Discovery of Hidden Variables Using Probabilistic Methods

Morgan Bryant (mrbryant@stanford.edu)
576 Alvarado Row
Department of Psychology
Stanford, CA 94305 USA

Jungmin Cho (joshcho@stanford.edu)
Storey House, 544 Lasuen Mall
Department of Psychology
Stanford, CA 94305 USA

Abstract

In our project for Stanford course Psych 204, we have designed a probabilistic framework for improving other probabilistic programs. The project can be described as an automated hypothesis tester for discovery of hidden variables/factors for a given program. We attempt to answer the following: is such a program tractable, and can generated models improve results?

Keywords: hypothesis testing, probabilistic models, WebPPL

Background

Previous work includes various methods for discovery of latent factors or learning the structure of a Bayesian network[1]. They identify three main methods of action: (1) constraint-based approaches, (2) score-based methods, and (3) ensembles. This project cites the second class of methods. Besides this work, we heavily leverage the Psych 204 coursework, the web-book Probabilistic Models of Cognition, and specifically work in Bayesian data analysis and statistics[2]. Our work falls in the realm of unsupervised learning, data analysis, and model selection.

Overview

Overall, we have designed a method within the WebPPL framework for implicitly improving probabilistic graphs. Our objective tasks are:

1. Baseline: First we present a solution to the benchmark researching-children-at-museum task, in which two independent researchers' results are combined in a way that is unlikely considering the data, in WebPPL. We use the results of the approximate hypothesis-testing inference to select a model among the proposed candidates and provide confidences over various alternative hypotheses. This task demonstrates the capability of a fully-Bayesian program for inferring about a model.

2. We generalized the baseline framework to consider other kinds of data, and we present the results at the end of the paper.

3. Hypothesis Class: Here we present a discussion around the properties of the hypothesis class, and how it interacts with our data.

4. Extend the system to various applied tasks. Extension (a) is a Lossy Recreation task that considers how well our system can estimate a model given data samples from a hidden model, and extension (b), an Encoding Compression task, tests the ability to recreate data using fewer variables than the model that generated such data.

See here[3] for a digital copy of the code.

Baseline

Our baseline WebPPL program examines two hypotheses on a given task inspired by "Posterior prediction and model checking"[4]:

> Imagine you're a developmental psychologist, piloting a two-alternative forced choice task on young children. (Just for fun, let's pretend it's a helping study, where the child either chooses to help or not help a confederate in need.) You have two research assistants that you send to two different preschools to collect data. You got your first batch of data back today: For one of your research assistants, 10 out of 10 children tested helped the confederate in need. For the other research assistant, 0 out of 10 children tested helped.

In this scenario, we first consider the original hypothesis H0 as P(child helps) depends only on the child, for which the optimal inference is 0.5 for a single parameter. Next, we instead consider the (sole) alternative hypothesis HA where P(child helps) depends on both the child and the specific

[1] Koller, 2009
[2] Goodman, 2016
[3] https://github.com/taoketao/Probabilistic-Variable-Discovery
[4] Goodman, 2016

experiment, where the optimal inferred parameters are just above 0 and just below 1 for two different, independently-chosen parameters.

Each hypothesized model is identified with optimal parameters for the given data using a standard Infer, run using Markov Chain Monte Carlo, with 30,000 samples.

Results

Let the notation [k1,k2] drawn from (n1,n2) represent an observation of two n-Binomial data, where each k is the number of 'true' observations in a trial of size n, denoted N when all the n values are the same. Over the single data point [0,10] for N=10, we see that the HA model with two free parameters can better explain the data than H0 model, with confidence above 0.99. This intuitively makes sense. In our hypothesis encoding, this represents the HA=(0,1) is much more likely than H0=(0,0), representing that the corresponding data points were drawn from the unique, independent variables labeled 0 or 1. Our data yielded this result:

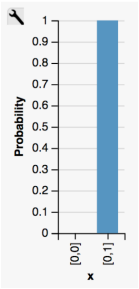

Figure 1: The probability that [0,10] was drawn from the same (0,0) or different (0,1) variables.

We also ran the program on other data, and other notable results are:

a) Data [5,5] yields that model H0 has probability about 0.57 and HA has 0.43. This follows the intuition around the Bayes Occam's Razor: while both models have all optimal parameters at 0.5, the original hypothesis is simpler and thus a better 'fit' compared to the overfit alternative model. Similar probabilities are found for data (2:10, 2:10).

b) For our tests on n1 = n2 = 10, The most indecisive data are: (0,1), (2,4), and (3,5).

c) The following is a heatmap of P(H0) per data pairs, where red indicates higher P(H0):

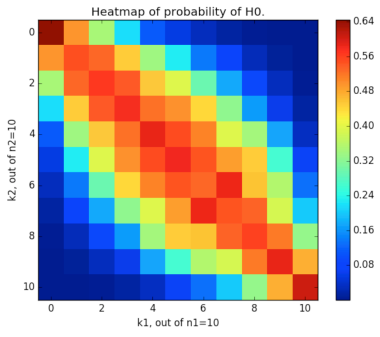

Figure 2: Heatmap

Consider all possible data that takes on values of [0,10]^2, we see that the original hypothesis H0 = (0,0) = the data were drawn from the same variable is strong along the axis k1=k2 and strongest at the corners k1=k2=0 or k1=k2=10, compared to the single alternative hypothesis HA = (0,1) = the data came from different variables.

Hypothesis Class Size

In our baseline system, we were able to exhaustively consider each hypothesis since there were only two reasonable options. As a general framework, these hypotheses represent the N=2 realization of the Unique Ordered Set Equivalence problem, where (0,0) and (0,1) where digits represent which sets are equivalent to each other as an encoding. In the context, these represent whether two or more datapoints are drawn from the same distribution. As N increases, the number of sets that must be considered increases in O(n^2), and specifically follows the pattern of Bell Exponential Numbers; their curve is indicated below:

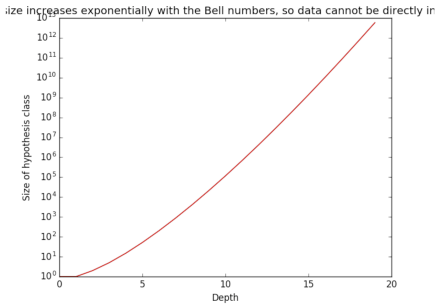

Figure 3: Depth vs. Hypothesis Class Size

As the depth of computation increases as the number N of data points increases, the number of possible hypotheses grows at N^2. At least it's not exponential!

Several values of the Bell numbers are 1, 2, 5, 15, 203, 877, 4140, and 21147 for N=1 to N=8.

As such, the ability to consider all hypotheses becomes considerably more difficult to either generate and consider: in another examination, we ran a study where we fixed the amount of compute and increased the size of the hypothesis class by increasing N, the number of dimensions of the data that we were considering. The result of this study confirms that the probable mass of the full hypothesis class that we can examine decreases rapidly as N grows:

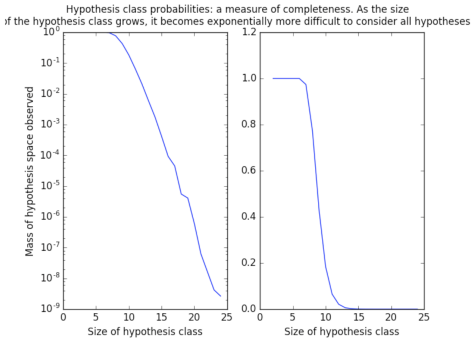

Figure 4: Hypothesis Class Probabilities

This problem is relevant because our ability to enumerate all hypotheses that might explain the data for large N becomes infeasible. For a child-psychologist modification that considers 10 groups of 10 children, the enumerative program takes nearly eight minutes to compute. One solution we tried was to infer hypotheses (using the Markov-Chain Monte Carlo or MCMC method) that were appropriate, in a necessary 'outer' inference for hypotheses wrapping the 'inner' inferences that compute the fitness of given hypothesized models to the data. We were able to decrease the runtime by about a factor of two with this sampling method before the hypothesized models deviated by more than 30% of the ideal labels.

Our ability to generate samples brings another issue: in order to generate unique hypotheses, we had to make a scheme that ensures uniqueness, since for two variables for example, (0,1) is effectively the same as (1,0). Our scheme for generating all hypotheses implicitly achieved such a scheme by ensuring that later-indexed digits for a hypothesis were no greater than one larger than any previous scheme; for example, for N=5, the scheme (0,1,0,0,2) would be valid while (0,2,3,4,0) would not. Notable traits are that the first digit is always zero and the last digit is the only slot that might take on the highest value of N-1.

Our initial trial simply sampled hypotheses by generating sequential digits by random uniform selection. Unfortunately, we realized that this scheme results in a skewed distribution for final digits, even with MCMC; sequences that start with zeros are considerably more likely (by a factor of O(n^2): see Bell's Triangle) than sequences that do not. This results in hypothesis class samples that are unable to consider data with more variation in the beginning of their data. We also figured that a normalization is definitely possible but that its implementation would require an algorithm that can generate Bell numbers and would be rather memory-intensive, as the probabilities of all the alternative selections at any given point in the sequence would need to be known. To verify, we generated the expected digit for a given index of a sequence, and we can see that the average value is monotonically increasing and increasingly favorable towards more 0s than higher digits.

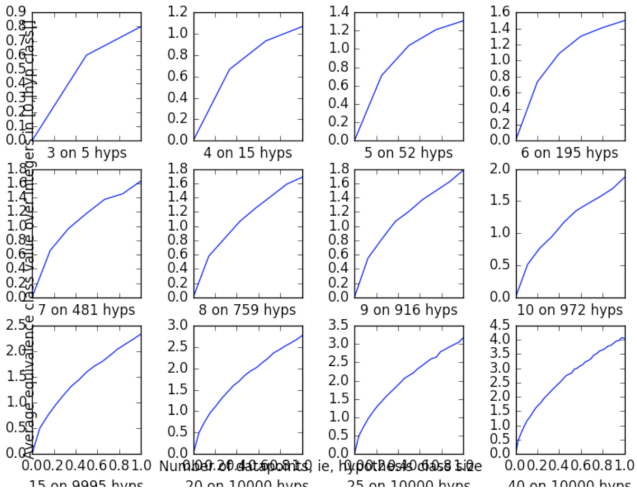

Figure 5: Expected Value of Sampled Hypotheses

Alternatively, this hypothesis-sample distribution actually lends well to a different interpretation. Since the average sampled hypothesis yields a determined distribution that favors more grouped elements at the start of the sequence, a valid preprocessing technique that clusters similar values at the start of a sequence would appropriately bias the data in a way that this system would generate centered, fitting hypotheses. Any clustering scheme that simply reorders the order of the sequential data points would be sufficient. Appropriate cluster sizes would set to be continually decreasing; for example, let the sequential clusters be roughly of size (½, ¼, ⅛, …) or even faster-decreasing and our algorithm should perform roughly well.

In Extension 4(a), where we consider a dataset generated from a sampled hypothesis and attempt to estimate the original hypothesis from that data, the hypotheses drawn from the hypothesis class distribution were not normalized, drawn instead from this biased distribution favoring many smaller digits earlier in the sequence. We make note that this can be understood as a preprocessed dataset that was indeed clustered in the scheme mentioned above. The sampled hypotheses that are later considered are drawn from this same distribution, and as such, can be considered relatively unbiased, regardless of the size of N.

Extension (a): Lossy Model Recreation

As an extension of our system to an applied task, we consider the problem of identifying an unknown model, knowing only data drawn from it.

In this experiment, we generated a hypothesis, then sample a small number of datapoints from it, and finally infer a reasonable hypothesis to explain that data. To make the task more difficult, we injected Gaussian noise into the generated data and we made no restriction on the repeatability of data values (eg, H=(0,1) may generate (5,5) as likely as any other pair).

We provide results for a test where N=4, the values in consideration lay within [0,6], and the number of sampled

data points was 20. We see that sometimes the system was able to recognize the correct original hypothesis, sometimes was able to promote the correct one but was less able to make a confident decision, and sometimes was surprisingly incorrect. We are unsure why the test was not as effective as imagined, and we hypothesize that increasing the number of values that the data can take (larger than 6) on would improve the system's ability to differentiate between models. We believe this mainly because the system was able to more reliably come up with the correct original hypothesis when more data values lay near the edge of the the dataset, i.e. just above 0 or just below 6.

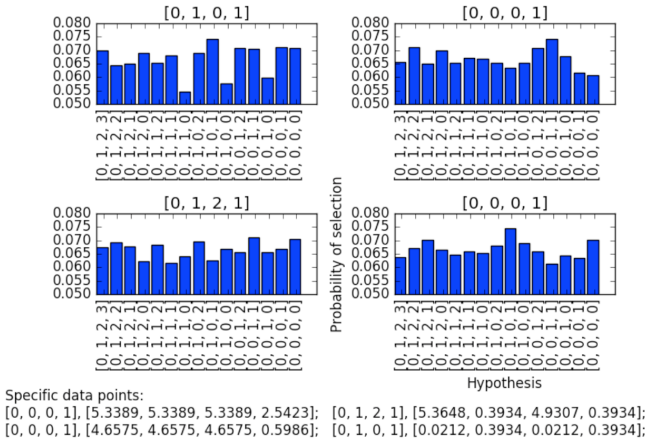

Figure 6: Hypothesis Selection

Extension (b): Encoding Compression

As another extension to our system, we consider the problem of restricting the maximum number of variables allowed, choosing the best hypothesis, and generating data from that hypothesis.

We modified the hypothesis-generation process by putting an upper bound to the number of variables. For instance, given data of four experiments, and two variables, we would generate [0, 0, 0, 0], [0, 0, 0, 1], [0, 0, 1, 0], [0, 0, 1, 1], [0, 1, 0, 0], [0, 1, 0, 1], [0, 1, 1, 0], and [0, 1, 1, 1]. We can view this restriction as a prior, or as a means of testing how much this restriction would alter our ability to generate data.

Based on the new hypothesis-generation process, we generate a distribution of hypotheses. Then we select the best hypothesis from the distribution (e.g. [0, 0, 1, 1] for experiment data [1/10, 2/10, 9/10, 10/10]).

Then we based on this hypothesis, we generate the data. We take the average of all experiment data attached to that variable — in the given example, you would have variable 0 map to 3/20, and variable 1 map to 19/20. Then we sample from that generated distribution for each experiment data.

The following is an example with data of 8 experiments, and recreating that data with fixed number of variables. The original experiment data had k values of [0, 0, 9, 8, 4, 1, 1, 10], and n values of [10,10,10,10,10,10,10,10], as shown in Figure 7.

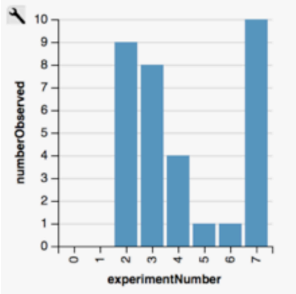

Figure 7: Original Experiment Data

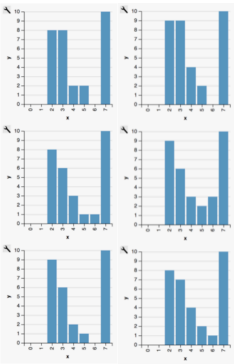

Figure 8: Recreated Data with 8 Variables

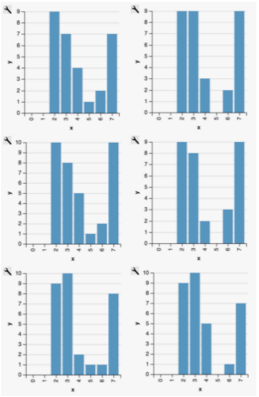

Figure 9: Recreated Data with 4 Variables

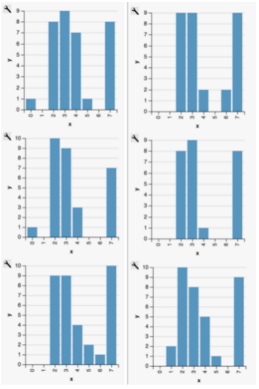

Figure 10: Recreated Data with 3 Variables

From here on out (in Figure 11 and 12), you can see that restricting the number of variables has a significant impact on the data we generate, as the shape of the sampled data

deviates heavily from the original experiment data. This might be a threshold at which restriction (and simplicity) does not pay off.

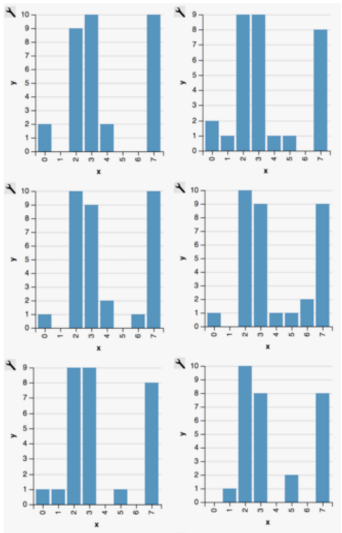

Figure 11: Recreated Data with 2 Variables

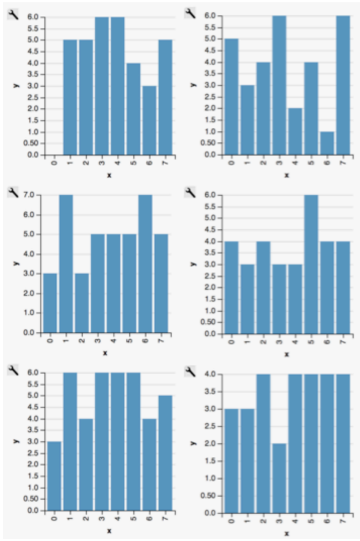

Figure 12: Recreated Data with 1 Variable

Conclusion

In this study, we are excited to have demonstrated the power of probabilistic programming. We demonstrated that the difficult tasks of unsupervised hidden variable discovery, model selection, and diverse data analysis can be effectively solved using probabilistic inference in a relatively concise framework. We are excited about the prospect that this model lends itself to self-reference, as it was fundamentally a probabilistic program that analyzes other probabilistic programs. We are enthralled by the accuracy of some of the results. We observed long compute times, but we are observed that the numerous hyper-parameters can be tweaked for individual problems to yield better performance. We showed that our system can extend to various other problems, given weak preprocessing conditions. We also showed that our system can eliminate variables, and see if the data generated from a more constrained model resembles the original data.

Use standard APA citation format. Citations within the text should include the author's last name and year. If the authors' names are included in the sentence, place only the year in parentheses, as in Newell and Simon (1972), but otherwise place the entire reference in parentheses with the authors and year separated by a comma (Newell & Simon, 1972). List multiple references alphabetically and separate them by semicolons (Chalnick & Billman, 1988; Newell & Simon, 1972). Use the "et al." construction only after listing all the authors to a publication in an earlier reference and for citations with four or more authors.

Additional Results

Besides the original motivating example of two researchers observing [0/10] and [10/10], we were able to find comparable results that match intuition well. These are presented below. In these graphs, as elsewhere, the horizontal axis points out the various hypotheses mapped to their inferred probabilities, for given K datasets drawn from integers up to N. We observe, as sounds reasonable, that the system can reliably weight hypotheses, expresses an Occam's Razor effect for preference for simpler models when appropriate, and finds that two pairs of datapoints with the same distances between them are more likely to be drawn from the same variable if they were centered nearer the center of the data space, around N/2. For example, in [0,2,4,6], the system is much more likely to think that 2 and 4 were from the same variable than 0 and 2 were.

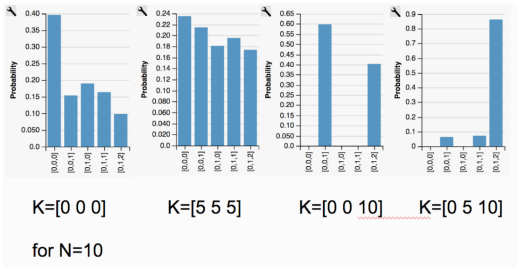

Figure 13: 3-Experiment Hypothesis Testing

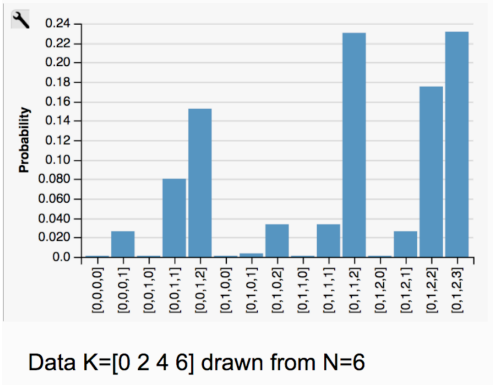

Figure 14: 4-Experiment Hypothesis Testing

References

Koller, Daphne, and Nir Friedman. Probabilistic graphical models: principles and techniques. MIT press, 2009.

OEIS Foundation Inc. (2011), The On-Line Encyclopedia of Integer Sequences, http://oeis.org.

N. D. Goodman and J. B. Tenenbaum (2016). Probabilistic Models of Cognition (2nd ed.). Retrieved 2016-12-14 from http://probmods.org/v2

Written December 2016.