Learning User Preferences in Interactive Constraint Programming

Mohamed Siala

LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France
msiala@laas.fr

Abstract

In real-world applications of prescriptive decision making, it is often the case that the decision maker is unable to provide a closed form of the objective function. However, they are able to express preferences over the possible solutions [Bessiere et al., 2017, Toffano et al., 2022]. In this paper, we assume that a decision problem is given without an explicit objective function. The framework that we consider is an interactive environment where the user is iteratively asked to rank certain solutions. This iterative process is bounded by a number of solutions and a total number of iterations. When the iterative process stops, the user picks one of the best solutions proposed in the aforementioned process. The challenge that we address in this paper is to to achieve the best possible approximation of the real objective function under limited resources based on the decision maker's preferences.

keywords Constraint Programming, Machine Learning, Preferences

1 Problem Definition

Let P be a constraint satisfaction problem (CSP) that is associated to an unknown objective function f. We denote by S the set of solutions of P. Assume w.o.l.g that the purpose is to maximize f. In order to capture the notion of user preferences, we define the absolute order function $order : S \to \mathbb{N}^*$ as follows:

- For each $s, s' \in S$, $order(s) < order(s')$ iff $f(s) < f(s')$
- For each $s \in S$, if $order(s) > 1$, then $\exists s'$ such that $order(s') = order(s) - 1$

Finding a solution that maximises $f(.)$ is equivalent to find a solution that maximises (its unknown) $order(.)$. We say that the user prefers s to s' if $order(s) > order(s')$.

The problem that we consider is defined as follows. Given a CSP P with an unknown objective function, the aim is to find a solution that maximizes its absolute order via a limited number of interactions with the user. We assume that a solver is available as an oracle to solve the initial problem P as well as answering certain queries. The global framework is depicted below:

- At each iteration i, the solver is asked to generate k solutions. The user is subsequently asked to rank all generated solutions
- The user preferences are leveraged at each iteration in order to guide the solver in its search for new solutions. The way the preferences are handled is called a *strategy*.
- When the resources are exhausted (number of iterations, number of solutions, runtime, etc), the iterative process stops and the user is given the set of the best solutions found so far

The crux of the problem lies in designing efficient and problem independent strategies.

2 Proposed Strategy

Our approach is essentially based on the notion of active learning. The prediction task takes as input a solution and predicts its order. The set of solutions constitutes the dataset and each solution is labelled by its order. Unlike active learning, drawing an example from the dataset is computationally hard since it involves solving a combinatorial problem. The idea is to build prediction models at each iteration that are compatible with the current preferences then inject them into the solver in order to guide it towards better solutions.

References

[Bessiere et al., 2017] Bessiere, C., De Raedt, L., Guns, T., Kotthoff, L., Nanni, M., Nijssen, S., O'Sullivan, B., Paparrizou, A., Pedreschi, D., and Simonis, H. (2017). The inductive constraint programming loop. *IEEE Intelligent Systems*, 32(5):44–52.

[Toffano et al., 2022] Toffano, F., Garraffa, M., Lin, Y., Prestwich, S. D., Simonis, H., and Wilson, N. (2022). A multi-objective supplier selection framework based on user-preferences. *Ann. Oper. Res.*, 308(1):609–640.

Acknowledgement

This work is funded by the **"Centre International de Mathématiques et Informatique de Toulouse (CIMI)".** `https://www.cimi.univ-toulouse.fr/en/`