Journal of the American Statistical Association

ISSN: 0162-1459 (Print) 1537-274X (Online) Journal homepage: https://www.tandfonline.com/loi/uasa20

Metropolized Knockoff Sampling

Stephen Bates, Emmanuel Candès, Lucas Janson & Wenshuo Wang

To cite this article: Stephen Bates, Emmanuel Candès, Lucas Janson & Wenshuo Wang (2020): Metropolized Knockoff Sampling, Journal of the American Statistical Association, DOI: 10.1080/01621459.2020.1729163

To link to this article: https://doi.org/10.1080/01621459.2020.1729163

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2020, VOL. 00, NO. 0, 1–15: Theory and Methods
https://doi.org/10.1080/01621459.2020.1729163

Taylor & Francis
Taylor & Francis Group

Metropolized Knockoff Sampling

Stephen Bates[a], Emmanuel Candès[b], Lucas Janson[c], and Wenshuo Wang[c]

[a]Department of Statistics, Stanford University, Stanford, CA; [b]Department of Mathematics and Statistics, Stanford University, Stanford, CA; [c]Department of Statistics, Harvard University, Cambridge, MA

ABSTRACT

Model-X knockoffs is a wrapper that transforms essentially any feature importance measure into a variable selection algorithm, which discovers true effects while rigorously controlling the expected fraction of false positives. A frequently discussed challenge to apply this method is to construct knockoff variables, which are synthetic variables obeying a crucial exchangeability property with the explanatory variables under study. This article introduces techniques for knockoff generation in great generality: we provide a sequential characterization of all possible knockoff distributions, which leads to a Metropolis–Hastings formulation of an *exact* knockoff sampler. We further show how to use conditional independence structure to speed up computations. Combining these two threads, we introduce an explicit set of sequential algorithms and empirically demonstrate their effectiveness. Our theoretical analysis proves that our algorithms achieve near-optimal computational complexity in certain cases. The techniques we develop are sufficiently rich to enable knockoff sampling in challenging models including cases where the covariates are continuous and heavy-tailed, and follow a graphical model such as the Ising model. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received March 2019
Accepted January 2020

KEYWORDS
False discovery rate; Graphical models, Ising model; Metropolis–Hastings; Treewidth.

1. Introduction

In modern science, researchers often have access to large datasets featuring comprehensive measurements about some phenomenon of interest. The question is then to discover meaningful relationships between an outcome and all the measured covariates. While it is often expected that only a small fraction of the covariates may be associated with the outcome, the relevance of any particular variable is unknown a priori. For instance, a researcher may be interested in understanding which of the thousands of gene-expression profiles may help determine the severity of a tumor. In such circumstances, the researcher often relies on statistical algorithms to sift through large datasets and find those promising candidates, making variable selection a topic of central importance in contemporary statistical research.

The knockoff filter (Barber and Candès 2015; Candès et al. 2018) has recently emerged as a useful framework for performing controlled variable selection, allowing the user to convert any black-box feature importance measure into a variable selection procedure while rigorously controlling the expected fraction of false positives. This means that the statistician can use essentially any black-box importance measure to return a list of variables with the guarantee that, on the average, the ratio between the number of false positives—loosely speaking, a false positive is a variable that does not influence the response, see Candès et al. (2018)—and the total number of reported variables is below a user-specified threshold. The strength of this method is that the guarantees hold in finite samples and in situations where nothing can be assumed about the dependence between the response and the explanatory variables. Instead, the statistician must have knowledge of the distribution of the explanatory variables. When this happens to be the case, a remaining challenge is the ability to generate the *knockoffs*, a set of synthetic variables, which can essentially be used as negative controls; these fake variables must mimic the original variables in a particular way without having any additional predictive power. In sum, constructing valid knockoff distributions and sampling mechanisms across a wide range of covariate models is critical to deploying the knockoff filter in a number of applications.

1.1. Our Contribution

This article describes a theory for sampling knockoff variables and introduces a general and efficient sampler inspired by ideas from Markov chain Monte Carlo (MCMC). Before moving on, we pause to explicitly mention the two main considerations one should keep in mind when constructing knockoffs:

Computation. How can we *efficiently* sample nontrivial knockoffs?

Statistical power. How can we generate knockoffs that will ultimately lead to *powerful* variable selection procedures? There are many different constructions that lead to Type I error control, but some knockoffs will have higher power than others. On this note, it has been observed that knockoffs that have smaller absolute correlation with the original variables

CONTACT Wenshuo Wang *wenshuo_wang@g.harvard.edu* Department of Statistics, Harvard University, Cambridge, MA 02138.
The authors are listed in alphabetical order.
Color versions of one or more of the figures in the article can be found online at *www.tandfonline.com/r/JASA*.
Supplementary materials for this article are available online. Please go to *www.tandfonline.com/r/JASA*.

lead to higher power (Barber and Candès 2015; Candès et al. 2018) and, therefore, low absolute correlation must be a design objective.[1]

Having said that, our work makes several specific contributions.

1. *Characterization of all knockoff distributions*. We provide a sequential characterization of *every* valid knockoff distribution. Furthermore, we introduce a connection linking pairwise exchangeability between original and knockoff variables to reversible Markov chains, enabling the use of powerful sampling tools from computational statistics.
2. *Complexity of knockoff sampling procedures*. We introduce a class of algorithms which use conditional independence information to efficiently generate knockoffs. The computational complexity of such procedures is shown to be determined by the complexity of the dependence structure in a precise way. Furthermore, we present a lower bound on complexity showing that structural assumptions are necessary for efficient computation, and that our procedure achieves the lower bound in certain cases.
3. *Practical sampling algorithms*. We develop a concrete knockoff sampler for a large number of distributions. This is achieved by constructing a family of MCMC tools—designed to have good performance—which only require the numerical evaluation of an *unnormalized* density. We identify a default parameter setting for the sampler that performs well across a variety of situations, producing a general and easy-to-use tool for practitioners.

We shall see that our ideas enable knockoff sampling in challenging models including situations where the covariates are continuous and heavy-tailed and where they follow an Ising model.

1.2. Related Literature

This article draws most heavily on Candès et al. (2018), which builds on Barber and Candès (2015) to introduce the model-X knockoff framework. In particular, the former reference proposes the *Sequential Conditional Independent Pairs* (SCIP) procedure for knockoff generation; this is the only known generic knockoff sampler to date, which shall serve as our starting point. The SCIP procedure, however, is only abstractly specified and prior to this article, implementations were only available for Gaussian distributions and discrete Markov chains. Briefly, Sesia, Sabatti, and Candès (2019) developed a concrete SCIP algorithm for discrete Markov chains, and then leveraged this construction to sample knockoffs for covariates following hidden Markov models widely used in genome-wide association studies. Similarly relevant is the work of Gimenez, Ghorbani, and Zou (2018), which developed a sampling strategy for a restricted class of Bayesian networks, most notably Gaussian mixture models. In contrast, we address here knockoff sampling for a much larger class of distributions, namely, arbitrary graphical models. We also describe the form of all valid knockoff sampling strategies, thereby providing a framework possibly enabling the construction of future knockoff sampling algorithms. Hence, our work may be of value to the increasing number of researchers deploying the knockoff framework for feature selection in a variety of applications including neural networks (Lu et al. 2018), time-series modeling (Fan et al. 2018), Gaussian graphical model structure learning (Zheng et al. 2018), and biology (Xiao et al. 2017; Gao et al. 2018). Lastly, we close by emphasizing that our contribution is very different from a new strand of research introducing approximate knockoffs generated with techniques from deep learning (Liu and Zheng 2018; Romano, Sesia, and Candès 2018; Jordon, Yoon, and van der Schaar 2019). While these approaches are tantalizing and demonstrate promising empirical performance in low-dimensional situations, they currently lack formal guarantees about their validity.

2. Characterizing Knockoff Distributions

2.1. Knockoff Variables

Consider random covariates $X = (X_1, X_2, \ldots, X_p)$. We say that the random variables $\tilde{X} = (\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_p)$ are *knockoffs* for X if for each $j = 1, \ldots, p$,

$$(X, \tilde{X})_{\text{swap}(j)} \overset{d}{=} (X, \tilde{X}). \tag{1}$$

Here, the notation swap(j) means permuting X_j and $\tilde{X}_j$; for instance, $(X_1, X_2, X_3, \tilde{X}_1, \tilde{X}_2, \tilde{X}_3)_{\text{swap}(2)}$ is the vector $(X_1, \tilde{X}_2, X_3, \tilde{X}_1, X_2, \tilde{X}_3)$.[2] Property (1) is known as the *pairwise exchangeability* property, and it is in general challenging to define joint distributions $(X, \tilde{X})$ satisfying this condition. Before continuing, we briefly pause to understand the meaning of pairwise exchangeability. A consequence of (1) is that for all sets $A \subseteq \{1, \ldots, p\}$,

$$(X, \tilde{X})_{\text{swap}(A)} \overset{d}{=} (X, \tilde{X}),$$

where $(X, \tilde{X})_{\text{swap}(A)}$ denotes the swapping of X_j and $\tilde{X}_j$ for all $j \in A$. Taking $A = \{1, \ldots, p\}$ and marginalizing, we immediately see that $\tilde{X} \overset{d}{=} X$; that is, X and $\tilde{X}$ are distributed in the same way. Also changing any subset of entries of X with their knockoff counterparts does not change the distribution either. Another consequence of the exchangeability property (1) is that the mixed second moments of $(X, \tilde{X})$ must match. Assume the second moments of X exist and write $\mathbf{\Sigma} = \text{cov}(X)$. Then the covariance of the vector $(X, \tilde{X})$ must take the form

$$\text{cov}(X, \tilde{X}) = \mathbf{\Gamma}(s) := \begin{bmatrix} \mathbf{\Sigma} & \mathbf{\Sigma} - \text{diag}(s) \\ \mathbf{\Sigma} - \text{diag}(s) & \mathbf{\Sigma} \end{bmatrix}, \tag{2}$$

where $s \in \mathbb{R}^p$ is any vector such that the right-hand side is positive semidefinite. In other words, for each pair (i, j) with $i \neq j$, we have $\text{cov}(X_i, \tilde{X}_j) = \text{cov}(X_i, X_j)$.

We are interested in constructing knockoff variables and below we call a *knockoff sampler* a procedure that takes as inputs a distribution $\mathbb{P}$ and a sample $X \sim \mathbb{P}$ and returns

[1] See Appendix F.6 for a simulation study demonstrating the relationship between power and absolute correlation.

[2] In the presence of a response Y, we also require $\tilde{X} \perp\!\!\!\perp Y \mid X$, which is easily satisfied by procedures that generate $\tilde{X}$ from X without looking at Y.

$\tilde{X}$ such that (1) holds. Nontrivial samplers have been demonstrated in a few cases, for instance, when $X \sim \mathcal{N}(0, \mathbf{\Sigma})$ is multivariate Gaussian. In this case, Candès et al. (2018) show that if $(X, \tilde{X})$ is jointly Gaussian with mean zero and covariance $\mathbf{\Gamma}(s)$, then the entries of $\tilde{X}$ are knockoffs for X. One can say that appropriately matching the first two moments is sufficient to generate knockoffs in the special case of the multivariate normal distribution. However, this does not extend and matching the first two moments is in general not sufficient; to be sure, (1) requires that all moments match appropriately.

As a motivating example, consider the Ising model, a frequently discussed family of Gibbs measures first introduced in the statistical physics literature (Ising 1925). In this model, the random vector $X \in \{-1, 1\}^{d_1 \times d_2}$ defined over a $d_1 \times d_2$ grid has a probability mass function (PMF) of the form

$$\mathbb{P}(X) = \frac{1}{Z(\beta, \alpha)} \exp\left(\sum_{\substack{s,t \in \mathcal{I} \\ \|s-t\|_1 = 1}} \beta_{st} X_s X_t + \sum_{s \in \mathcal{I}} \alpha_s X_s \right); \quad (3)$$

here, $\mathcal{I} = \{(i_1, i_2) : 1 \le i_1 \le d_1, 1 \le i_2 \le d_2\}$ is the grid and α and β are parameters. As we have seen, knockoffs $\tilde{X}$ for X must marginally follow the Ising distribution (3). Furthermore, $\tilde{X}$ must be dependent on X in such a way that any vector of the form $\{(Z_1, \dots, Z_p) : Z_j = X_j \text{ or } Z_j = \tilde{X}_j, 1 \le j \le p\}$ has PMF given by (3). It is tempting to naïvely define a joint PMF for $(X, \tilde{X})$ as

$$\mathbb{P}(X, \tilde{X}) \propto \exp\left(\sum_{\substack{s,t \in \mathcal{I} \\ \|s-t\|_1 = 1}} \beta_{st}(X_s X_t + \tilde{X}_s \tilde{X}_t + X_s \tilde{X}_t + \tilde{X}_s X_t) + \sum_{s \in \mathcal{I}} \alpha_s (X_s + \tilde{X}_s) \right).$$

Although the joint distribution is symmetric in X_s and $\tilde{X}_s$, the marginal distribution of X is not an Ising model! Hence, this is not a valid joint distribution. Other than the trivial construction $\tilde{X} = X$, it is a priori unclear how one would construct knockoffs. Any distribution continuous or discrete factoring over a grid poses a similar challenge.

2.2. SCIP and Its Limitations

The only generic knockoff sampler one can find in the literature is SCIP from Candès et al. (2018), given in Procedure 1. While this procedure provably generates valid knockoffs for any input distribution, there are two substantial limitations. The first is that SCIP is only given abstractly; it is challenging to specify $\mathcal{L}(X_j \mid X_{-j}, \tilde{X}_{1:(j-1)})$,[3] let alone to sample from it. As a result, it is only known how to implement SCIP for very special models such as discrete Markov chains and Gaussian distributions. The second limitation is that SCIP is not able to generate all valid knockoff distributions. Recall that we want knockoffs to have low absolute correlations with the original variables so that a feature importance statistic will correctly detect true effects. To achieve this goal, we might need a wider range of sampling mechanisms.

Procedure 1: Sequential Conditional Independent Pairs (SCIP)

for $j = 1$ **to** p **do**
  Sample $\tilde{X}_j$ from $\mathcal{L}(X_j \mid X_{-j}, \tilde{X}_{1:(j-1)})$, conditionally independently from X_j
end

2.3. Sequential Formulation of Knockoff Distributions

We begin by introducing a sequential characterization of *all* valid knockoff distributions, which will later lead to a new class of knockoff samplers.

Theorem 1 (Sequential characterization of knockoff distributions). Let $(X, \tilde{X}) \in \mathbb{R}^{2p}$ be a random vector. Then pairwise exchangeability (1) holds if and only if both of the following conditions hold:

Conditional exchangeability. For each $j \in \{1, \dots, p\}$,

$$(X_j, \tilde{X}_j) \mid X_{-j}, \tilde{X}_{1:(j-1)} \overset{d}{=} (\tilde{X}_j, X_j) \mid X_{-j}, \tilde{X}_{1:(j-1)}. \quad (4)$$

Knockoff symmetry. For each $j \in \{1, \dots, p\}$,

$$\mathbb{P}((X_j, \tilde{X}_j) \in A \mid X_{-j}, \tilde{X}_{1:(j-1)}) \quad (5)$$

is $\sigma(X_{(j+1):p}, \{X_1, \tilde{X}_1\}, \dots, \{X_{j-1}, \tilde{X}_{j-1}\})$-measurable for any Borel set A, where $\{\cdot, \cdot\}$ denotes the unordered pair. That is, the conditional distribution does not change if we swap previously sampled knockoffs with the original features.

Theorem 1 implies that a sequential knockoff sampling algorithm faithful to these two conditions is as general as it gets. The challenge now becomes creating exchangeable random variables at each step (with a little caution on the dependence on the previous pairs of variables). In turn, this task happens to be equivalent to designing a time-reversible Markov chain, as formalized below.

Proposition 1. A pair of random variables $(Z, \tilde{Z})$ is exchangeable, that is, $(Z, \tilde{Z}) \overset{d}{=} (\tilde{Z}, Z)$, with marginal distribution π for Z—and, therefore, for $\tilde{Z}$ as well—if and only if there exists a time-reversible Markov chain $\{Z_n\}_{n=1}^{\infty}$ such that $Z_1 \sim \pi$ is a stationary distribution of the chain, and $(Z_1, Z_2) \overset{d}{=} (Z, \tilde{Z})$.

Combining these two results gives SCEP (Procedure 2), which is a completely general strategy for generating knockoffs: at each step j, we design a time-reversible Markov chain with stationary distribution $\mathcal{L}(X_j \mid X_{-j}, \tilde{X}_{1:(j-1)})$, and draw a sample by taking one step of this chain starting from X_j. Proposition 1 implies that the conditional exchangeability (4)

[3] We use $\mathcal{L}(W_1 \mid W_2)$ to denote the conditional distribution of W_1 given W_2. We use subscript $-k$ to mean the vector with the kth coordinate removed, and $1{:}k$ to mean the first k coordinates of the vector. We use the subscript 1:0 to mean an empty vector.

holds. Furthermore, the symmetry requirement on the transition kernel implies that SCEP does not break the exchangeability from previous steps; that is, the knockoff symmetry (5) also holds. Theorem 1 then implies that such a procedure produces valid knockoffs.

Procedure 2: Sequential Conditional Exchangeable Pairs (SCEP)

for $j = 1$ **to** p **do**
  Sample $\tilde{X}_j$ by taking one step of a time-reversible Markov chain starting from X_j.
  The transition kernel must be such that it depends only on $X_{(j+1):p}$ and the unordered pairs $\{X_1, \tilde{X}_1\}, \ldots, \{X_{j-1}, \tilde{X}_{j-1}\}$, and admits $\mathcal{L}(X_j \mid X_{-j}, \tilde{X}_{1:(j-1)})$ as a stationary distribution.
end

To rehearse the universality of SCEP, consider an arbitrary knockoff sampler producing $\tilde{X}_1, \ldots, \tilde{X}_p$. Then from Theorem 1 we know that X_1 and $\tilde{X}_1$ must be exchangeable conditional on X_{-1}. Therefore, $\tilde{X}_1$ may be sampled by taking one step of a reversible Markov chain starting at X_1. Moving on to X_2, Theorem 1 informs us that X_2 and $\tilde{X}_2$ are exchangeable conditional on $\{X_1, \tilde{X}_1\}, X_3, \ldots, X_p$, so $\tilde{X}_2$ can again be viewed as taking one step of a reversible Markov chain starting at X_2. Continuing in this fashion for $j = 3, \ldots, p$ establishes our claim.

SCEP as stated remains too abstract to be considered an implementable algorithm, so we will next develop a concrete version of this procedure. Although this may not yet be clear, we would like to stress that formulating a knockoff sampler in terms of reversible Markov chains is an important step forward because it will ultimately enable the use of flexible MCMC tools.

3. The Metropolized Knockoff Sampler

We now demonstrate how one can generate knockoffs in a sequential manner by making proposals which are either accepted or rejected in a Metropolis–Hastings-like fashion as to ensure pairwise exchangeability.

3.1. Algorithm Description

The celebrated Metropolis–Hastings (MH) algorithm (Metropolis et al. 1953; Hastings 1970) provides a general time-reversible Markov transition kernel whose stationary distribution is an arbitrary density function π. To construct a transition from x to y, MH operates as follows: generate a proposal x^* from a distribution $q(\cdot \mid x)$ (any distribution depending on x) and set[4]

$$y = \begin{cases} x^* & \text{with prob. } \alpha, \\ x & \text{with prob. } 1-\alpha, \end{cases} \qquad \alpha = \min\left(1, \frac{\pi(x^*)q(x \mid x^*)}{\pi(x)q(x^* \mid x)}\right).$$

This can be implemented even when the density π is unnormalized, as the normalizing constants cancel. In our setting, we shall make sure that the choice of the proposal distribution depends on the previously sampled pairs in a symmetric fashion, thereby remaining faithful to the knockoff symmetry condition (5) in Theorem 1. As such, we call such proposals *faithful*.

Consider now running SCEP (Procedure 2) with the MH kernel, where at the jth step, the target distribution π is taken to be $\mathcal{L}(X_j \mid X_{-j}, \tilde{X}_{1:j-1})$. The issue with such a naïve implementation is that the target π cannot be readily evaluated. To understand why this is the case, set $j = 2$ and consider $\mathcal{L}(X_2 \mid X_{-2}, \tilde{X}_1)$. This distribution has density proportional to $\mathbb{P}(X = x)\mathbb{P}(\tilde{X}_1 = \tilde{x}_1 \mid X = x)$, which is equal to

$$\mathbb{P}(X = x)\Bigg[q(\tilde{x}_1 \mid x_1)\min\left(1, \frac{q(x_1 \mid \tilde{x}_1)\mathbb{P}(X_1 = \tilde{x}_1, X_{-1} = x_{-1})}{q(\tilde{x}_1 \mid x_1)\mathbb{P}(X_1 = x_1, X_{-1} = x_{-1})}\right) + \delta(\tilde{x}_1 - x_1)\int q(x^* \mid x_1)\left(1 - \min\left(1, \frac{q(x_1 \mid x^*)\mathbb{P}(X_1 = x^*, X_{-1} = x_{-1})}{q(x^* \mid x_1)\mathbb{P}(X_1 = x_1, X_{-1} = x_{-1})}\right)\right)dx^*\Bigg]. \quad (6)$$

The first term in the summation within the brackets corresponds to the acceptance case while the second corresponds to the rejection case. This latter term cannot be evaluated because of the integral over x^*. Hence, the target density cannot be evaluated either.

We propose an effective solution to this problem: *condition on the proposals* and at step j, let the target distribution be $\mathcal{L}(X_j \mid X_{-j}, \tilde{X}_{1:j-1}, X^*_{1:j-1})$ rather than $\mathcal{L}(X_j \mid X_{-j}, \tilde{X}_{1:j-1})$. This has the effect of removing the integral and makes computing the rejection probability tractable. This is best seen by returning to our example where $j = 2$. Here, $\mathcal{L}(X_j \mid X_{-j}, \tilde{X}_{1:j-1}, X^*_{1:j-1})$ has density now proportional to

$$\mathbb{P}(X = x)q(x_1^* \mid x_1)\Bigg[\delta(\tilde{x}_1 - x_1^*)\min\left(1, \frac{q(x_1 \mid \tilde{x}_1)\mathbb{P}(X_1 = \tilde{x}_1, X_{-1} = x_{-1})}{q(\tilde{x}_1 \mid x_1)\mathbb{P}(X_1 = x_1, X_{-1} = x_{-1})}\right) + \delta(\tilde{x}_1 - x_1)\left(1 - \min\left(1, \frac{q(x_1 \mid x_1^*)\mathbb{P}(X_1 = x_1^*, X_{-1} = x_{-1})}{q(x_1^* \mid x_1)\mathbb{P}(X_1 = x_1, X_{-1} = x_{-1})}\right)\right)\Bigg]. \quad (7)$$

We will show in Section 4 how such terms can be efficiently computed. Leaving aside implementation details for the moment, this strategy leads to Algorithm 1. Here and elsewhere, $\mathbb{P}$ denotes the density of the variables under study, or formally, the Radon–Nikodym derivative with respect to a common dominating measure.

Algorithm 1: Metropolized knockoff sampling (Metro).

for $j = 1$ **to** p **do**
  Sample $X_j^* = x_j^*$ from a faithful proposal distribution q_j.
  Accept the proposal with probability
  $$\min\left(1, \frac{q_j(x_j \mid x_j^*)\mathbb{P}\left(X_{-j} = x_{-j}, X_j = x_j^*, \tilde{X}_{1:(j-1)} = \tilde{x}_{1:(j-1)}, X^*_{1:(j-1)} = x^*_{1:(j-1)}\right)}{q_j(x_j^* \mid x_j)\mathbb{P}\left(X_{-j} = x_{-j}, X_j = x_j, \tilde{X}_{1:(j-1)} = \tilde{x}_{1:(j-1)}, X^*_{1:(j-1)} = x^*_{1:(j-1)}\right)}\right).$$
  Upon acceptance, set $\tilde{x}_j = x_j^*$; otherwise, set $\tilde{x}_j = x_j$.
end
Return $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_p)$

Note that even with the additional conditioning on $X^*_{1:(j-1)}$ this is a SCEP procedure, because if X_j and $\tilde{X}_j$ are exchangeable conditional on $(X_{-j}, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})$, then by marginalizing

[4] More generally, we take as acceptance probability $\gamma\,\alpha$ with $\gamma \in (0, 1]$. In this work, γ is set to 1 as default, except in Section 3.3 and Appendix F.2, which are cases where tuning γ is recommended.

out $X^*_{1:(j-1)}$, we see that they are also exchangeable given only $X_{-j}, \tilde{X}_{1:(j-1)}$. As a result, Algorithm 1 produces valid knockoffs, which we record formally below.

Corollary 1. Metropolized knockoff sampling (Metro) produces valid knockoffs.

Proof. For the sake of the proof, let U_j be the indicator of acceptance at step j, and $Z_j = (1-U_j)X^*_j$. We will prove pairwise exchangeability jointly with the U_j's and Z_j's; marginalizing out these variables will establish the claim. For $1 \le j \le p$, let $f_j(x_j, x_{-j}, \tilde{x}_{1:(j-1)}, u_{1:(j-1)}, z_{1:(j-1)})$ be the joint density function of $(X_j, X_{-j}, \tilde{X}_{1:(j-1)}, U_{1:(j-1)}, Z_{1:(j-1)})$, in this order. We will use induction to show that the density of $(X, \tilde{X}_{1:j}, U_{1:j}, Z_{1:j})$ is symmetric in X_k and $\tilde{X}_k$ for $1 \le k \le j$. For $1 \le j \le p$, the inductive hypothesis is that f_j is symmetric in x_k and $\tilde{x}_k$ for $1 \le k \le j-1$ (since f_j is just the density of $(X, \tilde{X}_{1:(j-1)}, U_{1:(j-1)}, Z_{1:(j-1)})$ after reordering the variables). For $1 \le j \le p$,

$$
\begin{aligned}
&\text{the density of } (X, \tilde{X}_{1:j}, U_{1:j}, Z_{1:j}) \text{ at } (x, \tilde{x}_{1:j}, u_{1:j}, z_{1:j}) \\
&= f_j(x_j, x_{-j}, \tilde{x}_{1:(j-1)}, u_{1:(j-1)}, z_{1:(j-1)}) \times \\
&\quad \Bigg[\mathbf{1}_{u_j=1} \delta(z_j - 0) q_j(\tilde{x}_j \mid x_j) \\
&\qquad \min\left(1, \frac{f_j(\tilde{x}_j, x_{-j}, \tilde{x}_{1:(j-1)}, u_{1:(j-1)}, z_{1:(j-1)}) q_j(x_j \mid \tilde{x}_j)}{f_j(x_j, x_{-j}, \tilde{x}_{1:(j-1)}, u_{1:(j-1)}, z_{1:(j-1)}) q_j(\tilde{x}_j \mid x_j)}\right) \\
&\qquad + \mathbf{1}_{u_j=0} \delta(\tilde{x}_j - x_j) q_j(z_j \mid x_j) \Bigg(1 - \\
&\qquad \min\left(1, \frac{f_j(z_j, x_{-j}, \tilde{x}_{1:(j-1)}, u_{1:(j-1)}, z_{1:(j-1)}) q_j(x_j \mid z_j)}{f_j(x_j, x_{-j}, \tilde{x}_{1:(j-1)}, u_{1:(j-1)}, z_{1:(j-1)}) q_j(z_j \mid x_j)}\right) \Bigg) \Bigg],
\end{aligned}
$$

which is symmetric in the first $j-1$ pairs by the inductive hypothesis. For the symmetry in the jth pair, when $u_j = 1$, the density simplifies to

$$
\begin{aligned}
\delta(z_j - 0) \times \min\big(& f_j(x_j, x_{-j}, \tilde{x}_{1:(j-1)}, u_{1:(j-1)}, z_{1:(j-1)}) q_j(\tilde{x}_j \mid x_j), \\
& f_j(\tilde{x}_j, x_{-j}, \tilde{x}_{1:(j-1)}, u_{1:(j-1)}, z_{1:(j-1)}) q_j(x_j \mid \tilde{x}_j) \big),
\end{aligned}
$$

which is invariant to swapping x_j and $\tilde{x}_j$; when $u_j = 0$, the delta function $\delta(\tilde{x}_j - x_j)$ ensures $x_j = \tilde{x}_j$, and thus swapping them has no effect. Hence, when the algorithm terminates, all pairs are exchangeable and therefore remain exchangeable after marginalizing out the U_j's and Z_j's. □

Anticipating possible future applications, we wish to remark that Metro can be easily adapted to sampling *group* knockoffs (Dai and Barber 2016); see Appendix E.

3.2. Covariance-Guided Proposals

Now that we have available a broad class of knockoff samplers, we turn to the question of finding faithful proposal distributions that will generate statistically powerful knockoffs. The overall challenge is to propose samples that are far away from X to make good knockoffs, but not as far that they are systematically rejected. A rejection at the jth step gives $\tilde{X}_j = X_j$, leading to a knockoff with poor contrast. Below, we shall borrow ideas from existing knockoff samplers for Gaussian models to make sensible proposals.

Suppose that X has mean μ and covariance $\mathbf{\Sigma}$, and consider $s \in \mathbb{R}^p$ with nonnegative entries such that $\mathbf{\Gamma}(s)$ from (2) is positive semidefinite. Such a vector s can be found with techniques from Barber and Candès (2015) and from Candès et al. (2018). We have seen that if X were Gaussian, this covariance matrix would induce a multivariate Gaussian joint distribution over X and $\tilde{X}$ with the correct symmetry. In non-Gaussian settings, our observation is that we can still make proposals as if the variables were Gaussian, but use the MH correction to guarantee exact conditional exchangeability. This can be viewed as a Metropolis-adjustment to the second-order knockoff construction of Candès et al. (2018). Concretely, the distribution q_j for a covariance-guided proposal—used to generate a proposal X^*_j—is normal with mean

$$
\mu_j + \left(\Gamma^{(j)}_{12}\right)^\top \left(\mathbf{\Gamma}^{(j)}_{11}\right)^\dagger \left(X - \mu, X^*_{1:(j-1)} - \mu_{1:(j-1)}\right)^\top
$$

and variance

$$
\Gamma^{(j)}_{22} - \left(\Gamma^{(j)}_{12}\right)^\top \left(\mathbf{\Gamma}^{(j)}_{11}\right)^\dagger \Gamma^{(j)}_{12};
$$

here, $X^*_{1:(j-1)}$ is the sequence of already generated proposals, $\mathbf{\Gamma}^{(j)}_{11} = \mathbf{\Gamma}_{1:(p+j-1),1:(p+j-1)}$, $\Gamma^{(j)}_{22} = \Gamma_{p+j,p+j}$, $\Gamma^{(j)}_{12} = \Gamma_{1:(p+j-1),p+j}$, μ is the mean of X, and † stands for the pseudoinverse. The parameters of q_j can be efficiently computed using the special structure of $\mathbf{\Gamma}$; see Appendix D. The faithfulness of the proposal is shown in Appendix B.

The covariance-guided proposals are valid even when $\mathbf{\Sigma}$ is replaced by any other positive semidefinite matrix—*any faithful proposal distribution will give valid knockoffs*. This allows us to use an empirical estimate of $\mathrm{cov}(X)$ based on simulated samples from $\mathcal{L}(X)$, or even to apply the covariance-guided proposals to discrete distributions by rounding each proposal X^*_j to the nearest point in the support of X_j. These proposals will be most successful when X is well-approximated by a Gaussian density, indeed when X is exactly Gaussian and the true covariance is used, the covariance-guided proposals will never be rejected. Numerical simulations in a variety of settings can be found in Section 5.

3.3. Multiple-Try Metropolis

A possibility for sampling $\tilde{X}_j$ "far away" from X_j is to run multiple MH steps instead of a single one. The issue with this is that this would make the conditional distributions from Metro prohibitively complex at later steps. Longer chains also require conditioning later proposals on a longer sequence of proposals and acceptances or rejections, which will constrain those proposals to be closer to their corresponding true variables and thus reduce power. Instead, we use the multiple-try Metropolis (MTM) technique introduced in Liu, Liang, and Wong (2000).

The key idea of MTM is to propose a set of several candidate moves to increase the probability of acceptance. As in Qin and Liu (2001), we take the candidate set to be $C^{m,t}_x = \{x \pm kt : 1 \le k \le m\}$, where m is a positive integer and t is a positive number; see Figure 1 for an illustration. MTM proceeds by choosing one

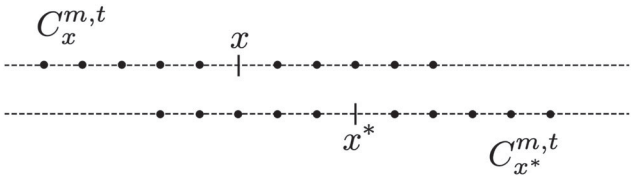

Figure 1. Multiple-try Metropolis (adapted from Figure 2 in Qin and Liu (2001)).

element x^* from the set $C_x^{m,t}$, with probability proportional to the target density, that is,

$$\mathbb{P}(\text{select } x^* \text{ from } C_x^{m,t}) = \frac{\pi(x^*)}{\sum_{u \in C_x^{m,t}} \pi(u)}. \tag{8}$$

This proposal is then accepted with probability

$$\gamma \min\left(1, \frac{\sum_{u \in C_x^{m,t}} \pi(u)}{\sum_{v \in C_{x^*}^{m,t}} \pi(v)}\right), \quad \gamma \in (0,1), \tag{9}$$

where γ is an additional tuning parameter explained in Appendix F.2. This parameter should be taken to be near 1 in most settings. If no element of $C_x^{m,t}$ has positive probability, then one automatically rejects. MTM is a special case of MH with the proposal $q(x^* \mid x)$ distribution defined implicitly by the above rules, and furthermore, the proposals are faithful.[5] Thus, MTM can be used in Metro.

While there is no universally optimal combination of m and t, we provide guidance about default values based on our experimental results from Section 5. To understand the choice of parameters, first observe that with a fixed t, large values of m intuitively induce high acceptance rate, but require more density evaluations. Turning our attention to t, smaller values cause higher acceptance rates, and at the same time, encourage $\tilde{X}_j$ to be close to X_j. Clearly, there is a trade-off. Based on our experiments, a sensible default setting is $m = 4$ and $t_j = 1.5\sqrt{1/(\boldsymbol{\Sigma}^{-1})_{jj}}$ where $\boldsymbol{\Sigma} = \text{cov}(X)$. In the Gaussian case, $\text{var}(X_j \mid X_{-j}) = 1/(\boldsymbol{\Sigma}^{-1})_{jj}$ for any observed value of X_{-j} (Anderson 2003), hence this choice of scaling is intuitive. In the non-Gaussian case $1/(\boldsymbol{\Sigma}^{-1})_{jj}$ should be viewed as an approximation to the conditional variance. We have found that this parameter setting achieves nearly the best performance in most of our experiments, so unless the distribution is thought to be nearly Gaussian, our recommendation is to use MTM with the above parameter settings to sample knockoffs.

4. Graphical Models and Conditional Independence

One outstanding issue is whether the Metropolized knockoff sampler can be run in reasonable time for cases of interest. We begin by showing why sequential knockoff sampling is prohibitively expensive without additional structure, and then turn our attention to a common type of structure that enables efficient sampling: graphical models. The central contribution of this section will be a complexity bound on Metro showing how the graphical structure affects the difficulty of sampling. To complete this line of investigation, we give a complexity lower bound for all knockoff samplers which shows that Metro is optimal in some cases.

4.1. Why Do We Need Structure?

Consider running Metro for some input distribution $\mathbb{P}$ and sample $X = x$. In view of (7), at step j we need to evaluate $\mathbb{P}(X_{-j}, X_j = z_j, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})$ for $z_j \in \{x_j, x_j^*\}$ up to a constant.[6] Metro defines a joint distribution on $(X, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})$ implicitly, so the only way to evaluate this density is to compute it step by step, from 1 to $j-1$, that is, through the sequential decomposition

$$\begin{aligned}\mathbb{P}(X_{-j}, X_j = z_j, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)}) &= \mathbb{P}(X_{-j}, X_j = z_j) \\ &\times \prod_{k=1}^{j-1} \Big[\mathbb{P}(\tilde{X}_k \mid X_{-j}, X_j = z_j, \tilde{X}_{1:(k-1)}, X^*_{1:k}) \\ &\qquad \mathbb{P}(X_k^* \mid X_{-j}, X_j = z_j, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)})\Big]. \end{aligned} \tag{10}$$

Consider the term $\mathbb{P}(\tilde{X}_k \mid X_{-j}, X_j = z_j, \tilde{X}_{1:(k-1)}, X^*_{1:k})$. By the definition of Metro, computing this term will require evaluating an acceptance probability of the form

$$\min\left(1, \frac{q_k(x_k \mid x_k^*)\mathbb{P}(X_{-(j,k)}, X_k = x_k^*, X_j = z_j, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)})}{q_k(x_k^* \mid x_k)\mathbb{P}(X_{-(j,k)}, X_k = x_k, X_j = z_j, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)})}\right). \tag{11}$$

Now, to compute the terms in the acceptance probability, we must use the same sequential decomposition (10) for the terms $\mathbb{P}(X_{-(j,k)}, X_k = z_k, X_j = z_j, \tilde{X}_{1:k-1}, X^*_{1:k-1})$ for $z_k \in \{x_k, x_k^*\}$. Considering $k = j-1$, we see that step j is making two calls to the probability at step $j-1$, each of which is in turn making two calls to the probability function at step $j-2$ and so on. Thus, each evaluation of (10) will require $\Omega(2^j)$ function calls. This behavior is not due to a shortcoming of Metro; any genuine knockoff sampler with access only to an unnormalized density will require time exponential in p. We will present the formal statement of this lower bound later in Theorem 3.

Although knockoff sampling with no restriction on the distribution is prohibitively slow, we will show how to avoid the exponential complexity when there is additional known structure. Consider a Markov chain, that is, a density that factors as $\mathbb{P}(x) = \prod_{j=1}^{p-1} \phi_j(x_j, x_{j+1})$. In this case, the joint density (10) can be evaluated efficiently provided we proceed along the chain in the natural order. Assume for simplicity that the proposal distribution is fixed in advance so that the second term within the square brackets in (10) does not depend on any variables and can be ignored. Due to the Markovian structure, only the $k = j-1$ term in the product depends on z_j, so it suffices to compute the acceptance probability (11) for $k = j-1$. Again using the Markovian structure, this simplifies to

[5] The proposal distribution at step j only depends on the conditional density of $\mathcal{L}(X_j \mid X_{-j}, X^*_{1:(j-1)}, \tilde{X}_{1:(j-1)})$ which can be easily shown to be symmetric in the first $j-1$ pairs.

[6] In this section, when not explicitly specified, a variable is set to its observed value, for example, $\mathbb{P}(X_1 \mid X_2 = z_2, X_3, \tilde{X}_1, X_1^*)$ is shorthand for $\mathbb{P}(X_1 = x_1 \mid X_2 = z_2, X_3 = x_3, \tilde{X}_1 = \tilde{x}_1, X_1^* = x_1^*)$.

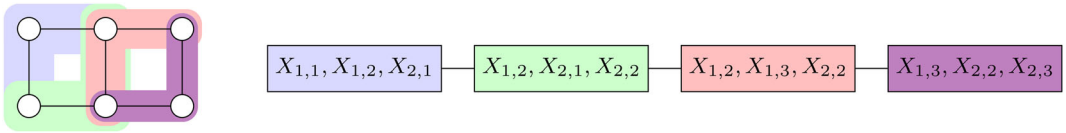

Figure 2. A junction tree of treewidth 2 for the 2×3 grid, which happens to be a chain.

$$\min\left(1, a_{j-1}\frac{q_{j-1}(x_{j-1} \mid x^*_{j-1})\mathbb{P}(X_{-(j,j-1)}, X_{j-1} = x^*_{j-1}, X_j = z_j)}{q_{j-1}(x^*_{j-1} \mid x_{j-1})\mathbb{P}(X_{-(j,j-1)}, X_{j-1} = x_{j-1}, X_j = z_j)}\right)$$
$$= \min\left(1, a_{j-1}\frac{q_{j-1}(x_{j-1} \mid x^*_{j-1})\phi_{j-2}(x_{j-2}, x^*_{j-1})\phi_{j-1}(x^*_{j-1}, z_j)}{q_{j-1}(x^*_{j-1} \mid x_{j-1})\phi_{j-2}(x_{j-2}, x_{j-1})\phi_{j-1}(x_{j-1}, z_j)}\right),$$

where a_{j-1} is the ratio

$$a_{j-1} = \frac{\mathbb{P}(X^*_{1:(j-2)}, \tilde{X}_{1:(j-2)} \mid X_{-(j,j-1)}, X_{j-1} = x^*_{j-1}, X_j = z_j)}{\mathbb{P}(X^*_{1:(j-2)}, \tilde{X}_{1:(j-2)} \mid X_{-(j,j-1)}, X_{j-1} = x_{j-1}, X_j = z_j)},$$

which does not depend on z_j by the Markov structure. The key here is that a_{j-1} was previously computed with $z_j = x_j$ when sampling $\tilde{X}_{j-1}$. Thus, the acceptance probability can be computed in constant time. Putting this all together, for a Markov chain, each of the necessary joint probabilities (10) can be computed in constant time, and the time to sample the entire vector $\tilde{X}$ is linear in the dimension p. Markov chains are not the only case where computing the acceptance probability can be done quickly; for other distributions with conditional independence structure, we next develop a systematic way of computing (10), using the graphical structure to control the depth of the recursion and hence control the running time.

4.2. Time Complexity of Metro for Graphical Models

We have seen that we must restrict our attention to a subset of distributions to efficiently sample knockoffs, so in this section we show how to implement Metropolized knockoff sampling for a very broad class of distributions: graphical models. Let $X \in \mathbb{R}^p$ be a random vector whose density factors over a graph G:

$$\mathbb{P}(x) \propto \Phi(x) = \prod_{c \in C} \phi_c(x_c); \tag{12}$$

here, C is the set of maximal cliques of the graph G and Φ is an unnormalized version of $\mathbb{P}$. The variables in X can be either discrete or continuous. All graphical models with positive density or mass take this form (Hammersley and Clifford 1971), and such distributions are known to have particular conditional independence properties. We refer the reader to Koller and Friedman (2009) for a general treatment.

To take advantage of the conditional independence structure of X, we use a graph-theoretical object known as a *junction tree* (Bertele and Brioschi 1972) which encodes properties of the graph G.

Definition 1 (Junction tree). Let T be a tree with vertices that are subsets of the nodes $\{1, \dots, p\}$ of a graph G. T is a *junction tree* for G if the following hold:

1. Each $j \in \{1, \dots, p\}$ appears in some vertex V of T.
2. For every edge (j, k) in G, $j \in V$ and $k \in V$ for some vertex V.
3. (Running intersection property) If the vertices V and V' both contain a node of G, then every vertex in the unique path from V to V' also contains this node.

Figure 2 gives an example of a junction tree over a 2×3 grid. The size of the largest vertex of T minus one is known as the *width* of the junction tree T, and the smallest width of a junction tree over G is called the *treewidth* of G, a measure of graph complexity. Finding the junction tree of lowest width for a graph G is known to be NP-hard (Arnborg, Corneil, and Proskurowski 1987), but there exist efficient heuristic algorithms for finding a junction tree with small width (Kjærulff 1990; Koller and Friedman 2009).

Given a junction tree T for the graph G, we will soon prove that Metro can be run with $O(p2^w)$ queries of the unnormalized density Φ, where w is the width of T. In view of (7), at step j of Metro we need to evaluate $\mathbb{P}(X_{-j}, X_j = z_j, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})$ for $z_j \in \{x_j, x^*_j\}$ up to a constant as well as sample from and evaluate the proposal distribution $q_j(\cdot \mid x_j)$. We can use the graphical model structure to make these operations tractable by both (1) sampling the variables in a specific order, and (2) choosing proposal distributions that are not unnecessarily complex. We formalize these two requirements below.

We first consider the order in which we sample the variables. Recalling (10), the complexity of the computations of $\mathbb{P}(X_{-j}, X_j = z_j, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})$ depends on the number of function calls implied by the recursion (10). For simplicity, assume that the proposal terms in the product, $\mathbb{P}(X^*_k \mid X_{-j}, X_j = z_j, \tilde{X}_{1:(k-1)}, X^*_{1:(k-1)})$, never depends on z_j; this will be relaxed soon. In that case, we need only consider the terms in (10) of the form $\mathbb{P}(\tilde{X}_k \mid X_{-j}, X_j = z_j, \tilde{X}_{1:(k-1)}, X^*_{1:k})$ for $k < j$. When there is graphical structure, not all such terms will depend on z_j, and the number of terms that do depend on z_j determines the recursion depth. In particular, if at step j only r_j terms depend on z_j, then there will be $O(2^{r_j})$ function calls in the recursion. A desirable ordering of the variables is then one that minimizes the largest r_j, and such an ordering can be extracted from a junction tree T using Algorithm 2.

Algorithm 2 is valid in that when a node is removed, no $j \in J$ remains in any node in T_{active}.[7] From now on we assume that the variables are numbered according to this ordering. Our second consideration is to create proposals that do not add unnecessary complexity. No matter which proposal distribution we choose, $\mathbb{P}(X_j = z_j \mid X_{-j}, \tilde{X}_{1:(j-1)}, X^*_{1:(j-1)})$ will still depend on some X_ℓ for $\ell > j$ due to dependencies among coordinates of X; we however restrict ourselves to proposal distributions that do not add any additional dependencies.

[7] This simple fact follows from the running intersection property; we refer the reader to Lemma 1 in Appendix A.

Algorithm 2: Junction tree variable ordering for Metro

Initialize tree $T_{\text{active}} = T$ and list $J = \{\}$.
while $T_{\text{active}} \neq \emptyset$ **do**
  Select a leaf node V of T_{active}. V is connected to at most one other node V' of T_{active} because it is a tree.
  In any order, append each $j \in V \setminus V'$ to the end of the list J. If no V' exists, append all $j \in V$ to J in any order.
  Remove V from the active tree T_{active}.
end
Return J

Definition 2 (Compatible proposal distributions). Let V_j be the node of the junction tree when j is appended to J from Algorithm 2. Set $\bar{V}_j = \{1, \ldots, j-1\} \cup V_j$. We say that proposal distributions q_j are *compatible* with a junction tree T if they depend only on $X_{\bar{V}_j}, \tilde{X}_{1:(j-1)}$, and $X^*_{1:(j-1)}$.

This definition is motivated by the property

$$X_{1:j} \perp\!\!\!\perp X_{\bar{V}_j^c} \mid X_{\bar{V}_j \setminus \{1,\ldots,j\}},$$

since $\bar{V}_j \setminus \{1, \ldots, j\}$ separates $\{1, \ldots, j\}$ from $\bar{V}_j^c$ in the graph G. Thus, a proposal distribution at step j that violates the compatibility property and relies on X_ℓ for some $\ell \notin \bar{V}_j$ will result in additional non-one terms in the product in (10) at step ℓ, so $\bar{V}_j$ is the largest set that the proposal can be allowed to depend on without increasing the number of function calls/runtime. Although not all proposals are compatible, it is a rich enough class to handle a broad range of knockoff distributions, including the distribution induced by SCIP.

With these two conditions in place, we now state our main result about the efficiency of knockoff sampling, giving an upper bound on the number of evaluations of the unnormalized density function Φ that is required by Metro when the graphical structure is known. Assuming the variable ordering from Algorithm 2 and faithful proposal distributions compatible for T such that sampling from and evaluating the proposal distributions does not require evaluating Φ, we reach the following result:

Theorem 2 (Computational efficiency of Metro). Let X be a random vector with a density which factors over a graph G as in (12). Let T be a junction tree of width w for the graph G. Under the conditions above, Metro uses $O(p2^w)$ queries of Φ.

This result means that we can efficiently implement Metropolized knockoff sampling for many interesting distributions, and it shows precisely how the complexity of the conditional independence structure of X affects the complexity of the sampling algorithm. Furthermore, in the next section we will prove that this is the optimal complexity in some cases.

4.3. Time Complexity of General Knockoff Sampling

In the previous section, we analyzed the runtime of Metro and showed that it will be tractable for graphs of sufficiently low treewidth. Now, we investigate the computational complexity of knockoff sampling in general. To formalize our investigation, we discuss a model of computation in which we have no information about the distribution of X beyond its graphical structure and the ability to query its (possibly unnormalized) density at any given point.

Formally, we consider the *oracle model*, where we are given as inputs (a) a p-dimensional vector X drawn from a density $\lambda\Phi$, where λ is a (possibly unknown) positive scalar so that we can think of Φ as an unnormalized density, (b) the support of Φ, (c) a black box capable of evaluating Φ at arbitrary query points, and (d) a graph G for which the density is known to have the form (12). No other information about Φ is available.

We show that in the oracle model with the complete graph, that is, when there is no graphical structure, knockoff sampling requires exponential time in the number of covariates, p. Please note that any complexity bound must take into account the quality of the generated knockoffs since $\tilde{X} = X$ is a trivial knockoff that can be sampled in no time.

Theorem 3 (Complexity lower bound for knockoff sampling). Consider a procedure operating in the oracle model which makes a finite number of calls to the black box Φ and returns $\tilde{X}$, thereby inducing a joint distribution $(X, \tilde{X})$ obeying the pairwise exchangeability (1) for all Φ. If G is the complete graph so that the procedure generates valid knockoffs for any input density, then the total number N of queries of Φ must obey $N \geq 2^{\#\{j: X_j \neq \tilde{X}_j\}} - 1$ a.s.

This result means that for any knockoff sampler, we cannot have both full generality and time efficiency. Put differently, to efficiently generate nontrivial knockoffs, we will need to restrict our attention to a subset of distributions for which we have structure. This fact justifies our decision to focus on distributions with graphical structure. We also derive a lower bound for the complexity of knockoff sampling for graphical models, stated next.

Corollary 2 (Complexity lower bound for graphical models). Consider the setting of Theorem 3. Fix a graph G with maximal cliques C. Suppose that for all Φ of the form $\Phi(x) = \prod_{c \in C} \phi_c(x_c)$, the procedure induces a joint distribution $(X, \tilde{X})$ obeying pairwise exchangeability (1). Then $N \geq \max_{c \in C} 2^{\#\{j \in c: X_j \neq \tilde{X}_j\}} - 1$ a.s..

This proposition shows that even after making some useful structural assumptions, there is still a trade-off between knockoff quality and computation. We next derive a byproduct, which proves that Metro is achieving a good runtime.

Proposition 2 (Optimality of Metro for chordal Gaussian graphical models). Consider continuous distributions of the form $\Phi(x) = \prod_{c \in C} \phi_c(x_c)$ over a chordal graph G.[8] On the one hand, for any input, Metro can be run with $O(p^2 + p2^w)$ queries of Φ. Furthermore, in the case where the distribution is Gaussian

[8] A chordal graph is a graph such that any cycle of length 4 or larger has a chord.

with zero mean and positive definite covariance (i.e., $\Phi(x) \propto \exp\left(-x\boldsymbol{\Sigma}^{-1}x^\top/2\right)$), Metro can produce knockoffs with $X_j \neq \tilde{X}_j$ for all j with probability 1. On the other hand, any general procedure that samples knockoffs such that $X_j \neq \tilde{X}_j$ for all j with probability $\epsilon > 0$ will require at least $2^w - 1$ queries of Φ with probability at least ϵ.

Proposition 2 means that for chordal graphs, any general knockoff sampling algorithm such that $\mathbb{P}(X_j \neq \tilde{X}_j \text{ for all } j)$ is bounded away from zero needs, in expectation, the same exponential order of queries as Metro (with the proviso that p is negligible compared to 2^w).

4.4. Divide-and-Conquer to Reduce Treewidth

Theorem 2 shows that Metro enables efficient computations for random vectors whose densities factor over a graph G of low treewidth. Not all graphs corresponding to random vectors of interest have low treewidth, however. A $d_1 \times d_2$ grid, for example, has treewidth $\min(d_1, d_2)$ (Diestel 2018). This section develops a mechanism for simplifying the graphical structure of a random vector X, allowing for faster computation of exact knockoffs at the cost of reduced knockoff quality.

To simplify graphical structure, we fix a set of variables C that separates the graph G into two subgraphs A and B. After fixing the variables in C, knockoffs can be constructed for the variables in A and B independently.

Proposition 3 (Validity of divide-and-conquer knockoffs). Suppose the sets A, B, C form a partition of $\{1, \dots, p\}$ such that C separates A and B in the graph G, that is, there is no path from some $j \in A$ to some $k \in B$ in G that does not contain some $\ell \in C$. Suppose $\tilde{X}$ is a random vector such that $X_C = \tilde{X}_C$ a.s. and for all $j_A \in A$ and $j_B \in B$,

$$(X_D, \tilde{X}_D) \overset{d}{=} (X_D, \tilde{X}_D)_{\text{swap}(j_D)} \mid X_C \qquad \text{for} \quad D = A, B.$$

Furthermore, assume we construct the knockoffs for A and B separately, that is $(X_A, \tilde{X}_A) \perp\!\!\!\perp (X_B, \tilde{X}_B) \mid X_C$. Then $\tilde{X}$ is a valid knockoff.

The divide-and-conquer technique can be applied recursively to split the graph into components of low treewidth until the junction-tree algorithm for constructing knockoffs can be used on each component. For example, for an arbitrary planar graph with p nodes, the planar separator theorem gives the existence of a subset of nodes C of size $O(\sqrt{p})$ that separates the graph into components A and B with $\max(|A|, |B|) \leq 2p/3$ (Lipton and Tarjan 1979), suggesting that this technique will apply to many cases of interest. Figure 3 illustrates this technique for a $d_1 \times d_1$ grid. We split the grid into rectangular ribbons of size $d_1 \times d_2$ for small d_2; each resulting ribbon has treewidth d_2.

The drawback of this approach is that for $j \in C$, we shall have $X_j = \tilde{X}_j$. When we think of deploying the knockoff framework in statistical applications, one should remember that we will work with multiple copies of X corresponding to distinct observations. We can then choose different separator sets for each observation so that in the end, $X_j \neq \tilde{X}_j$ for most of the observations. For example, in the setting of Figure 3, one would randomly choose between the two choices of C for each observation. This technique is explored numerically in Section 5.3.3.

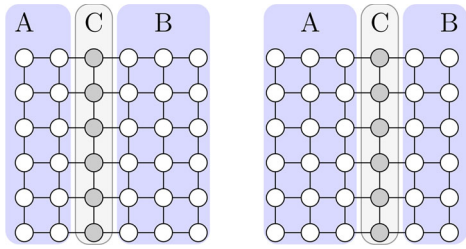

Figure 3. Two examples of conditioning to reduce the treewidth of a 6×6 grid from 6 to 3.

4.5. Discrete Distributions

For discrete distributions with a small number of states for each coordinate X_j, the junction tree techniques from Section 4.2 can be directly applied without using Metropolized knockoff sampling. When each variable X_j can take on at most K values, the probability mass function $\mathbb{P}(X_j \mid X_{-j}, \tilde{X}_{1:(j-1)})$ can be represented as a vector in $\mathbb{R}^K$, so at step j of the algorithm we simply need to evaluate $\mathbb{P}(X_j = z_j, X_{-j}, \tilde{X}_{1:(j-1)})$ for z_j in the support of X_j. This is the same quantity we computed in Section 4.2; see, for example, (10). Once these probabilities have been computed, sampling from the resulting multinomial probability gives the SCIP procedure. In principle, this can be viewed as a special case of Metro, but for a practical implementation it is simpler to work directly with the probability vectors. A similar analysis to the proof of Theorem 2 then shows that the procedure requires $O(pK^w)$ queries of the density Φ; see Appendix C.5 for details. For discrete distributions with infinite or large K, this is not tractable. However, Metro still applies and is much faster.

4.6. Knockoffs for the Ising Model

The tools from this section have the power to generate knockoffs for the Ising model on a grid (3). To construct an efficient knockoff sampler for this distribution, we need to find a junction tree of minimal width for the $d_1 \times d_2$ grid so that we can apply the technique from Section 4.5. A junction tree for the 2×3 grid of width 2 is shown in Figure 2, and the construction immediately generalizes to a junction tree of width $\min(d_1, d_2)$ for the $d_1 \times d_2$ grid, which is the optimal width. When $d_1 \geq d_2$, this leads to a knockoff sampler that proceeds from left to right, top to bottom; when variable $X_{i,j}$ is sampled, the other variables in the active node of the junction tree are $X_{i,j+1:p}$ and $X_{i+1,1:j}$; see Figure 4. Per our upper bound, this knockoff sampler will have runtime $O(d_1 d_2 2^{\min(d_1,d_2)})$. If $\min(d_1, d_2)$ is large, this runtime may still be prohibitively long, but the divide-and-conquer technique from Section 4.4 greatly increases speed at the cost of slightly worse knockoffs than the impractical full procedure. We conduct a simulation experiment of both the small-grid and large-grid setting in Section 5.3.3.

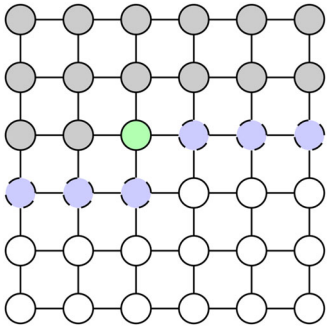

Figure 4. An illustration of sampling knockoffs for an Ising model on a grid from Section 4.6. The blue dashed nodes represent the active variables of the junction tree when variable $X_{3,3}$ (shown in green) is being sampled. Gray nodes indicate variables that have already been sampled, and white nodes indicate variables that have not been sampled yet and are not in the active node of the junction tree.

5. Numerical Experiments

We now empirically examine the Metropolized knockoff sampler, beginning with the few models where previously known samplers are available as a baseline, and then continuing on to cases with no previously known samplers. Condensed plots are presented in the main text, while more comprehensive versions can be found in Appendix F. We provide approximate runtimes with a single-core[9] implementation in either R or Python. All source code is available from *https://github.com/wenshuow/metro* with interactive computing notebooks at *http://web.stanford.edu/group/candes/metro* demonstrating the usage of the code and presenting further experimental results.

5.1. Measuring Knockoff Quality

The *mean absolute correlation* (MAC) is a useful measure of knockoff quality for a joint distribution of $(X, \tilde{X})$:

$$\mathrm{MAC}(\mathcal{L}(X,\tilde{X})) := \frac{1}{p}\sum_{j=1}^{p}|\mathrm{cor}(X_j,\tilde{X}_j)|. \tag{13}$$

We will use this as our measure of knockoff quality in our simulation experiments. Lower values of MAC are preferred. Let $\mathbf{\Gamma}$ be the correlation matrix of $(X, \tilde{X})$; pairwise exchangeability implies $\mathbf{\Gamma}$ is of the form (2). The MAC is then $\frac{1}{p}\sum_{j=1}^{p}|1-s_j|$. Since $\mathbf{\Gamma} = \mathbf{\Gamma}(s)$ has to be positive semidefinite, a lower bound on the MAC achievable by any knockoff-generation algorithm for a given distribution is the optimal value of the program

$$\min_{s} \frac{1}{p}\sum_{j=1}^{p}|1-s_j|, \text{ subject to } \mathbf{\Gamma}(s) \succeq 0. \tag{14}$$

This minimization problem can be solved efficiently with semidefinite programming (Barber and Candès 2015); we call the solution the *SDP lower bound* for the MAC. This lower bound can be achieved for Gaussian distributions (Candès et al. 2018). Valid knockoffs, however, must match *all* moments, not just the second moments, so this lower bound is not expected to be achievable in general; still it provides a useful goalpost in our simulations.

[9]The hardware varies across simulations, but each CPU is between 2.5 GHz and 3.3 GHz.

5.2. Models With Previously Known Knockoff Samplers

5.2.1. Gaussian Markov Chains

We first apply our algorithm to Gaussian Markov chains and compare with the SDP Gaussian knockoffs, whose MAC achieves the SDP lower bound exactly, and SCIP knockoffs, both from Candès et al. (2018). We take $p = 500$ features such that $X_1 \sim \mathcal{N}(0,1)$ and $X_{j+1} \mid X_{1:j} \sim \mathcal{N}(\rho_j X_j, 1-\rho_j^2)$. First, since the model is multivariate Gaussian, the covariance-guided proposal with s computed by the SDP method (14) will be identical to the SDP Gaussian knockoffs, so already a clever implementation of Metro is as good as a method specifically designed for Gaussian distributions, and since both achieve the SDP lower bound, one cannot do better in terms of MAC. Thus, we only investigate the MTM-proposals for implementing Metro. Note that the Gaussian knockoffs from Candès et al. (2018) do not use the Markovian structure of this problem, but instead rely on operations on $2p \times 2p$ matrices, whereas the MTM knockoffs from this work utilize the Markovian structure to achieve time complexity linear in p.

The results are presented in Figure 5. Following Section 3.3, we vary the number of proposals and the step size. We find that choosing the step size for X_j to be proportional to $\sqrt{1/(\mathbf{\Sigma}^{-1})_{jj}}$ gives consistent results across different sets of ρ_j's. The MTM consistently outperforms the SCIP procedure, and is reasonably close to the SDP procedure. It is observed that the defaults from Section 3.3 of eight proposals ($m = 4$) and $t_j = 1.5\sqrt{1/(\mathbf{\Sigma}^{-1})_{jj}}$ performs nearly the best in all settings. Confirming our reasoning in Section 3.3, we find that the performance stabilizes as m grows and the step size should not be too large or too small, although for sufficiently large m the MAC is fairly stable to the choice of t. In this setting, it takes around 1 sec for MTM to sample one knockoff vector with $m = 4$ and $t_j = 1.5\sqrt{1/(\mathbf{\Sigma}^{-1})_{jj}}$. The optimal Gaussian knockoffs and the SCIP knockoffs require a one-time computation of 20 sec and 90 sec, respectively. After that, each knockoff generation requires less than 0.01 sec.

5.2.2. Discrete Markov Chains

For discrete Markov Chains there is one previously-known knockoff sampler, which is an implementation of the SCIP procedure (Sesia, Sabatti, and Candès 2019). We consider here Metro with MTM proposals. (The covariance-guided proposals would require ad-hoc rounding so we do not consider this here.) We take a simple Markov Chain with $K \in \{5, 10\}$ states with uniform initial distribution and transition probabilities $Q(j, j')$ defined as

$$Q(j,j') = \frac{(1-\alpha)^{|j-j'|}}{\sum_{\ell=1}^{K}(1-\alpha)^{|j-\ell|}}. \tag{15}$$

We examine α from 0 (independent coordinates) to 0.5 (strong dependence between adjacent coordinates), with $p = 500$ features.

We examine the MTM methods across a range of values of the tuning parameters, and the results are presented in Figure 6. Full simulation results are given in Appendix F. Note that the cases with $K = 5$ and $\alpha \leq 0.15$ are tuned with the additional

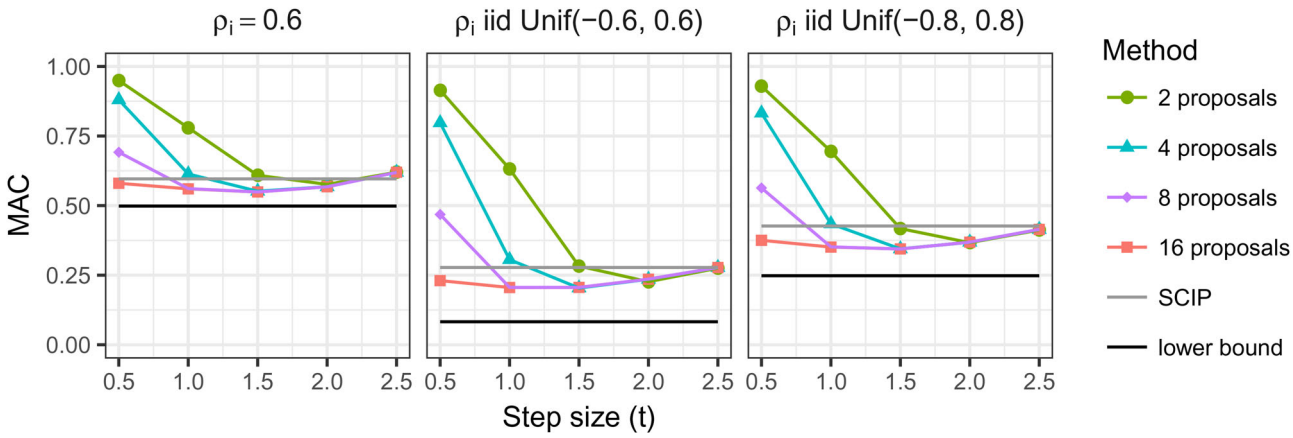

Figure 5. Simulation results for Gaussian Markov chains. The unit of step sizes is $\sqrt{1/(\boldsymbol{\Sigma}^{-1})_{jj}}$. All standard errors are below 0.001. In this case, the lower bound is achieved by the SDP Gaussian knockoffs, or equivalently, the covariance-guided proposal with an s given by the SDP (14).

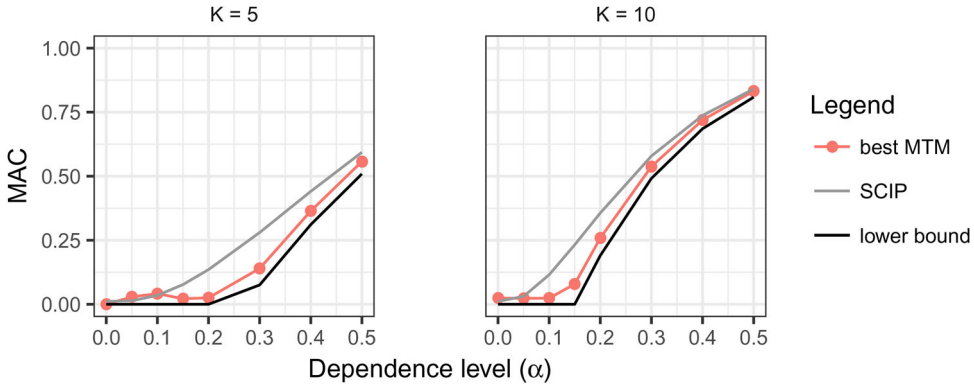

Figure 6. A comparison of the MTM procedure for discrete Markov chains with SCIP and the SDP lower bound. All standard errors are below 0.002.

parameter γ from (9), as detailed in Appendix F.2. We find that the best-tuned MTM method outperforms the SCIP method and achieves MAC near the lower bound for all dependence levels α. It takes around 0.5 sec and 0.7 sec, respectively, to run MTM ($m = 4$ and $t = 1$) for $K = 5$ and $K = 10$. It takes around 0.02 sec and 0.05 sec, respectively, to run SCIP for $K = 5$ and $K = 10$.

5.3. Models With No Previously Known Knockoff Sampler

5.3.1. Heavy-Tailed Markov Chains

As an example of a heavy-tailed distribution, we consider a Markov chain with t-distributed tails. The results are presented in Figure 7.

$$\begin{aligned} X_1 &= \sqrt{\frac{\nu-2}{\nu}} Z_1, \\ X_{j+1} &= \rho_j X_j + \sqrt{1-\rho_j^2}\sqrt{\frac{\nu-2}{\nu}} Z_{j+1}, \\ Z_j &\overset{\text{iid}}{\sim} t_\nu, \end{aligned} \tag{16}$$

for $j = 1, \ldots, p = 500$ where t_ν represents the Student's t-distribution with $\nu > 2$ degrees of freedom (note this is not a multivariate t-distribution). We try both the covariance-guided proposal with s provided by the SDP method (14) and the MTM proposals. We set $\nu = 5$ and use the same ρ_j's as in the Gaussian setting. As in Section 5.2.1, a step size of $1.5\sqrt{1/(\boldsymbol{\Sigma}^{-1})_{jj}}$ again performs well. The covariance-guided proposals also perform well, although unlike the Gaussian case, there is now a gap between the lower bound and the performance of the covariance-guided proposals. In this setting, it takes around 1.6 sec for MTM to sample one knockoff vector with $m = 4$ (eight proposals) and $t_j = 1.5\sqrt{1/(\boldsymbol{\Sigma}^{-1})_{jj}}$. For the covariance-guided proposals, it takes around 12.5 sec for the one-time computation of the parameters (excluding time used for computing s, which varies depending on the method; the most expensive one is the SDP, which takes 20 sec) and then 0.3 sec to sample each knockoff vector.

5.3.2. Asymmetric Markov Chains

As an example of asymmetric, continuous distributions, we take a standardized equal mixture of Gaussian and exponential random variables and then form a Markov chain. The results are presented in Figure 8. Explicitly,

$$Z_j \overset{\text{iid}}{\sim} \frac{I \cdot \mid Y_G \mid -(1-I) \cdot Y_E - \mu}{\sigma} \quad \text{for } j = 1, \ldots, p = 500,$$

where $Y_G \sim \mathcal{N}(0,1)$, $Y_E \sim \text{Expo}(1)$, and $I \sim \text{Bern}(1/2)$ are independent. The parameters μ and σ are chosen so that Z_j has mean 0 and variance 1. We then take

$$X_1 = Z_1, \quad X_{j+1} = \rho_j X_j + \sqrt{1-\rho_j^2} Z_{j+1} \quad \text{for } j = 2, \ldots, p.$$

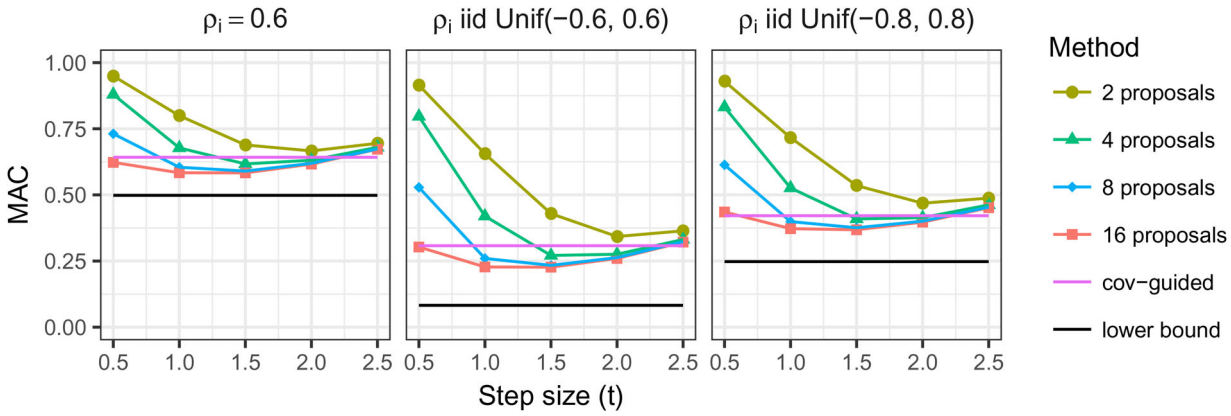

Figure 7. Simulation results for the t-distributed Markov chains. The unit of step sizes is $\sqrt{1/(\boldsymbol{\Sigma}^{-1})_{jj}}$. All standard errors are below 0.001.

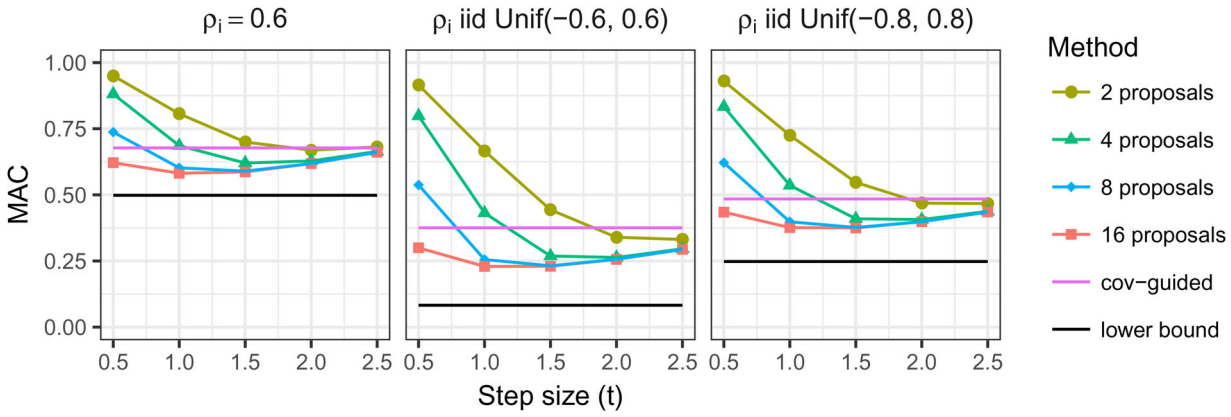

Figure 8. Simulation results for the asymmetric Markov chains. The unit of step sizes is $\sqrt{1/(\boldsymbol{\Sigma}^{-1})_{jj}}$. All standard errors are below 0.001.

We examine both the covariance-guided proposal with s provided by the SDP (14) and the multiple-try proposals. We use the same ρ_j's as in the Gaussian setting. As in the previous case, $m = 4$ (eight proposals) and $t_j = 1.5\sqrt{1/(\boldsymbol{\Sigma}^{-1})_{jj}}$ performs essentially as well as any other MTM parameter choices, and significantly outperforms the covariance-guided proposals. The timing results are the same as in the heavy-tailed Markov chains.

5.3.3. Ising Model

In this section, we consider an Ising model over a square grid (3). We generate knockoffs with the method for discrete random variables from Section 4.5 combined with the divide-and-conquer technique, the combination of which was described for Ising models in Section 4.6; no other exact knockoff samplers are known for the Ising model. Although our sampling procedures for the Ising model do not explicitly use the Metropolis–Hastings step, as explained in Section 4.5, we will refer to the sampler as "Metro" in this section for simplicity.

First, we take a 10×10 grid and set all $\beta_{i,j,i',j'} = \beta_0$ and all $\alpha_{i,j} = 0$. The results are presented in Figure 9. The left panel shows how the MAC increases—or, the quality decreases—as the dependence between adjacent variables—β_0—increases. We see that the procedure is close to the lower bound for large β_0. In the middle panel, we plot $\text{cor}(X_{j,k}, \tilde{X}_{j,k})$ across different coordinates (j, k). We see that on the edges of the grid, especially on the corners, knockoffs have lower absolute correlation with their original counterparts. These variables are less determined by the values of the rest of the grid, so this is expected. In this setting, it takes about 12 sec to sample a knockoff.

Next, we demonstrate the divide-and-conquer technique from Section 4.4. Here we consider the Ising model from above on a 100×100 grid, for a total dimension of 10,000. The 100×100 grid has treewidth 100, so Metro would not be tractable without the divide-and-conquer technique. We divide the graph into subgraphs of width w, by fixing entire columns as in Figure 3. To measure the effect of the slicing, we compute the MAC on the interior points and compare this to the MAC of the interior points of a smaller grid for a procedure without slicing, see Appendix F.3 for details. We find that the quality of the knockoffs increases as we take larger slices, as expected. Furthermore, even modest values of w such as $w = 5$ result in a procedure that achieves a MAC close to that of the baseline. Recall that the complexity of Metro scales as 2^w, so fixing $w = 5$ dramatically reduces the computation time compared to $w = 100$. With $w = 5$, it takes about 2.5 min to generate one knockoff for the 100×100 grid.

5.3.4. Gibbs Measure on a Grid

Lastly, we demonstrate the MTM proposals simultaneously with the junction tree techniques for complex dependence structure. Consider a Gibbs measure on $\{1, \ldots, K\}^{d \times d}$, with a probability mass function

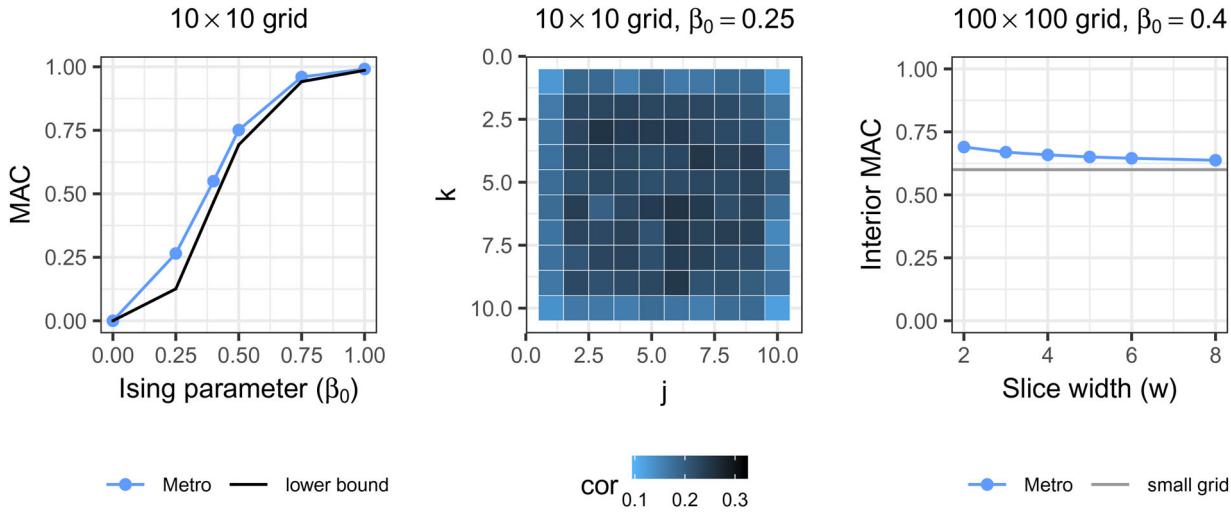

Figure 9. Results of the Ising model experiments. All standard errors in the line plots are less than 0.005.

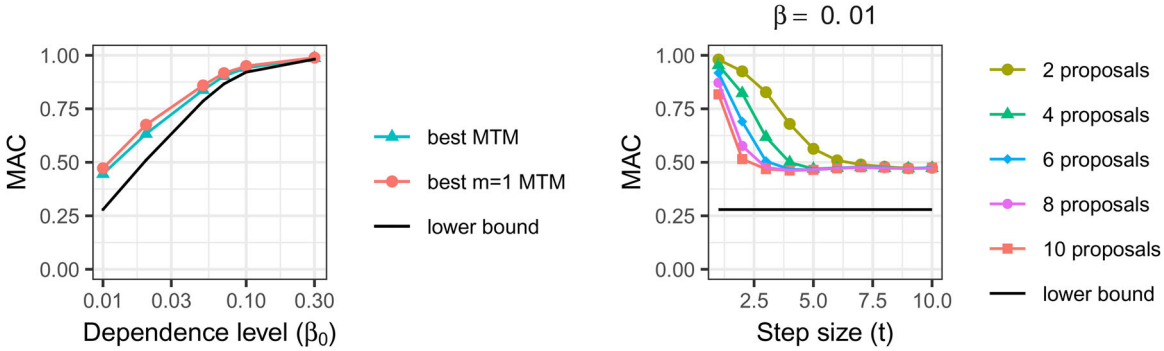

Figure 10. Results of the Gibbs measure experiments. All standard errors are below 0.002. In the left panel, β_0 is shown in logarithmic scale.

$$\mathbb{P}(X) = \frac{1}{Z(\beta_0)} \exp\left(-\beta_0 \sum_{\substack{s,t\in\mathcal{I} \\ \|s-t\|_1=1}} (x_s - x_t)^2\right),$$
$$\mathcal{I} = \{(i_1, i_2) : 1 \le i_1, i_2 \le d\},$$

and note that like the Ising model, this density factors over the grid. For our experiment, we take a 10×10 grid and examine different dependence levels β_0 with $K = 20$ possible states for each variable. We apply Metro with the MTM proposals and the divide-and-conquer technique on the grid, tuning the procedure across a range of parameters as detailed in Appendix F. The condensed results are given in Figure 10. We do not know of another knockoff sampler in this setting. Having said this, we observe that our procedure has MAC close to the lower bound. We also observe that in the case where $w = 3$, with as few as two proposals, our procedure performs well and takes about half a second to generate a knockoff copy; when we increase the number of proposals to ten, the compute time is around 2 min. When w is set to 5, the slowest setting is $m = t = 1$, which takes less than 4 min.

5.3.5. Potts Model in Protein Contact Prediction

To demonstrate Metro in a case with an even more complex graphical structure, we apply it to a Potts model over a graph arising in protein residual contact prediction (see, e.g., Weigt et al. 2009; Marks et al. 2011; Ekeberg et al. 2013). In this line of work, researchers model the distribution of an amino acid sequence with length p, $X = (X_1, X_2, \ldots, X_p) \in \{0, 1, \ldots, 20\}^p$, as

$$\mathbb{P}_{h,J}(X) = \frac{1}{Z(h,J)} \exp\left(\sum_{j=1}^{p} h_i(X_i) + \sum_{1\le i<j\le p} J_{ij}(X_i, X_j)\right),$$

where the 21 states represent 20 possible amino acids and one gap, each h_i is a 21-dimensional vector, each J_{ij} is a 21×21 matrix, and $Z(h, J)$ is a normalizing constant. The studies above fit this model to identify sites far away that are dependent, which suggests that they are nearby in three-dimensional space, but in our case we are interested in this model only as an example of a complex graphical model arising in a scientific application. Using the method and data in Ekeberg et al. (2013), we estimate such a model for protein family PF00006 which has $p = 213$ features. The graphical model we estimate admits a junction tree of width 9, which indicates that the graph is moderately complex, and unlike in previous examples it does not have a simple description; see Figure 11 for a visualization. Nonetheless, we can construct knockoffs with the Metro algorithm. Using a proposal distribution that is uniform across the 21 possible states, sampling one knockoff takes about 10 sec, and using 5000 independent samples we find that the resulting MAC is 0.26, which is reasonably close to the lower bound of 0.14 for this distribution. Note that in this example, the variables are categorical, so the MAC depends on the encoding of the

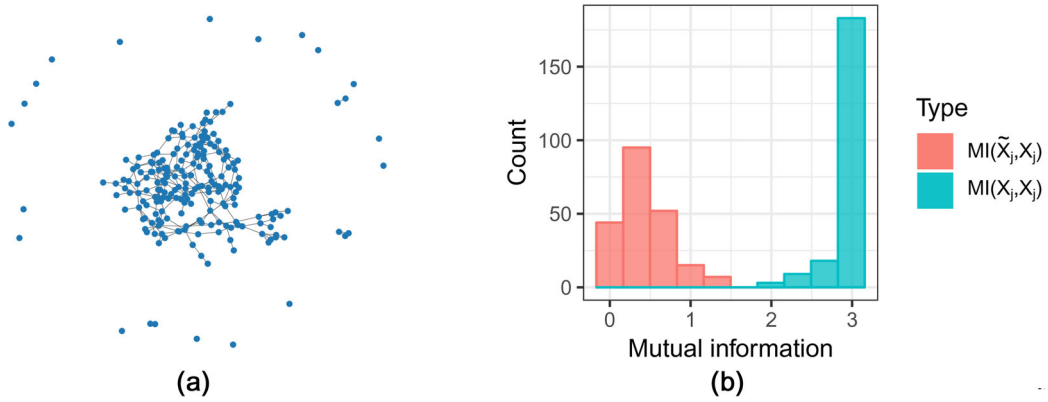

Figure 11. Results of the protein residual contact example. (a) The fitted graph. (b) The mutual information between X_j and $\tilde{X}_j$, a measure of knockoff quality. As a point of reference, the mutual information between X_j and X_j (i.e., the entropy) is also plotted.

variables. Because of this, in Figure 11 we instead plot the empirical mutual information between X_j and $\tilde{X}_j$, which is an alternative measure of knockoff quality that does not depend on the choice of encoding. Here, we find that the mutual information is very small relative to the mutual information of the baseline of $\tilde{X} = X$. As with our MAC metric, this again indicates that the knockoffs we have constructed have high contrast with the original variables, which was our aim. Lastly, we emphasize that this experiment was carried out with our general-purpose software that samples knockoffs while only requiring the user to specify (i) a function evaluating the unnormalized density, (ii) a graph encoding the conditional independence structure, and (iii) a symmetric proposal distribution. As such, this method is straightforward to use and is ready for deployment in scientific settings. We again refer the reader to our notebook tutorials at *http://web.stanford.edu/group/candes/metro*.

6. Discussion

This article introduced a sequential characterization of all valid knockoff-generating procedures and used it along with ideas from MCMC and graphical models to create Metropolized knockoff sampling, an algorithm which generates valid knockoffs in complete generality with access only to X's unnormalized density. Although we proved in Theorem 3 that no algorithm (including Metro) can sample exact knockoffs *efficiently* for arbitrary X distributions, we characterized one way out of this impossibility result: conditional independence structure in X. An interesting future direction would be to establish other sufficient conditions on a model family that would allow one to sample knockoffs efficiently. Another way out of the lower bound in Theorem 3 is to forgo exact knockoffs and settle for approximations. Although this arguably is a tall order, it would be interesting to establish theoretical guarantees on the approximation quality of these or other approximate knockoff constructions, and better understand the tradeoff between knockoff approximation quality and time complexity.

Supplementary Materials

Appendix: Appendix of the article. (.pdf file)

Acknowledgments

S. B. and E.C. would like to thank Yaniv Romano and Matteo Sesia for useful comments on an early version of this work. L. J. and W. W. would like to thank Jun Liu for fruitful discussions on MCMC.

Funding

E. C. was partially supported by the Office of Naval Research under grant N00014-16-1-2712, by the National Science Foundation via DMS 1712800, and by a generous gift from TwoSigma. S. B. was supported by a Ric Weiland Graduate Fellowship.

References

Anderson, T. W. (2003), *An Introduction to Multivariate Statistical Analysis* (3th ed.), Hoboken, New Jersey: Wiley. [6]

Arnborg, S., Corneil, D. G., and Proskurowski, A. (1987), "Complexity of Finding Embeddings in a k-Tree," *SIAM Journal on Algebraic Discrete Methods*, 8, 277–284. [7]

Barber, R. F., and Candès, E. J. (2015), "Controlling the False Discovery Rate via Knockoffs," *The Annals of Statistics*, 43, 2055–2085. [1,2,5,10]

Bertele, U., and Brioschi, F. (1972), *Nonserial Dynamic Programming*, Orlando, FL: Academic Press, Inc. [7]

Candès, 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, 80, 551–577. [1,2,3,5,10]

Dai, R., and Barber, R. (2016), "The Knockoff Filter for FDR Control in Group-Sparse and Multitask Regression," in *Proceedings of the 33rd International Conference on Machine Learning, Volume 48 of Proceedings of Machine Learning Research (PMLR)*, pp. 1851–1859. [5]

Diestel, R. (2018), *Graph Theory* (5th ed.), Berlin, Heidelberg: Springer Publishing Company, Inc. [9]

Ekeberg, M., Lövkvist, C., Lan, Y., Weigt, M., and Aurell, E. (2013), "Improved Contact Prediction in Proteins: Using Pseudolikelihoods to Infer Potts Models," *Physical Review E*, 87, 012707. [13]

Fan, Y., Lv, J., Sharifvaghefi, M., and Uematsu, Y. (2018), "IPAD: Stable Interpretable Forecasting With Knockoffs Inference," available at SSRN 3245137. [2]

Gao, C., Sun, H., Wang, T., Tang, M., Bohnen, N. I., Müller, M. L. T. M., Herman, T., Giladi, N., Kalinin, A., Spino, C., Dauer, W., Hausdorff, J. M., and Dinov, I. D. (2018), "Model-Based and Model-Free Machine Learning Techniques for Diagnostic Prediction and Classification of Clinical Outcomes in Parkinson's Disease," *Scientific Reports*, 8, 7129. [2]

Gimenez, J. R., Ghorbani, A., and Zou, J. (2018), "Knockoffs for the Mass: New Feature Importance Statistics With False Discovery Guarantees," arXiv no. 1807.06214. [2]

Hammersley, J. M., and Clifford, P. E. (1971), "Markov Random Fields on Finite Graphs and Lattices," unpublished manuscript. [7]

Hastings, W. K. (1970), "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika*, 57, 97–109. [4]

Ising, E. (1925), "Beitrag zur theorie des ferromagnetismus," *Zeitschrift für Physik*, 31, 253–258. [3]

Jordon, J., Yoon, J., and van der Schaar, M. (2019), "KnockoffGAN: Generating Knockoffs for Feature Selection Using Generative Adversarial Networks," in *International Conference on Learning Representations*. [2]

Kjærulff, U. (1990), "Triangulation of Graphs—Algorithms Giving Small Total State Space," Technical Report. [7]

Koller, D., and Friedman, N. (2009), *Probabilistic Graphical Models: Principles and Techniques*, Adaptive Computation and Machine Learning, Cambridge, MA: MIT Press. [7]

Lipton, R. J., and Tarjan, R. E. (1979), "A Separator Theorem for Planar Graphs," *SIAM Journal on Applied Mathematics*, 36, 177–189. [9]

Liu, J. S., Liang, F., and Wong, W. H. (2000), "The Multiple-Try Method and Local Optimization in Metropolis Sampling," *Journal of the American Statistical Association*, 95, 121–134. [5]

Liu, Y., and Zheng, C. (2018), "Auto-Encoding Knockoff Generator for FDR Controlled Variable Selection," arXiv no. 1809.10765. [2]

Lu, Y., Fan, Y., Lv, J., and Noble, W. S. (2018), "DeepPINK: Reproducible Feature Selection in Deep Neural Networks," in *Advances in Neural Information Processing Systems*, pp. 8689–8699. [2]

Marks, D. S., Colwell, L. J., Sheridan, R., Hopf, T. A., Pagnani, A., Zecchina, R., and Sander, C. (2011), "Protein 3D Structure Computed From Evolutionary Sequence Variation," *PLoS One*, 6, e28766. [13]

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953), "Equation of State Calculations by Fast Computing Machines," *The Journal of Chemical Physics*, 21, 1087–1092. [4]

Qin, Z. S., and Liu, J. S. (2001), "Multipoint Metropolis Method With Application to Hybrid Monte Carlo," *Journal of Computational Physics*, 172, 827–840. [5,6]

Romano, Y., Sesia, M., and Candès, E. (2018), "Deep Knockoffs," arXiv no. 1811.06687. [2]

Sesia, M., Sabatti, C., and Candès, E. J. (2019), "Gene Hunting With Hidden Markov Model Knockoffs," *Biometrika*, 106, 1–18. [2,10]

Weigt, M., White, R. A., Szurmant, H., Hoch, J. A., and Hwa, T. (2009), "Identification of Direct Residue Contacts in Protein–Protein Interaction by Message Passing," *Proceedings of the National Academy of Sciences of the United States of America*, 106, 67–72. [13]

Xiao, Y., Angulo, M., Friedman, J., Waldor, M., Weiss, S., and Liu, Y.-Y. (2017), "Mapping the Ecological Networks of Microbial Communities From Steady-State Data," *bioRxiv*, 8, 150649. [2]

Zheng, Z., Zhou, J., Guo, X., and Li, D. (2018), "Recovering the Graphical Structures via Knockoffs," *Procedia Computer Science*, 129, 201–207. [2]