What is the Population of Interest?

Population Modeling for BayesDB

Richard Tibbetts
tibbetts@mit.edu

Vikash Mansinghka
vkm@mit.edu

BayesDB [1, 2] is a probabilistic programming platform that enables users to solve probabilistic data analysis problems using a simple, SQL-like language. Queries execute against *generative population models* (GPMs), a new abstraction that can be used to integrate data, metadata, qualitative domain knowledge, and quantitative models. Baseline quantitative models are typically built via an AI modeling assistant then refined by end users. A key challenge in using BayesDB is designing the template, or *population schema*, that defines the conceptual population of interest.

The issues in population design are broadly analogous to relational database schema design, with additional concerns for statistical validity, modeling, and inference queries. Basic population modeling starts with considerations like:

1. Defining entities and variables. The core of a generative population model is the set of real-world entities and variables of interest. These reflect a trade-off between the available data and the questions of intrinsic interest, even if those questions refer to entities and/or variables that cannot be observed.

2. Integrating computationally derived variables. For example, financial data about companies often includes functions of one or more fundamental indicators, such as price-to-earnings ratios. If these are not explicitly identified, then these relationships will be re-inferred from the data, which may be wasteful and inaccurate, but also more robust.

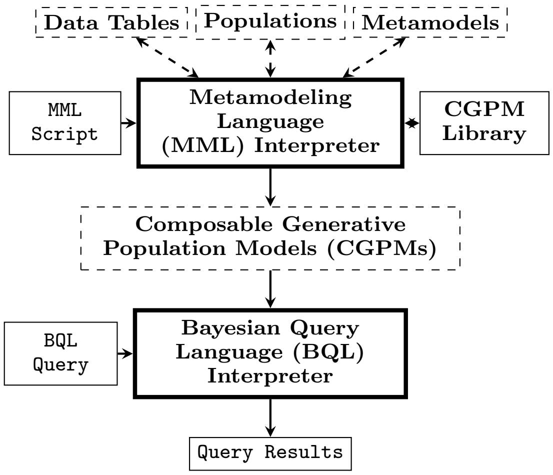
3. Choosing statistical types for all variables. This includes information about allowable values, but also the arbitrariness of the labels used to distinguish so-called "nominal" data; the significance of ordering for "ordinal" data; and the geometry of "numerical" data.

Fundamental issues remain open. Examples include (i) how to integrate data from random samples with data from "convenience" samples; (ii) the relationship between a "base" population and others that can be defined by segmenting it (e.g. by distinct values of some variable) or by aggregating it (e.g. by univariate statistics such as the sum, max, or median); and (iii) how to "join" or "merge" two populations and reconcile the quantitative models associated with each one. Each issue touches on both statistical inference and data management. Addressing these challenges will require the creation of new data modeling patterns.

References

[1] Vikash Mansinghka et al. "BayesDB: A probabilistic programming system for querying the probable implications of data". In: *arXiv preprint arXiv:1512.05006* (2015).

[2] Feras Saad and Vikash Mansinghka. "Probabilistic Data Analysis with Probabilistic Programming". In: *arXiv preprint arXiv:1608.05347* (2016).


Data Tables
Populations
Metamodels
MML Script
Metamodeling Language (MML) Interpreter
CGPM Library
Composable Generative Population Models (CGPMs)
BQL Query
Bayesian Query Language (BQL) Interpreter
Query Results

```
%mml CREATE TABLE t FROM "customers.csv"
%mml CREATE POPULATION p FOR t(
....   GUESS STATTYPES FOR (*);
....   MODEL age AS MAGNITUDE
.... );

%mml CREATE METAMODEL m FOR p
....  WITH BASELINE crosscat(
....   SET CATEGORY MODEL
....     FOR age TO lognormal;
....   OVERRIDE GENERATIVE MODEL
....     FOR income GIVEN age, state
....     USING linear_regression
.... );

%mml INITIALIZE 4 MODELS FOR m;
%mml ANALYZE m FOR 1 MINUTE;

%bql SIMULATE age, state
....   GIVEN income = 145000
....   FROM p LIMIT 3;
```

age	state	income
29	CA	145000
61	TX	145000
48	MA	145000

Figure 1: System architecture and modules that comprise BayesDB. The MML interpreter reads schemas to define variables and statistical types, metamodel definitions to apply automatic and custom modeling strategies for groups of variables in the population. BQL is a model-independent probabilistic query language to `ESTIMATE` properties of GPMs such as dependence relationships between variables, similarity between members, and conditional density queries, and `SIMULATE` missing or hypothetical observations subject to ad hoc constraints. Together, these components allow users to build population models and query the probable implications of their data.