Detecting Dependencies in Sparse, Multivariate Databases Using Probabilistic Programming and Non-parametric Bayes

Feras Saad
Probabilistic Computing Project
Massachusetts Institute of Technology

Vikash Mansinghka
Probabilistic Computing Project
Massachusetts Institute of Technology

Abstract

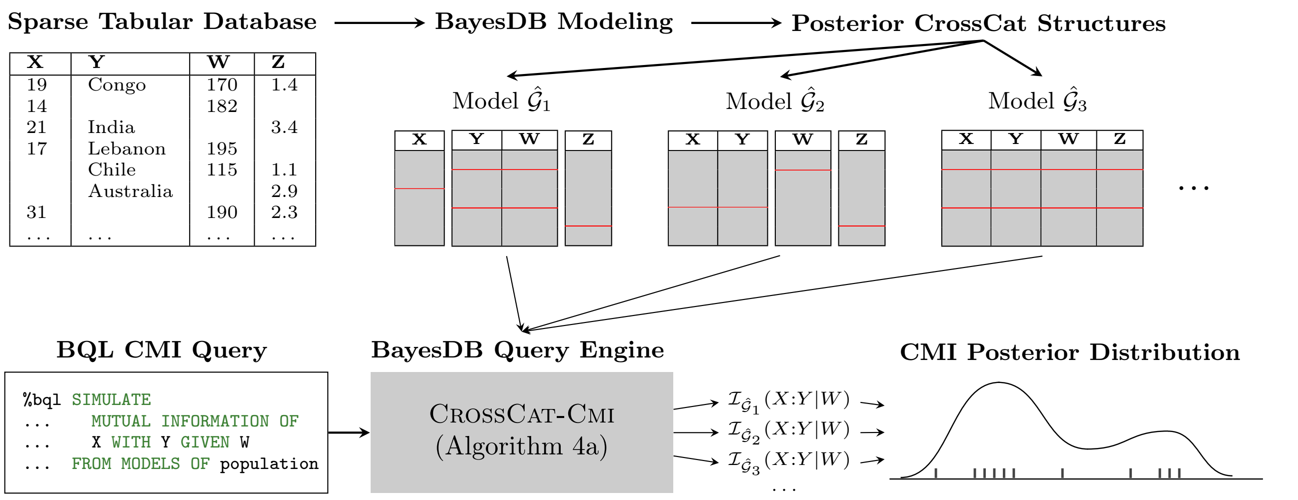
Datasets with hundreds of variables and many missing values are commonplace. In this setting, it is both statistically and computationally challenging to detect true predictive relationships between variables and also to suppress false positives. This paper proposes an approach that combines probabilistic programming, information theory, and non-parametric Bayes. It shows how to use Bayesian non-parametric modeling to (i) build an ensemble of joint probability models for all the variables; (ii) efficiently detect marginal independencies; and (iii) estimate the conditional mutual information between arbitrary subsets of variables, subject to a broad class of constraints. Users can access these capabilities using BayesDB, a probabilistic programming platform for probabilistic data analysis, by writing queries in a simple, SQL-like language. This paper demonstrates empirically that the method can (i) detect context-specific (in)dependencies on challenging synthetic problems and (ii) yield improved sensitivity and specificity over baselines from statistics and machine learning, on a real-world database of over 300 sparsely observed indicators of macroeconomic development and public health.

1 Introduction

Sparse databases with hundreds of variables are commonplace. In these settings, it can be both statistically and computationally challenging to detect predictive relationships between variables [4]. First, the data may be incomplete and require cleaning and imputation before pairwise statistics can be calculated. Second, parametric modeling assumptions that underlie standard hypothesis testing techniques may not be appropriate due to nonlinear, multivariate, and/or heteroskedastic relationships. Third, as the number of variables grows, it becomes harder to detect true relationships while suppressing false positives. Many approaches have been proposed (see [17, Table 1] for a summary), but they each exhibit limitations in practice. For example, some only apply to fully-observed real-valued data, and most do not produce probabilistically coherent measures of uncertainty. This paper proposes an approach to dependence detection that combines probabilistic programming, information theory, and non-parametric Bayes. The end-to-end approach is summarized in Figure 1. Queries about the conditional mutual information (CMI) between variables of interest are expressed using the Bayesian Query Language [18], an SQL-like probabilistic programming language. Approximate inference with CrossCat [19] produces an ensemble of joint probability models, which are analyzed for structural (in)dependencies. For model structures in which dependence cannot be ruled out, the CMI is estimated via Monte Carlo integration.

In principle, this approach has significant advantages. First, the method is scalable to high-dimensional data: it can be used for exploratory analysis without requiring expensive CMI estimation for all pairs of variables. Second, it applies to heterogeneously typed, incomplete datasets with minimal pre-processing [19]. Third, the non-parametric Bayesian joint density estimator used to form CMI estimates can model a broad class of data patterns, without overfitting to highly irregular data. This paper shows that the proposed approach is effective on a real-world database with hundreds of variables and a missing data rate of ∼35%, detecting common-sense predictive relationships that are missed by baseline methods while suppressing spurious relationships that baselines purport to detect.

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54. Copyright 2017 by the author(s).



X	Y	W	Z
19	Congo	170	1.4
14		182	
21	India		3.4
17	Lebanon	195	
	Chile	115	1.1
	Australia		2.9
31		190	2.3
...	...	...	...

```
%bql SIMULATE
...     MUTUAL INFORMATION OF
...     X WITH Y GIVEN W
... FROM MODELS OF population
```

Sparse Tabular Database
BayesDB Modeling
Posterior CrossCat Structures
Model $\hat{\mathcal{G}}_1$
Model $\hat{\mathcal{G}}_2$
Model $\hat{\mathcal{G}}_3$
X
Y
W
Z
...
BQL CMI Query
BayesDB Query Engine
CrossCat-Cmi
(Algorithm 4a)
$\mathcal{I}_{\hat{\mathcal{G}}_1}(X{:}Y|W)$
$\mathcal{I}_{\hat{\mathcal{G}}_2}(X{:}Y|W)$
$\mathcal{I}_{\hat{\mathcal{G}}_3}(X{:}Y|W)$
...
CMI Posterior Distribution

Figure 1: Workflow for computing posterior distributions of the CMI for variables in a data table using BayesDB. Modeling and inference in BayesDB produces an ensemble of posterior CrossCat samples. Each model learns a factorization of the joint distribution of all variables in the database, and a Dirichlet process mixture within each block of dependent variables. For instance, model $\hat{\mathcal{G}}_1$ specifies that X is independent of (Y, W) which in turn is independent of Z, while in $\hat{\mathcal{G}}_3$, all variables are (structurally) dependent. End-user queries for the CMI are expressed in the Bayesian Query Language. The BQL interpreter uses CrossCat structures to optimize the query where possible, by (i) bypassing Monte Carlo estimation completely when the queried variables are structurally independent, and/or (ii) dropping redundant constraints which are structurally independent of the queried variables. Values of CMI returned by each model constitute samples from the posterior CMI distribution.

2 Drawing Bayesian inferences about conditional mutual information

Let $\boldsymbol{x} = (x_1, x_2, \dots, x_D)$ denote a D-dimensional random vector, whose sub-vectors we denote $\boldsymbol{x}_\mathcal{A} = \{x_i : i \in \mathcal{A}\}$ with joint probability density $p_\mathcal{G}(\boldsymbol{x}_\mathcal{A})$. The symbol $\mathcal{G}$ refers to an arbitrary specification for the "generative" process of $\boldsymbol{x}$, and parameterizes all its joint and conditional densities. The *mutual information* (MI) of the variables $\boldsymbol{x}_\mathcal{A}$ and $\boldsymbol{x}_\mathcal{B}$ (under generative process $\mathcal{G}$) is defined in the usual way [5]:

$$\mathcal{I}_\mathcal{G}(\boldsymbol{x}_\mathcal{A}{:}\boldsymbol{x}_\mathcal{B}) = \mathop{\mathbb{E}}_{(\boldsymbol{x}_\mathcal{A}, \boldsymbol{x}_\mathcal{B})}\left[\log\left(\frac{p_\mathcal{G}(\boldsymbol{x}_\mathcal{A}, \boldsymbol{x}_\mathcal{B})}{p_\mathcal{G}(\boldsymbol{x}_\mathcal{A})p_\mathcal{G}(\boldsymbol{x}_\mathcal{B})}\right)\right]. \quad (1)$$

The mutual information can be interpreted as the KL-divergence from the product of marginals $p_\mathcal{G}(\boldsymbol{x}_\mathcal{A})p_\mathcal{G}(\boldsymbol{x}_\mathcal{B})$ to the joint distribution $p_\mathcal{G}(\boldsymbol{x}_\mathcal{A}, \boldsymbol{x}_\mathcal{B})$, and is a well-established measure for both the existence and strength of dependence between $\boldsymbol{x}_\mathcal{A}$ and $\boldsymbol{x}_\mathcal{B}$ (Section 2.2). Given an observation of the variables $\{\boldsymbol{x}_\mathcal{C} {=} \hat{\boldsymbol{x}}_\mathcal{C}\}$, the *conditional mutual information* (CMI) of $\boldsymbol{x}_\mathcal{A}$ and $\boldsymbol{x}_\mathcal{B}$ given $\{\boldsymbol{x}_\mathcal{C} {=} \hat{\boldsymbol{x}}_\mathcal{C}\}$ is defined analogously:

$$\mathcal{I}_\mathcal{G}(\boldsymbol{x}_\mathcal{A}{:}\boldsymbol{x}_\mathcal{B}|\boldsymbol{x}_\mathcal{C}{=}\hat{\boldsymbol{x}}_\mathcal{C}) = \mathop{\mathbb{E}}_{(\boldsymbol{x}_\mathcal{A}, \boldsymbol{x}_\mathcal{B})|\hat{\boldsymbol{x}}_\mathcal{C}}\left[\log\left(\frac{p_\mathcal{G}(\boldsymbol{x}_\mathcal{A}, \boldsymbol{x}_\mathcal{B}|\hat{\boldsymbol{x}}_\mathcal{C})}{p_\mathcal{G}(\boldsymbol{x}_\mathcal{A}|\hat{\boldsymbol{x}}_\mathcal{C})p_\mathcal{G}(\boldsymbol{x}_\mathcal{B}|\hat{\boldsymbol{x}}_\mathcal{C})}\right)\right]. \quad (2)$$

Estimating the mutual information between the variables of $\boldsymbol{x}$ given a dataset of observations $\mathcal{D}$ remains an open problem in the literature. Various parametric and non-parametric methods for estimating MI exist [21, 22, 15]; see [24] for a comprehensive review. Traditional approaches typically construct a point estimate $\hat{\mathcal{I}}(\boldsymbol{x}_\mathcal{A}{:}\boldsymbol{x}_\mathcal{B})$ (and possible confidence intervals) assuming a "true value" of $\mathcal{I}(\boldsymbol{x}_\mathcal{A}{:}\boldsymbol{x}_\mathcal{B})$. In this paper, we instead take a non-parametric Bayesian approach, where the mutual information itself is a derived random variable; a similar interpretation was recently developed in independent work [16]. The randomness of mutual information arises from treating the data generating process and parameters $\mathcal{G}$ as a random variable, whose prior distribution we denote π. Composing $\mathcal{G}$ with the function $h : \hat{\mathcal{G}} \mapsto \mathcal{I}_{\hat{\mathcal{G}}}(\boldsymbol{x}_\mathcal{A}{:}\boldsymbol{x}_\mathcal{B})$ induces the derived random variable $h(\mathcal{G}) \equiv \mathcal{I}_\mathcal{G}(\boldsymbol{x}_\mathcal{A}{:}\boldsymbol{x}_\mathcal{B})$. The distribution of the MI can thus be expressed as an expectation under distribution π:

$$\begin{aligned}\mathbb{P}\left[\mathcal{I}_\mathcal{G}(\boldsymbol{x}_\mathcal{A}{:}\boldsymbol{x}_\mathcal{B}) \in S\right] &= \int \mathbb{I}\left[\mathcal{I}_{\hat{\mathcal{G}}}(\boldsymbol{x}_\mathcal{A}{:}\boldsymbol{x}_\mathcal{B}) \in S\right] \pi(d\hat{\mathcal{G}}) \\ &= \mathop{\mathbb{E}}_{\hat{\mathcal{G}} \sim \pi}\left[\mathbb{I}\left[\mathcal{I}_{\hat{\mathcal{G}}}(\boldsymbol{x}_\mathcal{A}{:}\boldsymbol{x}_\mathcal{B}) \in S\right]\right]. \quad (3)\end{aligned}$$

Given a dataset $\mathcal{D}$, we define the posterior distribution of the mutual information, $\mathbb{P}\left[\mathcal{I}_\mathcal{G}(\boldsymbol{x}_\mathcal{A}{:}\boldsymbol{x}_\mathcal{B}) \in S|\mathcal{D}\right]$ as the expectation in Eq (3) under the posterior $\pi(\cdot|\mathcal{D})$. We define the distribution over conditional mutual information $\mathbb{P}\left[\mathcal{I}_\mathcal{G}(\boldsymbol{x}_\mathcal{A}{:}\boldsymbol{x}_\mathcal{B}|\hat{\boldsymbol{x}}_\mathcal{C}) \in S\right]$ analogously to Eq (3), substituting the CMI (2) inside the expectation.

2.1 Estimating CMI with generative population models

Monte Carlo estimates of CMI can be formed for models expressed as *generative population models* [18, 28], a probabilistic programming formalism for characterizing the data generating process of an infinite array of realizations of random vector $\boldsymbol{x} = (x_1, x_2, \dots, x_D)$. Listing 1 summarizes elements of the GPM interface.

Listing 1 GPM interface for simulating from and assessing the density of conditional and marginal distributions of a random vector $\boldsymbol{x}$.

```
Simulate($\mathcal{G}$, query: $\mathcal{Q}=\{q_j\}$, condition: $\hat{\boldsymbol{x}}_{\mathcal{E}}=\{\hat{x}_{e_j}\}$)
  Return a sample $\mathbf{s} \sim p_{\mathcal{G}}(\boldsymbol{x}_{\mathcal{Q}}|\hat{\boldsymbol{x}}_{\mathcal{E}}, \mathcal{D})$.

LogPdf($\mathcal{G}$, query: $\hat{\boldsymbol{x}}_{\mathcal{Q}}=\{\hat{x}_{q_j}\}$, condition: $\hat{\boldsymbol{x}}_{\mathcal{E}}=\{\hat{x}_{e_j}\}$)
  Return the joint log density $p_{\mathcal{G}}(\hat{\boldsymbol{x}}_{\mathcal{Q}}|\hat{\boldsymbol{x}}_{\mathcal{E}}, \mathcal{D})$
```

These two interface procedures can be combined to derive a simple Monte Carlo estimator for the CMI (2), shown in Algorithm 2a.

Algorithm 2a Gpm-Cmi

```
Require: GPM $\mathcal{G}$; query $\mathcal{A}$, $\mathcal{B}$; condition $\hat{\boldsymbol{x}}_{\mathcal{C}}$; accuracy $T$
Ensure: Monte Carlo estimate of $\mathcal{I}_{\mathcal{G}}\left(\boldsymbol{x}_{\mathcal{A}}{:}\boldsymbol{x}_{\mathcal{B}}|\boldsymbol{x}_{\mathcal{C}}{=}\hat{\boldsymbol{x}}_{\mathcal{C}}\right)$
for $t = 1, \dots, T$ do
    $(\hat{\boldsymbol{x}}_{\mathcal{A}}, \hat{\boldsymbol{x}}_{\mathcal{B}}) \leftarrow$ Simulate($\mathcal{G}, \mathcal{A} \cup \mathcal{B}, \hat{\boldsymbol{x}}_{\mathcal{C}}$)
    $m^t_{\mathcal{A}\cup\mathcal{B}} \leftarrow$ LogPdf($\mathcal{G}, \hat{\boldsymbol{x}}_{\mathcal{A}\cup\mathcal{B}}, \hat{\boldsymbol{x}}_{\mathcal{C}}$)
    $m^t_{\mathcal{A}} \leftarrow$ LogPdf($\hat{\boldsymbol{x}}_{\mathcal{A}}, \hat{\boldsymbol{x}}_{\mathcal{C}}$)
    $m^t_{\mathcal{B}} \leftarrow$ LogPdf($\hat{\boldsymbol{x}}_{\mathcal{B}}, \hat{\boldsymbol{x}}_{\mathcal{C}}$)
return $\frac{1}{T}\sum_{t=1}^{T}\left(m^t_{\mathcal{A}\cup\mathcal{B}} - (m^t_{\mathcal{A}} + m^t_{\mathcal{B}})\right)$
```

While Gpm-Cmi is an unbiased and consistent estimator applicable to any probabilistic model implemented as a GPM, its quality in detecting dependencies is tied to the ability of $\mathcal{G}$ to capture patterns from the dataset $\mathcal{D}$; this paper uses baseline non-parametric GPMs built using CrossCat (Section 3).

2.2 Extracting conditional independence relationships from CMI estimates

An estimator for the CMI can be used to discover several forms of independence relations of interest.

Marginal Independence It is straightforward to see that $(\boldsymbol{x}_{\mathcal{A}} \perp\!\!\!\perp_{\mathcal{G}} \boldsymbol{x}_{\mathcal{B}})$ if and only if $\mathcal{I}_{\mathcal{G}}(\boldsymbol{x}_{\mathcal{A}}{:}\boldsymbol{x}_{\mathcal{B}}) = 0$.

Context-Specific Independence If the event $\{\boldsymbol{x}_{\mathcal{C}}{=}\hat{\boldsymbol{x}}_{\mathcal{C}}\}$ decouples $\boldsymbol{x}_{\mathcal{A}}$ and $\boldsymbol{x}_{\mathcal{B}}$, then they are said to be independent "in the context" of $\hat{\boldsymbol{x}}_{\mathcal{C}}$, denoted $(\boldsymbol{x}_{\mathcal{A}} \perp\!\!\!\perp_{\mathcal{G}} \boldsymbol{x}_{\mathcal{B}}|\,\{\boldsymbol{x}_{\mathcal{C}}{=}\hat{\boldsymbol{x}}_{\mathcal{C}}\})$ [3]. This condition is equivalent to the CMI from (2) equaling zero. Thus by estimating CMI, we are able to detect finer-grained independencies than can be detected by analyzing the graph structure of a learned Bayesian network [30].

Conditional Independence If context-specific independence holds for all possible observation sets $\{\boldsymbol{x}_{\mathcal{C}}{=}\hat{\boldsymbol{x}}_{\mathcal{C}}\}$, then $\boldsymbol{x}_{\mathcal{A}}$ and $\boldsymbol{x}_{\mathcal{B}}$ are *conditionally independent* given $\boldsymbol{x}_{\mathcal{C}}$, denoted $(\boldsymbol{x}_{\mathcal{A}} \perp\!\!\!\perp_{\mathcal{G}} \boldsymbol{x}_{\mathcal{B}}|\boldsymbol{x}_{\mathcal{C}})$. By the non-negativity of CMI, conditional independence implies the CMI of $\boldsymbol{x}_{\mathcal{A}}$ and $\boldsymbol{x}_{\mathcal{B}}$, marginalizing out $\boldsymbol{x}_{\mathcal{C}}$, is zero:

$$\mathcal{I}_{\mathcal{G}}(\boldsymbol{x}_{\mathcal{A}}{:}\boldsymbol{x}_{\mathcal{B}}|\boldsymbol{x}_{\mathcal{C}}) = \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}_{\mathcal{C}}}\left[\mathcal{I}_{\mathcal{G}}(\boldsymbol{x}_{\mathcal{A}}{:}\boldsymbol{x}_{\mathcal{B}}|\boldsymbol{x}_{\mathcal{C}} = \hat{\boldsymbol{x}}_{\mathcal{C}})\right] = 0. \quad (4)$$

Figure 2 illustrates different CMI queries which are used to discover these three types of dependencies in various data generators; Figure 3 shows CMI queries expressed in the Bayesian Query Language.

3 Building generative population models for CMI estimation with non-parametric Bayes

Our approach to estimating the CMI requires a prior π and model class $\mathcal{G}$ which is flexible enough to emulate an arbitrary joint distribution over $\boldsymbol{x}$, and tractable enough to implement Algorithm 2a for its arbitrary sub-vectors. We begin with a Dirichlet process mixture model (DPMM) [12]. Letting L_d denote the likelihood for variable d, V_d a prior over the parameters of L_d, and $\boldsymbol{\lambda}_d$ the hyperparameters of V_d, the generative process for N observations $\mathcal{D}=\left\{\boldsymbol{x}_{[i,1:D]} : 1 \le i \le N\right\}$ is:

<u>DPMM-Prior</u>

$$\begin{aligned}
&\alpha \sim \text{Gamma}(1,1) \\
&\boldsymbol{z} = (z_1, \dots, z_N) \sim \text{CRP}(\cdot|\alpha) \\
&\boldsymbol{\phi}_{[d,k]} \sim V_d(\cdot|\boldsymbol{\lambda}_d) && d \in [D], k \in \text{Unique}(\boldsymbol{z}) \\
&x_{[i,d]} \sim L_d(\cdot|\boldsymbol{\phi}_{[d,z_i]}) && i \in [N], d \in [D]
\end{aligned}$$

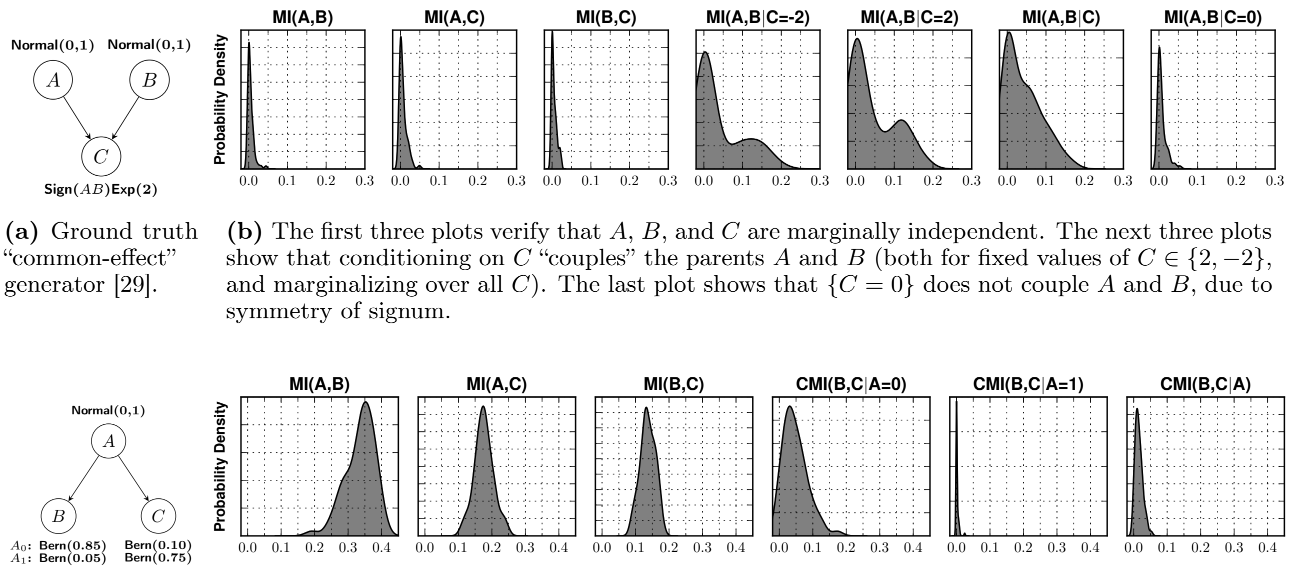
We refer to [7, 14] for algorithms for posterior inference, and assume we have a posterior sample $\hat{\mathcal{G}} = (\alpha, \boldsymbol{z}_{[1:N]}, \{\boldsymbol{\phi}_d\})$ of all parameters in the DPMM. To compute the CMI of an arbitrary query pattern $\mathcal{I}_{\hat{\mathcal{G}}}(\boldsymbol{x}_{\mathcal{A}}{:}\boldsymbol{x}_{\mathcal{B}}|\boldsymbol{x}_{\mathcal{C}}{=}\hat{\boldsymbol{x}}_{\mathcal{C}})$ using Algorithm 2a, we need implementations of Simulate and LogPdf for $\hat{\mathcal{G}}$. These two procedures are summarized in Algorithms 3a, 3b.

Algorithm 3a DPMM-Simulate

```
Require: DPMM $\mathcal{G}$; target $\mathcal{A}$; condition $\hat{\boldsymbol{x}}_{\mathcal{C}}$
Ensure: joint sample $\hat{\boldsymbol{x}}_{\mathcal{A}} \sim p_{\mathcal{G}}(\cdot|\hat{\boldsymbol{x}}_{\mathcal{C}})$
$(l_i)_{i=1}^{K+1} \leftarrow$ DPMM-Cluster-Posterior($\mathcal{G}, \hat{\boldsymbol{x}}_{\mathcal{C}}$)
$z_{N+1} \sim$ Categorical($l_1, \dots, l_{K+1}$)
for $a \in \mathcal{A}$ do
    $\hat{x}_a \sim L_a(\cdot|\boldsymbol{\phi}_{[a,z_{N+1}]})$
return $\hat{\boldsymbol{x}}_{\mathcal{A}}$
```

Algorithm 3b DPMM-LogPdf

```
Require: DPMM $\mathcal{G}$; target $\hat{\boldsymbol{x}}_{\mathcal{A}}$; condition $\hat{\boldsymbol{x}}_{\mathcal{C}}$
Ensure: log density $p_{\mathcal{G}}(\hat{\boldsymbol{x}}_{\mathcal{A}}|\hat{\boldsymbol{x}}_{\mathcal{C}})$
$(l_i)_{i=1}^{K+1} \leftarrow$ DPMM-Cluster-Posterior($\mathcal{G}, \hat{\boldsymbol{x}}_{\mathcal{C}}$)
for $k = 1, \dots, K+1$ do
    $t_k \leftarrow \prod_{a\in\mathcal{A}} L_a(\hat{x}_a|\boldsymbol{\phi}_{[a,k]})$
return log $\left(\sum_{k=1}^{K+1}(t_k l_k)\right)$
```



(a) Ground truth "common-effect" generator [29].

(b) The first three plots verify that A, B, and C are marginally independent. The next three plots show that conditioning on C "couples" the parents A and B (both for fixed values of $C \in \{2, -2\}$, and marginalizing over all C). The last plot shows that $\{C = 0\}$ does not couple A and B, due to symmetry of signum.

(c) Ground truth "common-cause" generator.

(d) The first three plots verify that A, B, and C are marginally dependent. The next three plots show that conditioning on A "decouples" the children B and C; the decoupling is weaker for $\{A = 0\}$, because it is 3.4 nats less likely that $\{A = 1\}$. The final plot shows the weighted CMI under these two possibilities.

Figure 2: Posterior distributions of CMI under the DPMM posterior, given 100 data points from canonical Bayes net structures. Distributions peaked at 0 indicate high probability of (conditional) independence. In both cases, the posterior CMI distributions correctly detect the marginal, conditional, and context-specific independences in the "ground truth" Bayes nets, despite the fact that both "common-cause" and "common-effect" structures are not in the (structural) hypothesis space of the DPMM prior.

Algorithm 3c DPMM-Cluster-Posterior

```
Require: DPMM $\mathcal{G}$; condition $\hat{\boldsymbol{x}}_\mathcal{C}$;
Ensure: $\{p_\mathcal{G}(z_{N+1} = k) : 1 \le k \le \max(\boldsymbol{z}_{1:N}) + 1\}$
$K \leftarrow \max(\boldsymbol{z}_{1:N})$
for $k = 1, \dots, K + 1$ do
    $n_k \leftarrow \begin{cases} |\{\boldsymbol{x}_i \in \mathcal{D} : z_i = k\}| & \text{if } k \le K \\ \alpha & \text{if } k = K + 1 \end{cases}$
    $l_k \leftarrow \left(\prod_{c \in \mathcal{C}} L_c(\hat{x}_c | \boldsymbol{\phi}_{[c,k]})\right) n_k$
return $(l_1, \dots, l_{K+1}) / \sum_{k=1}^{K+1} (l_k)$
```

The subroutine DPMM-Cluster-Posterior is used for sampling (in DPMM-Simulate) and marginalizing over (in DPMM-LogPdf) the non-parametric mixture components. Moreover, if L_d and V_d form a conjugate likelihood-prior pair, then invocations of $L_d(\hat{x}_d | \boldsymbol{\phi}_{[d,k]})$ in Algorithms 3a:4 and 3b:3 can be Rao-Blackwellized by conditioning on the sufficient statistics of data in cluster k, thus marginalizing out $\boldsymbol{\phi}_{[d,k]}$ [26]. This optimization is important in practice, since analytical marginalization can be obtained in closed-form for several likelihoods in the exponential family [9]. Finally, to approximate the posterior distribution over CMI in (2), it suffices to aggregate DPMM-Cmi from a set of posterior samples $\left\{\hat{\mathcal{G}}_1, \dots, \hat{\mathcal{G}}_H\right\} \sim^{\text{iid}} \pi(\cdot|\mathcal{D})$. Figure 2 shows posterior CMI distributions from the DPMM successfully recovering the marginal and conditional independencies in two canonical Bayesian networks.

3.1 Inducing sparse dependencies using the CrossCat prior

The multivariate DPMM makes the restrictive assumption that all variables $\boldsymbol{x} = (x_1, \dots, x_D)$ are (structurally) marginally dependent, where their joint distribution fully factorizes conditioned on the mixture assignment z. In high-dimensional datasets, imposing full structural dependence among all variables is too conservative. Moreover, while the Monte Carlo error of Algorithm 2a does not scale with the dimensionality D, its runtime scales linearly for the DPMM, and so estimating the CMI is likely to be prohibitively expensive. We relax these constraints by using CrossCat [19], a structure learning prior which induces sparsity over the dependencies between the variables of $\boldsymbol{x}$. In particular, CrossCat posits a factorization of $\boldsymbol{x}$ according to a *variable partition* $\boldsymbol{\gamma} = \left\{\mathcal{V}_1, \dots, \mathcal{V}_{|\boldsymbol{\gamma}|}\right\}$, where $\mathcal{V}_i \subseteq [D]$. For $i \neq j$, all variables in block $\mathcal{V}_i$ are mutually (marginally and conditionally) independent of all variables in $\mathcal{V}_j$. The factorization of $\boldsymbol{x}$ given the

English Summary of CMI Query	CMI Query in Bayesian Query Language	Inference Algorithm Invoked by Query Interpreter

Simulate from the posterior distribution of the mutual information of (x_1, x_2) with x_3, given $x_4 = 14$.	`SIMULATE`
	`  MUTUAL INFORMATION OF`
	`    (x1, x2) WITH (x3)`
	`  GIVEN (x4 = 14)`
	`FROM MODELS OF population`
		for $\mathcal{G}_k \in \mathcal{M}$ **do**
		  $\mathcal{I}_{\mathcal{G}_k} \leftarrow \textsc{Gpm-Cmi}(\mathcal{G}_k, \{x_1, x_2\}, \{x_3\}, \{(x_4, 14)\})$
		return $(\mathcal{I}_{\mathcal{G}_1}, \dots, \mathcal{I}_{\mathcal{G}_{\lvert\mathcal{M}\rvert}})$

Estimate the probability that the mutual information of (x_1, x_2) with x_3, given $x_4 = 14$ and marginalizing over x_5, is less than 0.1 nats.	`ESTIMATE PROBABILITY OF`
	`  MUTUAL INFORMATION OF`
	`    (x1, x2) WITH (x3)`
	`    GIVEN (x4 = 14, x5)`
	`  < 0.1`
	`BY population`
		for $\mathcal{G}_k \in \mathcal{M}$ **do**
		  for $t = 1, \dots, T$ **do**
		    $\hat{x}_5^t \leftarrow \textsc{Simulate}(\mathcal{G}_k, x_5, \{(x_4, 14)\})$
		    $\mathcal{I}^t_{\mathcal{G}_k} \leftarrow \textsc{Gpm-Cmi}($
		      $\mathcal{G}_k, \{x_1, x_2\}, \{x_3\}, \{(x_4, 14), (x_5, \hat{x}_5^t)\})$
		  $\mathcal{I}_{\mathcal{G}_k} \leftarrow \frac{1}{T}\sum_t \left(\mathcal{I}^t_{\mathcal{G}_k}\right)$
		return $\frac{1}{\lvert\mathcal{M}\rvert}\sum_j \left(\mathbb{I}\left[\mathcal{I}_{\mathcal{G}_k} < 0.1\right]\right)$

Synthesize a hypothetical dataset with 100 records, including only those variables which are probably independent of x_2.	`SIMULATE (`
	`  SELECT * FROM`
	`  VARIABLES OF population`
	`  WHERE PROBABILITY OF`
	`    MUTUAL INFORMATION`
	`    WITH x2 < 0.1`
	`  > 0.9)`
	`FROM population`
	`LIMIT 100;`
		$\mathcal{S} \leftarrow \varnothing$
		for $x_i \in (x_1, \dots, x_D)$ **do**
		  for $\mathcal{G}_k \in \mathcal{M}$ **do**
		    $\mathcal{I}_{\mathcal{G}_k} \leftarrow \textsc{Gpm-Cmi}(\mathcal{G}_k, x_i, x_2, \varnothing)$
		  $p_i \leftarrow \frac{1}{\lvert\mathcal{M}\rvert}\sum_k \mathbb{I}\left[\mathcal{I}_{\mathcal{G}_k} < 0.1\right]$
		  if $p_i > 0.9$ **then**
		    $\mathcal{S} \leftarrow \mathcal{S} \cup \{x_i\}$
		for $t = 1 \dots, 100$ **do**
		  $s_t \leftarrow \textsc{Simulate}(\mathcal{M}, \mathcal{S}, \varnothing)$
		return $(s_1, \dots, s_{100})$

Figure 3: End-user CMI queries in the Bayesian Query Language for three data analysis tasks; (top) evaluating the strength of predictive relationships; (middle) specifying the amount of evidence required for a "predictively" significant relationship; (bottom) synthesizing a hypothetical population, censoring probably sensitive variables.

variable partition $\boldsymbol{\gamma}$ is therefore given by:

$$p_{\mathcal{G}}(\boldsymbol{x}|\mathcal{D}) = \prod_{\mathcal{V} \in \boldsymbol{\gamma}} p_{\mathcal{G}_\mathcal{V}}(\boldsymbol{x}_\mathcal{V}|\mathcal{D}_\mathcal{V}). \tag{5}$$

Within block $\mathcal{V}$, the variables $\boldsymbol{x}_\mathcal{V} = \{x_d : d \in \mathcal{V}\}$ are distributed according to a multivariate DPMM; subscripts with $\mathcal{V}$ (such as $\mathcal{G}_\mathcal{V}$) now index a set of block-specific DPMM parameters. The joint predictive density $p_{\mathcal{G}_\mathcal{V}}$ is given by Algorithm 3b:

$$p_{\mathcal{G}_\mathcal{V}}(\boldsymbol{x}_\mathcal{V}|\mathcal{D}) = \sum_{k=1}^{K_V+1} \left(\frac{n_{[\mathcal{V},k]} \prod_{d \in \mathcal{V}} p_{\mathcal{G}_\mathcal{V}}(x_d|\boldsymbol{\phi}_{[d,k]})}{\sum_{k'} n_{[\mathcal{V},k']}} \right). \tag{6}$$

The CrossCat generative process for N observations $\mathcal{D} = \left\{\boldsymbol{x}_{[i,1:D]} : 1 \le i \le N\right\}$ is summarized below.

```
$\underline{\textsc{CrossCat-Prior}}$
$\alpha' \sim \textsc{Gamma}(1, 1)$
$\boldsymbol{v} = (v_1, \dots, v_D) \sim \text{CRP}(\cdot|\alpha')$
$\mathcal{V}_k \leftarrow \{i \in [D] : v_i = k\}$            $k \in \textsc{Unique}(\boldsymbol{v})$
$\left\{x_{[i,\mathcal{V}_k]}\right\}_{i=1}^{N} \sim \text{DPMM-}\textsc{Prior}$     $k \in \textsc{Unique}(\boldsymbol{v})$
```

We refer to [19, 23] for algorithms for posterior inference in CrossCat, and assume we have a set of approximate samples $\left\{\hat{\mathcal{G}}_i : 1 \le i \le H\right\}$ of all latent CrossCat parameters from the posterior $\pi(\cdot|\mathcal{D})$.

3.2 Optimizing a CMI query

The following lemma shows how CrossCat induces sparsity for a multivariate CMI query.

Lemma 1. *Let $\mathcal{G}$ be a posterior sample from CrossCat, whose full joint distribution is given by* (5) *and* (6)*. Then, for all $\mathcal{A}, \mathcal{B}, \mathcal{C} \subseteq [D]$,*

$$\mathcal{I}_{\mathcal{G}}\left(\boldsymbol{x}_\mathcal{A} : \boldsymbol{x}_\mathcal{B} \mid \hat{\boldsymbol{x}}_\mathcal{C}\right) = \sum_{\mathcal{V} \in \boldsymbol{\gamma}} \mathcal{I}_{\mathcal{G}_\mathcal{V}}(\boldsymbol{x}_{\mathcal{A}\cap\mathcal{V}} : \boldsymbol{x}_{\mathcal{B}\cap\mathcal{V}} \mid \hat{\boldsymbol{x}}_{\mathcal{C}\cap\mathcal{V}}),$$

where $\mathcal{I}_{\mathcal{G}_\mathcal{V}}(\boldsymbol{x}_{\mathcal{A}\cap\mathcal{V}} : \varnothing | \hat{\boldsymbol{x}}_{\mathcal{C}\cap\mathcal{V}}) \equiv 0.$

Proof. Refer to Appendix A.

An immediate consequence of Lemma 1 is that structure discovery in CrossCat allows us to optimize Monte Carlo estimation of $\mathcal{I}_\mathcal{G}(\boldsymbol{x}_\mathcal{A} : \boldsymbol{x}_\mathcal{B} | \boldsymbol{x}_\mathcal{C} = \hat{\boldsymbol{x}}_\mathcal{C})$ by ignoring all target and condition variables which are not in the same block $\mathcal{V}$, as shown in Algorithm 4a and Figure 4.

Algorithm 4a CrossCat-Cmi

```
Require: CrossCat $\mathcal{G}$; query $\mathcal{A}$, $\mathcal{B}$; condition $\hat{\boldsymbol{x}}_\mathcal{C}$; acc. $T$
Ensure: Monte Carlo estimate of $\mathcal{I}_\mathcal{G}(\boldsymbol{x}_\mathcal{A}{:}\boldsymbol{x}_\mathcal{B}|\boldsymbol{x}_\mathcal{C}{=}\hat{\boldsymbol{x}}_\mathcal{C})$
for $\mathcal{V} \in \boldsymbol{\gamma}$ do
  if $\mathcal{A} \cap \mathcal{V}$ AND $\mathcal{B} \cap \mathcal{V}$ then
    $i_\mathcal{V} \leftarrow \textsc{Gpm-Cmi}(\mathcal{G}_\mathcal{V}, \mathcal{A} \cap \mathcal{V}, \mathcal{B} \cap \mathcal{V}, \hat{\boldsymbol{x}}_{\mathcal{C}\cap\mathcal{V}}, T)$
  else
    $i_\mathcal{V} \leftarrow 0$
return $\sum_{\mathcal{V} \in \boldsymbol{\gamma}} i_\mathcal{V}$
```

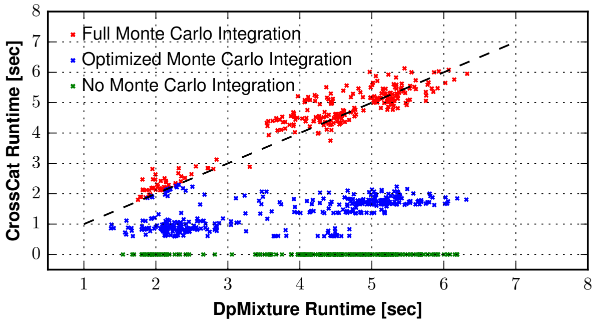


Figure 4: Comparing the runtime of CrossCat-Cmi (Alg 4a) and Gpm-Cmi (Alg 2a) (using the DPMM), on 1000 randomly generated CMI queries from an 8-dimensional dataset. The dashed curve shows the 45-degree line. The green dots at 0 correspond to CrossCat detecting structural independence between query variables, bypassing Monte Carlo estimation completely. The blue dots (below diagonal) correspond to CrossCat optimizing the Monte Carlo estimator by ignoring constraint variables which are structurally independent of the target variables. The red dots (along diagonal) correspond to CrossCat learning no structural independences, requiring full Monte Carlo estimation and resulting in comparable runtime to DPMM. These three cases correspond to the three posterior CrossCat structures illustrated in Figure 1, when the targets variables are X and Y conditioned on W.

3.3 Upper bounding the pairwise dependence probability

In exploratory data analysis, we are often interested in detecting pairwise predictive relationships between variables (x_i, x_j). Using the formalism from Eq (3), we can compute the probability that their MI is nonzero: $\mathbb{P}\left[\mathcal{I}_{\mathcal{G}}(x_i : x_j) > 0\right]$. This quantity can be upper-bounded by the posterior probability that x_i and x_j have the same assignments v_i and v_j in the CrossCat variable partition $\boldsymbol{\gamma}$:

$$\begin{aligned}
&\mathbb{P}\left[\mathcal{I}_{\mathcal{G}}(x_i : x_j) > 0\right] \\
&= \mathbb{P}\left[\mathcal{I}_{\mathcal{G}}(x_i : x_j) > 0 \mid \{\mathcal{G} : v_i = v_j\}\right] \mathbb{P}\left[\{\mathcal{G} : v_i = v_j\}\right] \\
&\quad + \mathbb{P}\left[\mathcal{I}_{\mathcal{G}}(x_i : x_j) > 0 \mid \{\mathcal{G} : v_i \neq v_j\}\right] \mathbb{P}\left[\{\mathcal{G} : v_i \neq v_j\}\right] \\
&= \mathbb{P}\left[\mathcal{I}_{\mathcal{G}}(x_i : x_j) > 0 \mid \{\mathcal{G} : v_i = v_j\}\right] \mathbb{P}\left[\{\mathcal{G} : v_i = v_j\}\right] \\
&< \mathbb{P}\left[\{\mathcal{G} : v_i = v_j\}\right] \approx \frac{1}{H} \sum_{h=1}^{H} \mathbb{I}[\hat{\mathcal{G}}_h : \hat{v}_{[h,i]} = \hat{v}_{[h,j]}], \quad (7)
\end{aligned}$$

where Lemma 1 has been used to set the addend in line 3 to zero. Also note that the summand in (7) can be computed in $O(1)$ for CrossCat sample $\hat{\mathcal{G}}_h$. When dependencies among the D variables are sparse such that many pairs (x_i, x_j) have MI upper bounded by 0, the number of invocations of Algorithm 4a required to compute pairwise MI values is $\ll O(D^2)$. A comparison of upper bounding MI versus exact MI estimation with Monte Carlo is shown in Figure 5.

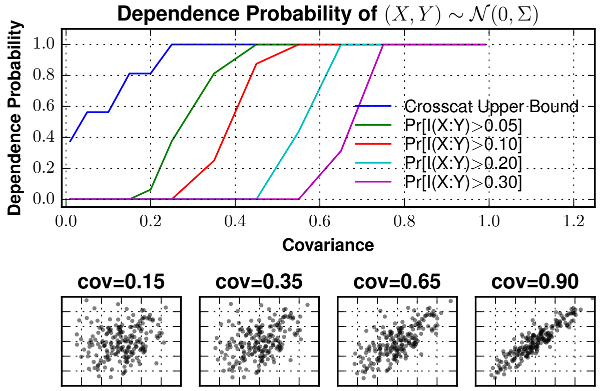
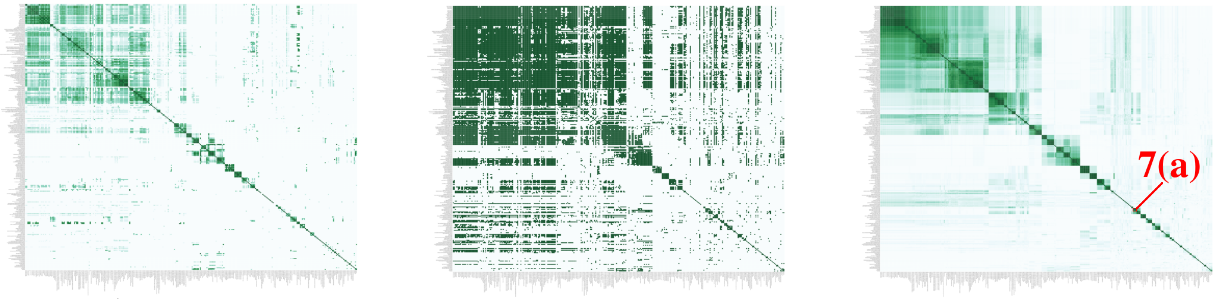


Figure 5: Posterior probability that dimensions of a bivariate Gaussian are dependent, vs the covariance (top). The CrossCat upper bound (7) is useful for detecting the existence of a predictive relationship; the posterior distribution of MI can determine whether the strength of the relationship is "predictively significant" based on various tolerance levels (0.05, 0.10, 0.20, and 0.30 nats).

4 Applications to macroeconomic indicators of global poverty, education, and health

This section illustrates the efficacy of the proposed approach on a sparse database from an ongoing collaboration with the Bill & Melinda Gates Foundation.[1] The Gapminder data set is an extensive longitudinal dataset of ∼320 global developmental indicators for 200 countries spanning over 5 centuries [27]. These include variables from a broad set of categories such as education, health, trade, poverty, population growth, and mortality rates. We experiment with a cross-sectional slice of the data from 2002. Figure 6a shows the pairwise R^2 correlation values between all variables; each row and column in the heatmap is an indicator in the dataset, and the color of a cell is the raw value of R^2 (between 0 and 1). Figure 6b shows pairwise binary hypothesis tests of independence using HSIC [13], which detects a dense set of dependencies including many spurious relationships (Appendix B). For both methods, statistically insignificant relationships ($\alpha = 0.05$ with Bonferroni correction for multiple testing) are shown as 0. Figure 6c shows an upper bound on the pairwise probability that the MI of two variables exceeds zero (also a value between 0 and 1). These entries are estimated using Eq (7) (bypassing Monte Carlo estimation) using $H{=}100$ samples of CrossCat. Note that the metric $\mathbb{P}[\mathcal{I}_{\mathcal{G}}(x_i{:}x_j) > 0]$ in Figure 6c only indicates the *existence* of a predictive relationship between x_i and x_j; it does not quantify either the strength or directionality of the relationship.

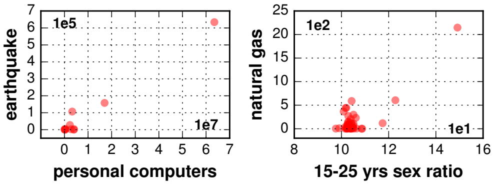
[1] A further application, to a real-world dataset of mathematics exam scores, is shown in Appendix C.



(a) R^2 Correlation Value **(b)** HSIC [13] Independence Test **(c)** $\mathbb{P}[\mathcal{I}_\mathcal{G}(x_i{:}x_j) > 0]$, Eq (7)

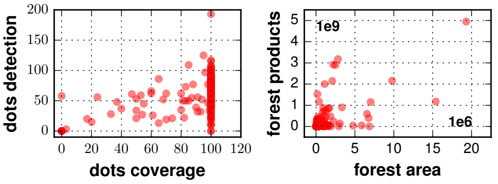
(d) Pairwise heatmaps of all 320 variables in the Gapminder dataset using three dependency detection techniques. Darker cells indicate a detected dependence between the two variables.

variable A	variable B	$\mathbb{P}[\mathcal{I}_\mathcal{G} > 0]$	R^2 $(p \ll 10^{-6})$
personal computers	earthquake affected	0.015625	0.974445
road traffic total deaths	people living w/ hiv	0.265625	0.858260
natural gas reserves	15-25 yrs sex ratio	0.031250	0.951471
forest products per ha	earthquake killed	0.046875	0.936342
flood affected	population 40-59 yrs	0.140625	0.882729



(e) Spurious relationships which are correlated under R^2, but probably independent according to posterior MI.

variable A	variable B	$\mathbb{P}[\mathcal{I}_\mathcal{G} > 0]$	R^2 $(p \ll 10^{-6})$
inflation	trade balance (% gdp)	0.859375	0.114470
DOTS detection rate	DOTS coverage	0.812500	0.218241
forest area (sq. km)	forest products (usd)	0.828125	0.206145
long term unemp. rate	total 15-24 unemp.	0.968750	NaN
male family workers	female self-employed	0.921875	NaN



(f) Common-sense relationships probably dependent according to posterior MI, but weakly correlated under R^2.

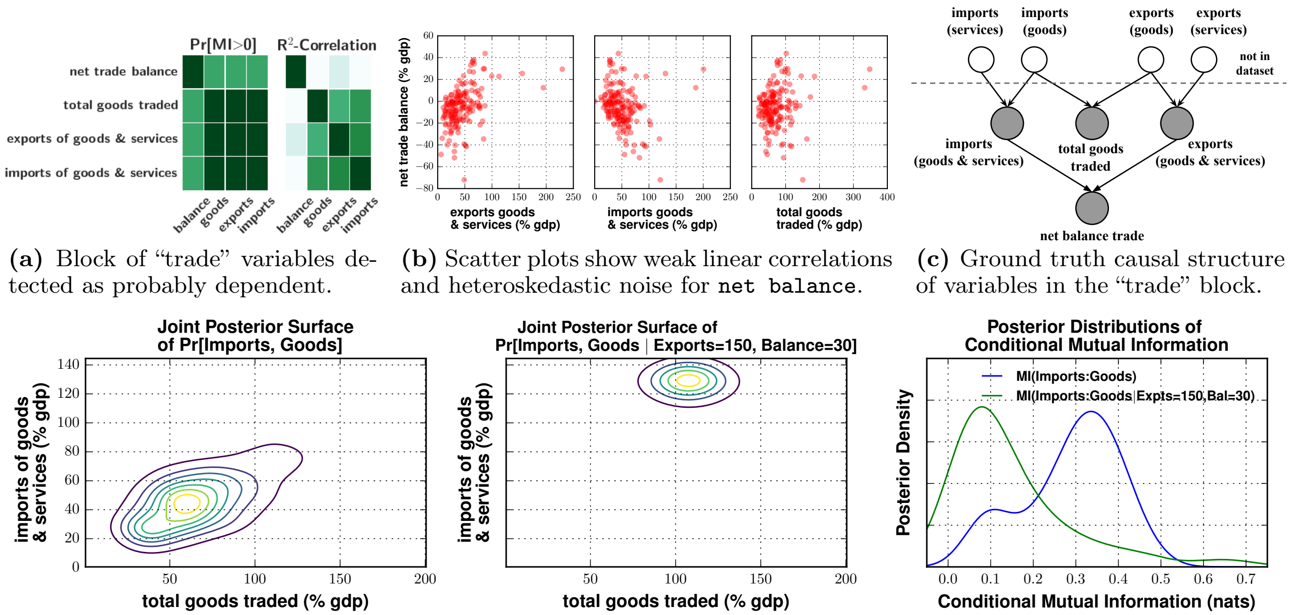
Figure 6: Comparing dependences between variables in the Gapminder dataset, as detected by R^2, HSIC (with Bonferroni correction for multiple testing), and posterior distribution over mutual information in CrossCat.

It is instructive to compare the dependencies detected by R^2 and CrossCat-Cmi. Table 6e shows pairs of variables that are spuriously reported as dependent according to correlation; scatter plots reveal they are either (i) are sparsely observed or (ii) exhibit high correlation due to large outliers. Table 6f shows common-sense relationships between pairs of variables that CrossCat-CMI detects but R^2 does not; scatter plots reveal they are either (i) non-linearly related, (ii) roughly linear with heteroskedastic noise, or (iii) pairwise independent but dependent given a third variable. Recall that CrossCat is a product of DPMMs; practically meaningful conditions for weak and strong consistency of Dirichlet location-scale mixtures have been established by [11, 33]. This supports the intuition that CrossCat can detect a broad class of predictive relationships that simpler parametric models miss.

Figure 7 focuses on a group of four "trade"-related variables in the Gapminder dataset detected as probably dependent: "net trade balance", "total goods traded", "exports of goods and services", and "imports of goods and services". R^2 fails to detect a statistically significant dependence between "net trade balance" and the other variables, due to weak linear correlations and heteroskedastic noise as shown in the scatter plots (Figure 7b). From economics, these four variables are causally related by the graphical model in Figure 7c, where the value of a node is a noisy addition or subtraction of the values of its parents. Figure 7d illustrates that CrossCat recovers predictive relationships between these variables: conditioning on "exports"=150 and "balance"=30 (a low probability event according to the left subplot) centers the posterior predictive distribution of "imports" around 120, and decouples it from "total goods". The posterior CMI curves of "imports" and "total goods", with and without the conditions on "exports" and "balance", formalize this decoupling (right subplot of Figure 7d).

5 Related Work

There is broad acknowledgment that new techniques for dependency detection beyond linear correlation are required. Existing approaches for conditional independence testing include the use of kernel methods [1, 10, 35, 29], copula functions [2, 25, 17], and char-



(a) Block of "trade" variables detected as probably dependent.

(b) Scatter plots show weak linear correlations and heteroskedastic noise for `net balance`.

(c) Ground truth causal structure of variables in the "trade" block.

(d) Left plot shows the joint posterior density of "imports" and "goods", where the marginal coupling is due to their common parent in (c). Center plot shows the same distribution conditioned on "exports"=150 and "balance"=30; "imports" now centers around its noiseless value of 120, and is decoupled from "goods". Right plot shows the CMI for these distributions.

Figure 7: CMI discovers existence and confirms strength of predictive relationships between "trade" variables.

acteristic functions [32], many of which capture non-linear and multivariate predictive relationships. Unlike these methods, however, our approach represents dependence in terms of conditional mutual information and is not embedded in a frequentist decision- theoretic framework. Our quantity of interest is a full posterior distribution over CMI, as opposed to a p-value to identify when the null hypothesis CMI=0 cannot be rejected. Dependence detection is much less studied in the Bayesian literature; [8] use a Polya tree prior to compute a Bayes Factor for the relative evidence of dependence versus independence. Their method is used only to quantify evidence for the existence, but not assess the strength, of a predictive relationship. The most similar approach to this work was proposed independently in recent work by [16], who compute a distribution over CMI by estimating the joint density using an encompassing non-parametric Bayesian prior. However, the differences are significant. First, the Monte Carlo estimator in [16] is based on resampling empirical data. However, real-world databases may be too sparse for resampling data to yield good estimates, especially for queries given unlikely constraints. Instead, we use a Monte Carlo estimator by simulating the predictive distribution. Second, the prior in [16] is a standard Dirichlet process mixture model, whereas this paper proposes a sparsity-inducing CrossCat prior, which permits optimized computations for upper bounds of posterior probabilities as well as simplifying CMI queries with multivariate conditions.

6 Discussion

This paper has shown it is possible to detect predictive relationships by integrating probabilistic programming, information theory, and non-parametric Bayes. Users specify a broad class of conditional mutual information queries using a simple SQL-like language, which are answered using a scalable pipeline based on approximate Bayesian inference. The underlying approach applies to arbitrary generative population models, including parametric models and other classes of probabilistic programs [28]; this work has focused on exploiting the sparsity of CrossCat model structures to improve scalability for exploratory analysis. With this foundation, one may extend the technique to domains such as causal structure learning. The CMI estimator can be used as a conditional-independence test in a structure discovery algorithm such as PC [31]. It is also possible to use learned CMI probabilities as part of a prior over directed acyclic graphs in the Bayesian setting. This paper has focused on detection and preliminary assessment of predictive relationships; confirmatory analysis and descriptive summarization are left for future work, and will require an assessment of the robustness of joint density estimation when random sampling assumptions are violated. Moreover, new algorithmic insights will be needed to scale the technique to efficiently detect pairwise dependencies in very high-dimensional databases with tens of thousands of variables.

Acknowledgements

This research was supported by DARPA (PPAML program, contract number FA8750-14-2-0004), IARPA (under research contract 2015-15061000003), the Office of Naval Research (under research contract N000141310333), the Army Research Office (under agreement number W911NF-13-1-0212), and gifts from Analog Devices and Google.

References

[1] Francis Bach and Michael Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[2] Taoufik Bouezmarni, Jeroen VK Rombouts, and Abderrahim Taamouti. Nonparametric copula-based test for conditional independence with applications to granger causality. *Journal of Business & Economic Statistics*, 30(2):275–287, 2012.

[3] Craig Boutilier, Nir Friedman, Moises Goldszmidt, and Daphne Koller. Context-specific independence in bayesian networks. In *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence*, pages 115–123. Morgan Kaufmann Publishers Inc., 1996.

[4] National Reseach Council. *Frontiers in massive data analysis.* The National Academies Press, 2013.

[5] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* Wiley Series in Telecommunications and Signal Processing. Wiley, 2012.

[6] David Edwards. *Introduction to graphical modelling.* Springer Texts in Statistics. Springer, 2012.

[7] Michael Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430): 577–588, 1995.

[8] Sarah Filippi and Chris Holmes. A bayesian nonparametric approach to testing for dependence between random variables. *Bayesian Analysis*, 2016. Advance publication.

[9] Daniel Fink. A compendium of conjugate priors. Technical report, Environmental Statistics Group, Department of Biology, Montana State University, 1997.

[10] Kenji Fukumizu, Arthur Gretton, Xiaohai Sun, and Bernhard Schölkopf. Kernel measures of conditional dependence. In *Proceedings of the Twentieth International Conference on Neural Information Processing Systems*, pages 489–496. Curran Associates Inc., 2007.

[11] Subhashis Ghosal, Jayanta Ghosh, and R.V. Ramamoorthi. Posterior consistency of dirichlet mixtures in density estimation. *The Annals of Statistics*, 27(1):143–158, 1999.

[12] Dilan Görür and Carl Edward Rasmussen. Dirichlet process gaussian mixture models: Choice of the base distribution. *Journal of Computer Science and Technology*, 25(4):653–664, 2010.

[13] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Proceedings of the Sixteenth International Conference Algorithmic Learning Theory*, pages 63–77. Springer, 2005.

[14] Sonia Jain and Radford M Neal. A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13(1):158–182, 2012.

[15] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69(6):066138, 2004.

[16] Tsuyoshi Kunihama and David B Dunson. Nonparametric bayes inference on conditional independence. *Biometrika*, 103(1):35–47, 2016.

[17] David Lopez-Paz, Philipp Hennig, and Bernhard Schölkopf. The randomized dependence coefficient. In *Proceedings of the Twenty-Sixth International Conference on Neural Information Processing Systems*, pages 1–9. Curran Associates Inc., 2013.

[18] Vikash Mansinghka, Richard Tibbetts, Jay Baxter, Pat Shafto, and Baxter Eaves. BayesDB: A probabilistic programming system for querying the probable implications of data. *CoRR*, abs/1512.05006, 2015.

[19] Vikash Mansinghka, Patrick Shafto, Eric Jonas, Cap Petschulat, Max Gasner, and Joshua B. Tenenbaum. CrossCat: A fully Bayesian nonparametric method for analyzing heterogeneous, high dimensional data. *Journal of Machine Learning Research*, 17(138):1–49, 2016.

[20] Kantilal Varichand Mardia, John T Kent, and John M Bibby. *Multivariate analysis.* Probability and Mathematical Statistics. Academic Press, 1980.

[21] Rudy Moddemeijer. On estimation of entropy and mutual information of continuous distributions. *Signal Processing*, 16(3):233–248, 1989.

[22] Young-Il Moon, Balaji Rajagopalan, and Upmanu Lall. Estimation of mutual information using kernel density estimators. *Physical Review E*, 52(3): 2318, 1995.

[23] Fritz Obermeyer, Jonathan Glidden, and Eric Jonas. Scaling nonparametric Bayesian inference via subsample-annealing. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 696–705. JMLR.org, 2014.

[24] Liam Paninski. Estimation of entropy and mutual information. *Neural Computation*, 15(6):1191–1253, 2003.

[25] Barnabás Póczos, Zoubin Ghahramani, and Jeff Schneider. Copula-based kernel dependency measures. *CoRR*, abs/1206.4682, 2012.

[26] Christian Robert and George Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer, 2005.

[27] Hans Rosling. Gapminder: Unveiling the beauty of statistics for a fact based world view. URL `https://www.gapminder.org/data/`.

[28] Feras Saad and Vikash Mansinghka. Probabilistic data analysis with probabilistic programming. *CoRR*, abs/1608.05347, 2016.

[29] Dino Sejdinovic, Arthur Gretton, and Wicher Bergsma. A kernel test for three-variable interactions. In *Proceedings of the Twenty-Sixth International Conference on Neural Information Processing Systems*, pages 1124–1132. Curran Associates Inc., 2013.

[30] Ross D Shachter. Bayes-ball: Rational pastime for determining irrelevance and requisite information in belief networks and influence diagrams. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 480–487. Morgan Kaufmann Publishers Inc., 1998.

[31] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. Adaptive Computation and Machine Learning. MIT Press, 2000.

[32] Liangjun Su and Halbert White. A consistent characteristic function-based test for conditional independence. *Journal of Econometrics*, 141(2): 807–834, 2007.

[33] Surya T Tokdar. Posterior consistency of dirichlet location-scale mixture of normals in density estimation and regression. *The Indian Journal of Statistics*, 68(1):90–110, 2006.

[34] Joe Whittaker. *Graphical models in applied multivariate statistics*. Wiley Series in Probability and Mathematical Statistics. Wiley, 1990.

[35] Kun Zhang, Jonas Peters, and Dominik Janzing. Kernel-based conditional independence test and application in causal discovery. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 804–813. AUAI Press, 2011.