Experiences with using Probabilistic Programming for Voucher Feature Extraction at Skanned.com

Project Status Report

Ahmad Salim Al-Sibahi
ahmad@bilagscan.dk

January 2019

1 Background

Probabilistic Programming is an emergent field of machine learning, that enriches general programming frameworks with probabilistic constructs from Bayesian reasoning. The core idea is that one specifies probabilistic models that describe the anticipated distribution of target data, and then these frameworks provide an automated way to learn parameters of the model when new data is observed.

There are three key advantages to probabilistic programming over existing machine learning technology: it is possible to directly incorporate *domain knowledge* using prior distributions, the constructed models allow for a systematic way to *quantify uncertainty*, and they are often directly *interpretable* by humans. All of these aspects are important for voucher information extraction, which is the core business of Skanned.com. Vouchers are almost always structured documents, where key information is explicitly labelled and where there are many legal rules about how information should be presented; the use of this domain knowledge is key to achieving good results, which is why the existing system relies heavily on hand-tuned heuristics. Quantifying uncertainty of a result is an important way for the system to specify its trust in the results it provides and make sure that customers only pay what is necessary: it is important for customers that when the system states that the total amount is "$1000", it is the right amount and not "$100" or "$10000". Skanned.com provides a human-based validation service, but such service is expensive and is a bottleneck with regards to scalability; it is therefore important to only rely on it is known to be necessary, which is not possible to do in a systematic way with the current system. Finally, if an error happens during information extraction, it is important that the system is easy to debug and explain to customers, which is hard to do for purely deep neural network-based architectures with millions of nuisance parameters.

The goal of the current Industrial PostDoc is to examine how to apply probabilistic programming in practice for voucher scanning systems. This is important both for the company, where getting a good solution can result in significant savings and make the company a leading expert in machine learning, and for science in general, since there are not many existing practical applications of probabilistic programming and new experiences provide an opportunity for improving the existing systems.

The PostDoc is mentored on the academic side by Dr. Thomas Hamelryck, who is an expert in Bayesian data analysis and its applications in Bioinformatics, and Dr. Fritz Henglein, who is an expert in programming languages and high-performance compilation to GPUs. On the business side, the mentor is Dan Rose Johansen, who is the Chief Operational Officer at BilagScan, and leads the daily operations.

2 Achievements

2.1 Knowledge Building

During my PostDoc at Skanned.com I have worked towards understanding how to apply probabilistic programming in practice, and sharing such knowledge amongst colleagues and fellow academics. Concretely:

- I have developed an understanding for Bayesian data analysis, including constructing probabilistic models, presenting them and evaluating them.

- I have gotten familiar with the wide range of inference techniques available for inference in probabilistic programs, gaining an understanding of their core theory and for the problems where they are applicable.
- I have shared my knowledge about various probabilistic programming frameworks with my colleagues, and discussed how they can potentially be applied in practice.
- I am a part of a weekly probabilistic programming discussion group at University of Copenhagen, where they are actively looking into how to use modern probabilistic programming frameworks for solving problems in academia in general.

2.2 Application

I have examined how we can use probabilistic programming frameworks for voucher information extraction, and developed various models aimed towards such goal.

- I have developed a model for grouping vouchers based on textual and visual features[1], that relies on latent Dirichlet allocation (LDA), which is a Bayesian model that identifies a common set of "topics" that summarize a set of documents. The grouping is useful for developing specialized algorithms for similar sets of vouchers and thus increase precision in the information extraction. This model was found useful by Skanned.com and is therefore planned to soon come in production.
- I have developed a series of probabilistic models from scratch for keyword-based feature localization. The problem is challenging to encode because of the varying number of keywords and features between each document, but initial results were promising in showing its potential use in practice: 80% of the time the target feature was the expected one, and 99% of the time it was within the 95% confidence interval.
- I am currently working with a more ambitious model that tries to assigns keywords to features explicitly, to make inference more precise. The model can also be extended to allow locating more features in the future, as well as possibly identify potential keywords.
- I have developed experience with various probabilistic programming frameworks like PyMC3, Pyro and Infer.Net, and their underlying technologies like Theano and PyTorch. This included getting experiences with neural network architectures that can potentially be used in the models or for inference. Furthermore, I have actively submitted bug reports, bug fixes and features to these languages.

2.3 Dissemination

I have tried to share my experiences using Probabilistic Programming as part of my Industrial PostDoc at Skanned.com, in order to generate excitement for the area. This is also a good branding opportunity for Skanned.com to position themselves as a cutting edge startup in artificial intelligence (AI).

- I have presented our work at the first international conference on probabilistic programming (PROBPROG 2018), which was held at Massachusetts Institute of Technology

[1] In collaboration with my mentors Fritz and Thomas

(MIT). This provided opportunity to learn about the latest technology within probabilistic programming by world's top universities and major IT companies (Microsoft/-Facebook/Google/Uber/Amazon/Oracle etc.)

- I have presented our plans for probabilistic programming at a public event organized by Skanned.com on Machine Learning and Artificial Intelligence (called Masters of AI/ML 2018). I was rated as one of the top speakers at this event by the audience, which was around 200 people.

- I have started a meet-up on probabilistic programming, to provide more focused technical discussions on the area. The meet-up is a collaboration between Skanned.com, University of Copenhagen and Hypefactors, and has already after one session over 75 members, which is great for such new techology. We already have scheduled multiple sessions and in talk to have exciting speakers from academia and industry (e.g., Facebook, Nordea) in the future.

In general, it should be mentioned that probabilistic programming as a technology is getting a lot of excitement. This is also felt by my mentors Thomas and Fritz, when they present and discuss this work with other academics and people from industry.

3 Challenges

Being a first mover in a new field of technology like probabilistic programming has its clear rewards, but also comes with its own set of challenges, both anticipated and unexpected. I will here describe some of the challenges encountered during the project and how I strived to solve them.

3.1 Theoretical Challenges

Today, there are few applications of general probabilistic programming in industry, and none we know of within the area of voucher processing. While the development of probabilistic programming framework is also done by research arms of industrial companies, most papers are written towards academics using academic language. There is still a reasonable gap between theory and practice, which we as part of this project hope to fill.

- Probabilistic Programming intersects many areas of theoretical and applied mathematics. Knowledge is required at least in the fields of probability theory, advanced calculus and programming language semantics (including category theory), and sometimes even areas like statistics, information theory, measure theory and real analysis. Some of these areas I had to refresh my knowledge in—I had not made heavy use of calculus or probability theory since high-school—and some I had to learn from scratch, like basic measure theory.

- The promise of probabilistic programming, where the user provides a model and where inference is fully automatic is perhaps not quite achieved. It seems certainly easier to do Bayesian modeling now than few years back, but knowledge of the particular inference methods is still necessary to construct models that work well. For example, sampling methods require good geometric ergodicity[2], variational methods require

[2]Distributions should not have heavy tails or complex modes.

estimators with low variance, and expectation propagation and similar algorithms require priors to be conjugate.

- Working with mixed discrete-continuous models is still a challenge from the side of inference. Discrete models make differentiation-based methods harder to use; those algorithms that rely on both differentiation and discrete methods work only well for small number of discrete choices, and those that do not rely on differentiation only work with fixed sets of continuous distributions.

- The maturity of existing probabilistic programming framework is still low. Frameworks like Anglican or Venture still seem to be at the academic stage, while more industry targeted frameworks like PyMC3, Pyro and Infer.Net still have limitations for more complex models. During the project, I had to debug [3] and profile [4] the frameworks themselves to identify bugs and limitations, discuss the issues with the framework developers and contribute code [5] to fix some of these issues.

- The performance and scalability of current probabilistic frameworks still needs improvement. We have so far not been able to train with more than around 1000 vouchers in reasonable time for more complex models, even though the available data set is orders of magnitude larger. Even though most framework support running models on the GPU, it is hard to achieve good performance improvements and sometimes the overhead is larger than running on CPUs directly.

3.2 Practical Issues

In the beginning of the project there were some practical issues at the University about the time of getting the budgeted research machine, which have now been solved. However, these issues made it a non-negligible impact on the project and caused some delays.

4 Future Directions

During this project there have been many challenges that have been overcome and many achievements that have been made. For the future, there are a few things which we want to work on:

- Finish the keyword assignment-based algorithm and publish the current results (including grouping) in a suitable conference (e.g. ECML PKDD). This might require

[3] See "Getting auto-enumeration working with transformed distributions" (`https://forum.pyro.ai/t/getting-auto-enumeration-working-with-transformed-distributions/440a`), "NaN occured in optimization" (`https://discourse.pymc.io/t/nan-occured-in-optimization-in-a-vonmises-mixture-model/1296`), "[bug] –jit option fails for HMM example" (`https://github.com/uber/pyro/issues/1435`), "pm.sample_prior_predictive fails when model contains a mixture-based distribution" (`https://github.com/pymc-devs/pymc3/issues/3101`), and "Updated mixture model breaks with multi-dimension mixtures" (`https://github.com/pymc-devs/pymc3/issues/3044`)

[4] See "Profiling TraceEnumELBO" (`https://forum.pyro.ai/t/invalid-index-in-gather-for-model-relying-on-auto-enumeration/479/5`)

[5] See "Working on component-wise transformations that mimic torch.cat and torch.stack" (`https://github.com/pytorch/pytorch/pull/11868`), "Add documentation for transform functions" (`https://github.com/pymc-devs/pymc3/pull/3192`), "Add site information for possible exception thrown when calculating logp" (`https://github.com/uber/pyro/pull/1509`) and "Clarify phrasing of installation instructions with regards to running examples" (`https://github.com/uber/pyro/pull/1533`)

some time to get working optimally with inference, since the model is complex and has many local optima, but can possibly be extended with more directions in the future.

  - Alternatively, work on extending the keyword-based localization that provided initially good results with more features to increase accuracy; this can provide a faster road to currently useful results, but is less extensible in the future.

- Tackle new problem in voucher processing with probabilistic programming. A potential problem is product line extraction, where the current algorithm of Skanned.com lacks behind.

- Work with the Futhark group at University of Copenhagen to see how we can compile probabilistic programming efficiently to the GPU. This includes looking into hybrid techniques for inference, based on e.g. those used for Augur by Oracle, and compiled sequential inference sampling. The hope is that this can allow scaling the capabilities of probabilistic programming systems, and e.g. allow more vouchers to be processed efficiently for Skanned.com