Probabilistic Search for Structured Data via Probabilistic Programming and Nonparametric Bayes

Feras Saad, Leonardo Casarsa, and Vikash Mansinghka
Probabilistic Computing Project
Massachusetts Institute of Technology

Abstract

Databases are widespread, yet extracting relevant data can be difficult. Without substantial domain knowledge, multivariate search queries often return sparse or uninformative results. This paper introduces an approach for searching structured data based on probabilistic programming and nonparametric Bayes. Users specify queries in a probabilistic language that combines standard SQL database search operators with an information theoretic ranking function called predictive relevance. Predictive relevance can be calculated by a fast sparse matrix algorithm based on posterior samples from CrossCat, a nonparametric Bayesian model for high-dimensional, heterogeneously-typed data tables. The result is a flexible search technique that applies to a broad class of information retrieval problems, which we integrate into BayesDB, a probabilistic programming platform for probabilistic data analysis. This paper presents an efficient implementation applying a simple sparse matrix algorithm to the results of inference in CrossCat. The result is a scalable, domain-general search technique for sparse, multivariate, structured data that combines the strengths of SQL search with probabilistic approaches to information retrieval. Users can query by example, using real records in the database if they are familiar with the domain, or partially-specified hypothetical records if they are less familiar. Users can then narrow search results by adding Boolean filters, and by including multiple records in the query set rather than a single record. An overview of the technique and its integration into BayesDB is shown in Figure 3.

We demonstrate the proposed technique with databases of (i) US colleges, (ii) public health and macroeconomic indicators, and (iii) cars from the late 1980s. The paper empirically confirms the scalability of the technique and shows that human evaluators often prefer results from the proposed technique to results from a standard baseline.

1 Introduction

We are surrounded by multivariate data, yet it is difficult to search. Consider the problem of finding a university with a city campus, low student debt, high investment in student instruction, and tuition fees within a certain budget. The US College Scorecard dataset contains these variables plus hundreds of others. However, choosing thresholds for the quantitative variables — debt, investment, tuition, etc — requires domain knowledge. Furthermore, results grow sparse as more constraints are added. Figure 1a shows results from an SQL SELECT query with plausible thresholds for this question that yields only a single match.

This paper shows how to formulate a broad class of probabilistic search queries on structured data using probabilistic programming and information theory. The core technical idea combines SQL search operators with a ranking function called predictive relevance that assesses the relevance of database records to some set of query records, in a context defined by a variable of interest. Figures 1b and 1c show two examples, expanding and then refining the result from Figure 1a by combining predictive relevance with SQL. Predictive relevance is the probability that a candidate record is informative about the answers to a specific class of predictive queries about unknown fields in the query records.

The paper presents an efficient implementation applying a simple sparse matrix algorithm to the results of inference in CrossCat. The result is a scalable, domain-general search technique for sparse, multivariate, structured data that combines the strengths of SQL search with probabilistic approaches to information retrieval. Users can query by example, using real records in the database if they are familiar with the domain, or partially-specified hypothetical records if they are less familiar. Users can then narrow search results by adding Boolean filters, and by including multiple records in the query set rather than a single record. An overview of the technique and its integration into BayesDB is shown in Figure 3.

We demonstrate the proposed technique with databases of (i) US colleges, (ii) public health and macroeconomic indicators, and (iii) cars from the late 1980s. The paper empirically confirms the scalability of the technique and shows that human evaluators often prefer results from the proposed technique to results from a standard baseline.
(a) Standard SQL. Using a SQL WHERE clause to search for a university with a city campus, low student debt (at most $10K), high investment in student instruction (at least $50K), and a tuition within their budget (at most $50K). Due to sparsity in the dataset for the chosen thresholds, the Boolean conditions in the clause have only a single matching result, shown in the table below. The user needs to iteratively adjust the thresholds in order to obtain more results which match the search query.

| Institute              | Admit | SAT  | Tuition | Debt | Investment | Locale     |
|------------------------|-------|------|---------|------|------------|------------|
| Duke University        | 11%   | 745  | 47,243  | 7,500| 50,756     | Midsize City|
| Princeton University   | 8%    | 755  | 41,820  | 7,500| 52,224     | Large Suburb|
| Harvard University     | 6%    | 755  | 43,938  | 6,500| 49,500     | Midsize City|
| Univ of Chicago        | 8%    | 758  | 49,380  | 12,500| 83,779     | Large City  |
| Mass Inst Technology   | 8%    | 770  | 45,016  | 14,990| 62,770     | Midsize City|
| Calif Inst Technology  | 8%    | 785  | 43,362  | 11,812| 92,590     | Midsize City|
| Stanford University    | 5%    | 745  | 48,105  | 12,782| 93,146     | Large Suburb|
| Yale University        | 6%    | 750  | 45,800  | 13,774| 107,982    | Midsize City|
| Columbia University    | 7%    | 745  | 51,008  | 23,000| 80,944     | Large City  |
| University of Penn.    | 10%   | 735  | 47,668  | 21,500| 49,018     | Large City  |

(b) Relevance to hypothetical record. If the search query is instead specified as a hypothetical record in a BQL RELEVANCE PROBABILITY query, then ORDER BY can give the top-10 ranked matches. The results are all top-tier schools with high teaching investment, a city or large suburban campus, and low student debt. However, the user is surprised by the highly stringent admission rates at these colleges, which are mostly below 10%.

| Institute                | Admit | SAT  | Tuition | Debt   | Investment | Locale     |
|--------------------------|-------|------|---------|--------|------------|------------|
| Duke University          | 11%   | 745  | 47,243  | 7,500  | 50,756     | Midsize City|
| Princeton University     | 8%    | 755  | 41,820  | 7,500  | 52,224     | Large Suburb|
| Harvard University       | 6%    | 755  | 43,938  | 6,500  | 49,500     | Midsize City|
| Univ of Chicago          | 8%    | 758  | 49,380  | 12,500 | 83,779     | Large City  |
| Mass Inst Technology     | 8%    | 770  | 45,016  | 14,990 | 62,770     | Midsize City|
| Calif Inst Technology    | 8%    | 785  | 43,362  | 11,812 | 92,590     | Midsize City|
| Stanford University      | 5%    | 745  | 48,105  | 12,782 | 93,146     | Large Suburb|
| Yale University          | 6%    | 750  | 45,800  | 13,774 | 107,982    | Midsize City|
| Columbia University      | 7%    | 745  | 51,008  | 23,000 | 80,944     | Large City  |
| University of Penn.      | 10%   | 735  | 47,668  | 21,500 | 49,018     | Large City  |

(c) Relevance to observed records combined with SQL. Combining BQL and SQL to search for colleges which are most relevant to the schools from (b), in the context of "instructional investment", but that must have (i) less stringent admissions (at least 10%) and (ii) city campuses only. The quantitative search metrics of interest for the colleges in the result set are all significantly better than the national average, but they are mostly below the more selective schools in (b).

| Institute                | Admit | SAT  | Tuition | Debt   | Investment | Locale     |
|--------------------------|-------|------|---------|--------|------------|------------|
| Duke University          | 11%   | 745  | 47,243  | 7,500  | 50,756     | Midsize City|
| Princeton University     | 8%    | 755  | 41,820  | 7,500  | 52,224     | Large Suburb|
| Johns Hopkins Univ       | 17%   | 710  | 46,744  | 17,000 | 31,102     | Midsize City|
| Vanderbilt Univ          | 13%   | 760  | 43,838  | 13,000 | 79,372     | Large City  |
| University of Penn.      | 10%   | 735  | 47,668  | 21,500 | 49,018     | Large City  |
| Carnegie Mellon          | 24%   | 750  | 49,022  | 25,250 | 31,807     | Midsize City|
| Rice University          | 15%   | 750  | 40,566  | 9,642  | 40,056     | Midsize City|
| Univ Southern Calif      | 18%   | 710  | 48,280  | 21,500 | 43,170     | Midsize City|
| Cooper Union             | 15%   | 710  | 41,400  | 18,250 | 21,635     | Large City  |
| New York University      | 30%   | 685  | 46,170  | 23,300 | 30,237     | Large City  |
2 Establishing an information theoretic definition of context-specific predictive relevance

In this section, we outline the basic set-up and notations for the database search problem, and establish a formal definition of the probability of “predictive relevance” between records in the database.

2.1 Finding predictively relevant records

Suppose we are given a sparse dataset \( D = \{x_1, x_2, \ldots, x_N\} \) containing \( N \) records, where each \( x_r = (x_{r,1}, \ldots, x_{r,p}) \) is an instantiation of a \( p \)-dimensional random vector, possibly with missing values. For notational convenience, we refer to arbitrary collections of observations using sets as indices, so that \( x_{[r,c]} = \{x_{[r,c]} : r \in R, c \in R\} \). Bold-face symbols denote multivariate entities, and variables are capitalized to denote multivariate entities, and variables are capitalized as \( X_{[r,c]} \) when they are unobserved (i.e., random).

Let \( Q \subset [N] \) index a small collection of “query records” \( x_Q = \{x_q : q \in Q\} \). Our objective is to rank each item \( x_{r} \in D \) by how relevant it is for formulating predictions about values of \( x_{Q,Q} \). “In the context” of a particular dimension \( c \). We formally define the context of \( c \) as a subset of dimensions \( V \subseteq [p] \) such that for an arbitrary record \( r^* \) and each \( v \in V \), the random variable \( X_{[r^*, v]} \) is statistically dependent with \( X_{[r^*, c]} \).

In other words, we are searching for records \( i \) where knowledge of \( x_{Q,i} \) is useful for predicting \( x_{Q,Q} \), had we not known the values of these observations.

2.2 Defining context-specific predictive relevance using mutual information

We now formalize the intuition from the previous section more precisely. Let \( \mathcal{R}_c(Q, r) \) denote the probability that \( r \) is predictively relevant to \( Q \), in the context of \( c \). Furthermore, let \( c^* \) denote the index of a new dimension in the length-\( p \) random vectors, which is statistically dependent on dimension \( c \) (i.e., is in its context) but is not one of the \( p \) existing variables in the database. Since \( c^* \) indexes a novel variable, its value for each row \( r \) is itself a random variable, which we denote \( X_{[r, c^*]} \). We now define the probability that \( r \) is predictively relevant to \( Q \) in the context of \( c \) as the posterior probability that the mutual information of \( X_{[r, c^*]} \) and each query record \( X_{[q, c^*]} \) is non-zero:

\[
\mathcal{R}_c(Q, r) = \mathbb{P}
\left[
\bigcap_{q \in Q}
(I(X_{[q, c^*]} : X_{[r, c^*]}) > 0) \mid \lambda_{c^*}, \alpha, \mathcal{D}
\right].
\]

The symbol \( \lambda_{c^*} \) refers to an arbitrary set of hyperparameters which govern the distribution of dimension \( c^* \), and \( \alpha \) is a context-specific hyperparameter which controls the prior on structural dependencies between the random variables \( \{X_{[r, c^*]} : r \in [N]\} \). Moreover, the mutual information \( I \), a well-established measure for the strength of predictive relationships between random variables (Cover and Thomas [2012]), is defined in the usual way,

\[
I(X_{[q, c^*]} : X_{[r, c^*]} \mid \lambda_{c^*}, \alpha, \mathcal{D}) =
\mathbb{E}
\left[
\log
\left(
\frac{p(X_{[q, c^*]}|X_{[r, c^*]}|\lambda_{c^*}, \alpha, \mathcal{D})}{p(X_{[q, c^*]}|\lambda_{c^*}, \alpha, \mathcal{D})p(X_{[r, c^*]}|\lambda_{c^*}, \alpha, \mathcal{D})}
\right)
\right].
\]

Figure 2 illustrates the predictive relevance probability in terms of a hypothesis test on two competing graphical models, where the mutual information is non-zero in panel (a) indicating predictive relevance; and zero in panel (b) indicating predictive irrelevance.

2.3 Related Work

Our formulation of predictive relevance in terms of mutual information between new variables \( X_{[r, c^*]} \) is related to the idea of “property induction” from the cognitive science literature (Rips, 1975; Osherson et al., 1990; Shafto et al., 2008), where subjects are asked to predict whether an entity has a property, given that some other entity has that property; e.g., how likely are cats to have some new disease, given that mice are known to have the disease?

It is also informative to consider the relationship between the predictive relevance \( \mathcal{R}_c(Q, r) \) in Eq (1) and the Bayesian Sets ranking function from the statistical modeling literature (Ghahramani and Heller, 2005):

\[
\text{score}_{\text{Bayes-Sets}} (Q, r) = \frac{p(x_r | x_Q)}{p(x_r)}.
\]

Bayes Sets defines a Bayes Factor, or ratio of marginal likelihoods, which is used for hypothesis testing without assuming a structure prior. On the other hand, predictive relevance defines a posterior probability, whose value is between 0 and 1, and therefore requires a prior over dependence structure between records (our approach outlined in Section 3 is based on nonparametric Bayes). While Bayes Sets draws inferences using only the query and candidate rows without considering the rest of the data, predictive relevance probabilities are necessarily...
Figure 2: The predictive relevance of a collection of query records \( \mathcal{Q} \) to a candidate record \( r \), in the context of variable \( c \), computes the probability that \( x_{[\mathcal{Q},c]} \) and \( x_{[r,c]} \) are drawn from the same generative process, versus different generative processes. The latent variables \( z_0 \) and \( z_1 \) are indicators for the generative process of the records; and \( \theta^r_k \) (resp. \( \theta^c_k \)) are distributional parameters of data under model \( z_0 \) (resp. \( z_1 \)) for variable \( c \). Hyperparameter \( \alpha \) dictates the prior on \( z \), and \( \lambda \) dictates the prior on distributional parameters \( \theta \). The symbol \( c^* \) denotes a new dimension which is statistically dependent on \( c \), and for which no values are observed for either \( \mathcal{Q} \) or \( r \). Conditioned on hyperparameters, knowing \( X_{[r,c^*]} \) in \( \mathcal{Q} \) carries information about the unknown values \( x_{[\mathcal{Q},c^*]} \), whereas in \( r \) it does not.

conditioned on \( \mathcal{D} \) as in Eq (1). Finally Bayes Sets considers the entire data vectors for scoring, whereas predictive relevance considers only dimensions which are in the context of a variable \( c \), making it possible for two records to be predictively relevant in some context but probably predictively irrelevant in another.

### 3 Computing the probability of predictive relevance using nonparametric Bayes

This section describes the cross-categorization prior (CrossCat, Mansinghka et al. (2016)) and outlines algorithms which use CrossCat to efficiently estimate predictive relevance probabilities Eq (1) for sparse, high-dimensional, and heterogeneously-typed data tables.

CrossCat is a nonparametric Bayesian model which learns the full joint distribution of \( p \) variables using structure learning and divide-and-conquer. The generative model begins by partitioning the set of \( p \) variables into blocks using a Chinese restaurant process. This step is CrossCat’s “outer” clustering, since it partitions the columns of a data table where variables correspond to columns, and records correspond to rows. Let \( \pi \) denote the partition of \( \{p\} \) whose \( k \)-th block is \( \mathcal{V}^k \subseteq \{p\} \); for \( j \neq k \), all variables in \( \mathcal{V}^k \) are mutually (marginally and conditionally) independent of all variables in \( \mathcal{V}^j \). Within block \( k \), the variables \( x_{[r,y^k]} \) follow a Dirichlet process mixture model (Escobar and West 1995), where we focus on the case the joint distribution factorizes given the latent cluster assignment \( z_{r,c}^k \). This step is an “inner” clustering in CrossCat, since it specifies a cluster assignment for each row in block \( k \). CrossCat’s combinatorial structure requires detailed notation to track the latent variables and dependencies between them. The generative process for an exchangeable sequence \( \{X_1, \ldots, X_N\} \) of \( N \) random vectors is summarized below.

### Table 1: Symbols used to describe CrossCat prior

| Symbol | Description |
|--------|-------------|
| \( \alpha_0 \) | Concentration hyperparameter of column CRP |
| \( \alpha_1 \) | Concentration hyperparameter of row CRP |
| \( v_c \) | Index of variable \( c \) in column partition |
| \( \mathcal{V}^k \) | List of variables in block \( k \) of column partition |
| \( z_{r,c}^k \) | Cluster index of \( r \) in row partition of block \( k \) |
| \( C_y^k \) | List of rows in cluster \( y \) of block \( k \) |
| \( M_c \) | Joint distribution of data for variable \( c \) |
| \( \lambda_c \) | Hyperparameters of \( M_c \) |
| \( X_{[r,c]} \) | \( r \)-th observation of variable \( c \) |
| \( \text{SET}(l) \) | Unique items in list \( l \) |

### CROSSCAT PRIOR

1. Sample column partition into blocks.
   \[ v = (v_1, \ldots, v_p) \sim \text{CRP}(\cdot|\alpha_0) \]
   \[ \mathcal{V}^k \leftarrow \{ c \in [p] : v_c = k \} \enspace \text{foreach } k \in \text{SET}(v) \]

2. Sample row partitions within each block.
   \[ z_{r,c}^k = (z_{r,1}^k, \ldots, z_{r,N}^k) \sim \text{CRP}(\cdot|\alpha_1) \enspace \text{foreach } k \in \text{SET}(v) \]
   \[ C_y^k \leftarrow \{ r \in [N] : z_{r,c}^k = y \} \enspace \text{foreach } k \in \text{SET}(v) \]
   \[ \text{SET}(z^k_c) \leftarrow \{ X_{[r,c]} : r \in C_y^k \} \sim M_c(\cdot|\lambda_c) \enspace \text{foreach } k \in \text{SET}(v) \]
   \[ \text{SET}(z^k_c) \leftarrow \{ y \in \text{SET}(z^k_c) \} \enspace \text{foreach } c \in \mathcal{V}^k \]
The representation of CrossCat in this paper assumes that data within a cluster is sampled jointly (step 3), marginalizing over cluster-specific distributional parameters:

\[ M_\theta(x_{[r,c]}, \lambda_c) = \int_\theta \prod_{r \in C^*_g} p(x_{[r,c]}|\theta)p(\theta|\lambda_c)d\theta. \]

This assumption suffices for our development of predictive relevance, and is applicable to a broad class of statistical data types with conjugate-prior-likelihood representations such as Beta-Bernoulli for binary, Dirichlet-Multinomial for categorical, Normal-Inverse-Gamma-Normal for real values, and Gamma-Poisson for counts.

Given dataset \( D \), we refer to [Obermeyer et al., 2014] and [Saad and Mansinghka, 2016] for scalable algorithms for posterior inference in CrossCat, and assume we have access to an ensemble of \( H \) posterior samples \( \{\hat{\phi}^1, \ldots, \hat{\phi}^H\} \) where each \( \hat{\phi}^h \) is a realization of all variables in Table 1.

### 3.1 Estimating predictive relevance using CrossCat

We now describe how to use posterior samples of CrossCat to efficiently estimate the predictive relevance probability \( R_c(Q,r) \) from Eq (1). Letting \( c \) denote the context variable, we formalize the novel variable \( c^* \) as a fresh column in the tabular population which is assigned to the same block \( k \) as \( c \) (i.e. \( k = v_c = v_{c^*} \)). As shown by [Saad and Mansinghka, 2017], structural dependencies induced by CrossCat’s variable partition are related to an upper-bound on the probability there exists a statistical dependence between \( c \) and \( c^* \). To estimate Eq (1), we first treat the mutual information between \( X_{[q,c^*]} \) and \( X_{[r,c^*]} \) as a derived random variable, which is a function of their random cluster assignments \( z_q^k \) and \( z_r^k \):

\[ (z_q^k, z_r^k) \rightarrow I(X_{[q,c^*]} : X_{[r,c^*]} | z_q^k, z_r^k, \alpha_1, \lambda_{c^*}) \quad (4) \]

The key insight, implied by step 3 of the CrossCat prior, is that, conditioned on their assignments, rows from different clusters are independent. Thus, we can estimate predictive relevance by maximizing the lower-bound on the mutual information. We then formalize predictive relevance as the relative importance of each context, which we formalize as a derived variable \( \hat{\phi}^1, \ldots, \hat{\phi}^H \) as

\[ \hat{\phi}^1, \ldots, \hat{\phi}^H \]

which are a realization of all variables in Table 1.
different clusters are sampled independently, which gives

\[ z_q^k \neq z_r^k \]

\[ \iff p(x_{[q,c^*]} : x_{[r,c^*]} | z_q^k, z_r^k, \lambda_{c^*}, \alpha_1, D) = p(x_{[q,c^*]} | z_q^k, \lambda_{c^*}, \alpha_1, D) \times p(x_{[r,c^*]} | z_r^k, \lambda_{c^*}, \alpha_1, D) \]

\[ \iff I(X_{[q,c^*]} : X_{[r,c^*]} | z_q^k, z_r^k, \lambda_{c^*}, \alpha_1, D) = 0, \quad (5) \]

where the final implication follows directly from the definition of mutual information in Eq (4). Note that Eq (5) does not depend on the particular choice of \( \lambda_{c^*} \), and indeed this hyperparameter is never represented explicitly. Moreover, hyperparameter \( \alpha_1 \) (corresponding to \( \alpha \) in Figure 2) is the concentration of the Dirichlet process for CrossCat row partitions.

Eq (5) implies that we can estimate the probability of non-zero mutual information between \( X_{[q,c^*]} \) and each \( X_{[r,c^*]} \) for \( q \in Q \) by forming a Monte Carlo estimate from the ensemble of posterior CrossCat samples,

\[
R_c(Q, r) = \Pr \left[ \bigcup_{q \in Q} I(X_{[q,c^*]} : X_{[r,c^*]}) > 0 \right | \lambda_{c^*}, \alpha_1, D] \]

\[
= \Pr \left[ \bigcup_{q \in Q} (z_q^c = z_r^c) \right | \alpha_1, D] \]

\[
\approx \frac{1}{H} \sum_{h=1}^{H} \left[ \prod_{q \in Q} \left( \frac{1}{H} \sum_{r=1}^{H} \mathbb{1}\left( z_q^c = z_r^c \right) \right) \right], \quad (6)
\]

where \( \phi_c^h \) indexes the context block, and \( z_r^{c,h} \) denotes cluster assignment of \( r \) in the row partition of \( \phi_c^h \), according to the sample \( \hat{\phi}^h \). Algorithm 1 outlines a procedure (used by the BayesDB query engine from Figure 3) for formulating a Monte Carlo based estimator for a predictive relevance query using CrossCat.

Algorithm 1: CrossCat-Predictive-Relevance

**Require:** CrossCat samples: \( \hat{\phi}^h \) for \( h = 1, \ldots, H \), query rows: \( Q = \{ q_i : 1 \leq i \leq |Q| \} \)

**Ensure:** predictive relevance of each existing row in \( D \) to \( Q \)

1: for \( r = 1, \ldots, N \) do \( \triangleright \) for each existing row
2: for \( h = 1, \ldots, H \) do \( \triangleright \) for each CrossCat sample
3: \( k \leftarrow \phi_c^h \) \( \triangleright \) retrieve context block
4: for \( q \in Q \) do \( \triangleright \) for each query row
5: if \( z_q^c \neq z_r^c \) then \( \triangleright \) \( r \) and \( q \) are different clusters
6: \( R_c^h(Q, r) \leftarrow 0 \) \( \triangleright \) \( r \) irrelevant to some \( q \)
7: break
8: else \( \triangleright \) \( r \) in same cluster as all \( q \) in \( Q \)
9: \( R_c^h(Q, r) \leftarrow 1 \) \( \triangleright \) \( r \) relevant to all \( q \)
10: \( R_c(Q, r) \leftarrow \frac{1}{H} \sum_{h=1}^{H} R_c^h(Q, r) \) \( \triangleright \) average relevances
11: return \( \{ R_c(Q, r) : 1 \leq r \leq N \} \)

3.2 Optimizing the estimator using a sparse matrix-vector multiplication

In this section, we show how to greatly optimize the naive, nested for-loop implementation in Algorithm 1 by instead computing predictive relevance for all \( r \) through a single matrix-vector multiplication.

Define the pairwise cluster co-occurrence matrix \( S_{c,h}^k \) for block \( k \) of CrossCat sample \( \hat{\phi}^h \) to have binary entries \( S_{c,h}^k = I(z_q^c = z_r^c) \). Furthermore, let \( \mathbf{1}_Q \) denote a length-\( N \) vector with a 1 at indices \( q \in Q \) and 0 otherwise. We vectorize \( R_c(Q, r) \) across \( r \in [N] \) by:

\[
\mathbf{u}^h = \frac{1}{|Q|} S_{c,h}^k \mathbf{1}_Q \quad h = 1, \ldots, H
\]

\[
R_c(Q, \cdot) = \frac{1}{H} \sum_{h=1}^{H} \mathbf{u}^h. \quad (8)
\]

The resulting length-\( N \) vector \( \mathbf{u}^h \) in Eq (7) satisfies \( \mathbf{u}^h_k = 1 \) if and only if \( z_q^c = z_r^c \) for all \( q \in Q \), which we identify as the argument of the indicator function in Eq (6). Finally, by averaging \( \mathbf{u}^h \) across the \( H \) samples in Eq (8), we arrive at the vector of relevance probabilities.

For large datasets, constructing the \( N \times N \) matrix \( S_{c,h}^k \) using \( \Theta(N^2) \) operations is prohibitively expensive. Algorithm 2 describes an efficient procedure that exploits CrossCat’s sparsity to build \( S_{c,h}^k \) in expected time \( \ll O(N^2) \) by using (i) a sparse matrix representation, and (ii) CrossCat’s partition data structures to avoid considering all pairs of rows. This fast construction means that Eq (7) is practical to implement for large data tables.

The algorithm’s running time depends on (i) the number of clusters \( |\text{Set}(\tilde{z}^k)| \) in line 1; (ii) the average number of rows per cluster \( |C_{p}^h| \) in line 2; and (iii) the data structures used to represent \( S_{c,h}^k \) in line 3. Under the CRP prior, the expected number of clusters is \( O(\alpha_1 \log(N)) \), which implies an average occupancy of \( O(N/(\alpha_1 \log(N))) \) rows per cluster. If the sparse binary matrix is stored with a list-of-lists representation, then the update in line 3 requires \( O(1) \) time. Furthermore, we emphasize that since \( S_{c,h}^k \) does not depend on \( Q \), its cost of construction is amortized over an arbitrary number of queries.

Algorithm 2: CrossCat-Co-Occurrence-Matrix

**Require:** CrossCat sample \( \hat{\phi}^h \); block index \( k \).

**Ensure:** Pairwise co-occurrence matrix \( S_{c,h}^k \)

1: for \( y \in \text{Set}(\tilde{z}^k) \) do \( \triangleright \) for each cluster in block \( k \)
2: for \( r \in C_{p}^h \) do \( \triangleright \) for each row in the cluster
3: Set \( S_{c,h}^k = 1 \), where \( j \in C_{p}^h \) \( \triangleright \) update the matrix
4: return \( S_{c,h}^k \)
3.3 Computing predictive relevance probabilities for query records that are not in the database

We have so far assumed that the query records must consist of items that already exist in the database. This section relaxes this restrictive assumption by illustrating how to compute relevance probabilities for search records which do not exist in \( \mathcal{D} \), and are instead specified by the user on a per-query basis (refer to the BQL query in Figure 3 for an example of a hypothetical query record). The key idea is to (i) incorporate the new records into each CrossCat sample \( \hat{\phi}^k \) by using a Gibbs-step to sample cluster assignments from the joint posterior (Neal 2000); (ii) compute Eq (7) on the updated samples; and (iii) unincorporate the records, leaving the original samples unmutated.

Letting \( \{x_{i|N+1} : 1 \leq i \leq t\} \) denote \( t \) (partially observed) new rows and \( \mathcal{Q} = \{N+1, \ldots, N+t\} \) the query, we compute \( \mathcal{R}_q(\mathcal{Q}, r) \) for all \( r \) by first applying CrossCat-Incorporate-Record (Algorithm 3) to each \( q \in \mathcal{Q} \) sequentially. Sequential incorporation corresponds to sampling from the sequence of predictive distributions, which, by exchangeability, ensures that each updated \( \hat{\phi}^k \) contains a sample of cluster assignments from the joint distribution, guaranteeing correctness of the Monte Carlo estimator in Eq (6). Note that since CrossCat specifies a non-parametric mixture, the proposal clusters include all existing clusters, plus one singleton cluster \( \max(z^k) + 1 \). We next update the co-occurrence matrices in time linear in the size of the sampled cluster and then evaluate Eq (7) and (8). To unincorporate, we reverse lines 5 and restore the co-occurrence matrices. Figure 4 confirms that the runtime scaling is asymptotically linear, varying the (i) number of new rows, (ii) fraction of variables specified for the new rows that are in the context block (i.e. query sparsity), (iii) number of clusters in the context block, and (iv) number of variables in the context block.

**Algorithm 3** CrossCat-Incorporate-Record

**Require:** CrossCat sample \( \hat{\phi} \); context \( c \); new row \( x_{N+1} \)

**Ensure:** Updated crosscat sample \( \hat{\phi}' \)

1. \( k \leftarrow r_{c} \) \text{ \quad \text{\triangleright retrieve block of context variable} \quad} 
2. \( Y \leftarrow \max(z^k) + 1 \) \text{ \quad \text{\triangleright retrieve proposal clusters} \quad} 
3. for \( y = 1, \ldots, Y \) do \text{ \quad \text{\triangleright compute cluster probabilities} \quad} 
4. \( n_y \leftarrow \begin{cases} \mathcal{C}_v^d & \text{if } y \in z^k \\ \mathcal{C}_y & \text{if } y = \max(z^k) + 1 \end{cases} \) 
5. \( l_y \leftarrow \prod_{v \in V} M_v(x_{|N+1,c|}, \tau_0) \) 
6. \( z^k_{N+1} \sim \text{Categorical}(l_1, \ldots, l_Y) \) \quad \text{\triangleright sample cluster} 
7. \( z^k \leftarrow z^k \cup \{z^k_{N+1}\} \) \quad \text{\triangleright append cluster assignment} 
8. \( \mathcal{C}^k_{z^k_{N+1}} \leftarrow \mathcal{C}^k_{z^k_{N+1}} \cup \{N+1\} \) \quad \text{\triangleright append row to cluster} 
9. \( \mathcal{D}' \leftarrow \mathcal{D} \cup \{x_{N+1, v^k}\} \) \quad \text{\triangleright append record to database} 
10. return \( \hat{\phi}' \) \quad \text{\triangleright return updated sample} 

Figure 4: Empirical measurements of the asymptotic scaling of CrossCat-Incorporate-Record (Algorithm 3) on the Gapminder dataset (Section 4). The color of each measurement indicates the number of variables in the block of the context variable; each column shows a different number of records (1, 2, 4, and 8) incorporated by the algorithm. The top panels show that, for a fixed number of variables in the context, the runtime (in milliseconds) decays linearly with the sparsity of the hypothetical records (dimensions which are not in the same block as the context variable are ignored). The lower panels show the runtime increasing linearly with the number of clusters in the context; the number of variables in the context dictates the slope of the curve.

4 Applications

This section illustrates the efficacy of predictive relevance in BayesDB by applying the technique to several search problems in real-world, sparse, and high-dimensional datasets of public interest

4.1 College Scorecard

The College Scorecard (Council of Economic Advisers 2015) is a federal dataset consisting of over 7000 colleges and 1700 variables, and is used to measure and improve the performance of US institutions of higher education. These variables include a broad set of categories

---

2 Appendix D contains a further application to a dataset of classic cars from 1987. Appendix A formally describes the integration of RELVANCE PROBABILITY into BayesDB as an expression in the Bayesian Query Language (Figure 3).
Pairwise CrossCat predictive relevances in different contexts

(a) CrossCat (life expectancy)  (b) CrossCat (exports, % gdp)

Pairwise cosine similarities in different contexts

(c) Cosine (life expectancy)  (d) Cosine (exports, % gdp)

| Concept                  | Representative Countries in the Concept |
|--------------------------|-----------------------------------------|
| Low-Income Nations       | Burundi, Ethiopia, Uganda, Benin, Malawi, Rwanda, Togo, Guinea, Senegal, Afghanistan, Malawi |
| Post-Soviet Nations      | Russia, Ukraine, Bulgaria, Belarus, Slovakia, Serbia, Croatia, Poland, Hungary, Romania, Latvia |
| Western Democracies      | France, Britain, Germany, Netherlands, Italy, Denmark, Finland, Sweden, Norway, Australia, Japan |
| Small Wealthy Nations    | Qatar, Bahrain, Kuwait, Emirates, Singapore, Israel, Gibraltar, Bermuda, Jersey, Cayman Islands |

(e) Countries which are mutually predictive in the context of “life expectancy” according to CrossCat’s relevance matrix

Figure 5: (a) – (d) Pairwise heatmaps of countries from the Gapminder dataset in the contexts of “life expectancy at birth” and “exports of goods and services (% of gdp “), using CrossCat predictive relevance and cosine similarity. Each row and column in a matrix is a country, and a cell value (between 0 and 1) indicates the strength of match between those two countries. (e) CrossCat learns a sparse set of relevances; for “life expectancy”, these broadly correspond to common-sense taxonomies of countries based on shared geographic, political and macroeconomic characteristics. These concepts were manually labeled by inspecting clusters of countries in matrix (a); the colors in the matrix correspond to countries in the table which belong to the concept of that color. Note that the relevance structure differs significantly when ranking in the context of “exports, % gdp”, as shown by the colors in matrix (b) where the clusters of mutually relevant countries form a different pattern than in (a). Cosine similarity learns dense, noisy sets of spuriously high-ranking countries with coarser structure, as shown in (c) and (d). Refer to Appendix C for more baselines.

4.2 Gapminder

Gapminder [Rosling 2008] is an extensive longitudinal dataset of over ~320 global macroeconomic variables of population growth, education, climate, trade, welfare and health for 225 countries. Our experiments are based on a cross-section of the data from the year 2002. The data is sparse, with 35% of the data missing. Figure 5 shows heatmaps of the pairwise predictive relevances for all countries in the dataset under different contexts, and compares the results to cosine similarity. Clusters of predictively relevant countries form common-sense taxonomies; refer to the caption for further discussion.

Figure 6 finds the top-15 countries in the dataset ordered by their predictive relevance to the United States, in the context of “life expectancy at birth”. Table 6b shows representative variables which are in the context; these variables have the highest dependence probability with the context variable, according a Monte Carlo estimate using 64 posterior CrossCat samples. The countries in Figure 6a are all rich, Western democracies with highly developed economies and advanced healthcare systems. To quantitatively evaluate the quality of top-ranked countries returned by predictive relevance, we ran the tech-
%bql .barplot
... ESTIMATE "country",
... RELEVANCE PROBABILITY
... TO EXISTING ROWS IN
... ('United States')
... IN THE CONTEXT OF
... "life expectancy at birth"
... AS "rel_us_lifexp"
... FROM gapminder
... ORDER BY "rel_us_lifexp" DESC
... LIMIT 15

United States
Iceland
Australia
Ireland
Andorra
New Zealand
Cyprus
San Marino
Austria
Belgium
Canada
Switzerland
Germany
Denmark

(a) Relevance to USA in the context of “life expectancy”

Measles, mumps, & rubella vaccines (% population)
Under 5 mortality rate
Dead children per woman
access to improved sanitation facilities (% population)
access to improved drinking water sources (% population)
human development index
body mass index (kg/m²)
murder rate (per 100,000)
food supply (kilocalories per person)
contraceptive prevalence (% women ages 15-49)
alcohol consumption (liters per adult)
prevalence of tobacco use among adults (% population)

(b) Variables in the context of “life expectancy at birth”

Figure 6: Using BQL to search for the top 15 countries in the Gapminder dataset ranked by their relevance to the United States in the context of “life expectancy at birth” finds rich, Western democracies with advanced healthcare systems.

Figure 7: Comparing human preferences for the top-ranked countries returned by cosine similarity versus CrossCat predictive relevance, in 10 representative search queries (shown on the y-axis). For each query, human subjects were given the top 10 most relevant countries, according to both cosine and CrossCat, and then asked to choose which results they preferred, if any. We scored the responses in the following way: “countries returned by cosine are more relevant” (score = -1); “countries returned by CrossCat are more relevant” (score = +1); “both results are equally relevant” (score = 0). The x-axis shows the scores averaged across 70 humans, surveyed on the cloud through crowdflower.com. Error bars represent one standard error of the mean. For most of the queries, human preferences are biased in favor of CrossCat’s rankings. Further details on the experimental design and results are given in Appendix B.

5 Discussion

This paper has shown how to perform probabilistic searches of structured data by combining ideas from probabilistic programming, information theory, and non-parametric Bayes. The demonstrations suggest the technique can be effective on sparse, real-world databases from multiple domains and produce results that human evaluators often preferred to a standard baseline.

More empirical evaluation is clearly needed, ideally including tests of hundreds or thousands of queries, more complex query types, and comparisons with query results manually provided by human domain experts. In fact, search via predictive relevance in the context of variables drawn from learned representations of data could potentially provide a meaningful way to compare representation learning techniques. It also may be fruitful to build a distributed implementation suitable for database representations of web-scale data, including photos, social network users, and web pages.

Relatively unstructured probabilistic models, such as topic models, proved sufficient for making unstructured text data far more accessible and useful. We hope this paper helps illustrate the potential for structured probabilistic models to improve the accessibility and usefulness of structured data.
Acknowledgments

The authors wish to acknowledge Ryan Rifkin, Anna Comerford, Marie Huber, and Richard Tibbets for helpful comments on early drafts. 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

Stef Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. Journal of Statistical Software, 45(3), 2011.

Council of Economic Advisers. Using federal data to measure and improve the performance of u.s. institutions of higher education. Technical report, Executive Office of the President of the United States, 2015.

Thomas Cover and Joy Thomas. Elements of Information Theory. Wiley Series in Telecommunications and Signal Processing. Wiley, 2012.

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

Zoubin Ghahramani and Katherine A. Heller. Bayesian sets. In Proceedings of the 18th International Conference on Neural Information Processing Systems, pages 435–442. MIT Press, 2005.

Dennis Kibler, David W Aha, and Marc K Albert. Instance-based prediction of real-valued attributes. Computational Intelligence, 5(2):51–57, 1989.

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.

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.

Radford M Neal. Markov chain sampling methods for dirichlet process mixture models. Journal of Computational and Graphical Statistics, 9(2):249–265, 2000.

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.

Daniel N Osherson, Edward E Smith, Ormond Wilkie, Alejandro Lopez, and Eldar Shafir. Category-based induction. Psychological review, 97(2):185, 1990.

Lance J. Rips. Inductive judgments about natural categories. Journal of Verbal Learning and Verbal Behavior, 14(6):665–681, 1975.

Hans Rosling. Gapminder: Unveiling the beauty of statistics for a fact based world view, 2008. URL https://www.gapminder.org/data/

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

Feras Saad and Vikash Mansinghka. Detecting dependencies in sparse, multivariate databases using probabilistic programming and non-parametric bayes. In Proceedings of the Twentieth International Conference on Artificial Intelligence and Statistics. JMLR.org, 2017.

Patrick Shafto, Charles Kemp, Elizabeth Bonawitz, John Coley, and Joshua Tenenbaum. Inductive reasoning about causally transmitted properties. Cognition, 109 (2):175–192, 2008.
Appendices

A Integrating predictive relevance as a ranking function in BayesDB

This section describes the integration of predictive relevance into BayesDB (Mansinghka et al., 2015; Saad and Mansinghka, 2016), a probabilistic programming platform for probabilistic data analysis.

New syntaxes in the Bayesian Query Language (BQL) allow a user to express predictive relevance queries where the query set can be an arbitrary combination of existing and hypothetical records. We implement predictive relevance in BQL as an expression with the following syntaxes, depending on the specification of the query records.

• Query records are existing rows.
  \[
  \text{RELEVANCE PROBABILITY TO EXISTING ROWS IN } \langle \text{expression} \rangle \text{ IN THE CONTEXT OF } \langle \text{context-var} \rangle
  \]

• Query records are hypothetical rows.
  \[
  \text{RELEVANCE PROBABILITY TO HYPOTHETICAL ROWS WITH VALUES } (\langle \text{values} \rangle) \text{ IN THE CONTEXT OF } \langle \text{context-var} \rangle
  \]

• Query records are existing and hypothetical rows.
  \[
  \text{RELEVANCE PROBABILITY TO EXISTING ROWS IN } \langle \text{expression} \rangle \text{ AND HYPOTHETICAL ROWS WITH VALUES } (\langle \text{values} \rangle) \text{ IN THE CONTEXT OF } \langle \text{context-var} \rangle
  \]

The expression is formally implemented as a 1-row BQL estimand, which specifies a map \( r \mapsto R_r(Q,r) \) for each record in the table. As shown in the expressions above, query records are specified by the user in two ways: (i) by giving a collection of \textsc{Existing Rows}, whose primary key indexes are either specified manually, or retrieved using an arbitrary BQL \textsc{Expression}; (ii) by specifying one or more \textsc{Hypothetical Records} with their \textsc{Values} as a list of column-value pairs. These new rows are first incorporated using Algorithm 3 from Section 3.3 and they are then unincorporated after the query is finished. The \textsc{Context-Var} can be any variable in the tabular population.

As a 1-row function in the structured query language, the \textsc{Relevance Probability} expression can be used in a variety of settings. Some typical use-cases are shown in the following examples, where we use only existing query rows for simplicity.

• As a column in an \textsc{Estimate} query.
  \[
  \text{ESTIMATE}
  \begin{align*}
  \text{"rowid"}, \\
  \text{RELEVANCE PROBABILITY TO EXISTING ROWS IN } \langle \text{expression} \rangle \\
  \text{IN THE CONTEXT OF } \langle \text{context-var} \rangle \\
  \text{FROM } \langle \text{table} \rangle
  \end{align*}
  \]

• As a filter in \textsc{Where} clause.
  \[
  \text{ESTIMATE}
  \begin{align*}
  \text{"rowid" FROM } \langle \text{table} \rangle
  \end{align*}
  \text{WHERE (}
  \begin{align*}
  \text{RELEVANCE PROBABILITY TO EXISTING ROWS IN } \langle \text{expression} \rangle \\
  \text{IN THE CONTEXT OF } \langle \text{context-var} \rangle
  \end{align*}
  \text{)} > 0.5
  \]

• As a comparator in an \textsc{Order By} clause.
  \[
  \text{ESTIMATE}
  \begin{align*}
  \text{"rowid" FROM } \langle \text{table} \rangle
  \end{align*}
  \text{ORDER BY}
  \begin{align*}
  \text{RELEVANCE PROBABILITY TO EXISTING ROWS IN } \langle \text{expression} \rangle \\
  \text{IN THE CONTEXT OF } \langle \text{context-var} \rangle
  \end{align*}
  \text{[ASC | DESC]}
  \]

It is also possible to perform arithmetic operations and Boolean comparisons on relevance probabilities.

• Finding the mean relevance probability for a set of \textsc{Rowids} of interest.
  \[
  \text{ESTIMATE}
  \begin{align*}
  \text{AVG (}
  \begin{align*}
  \text{RELEVANCE PROBABILITY TO EXISTING ROWS IN } \langle \text{expression} \rangle \\
  \text{IN THE CONTEXT OF } \langle \text{context-var} \rangle
  \end{align*}
  \text{) FROM } \langle \text{table} \rangle
  \end{align*}
  \text{WHERE "rowid" IN } \langle \text{expression} \rangle
  \]

• Finding rows which are more relevant in some context \( c_0 \) than in another context \( c_1 \).
  \[
  \text{ESTIMATE}
  \begin{align*}
  \text{"rowid" FROM } \langle \text{table} \rangle
  \end{align*}
  \text{WHERE (}
  \begin{align*}
  \text{RELEVANCE PROBABILITY TO EXISTING ROWS IN } \langle \text{expression} \rangle \\
  \text{IN THE CONTEXT OF } \langle \text{context-var-0} \rangle
  \end{align*}
  \text{)} > (}
  \begin{align*}
  \text{RELEVANCE PROBABILITY TO EXISTING ROWS IN } \langle \text{expression} \rangle \\
  \text{IN THE CONTEXT OF } \langle \text{context-var-1} \rangle
  \end{align*}
  \]


## B Predictive relevance and cosine similarity on Gapminder human evaluation queries

| Country Pair                      | Predictive Relevance | Cosine Similarity |
|-----------------------------------|----------------------|-------------------|
| **Saudi Arabia, Democracy**       |                      |                   |
| A: Saudi Arabia                   |                      |                   |
| B: Venezuela                      |                      |                   |
| A: Libya                          |                      |                   |
| B: Israel                         |                      |                   |
| A: Kuwait                         |                      |                   |
| B: Saudi Arabia                   |                      |                   |
| A: W. Sahara                      |                      |                   |
| B: Malta                          |                      |                   |
| A: Qatar                          |                      |                   |
| B: Puerto Rico                    |                      |                   |
| A: Algeria                        |                      |                   |
| B: Spain                          |                      |                   |
| A: Bangladesh                      |                      |                   |
| B: United States                   |                      |                   |
| A: Australia                      |                      |                   |
| B: Australia                      |                      |                   |
| A: Iceland                        |                      |                   |
| B: Ireland                        |                      |                   |
| A: Andorra                        |                      |                   |
| B: Canada                         |                      |                   |
| A: United States                   |                      |                   |
| B: Iceland                        |                      |                   |
| A: New Zealand                     |                      |                   |
| B: Malta                          |                      |                   |
| A: Austria                        |                      |                   |
| B: Ireland                        |                      |                   |
| A: Belgium                        |                      |                   |
| B: Finland                        |                      |                   |
| A: Germany                        |                      |                   |
| B: Switzerland                    |                      |                   |
| A: Switzerland                    |                      |                   |
| B: Japan                          |                      |                   |
| A: Sri Lanka                       |                      |                   |
| B: Australia                      |                      |                   |

**Figure 8:** The top-10 ranking countries returned by predictive relevance and cosine similarity for each of the 10 queries used for the human evaluation in Figure 7. For each country-context search query, we showed seventy subjects (surveyed on the AI crowdsourcing platform) a pair of tables. We then asked each subject to select the table which contains more relevant results to the search query, or report that both tables contain equally relevant results. The tables above show the top-ranked countries using CrossCat predictive relevance and cosine similarity, with a histogram of the human responses. The caption of Figure 7 describes how we converted these raw histograms into scores between -1 and 1 that are displayed in the main text. The tables showing countries ranked using CrossCat predictive relevance are: Saudi Arabia (A); United States (B); Australia (A); Bangladesh (B); Bulgaria (B); Japan (B); Qatar (A); UK (B); Hong Kong (B); Singapore (B).
C Pairwise heatmaps on Gapminder countries using baseline methods

**COSINE SIMILARITY**

- Median Imputation (5 vars)
- Median Imputation (10 vars)
- Median Imputation (15 vars)
- Median Imputation (20 vars)

- MICE Imputation (5 vars)
- MICE Imputation (10 vars)
- MICE Imputation (15 vars)
- MICE Imputation (20 vars)

**BRAY-CURTIS COEFFICIENT**

- Median Imputation (5 vars)
- Median Imputation (10 vars)
- Median Imputation (15 vars)
- Median Imputation (20 vars)

- MICE Imputation (5 vars)
- MICE Imputation (10 vars)
- MICE Imputation (15 vars)
- MICE Imputation (20 vars)

**EUCLIDEAN DISTANCE**

- Median Imputation (5 vars)
- Median Imputation (10 vars)
- Median Imputation (15 vars)
- Median Imputation (20 vars)

- MICE Imputation (5 vars)
- MICE Imputation (10 vars)
- MICE Imputation (15 vars)
- MICE Imputation (20 vars)

**Figure 9:** Pairwise heatmaps of countries in Gapminder dataset in the context of “life expectancy at birth”, using various distance and similarity measures on the country vectors. Each heatmap is labeled with the imputation technique (median or MICE [Buuren and Groothuis-Oudshoorn, 2011]), and the number of variables in the context (i.e. dimensionality of the vectors). These techniques struggle with sparsity and their structures are much noisier than the results of relevance probability shown in Figure 5a and Table 5c.
D Application to a dataset of 1987 cars

%bql CREATE TABLE cars_1987_raw
... FROM 'cars_1987.csv'

%bql SELECT
... "make",
... "price",
... "wheels",
... "doors",
... "engine",
... "horsepower",
... "body"
... FROM cars_1987_raw
... WHERE "price" < 45000
... AND "wheels" = 'rear'
... AND "doors" = 'four'
... AND "engine" >= 250
... AND "horsepower" > 180
... AND "body" = 'sedan'

(a) Suppose a customer wishes to purchase a classic car from 1987 with a budget of $45,000 and a desired set of technical specifications. They first load a csv file of 200 cars with 26 variables into a BayesDB table, and then specify the search conditions as Boolean filters in a SQL WHERE clause. Due to sparsity in the table, only one record is returned. To obtain more relevant results, the user needs to broaden the specifications in the query.

| make    | price    | wheels | doors | engine | horsepower | body   |
|---------|----------|--------|-------|--------|------------|--------|
| mercedes| 40,960   | rear   | four  | 308    | 184        | sedan  |

%mm1 CREATE POPULATION
... cars_1987
... FOR cars_1987_raw
... WITH SCHEMA (
... GUESS STATISTICAL
... TYPES FOR (*);
... )

%mm1 CREATE METAMODEL m FOR cars_1987
... WITH BASELINE crosscat;

%mm1 INITIALIZE 100 MODELS FOR m;
%mm1 ANALYZE m FOR 1 MINUTE;

%bql .heatmap ESTIMATE
... DEPENDENCE PROBABILITY
... FROM PAIRWISE VARIABLES
... OF cars_1987

%bql SELECT
... "make",
... "price",
... "wheels",
... "doors",
... "engine-size",
... "horsepower",
... "style"
... FROM cars_1987
... ORDER BY
... RELEVANCE PROBABILITY
... TO HYPOTHETICAL ROW ((
... "price" = 42000,
... "wheels" = 'rear',
... "doors" = 'four',
... "engine" = 250,
... "horsepower" = 180,
... "body" = 'sedan')
... )
... IN THE CONTEXT OF
... "price"
... LIMIT 10

(b) Building CrossCat models in BayesDB for the cars_1987 population learns a full joint probabilistic model over all variables. The ESTIMATE DEPENDENCE PROBABILITY query allows the user to plot a heatmap of probable dependencies between car characteristics. The context of "price" probably contains the majority of other variables in the search query.

(c) Using ORDER BY RELEVANCE PROBABILITY in BQL ranks each car in the table by its relevance to the user’s specifications, which are specified as a hypothetical row. The top-10 ranked cars by probability of relevance to the search query, in the context of price, are shown below in the table below. The user can now inspect further characteristics of this subset of cars, to find ones that they like best.

| make    | price    | wheels | doors | engine | horsepower | body   |
|---------|----------|--------|-------|--------|------------|--------|
| jaguar  | 35,550   | rear   | four  | 258    | 176        | sedan  |
| jaguar  | 32,250   | rear   | four  | 258    | 176        | sedan  |
| mercedes| 40,960   | rear   | four  | 308    | 184        | sedan  |
| mercedes| 45,400   | rear   | two   | 304    | 184        | hardtop|
| mercedes| 34,184   | rear   | four  | 234    | 155        | sedan  |
| mercedes| 35,056   | rear   | two   | 234    | 155        | convertible|
| bmw     | 36,880   | rear   | four  | 209    | 182        | sedan  |
| bmw     | 41,315   | rear   | two   | 209    | 182        | sedan  |
| bmw     | 30,760   | rear   | four  | 209    | 182        | sedan  |
| jaguar  | 36,000   | rear   | two   | 326    | 262        | sedan  |

Figure 10: A session in BayesDB for probabilistic model building and search in the cars dataset (Kibler et al., 1989).