ARTICLE IN PRESS
ARTINT:2918

Artificial Intelligence ••• (••••) •••–•••

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Empirical decision model learning

Michele Lombardi [a,*], Michela Milano [a], Andrea Bartolini [b]

[a] DISI, University of Bologna, Italy
[b] DEI, University of Bologna, Italy

ARTICLE INFO

Article history:
Received in revised form 21 December 2015
Accepted 10 January 2016
Available online xxxx

Keywords:
Combinatorial optimization
Machine learning
Complex systems
Local search
Constraint programming
Mixed integer non-linear programming
SAT modulo theories
Artificial neural networks
Decision trees

ABSTRACT

One of the biggest challenges in the design of real-world decision support systems is coming up with a good combinatorial optimization model. Often enough, accurate predictive models (e.g. simulators) can be devised, but they are too complex or too slow to be employed in combinatorial optimization.

In this paper, we propose a methodology called Empirical Model Learning (EML) that relies on Machine Learning for obtaining *components* of a prescriptive model, using data either extracted from a predictive model or harvested from a real system. In a way, *EML can be considered as a technique to merge predictive and prescriptive analytics*.

All models introduce some form of approximation. Citing G.E.P. Box [1] "Essentially, all models are wrong, but some of them are useful". In EML, models are useful if they provide adequate accuracy, and if they can be *effectively exploited by solvers for finding high-quality solutions*.

We show how to ground EML on a case study of thermal-aware workload dispatching. We use two learning methods, namely Artificial Neural Networks and Decision Trees and we show how to encapsulate the learned model in a number of optimization techniques, namely Local Search, Constraint Programming, Mixed Integer Non-Linear Programming and SAT Modulo Theories. We demonstrate the effectiveness of the EML approach by comparing our results with those obtained using expert-designed models.

1. Introduction

Advances in Combinatorial Optimization methods in the last decades have enabled their successful application to a broad range of industrial problems. Many of such approaches rely on the availability of some declarative system description. This typically consists of a hand-crafted mathematical model, obtained after thorough discussion with the domain experts by introducing some simplifying assumptions.

Devising a good model is a complex task, especially challenging when dealing with real-world systems. A good model finds a proper balance between model complexity and model accuracy: on the one hand, excessive simplification may lead to "optimal" – but completely useless – solutions. On the other hand, incorporating too many details results in extremely hard computational issues. Despite this, a number of successful optimization approaches have been proposed in the literature and

* Corresponding author.
E-mail addresses: michele.lombardi2@unibo.it (M. Lombardi), michela.milano@unibo.it (M. Milano), a.bartolini@unibo.it (A. Bartolini).

http://dx.doi.org/10.1016/j.artint.2016.01.005
0004-3702/

applied to real-life industrial problems, enabling in many cases[1] huge savings in terms of resources (time, money, machines, energy).

Nevertheless, many systems are still impervious to approaches such as Mixed Integer Linear Programming (MILP), Constraint Programming (CP), or SAT (propositional SATisfiability) and this is often due to modeling issues. There are basically two kinds of "high-complexity systems" that are out-of-reach for traditional combinatorial approaches: (1) Complex Systems, which exhibit phenomena that emerge from a collection of interacting objects capable of self-organization and affected by memory or feedback; and (2) physical systems whose dynamic model is known, but its embedding in a combinatorial model is computationally intractable.

A very common way for supporting decision-making in these systems is to design a predictive model (e.g., a simulator) based on real data and to use it via what-if analysis (see [2] for a recent reference). In what-if analysis, the decision maker repeatedly feeds scenarios (i.e. sets of decisions) to the predictive model to extract the values of certain observables of interest (e.g. quality measures). Inevitably, only a limited number of scenarios is investigated, and then the decision maker commits to the one showing the best behavior. In combinatorial problems the decision space might be so large that selecting scenarios manually or in isolation results in far-from-optimal choices.

The aim of this paper is to bring such high-complexity systems within the reach of combinatorial decision making and optimization. The idea is to use Machine Learning (ML) to learn an approximate relation between decisions and their impact on the system. In particular, we devise a methodology, called *Empirical Model Learning (EML)* that: (1) learns relations between decidables and observables[2] from data, and (2) encapsulates these relations into components of an optimization model, namely objective functions or constraints. The training data for the learning techniques can be harvested from the real system or extracted from a predictive model (e.g. simulator). The integration into model components is not merely a matter of encoding, since in some cases an operational semantics for the efficient use of the component should be defined.

The ability to integrate Machine Learning models in combinatorial optimization has the potential to play a major role in *bridging the gap between predictive and prescriptive analytics*. An EML based system may be capable of suggesting optimal decisions in a complex real-world setting, by taking advantage of recent developments in big data analysis and predictive model design.

This paper provides three main contributions. First, we introduce the Empirical Model Learning approach in a general fashion. Second, we present a number of methods for embedding Machine Learning models (namely Decision Trees and Artificial Neural Networks) into several optimization techniques (Local Search, Mixed Integer Non-Linear Programming, Constraint Programming, SAT Modulo Theories). Some of our embedding techniques have been presented in previous papers of ours [4,5]. Third, we show that despite the main idea behind EML being very simple, its application requires some care for obtaining an effective optimization approach. We highlight the main difficulties and suggest possible solutions by applying the EML approach on two practical examples.

As motivating (and running) examples, we use two thermal-aware workload dispatching problems, defined over an experimental multicore Intel CPU called "Single-chip Cloud Computer" (SCC, see [6]). Both problems consist in mapping a set of heterogeneous jobs on the platform cores so as to maximize some cost metric involving the platform efficiency. The efficiency of each core is affected by a number of complex factors including the thermal dynamics of the chip, the workload distribution, and the presence of low-level schedulers and thermal controllers. Although an accurate system simulator for the platform is available, it cannot be inserted into a decision model due to its high complexity and large run time. We show that EML allows considerable improvements over simpler optimization approaches either based on expert-designed heuristics, or on expert-designed models refined via function fitting.

The paper is structured as follows: in Section 2 we provide a comparative analysis of related work. In Section 3 we introduce the example problems. In Section 4 we give a brief overview of the EML approach. Section 5 presents techniques for embedding Machine Learning models into Combinatorial Optimization models. Sections 6 and 7 discuss respectively how to design the *core combinatorial structure* of the optimization problem, and how to extract a system model from data: in both cases, our example problems are employed to present the process. We provide experimental results in Section 8 and concluding remarks in Section 9.

2. Comparative analysis of related work

The EML approach combines elements of Combinatorial Optimization, Machine Learning, and Complex Systems/Simulation. In this section we provide a brief overview of approaches related to the integration of such research fields.

Loosely related approaches Researchers have been interested for a long time in the integration of optimization techniques in Machine Learning. This is not surprising, given that training problems are fundamentally (very peculiar) optimization problems. Works such as [7,8] have studied the core optimization problems in ML algorithms and proposed efficient methods

[1] The reader may find some examples on the web page dedicated to the Franz Edelman Award at https://www.informs.org/Recognize-Excellence/Franz-Edelman-Award.

[2] The names decidables and observables have been suggested by Peter Flach [3].