Building an interface between probabilistic programming languages and lumen

Jonas Aaron Gütter
Friedrich Schiller Universität Jena
Matrikelnr 152127
Prof.Dr. Joachim Giesen
M. Sc. Phillip Lucas

20. September 2018

Zusammenfassung

(1) grober Fahrplan

- read related material:
  - understand what Probabilistic Programming Languages (PPLs) are
  - understand main idea of Lumen and what we want to do with it
  - understand and formulate "why & what"
- choose a PPL
  - work out requirements to chose PPL
  - work out preferred (but not necessarily required) features
  - chose a PPL based on these requirements and preferences
- get started with PPL
  - play around, learn how to use it, what it can do, etc also confirm identified 'pain point', i.e. understand and formulate what problem you are trying to solve, why this is relevant and outline how you plan to solve that pain point
- design a wrapper of chose PPL for Backend of Lumen
- give presentation about work so far, its justification, relevance, verification ideas, etc etc
- implement and test wrapper
- evaluate implementation in terms of goals set in beginning

(2) interessante PPLs:

- stan for python: https://pystan.readthedocs.io/en/latest/
- pymc3: https://docs.pymc.io/notebooks/getting_started.html#Case-study-2:-Coal-mining-disasters

- edward: http://edwardlib.org/getting-started
- pyro: http://pyro.ai/

Wegen eines Arbeitsplatzes und eines PCs erkundigen wir uns. Als Anmelde und Starttermin für Deine MA halten wir Mitte September (Mo, 17. September) im Auge.

Inhaltsverzeichnis

1 Rules of Probabilistic Inference

There are three rules of probabilistic inference: The chain rule, the total probability rule, and the Bayes' rule. The following explanations are taken from [?].

1.1 Chain rule

The chain rule is used to calculate a joint probability distribution of several variables from local conditional probability distributions of these variables:

$$P(X_1, X_2, ...X_n) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)...P(X_n|X_1, X_2, ...X_{n-1})) \tag{1}$$

1.2 Total probability rule

The total probability rule calculates the probability distribution over a subset of variables, also called a marginal distribution, by summing out all the other variables, that is by summing the probability distributions for each combination of values of these variables:

$$P(\boldsymbol{X}|\boldsymbol{Z}) = \sum_{\boldsymbol{y}} P(\boldsymbol{X}, \boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{Z}) \tag{2}$$

1.3 Bayes' rule

Bayes' rule calculates the probability of a cause, given aneffect, by using the prior probability of the cause and the probability of the effect, given the cause. The Bayes' rule can be derived from the chain rule.

$$P(X|Y) = (P(Y|X) * P(X))/P(Y) \tag{3}$$

2 Bayesian models

A Bayesian model is about assigning probabilities to model parameters. The model parameters β themselves are considered as random variables which depend on hyperparameters α.
According to [?], the likelihood of a new datapoint can be calculated by integrating the product of the prior likelihood and conditional probability of β over

the space of β, as shown in equation 4. This formula can also be derived using the chain rule (I think).

$$p(x_{new}|\boldsymbol{x}, \alpha) = \int p(x_{new}|\beta) * p(\beta|\boldsymbol{x}, \alpha) d\beta \tag{4}$$

3 What is Probabilistic Programming

Modelle spezifizieren/beschreiben

[?]

effizienter in der Beschreibung von Modellen als herkömmliche Programmiersprachen [?]

unifying general purpose programming with probabilistic modeling [?]

A probabilistic reasoning system uses prior knowledge in the form of a probabilistic model to answer a certain query. The particular propertes of the query as well as the prior knowledge are given to an inference algorithm which returns an answer in the form of probabilities. Example is shown in figure 1. Probabilistic Programming is the implementation of a probabilistic reasoning system by using a programming language.

Traditional means for representing models are not always sufficient for probabilistic models. Therefore, probabilistic programming languages were introduced to be able to represent models with the full power of a programming language (http://www.probabilistic-programming.org/wiki/Home).

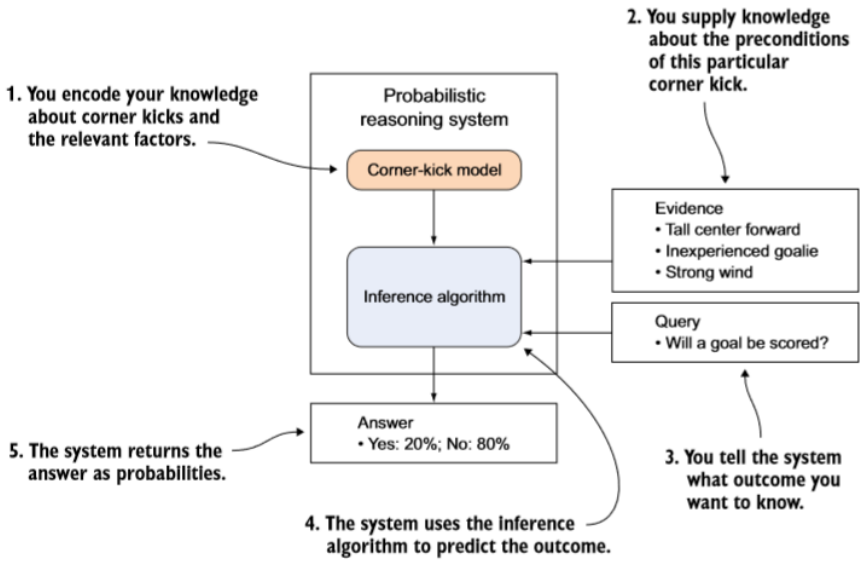

Abbildung 1: General workflow example of a probabilistic reasoning system

4 Existing PPLs

4.1 Stan for python

4.2 Pymc3

python library

fit Bayesian models, including Markov Chai Monte Carlo (MCMC) and variational inference (VI)

4.3 Edward

4.4 Pyro

5 Lumen

Abbildungsverzeichnis

6 Literatur

Literatur