Poster: Bayesian Synthesis of Probabilistic Programs for Automatic Data Modeling*

POPL 2019

Dylan Wilson
dgwilson@ncsu.edu

ABSTRACT

In this poster we present the key aspects of an automated data modeling regime detailed in [1]. This paper shows how to represent data models as programs and uses Bayesian inference to search over the possible programs in a custom Domain Specific Languages (DSL) resulting in an ensemble of programs used to model and predict the data. They present two such DSLs, one for univariate time series and a second for multivariate tabular data. They illustrate the performance of the approach on a wide variety of data sets and show its superiority to a set of existing techniques.

1 INTRODUCTION

Given a set of data you wish to model, standard practice involves manually selecting and testing candidate models and their parameters until a candidate solution is found that satisfies the investigator's needs. This paper contributes a new technique where the model and it's parameters are automatically found without specifying any information about the model. Using a customized Domain Specific Language (DSL) that leverages a set of basic building blocks and a Bayesian inference process, the paper demonstrates a probabilistic approach to generate a seed program in the DSL, and describes a Bayesian inference process to refine both the program's structure and parameters. The final output for any dataset being modeled is a set of programs with a high probability of having generated the data. Thus, as an ensemble, this set of programs on average accurately model and also predict future data. In addition, the programs can be characterized by the presence or absence (and probability) of various syntactic features available in the DSL to create an explanation of the model and what likely characteristics are inherent in the data. They demonstrate the power of their approach on both univariate and tabular data, showing how it outperforms many standard methods on a wide variety of data sets.

2 METHODOLOGY

The authors define a DSL for each of the two problem domains. One for univariate time-series analysis and one for multivariate tabular data. For each data set to be modeled they generate 60 random programs using a Venture Probabilistic program that samples from the space of all programs representable in each DSL. With these in hand, they turn to Metropolis Hastings search to improve each program with respect to the likelihood each program generated the training data. They alternate search steps between program (structural) modifications and parameter modifications. The result is a set of programs that are drawn from the posterior distribution of programs that generated the data and as a set, provide a good model for the data.

In an affirmative nod to explainable models, they also show how to post-process the final program(s) to characterize the underlying nature of the data being modeled. In the univariate time series data, they can directly express whether the data had a linear trend, periodicity, a change point or white noise. In their multivariate example, by analyzing the synthesized programs, they are able to characterize whether any two variables had a predictive relationship and demonstrated success under a wide variety of possible relationships. Specifically, when a complex relationship (e.g. linear+bimodal) existed between two variables, Bayesian synthesis outperformed Pearson correlation by a wide margin and was not subject to false alarm in four of the reported variable pairs.

```
['+',
  ['*',
    ['+', ['WN',49.5], ['C',250.9]],
    ['+', ['PER', 13.2, 8.6],
      ['+',
        ['LIN', 1.2],
        ['LIN', 4.9]]]],
  ['WN', 0.1]]
```

Figure 1: Example Univariate Program in Custom DSL

3 RESULTS

The authors provide a wide variety of interesting results. For univariate data, they demonstrate the structure prediction

*https://dl.acm.org/citation.cfm?doid=3302515.3290350

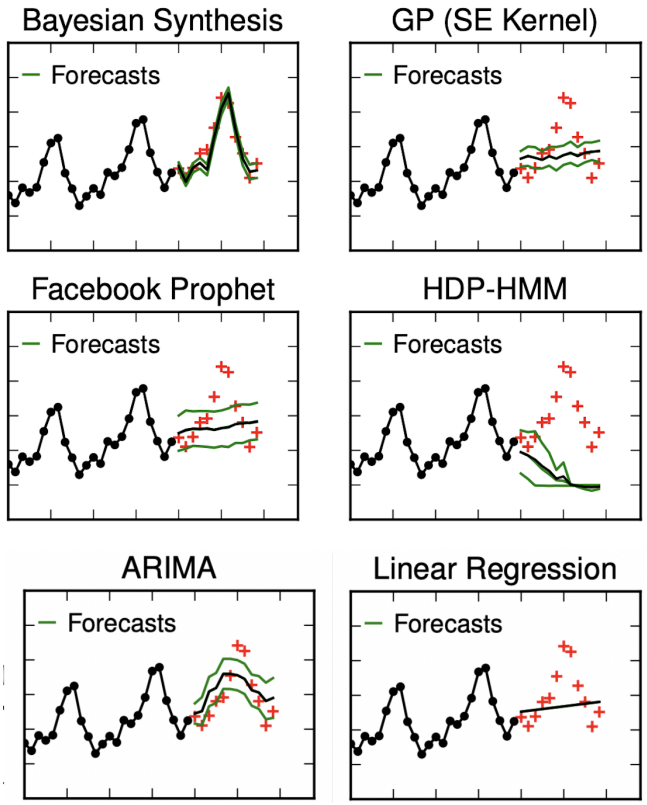
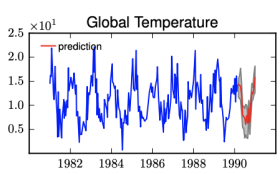


Figure 2: Predictive Performance of Bayesian Synthesis vs other methods on univariate time series data.

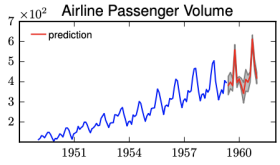
in Figure 3 and predictive output for six different datasets in Figure 2. They then compare their method on seven data sets to five other commonly used methods. In all but two datasets, the Bayesian Synthesis method beat the other methods in root mean square forecasting error for held out data show. They then used the time-series data to discuss and plot the runtime vs predictive likelihood for six different datasets. In their analysis, they observe that runtime is dependent on both number of observations and complexity of the data. Data complexity causes runtimes to grow faster than number of observations, as the length of the sampled programs increases in size to capture the complexity of the underlying data.

For multivariate data, they demonstrate the ability of Bayesian synthesis to capture pairwise predictive relationships across sixteen different pairs with a wide variety of underlying relationships. Bayesian synthesis performs much better than Pearson correlation in this head-to-head comparison. The include a series of scatter plots showing the various relationships that Bayesian Synthesis was able to identify. Finally, they compare Kernel Density Estimation (KDE) to their technique across 13 different "benchmark" datasets. Their method handily beat KDE on all, in some cases by orders of magnitude.



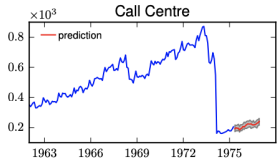
Temporal Structure	p^{synth}
✓ White Noise	100%
× Linear Trend	16%
✓ Periodicity	92%
× Change Point	4%

(a)



Temporal Structure	p^{synth}
✓ White Noise	100%
✓ Linear Trend	95%
✓ Periodicity	95%
× Change Point	22%

(c)



Temporal Structure	p^{synth}
✓ White Noise	100%
✓ Linear Trend	93%
✓ Periodicity	97%
✓ Change Point	90%

(e)

Figure 3: Temporal structure examples from Bayesian Synthesis method. Probability of each structure is show in the table.

4 CONTRIBUTIONS

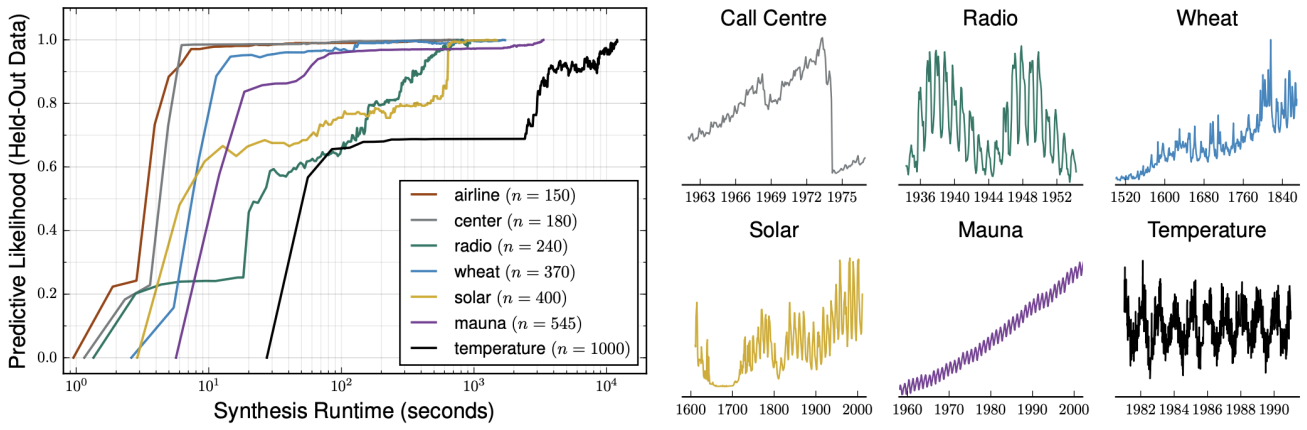
This paper claims to be the first fully Bayesian synthesis of probabilistic programs. A key contribution is the definition and formalization of the necessary conditions to ensure that the system is well-formed and that convergence is “almost surely” guaranteed. They also provide concrete examples including a DSL and transition rules for both univariate time-series data and multivariate tabular data. In addition, they demonstrate a method to post-process the resultant programs to identify the probability of various structural features in the dataset. They also describe the method by which they convert from their custom DSL to Venture to allow for the Synthesis process to be carried out and evaluated in a generic fashion. Finally, the provide an executable download to allow readers to use or extend the system.

REFERENCES

[1] Feras A. Saad, Marco F. Cusumano-Towner, Ulrich Schaechtle, Martin C. Rinard, and Vikash K. Mansinghka. 2019. Bayesian synthesis of probabilistic programs for automatic data modeling. *Proceedings of the ACM on Programming Languages* 3, POPL (Jan 2, 2019), 1–32. http://dl.acm.org/citation.cfm?id=3290350

Variable 1	Variable 2	True Predictive Structure		Predictive Relationship Detected By Pearson Correlation		Predictive Relationship Detected By Bayesian Synthesis	
(a) flavanoids	color-intensity	linear + bimodal	✓	×	(0.03)	✓	(0.97)
(b) A02	A07	linear + heteroskedastic	✓	×	(0.16)	✓	(0.89)
(c) A02	A03	linear + bimodal + heteroskedastic	✓	×	(0.03)	×	(0.66)
(d) proline	od280-of-wines	nonlinear + missing regime	✓	×	(0.09)	✓	(0.97)
(e) compression-ratio	aspiration	mean shift	✓	×	(0.07)	✓	(0.98)
(f) age	income	different group tails	✓	×	(0.06)	✓	(0.90)
(g) age	varices	scale shift	✓	×	(0.00)	✓	(0.90)
(h) capital-gain	income	different group tails	✓	×	(0.05)	×	(0.77)
(i) city-mpg	highway-mpg	linearly increasing	✓	✓	(0.95)	✓	(1.00)
(j) horsepower	highway-mpg	linearly decreasing	✓	✓	(0.65)	✓	(1.00)
(k) education-years	education-level	different group means	✓	✓	(1.00)	✓	(1.00)
(l) compression-ratio	fuel-type	different group means	✓	✓	(0.97)	✓	(0.98)
(m) cholesterol	max-heart-rate	none (+ outliers)	×	×	(0.00)	×	(0.08)
(n) cholesterol	st-depression	none (+ outliers)	×	×	(0.00)	×	(0.00)
(o) blood-pressure	sex	none	×	×	(0.01)	×	(0.26)
(p) st-depression	electrocardiography	none	×	×	(0.04)	×	(0.00)

Figure 4: Pairwise Predictions: Bayesian Synthesis vs. Pearson Correlation.



(a) Runtime vs. Accuracy Profiles (b) Time Series Datasets

Figure 5: Runtime vs. accuracy across six datasets.