Optimal decision-making under constraints and uncertainty
Latour, A.L.D.

Abstract

Scientists, policy makers and individuals must take decisions while dealing with uncertainty on a daily basis. Often-times, there also are limitations on the number or kind of decisions we can take. Especially when the stakes are high, we want to take an optimal decision: one that has, in expectation, the best possible outcome, according to some predefined metric of success.

Uncertainty can, for example, stem from randomisation in algorithms or human behaviour. Even if you follow someone on Twitter, does their post show up on your timeline? And if it does, do you read it? Alternatively, uncertainty may, for example, originate in the randomness of nature. When an earthquake strikes, which power lines in an electric grid survive this disaster? And how many households lose power as a result? In any application, uncertainty can stem from the practical problem that some things are hard to accurately measure or quantify.

We must also deal with constraints. In the realm of marketing, there is a limit on how many people we can reach with a single ad campaign or newspaper article. In the context of maintenance, there is a limit on how much money we can spend on making our electric grid strong and resilient.

The examples above all involve relationships between entities. People may have a (mutual) follower relationship with each other on social media. Power grids connect power stations to households. Because of the uncertainty in these examples, we can view these relationships as *probabilistic relational data*, which we

can model using *probabilistic networks*.

In this work, we develop new methods for exact optimal decision-making under constraints and uncertainty, specifically for problems that involve some form of probabilistic relational data. These *stochastic constraint (optimisation) problems (SCPs)* are hard to solve. Uncertainty about the future means that we must consider many different, and possibly overlapping, scenarios, to assess the effects of decisions. In addition, the number of potential strategies (sets of decisions) that we can adopt for solving the problem can grow exponentially with the number of individual decisions in the problem. In general, SCPs are $\mathcal{NP}$-hard problems.

We aim to design solving tools that are convenient to use, and thus accessible to people with limited programming skills, general enough to support a wide range of SCPs, and fast enough to be practical. In order to achieve these goals, we propose to build on technology from the probabilistic inference literature, and the *constraint programming (CP)* and *mixed integer programming (MIP)* literature.

In particular, we present a new probabilistic logic programming language, SC-ProbLog, especially designed for modelling SCPs formulated on probabilistic networks, and based on ProbLog. We show how to formulate constraints on the probability distributions induced by these networks, and how to model these constraints such that they can be solved by a CP or MIP solver. Here, we make use of existing tools to create *decision diagram (DD)* representations of probability distributions, which support tractable probabilistic inference. In doing so, we identify specific properties of these DDs that we can exploit to speed up the inference process, and thus the SCP solving process.

We take a modular approach to building our SCP solving pipelines. This ensures that, for each subtask, we can use any relevant state-of-the-art tools, which helps to keep the pipeline up-to-date with the latest developments, without having to (re)implement the latest techniques ourselves or having to integrate them into a monolithic design.

We do this following the paradigm of *programming by optimisation (PbO)*, implementing alternative design choices for different parts of the pipeline. The resulting pipelines are highly configurable, meaning that choosing the right components, and the right parameters for those components, for a specific type of problem may be difficult. We therefore use *automated algorithm configuration (AAC)*, to not only evaluate the performances of the SCP solving pipelines in a fair manner, but also find which parameter settings work well for problems from specific application domains.

The work presented in this dissertation advances the state of the art in SCP solving by proposing a new programming language to model SCPs, demonstrating how off-the-shelf CP and MIP technology can be leveraged for fast SCP solving, and by proposing an efficient propagation algorithm for stochastic constraints. We take a PbO-based approach, being, to the best of our knowledge, the first to do so in the field of exact probabilistic inference. We demonstrate the effectiveness of our methods on problems that are known in the data mining literature, including spread-of-influence problems and *frequent itemset mining (FIM)* problems.