Knockoffs filter and its extensions

Heqi Yin*

December 7, 2023

Contents

*This is report for Topics in Knockoff Filter.

1 Introduction

1.1 Variable Selection and false discovery rate

Variable selection is concerned in the regression setting. We are given a set of sampling data $\mathcal{D} = (X, Y)$, where the co-variances X is a n-by-p matrix and the respond variable Y is a vector of n dimension. It is assumed they are sampled n times from the true model $y = f(x) + \epsilon$, where $x = (x_1, \ldots, x_p) \in \mathbb{R}^p$, $y \in \mathbb{R}$ and ϵ is white noise. Here the function $f(x)$ is unknown and the target is to develop a model based on $\mathcal{D}$ that can predict f on a test data set other than D while minimizing the risk.

Classical linear regression reveals the bias-variances trade-off problem — the more variables we select, the more precise we can reach on the training data $\mathcal{D}$, but also the more variance we introduced to the model, and thus more variables may not have better performance on prediction. Variable selection is to address this trade-off by choosing a subset of the current features aiming at having better prediction precision. Besides the root mean square derivation as a common criteria for variable selection, the false discovery rate is also of great importance in many scenario.

Denote the true model by the set S of indices j that f is a function of x_j and denote $\hat{S}$ by a model driven from $\mathcal{D}$. The false discovery rate is the size of false discovery, which is the features in the model $\hat{S}$ but not in the true model, over the size of $\hat{S}$. It is often desired that a variable selection model cam control the FDR within a preset threshold q.

1.2 Knockoffs and its extensions

Knockoffs filters is a variable selection method which guarantees control of false discovery rate. In our report, we will discussion the classical knockoff filters defined in [Barber and Candès, 2015] with fixed design matrix, the Model-X knockoff filters introduced in [Candes et al., 2018] with random design matrix and dimension extension, and we will introduce deep knockoffs [Romano et al., 2020] as a way to construct knockoff matrices using deep neural networks. We also covered discussion on robust inference with knockoffs in [Barber et al., 2019].

2 Knockoffs filters with Fix-X design

Consider linear model $Y = X\beta + \epsilon$, where ϵ are Gaussian noises, $X = (X_1, \ldots, X_p)$ is a fixed design matrix and β is the parameter. Denote the true model by $S = \{j \mid \beta_j \neq 0\}$ and the model of interest by $\hat{S} = \#\left\{j]mid\hat{\beta}_j \neq 0\right\}$. In this session, we will introduce the classic knockoff filters in the case $n > 2p$, which can be generalized to $n > p$.

Our goal is to control the false discover rate under a preset threshold q, in this setting,

$$\text{FDR} = \frac{\#(\hat{S} - S)}{\#\hat{S}} = \frac{\#\{j \mid \hat{\beta}_j \neq 0 \text{ and } \beta_j = 0\}}{\#\{j \mid \hat{\beta}_j \neq 0\}}.$$

2.1 Knockoff Matrix

We define the knockoff matrix $\tilde{X} = (\tilde{X}_1, \ldots, \tilde{X}_p)$ of X, as a matrix of the same size, by the following axioms:

1. $\tilde{X}$ recovers the correlations of X :

$$\tilde{X}^T\tilde{X} = X^T X, \text{ i.e } \langle X_i, X_j \rangle = \left\langle \tilde{X}_i, \tilde{X}_j \right\rangle$$

2. feature X_j and knockoff feature $\tilde{X}_j$ are not identical

3. cannot distinguish X and $\tilde{X}$ using inner product: $\left\langle \tilde{X}_i, X_j \right\rangle = \langle X_i, X_j \rangle$ for $i \neq j$.

Denote $\Sigma = X^T X$, the above three conditions can be written as

$$G = \begin{pmatrix} X^T X & X^T \tilde{X} \\ \tilde{X}^T X & \tilde{X}^T \tilde{X} \end{pmatrix} = \begin{pmatrix} \Sigma & \Sigma - \operatorname{diag}\{s\} \\ \Sigma - \operatorname{diag}\{s\} & \Sigma \end{pmatrix}, \tag{1}$$

where $\operatorname{diag}\{s\}$ is a diagonal matrix from a vector $s \in \mathbb{R}^p$ with positive entries.

One concrete construction of the knockoff matrix when $\operatorname{diag}\{s\} \preccurlyeq 2\Sigma$ is

$$\tilde{X} = X(I - \Sigma^{-1} \operatorname{diag}\{s\}) + \tilde{U}C,$$

where $\tilde{U}$ has orthogonal columns and orthogonal to columns of X, and C is the Cholesky decomposition, that is, C satisfies $C^T C = 2 \operatorname{diag}\{s\} - \operatorname{diag}\{s\}\Sigma^{-1} \operatorname{diag}\{s\}$.

2.2 Knockoff Statistic

Base on the knockoff matrix $\tilde{X}$, we now create knockoff statistic $W = (W_1, \ldots, W_p)$ that have the following properties:

- sufficient: W is a function of G in eq. (1), and $[X\tilde{X}]^T Y$.
- anti-symmetric : If we swap X_j with $\tilde{X}_j$ in $[X\tilde{X}]$, then W_j will flip a sign while other W_j remains the same. If swapping a subset $I \subset 1, 2, \ldots, n$, then W_j will flip sign iff $j \in I$.

There are many constructions of W_j that satisfy these properties and the choice of W_j effects the power of our discover rate (since we are controlling the Type I error). The initial work in [Barber and Candès, 2015] does not explicitly talk about which W is ideal and proposes it as an open problem.

As an example we will use throughout this session, we'll consider the Lasso regression problem with augmented design matrix $[X\ \tilde{X}]$, i.e.

$$Y = [X\ \tilde{X}]b + \epsilon$$

Lasso estimator $\hat{b}$ satisfies $\hat{b}(\lambda) = \arg\min_b \frac{1}{2} \left\| y - [X\tilde{X}]b \right\|_2^2 + \lambda \|b\|_1$. As λ increases more features will disappear from the model, let

$$\begin{aligned} Z_j &= \sup\{\lambda \colon \hat{b}_j(\lambda) \neq 0\}, \quad \text{for } j = 1 \ldots p, \\ \tilde{Z}_j &= \sup\{\lambda \colon \hat{b}_{j+p}(\lambda) \neq 0\}, \quad \text{for } j = 1 \ldots p. \end{aligned}$$

Note Z_j measures how long feature i survived. $Z_j > \tilde{Z}_j$ indicates X_j is more likely to be a true feature. We should pick a statistic W_j for each feature such that high value of W_j indicates X_j is a true feature. Hence we define

$$W_j = \begin{cases} Z_j \vee \tilde{Z}_j, & \text{if } Z_j > \tilde{Z}_j, \\ 0, & \text{if } Z_j = \tilde{Z}_j, \\ -Z_j \vee \tilde{Z}_j, & \text{if } Z_j > \tilde{Z}_j, \end{cases}$$

where $\vee$ represents the maximum. The knockoff statistic W here satisfies the two properties, since W is obtained from the lasso model and W is defined anti-symmetric.

Here are some other constructions of W_j that satisfies the two properties:

1. $W_j = X_j^T Y - \tilde{X}_j^T Y$,
2. $W_j = \hat{\beta}_j^{OLS} - \hat{\beta}_{j+p}^{OLS}$, using the augmented design $[X\tilde{X}]$,
3. $W_j = Z_j - \tilde{Z}_j$, Z_j and $\tilde{Z}_j$ means the same as above.

2.3 FDR Control

Based on the knockoff statistic W, we will pick a feature X_j if W_j exceed some threshold t. Explicitly, the selected model is $\hat{S}(t) = \{j \mid W_j \geq t\}$. To control the true FDR, we need an estimator of the false discovery. Here we choose $\#\{j \mid W_j \leq -t\}+1$. It will be clear soon why this works. Therefore the estimator of FDR with threshold t is

$$\widehat{\mathrm{FDR}}(t) = \frac{\text{false discovery size estimator}}{\#\hat{S}(t)} = \frac{\#\{j \mid W_j \leq -t\} + 1}{\#\hat{S}(t)}$$

The smaller t is, the more feature we choose. We want t to be as large as possible to reduce Type II error while controlling the Type I error, i.e. FDR, hence set the knockoff threshold

$$T = \min\left\{t \in \mathcal{W} \mid \widehat{\mathrm{FDR}}(t) \leq q\right\},$$

where $\mathcal{W} = \{W_j \mid j = 1 \dots p\}$. The following key theorem shows why the true FDR is under control.

Theorem 1. *Chose a threshold T by setting*

$$T = \min\left\{t \in \mathcal{W} : \frac{1 + \#\{j : W_j \leq -t\}}{\#\{j : W_j \geq t\} \vee 1} \leq q\right\}$$

where q is the target FDR level. Then setting $\hat{S} = \{j : W_j \geq T\}$, controls the FDR

$$\mathbb{E}\left[\frac{|\{j : j \in \hat{S}, \beta_j = 0\}|}{|\hat{S}| \vee 1}\right] \leq q$$

Lemma 2. *Let $\varepsilon \in \{\pm 1\}^p$ with $\varepsilon_j = 1$ for all non-null j and $\varepsilon_j \overset{i.i.d.}{\sim} \{\pm 1\}$ for null j. Then*

$$(W_1, \dots, W_p) \overset{d}{=} (\varepsilon_1 W_1, \dots, \varepsilon_p W_p)$$

Proof. For the lemma, the anti-symmetry property gives

$$W_{\mathrm{swap}(S)} = (\varepsilon_1 W_1, \dots, \varepsilon_p W_p), \qquad \varepsilon_j = \begin{cases} +1 & j \notin S \\ -1 & j \in S \end{cases}$$

On the other hand, since $[X\tilde{X}]^T[X\tilde{X}] = [X\tilde{X}]^T_{\mathrm{swap}(S)}[X\tilde{X}]_{\mathrm{swap}(S)}$ and for $S \subset \{j : \beta_j = 0\}$ we have $[X\tilde{X}]^T y \overset{d}{=} [X\tilde{X}]^T_{\mathrm{swap}(S)} y$, we have

$$W_{\mathrm{swap}(S)} \overset{d}{=} W$$

□

Proof. Without loss of generality we assume that $|W_1| \geq \cdots \geq |W_p|$. Then our t value in question can be found by testing $t = |W_i|$ starting with $i = p$ and decreasing. This results in T being a stopping time for the supermartingale that is given by

$$\frac{\#\{j : \beta_j = 0, W_j \leq -t\}}{1 + \#\{j : \beta_j = 0, W_j \geq t\}}$$

The expected value of this supermartingale at the random time $t = T$ is bounded by its expected value at time $t = 0$.

$$\begin{aligned}\mathbb{E}\left[\frac{\#\{j : \beta_j = 0, W_j \leq -T\}}{1 + \#\{j : \beta_j = 0, W_j \geq T\}}\right] \leq &\mathbb{E}\left[\frac{\#\{j : \beta_j = 0, W_j \leq 0\}}{1 + \#\{j : \beta_j = 0, W_j \geq 0}\right] \\ = &\mathbb{E}\left[\frac{Y}{1 + \#\{j : \beta_j = 0\} - Y}\right]\end{aligned}$$

Where $Y = \#\{j : \beta_j = 0, W_j \leq 0\}$. Since we have $Y \sim \text{Binom}(\#\{j : \beta_j = 0\}, 1/2)$ we let $n = \#\{j : \beta_j = 0\}$

$$\mathbb{E}\left[\frac{Y}{1 + \#\{j : \beta_j = 0\} - Y}\right] = \frac{1}{2^n}\sum_{k=0}^{n} \frac{k}{1 + n - k}\binom{n}{k} = \frac{1}{2^n}\sum_{k=0}^{n}\binom{n}{k-1} = 1 - \frac{1}{2^n}$$

Giving us that

$$\mathbb{E}\left[\frac{\#\{j : \beta_j = 0, W_j \leq -T\}}{1 + \#\{j : \beta_j = 0, W_j \geq T\}}\right] \leq 1$$

With this bound in place, we have

$$\begin{aligned}\text{FDR} = &\mathbb{E}\left[\frac{\#\{j : \beta_j = 0, W_j \geq T\}}{\#\{j : W_j \geq T\} \vee 1}\right] \\ \leq &\mathbb{E}\left[\frac{\#\{j : \beta_j = 0, W_j \geq T\}}{1 + \#\{j : \beta_j = 0, W_j \leq -T\}} \cdot \frac{1 + \#\{j : W_j \leq -T\}}{\#\{j : W_j \geq T\} \vee 1}\right] \\ \leq &\mathbb{E}\left[\frac{\#\{j : \beta_j = 0, W_j \geq T\}}{1 + \#\{j : \beta_j = 0, W_j \leq -T\}} \cdot q\right] \\ \leq &q\end{aligned}$$

□

3 Model X

3.1 Knockoff Matrix & Statistics

For the case $n < p$ we cannot use the above Fixed-X paradigm since we cannot find $\tilde{U}$ as specified in equation (2.1). Instead we'll used the Model-X paradigm as described in [Candes et al., 2018]. In this paradigm the following restrictions are introduced or relaxed.

- We expand our range of dimensions to allow any p.
- We assume that we know the covariate distribution, F_X.

- We relax the condition of $Y = X\beta + \varepsilon$ to allow $F_{Y|X}$ to be arbitrary and unknown. (We generalize $\{j : \beta_j = 0\}$ to $\mathcal{H}_0$)
- We also relax the restriction on w_j depending only on $[X, \tilde{X}]^T[X, \tilde{X}]$ and $[X, \tilde{X}]^T y$. Instead we only require the sign switching property

$$w_j([X, \tilde{X}]_{\text{swap(S)}}, y) = \begin{cases} w_j([X, \tilde{X}], y) & j \notin S \\ -w_j([X, \tilde{X}], y) & j \in S \end{cases}$$

Unlike above, we create $\tilde{X}$ in a random way. Since we know the full covariate distribution, we can iteratively sample $\tilde{X}_j$ from $\mathcal{L}(X_j|X_{-j}, \tilde{X}_{1:j-1})$ where $X_{-j} = \{x_1, \ldots, X_p\} - \{X_j\}$. This results in knockoff variables such that

$$(X, \tilde{X})_{\text{swap}(S)} \stackrel{d}{=} (X, \tilde{X}) \qquad \tilde{X} \perp Y|X \tag{2}$$

We then create our statistics W_j similar to above. Like before, we have the following mirror of Lemma 2.

Lemma 3. *Conditional on $(|W_1|, \ldots, |W_p|)$, the signs of the null W_j's, $j \in \mathcal{H}_0$, are i.i.d. coin flips.*

Proof. Let $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_p)$ be a sequence of independent random variables such that $\varepsilon_j = \pm 1$ with probability $1/2$ if $j \in \mathcal{H}_0$, and $\varepsilon_j = 1$ otherwise. It suffices to establish that

$$W \stackrel{d}{=} (\varepsilon_1 W_1, \ldots, \varepsilon_p W_p)$$

Allow $S = \{j : \varepsilon_j = -1\} \subset \mathcal{H}_0$. We see from the flip-sign property that $W_{\text{swap}(S)} = (\varepsilon_1 W_1, \ldots, \varepsilon_p W_p)$. On the other hand, since $([X, \tilde{X}], y) \stackrel{d}{=} ([X, \tilde{X}]_{\text{swap}(S)}, y)$ for $S \subset \mathcal{H}_0$, we have $W_{\text{swap}(S)} \stackrel{d}{=} W$. □

3.2 FDR Control

This gives us the same method of FDR control extending Theorem 1.

Theorem 4. *Chose a threshold $\tau > 0$ by setting*

$$\tau = \min\left\{t > 0 : \frac{1 + \#\{j : W_j \leq -t\}}{\#\{j : W_j \geq t\}} \leq q\right\}$$

where q is the target FDR level. Then setting $\hat{S} = \{j : W_j \geq \tau\}$, controls the FDR

$$\mathbb{E}\left[\frac{|\{j \in \hat{S} \cap \mathcal{H}_0\}|}{|\hat{S}| \wedge 1}\right] \leq q$$

The proof is the same as Theorem 1 using Lemma 3 for the signs of W_j being i.i.d. coinflips.

4 Deep Knockoffs

From the above sections, we see that Model-X knockoffs can be used to identify a subset of important variables from a larger pool that could potentially explain a phenomenon under study while rigorously controlling the false discovery rate [Benjamini and Hochberg, 1995] in very complex statistical models.

Why we care the Deep Knockoffs method? We want to extend the applicability of the knockoffs framework as to make it practically model-free and, therefore, widely applicable. In words, we want to extend it to the models that we do not have reliable prior knowledge about the distribution of the covariates but perhaps sufficiently many labeled or unlabeled samples to "learn" this distribution to a suitable level of approximation.

In general, the procedure is: taking the data X as input and generates $\tilde{X}$ through a mapping $f_\theta(X, V)$, where V is random noise, and f_θ is a deep neural network. The main idea is to iteratively refine a knockoff sampling mechanism until a criterion measuring the validity of the produced knockoffs is optimized; this criterion is inspired by the popular maximum mean discrepancy in machine learning and can be thought of as measuring the distance to pairwise exchangeability between original and knockoff features. By building upon the existing model-X framework, it thus obtains a flexible and model-free statistical tool to perform controlled variable selection.

4.1 Recap of Model-X Knockoffs

We have introduced the basic idea of knockoff filter. To help understand Deep Knockoffs, we need to emphasize a few points in this section. Recall the each pair of Z_j and $\tilde{Z}_j$ can be chosen to be combined through an antisymmetric function into the statistics W_j, e.g. $W_j = Z_j - \tilde{Z}_j$. If W_j is large and positive, it can be shown that exact control of the false discovery rate below the nominal level q can be obtained by selecting $\hat{\mathcal{S}} = \{j : W_j \geq \tau_q\}$, where

$$\tau_q = \min\left\{t > 0 : \frac{1 + |\{j : W_j \leq -t\}|}{|\{j : W_j \geq t\}|} \leq q\right\}. \tag{3}$$

This is an adaptive significance threshold since the numerator above can be regarded as a conservative estimate of the number of false positives at the fixed level t [Barber and Candès, 2015]. And notice that the choice of the test statistics W_j may be different [Candes et al., 2018].

4.2 Second-Order Knockoff

A general question arises, how to construct a good copy of X, i.e., $\tilde{X}$. Recall that We have talked about an example for X is Gaussian distributions, but it is usually not the case in real life. Therefore, we introduce deep knockoff machine, which can help us get $\tilde{X}$, which benefits from deep generative models. A flow chart of simplest knockoff machine is shown in 4.2. We can then summarize the algorithm briefly as

1. **Input:**
   - n realizations of a random vector X independently sampled from an unknown distribution P_X.
   - Independent noise vectors $V^i \sim \mathcal{N}(0, I)$ for each $i \in \{1, \ldots, n\}$.
2. **Knockoff Generation:**

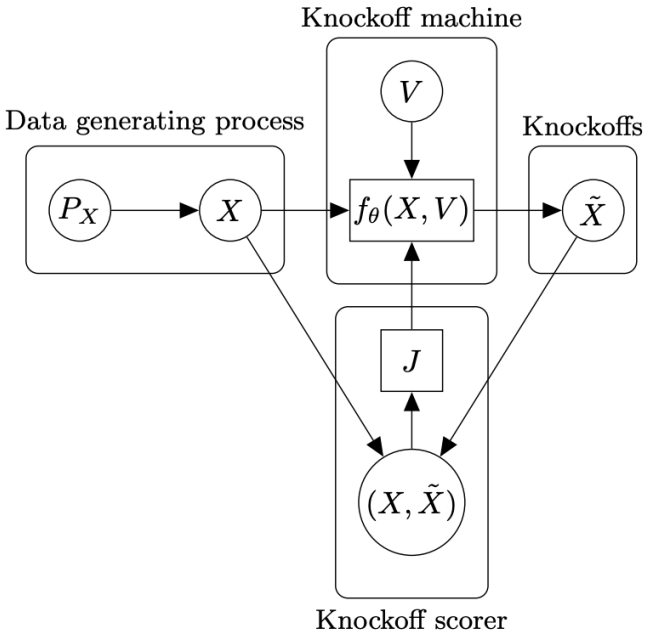

Figure 1: Second-Order Deep Knockoff Machine

- For each observation i, compute the knockoff copy $\tilde{X}^i = f_\theta(X^i, V^i)$ using the random mapping f_θ characterized by parameters θ.
- The noise vector V^i is independently resampled for each observation and each time the machine is called.

3. **Scoring Function** J**:** Examine the empirical distribution of $(X, \tilde{X})$ and quantify its compliance with the exchangeability property. i.e., $(X, \tilde{X}) \overset{d}{=} (X, \tilde{X})_{\text{swap}(j)}$ for each $j \in \{1, \ldots, p\}$
4. **Parameter Update:** After each iteration, update the parameters θ in an attempt to improve future scores.
5. **Output:** Ideally, after sufficient iterations, the machine should be able to generate high-quality knockoff copies $\tilde{X}$ for new observations of X drawn from the same distribution P_X.

Definition 1 (Second-Order)**.**

$$J_{\textit{second-order}}\left(\mathbf{X}, \tilde{\mathbf{X}}\right) = \lambda_1 \frac{\left\| \frac{1}{n} \sum_{i=1}^{n} \left(X^i - \tilde{X}^i \right) \right\|_2^2}{p} + \lambda_2 \frac{\left\| \hat{G}_{XX} - \hat{G}_{\tilde{X}\tilde{X}} \right\|_F^2}{\left\| \hat{G}_{XX} \right\|_F^2} + \lambda_3 \frac{\left\| M \circ \left(\hat{G}_{XX} - \hat{G}_{X\tilde{X}} \right) \right\|_F^2}{\left\| \hat{G}_{XX} \right\|_F^2}, \tag{4}$$

where symbol $\circ$ *indicates element-wise multiplication,* $M = E - I \in \mathbb{R}^{p \times p}$*, with* E *being a matrix of one and* I *the identity matrix.*

For simplicity, the weights $\lambda_1, \lambda_2, \lambda_3$ will can set equal to one in our setting. The first term in (4) penalizes differences in expectation, while the second and third terms encourage the matching of the second moments. Naturally, smaller values of this loss function suggest that $\tilde{X}$ is a better second-order approximate knockoff copy of X. Since J is smooth, a second-order knockoff machine can be trained with standard techniques of stochastic gradient descent. Since knockoff filter may

not be unique. And our goal is to make $\tilde{X}$ as different as possible from X. Thus, the machine might have many different solutions to our problem. Adapted from [Candes et al., 2018], we can introduce a penalty term into the loss function as:

Definition 2 (Regularization term)**.**

$$J_{decorrelation}\left(\mathbf{X}, \tilde{\mathbf{X}}\right) = \sum_{j=1}^{p} \widehat{\operatorname{corr}}\left(X_j, \tilde{X}_j\right), \tag{5}$$

where $\widehat{\operatorname{corr}}\left(X_j, \tilde{X}_j\right)$ *indicates empirical estimate of Pearson correlation coefficient for the* j*th columns of* $\mathbf{X}$ *and* $\tilde{\mathbf{X}}$*.*

To summarize, we introduced a new general procedure for sampling approximate second-order knockoffs by using neural networks.

4.3 Higher-Order Knockoff

Compared to [Candes et al., 2018], we may have a question that the computational cost is lot in our neural network generating method. In this section, we will tend to explain why this design is valuable.

Before we move into the section of introducing the deep knockoff machine, we first determine the loss function, i.e., the metric we use to measure the difference in distribution between $(X, \tilde{X})$ and $(X, \tilde{X})_{\text{swap}(j)}$ for each $j \in \{1, \ldots, p\}$. Inspired by [Gretton et al., 2012], we consider a test statistic, called the maximum mean discrepancy, which has led to the development of generative moment matching algorithms [Li et al., 2015], [Dziugaite et al., 2015].

Definition 3. *Let* X, X', Z, Z' *be independent samples drawn from* P_X *and* P_Z*, respectively, and define the maximum mean discrepancy between* P_X *and* P_Z *as*

$$\mathcal{D}_{\text{MMD}}\left(P_X, P_Z\right) = \mathbb{E}_{X,X'}\left[k\left(X, X'\right)\right] - 2\mathbb{E}_{X,Z}[k(X, Z)] + \mathbb{E}_{Z,Z'}\left[k\left(Z, Z'\right)\right], \tag{6}$$

where k *is a kernel function.*

The quantity in (6) can be shown to be zero if and only if $P_X = P_Z$ if the characteristic kernel of a reproducing kernel Hilbert space (RKHS) is used. In our case, we are able to choose Gaussian kernel function, i.e.,

$$k\left(X, X'\right) = \exp\left\{-\left\|X - X'\right\|_2^2 / \left(2\xi^2\right)\right\}, \tag{7}$$

where ξ is bandwidth of kernel. Gaussian kernels are valid since expanding into a power series allows one to characterize (6) as the distance between vectors containing all higher-moments. Also, by [Gretton et al., 2012], the maximum mean discrepancy is always non-negative and it can be estimated from finite samples $\mathbf{X}, \mathbf{Z} \in \mathbb{R}^{n \times p}$ in an unbiased formula via

$$\widehat{\mathcal{D}}_{\text{MMD}}(\mathbf{X}, \mathbf{Z}) = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} k\left(X^i, X^j\right) - \frac{2}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} k\left(X^i, Z^j\right) + \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} k\left(Z^i, Z^j\right), \tag{8}$$

In practice, since (8) is computationally feasible, it can be used to be the objective function during training the deep generative model. For example, the worst-case perspective in the robustness theory may suggest the generative adversarial networks. Thus, we are allowed to replace the discriminator

effectively by (8). In words, by applying gradient descent techniques to $\mathbf{X}$, the generator produces samples $\mathbf{Z}$ that minimize (8). After determining the loss function, we then back to the algorithm in deep knockoff machine. In general, the algorithm is similar as second-order knockoff machine except we should define more hyperparameters in deep knockoff setting:

1. **Input:**
   - $\mathbf{X} \in \mathbb{R}^{n \times p}$ - Training data.
   - $V^i \sim \mathcal{N}(0, I)$ for each $i \in \{1, \ldots, n\}$ - Independent noise vectors.
   - γ - Higher-order penalty hyperparameter.
   - λ - Second-order penalty hyperparameter.
   - δ - Decorrelation penalty hyperparameter.
   - θ_1 - Initialization values for the weights and biases of the network.
   - μ - Learning rate.
   - T - Number of iterations.
2. **Output:** f_{θ_T} – A knockoff machine.

We can then illustrate the training procedure as:

Algorithm 1 Training a deep knockoff machine

for $t = 1$ to T **do**
  Sample the random noises: $V_i \sim \mathcal{N}(0, I), \forall 1 \leq i \leq n$;
  Randomly divide X into two disjoint mini-batches X', X'';
  Pick a subset of swapping indices $S \subseteq \{1, \ldots, p\}$ uniformly at random;
  Generate the knockoffs as a deterministic function of θ: $\tilde{X}_i = f_\theta(X'_i, V_i), \forall 1 \leq i \leq n$;
  Evaluate the loss function, using the batches and swapping indices fixed above:
  $J_{\theta_t}(X, \tilde{X}) = \gamma J_{MMD}(X, \tilde{X}) + \lambda J_{\text{second-order}}(X, \tilde{X}) + \delta J_{\text{decorrelation}}(X, \tilde{X})$;
  Compute the gradient of $J_{\theta_t}(X, \tilde{X})$, which is now a deterministic function of θ:$\nabla_\theta J_{\theta_t}(X, \tilde{X})$;
  Update the parameters: $\theta_{t+1} = \theta_t - \mu \nabla_\theta J_{\theta_t}(X, \tilde{X})$;
end for

Note that we randomly split the data into a partition $\mathbf{X}', \mathbf{X}'' \in \mathbb{R}^{n/2 \times p}$ and define the corresponding output of the machine as $\tilde{\mathbf{X}}', \tilde{\mathbf{X}}''$. The target objective function is:

$$\sum_{j=1}^{p} \widehat{\mathcal{D}}_{\text{MMD}} \left[\left(\mathbf{X}', \tilde{\mathbf{X}}' \right), \left(\mathbf{X}'', \tilde{\mathbf{X}}'' \right)_{\text{swap}(j)} \right], \tag{9}$$

where $\widehat{\mathcal{D}}_{\text{MMD}}$ stands for the empirical estimate in (6), evaluated with a Gaussian kernel. However, computing the empirical loss in (9) at each iteration might be really expensive (since there are p swaps). In practice, we can only consider two swaps:

$$J_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}}) = \widehat{\mathcal{D}}_{\text{MMD}} \left[\left(\mathbf{X}', \tilde{\mathbf{X}}' \right), \left(\tilde{\mathbf{X}}'', \mathbf{X}'' \right) \right] + \widehat{\mathcal{D}}_{\text{MMD}} \left[\left(\mathbf{X}', \tilde{\mathbf{X}}' \right), \left(\mathbf{X}'', \tilde{\mathbf{X}}'' \right)_{\text{swap}(S)} \right], \tag{10}$$

where S stands for a uniformly chosen random subset of $\{1, \ldots, p\}$ such that $j \in S$ with probability $1/2$. The effectiveness of this substitution is guaranteed by the following theorem:

Theorem 5. *Let $\mathbf{X} \in \mathbb{R}^{n \times p}$ be a collection of independent observations drawn from P_X, and define $\tilde{\mathbf{X}}$ as the corresponding random output of a fixed machine f_θ. Then for J_{MMD} defined as in (10),*

$$\mathbb{E}\left[J_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}})\right] \geq 0. \tag{11}$$

Notice that the equality holds if and only if the machine produces valid knockoffs for P_X.

Proof. Firstly, we rewrite the objective function J as:

$$\begin{aligned}
&\mathbb{E}\left[J_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}})\right] \\
&= \mathbb{E}\left\{\widehat{\mathcal{D}}_{\text{MMD}}\left[\left(\mathbf{X}', \tilde{\mathbf{X}}'\right), \left(\tilde{\mathbf{X}}'', \mathbf{X}''\right)\right]\right\} + \mathbb{E}\left\{\widehat{\mathcal{D}}_{\text{MMD}}\left[\left(\mathbf{X}', \tilde{\mathbf{X}}'\right), \left(\mathbf{X}'', \tilde{\mathbf{X}}''\right)_{\text{swap}(S)}\right]\right\} \\
&= \mathbb{E}\left\{\widehat{\mathcal{D}}_{\text{MMD}}\left[\left(\mathbf{X}', \tilde{\mathbf{X}}'\right), \left(\tilde{\mathbf{X}}'', \mathbf{X}''\right)\right]\right\} + \mathbb{E}\left\{\mathbb{E}\left[\widehat{\mathcal{D}}_{\text{MMD}}\left[\left(\mathbf{X}', \tilde{\mathbf{X}}'\right), \left(\mathbf{X}'', \tilde{\mathbf{X}}''\right)_{\text{swap}(S)}\right] \mid S\right]\right\}.
\end{aligned}$$

From [Gretton et al., 2012], we know that $\widehat{\mathcal{D}}_{\text{MMD}}$ is an unbiased estimator of $\mathcal{D}_{\text{MMD}}$. Intuitively, $\mathbb{E}\left[J_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}})\right] \geq 0$ since $\mathcal{D}_{\text{MMD}}$ is non-negative. And $\mathbf{X}$ is *i.i.d*, the partition is randomly chosen. Thus,

$$\begin{aligned}
\mathbb{E}\left[J_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}})\right] &= \mathcal{D}_{\text{MMD}}\left[P_{(X', \tilde{X}')}, P_{(\tilde{X}'', X'')}\right] + \mathbb{E}\left\{\mathcal{D}_{\text{MMD}}\left[P_{(X', \tilde{X}')}, P_{(X'', \tilde{X}'')_{\text{swap}(S)}}\right]\right\} \\
&= \mathcal{D}_{\text{MMD}}\left[P_{(X, \tilde{X})}, P_{(\tilde{X}, X)}\right] + \mathbb{E}\left\{\mathcal{D}_{\text{MMD}}\left[P_{(X, \tilde{X})}, P_{(X, \tilde{X})_{\text{swap}(S)}}\right]\right\}.
\end{aligned}$$

The first term $\mathcal{D}_{\text{MMD}}\left[P_{(X, \tilde{X})}, P_{(\tilde{X}, X)}\right]$, measures the Maximum Mean Discrepancy (MMD) between the joint distribution of the original variable X and its knockoff copy $\tilde{X}$, and the distribution after swapping these two. This term equals zero if and only if $(\tilde{X}, X)$ has the same distribution as $(X, \tilde{X})$.Then, the second term involves the expectation of the swap operation over a random subset S, expressed as:

$$\mathbb{E}\left\{\mathcal{D}_{\text{MMD}}\left[P_{(X, \tilde{X})}, P_{(X, \tilde{X})_{\text{swap}(S)}}\right]\right\}.$$

This term implies that if $(X, \tilde{X})_{\text{swap}(j)} \overset{d}{=} (X, \tilde{X})$ for all $j \in \{1, \ldots, p\}$, then $(X, \tilde{X})_{\text{swap}(S)} \overset{d}{=} (X, \tilde{X})$ for all subsets $S \subset \{1, \ldots, p\}$, thus making this term also zero. □

Finally, taking the idea from the second-order knockoff case (especially, the regularization term [Candes et al., 2018]), we get the full objective (loss) function as:

$$J(\mathbf{X}, \tilde{\mathbf{X}}) = \gamma J_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}}) + \lambda J_{\text{second-order}}(\mathbf{X}, \tilde{\mathbf{X}}) + \delta J_{\text{decorrelation}}(\mathbf{X}, \tilde{\mathbf{X}}) \tag{12}$$

For optimization, the hyperparameters should be tuned to the specific data distribution. There are many different practical tools to measure goodness of fit. Note that for any fixed choice of $(\gamma, \lambda, \delta)$, the learning strategy is summarized in 4.3. Finally, it is worthy to mention that there could be other types of metric for measuring discrepancy between two distributions, for example, the Kullback-Leibler divergence[Jiang, 2018]. Actually, there are some discussions on empirical estimators of this divergence on deep generative models [Jiang, 2018] [Nguyen et al., 2010], that could also be applied to the problem at our hand.

4.4 Discussions

We omitted the optimization algorithm in this section, instead focusing on applications and potential future directions. Basically, deep generative model is useful for complex distributions. Recall that the simplest example in [Barber and Candès, 2015] is $X \sim \mathcal{N}(0, \mathbf{I})$, where we can directly derive the closed form for the covariance matrix. However, in practice, we can only have the empirical covariance matrix of $(X, \tilde{X}) \in \mathbb{R}^{2p}$ defined as:

$$\hat{G} = \begin{bmatrix} \hat{G}_{XX} & \hat{G}_{X\tilde{X}} \\ \hat{G}_{X\tilde{X}} & \hat{G}_{\tilde{X}\tilde{X}}. \end{bmatrix} \tag{13}$$

And $\hat{G}_{X\tilde{X}}$-should be small as possible for the knockoffs to be powerful. Thus, by using our generative moment matching method, we can extend the previous method to more complex distributions. For example, a multivariate Gaussian mixture model: we assume that each $X \in \mathbb{R}^p$ is independently sampled from

$$X \sim \begin{cases} \mathcal{N}(0, \Sigma_1), & \text{with probability } \frac{1}{3} \\ \mathcal{N}(0, \Sigma_2), & \text{with probability } \frac{1}{3} \\ \mathcal{N}(0, \Sigma_3), & \text{with probability } \frac{1}{3} \end{cases}.$$

Meanwhile, deep knockoffs can also be useful in real-world data. As mentioned in [Romano et al., 2020], it can be used to detect important mutations to a study of variations in drug resistance among human immunodeficiency viruses of type I [Rhee et al., 2006].
Lastly, we briefly discuss some potential points that might be changed. We used higher-order moment matching network. However, the worst-case perspective in the robustness theory may suggest an approach based on generative adversarial networks; Moreover, instead of multi-layer perceptron, deep Boltzmann machines is another alternative choice. In practice, however, even when $(X, \tilde{X})$ is far from respecting the exchangeability property in (2), the false discovery rate may sometimes be controlled in certain real datasets. Therefore, it is necessary to develop a more robust set of validation tools.

5 Robustness

As shown in Section 3 and Section 4, the challenge of estimating the covariate distribution from the sample data can pose a problem for the reliability of our control over the FDR. Below we'll investigate how errors in our distribution approximations effect our control on the FDR. All of the ideas below are from [Barber et al., 2019].

5.1 Distribution estimates

Let $P_j(\cdot|x_{-j})$ denote the true distribution of X_j given $\{X_1, \ldots, X_p\} - \{X_j\}$. Similarly let $Q_j(\cdot|x_{-j})$ be the estimate of the distribution we get from our data. Then we require the following definition:

Definition 4. *$P_{\tilde{X}|X}$ is pairwise exchangeable with respect to Q_j if for any distribution $D^{(j)}$ on $\mathbb{R}^p$ with jth condition Q_j, drawing $X \sim D^{(j)}$ and $\tilde{X}|X \sim P_{\tilde{X}|X}(\cdot|X)$ we have*

$$(X_j, \tilde{X}_j, X_{-j}, \tilde{X}_{-j}) \stackrel{d}{=} (\tilde{X}_j, X_j, X_{-j}, \tilde{X}_{-j})$$

We then measure the discrepancy between the true conditional P_j and its estimate Q_j with the following

$$\widehat{KL}_j := \sum_i \log\left(\frac{P_j(X_{ij}|X_{i,-j}) \cdot Q_j(\tilde{X}_{ij}|X_{i,-j})}{Q_j(X_{ij}|X_{i,-j}) \cdot P_j(\tilde{X}_{ij}|X_{i,-j})}\right)$$

We see if our swap property holds exactly $(X_j, \tilde{X}_j, X_{-j}, \tilde{X}_{-j}) \stackrel{d}{=} (\tilde{X}_j, X_j, X_{-j}, \tilde{X}_{-j})$ then we have the following.

$$\mathbb{E}[\widehat{KL}_j] = \mathbb{E}\left[\sum_i \log\left(\frac{P_j(X_{ij}|X_{i,-j}) \cdot Q_j(\tilde{X}_{ij}|X_{i,-j})}{Q_j(X_{ij}|X_{i,-j}) \cdot P_j(\tilde{X}_{ij}|X_{i,-j})}\right)\right] = \sum_i \mathbb{E}\left[\log\left(\frac{P_j(X_{ij}|X_{i,-j}) \cdot P_j(\tilde{X}_{ij}|X_{i,-j})}{P_j(X_{ij}|X_{i,-j}) \cdot P_j(\tilde{X}_{ij}|X_{i,-j})}\right)\right] = 0$$

This quantity gives us the modified Theorem 4.

Theorem 6. *Under the definitions above, for any $\varepsilon \geq 0$, consider the null variables for which $\widehat{KL}_j \leq \varepsilon$. If we use the knockoff threshold from Theorem 4, then the fraction of the rejections that correspond to such nulls obey*

$$\mathbb{E}\left[\frac{|\{j : j \in \widehat{S} \cap \mathcal{H}_0, \widehat{KL}_j \leq \varepsilon\}|}{|\widehat{S}| \wedge 1}\right] \leq q \cdot e^{\varepsilon}$$

Lemma 7. *If W obeys the "flip-sign" property, the random variables $\widehat{KL}_j$ satisfies the following*

$$\mathbb{P}(W_j > 0, \widehat{KL}_j \leq \varepsilon \big| |W_j|, W_{-j}) \leq e^{\varepsilon} \cdot \mathbb{P}(W_j < 0 \big| |W_j|, W_{-j}) \qquad \forall \varepsilon \geq 0, j \in \mathcal{H}_0$$

We will not prove Lemma 7 here, but its proof can be found in the appendix of [Barber et al., 2019]

Proof. First take any $\varepsilon > 0$ and threshold $t > 0$, then define the following

$$R_\varepsilon(t) := \frac{\sum_{j \in \mathcal{H}_0} \mathbb{1}\{W_j \geq t, \widehat{KL}_j \leq \varepsilon\}}{1 + \sum_{j \in \mathcal{H}_0} \mathbb{1}\{W_j \leq -t\}}$$

$$\begin{aligned}
\frac{|\{j : j \in \widehat{S} \cap \mathcal{H}_0, \widehat{KL}_j \leq \varepsilon\}|}{|\widehat{S}| \wedge 1} =& \frac{\sum_{j \in \mathcal{H}_0} \mathbb{1}\{W_j \geq \tau, \widehat{KL}_j \leq \varepsilon\}}{1 \wedge \sum_j \mathbb{1}\{W_j \geq \tau\}} \\
=& \frac{1 + \sum_j \mathbb{1}\{W_j \leq -\tau\}}{1 \wedge \sum_j \mathbb{1}\{W_j \geq \tau\}} \cdot \frac{\sum_{j \in \mathcal{H}_0} \mathbb{1}\{W_j \geq \tau, \widehat{KL}_j \leq \varepsilon\}}{1 + \sum_j \mathbb{1}\{W_j \leq -\tau\}} \\
\leq& \frac{1 + \sum_j \mathbb{1}\{W_j \leq -\tau\}}{1 \wedge \sum_j \mathbb{1}\{W_j \geq \tau\}} \cdot R_\varepsilon(\tau) \\
\leq& q \cdot R_\varepsilon(\tau)
\end{aligned}$$

Then we find the expectation of $R_\varepsilon(\tau)$. First define $T_j = T((W_1, \ldots, W_{j-1}, |W_j|, W_{j+1}, \ldots, W_p)) > 0$

where T is the threshold rule in Theorem 4. The lemma 7 we get the following.

$$\begin{aligned}
\mathbb{E}[R_\varepsilon(\tau)] &= \sum_{j\in\mathcal{H}_0} \mathbb{E}\left[\frac{\mathbb{1}\{W_j>0, \widehat{KL}_j \le \varepsilon\}\cdot \mathbb{1}\{|W_j|\ge T_j\}}{1+\sum_{k\in\mathcal{H}_0,k\neq j}\mathbb{1}\{W_k\le -T_j\}}\right] \\
&= \sum_{j\in\mathcal{H}_0} \mathbb{E}\left[\frac{\mathbb{P}\{W_j>0, \widehat{KL}_j \le \varepsilon \big| |W_j|, W_{-j}\}\cdot \mathbb{1}\{|W_j|\ge T_j\}}{1+\sum_{k\in\mathcal{H}_0,k\neq j}\mathbb{1}\{W_k\le -T_j\}}\right] \\
&\le e^\varepsilon \cdot \sum_{j\in\mathcal{H}_0} \mathbb{E}\left[\frac{\mathbb{P}\{W_j<0 \big| |W_j|, W_{-j}\}\cdot \mathbb{1}\{|W_j|\ge T_j\}}{1+\sum_{k\in\mathcal{H}_0,k\neq j}\mathbb{1}\{W_k\le -T_j\}}\right] \\
&= e^\varepsilon \cdot \sum_{j\in\mathcal{H}_0} \mathbb{E}\left[\frac{\mathbb{1}\{W_j<0\}\cdot \mathbb{1}\{|W_j|\ge T_j\}}{1+\sum_{k\in\mathcal{H}_0,k\neq j}\mathbb{1}\{W_k\le -T_j\}}\right] \\
&= e^\varepsilon \cdot \mathbb{E}\left[\sum_{j\in\mathcal{H}_0} \frac{\mathbb{1}\{W_j<-T_j\}}{1+\sum_{k\in\mathcal{H}_0,k\neq j}\mathbb{1}\{W_k\le -T_j\}}\right]
\end{aligned}$$

If for all null j, we have $W_j > -T_j$, then the sum is equal to zero, while otherwise, we can write

$$\begin{aligned}
\sum_{j\in\mathcal{H}_0} \frac{\mathbb{1}\{W_j\le -T_j\}}{1+\sum_{k\in\mathcal{H}_0,k\neq j}\mathbb{1}\{W_k\le -T_j\}} &= \sum_{j\in\mathcal{H}_0} \frac{\mathbb{1}\{W_j\le -T_j\}}{1+\sum_{k\in\mathcal{H}_0,k\neq j}\mathbb{1}\{W_k\le -T_k\}} \\
&= \sum_{j\in\mathcal{H}_0} \frac{\mathbb{1}\{W_j\le -T_j\}}{\sum_{k\in\mathcal{H}_0}\mathbb{1}\{W_k\le -T_k\}} \\
&= 1
\end{aligned}$$

Combining everything, we have shown that $\mathbb{E}[R_\varepsilon(\tau)] \le e^\varepsilon$ and therefore Theorem 6

□

References

[Barber and Candès, 2015] Barber, R. F. and Candès, E. J. (2015). Controlling the false discovery rate via knockoffs. *The Annals of Statistics*, 43(5).

[Barber et al., 2019] Barber, R. F., Candès, E. J., and Samworth, R. J. (2019). Robust inference with knockoffs.

[Benjamini and Hochberg, 1995] Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300.

[Candes et al., 2018] Candes, E., Fan, Y., Janson, L., and Lv, J. (2018). Panning for gold:'model-x'knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 80(3):551–577.

[Dziugaite et al., 2015] Dziugaite, G. K., Roy, D. M., and Ghahramani, Z. (2015). Training generative neural networks via maximum mean discrepancy optimization. *arXiv preprint arXiv:1505.03906*.

[Gretton et al., 2012] 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.

[Jiang, 2018] Jiang, B. (2018). Approximate bayesian computation with kullback-leibler divergence as data discrepancy. In *International conference on artificial intelligence and statistics*, pages 1711–1721. PMLR.

[Li et al., 2015] Li, Y., Swersky, K., and Zemel, R. (2015). Generative moment matching networks. In *International conference on machine learning*, pages 1718–1727. PMLR.

[Nguyen et al., 2010] Nguyen, X., Wainwright, M. J., and Jordan, M. I. (2010). Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861.

[Rhee et al., 2006] Rhee, S.-Y., Taylor, J., Wadhera, G., Ben-Hur, A., Brutlag, D. L., and Shafer, R. W. (2006). Genotypic predictors of human immunodeficiency virus type 1 drug resistance. *Proceedings of the National Academy of Sciences*, 103(46):17355–17360.

[Romano et al., 2020] Romano, Y., Sesia, M., and Candès, E. (2020). Deep knockoffs. *Journal of the American Statistical Association*, 115(532):1861–1872.