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

23. August 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.

1 What are Probabilistic Programming Languages

Modelle spezifizieren/beschreiben

[1]

effizienter in der Beschreibung von Modellen als herkömmliche Programmiersprachen [2]

unifying general purpose programming with probabilistic modeling [3]

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

judgement call

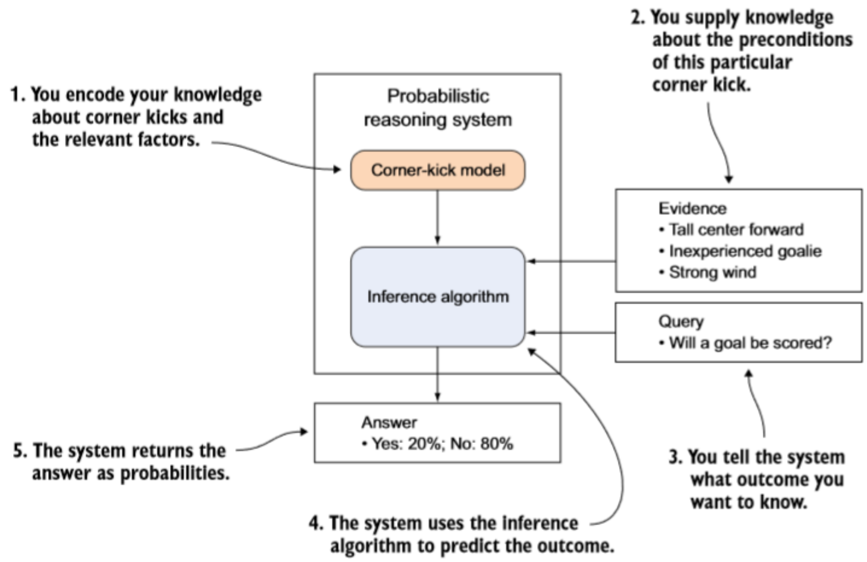
1. You encode your knowledge about corner kicks and the relevant factors.
Probabilistic reasoning system
Corner-kick model
Inference algorithm
2. You supply knowledge about the preconditions of this particular corner kick.
Evidence
• Tall center forward
• Inexperienced goalie
• Strong wind
Query
• Will a goal be scored?
3. You tell the system what outcome you want to know.
Answer
• Yes: 20%; No: 80%
5. The system returns the answer as probabilities.
4. The system uses the inference algorithm to predict the outcome.

Abbildung 1: General workflow example of a probabilistic reasoning system

2 Description of existing PPLs

2.1 Stan for python

2.2 Pymc3

python library

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

2.3 Edward

2.4 Pyro

3 Lumen

Abbildungsverzeichnis

4 Literatur

Literatur

[1] Wikipedia contributors, “Probabilistic programming language — Wikipedia, the free encyclopedia,” 2018. [Online; accessed 23-August-2018].

[2] L. Hardesty, “Probabilistic programming does in 50 lines of code what used to take thousands,” Apr. 2015. [Online; accessed 23-August-2018].

[3] “probabilistic-programming.org.” [Online; accessed 23-August-2018].

[4] A. Pfeffer, *Practical Probabilistic Programming*. Manning Publications, 2016.