HHS Public Access

Author manuscript

Biometrika. Author manuscript; available in PMC 2023 July 06.

Published in final edited form as:

Biometrika. 2022 June ; 109(2): 277–293. doi:10.1093/biomet/asab039.

Fast and powerful conditional randomization testing via distillation

MOLEI LIU,

Department of Biostatistics, Harvard Chan School of Public Health, 677 Huntington Avenue, Boston, Massachusetts 02115, U.S.A.

EUGENE KATSEVICH,

Department of Statistics and Data Science, Wharton School of the University of Pennsylvania, 265 South 37th Street, Philadelphia, Pennsylvania 19104, U.S.A.

LUCAS JANSON,

Department of Statistics, Harvard University, One Oxford Street, Cambridge, Massachusetts 02138, U.S.A.

AADITYA RAMDAS

Department of Statistics & Data Science, Carnegie Mellon University, 132H Baker Hall, Pittsburgh, Pennsylvania 15213, U.S.A.

Summary

We consider the problem of conditional independence testing: given a response Y and covariates (X, Z), we test the null hypothesis that $Y \perp\!\!\!\perp X \mid Z$. The conditional randomization test was recently proposed as a way to use distributional information about $X \mid Z$ to exactly and nonasymptotically control Type-I error using any test statistic in any dimensionality without assuming anything about $Y \mid (X, Z)$. This flexibility, in principle, allows one to derive powerful test statistics from complex prediction algorithms while maintaining statistical validity. Yet the direct use of such advanced test statistics in the conditional randomization test is prohibitively computationally expensive, especially with multiple testing, due to the requirement to recompute the test statistic many times on resampled data. We propose the distilled conditional randomization test, a novel approach to using state-of-the-art machine learning algorithms in the conditional randomization test while drastically reducing the number of times those algorithms need to be run, thereby taking advantage of their power and the conditional randomization test's statistical guarantees without suffering the usual computational expense. In addition to distillation, we propose a number of other tricks, like screening and recycling computations, to further speed up the conditional randomization test without sacrificing its high power and exact validity. Indeed, we show in simulations that all our proposals combined lead to a test that has similar power to the most powerful existing conditional randomization test implementations, but requires orders of magnitude less computation, making it

molei_liu@g.harvard.edu .

Supplementary material

Supplementary Material available at *Biometrika* online contains further comparisons, simulation results and details of the breast cancer data analysis.

a practical tool even for large datasets. We demonstrate these benefits on a breast cancer dataset by identifying biomarkers related to cancer stage.

Some key words:

Conditional independence test; Conditional randomization test; High-dimensional inference; Machine learning; Model-X

1. Introduction

1.1. Background

In our increasingly data-driven world, it has become the norm, in applications from genetics and health care to policy evaluation and e-commerce, to seek to understand the relationship between a response variable of interest and a high-dimensional set of potential explanatory variables or covariates. While accurately estimating this entire relationship would generally require a nearly infinite sample size, a less intractable, but still extremely useful question is to ask, for any given covariate, whether it actually contributes to the response variable's high-dimensional conditional distribution. We address this problem by encoding a covariate's relevance as its conditional dependence with the response, which can be defined without requiring any modelling assumptions. Denoting the response random variable by Y, a given covariate of interest by X, and a multidimensional set of further covariates by $Z = (Z_1, \ldots, Z_p)$, the null hypothesis we seek to test is $H_0 : Y \perp\!\!\!\perp X \mid Z$ against the alternative $H_1 : Y \not\perp\!\!\!\perp X \mid Z$. Testing this hypothesis for just a single covariate is sometimes all that is needed, such as in an observational study investigating whether a particular treatment (X) causes a change in a response (Y) after controlling for a set of measured confounding variables (Z). But in other applications no one covariate holds a priori precedence over another, and a researcher seeks any and all covariates that contribute to Y's conditional distribution. This variable selection objective can also be achieved by testing H_0 for each covariate in turn and plugging the resulting p-values into one of the many procedures from the extensive literature on multiple testing. In addition to the considerable statistical challenge of providing a valid and powerful test of H_0, it is of paramount importance to also ensure that the test is computationally efficient, especially, as is often the case in modern applications, when either or both the sample size and dimension are large, and even more so when a variable selection objective requires the test to be run many times. Thus, the goal of this paper is to present a test for conditional independence that is provably valid, empirically powerful and computationally efficient when the distribution of $X \mid Z$ is known or can be well approximated.

Our work builds on the conditional randomization test, CRT, introduced by Candès et al. (2018). The CRT is a general framework for conditional independence testing that can use any test statistic one chooses, and exactly and nonasymptotically control the Type-I error regardless of the data dimensionality. The CRT's guarantees assume nothing whatsoever about $Y \mid (X, Z)$, but instead assumes $X \mid Z$ is known. This so-called model-X framework, in contrast to the canonical approach of assuming a strong model for $Y \mid (X, Z)$, is perhaps

easiest to justify when a wealth of unlabelled data, i.e., pairs (X_i, Z_i) without corresponding Y_i, is available, but has also been found to be quite robust even when $X \mid Z$ is estimated using only the labelled data.

In order to define the CRT, we first need notation for our data. For $i \in \{1, \ldots, n\}$, let $(Y_i, X_i, Z_i) \in \mathbb{R}^{p+2}$ be independent and identically distributed copies of (Y, X, Z), and denote the column vector of the Y_i by $y \in \mathbb{R}^n$, the column vector of the X_i by $x \in \mathbb{R}^n$, and the matrix whose rows are the Z_i by $\mathbb{Z} \in \mathbb{R}^{n \times p}$. The CRT is given by Algorithm 1, and its Type-I error guarantee follows.

Algorithm 1. The conditional randomization test.

Input: The distribution of $x \mid \mathbb{Z}$, data $(y, x, \mathbb{Z})$, test statistic function T, number of randomizations M.

For $m = 1, 2, \ldots, M$: Sample $x^{(m)}$ from the distribution of $x \mid \mathbb{Z}$, conditionally independently of x and y.

Output: CRT p-value $\frac{1}{M+1}\left[1 + \sum_{m=1}^{M} 1_{\{T(y, x^{(m)}, \mathbb{Z}) \geq T(y, x, \mathbb{Z})\}}\right]$.

Theorem 1 (Candès et al. (2018)). *The CRT p-value* $p(y, x, \mathbb{Z})$ *satisfies* $P_{H_0}(p(y, x, \mathbb{Z}) \leqslant \alpha) \leqslant \alpha$ *for all* $\alpha \in [0, 1]$.

For many common models of $X \mid Z$, the conditionally independent sampling of $x^{(m)}$ is straight-forward. And even in more complex models it is still often easy to sample $x^{(m)}$ conditionally exchangeably with x and conditionally independently of y, for instance by conditioning on an inferred latent variable, which is sufficient for Theorem 1 to hold. Because Theorem 1 only relies on the exchangeability of the vectors $x, x^{(1)}, \ldots, x^{(M)}$ under H_0, it is entirely agnostic to the choice of test statistic T. This enables some very powerful choices, such as test statistics derived from modern machine learning algorithms, from Bayesian inference, though neither the prior nor model for $Y \mid (X, Z)$ need be well-specified, or from highly domain-specific knowledge or intuition. Unfortunately, the most powerful statistics are often particularly expensive to compute and, as can be seen from Algorithm 1, T must be applied $M + 1$ times in order to compute a single p-value. When testing all the covariates at once, this computational problem is compounded since, not only does each test require $M + 1$ applications of T, but M must be roughly of order p to ensure the p-values are sufficiently high resolution to make any discoveries with multiple-testing procedures such as Benjamini–Hochberg (Benjamini & Hochberg, 1995).

1.2. Our contribution

We resolve this computational challenge in § 2 by introducing a technique we call distillation that can still leverage any high-dimensional modelling or supervised learning algorithm, but presents dramatic computational savings by only requiring the expensive high-dimensional computation to be performed once, instead of $M + 1$ times. We call our

proposed method the distilled CRT, or dCRT, and we show how to further improve its computation in multiple-testing settings in § 3.

In this paper we will refer to the CRT implementation as originally proposed without distillation or HRT, holdout randomization test, speedup as the original CRT, or oCRT. We demonstrate in simulations in § 4 that there is little difference in power between the dCRT and its more expensive CRT counterpart, and what small differences exist can be explained by factors that are separate from distillation. Meanwhile, our proposals save orders of magnitude in computation over the oCRT even for medium-scale problems. We also show in simulations that the dCRT is comparably powerful to other state-of-the-art conditional independence tests, and is also robust to misspecification in the distribution of X.

The dCRT inherits several attractive properties of the CRT: it can be derandomized to an arbitrary extent through computation with increasing M and yields finite-sample valid p-values for all variables that can be used for downstream multiple-testing analyses with a variety of error metrics, including not only the false discovery rate, but also the familywise error rate and others.

1.3. Related work

Our work builds upon the CRT framework of Candès et al. (2018), with the goal of making it computationally tractable without sacrificing power. Our work is perhaps most similar in its goal to the HRT of Tansey et al. (2021), which uses data splitting to enable the use of complex modelling in the CRT with far less computation by doing all the complex modelling on the first part of the data and testing on the second part. A domain-specific version of the HRT was applied by Bates et al. (2020) to genetic trio studies by using causal terminology, learning a model on observational data and using it within the CRT on randomized experimental data; the power of a similar hybrid CRT approach was studied by Katsevich & Ramdas (2021). We show in § 4 that data splitting comes with a substantial power loss compared to the dCRT and oCRT. Tansey et al. (2021) addressed this with cross-fitting, but in doing so lost the guarantee on Type-I error control of the CRT and dCRT. Other works have extended the CRT (Bellot & van der Schaar, 2019; Berrett et al., 2020) in ways that do not address its computational intractability. For the variable selection problem, model-X knockoffs (Candès et al., 2018) can simultaneously test conditional independence for each covariate, yielding a false discovery rate-controlling rejection set. Model-X knockoffs is inherently a multiple-testing method, with power to detect groups of nonnull variables without quantifying their individual significances. On the other hand, the dCRT is a single-testing method which can be paired with multiple-testing procedures if desired. We elaborate further on the comparison between dCRT and model-X knockoffs in the discussion.

We note a pair of methods, double machine learning (Chernozhukov et al., 2018) and the generalized covariance measure (Shah & Peters, 2018), that both test conditional independence under assumptions that nearly, but not quite due to moment conditions on Y, subsume ours, and whose test statistic resembles and can even be identical to certain special cases of the dCRT. However, their statistics only resemble a special case of the dCRT; the

dCRT framework includes many other statistics which deviate substantially from double machine learning/generalized covariance measure and can be more powerful in certain settings. Furthermore, the cut-offs for their test statistics are both based on asymptotic normality, while the dCRT is nonasymptotically exact regardless of the distribution of its test statistic; see the Supplementary Material.

1.4. Notation

Let $I = (i_i, i_2, \ldots, i_k) \subseteq \{1, \ldots, n\}$ and $J = (j_1, j_2, \ldots, j_\ell) \subseteq \{1, 2, \ldots, p\}$ be subsets of samples and variables, respectively, and consider a matrix $A = (a_1, a_2, \ldots, a_p) \in \mathbb{R}^{n \times p}$ with $a_j = (A_{1j}, A_{2j}, \ldots, A_{nj})^{\mathrm{T}}$. We denote by $A_{I,J}$ the submatrix of A with rows in I and columns in J. We use the subscripts j, $-J'$ and $\bullet$ as shorthand for $J = \{j\}$, $\{1, \ldots, p\} \setminus J'$ and $\{1, \ldots, p\}$, respectively, and the same for the first index. For example, $A_{\bullet, -j}$ represents the matrix A with the jth column removed. For any two vectors a_j and a_ℓ, let $a_j \odot a_\ell = (A_{1j}A_{1\ell}, A_{2j}A_{2\ell}, \ldots, A_{nj}A_{n\ell})^{\mathrm{T}}$ denote their elementwise product, and for $L = \{\ell_1, \ell_2, \ldots, \ell_k\}$ let $a_j \odot A_L = (a_j \odot a_{\ell_1}, \ldots, a_j \odot a_{\ell_k})$; these will be used when fitting interaction effects.

2. The distilled conditional randomization test

2.1. Main idea

It is natural to derive CRT test statistics from machine learning methods with high predictive and estimation accuracy. Indeed, the original paper proposing the CRT (Candès et al., 2018) used the test statistic $T_{\mathrm{oCRT}}(y, x, \mathbb{Z}) := \left|\hat{\beta}_x^{\mathrm{lasso}}\right|$, the absolute value of the fitted coefficient on x from the lasso (Tibshirani, 1996) of y on $(x, \mathbb{Z})$ with penalty parameter chosen by cross-validation. Although powerful and computationally much faster than many other machine learning algorithms, it is still expensive to repeatedly run the lasso on large datasets hundreds or more times just to compute a single CRT p-value, and many times more than that in multiple-testing scenarios when a CRT p-value for each covariate is needed.

Consider now the following alternative test statistic which captures the essence of our proposal. First fit a cross-validated lasso of y on only $\mathbb{Z}$ to obtain the p-dimensional coefficient vector $\hat{\beta}_z^{\mathrm{loco}}$. Then fit a least-squares regression of the residual $\left(y - \mathbb{Z}\hat{\beta}_z^{\mathrm{loco}}\right)$ on x to obtain the scalar coefficient $\hat{\beta}_x^{\mathrm{loco}}$ and take its absolute value $T_{\mathrm{dCRT}}(y, x, \mathbb{Z}) := \left|\hat{\beta}_x^{\mathrm{loco}}\right|$ as the test statistic. Here, the superscript loco represents leave-one-covariate-out regression as x is left out when regressing y solely on $\mathbb{Z}$. We introduce this notation to distinguish the leave-one-covariate-out construction from the oCRT lasso statistics when needed, although in the remainder of the paper we will just use $\left(\hat{\beta}_x, \hat{\beta}_z\right)$ to represent $\left(\hat{\beta}_x^{\mathrm{loco}}, \hat{\beta}_z^{\mathrm{loco}}\right)$ when there is no need to distinguish them from $\left(\hat{\beta}_x^{\mathrm{lasso}}, \hat{\beta}_z^{\mathrm{lasso}}\right)$. It may seem as though little has changed from the preceding paragraph; we would expect T_{oCRT} and T_{dCRT} to have similar statistical properties and require nearly the same computation. Although the statistical properties of T_{oCRT} and T_{dCRT} are indeed very similar and they do require nearly the same time to compute once, they require dramatically different computation within the CRT. The key difference is that

the expensive $(p+1)$-dimensional lasso fit in T_{oCRT} must be recomputed for each resample of x, while the expensive p-dimensional lasso fit in T_{dCRT} must only be computed once, since that lasso does not depend on x and hence is identical for all its resamples. In the CRT, neither y nor $\mathbb{Z}$ change during the resampling procedure, and we take advantage of this by applying our expensive computation to only y and $\mathbb{Z}$ so it only has to be done once. All that is required for each resample's computation of T_{dCRT} is a univariate regression, whose computational expense is much lower than a p-dimensional lasso.

We can generalize this idea far beyond the lasso or linear regressions. The core proposal is to distil all the high-dimensional information in $\mathbb{Z}$ about y into a low-dimensional representation, without looking at x. Then the test statistic estimates a relationship between x and the leftover information in y by only looking at x, y and the distilled low-dimensional function of $\mathbb{Z}$. Thus, all the computation on high-dimensional data, namely the distillation, only needs to be performed once, while the computation that is repeatedly applied to the resampled data is low-dimensional and hence relatively fast.

2.2. Formal presentation of dCRT

We now formalize the idea from the previous subsection in Algorithm 2.

Algorithm 2. The distilled conditional randomization test.

Input: The distribution of $x \mid \mathbb{Z}$, data $(y, x, \mathbb{Z})$, y-distillation-fitting function $\mathscr{D}_y$, x-distillation function $\mathscr{D}_x$, test statistic function T, number of randomizations M.

Distil $\mathbb{Z}$'s information about y into $d_y = \mathscr{D}_y(y, \mathbb{Z})$ and about x into $d_x = \mathscr{D}_x(\mathbb{Z})$.

For $m = 1, 2, \ldots, M$: Sample $x^{(m)}$ from the distribution of $x \mid \mathbb{Z}$, conditionally independently of x and y.

Output: dCRT p-value $\frac{1}{M+1}\left[1 + \sum_{m=1}^{M} 1_{\left\{T\left(y, x^{(m)}, d_y, d_x\right) \geq T\left(y, x, d_y, d_x\right)\right\}}\right]$.

The key difference from the more general CRT in Algorithm 1 is that the test statistic function T in Algorithm 2 only sees information about the high-dimensional $\mathbb{Z}$ through its y- and x-distillations d_y and d_x, which are both computed just once in the first line of the algorithm. The functions $\mathscr{D}_y$ and $\mathscr{D}_x$ should be chosen such that the distillation step produces d_y and d_x with dimension much less than p, so that T's inputs are low-dimensional. Then, since T is the only repeatedly applied function and its computation does not suffer from the high-dimensionality of the original data, the dCRT's computation will be dominated by the single application of $\mathscr{D}_y$. For instance, in the dCRT example in § 2.1, d_x is not used and $\mathscr{D}_y$ fits a lasso of y on $\mathbb{Z}$ and returns $d_y = \mathbb{Z}\hat{\beta}_z$, while $T(y, x, d_y) = \left|(y - d_y)^{\mathrm{T}} x\right| / \| x \|^2$ requires negligible computation by comparison.

We emphasize that $\mathscr{D}_y$ can really be any regression algorithm and Theorem 1 still holds, since for any choice of $\mathscr{D}_y$ the dCRT is still a special case of the CRT. Thus, it can take advantage of the predictive power of state-of-the-art machine learning algorithms, precise

knowledge in the form of a Bayesian prior, or even imprecise domain expertise or intuition applied by trying many different regressions of y on $\mathbb{Z}$ and choosing whichever feels best as long as x is not factored into that decision. In the following we provide some suggestions and default choices.

2.3. d_0CRT : Fast, powerful and intuitive

The most computationally efficient and intuitive class of dCRT procedures has both y- and x-distillations reduce $\mathbb{Z}$ to an output with a single column. We label this subclass of dCRT procedures as d_0CRT because it represents the choice to maximally distil each row of $\mathbb{Z}$ down to a single scalar. Assuming T's computation generally increases with the dimension of its inputs, the d_0CRT also represents a particularly computationally efficient class of dCRTs.

A natural approach to constructing a d_0CRT, especially when Y is continuous, is to have distillation take the form of conditional mean functions. That is, let $\mathcal{D}_x(\mathbb{Z}) = E[x \mid \mathbb{Z}]$ and have $\mathcal{D}_y$ fit an estimate of the analogous regression function for y, i.e., $\mathcal{D}_y(y, \mathbb{Z}) \approx E[y \mid \mathbb{Z}]$. Then T can be chosen as an empirical measure of dependence between the residuals $y - d_y$ and $x - d_x$, such as the square of the fitted coefficient when regressing the former on the latter. This approach is also easy to understand and implement since it just requires choosing $\mathcal{D}_y$ and T, with $\mathcal{D}_y$ just performing a possibly nonparametric regression, while T can be thought of as computing a test statistic for testing the independence between two scalar random variables from a paired sample of size n: $(y - d_y, x - d_x)$. As both regression and bivariate independence testing are highly studied topics, users can easily draw from their statistical training, domain expertise and a rich literature in order to design an appropriate d_0CRT for their particular problem. The following is a generic example we have found to be computationally efficient and powerful in our simulations.

Example 1 (*Lasso-based* d_0CRT). The fitted predictions from a cross-validated lasso of y on $\mathbb{Z}$ are $d_y = \mathbb{Z}\hat{\beta}_z, d_x = E[x \mid \mathbb{Z}]$ and $T(y, x, d_y, d_x) = \left|\hat{\beta}_x\right| := \dfrac{\left|(y - d_y)^{\mathrm{T}}(x - d_x)\right|}{\|x - d_x\|^2}$.

More generally, the d_0CRT's distillation need not be couched in terms of finding conditional means. For instance, an appealing analogue of Example 1 for binary Y might fit $\hat{\beta}_z$ by a cross-validated L_1-penalized logistic regression of y on $\mathbb{Z}$, and otherwise leave $\mathcal{D}_y$ and $\mathcal{D}_x$ unchanged, and take $T(y, x, d_y, d_x)$ to be the absolute value of the fitted coefficient from a logistic regression of y on $x - d_x$ with offset d_y.

2.4. d_1CRT: Accounting for interactions

Of the three functions applied in Algorithm 2, only T takes both y and x as arguments, and hence the choice of T is how a user can encode the kinds of nonnull relationships between Y and X that are deemed plausible. But, because T only sees $\mathbb{Z}$ through d_y and d_x, any plausible models for Y must be expressed using only x, d_y and d_x. This means that the d_0CRT has almost no capacity to model even first-order interactions between X and Z. For instance, suppose $p = 3$ and $Z_j \sim N(0, 1)$ are independent and identically distributed,

$X \sim Z_1 + N(0,1)$ and $Y \sim Z_2 + XZ_3 + N(0,1)$. Then the best possible distillations of x and y are $d_x = \mathbb{Z}_1$ and $d_y = \mathbb{Z}_2 + \mathbb{Z}_1 \odot \mathbb{Z}_3$, making it impossible for T to encode the true conditional mean of y, namely $\mathbb{Z}_2 + x \odot \mathbb{Z}_3$, from just x, d_x and d_y.

To address this limitation of the d_0CRT, one can simply increase the dimension of d_y and d_x to explicitly include possible columns of $\mathbb{Z}$ with which x might be expected to interact. But of course increasing the dimension of d_y and d_x tends to come at a computational cost, since their low dimensionality is exactly what makes the dCRT fast in the first place. Thus, one needs some sort of prior, domain knowledge or heuristic for choosing based on either the pair $(y, \mathbb{Z})$ or $(x, \mathbb{Z})$ a small subset of columns of $\mathbb{Z}$ that x might plausibly interact with. One option is to split the data into two independent parts and use one part in an unconstrained way to select columns of $\mathbb{Z}$ that are likely to interact with x, and then to leverage these selections in a dCRT run only on the other part. Here we propose an alternative that avoids sample splitting, based on the common statistical practice of only allowing for interactions between variables with strong main effects. This practice of enforcing hierarchy in interactions has a long history in applied and theoretical statistics under many different names (Nelder, 1977; Cox, 1984; Peixoto, 1987; Hamada & Wu, 1992; Chipman, 1996; Bien et al., 2013).

Our proposed method for incorporating interactions, which we call the d_lCRT, is to still have $\mathscr{D}_y$ distil $\mathbb{Z}$ into one column to best capture the relationship between y and $\mathbb{Z}$, but then to additionally return a limited subset of columns of $\mathbb{Z}$ as further columns of d_y whose contributions to that fitted relationship are strongest. Then T can be chosen as a test statistic that allows x to interact with those columns of $\mathbb{Z}$ contained in d_y, while still prioritizing the main effect of x. As a generic example that we found to be powerful to detect hierarchical interactions without losing much power in the absence of interactions, consider the following.

Example 2 (*Lasso-based* d_l*CRT*). The fitted predictions from a cross-validated lasso of y on $\mathbb{Z}$ concatenated with the columns of $\mathbb{Z}$ corresponding to the k largest entries of $\left|\hat{\beta}_z\right|$ are $d_y = \left(\mathbb{Z}\hat{\beta}_z, \mathbb{Z}_{\bullet, \mathrm{top}(k)}\right) \colon = (d_{y,1}, d_{y,-1}), d_x = E[x \mid \mathbb{Z}]$ and $T(y, x, d_y, d_x) = \hat{\beta}^2_{x,1} + \frac{1}{k}\sum_{j=2}^{k+1} \hat{\beta}^2_{x,j}$, where $\hat{\beta}_x \in \mathbb{R}^{k+1}$ are the fitted coefficients from a least-squares fit of $(y - d_{y,1})$ on $(x - d_x)$ and $(x - d_x) \odot d_{y,-1}$.

The normalization by $1/k$ of $\sum_{j=2}^{k+1} \hat{\beta}^2_{x,j}$ encodes our hierarchical prioritization of the main effect $\hat{\beta}_{x,1}$ over the interaction effects. For small k we still expect the computation to be dominated by $\mathscr{D}_y$, but it also represents a statistical trade-off in how widely to search for interactions; we found the performance to be quite stable to k in our simulations, but set as a default $k = \lceil 2\log(p) \rceil$. Note that k could also be chosen after looking at $(y, \mathbb{Z})$, and more generally one can construct many different types of d_lCRT. For instance, one can adapt Example 2 to binary Y in an analogous way as was done for Example 1 by replacing linear regressions with logistic regressions and using $d_{y,1}$ as an offset in T. Or one could have $\mathscr{D}_y$ and/or T use the predictions and default variable importance measures from a random forest. We explore some of these options in simulations in § 4.

2.5. Running the dCRT without resampling

Distillation provides massive computational savings within the CRT by only requiring a single evaluation of by far the most expensive function, $\mathscr{D}_y$. But it still requires $M + 1$ evaluations of T, which can sometimes still contribute nontrivially to the computation time, and requires the user to choose the tuning parameter M which trades off computation and statistical power. It turns out that in certain cases the simplicity of T in the dCRT can be leveraged to remove the resampling of $x^{(m)}$ entirely and compute an exact p-value directly from the single function evaluation $T(y, x, d_y, d_x)$.

For intuition, suppose $X \mid Z \sim N\left(Z^{\mathrm{T}}\gamma, \sigma^2\right)$, and consider the d_0CRT with T as in Example 1,

$$T(y, x, d_y, d_x) = \frac{\left|(y - d_y)^{\mathrm{T}}(x - d_x)\right|}{\|x - d_x\|^2}.$$

Then, since the (d)CRT conditions on y and $\mathbb{Z}$, and hence also d_y and $d_x = \mathbb{Z}\gamma$,

$$(y - d_y)^{\mathrm{T}}(x - d_x) \sim N\left(0, \sigma^2\|y - d_y\|^2\right). \tag{1}$$

The denominator of T makes things a bit more complicated, but the nature of the statistic does not change much if we replace the denominator by its expectation or, equivalently, since multiplying T by a fixed constant has no effect on its resulting p-value, simply replace it by $T'(y, x, d_y, d_x) = \left|(y - d_y)^{\mathrm{T}}(x - d_x)\right|$. We then get immediately that the exact p-value, i.e., the p-value that would result from taking the limit as $M \to \infty$, can be computed as $2\left\{1 - \Phi\left(\frac{T'(y, x, d_y, d_x)}{\sigma\|y - d_y\|}\right)\right\}$ without ever resampling $x^{(m)}$ or recomputing T', where Φ is the standard normal cumulative distribution function.

The same principle can be applied to non-Gaussian X: since the distribution of $(x - d_x) \mid \mathbb{Z}$ is known and the rows are independent, $(x - d_x)$ can be elementwise transformed via scalar monotone functions to be an independent and identically distributed $N(0, 1)$ given $\mathbb{Z}$. For conditionally continuously distributed $(x - d_x)$, this can be done via the probability inverse transform, while for distributions with atoms the atoms need to be carefully randomized, though just once; see the Supplementary Material for details.

As long as $(x - d_x)$ is independent Gaussian or transformed to be, the same principle can also be applied to some more complex T functions. For instance, in Example 2 we can again replace the random denominator, in this case the matrix inverse in the least-squares formula for $\hat{\beta}_x$, with its conditional expectation given $\mathbb{Z}$ and end up with a quadratic form in Gaussian random variables. Efficient algorithms for computing the quantiles of a quadratic form in Gaussian random variables exist (Duchesne & De Micheaux, 2010) and can be applied to again compute the exact dCRT p-value without any resampling; see the Supplementary Material for details.

3. Variable selection and multiple testing via the dCRT

3.1. Outline

Conditional independence testing is often performed in the context of a variable selection problem. Given p covariates $X_1, \ldots, X_p$ and a response Y, the goal is to discover the covariates X_j that are conditionally associated with the response, i.e., $Y \not\perp\!\!\!\perp X_j \mid X_{-j}$. For a given j, we arrive at the problem formulation from the previous two sections by setting $X = X_j$ and $Z = X_{-j}$. This change of notation highlights the fact that the effects of all variables are of interest, rather than that of one special variable. Given a design matrix $\mathbb{X} \in \mathbb{R}^{n \times p}$ and a response vector y, we propose to approach the variable selection problem by applying the dCRT to $(y, x, \mathbb{Z}) = (y, \mathbb{X}_{\bullet,j}, \mathbb{X}_{\bullet,-j})$ for each covariate j, followed by a multiple-testing procedure on the resulting p-values. Two common error rates to control are the familywise error rate and the false discovery rate. The former can be easily achieved based on the Bonferroni correction, which works under arbitrary p-value dependence. The latter is usually done via the Benjamini–Hochberg procedure. Even though the p-values are technically not positively dependent in the sense required for mathematical false discovery rate control (Benjamini & Yekutieli, 2001), the Benjamini–Hochberg procedure is known to be very robust to dependent p-values in all but adversarially constructed settings, as confirmed in our simulations.

Regardless of error rate, the straightforward application of the dCRT to the variable selection problem requires computing $\mathscr{D}_y$ a total of p times, once for each variable. These are entirely parallel computations, so for certain problem dimensionalities and parallel computing resources this is entirely feasible. However, in large-scale variable selection applications such as genome-wide association studies there may be too many covariates for the direct application of dCRT to each. In the following subsections we present two computational shortcuts that make variable selection via the dCRT feasible for large-scale applications.

3.2. Data-dependent screening of variables

A natural acceleration of the dCRT for variable selection is to first use the data to identify a preliminary subset $\mathcal{S} \subseteq \{1, \ldots, p\}$ of promising covariates via a screening function $S: (\mathbb{X}, y) \mapsto \mathcal{S}$. We can then compute (d)CRT p-values $p_j(\mathbb{X}, y)$, via Algorithm 1 or 2, for only $j \in \mathcal{S}$ while setting the p-values for all the other covariates to 1, yielding the screened p-values

$$p_j^{\cdot}(\mathbb{X}, y) = \begin{cases} p_j(\mathbb{X}, y) & \text{if } j \in S(\mathbb{X}, y), \\ 1 & \text{if } j \notin S(\mathbb{X}, y). \end{cases} \tag{2}$$

For instance, $\mathcal{S}$ could be the active set of a cross-validated lasso fit of y on all the covariates.

In general, a screening step like this applied before the (d)CRT breaks the exchangeability between the original and resampled test statistics which Theorem 1 relies on to guarantee p-value validity. Despite this failure of exchangeability, the screening can only inflate a p-value and thus does not affect its validity.

Theorem 2. *Let j be a null variable. For any screening rule S, the screened p-value $p_j^{'}(\mathbb{X}, y)$ obtained from* (2) *is stochastically larger than uniform.*

Proof. By (2), for any $u \in [0, 1]$, $P(p_j^{'}(\mathbb{X}, y) \leqslant u) \leqslant P(p_j(\mathbb{X}, y) \leqslant u) \leqslant u$. □

Thus, with the small computational overhead of a single well-chosen screening function, we can expect to dramatically cut the computation time of using the (d)CRT for variable selection. Indeed, we found in our simulations that simple screenings substantially decreased computation time without affecting the power.

3.3. Recycling computation for L_1-regularized M-estimators

In some cases, we may want to compute p-values for all variables under consideration, even if only a small fraction of these are statistically significant. For instance, these may be needed for downstream analysis tasks like calibration assessment or meta-analysis. In such cases we must look beyond the screening approach. In this section we present a way of recycling computation for L_1-regularized M-estimators including the lasso. This reduces the number of $\mathscr{D}_y$ computations from p to $|\mathscr{A}|$, where $\mathscr{A}$ is the active set of the lasso on $(\mathbb{X}, y)$.

Let $\mathscr{D}_y$ be the cross-validated lasso with strictly convex and differentiable loss function ℓ. Variable selection via the dCRT based on this distillation function requires computing

$$\hat{\beta}(\mathbb{X}_{\bullet, -j}, y; \lambda) := \underset{\beta \in \mathbb{R}^{p-1}}{\arg\min} \sum_{i=1}^{n} \ell\left(Y_i, X_{i, -j}\beta\right) + \lambda \parallel \beta \parallel_1 \tag{3}$$

for each $j = 1, \ldots, p$ along a grid of regularization parameters. There is redundancy among these p lasso problems; they all differ from the the full lasso problem on $(\mathbb{X}, y)$ by just one variable. We may therefore expect that we can save computation by somehow recycling computation across these lasso problems. The next lemma suggests a means to this end.

Lemma 1. *Suppose the columns of $\mathbb{X}$ are in a general position and that the loss ℓ is differentiable and strictly convex. Then, for any $\lambda > 0$,*

$$\hat{\beta}_j(\mathbb{X}, y; \lambda) = 0 \;\Rightarrow\; \hat{\beta}(\mathbb{X}_{\bullet, -j}, y; \lambda) = \hat{\beta}_{-j}(\mathbb{X}, y; \lambda)\,. \tag{4}$$

In other words, Lemma 1 states that removing an inactive variable from the lasso does not change the fitted coefficient vector. This has important computational implications, potentially even outside the scope of this paper: it suggests that we can avoid refitting the lasso (3) for most variables j, instead recycling the lasso fit on the full design matrix. Of course, the parameter λ is usually tuned via cross-validation, which introduces extra complications. However, we claim that if λ is chosen in an appropriate data-dependent way, then an analogous result will still hold.

To make this precise, consider a grid of regularization parameters

$$\lambda(1) > \lambda(2) > \cdots > \lambda(G) > 0 \tag{5}$$

and a corresponding set of cross-validation errors $\mathscr{E}_1, \ldots, \mathscr{E}_G$. Define a rule $\hat{g}$ to select the penalty parameter λ based on cross-validation errors $\mathscr{E}_1, \ldots, \mathscr{E}_G$ to be sequential if these values are traversed in this order, and at some stopping time $\tilde{g}$, the algorithm terminates and chooses $\lambda(\hat{g})$ for some $\hat{g} \leqslant \tilde{g}$. For example, for any integer $\Delta \geqslant 1$, the following rule is sequential: $\hat{g} \equiv \min\{g : \mathscr{E}_g \leqslant \min(\mathscr{E}_{g+1}, \ldots, \mathscr{E}_{g+\Delta})\}$, which is the first time along the regularization path that the cross-validation error is smaller than the following Δ steps, i.e., the first local minimum on the cross-validation path, and the sparsest of all such local minima. In this case the stopping time is $\tilde{g} = \hat{g} + \Delta$. The lasso with any sequential rule $\hat{g}$ has the property (4).

Theorem 3. *Fix a grid of regularization parameters* (5). *Consider applying* L_1*-regularized regression with loss* ℓ *on the whole data* $(\mathbb{X}, y)$, *with* λ *selected by* K*-fold cross-validation and a sequential stopping rule* $\hat{g}$. *Let* $\hat{g}(\mathbb{X}, y)$ *and* $\tilde{g}(\mathbb{X}, y)$ *be the resulting grid point and stopping time, respectively. Letting* $\{1, \ldots, n\} = D_1 \cup \cdots \cup D_K$ *denote the split of the data into non-overlapping folds, define the active set*

$$\mathscr{A} = \Big(j \in \{1, \ldots, p\} : \hat{\beta}_j[\mathbb{X}, y; \lambda\{\hat{g}(\mathbb{X}, y)\}] \neq 0 \text{ or } \hat{\beta}_j\{\mathbb{X}_{D_k, \bullet}, y_{-D_k}; \lambda(g)\} \neq 0 \text{ for some } k, g \leqslant \tilde{g}(\mathbb{X}, y)\Big). \tag{6}$$

If the loss ℓ *is differentiable and strictly convex, and the columns of* $\mathbb{X}$ *and* $\mathbb{X}_{-D_k, \bullet}$ *are in general position for each* k, *then excluding nonactive variables* j *does not alter the fitted coefficients: for each* $j \notin \mathscr{A}$,

$$\hat{g}(\mathbb{X}_{\bullet, -j}, y) = \hat{g}(\mathbb{X}, y) \text{ and } \hat{\beta}[\mathbb{X}_{\bullet, -j}, y; \lambda\{\hat{g}(\mathbb{X}_{\bullet, -j}, y)\}] = \hat{\beta}_{-j}[\mathbb{X}, y; \lambda\{\hat{g}(\mathbb{X}, y)\}] \tag{7}$$

Theorem 3 states that for each variable j not in the active set we need not rerun the lasso holding out variable j; we can instead fit the full lasso once and then read off the coefficient vector. This computational shortcut, summarized in the Supplementary Material, reduces the number of lasso applications required by the dCRT from p to $|\mathscr{A}|$. Depending on the sparsity of the problem, this reduction can save several orders of magnitude of computation. It is known that, at most, the lasso solution has $\min(p, n)$ nonzero entries (Tibshirani, 2013), though often it is much sparser.

4. Statistical performance of the dCRT

4.1. Implications of distillation for power

Our motivation for proposing the dCRT is computational; using distilled test statistics accelerates the CRT by orders of magnitude compared to the originally proposed lasso coefficient test statistic. In this section we discuss the statistical implications of this computational acceleration. While distillation is a flexible framework that can encompass a variety of test statistics, for concreteness in this section we narrow our focus to the d_0CRT. Our goal is to carefully compare the d_0CRT to its undistilled counterpart, the oCRT based on the absolute lasso coefficient. Our main conclusion is that, perhaps surprisingly, distillation

does not have much effect on the power of the CRT. We present the main reasoning behind this conclusion here and defer the details to the Supplementary Material.

To emphasize the exclusion of x from the lasso regression, let $\hat{\beta}_z^{\text{loco}}$ be the fitted coefficients in the lasso regression of y on $\mathbb{Z}$ and let $\hat{\beta}_x^{\text{loco}} = (x - d_x)^{\mathrm{T}}\left(y - \mathbb{Z}\hat{\beta}_z^{\text{loco}}\right)/\|x - d_x\|^2$ as in the definition of the $\mathrm{d_0CRT}$. By contrast, let $\left(\hat{\beta}_x^{\text{lasso}}, \hat{\beta}_z^{\text{lasso}}\right)$ denote the fitted coefficients in the lasso regression of y on x and $\mathbb{Z}$, so the oCRT is based on the test statistic $\left|\hat{\beta}_x^{\text{lasso}}\right|$. Let us also assume in this section, as we did in § 2.5, that $X \mid Z \sim N\left(Z^{\mathrm{T}}\gamma, s^2\right)$. Finally, for intuition, suppose that $Y \mid X, Z$ follows a Gaussian linear model with coefficients β_x and β_z.

The obvious difference between oCRT and dCRT is that the latter is based on a lasso regression excluding the variable of interest while the former is based on a lasso regression on all variables. Thus, $\hat{\beta}_z^{\text{loco}} \neq \hat{\beta}_z^{\text{lasso}}$, and so of course $\hat{\beta}_x^{\text{boco}} \neq \hat{\beta}_x^{\text{lasso}}$. However, there are two additional differences that must be accounted for in order to understand the relationship between the two methods. First of all, $\hat{\beta}_x^{\text{lasso}}$ has a nonzero probability of being equal to zero, while $\hat{\beta}_x^{\text{loco}}$ is almost surely nonzero. Secondly, $\hat{\beta}_x^{\text{lasso}}$ does not necessarily have a null distribution centred on zero, whereas $\hat{\beta}_x^{\text{loco}}$ does according to (1).

In the Supplementary Material we examine the impacts of these two properties of the oCRT. We find that the sparsity that $\hat{\beta}_x^{\text{lasso}}$ inherits from the lasso can only hurt the power of the oCRT, and propose a simple alternative based on removing the soft threshold operator. Furthermore, we show that, depending on the distribution of $X \mid Z$ and on the locations and signs of the nonzero elements of (β_x, β_z), the null distribution of $\hat{\beta}_x^{\text{lasso}}$ can either be centred at the origin, or left or right of the origin. By using the absolute value of the potentially off-centre test statistic $\hat{\beta}_x^{\text{lasso}}$, the oCRT gains or loses power to the extent that the null distribution is shifted to the right or left, respectively. This motivates us to propose a centred and non-soft-thresholded version of the oCRT test statistic.

Using numerical simulations, we found essentially no difference between the performance of the dCRT and the centred, non-soft-thresholded version of the oCRT. In other words, after accounting for the aforementioned two differences, the distillation step has little or no impact on the power of the CRT. This conclusion may seem surprising, since on first glance leaving x out appears to cause some of the signal, namely the contribution of x to y that can instead be explained by $\mathbb{Z}$, to be regressed out. One may expect this effect to decrease the power of the dCRT. However, this is not the case because it is precisely the component of x that cannot be explained by $\mathbb{Z}$ that carries signal. Therefore, dropping x and regressing $\mathbb{Z}$ out of y first does not have much effect on the power of the dCRT. This intuition would be precise if $\hat{\beta}_z^{\text{loco}}$ were obtained from unpenalized linear regression of y on $\mathbb{Z}$. Indeed, it is a well-known property of linear regression that the coefficient of x can be obtained by first regressing $\mathbb{Z}$ out of x and y, and then regressing the residual of y onto that of x.

4.2. Numerical comparisons of power, speed, robustness and stability

Going beyond the numerical simulations in the previous section, we designed an extensive simulation suite to systematically assess the power and other operating characteristics of dCRT, and compare it with several alternative methods. Preferring to compare the dCRT to existing methods, we chose to benchmark it against the originally proposed oCRT instead of the modified version considered above. We must keep in mind, however, that this choice also complicates the comparison for the aforementioned reasons. Furthermore, we suspect that the centring and soft-thresholding issues may impact the performance of knockoffs as well. Unlike for the CRT, however, it is harder to pull these aspects apart for knockoffs. The soft thresholding affects both the one-bit p-values and the ordering of the variables, so removing it may not result in a uniform improvement like it did for oCRT. Regarding centring, it is not obvious how to recentre the knockoffs null distribution because knockoffs does not really use a null distribution. We leave the study of these phenomena for knockoffs to future work, and in the meantime compare dCRT to published implementations of the latter.

In the interest of space we defer the details of our simulations to the Supplementary Material and present here a detailed summary of the takeaways. The main focus of our simulations is examining the performance of the dCRT through the d_0CRT and d_lCRT given by Examples 1 and 2, respectively. Except where explicitly stated otherwise, we apply them in a resampling-free manner as per § 2.5 and, when simulating a variable selection task, with screening using the cross-validated lasso for selection as per § 3.2. For variable selection simulations, we take each of the p-value methods, i.e., oCRT, dCRT and HRT, and apply the Benjamini–Hochberg procedure when targeting false discovery rate control and the Bonferroni correction when targeting familywise error rate control. Source code for running the dCRT and reproducing our results, along with example scripts for illustration, can be found at https://github.com/moleibobliu/Distillation-CRT.

We compared the dCRT to the oCRT in a broader set of simulations than those referenced in § 4.1, including linear and logistic regression models and d_lCRT as well as d_0CRT. We chose the smaller problem size of $n = p = 300$ to accommodate the computational burden of the oCRT. We found that distillation dramatically reduces CRT computation; both the d_0CRT and d_lCRT conferred computational savings of approximately 500 times over the oCRT. The relative powers of dCRT and oCRT were consistent with what we found in § 4.1, with dCRT sometimes more powerful and sometimes less powerful than the oCRT. The oCRT was more powerful when signals were equally spaced, while the dCRT was more powerful when signals were adjacent to each other. We suspect these differences to be caused mainly by the discussed soft thresholding and centring issues.

The dCRT is more powerful than the HRT. In both the aforementioned $n = p = 300$ simulations and a larger simulation with $n = p = 800$, the dCRT computation times were mostly within an order of magnitude of the HRT. But, across settings that included a range of n up to 1400, a range of p up to 3200, a range of signal magnitudes, a range of sparsities, a range of covariance structures for X and a range of models for $Y \mid X$, both dCRT methods had consistently up to about 50 percentage points higher power than the HRT.

When controlling false discovery rate, the relative performance of dCRT and knockoffs varies across simulation settings, similar to the relative performance of the dCRT and oCRT. The dCRT methods tend to have higher power than knockoffs when signal variables are adjacent, and lower power than knockoffs when the signal variables are equally spaced. The power comparison between dCRT and knockoffs is a subtle one, and we leave its further investigation for future work. In very sparse settings dCRT still has power, while knockoffs does not due to its reliance on the Selective SeqStep+ procedure (Barber & Candès, 2015). In such regimes, the familywise error rate may be more appropriate, and the dCRT can be used to control this error rate as well. The dCRT is more computationally expensive than knockoffs, but usually within an order of magnitude. Finally, the dCRT has substantially less algorithmic variability than knockoffs, as measured by the expected Jaccard similarity between two rejection sets obtained by rerunning the methods with different seeds.

The d_1CRT is stable to the choice of k and has slightly less power than the d_0CRT in additive models, but can have substantially higher power in the presence of interactions. In a simulation with an additive model, the power of the d_1CRT was identical as k ranged from 2 to 22, noting that the default value of $k = 2\log(p)$ would have been 13, while in a model with five true interactions, the power only varied from about 50% to about 40% over the same range of k. Throughout all our simulations in additive models we found the d_0CRT to be slightly, but consistently, more powerful than the d_1CRT, but, in the presence of interactions obeying the hierarchy principle discussed in § 2.4, we found that the d_1CRT could be up to about 25 percentage points more powerful than the d_0CRT.

The dCRT can leverage nonparametric machine learning algorithms for substantial power gains in highly nonlinear models. In a simulation in which X's relationship with Y was highly nonlinear and interacted with five Z_j, our default lasso-based d_1CRT had about 20 percentage points higher power than d_0CRT, but a different, random-forest-based d_1CRT had far higher power than the lasso-based d_1CRT by as much as about 50 percentage points.

The dCRT is quite robust to misspecification of X's distribution. When the distribution of $X \mid Z$ is Poisson even with a very small mean parameter, making it highly discrete and heavily skewed, but approximated by a Gaussian with matching mean and variance, both the d_0CRT and d_1CRT maintain Type-I error control and high power. Furthermore, when the covariates are jointly Gaussian and the $X \mid Z$ distributions are estimated in-sample using any of three standard methods detailed in the Supplementary Material, the Type-I error of both dCRT methods always remains close to the nominal level.

The resampling-free versions of the dCRT are faster and just as powerful as the non-resampling-free dCRT except when $X \mid Z$ is highly discrete. The resampling-free modification sped up the d_0CRT by 2.5 times in an $n = p = 800$ simulation and sped up the d_1CRT by 11 times in an $n = p = 800$ simulation, even after applying screening. When $X \mid Z$ is Gaussian, changing the form of the test statistics of the d_0CRT and d_1CRT as proposed in paragraphs 2 and 4, respectively, of § 2.5, had a negligible effect on their power. When $X \mid Z$ is non-Gaussian and must be transformed to Gaussian, we found essentially no power loss for the resampling-free d_0CRT and d_1CRT relative to their non-resampling-free

counterparts when $X \mid Z$ was Gamma-distributed with skew > 1 and excess kurtosis = 2, while there was up to about 40 percentage points power loss when $X \mid Z$ was binary, and hence required substantial exogenous randomization to be transformed to Gaussian, though the resampling-free dCRTs were still up to about 10 percent more powerful than the HRT.

Screening makes the dCRT faster without affecting its power. In a simulation with $n = p = 800$, screening reduced the computation time by a factor of about five for both d_0CRT and d_1CRT without perceptibly hurting power.

5. Identifying biomarkers for breast cancer

As a final demonstration of the effectiveness of the dCRT, we apply it to the dataset from Curtis et al. (2012), consisting of $n = 1396$ staged oestrogen-receptor-positive cases of breast cancer, each with expression level (mRNA) and copy number aberration measured for $p = 164$ genes, which was studied in Pereira et al. (2016). Our goal is to find genes on which the cancer stage depends, conditional on the remaining genes and all copy number aberrations, while controlling either the false discovery rate or familywise error rate at level 0.1. Discovering such biomarkers for cancer can reveal new pathways and mechanisms for cancer progression; see Shen et al. (2019) for a recent application of model-X knockoffs to the same end.

After log-transforming the gene expressions, we adjusted them using the copy number aberration data with a linear model as in Solvang et al. (2011), Lahti et al. (2012) and Leday et al. (2013), and modelled the processed gene expressions jointly as multivariate Gaussian, similar to Shen et al. (2019). We applied the d_0CRT, the d_1CRT, the oCRT, the HRT and model-X knockoffs, and compared the results. Each method was run 300 times. Table 1 contains average runtimes in `R` (R Development Core Team, 2022) for all methods, showing that the dCRTs are quite fast compared to the oCRT. In particular, the oCRT takes over 7 hours to run while the dCRTs take under a minute.

Figure 1 presents the distribution of the numbers of discoveries among the 300 repetitions for all the methods. Methods including dCRT, oCRT and HRT have stable outputs about the number of detected genes. In terms of false discovery rate control, d_0CRT and d_1CRT detect exactly 5 genes in more than 80% of repetitions and $\geqslant 5$ genes at all times. Methods oCRT and HRT detect exactly 3 genes in more than 70% of repetitions and always have fewer discoveries than dCRT, while the knockoffs have 0 discoveries in about 45% of repetitions, but $\geqslant 10$ discoveries in the remaining times, which implies that knockoffs fail to produce stable output. Knockoffs' instability and lack of power is due to the sparsity of discoverable genes. In terms of familywise error rate control, d_0CRT and HRT have three discoveries in most runs, d_1CRT has four discoveries and oCRT has two.

When used to control the false discovery rate, it turns out that all five genes discovered by the dCRT, *FBXW7*, *MAP3K13*, *HRAS*, *GPS2* and *RUNX1*, have been linked in independent research to cancer, suggesting the dCRT makes promising discoveries. In particular, *FBXW7* encodes a member of the F-box protein family, and its mutations are detected in ovarian and breast cancer cell lines (Kirzinger et al., 2019; Liu et al., 2019); *MAP3K13* belongs to the

serine/threonine protein kinase family acting as a regulator for cancer (Han et al., 2016); *HRAS* belongs to the RAS oncogene family which is related to the transforming of genes of mammalian sarcoma retro-viruses, and defects in this gene have been implicated in a variety of cancers (Geyer et al., 2018); overexpression of *GPS2* in mammalian cells may suppress signals mediated by RAS/MAPK and interfere with JNK activity, all of which are cancer related (Jarmalavicius et al., 2010; Huang et al., 2016); *RUNX1* has been found to activate certain signalling pathways that promote tumor metastasis (Li et al., 2019).

6. Discussion

The HRT provided the first indication that a variant of the CRT could be computationally tractable, albeit at the cost of statistical performance. In this paper we demonstrate that leaving out variables instead of samples creates a procedure that is not quite as fast, though still a tiny fraction of the oCRT's computational cost, but much more powerful. This brings the dCRT into the realm of fast and powerful model-X methods, where knockoffs is currently the methodology of choice. Knockoffs and dCRT have complementary strengths, which we discuss briefly below.

Model-X knockoffs address the variable selection problem, targeting false discovery rate control. They are computationally very efficient, requiring just one high-dimensional model fit. Furthermore, our simulations confirm that knockoffs are quite powerful in several settings. These advantages have led to the successful application of knockoffs to genome-wide association studies (Sesia et al., 2019, 2020). By comparison, the dCRT still requires several high-dimensional model fits and is therefore more computationally costly. On the other hand, dCRT computation benefits from being embarrassingly parallelizable, so modern parallel computing resources can greatly reduce its runtime. As far as power goes, the relative performance of the two methods varies with simulation setting, see § 4 and the Supplementary Material; neither procedure uniformly dominates the other when controlling the false discovery rate.

Aside from these considerations, the dCRT provides a few important advantages over knockoffs. The first is that, unlike knockoffs, the dCRT provides p-values arbitrarily fine-grained and essentially exact for each conditional independence hypothesis. In addition to providing an inter-pretable measure of significance, this decoupling of statistical significance quantification from downstream analyses such as multiple testing brings great versatility. Indeed, dCRT p-values can be used for single-hypothesis testing, multiple-hypothesis testing with a variety of error rates, and any number of other tasks that take p-values as input. While the knockoffs framework has gradually been extended to handle analysis tasks beyond false discovery rate control, e.g., k-familywise error rate control by Janson & Su (2016) and simultaneous false discovery probability control by Katsevich & Ramdas (2020), such extensions require custom solutions and some are currently out of reach, such as single testing or familywise error rate control. Another advantage of the dCRT is that it has little or no variability across runs. On the other hand, knockoffs is a randomized procedure and this randomization can lead to variability in the performance of the procedure on a given dataset; see the Supplementary Material and Fig. 4 of Sesia et al. (2019).

The dCRT is therefore a useful addition to the model-X methodology toolbox. Much work still remains to refine this new tool for better power and even faster computation. Indeed, many degrees of freedom in the construction of the dCRT test statistic remain to be explored. For example, should the statistic be based on the fitted coefficient of a variable or on the loss function? What is the best way to test groups of variables? The recent theoretical exploration of the CRT (Katsevich & Ramdas, 2021) may help guide the search for powerful test statistics. Another open question is whether there are efficient resampling-free dCRT variants for highly discrete covariates. Finally, the dependence structure of (d)CRT p-values is an important subject for further exploration. We may not always be able to plug-and-play (d)CRT p-values in multiple-testing procedures, since their dependency structure is currently unknown. In a related development, Bates et al. (2020) recently proposed a clever method of generating independent HRT p-values for groups of linearly structured covariates.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgement

We thank Siyuan Ma, Wenshuo Wang, Dae Woong Ham, Lu Zhang, Shuangning Li and Emmanuel Candès, as well as two referees and an associate editor for helpful discussions and constructive feedback that helped improve our paper. This work used the Extreme Science and Engineering Discovery Environment (Towns et al., 2014), supported by the National Science Foundation (ACI-1548562). Specifically, it used the Bridges system (Nystrom et al., 2015), which is supported by the National Science Foundation (ACI-1445606), Ramdas is also affiliated with the Machine Learning Department at Carnegie Mellon University.

References

Barber RF & Candès EJ (2015). Controlling the false discovery rate via knockoffs. Ann. Statist 43, 2055–85.

Bates S, Sesia M, Sabatti C & Candès E (2020). Causal inference in genetic trio studies. Proc. Nat. Acad. Sci 117, 24117–26. [PubMed: 32948695]

Bellot A & van der Schaar M (2019). Conditional independence testing using generative adversarial networks. Proc. Adv. Neural Inf. Proc. Syst 32, 2199–208.

Benjamini Y & Hochberg Y (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. J. R. Statist. Soc. B 57, 289–300.

Benjamini Y & Yekutieli D (2001). The control of the false discovery rate in multiple testing under dependency. Ann. Statist 29, 1165–88.

Berrett TB, Wang Y, Barber RF & Samworth RJ (2020). The conditional permutation test for independence while controlling for confounders. J. R. Statist. Soc. B 82, 175–97.

Bien J, Taylor J & Tibshirani R (2013). A lasso for hierarchical interactions. Ann. Statist 41, 1111–41.

Candès E, Fan Y, Janson L & Lv J (2018). Panning for gold: Model-X knockoffs for high dimensional controlled variable selection. J. R. Statist. Soc. B 80, 551–77.

Chernozhukov V, Chetverikov D, Demirer M, Duflo E, Hansen C, Newey W & Robins J (2018). Double/debiased machine learning for treatment and structural parameters. Economet. J 21, C1–C68.

Chipman H (1996). Bayesian variable selection with related predictors. Can. J. Statist 24, 17–36.

Cox DR (1984). Interaction. Int. Statist. Rev 52, 1–31.

Curtis C, Shah SP, Chin SF, Turashvili G, Rueda OM, Dunning MJ, Speed D, Lynch AG, Samarajiwa S, Yuan Y and Gräf S (2012). The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. Nature 486, 346–52. [PubMed: 22522925]

Duchesne P & De Micheaux PL (2010). Computing the distribution of quadratic forms: Further comparisons between the Liu–Tang–Zhang approximation and exact methods. Comp. Statist. Data Anal 54, 858–62.

Geyer FC, Li A, Papanastasiou AD, Smith A, Selenica P, Burke KA, Edelweiss M, Wen HC, Piscuoglio S & Schultheis AM (2018). Recurrent hotspot mutations in HRAS-Q61 and PI3K-AKT pathway genes as drivers of breast adenomyoepitheliomas. Nature Commun 9, 1–16. [PubMed: 29317637]

Hamada M & Wu CJ (1992). Analysis of designed experiments with complex aliasing. J. Qual. Technol 24, 130–7.

Han H, Chen Y, Cheng L, Prochownik EV & Li Y (2016). microRNA-206 impairs c-Myc-driven cancer in a synthetic lethal manner by directly inhibiting MAP3K13. Oncotarget 7, 16409–19. [PubMed: 26918941]

Huang X, Xiao F, Wang S, Yin R, Lu C, Li Q, Liu N, Wang L & Li P (2016). G protein pathway suppressor 2 (GPS2) acts as a tumor suppressor in liposarcoma. Tumor Biol 37, 13333–43.

Janson L & Su W (2016). Familywise error rate control via knockoffs. Electron. J. Statist 10, 960–75

Jarmalavicius S, Trefzer U & Walden P (2010). Differential arginine methylation of the G-protein pathway suppressor GPS-2 recognized by tumor-specific T-cells in melanoma. FASEB J 24, 937–46. [PubMed: 19917673]

Katsevich E & Ramdas A (2020). Simultaneous high-probability bounds on the false discovery proportion in structured, regression, and online settings. Ann. Statist 48, 3465–87.

Katsevich E & Ramdas A (2021). A theoretical treatment of conditional independence testing under model-X. arXiv:2005.05506v3.

Kirzinger MW, Vizeacoumar FS, Haave B, Gonzalez Lopez C, Bonham K, Kusalik A & Vizeacoumar FJ (2019). Humanized yeast genetic interaction mapping predicts synthetic lethal interactions of FBXW7 in breast cancer. BMC Med. Genom 12, 112.

Lahti L, Schäfer M, Klein HU, Bicciato S & Dugas M (2012). Cancer gene prioritization by integrative analysis of mRNA expression and DNA copy number data: A comparative review. Brief. Bioinform 14, 27–35. [PubMed: 22441573]

Leday GG, van der Vaart AW, van Wieringen WN & van de Wiel MA (2013). Modeling association between DNA copy number and gene expression with constrained piecewise linear regression splines. Ann. Appl. Statist 7, 823–45.

Li Q, Lai Q, He C, Fang Y, Yan Q, Zhang Y, Wang X, Gu C, Wang Y, Ye et al. (2019). RUNX1 promotes tumour metastasis by activating the Wnt/β-catenin signalling pathway and EMT in colorectal cancer. J. Exp. Clin. Cancer Res 38, 334. [PubMed: 31370857]

Liu F, Zou Y, Wang F, Yang B, Zhang Z, Luo Y, Liang M, Zhou J & Huang O (2019). FBXW7 mutations promote cell proliferation, migration, and invasion in cervical cancer. Genet. Test. Molec. Biomarkers 23, 409–17. [PubMed: 31161818]

Nelder J (1977). A reformulation of linear models. J. R. Statist. Soc. A 140, 48–63.

Nystrom NA, Levine MJ, Roskies RZ & Scott JR (2015). Bridges: A uniquely flexible HPC resource for new communities and data analytics. In Proc. 2015 XSEDE Conf. Sci. Adv. Enabled by Enhanced Cyberinfrastructure New York, NY, USA: ACM.

Peixoto JL (1987). Hierarchical variable selection in polynomial regression models. Am. Statistician 41, 311–13.

Pereira B, Chin S-F, Rueda OM, Vollan H-KM, Provenzano E, Bardwell HA, Pugh M, Jones L, Russell R, Sammut S-J et al. (2016). The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes. Nature Commun 7, 11479. [PubMed: 27161491]

R Development Core Team (2022). R: A Language and Environment for Statistical Computing, Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0, http://www.R-project.org.

Sesia M, Katsevich E, Bates S, Candès E & Sabatti C (2020). Multi-resolution localization of causal variants across the genome. Nature Commun 11, 1093. [PubMed: 32107378]

Sesia M, Sabatti C & Candès EJ (2019). Gene hunting with hidden Markov model knockoffs. Biometrika 106, 1–18. [PubMed: 30799875]

Shah RD & Peters J (2018). The hardness of conditional independence testing and the generalised covariance measure. Ann. Statist 48, 1514–38.

Shen A, Fu H, He K & Jiang H (2019). False discovery rate control in cancer biomarker selection using knockoffs. Cancers 11, 744. [PubMed: 31146393]

Solvang HK, Linguaerde OC, Frigessi A, Børresen Dale AL & KristenSen VN (2011). Linear and nonlinear dependencies between copy number aberrations and mRNA expression reveal distinct molecular pathways in breast cancer. BMC Bioinform 12, 197.

Tansey W, Veitch V, Zhang H, Rabadan R & Blei DM (2021). The holdout randomization test: Principled and easy black box feature selection. arXiv:1811.00645v4.

Tibshirani R (1996). Regression shrinkage and selection via the lasso. J. R. Statist. Soc. B 58, 267–88.

Tibshirani RJ (2013). The lasso problem and uniqueness. Electron. J. Statist 7, 1456–90

Towns J, Cockerill T, Dahan M, Foster I, Gaither K, Grimshaw A, Hazlewood V, Lathrop S, Lifka D, Peterson GD et al. (2014). XSEDE: Accelerating scientific discovery. Comp. Sci. Eng 16, 62–74.

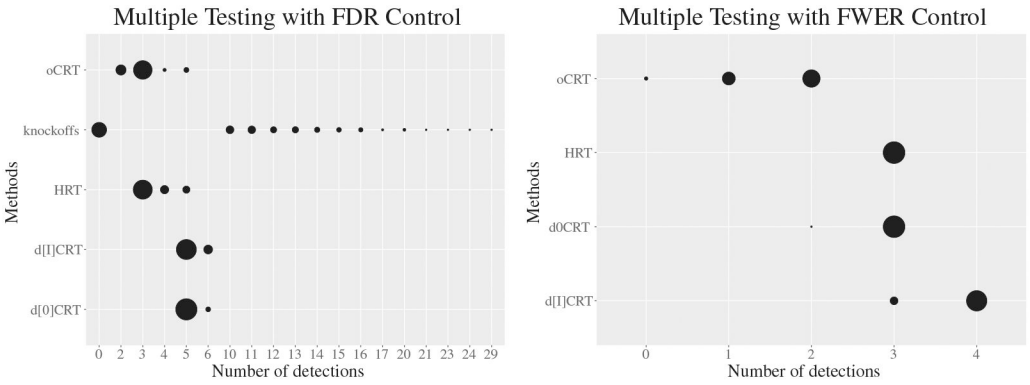

Fig. 1.

Summary of the numbers of discoveries over 300 repetitions, with false discovery rate and familywise error rate control, in the breast cancer application. The area of each black point is proportional to the frequency that the corresponding method makes this number of discoveries in the 300 repetitions. The dCRT approaches are more powerful than oCRT and HRT. The knockoffs have no discoveries in around 45% of the experiments.

Table 1.

Average computation times of 300 repetitions in R in the breast cancer application. Our use of the resampling-free version of the dCRT makes it faster than the HRT in this case

d_0CRT	d_1CRT	**oCRT**	**Knockoffs**	**HRT**
0.8	0.8	443.3	0.3	3.1