and STAD. Among these genes, BARX1 is a stomach mesenchymal transcription factor. Kim et al. (2005) showed that BARX1 loss in the mesenchyme prevents stomach epithelial differentiation of overlying endoderm and instead induces intestinespecific genes. Their results defined a transcriptional and signaling pathway of inductive cell interactions in vertebrate organogenesis. Kim et al. (2011) proved that BARX1 controls mouse stomach morphogenesis and is required to specify stomach-specific epithelium in adjacent endoderm. HAND2 is an RNA gene that is affiliated with the long non-coding RNA class. Tsukobawa et al. (2002) stated that KRT14 is often expressed by tumor cells in the trabecular nests of the primary carcinoma. Supplementary Figure S9 presents the box plots of gene expression levels for all the detected genes. All samples are grouped according to their cancer types, with ESCA samples in blue and STAD samples in yellow. A non-parametric Wilcoxon signed-rank test was conducted for each gene, and the corresponding P-values are included at the top of the plot. This figure indicates that gene expression levels of all the selected genes are highly correlated with cancer type.

Supplementary Table S14 shows the detected genes that can be used for READ and COAD classification. HOXC4 and HOXC8 are homeobox genes, which comprise a large family of transcription factors that direct the formation of many bodily structures during early embryonic development. It has been observed that HOX family gene expression is upregulated in most solid tumor types (Bhatlekar et al., 2014). Supplementary Figure S10 shows the box plots of gene expression levels for all detected genes. All samples are grouped according to their cancer type, with COAD samples in blue and READ samples in yellow. A non-parametric Wilcoxon signed-rank test was conducted for each gene, and the corresponding P-values are included at the top of the plot. This figure indicates that gene expression levels of all selected genes are highly correlated with cancer type.

Compared to tumor purity estimation, cancer genes play a more important role in tumor-type classification. We detected some promising new cancer genes and previously published literature supports our findings for both classification tests. Given the fact that the mechanism of cancers is complicated, and the cancer types we chose have been reported to be difficult to distinguish, we show our proposed KOBT algorithm to be robust for real data with unknown models.

5 Conclusions

This study presents several important methodological advances. First, we introduce KOBT, a new model-free variable selection method. Given the nature of boosted tree models, no prior model topology knowledge is required. This method extends the application of knockoff methods to highly successful tree-based models. Additionally, we propose two new sampling methods to generate knockoffs, namely the PCC knockoff and the sparse Gaussian knockoff. Unlike currently available methods, the PCC knockoff does not depend on Gaussian assumptions for the design matrix. Additionally, to evaluate the power of our generated knockoffs in simulation tests, we define a new test statistic, mean absolute angle of columns, which represents the average distance between column vectors in two matrices. We apply kernel maximum mean discrepancy to test whether our proposed knockoffs are drawn from the same distributions as the original vectors. In our boosted tree framework, we fix most hyperparameters at multiple reasonable levels and leave regularization parameters to be tuned by Bayesian optimization. We consider different importance scores in tree models.
We test the results, which are based on multiple combinations of different knockoffs and importance test statistics, for their impact on overall performance. This can help users of the method make appropriate parameter choices. We outline the simulation experiments that we conducted to test KOBT on a range of models, including main effect, interaction, exponential and second-order models. Finally, we apply our algorithm for tumor purity estimation and tumor-type classification using TCGA gene expression data. KOBT performs well in this application, and our results both recapitulate previously published literature and reveal additional genes. Finally, we provide an R package to implement our proposed approaches, which is available at https://cran.r-project.org/web/packages/KOBT/index.html.

Funding

This research was supported by funds from the Intramural Research Program of the NIH, National Institute of Environmental Health Sciences.

Conflict of Interest: none declared.

Data availability

Data used is publicly available through the GEO accession number GSE107650.

References

Aran,D. et al. (2015) Systematic pan-cancer analysis of tumour purity. Nat. Commun., 6, 8971.
Barber,R.F. and Candès,E.J. (2015) Controlling the false discovery rate via knockoffs. Ann. Stat., 43, 2055-2085.
Bhatkhande.S et al. (2014) Hox genes and their role in the development of human cancers. J. Mol. Med., 92, 811-823.
Bien,J. and Tibshirani,R.J. (2011) Sparse estimation of a covariance matrix. Biometrika, 98, 807-820.
Breneman,L. (2001) Random forests. Mach. Learn., 45, 5-32.
Candes,E. et al. (2018) Panning for gold: ‘model-x’ knockoffs for high dimensional controlled variable selection. J. R. Stat. Soc. Ser. B (Stat. Methodol.), 80, 531-577.
Carter,S.L. et al. (2012) Absolute quantification of somatic dna alterations in human cancer. Nat. Biotechnol., 30, 413-421.
Chen,T. and Gaither,C. (2016) XGBoost: a scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, San Francisco, California, pp. 785-794.
Chung,E. and Romano,J.P. (2013) Exact and asymptotically robust permutation tests. Ann. Stat., 41, 484-507.
Friedman,M.A. et al. (2018) Synthetic data augmentation using gan for improved liver lesion classification. In 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018). IEEE, San Francisco, California, pp. 289-293.
Friedman,J. et al. (2000) Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). Ann. Stat., 28, 337-407.
Friedman,J.H. (2001) Greedy function approximation: a gradient boosting machine. Ann. Stat., 29, 1189-1232.
Golub,T.R. et al. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science, 286, 531-537.
Gretton,A. et al. (2007) A kernel method for the two-sample-problem. Adv. Neural Inf. Process. Syst., 513-520.
Hastie,T. et al. (2005) The elements of statistical learning: data mining, inference and prediction. Math. Intell., 27, 83-85.
Hoadley,K.A. et al. (2018) Cell-of-origin patterns dominate the molecular classification of 10,000 tumors from 33 types of cancer. Cell, 173, 291-304.
Hosea,E.A. et al. (2012) DNA methylation arrays as surrogate measures of cell mixture distribution. BMC Bioinformatics, 13, 86.
Kim,B.-M. et al. (2005) The stomach mesenchymal transcription factor Barx1 specifies gastric epithelial identity through inhibition of transient wnt signalling. Dev. Cell, 8, 611-622.
Kim,B.-M. et al. (2011) Regulation of mouse stomach development and Barx1 expression by specific miRNAs. Development, 138, 1081-1086.
Kim,H. and Loh,W.-Y. (2001) Classification trees with unbiased multitree splits. J. Am. Stat. Assoc., 96, 589-604.
LeCun,Y. et al. (1995) Convolutional networks for images, speech, and time series. In: Arib, M.A. (ed.) The Handbook of Brain Theory and Neural Networks, Vol. 3361. MIT Press, Boston, MA.
Li,L. et al. (2005) Model-free variable selection. J. R. Stat. Soc. Ser. B (Stat. Methodol.), 67, 285-299.
Li,Y. et al. (2017) A comprehensive genomic pan-cancer classification using the cancer genome atlas gene expression data. BMC Genomics, 18, 508.
Li,Y. et al. (2019) Putative biomarkers for predicting tumor sample purity based on gene expression data. BMC Genomics, 20, 1-12.
Liu,Y. et al. (2019) Bioinformatic identification of miR-622 key target genes and experimental validation of the miR-622-RNF8 axis in breast cancer. Front. Oncol., 9, 1114.
Loh,W.-Y. (2002) Regression trees with unbiased variable selection and interaction detection. Stat. Sin., 12, 361-386.
Lundberg,S.M. et al. (2018) Consistent individualized feature attribution for tree ensembles. arXiv preprint arXiv: 1802.03888.
Lundberg,S.M. and Lee,S.-L. (2017a) Consistent feature attribution for tree ensembles. arXiv preprint arXiv: 1706.06060.
Lundberg,S.M. and Lee,S.-L. (2017b) A unified approach to interpreting model predictions. Adv. Neural Inf. Process. Syst., 4765-4774.
Miller,P.J. et al. (2016) Finding structure in data using multivariate tree boosting. Psychol. Methods, 21, 583-602.
Nielsen,D. (2016) Tree Boosting with XGBoost. NTNU Norwegian University of Science and Technology, Norway.
Oppeg-Rhine,R. and Strimmer,K. (2007) Accurate ranking of differentially expressed genes by a distribution-free shrinkage approach. Stat. Appl. Genet. Mol. Biol., 6.
Ratsch,G. et al. (2001) Soft margins for adaboost. Mach. Learn., 42, 287-320.
Schuur,F. et al. (2018) Using regression tree ensembles to model interaction effects: a graphical approach. Appl. Econ., 50, 6341-6354.
Shen,A. et al. (2019) False discovery rate control in cancer biomarker selection using knockoffs. Cancers, 11, 744.
Snook,J. et al. (2012) Practical bayesian optimization of machine learning algorithms. Adv. Neural Inf. Process. Syst., 2951-2959.
Song,Y.-Y. and Ying,L. (2015) Decision tree methods: applications for classification and prediction. Shanghai Arch. Psychiatry, 27, 130-135.
Su,X. et al. (2011) Interaction trees: exploring the differential effects of an intervention programme for breast cancer survivors. J. R. Stat. Soc. Ser. C (Appl. Stat.), 60, 457-474.
Tibshirani,R. (1996) Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B (Stat. Methodol.), 50, 634-654.
Turley,S.J. et al. (2015) Functional hallmarks of stromal cells in the tumour microenvironment. Nat. Rev. Immunol., 15, 669-682.
Yoshihara,K. et al. (2013) Inferring tumour purity and stromal and immune cell admixture from expression data. Nat. Commun., 4, 2612.
Zhang,T. and Yu,B. (2005) Boosting with early stopping: convergence and consistency. Ann. Stat., 33, 1538-1579.
Zou,H. and Hastie,T. (2005) Regularization and variable selection via the elastic net. J. R. Stat. Soc. Ser. B (Stat. Methodol.), 67, 301-320.