Invariant Causal Set Covering Machines

Thibaud Godon [1] **Baptiste Bauvin** [1] **Pascal Germain** [1 2] **Jacques Corbeil** [1] **Alexandre Drouin** [1 2 3]

Abstract

Rule-based models, such as decision trees, appeal to practitioners due to their interpretable nature. However, the learning algorithms that produce such models are often vulnerable to spurious associations and thus, they are not guaranteed to extract causally-relevant insights. In this work, we build on ideas from the invariant causal prediction literature to propose *Invariant Causal Set Covering Machines*, an extension of the classical Set Covering Machine algorithm for conjunctions/disjunctions of binary-valued rules that provably avoids spurious associations. We demonstrate both theoretically and empirically that our method can identify the causal parents of a variable of interest in polynomial time.

1. Introduction

In some fields of application of machine learning, the use of learned models goes well beyond prediction. Domain experts often rely on a deeper inspection of such models to extract mechanistic insights into complex systems. For instance, in healthcare, one can train a model to predict predisposition to a disease based on genomics data (Szymczak et al., 2009). Significant insights can then be obtained by understanding which genomic traits are used for prediction (biomarkers). These constitute potential causes of the disease, which might be relevant targets in the elaboration of new therapies or drugs.

One kind of machine learning model that has been shown to allow for such scrutiny is rule-based models, such as decision trees (Breiman et al., 1984). Such models make inferences by applying a series of binary-valued rules to their input (e.g., the presence or absence of a mutation). In addition to their high level of interpretability, such models have been shown to scale particularly well to large feature sets and resist overfitting in the small data regime that is common in applications of machine learning to healthcare (Drouin et al., 2019). However, one must not be fooled by the ease of interpretation of such models, since the variables used for prediction may very well be spuriously associated with the outcome of interest.

In this work, we make a step towards alleviating this issue by proposing *Invariant Causal Set Covering Machines*, an extension of the Set Covering Machine algorithm (Marchand & Shawe-Taylor, 2002) for conjunctive and disjunctive classifiers, that avoids, as much as possible, relying on spurious associations for prediction. Our work builds on previous advances in invariant causal prediction (Peters et al., 2016; Heinze-Deml et al., 2018; Bühlmann, 2020), where data is assumed to be collected in multiple environments (e.g., populations from different geographic locations, measurement devices with different calibrations, etc.).

Contributions: **(1)** We propose Invariant Causal Set Covering Machines (ICSCM), an extension of the Set Covering Machine algorithm (Marchand & Shawe-Taylor, 2002) that relies on invariant causal prediction to avoid spurious associations (Section 3). **(2)** We support this new algorithm with theoretical results expressing conditions under which the causal parents of an outcome of interest can be recovered in polynomial time (Theorem 3.1). **(3)** We conduct a brief empirical study with simulated data to verify the correctness of the theory and the efficiency of the algorithm (Section 4). Note that this is an initial work and that future work will significantly extend the empirical analysis, e.g., with real data.

2. Background and Related Work

2.1. Problem setting

We consider a standard supervised binary classification setting, where the observations are feature-label pairs $(\mathbf{x}, y)$, with $\mathbf{x} \in \mathcal{X}$ and $y \in \{0, 1\}$. Further, as in Heinze-Deml et al. (2018), we consider the case where the observations have been collected in multiple environments $e \in \mathcal{E}$, which correspond to various experimental conditions (e.g., populations from different geographic locations, measurement devices with different calibrations, etc.). Hence, we assume access to observations $(\mathbf{x}, y, e) \sim P(\mathbf{X}, Y, E)$, where the data-generating process P factorizes according to G depicted at Fig. 1. The random variable $\mathbf{X}$ is segmented

*Equal contribution [1]Université Laval [2]Mila-Québec [3]ServiceNow Research. Correspondence to: Thibaud Godon <thibaud.godon.1@ulaval.ca>, Alexandre Drouin <alexandre.drouin@servicenow.com>.

Published at the ICML 2023 Workshop on Spurious Correlations, Invariance, and Stability. Honolulu, Hawaii, USA.

as $\mathbf{X} = [\mathbf{X}_A, \mathbf{X}_B, \mathbf{X}_C]$, respectively denoting the causal parents of Y, variables that are not directly related to Y, and the causal children of Y. Formally, we make the following assumptions, which are common in the causal discovery literature (see Glymour et al., 2019):

Assumption 2.1. (Causal Markov assumption) We assume that *d-separation*[1] in G implies conditional independence in P, i.e., for arbitrary random variables U, V, W: $U \perp\!\!\!\perp_G V \mid W \Rightarrow U \perp\!\!\!\perp_P V \mid W$, where $\perp\!\!\!\perp_G$ denotes d-separation and $\perp\!\!\!\perp_P$ denotes independence in distribution.

Assumption 2.2. (Faithfulness assumption) For arbitrary random variables U, V, W: $U \perp\!\!\!\perp_P V \mid W \Rightarrow U \perp\!\!\!\perp_G V \mid W$.

Further, we assume that the environments are defined as follows, matching the setting of Heinze-Deml et al. (2018).

Assumption 2.3. (Environments) The environment E is a causal parent of $\mathbf{X}$, but it is not a causal parent of Y (see Fig. 1). In other words, the structural equation $Y := f(\mathbf{X}_A, \epsilon)$, with noise $\epsilon \perp\!\!\!\perp_P \mathbf{X}_A$, is invariant across environments.

Intuitively, this means that the distribution of features $\mathbf{X}$ can change across environments, but the mechanism that produces Y from its causal parents $\mathbf{X}_A$ must remain stable.

Goal: We aim to learn a classifier $h : \mathcal{X} \to \{0, 1\}$, such that $h(\mathbf{x}) = h(\mathbf{x}_A) = y$ with high probability, i.e., that closely approximates Y while relying solely on its *causal parents* $\mathbf{X}_A$ and no other *spurious association*.

Moreover, we are interested in learning classifiers that are conjunctive in nature, so we make the following additional assumption on the functional form of $Y := f(\mathbf{X}_A, \epsilon)$:

Assumption 2.4. (Functional form) The function f is s.t.

$$f(\mathbf{X}_A, \epsilon) \stackrel{\text{def}}{=} g(r_1(\mathbf{X}_A, \epsilon_1) \wedge \ldots \wedge r_d(\mathbf{X}_A, \epsilon_d), \epsilon_g), \quad (1)$$

where $\langle r_1, \ldots, r_d \rangle$ are arbitrary binary-valued rules (e.g., threshold functions on the value of features), $\epsilon := \langle \epsilon_1, \ldots, \epsilon_d, \epsilon_g \rangle$ are noise terms sampled from arbitrary independent distributions, and g is a function that tampers with the outcome based on ϵ_g.

2.2. Set Covering Machines

We now introduce the Set Covering Machine (SCM) algorithm[2] (Marchand & Shawe-Taylor, 2002), which can be used to learn classifiers that are conjunctive in nature. Later on, at Section 3, we will explain how this algorithm can be extended to achieve the goal defined at Section 2.1.

Let $(\mathbf{x}, y)$ be a pair of features and binary label, as described at Section 2.1. The SCM algorithm is a greedy learning algorithm that attempts to find the shortest conjunction $h(\mathbf{x}) = r_1(\mathbf{x}) \wedge \ldots \wedge r_d(\mathbf{x})$ or disjunction $h(\mathbf{x}) = r_1(\mathbf{x}) \vee \ldots \vee r_d(\mathbf{x})$, where the r_i are binary-valued rules, that minimizes the expected prediction error, $\mathbb{E}_{(\mathbf{x},y)\sim P(\mathbf{X},Y)}\, I[h(\mathbf{x}) \neq y]$, where $I[\text{True}] = 1$ and 0 otherwise.

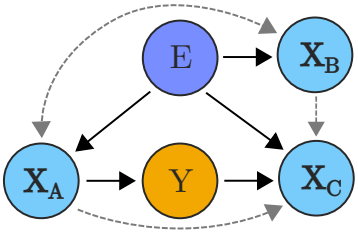

Figure 1. Graphical assumptions: the edge between $\mathbf{X}_A$ and $\mathbf{X}_B$ can be oriented in either way, but the resulting G must be a Directed Acyclic Graph (DAG). Dashed edges are optional.

For conciseness, the rest of the presentation will focus on learning conjunctions. This is not restrictive since any algorithm for learning conjunctions can also be applied to learning disjunctions by the simple application of De Morgan's law, i.e., $\neg\big(r_1(\mathbf{x}) \wedge r_2(\mathbf{x})\big) = \neg r_1(\mathbf{x}) \vee \neg r_2(\mathbf{x})$. Hence, all the forthcoming findings and observations equally apply to the case of learning disjunctions.

Algorithm 1 shows the pseudocode for learning a conjunction with the SCM algorithm. The algorithm starts with a data sample $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^m \sim P(\mathbf{X}, Y)^m$ and a set $\mathcal{R}$ of candidate binary-valued rules. It builds a conjunction by iteratively adding rules $r_i \in \mathcal{R}$. At each iteration, the best rule is selected based on a utility score U_i, which is a tradeoff between the number of negative examples correctly classified (i.e., covered by the rule[3]) and the number of positive examples misclassified by the rule. Then, the examples for which the model's outcome is settled (i.e., $h(\mathbf{x}) = 0$) are discarded, and the next iteration is performed considering the examples that remain to be classified.[4] The training stops when no negative examples remain to cover or when the maximum conjunction length n, which is a hyperparameter, is reached. Overall, the running time complexity of this algorithm is $O(m \cdot |\mathcal{R}| \cdot n)$.

The SCM algorithm is thus a simple and efficient way of learning conjunctive classifiers that minimize the expected prediction error. These predictive models have the benefit of being highly interpretable since their decision function is simple, and they typically rely on a few rules that can be inspected by domain experts (e.g., as in Drouin et al., 2019). However, it has one significant pitfall: nothing prevents $h(\mathbf{x})$ from relying on spurious associations between $\mathbf{X}$ and Y. In Section 3, we set out to alleviate this issue.

[1] See Koller & Friedman (2009) (Chap. 3) for an introduction.

[2] Not to be confused with ***S**tructural **C**ausal **M**odels*. This low-probability clash is unfortunately beyond our control.

[3] Hence the name: Set *Covering* Machines

[4] Since h is a conjunction, $h(\mathbf{x}) = 0$ if any $r_i(\mathbf{x}) = 0$. The outcome of the model cannot be changed by subsequent rules.

Algorithm 1 Set Covering Machine

Input: $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ a data sample
Input: $\mathcal{R}$ a set of candidate binary-valued rules
Input: $p \in \mathbb{R}^+$ hyperparameter of utility score
Input: $n \in \mathbb{N}$ hyperparameter for conjunction length
$\mathcal{P} \leftarrow \{(\mathbf{x}, y) \in \mathcal{S} \mid y = 1\}$ ▷ Positive examples
$\mathcal{N} \leftarrow \{(\mathbf{x}, y) \in \mathcal{S} \mid y = 0\}$ ▷ Negative examples
$\mathcal{H} \leftarrow \{\}$ ▷ Rules in the conjunction
while $|\mathcal{H}| < n$ and $\mathcal{N}$ is not empty **do**
  for each rule $r_i \in \mathcal{R}$ **do**
    $\mathcal{A}_i \leftarrow \{(\mathbf{x}, y) \in \mathcal{N} \mid r_i(\mathbf{x}) = y\}$
    $\mathcal{B}_i \leftarrow \{(\mathbf{x}, y) \in \mathcal{P} \mid r_i(\mathbf{x}) \neq y\}$
    $U_i = |\mathcal{A}_i| - p \cdot |\mathcal{B}_i|$ ▷ Utility computation
  end for
  $i^\star = \underset{i}{\operatorname{argmax}}\, U_i$
  $\mathcal{H} \leftarrow \mathcal{H} \cup \{r_{i^\star}\}$ ▷ Append best utility rule to the model
  $\mathcal{N} \leftarrow \mathcal{N} \setminus \mathcal{A}_{i^\star}$ ▷ Remove final samples
  $\mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{B}_{i^\star}$
end while
output the conjunction $h(x) = \bigwedge_{r \in \mathcal{H}} r(x)$

2.3. Invariant Causal Prediction

The two most related works from the causal inference literature are the seminal works of Peters et al. (2016) and Heinze-Deml et al. (2018) on invariant causal prediction. Both of these works rely on a multi-environment setting like the one described at Section 2.1. Moreover, both are based on the idea that the conditional distribution of Y, given all of its causal parents $\mathbf{X}_A$, should be invariant across environments; an idea that we also exploit in Section 3. In contrast with our work, Peters et al. (2016) require the structural equation between Y and $\mathbf{X}_A$ to be linear, while we assume it to be conjunctive (see Assumption 2.4). As for Heinze-Deml et al. (2018), they consider non-linear structural equations, which are compatible with our setting. However, identifying $\mathbf{X}_A$ using their approach requires to perform conditional independence tests for all sets of potential parents, which requires $O(2^{|\mathbf{X}|})$ time, where $|\mathbf{X}|$ is the number of feature variables (see Section 2.1). In sharp contrast, we show that, by exploiting the conjunctive nature of the structural equation of Y, the causal parents $\mathbf{X}_A$ can be identified in polynomial time.

3. Invariant Causal Set Covering Machines

We now propose an extension of the classical Set Covering Machine algorithm that exploits invariances across environments (see Assumption 2.3) to learn classifiers that rely solely on the causal parents $\mathbf{X}_A$ of Y.

Tree-based representation: To facilitate the presentation, let us introduce an alternative perspective on conjunctions. Let $f(x) \stackrel{\text{def}}{=} r_1(x) \wedge \ldots \wedge r_d(x)$ be an arbitrary conjunction of binary-valued rules r_i applied to some input x. If one assumes an ordering in the evaluation of the rules $r_i(x)$, then $f(x)$ corresponds to a simple decision tree composed of a single branch. This is illustrated at Fig. 2. Such a tree has d *negative leaves* where $f(x) = 0$, which are attained when a rule $r_i(x) = 0$, and a single *positive leaf* where $f(x) = 1$, which is attained when $r_i(x) = 1\ \forall i \in \{1, \ldots, d\}$.

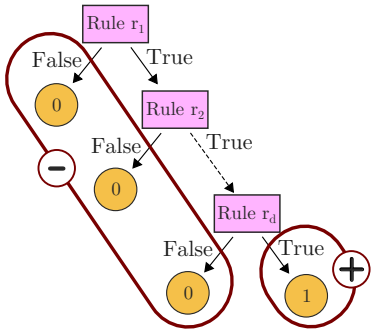

Figure 2. Tree-based representation of a d-rule conjunction. Positive and negative leaves are emphasized with $+$ and $-$, respectively.

With this perspective in mind, we propose the following theoretical result.

Theorem 3.1. *(Model construction criteria) Assume that the data-generating process follows the causal graph depicted at Fig. 1 and that Assumptions 2.1, 2.2, 2.3, and 2.4 hold. Let $f(\mathbf{X}^\star, \epsilon) = g(r_1(\mathbf{X}^\star, \epsilon_1) \wedge \ldots \wedge r_d(\mathbf{X}^\star, \epsilon_d), \epsilon_g)$, with $\mathbf{X}^\star \subseteq \mathbf{X}$, be an arbitrary conjunction of d binary-valued rules. Without loss of generality, assume an arbitrary ordering of the rules $1 \ldots d$ and consider the tree-based representation depicted in Fig. 2. We have that, if:*

i) the distribution of Y in the i-th negative leaf satisfies

$$Y \perp\!\!\!\perp_P E \mid \mathbf{r}_{<i}(\mathbf{X}^\star, \epsilon_{<i}) = \mathbf{1}, r_i(\mathbf{X}^\star, \epsilon_i) = 0, \quad (2)$$

where $\mathbf{r}_{<i}(\mathbf{X}^\star, \epsilon_{<i}) = \mathbf{1}$ denotes that all rules preceding r_i in the ordering have value 1, and

ii) the distribution of Y in the positive leaf satisfies

$$Y \perp\!\!\!\perp_P E \mid \mathbf{r}_{<d}(\mathbf{X}^\star, \epsilon_{<d}) = \mathbf{1}, r_d(\mathbf{X}^\star, \epsilon_d) = 1, \quad (3)$$

then $\mathbf{X}_A \subseteq \mathbf{X}^\star$ and $\mathbf{X}_C \cap \mathbf{X}^\star = \emptyset$.

The proof exploits the conjunctive nature of f (see App. A.1).

Model construction: Theorem 3.1 gives us criteria that can be evaluated at each step of building a conjunction and guarantee reliance on all $\mathbf{X}_A$ but none of the $\mathbf{X}_C$. That is, we can simply modify Algorithm 1 to i) disregard all rules where Criterion (2) is not satisfied when searching for the rule of maximal utility and ii) continue adding rules to the conjunction until Criterion (3) is satisfied. A detailed pseudocode and implementation details are given at App. B.

However, note that Theorem 3.1 does not guarantee the *minimality* of $\mathbf{X}^\star$, i.e., Criteria (2) and (3) could be satisfied

even if $\mathbf{X}_B \cap \mathbf{X}^\star \neq \emptyset$. Hence, we propose the following pruning procedure, which can be applied as long as none of the rules r_i jointly relies on elements from both $\mathbf{X}_A$ and $\mathbf{X}_B$. For example, this holds in the common setting where the rules are threshold functions on the value of a single feature.

Proposition 3.2. *(Pruning) Assume that Assumptions 2.1 and 2.2 hold. Let $f(\mathbf{X}^\star, \epsilon)$ be a conjunction that satisfies Equations 2 and 3, but is non-minimal, i.e., $\mathbf{X}_B \cap \mathbf{X}^\star \neq \emptyset$. For any $\mathbf{X}^\dagger \in \mathbf{X}^\star$, we have that $Y \perp\!\!\!\perp_P E \mid \mathbf{X}^\star \setminus \mathbf{X}^\dagger$ if and only if $\mathbf{X}^\dagger \notin \mathbf{X}_A$.*

Hence, the $f(\mathbf{X}^\star, \epsilon)$ can be pruned, iteratively, by applying a conditional independence test to each $\mathbf{X}^\dagger \in \mathbf{X}^\star$ and removing any rule that makes use of non-causal-parents of Y. The proof is given at App. A.2.

Invariant Causal Set Covering Machines: By combining the construction Criteria (2) and (3) with Proposition 3.2, we obtain a simple modification of the Set Covering Machine algorithm that can provably *identify* all the causal parents X_A of Y. Of note, the runtime complexity is $O(m \cdot |\mathcal{R}| \cdot n)$, where $|\mathcal{R}|$ is typically linear w.r.t. $|\mathbf{X}|$. This is in sharp contrast with the exponential runtime complexity of non-linear ICP.

4. Results

We now report a brief empirical study. These experiments are by no means extensive, but they support the validity of the proposed algorithm and theory.

Simulated Data: We parametrize a discrete Bayesian network that satisfies the assumptions at Section 2.1. We define two causal parents $\mathbf{X}_A = [\mathbf{X}_{A1}, \mathbf{X}_{A2}]$ and a single descendent $\mathbf{X}_C$ for Y. We let $\mathbf{X}_B$ be a distractor, unrelated to Y and vary $|\mathbf{X}_B|$ throughout the experiments. Of particular note, the Bayesian network is designed such that the relationship between Y and $\mathbf{X}_c$ is less noisy than the relationship between Y and its causal parents $\mathbf{X}_A$. As such, algorithms that are vulnerable to spurious associations will tend to use $\mathbf{X}_C$ instead of $\mathbf{X}_A$. In all experiments, we use $m = 10^4$ samples from each of two environments (E). See App. C.1 for details.

Baselines: The baselines that we consider are Set Covering Machines (SCM; Marchand & Shawe-Taylor (2002)), decision trees (DT; Breiman et al. (1984)), and non-linear ICP (ICP; Heinze-Deml et al. (2018)). ICP should be robust to spurious associations, while DT and SCM should not. See App. C.2 for implementation details.

Identification of causal parents: Table 1 compares the ability of all methods to identify the causal parents $\mathbf{X}_A$ of Y. As expected, SCM and DT always rely on spurious associations and fail to identify $\mathbf{X}_A$. On the contrary, both ICP and ICSCM succeed at identifying $\mathbf{X}_A$ in most cases. The performance of ICP degrades as dimensionality increases, likely due to type II errors that arise due to the vanishing statistical power of its conditional independence tests. In contrast, ICSCM appears less affected by dimensionality; this seems to hold even up to hundreds of variables (see Table 3). At App. D, we report additional results that support these claims and provide an extensive discussion on the effect of type I and II errors on both methods. Finally, it is clear from these results that ICSCM endows SCM with the ability to identify causal parents, which was the main objective of this work.

Table 1. Identification of the causal parents $\mathbf{X}_A$: proportion of 100 training runs where the model relied solely on $\mathbf{X}_A$, for an increasing number of distractor features $\mathbf{X}_B$ (1 to 7). Worst values colored.

Model	1	2	3	4	5	6	7
SCM	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DT	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ICP	0.96	0.98	0.99	0.99	0.97	0.09	0.00
ICSCM	0.96	0.97	0.99	0.96	0.97	0.96	0.96

Runtime complexity analysis: Fig. 3 shows the running time of all methods with respect to dimensionality. Clearly, the running time of ICP increases exponentially fast, while that of ICSCM increases linearly, making it more amenable to real-world applications where variables are plentiful.

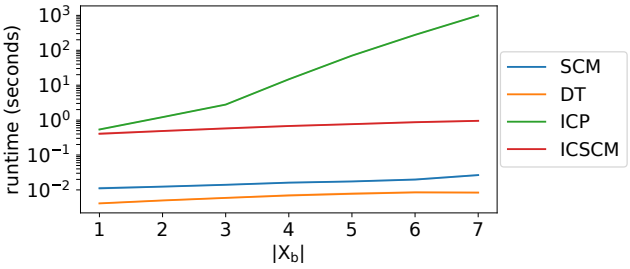

Figure 3. Running time w.r.t. the size of $\mathbf{X}_B$

5. Conclusion

This work introduced ICSCM, a learning algorithm that builds conjunctive and disjunctive models that, in some settings, are guaranteed to rely exclusively on the causal parents of a variable of interest. The algorithm is supported by a rigorous theoretical foundation that exploits invariances that hold in a multi-environment setting to avoid spurious associations. In contrast with previous work, such as non-linear ICP (Heinze-Deml et al., 2018), ICSCM leverages the nature of the models to considerably reduce the number of statistical tests required, making it more amenable to practical applications. In future work, we will consider relaxations of the current assumptions (e.g., variants of graph G, unobserved confounding), conduct a deeper study of ICSCM's empirical performance, and explore applications to real-world problems, such as the identification of biomarkers in high-dimensional genomics data (Drouin et al., 2019).

References

Breiman, L., Friedman, J., Olshen, R., and Stone, C. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.

Bühlmann, P. Invariance, causality and robustness. *Statistical Science*, 35(3), Aug 2020. ISSN 0883-4237. doi: 10.1214/19-STS721.

Drouin, A., Letarte, G., Raymond, F., Marchand, M., Corbeil, J., and Laviolette, F. Interpretable genotype-to-phenotype classifiers with performance guarantees. *Scientific Reports*, 9(11):4071, Mar 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-40561-2.

Glymour, C., Zhang, K., and Spirtes, P. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

Heinze-Deml, C., Peters, J., and Meinshausen, N. Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 6(2), 2018.

Koller, D. and Friedman, N. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Marchand, M. and Shawe-Taylor, J. The set covering machine. *Journal of Machine Learning Research*, 3(4-5): 723–746, 2002.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Peters, J., Bühlmann, P., and Meinshausen, N. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 78(5): 947–1012, 2016. ISSN 1369-7412.

Szymczak, S., Biernacka, J. M., Cordell, H. J., González-Recio, O., König, I. R., Zhang, H., and Sun, Y. V. Machine learning in genome-wide association studies. *Genetic epidemiology*, 33(S1):S51–S57, 2009.

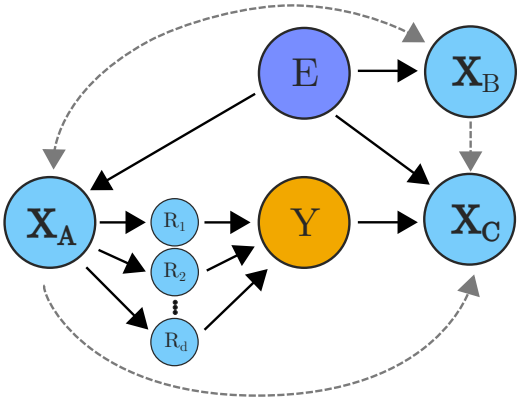

Figure 4. Implicit variables for binary-valued rules: this figure shows an expanded version of the causal graph illustrated in Fig. 1 where the rules in the conjunction (see Eq. (1)) are represented as random variables that mediate all paths from $\mathbf{X}_A$ to Y.

A. Proofs

A.1. Proof of Theorem 3.1

Theorem 3.1. (Model construction criteria) Assume that the data-generating process follows the causal graph depicted at Fig. 1 and that Assumptions 2.1, 2.2, 2.3, and 2.4 hold. Let $f(\mathbf{X}^\star, \epsilon) = g(r_1(\mathbf{X}^\star, \epsilon_1) \wedge \ldots \wedge r_d(\mathbf{X}^\star, \epsilon_d), \epsilon_g)$, with $\mathbf{X}^\star \subseteq \mathbf{X}$, be an arbitrary conjunction of d binary-valued rules. Without loss of generality, assume an arbitrary ordering of the rules $1 \ldots d$ and consider the tree-based representation depicted in Fig. 2. We have that, if:

i) the distribution of Y in the i-th negative leaf satisfies:

$$Y \perp\!\!\!\perp_P E \mid \mathbf{r}_{<i}(\mathbf{X}^\star, \epsilon_{<i}) = \mathbf{1}, r_i(\mathbf{X}^\star, \epsilon_i) = 0, \tag{4}$$

where $\mathbf{r}_{<i}(\mathbf{X}^\star, \epsilon_{<i}) = \mathbf{1}$ denotes that all rules preceding r_i in the ordering have value 1, and

ii) the distribution of Y in the positive leaf satisfies:

$$Y \perp\!\!\!\perp_P E \mid \mathbf{r}_{<d}(\mathbf{X}^\star, \epsilon_{<d}) = \mathbf{1}, r_d(\mathbf{X}^\star, \epsilon_d) = 1, \tag{5}$$

then we have that $\mathbf{X}_A \subseteq \mathbf{X}^\star$ and $\mathbf{X}_C \cap \mathbf{X}^\star = \emptyset$.

Proof. The proof is divided in two cases and we proceed by contradiction. For simplicity, we use R_i to denote the random variable $r_i(\mathbf{X}^\star, \epsilon_i)$ and will abuse the notation by using r_i to denote a value taken on by R_i.

Case 1: Suppose that the properties at Eq. (4) and Eq. (5) hold, but that $\mathbf{X}_A \not\subseteq \mathbf{X}^\star$. We have that $\mathbf{X}_A \not\subseteq \mathbf{X}^\star$ and thus some of the causal parents of Y are not in $\mathbf{X}^\star$, i.e., there exists $X_{A_i} \in \mathbf{X}_A$ such that $X_{A_i} \notin \mathbf{X}^\star$. Hence, we have that $Y \not\perp\!\!\!\perp_P E \mid \mathbf{X}^\star$, because there exists an unblocked path $E \to X_{A_i} \to Y$. Because $\mathbf{X}^\star$ contains all the causal parents of $R_1, \ldots, R_d$, we also have that $Y \not\perp\!\!\!\perp_P E \mid R_1, \ldots, R_d$. By the definition of conditional dependence, this means that $\exists y, e, r_1, \ldots, r_d$ such that $P(Y = y \mid R_1 = r_1, \ldots, R_d = r_d, E = e) \neq P(Y = y \mid R_1 = r_1, \ldots, R_d = r_d)$. Recall that, given the conjunctive nature of $f(\mathbf{X}^\star, \epsilon)$, each combination of $r_1, \ldots, r_d$ corresponds to a leaf in the conjunction (see Fig. 2). Therefore, depending on the value of $r_1, \ldots, r_d$, we will reach a contradiction for either a negative or a positive leaf:

- **Negative leaves:** $\exists j, r_j = 0$, then we have that $P(Y \mid R_1 = 1, \ldots, R_{j-1} = 1, R_j = 0, E = e) \neq P(Y \mid R_1 = 1, \ldots, R_{j-1} = 1, R_j = 0)$, this is in contradiction with Eq. (4).
- **Positive leaf:** $\forall j, r_j = 1$, then we have that $P(Y = y \mid R_1 = 1, \ldots, R_d = 1, E = e) \neq P(Y = y \mid R_1 = 1, \ldots, R_d = 1)$, this is in contradiction with Eq. (5).

Case 2 : Suppose that the properties at Eq. (4) and Eq. (5) hold, but that $\mathbf{X}_C \cap \mathbf{X}^\star \neq \emptyset$. There are some descendants of Y in $\mathbf{X}^\star$, i.e., there exists $X_{C_i} \in \mathbf{X}_C$ such that $X_{C_i} \in \mathbf{X}^\star$. Recall, from our graphical assumptions (see Fig. 1), the existence

Table 2. Probability table used in random generation of causal parents in simulated data.

Environment	$E = 0$	$E = 1$
$P(X_{A_1} = 1)$	0.1	0.5
$P(X_{A_2} = 1)$	0.5	0.3

of the v-structure $Y \rightarrow X_C \leftarrow E$. Since, $X_{C_i} \in \mathbf{X}^\star$, there exists at least one R_j such that $X_{C_i} \rightarrow R_j$. Because conditioning on R_j opens the path $E - X_{C_i} - Y$, we have that $Y \not\perp\!\!\!\perp_P E \mid R_1, \ldots, R_d$.

By the definition of conditional dependence, this means that $\exists y, e, r_1, \ldots, r_d$ such that $P(Y = y \mid R_1 = r_1, \ldots, R_d = r_d, E = e) \neq P(Y = y \mid R_1 = r_1, \ldots, R_d = r_d)$. Recall that, given the conjunctive nature of $f(\mathbf{X}^\star, \epsilon)$, each combination of $r_1, \ldots, r_d$ corresponds to a leaf in the conjunction (see Fig. 2). Therefore, depending on the value of $r_1, \ldots, r_d$, we will reach a contradiction for either a negative or a positive leaf:

- **Negative leaves:** $\exists k, r_k = 0$, then we have that $P(Y \mid R_1 = 1, \ldots, R_{k-1} = 1, R_k = 0, E = e) \neq P(Y \mid R_1 = 1, \ldots, R_{k-1} = 1, R_k = 0)$, this is in contradiction with Eq. (4).
- **Positive leaf:** $\forall k, r_k = 1$, then we have that $P(Y = y \mid R_1 = r_1, \ldots, R_d = r_d, E = e) \neq P(Y = y \mid R_1 = 1, \ldots, R_d = 1)$, this is in contradiction with Eq. (5).

□

A.2. Proof of Proposition 3.2

Proposition 3.2. (Pruning) Assume that Assumptions 2.1 and 2.2 hold. Let $f(\mathbf{X}^\star, \epsilon)$ be a conjunction that satisfies Equations (2) and (3), but is non-minimal, i.e., $\mathbf{X}_B \cap \mathbf{X}^\star \neq \emptyset$. For any $\mathbf{X}^\dagger \in \mathbf{X}^\star$, we have that $Y \perp\!\!\!\perp_P E \mid \mathbf{X}^\star \setminus \mathbf{X}^\dagger$ if and only if $\mathbf{X}^\dagger \notin \mathbf{X}_A$.

Proof. Since $f(\mathbf{X}^\star, \epsilon)$ satisfies Equations (2) and (3), we know that $\mathbf{X}_A \subseteq \mathbf{X}^\star$ and $\mathbf{X}_C \cap \mathbf{X}^\star = \emptyset$. We now consider two cases. First, let us reason about sufficiency and show that $Y \perp\!\!\!\perp_P E \mid \mathbf{X}^\star \setminus \mathbf{X}^\dagger \Rightarrow \mathbf{X}^\dagger \notin \mathbf{X}_A$. This follows directly from the faithfulness assumption (Assumption 2.2), since taking $\mathbf{X}^\dagger$ from $\mathbf{X}_A$ would open at least one path from E to Y in G (see Fig. 1), contradicting the conditional independence statement. Second, let us reason about necessity and show that $\mathbf{X}^\dagger \notin \mathbf{X}_A \Rightarrow Y \perp\!\!\!\perp_P E \mid \mathbf{X}^\star \setminus \mathbf{X}^\dagger$. This follows directly from the causal Markov assumption (Assumption 2.1) since removing an element of $\mathbf{X}_B$ leaves all paths in G from E to Y blocked. □

B. ICSCM: pseudocode and implementation

The detailed pseudo-code of ICSCM is presented in Algorithm 2. The differences with the standard Set Covering Machine algorithm (Algorithm 1) are emphasized in color.

Implementation: The code for ICSCM is available at https://github.com/GRAAL-Research/icscm.

C. Experimental details

C.1. Simulated Data Generation

The code for all the experiments is available at: https://github.com/GRAAL-Research/icscm.

Simulated data is generated to follow the causal graph in Fig. 1. We define $n_{env} = 2$ environments, each one contains $n_{samples} = 10000$ observations. For each observation, the causal parent variables X_{A_1} and X_{A_2} are binary values randomly generated given the probability Table 2. Note that the environment E affects X_A by changing the distributions of X_{A_1} and X_{A_2}. The value of Y is generated with the function described in Assumption 2.4 : $f(\mathbf{X}_A, \epsilon) \stackrel{\text{def}}{=} g_y(r_1(\mathbf{X}_A, \epsilon_1) \wedge r_d(\mathbf{X}_A, \epsilon_d), \epsilon_y)$, with

$$Y = g_y(r_1(\mathbf{X}_A, \epsilon_1) \wedge r_2(\mathbf{X}_A, \epsilon_2), \epsilon_y) \stackrel{\text{def}}{=} g_y(X_{A_1} \wedge X_{A_2}, \epsilon_y) \stackrel{\text{def}}{=} \begin{cases} X_{A_1} \wedge X_{A_2}, & \text{if } t = 0 \\ 1 - X_{A_1} \wedge X_{A_2}, & \text{if } t = 1 \end{cases}, t \sim \mathrm{B}(\epsilon_y) \tag{6}$$

Algorithm 2 Invariant Causal Set Covering Machine

Input: $\mathcal{S} = \{(\mathbf{x}_i, y_i, e_i)\}_{i=1}^{m}$ a data sample
Input: $\mathcal{R}$ a set of candidate binary-valued rules
Input: $p \in \mathbb{R}^+$ hyperparameter of utility score
Input: $n \in \mathbb{N}$ hyperparameter for conjunction length
Input: $\alpha \in \mathbb{R}^+$ Hyperparameter for independence threshold

// Initialization
$\mathcal{P} \leftarrow \{(\mathbf{x}, y, e) \in \mathcal{S} \mid y = 1\}$ ▷ Positive examples
$\mathcal{N} \leftarrow \{(\mathbf{x}, y, e) \in \mathcal{S} \mid y = 0\}$ ▷ Negative examples
$\mathcal{H} \leftarrow \{\}$ ▷ Rules in the conjunction
$\gamma \leftarrow 1$ ▷ A stopping criterion indicator

// Iterative construction of the conjunction
while $|\mathcal{H}| < n$ and $\mathcal{N}$ is not empty and $\gamma < \alpha$ **do**

// Calculate a utility value for each candidate rule
for each rule $r_i \in \mathcal{R}$ **do**
$\mathcal{A}_i \leftarrow \{(\mathbf{x}, y, e) \in \mathcal{N} \mid r_i(\mathbf{x}) = y\}$
$\mathcal{B}_i \leftarrow \{(\mathbf{x}, y, e) \in \mathcal{P} \mid r_i(\mathbf{x}) \neq y\}$
$\pi \leftarrow$ p-value of a statistical independence test between $\{e_j \mid (\mathbf{x}_j, y_j, e_j) \in \mathcal{A}_i \cup \mathcal{B}_i\}$ and $\{y_j \mid (\mathbf{x}_j, y_j, e_j) \in \mathcal{A}_i \cup \mathcal{B}_i\}$
$U_i = (|\mathcal{A}_k| - p \cdot |\mathcal{B}_k|)$ **if** $\pi > \alpha$ **else** $-\infty$
end for

// Add the best rule to the conjunction
$i^\star = \underset{i}{\operatorname{argmax}}\, U_i$
if $U_{i^\star} = -\infty$ **then**
break ▷ Stop adding rules to the conjunction
end if
// Remove examples for which $h(\mathbf{x})$ is final
$\mathcal{N} \leftarrow \mathcal{N} \setminus \mathcal{A}_{i^\star}$
$\mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{B}_{i^\star}$
$\gamma \leftarrow$ the p-value of a statistical independence test between $\{e_j \mid (\mathbf{x}_j, y_j, e_j) \in \mathcal{P} \cup \mathcal{N}\}$ and $\{y_j \mid (\mathbf{x}_j, y_j, e_j) \in \mathcal{P} \cup \mathcal{N}\}$
end while
// [Optionally] Conduct pruning according to the procedure based on Proposition 3.2 described at Section 3.
output the conjunction $h(x) = \bigwedge_{r \in \mathcal{H}} r(x)$

In our experiments, $\epsilon_y = 0.05$. Note that Y depends only on its causal parents and not directly on E.

Then, the value of X_C is generated given the following function :

$$X_C = \begin{cases} Y, & \text{if } u = 0 \\ E, & \text{if } u = 1 \end{cases}, u \sim \mathrm{B}(\epsilon_{X_C}) \tag{7}$$

This reflects the effect of both Y and E on $\mathbf{X}_C$. In our experiments, $\epsilon_{\mathbf{X}_C} = 0.05$. Note that, when $\epsilon_y = \epsilon_{\mathbf{X}_C}$, $\mathbf{X}_C$ is a better predictor of Y than $X_{A_1} \wedge X_{A_2}$, because even if the noise term $u = 1$, X_C can still have the same value than Y when $E = Y$.

Then, the variables in $\mathbf{X}_B$ are generated as Bernoulli random variables $X_{B_i} \sim \mathrm{B}(\epsilon_{X_{B_i}})_{i=1}^{|\mathbf{X}_B|}$, with $\epsilon_{X_{B_i}} = 0.5$. In our experiments, we made $|\mathbf{X}_B|$ vary from 1 to 7.

C.2. Baselines and implementation details

For the Set Covering Machine (SCM) and classification tree (DT) baselines, as well as ICSCM, we conducted a hyperparameter search using 5-fold cross-validation and selected the values that led to the highest binary accuracy, on average, over all folds. Details are provided below:

Table 3. Identification of the causal parents $\mathbf{X}_A$: proportion of 100 training runs where the model relied solely on $\mathbf{X}_A$, for an increasing number of distractor features $\mathbf{X}_B$ (1 to 200).

$\mathbf{X}_B$	1	2	3	4	5	6	7	10	15	20	25	50	100	200
ICSCM	0.96	0.97	0.99	0.96	0.97	0.96	0.96	0.89	0.89	0.86	0.96	0.90	0.91	0.94

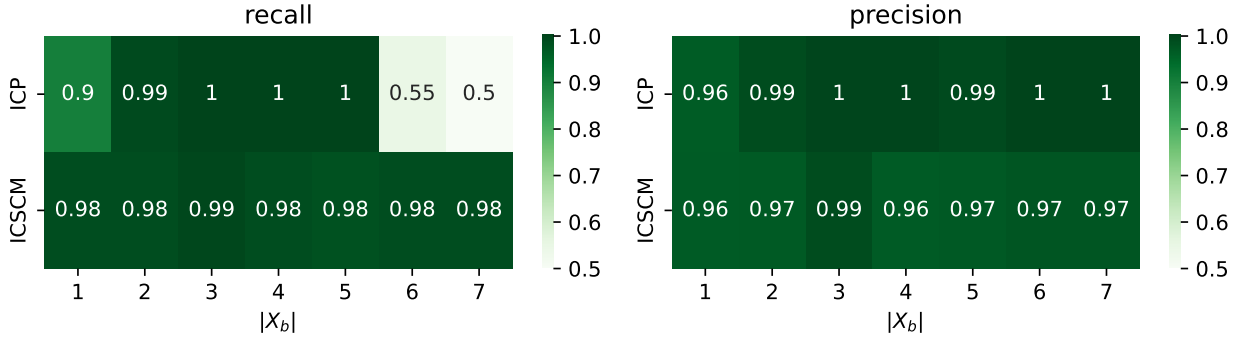

Figure 5. Precision and recall metrics on the task of identifying the set of causal parents for ICP and ICSCM. The scores are computed for several values of $|X_B|$, and averaged over 100 randomly generated datasets, using the experimental design presented in App. C.1.

- Set Covering Machine (SCM; Marchand & Shawe-Taylor (2002)): We used the implementation available at https://github.com/aldro61/pyscm (version 1.1.0) and considered the following hyperparameter values for trade-off p: $\{0.1, 0.5, 0.75, 1.0, 2.5, 5, 10\}$. The models built were conjunctions.
- Classification trees (DT; Breiman et al. (1984)): We used the implementation available in Scikit-Learn (version 1.2.2; Pedregosa et al. (2011)) and considered the following hyperparameter values: i) maximum depth: $\{1, 2, 3, 4, 5, 10\}$, ii) minimum samples split: $\{2, 0.01, 0.05, 0.1, 0.3\}$.
- ICSCM: We implemented the pseudo-code showed in Algorithm 2 in Python and considered the following hyperparameter values for trade-off p: $\{0.1, 0.5, 0.75, 1.0, 2.5, 5, 10\}$. The models built were conjunctions. Conditional independence was tested using a χ^2 test with $\alpha = 0.05$. For the pruning procedure, we used the conditional G-test implementation available at https://github.com/keiichishima/gsq, also with $\alpha = 0.05$.

For the ICP baseline, we reimplemented the method in Heinze-Deml et al. (2018). The implementation is available in our main codebase. Conditional independence was tested using a conditional G test with $\alpha = 0.05$. We used the conditional G-test implementation available at https://github.com/keiichishima/gsq.

D. Robustness to Type I and Type II Errors

While algorithms like ICP and ICSCM offer strong theoretical guarantees, their empirical performance is subject to the reliability of the (conditional) independence tests that they perform. Empirical phenomena such as *type I* errors (false positive: finding dependence when there is none) and *type II* errors (false negative: finding independence when there is dependence) can affect their performance in practice. Type I errors can be controlled via the α threshold on the p-value of the statistical tests. However, type II errors are more difficult to control, as the false negative rate β depends on the statistical power of a test, which in turn depends on factors such as the effect size (which we do not control) and the number of available data samples. Here, we discuss the effect of each type of error on both algorithms.

ICP: To find the set of causal parents $\mathbf{X}_A$, ICP tests the independence of Y and E conditioned on every possible set of variables, resulting in $2^{|\mathbf{X}|}$ conditional independence tests. Then, the algorithm takes the intersection of all sets that were found to lead to independence (see Heinze-Deml et al. (2018), line 4 of Algorithm 1 in Appendix B). The intersection has an effect similar to the pruning procedure of ICSCM (see Proposition 3.2) and serves to filter out $\mathbf{X}_B$ from the solution. ICP is quite robust to type I errors since such errors result in discarding only a few of the sets that contain $\mathbf{X}_A$. Apart from rare cases, such as making a type I error for the set $\{\mathbf{X}_A\}$ and not for all sets containing both $\mathbf{X}_A$ and $\mathbf{X}_B$, the intersection renders the algorithm

robust to type I errors. On the other hand, ICP is vulnerable to type II errors, which can cause it to incorrectly include a set that does not lead to independence, e.g., $\{\mathbf{X}_C\}$ or $\{X_{A1}\}$ (but missing some $X_{Ai} \in \mathbf{X}_A$) in the intersection. If this happens, the intersection returns either the empty set or a partial set of causal parents. As the dimensionality of $\mathbf{X}$ increases (for a fixed sample size), the power of the statistical tests decreases and the probability β of making such errors increases, amplifying the problem. We hypothesize that this is what causes the poor performance of ICP for the larger sizes of $\mathbf{X}_B$ at Table 1. This is supported by the results at Fig. 5, which show that the *recall* of ICP on the task of identifying the set of causal parents drops for $|\mathbf{X}_B| \geq 6$. In contrast, note that the *precision* remains stables (see Fig. 5), illustrating the robustness of this method to type I errors.

ICSCM: We separate the discussion of how ICSCM can be affected by type I and II errors into three parts, based on the different stages of the algorithm. For simplicity, let us assume that the set of candidate rules $\mathcal{R}$ contains a single rule per causal parent.

1. **Effect on Eq. (2):** This criterion is used to select which rules are permitted to be added to the model. A type I error has the effect of rejecting a rule that actually satisfies the criterion, e.g., rejecting a causal parent in $\mathbf{X}_A$. If this happens, the resulting conjunction might not contain all the causal parents. However, note that, even if a causal parent is rejected at one stage of building the conjunction, the data filtering that occurs at the end of every iteration in Algorithm 2 results in re-testing the same rule with a subset of the data at the next iteration, which might offer some resistance to type I errors. As for type II errors, these correspond to incorrectly believing that the criterion is satisfied and could result in adding rules that depend on $\mathbf{X}_C$ to the conjunction. In both cases, if such errors were to be made, the stopping criterion at Eq. (3) would not be satisfiable due to unblocked paths between Y and E in G. In terms of precision and recall w.r.t. the causal parents, type I and type II errors would result in lower recall and precision, respectively.

2. **Effect on Eq. (3):** This criterion is used to determine when to stop adding rules to the model. Here, a type I error would result in a failure to stop. However, note that, even if the algorithm failed to stop, as long as all R_i have been added to the conjunction, the algorithm should find that no other R_j satisfies Eq. (2) and stop, offering some robustness to such errors. As for type II errors, these would result in incorrectly concluding that the criterion is satisfied, resulting in premature stopping and missing causal parents in $\mathbf{X}_A$. Hence, type I and type II errors would result in lower precision and recall, respectively.

3. **Effect on Proposition 3.2:** This result is the foundation for the pruning procedure of ICSCM. Here, a type I error would result in incorrectly keeping a variable in $\mathbf{X}_B$ in the conjunction instead of pruning it. In contrast, a type II error would result in incorrectly removing a variable in $\mathbf{X}_A$ from the conjunction. Type I and type II errors would therefore result in lower precision and recall, respectively.

Finally, note that while we typically cannot control all the factors that govern type II errors (β), there is one key element that distinguishes ICSCM from ICP: ICP must conduct conditional independence testing for every possible set of parents, resulting in large conditioning sets. In contrast, ICSCM's tests are only conditioned on a number of variables that grows linearly with the length of the conjunction. As such, ICSCM's tests may have greater power and the algorithm may be less affected by type II errors. The results at Fig. 5, show that both the precision and recall of ICSCM remain high as the number of variables increases, in contrast with ICP whose recall decreases significantly.