Learning and Intelligent Optimization (LION): One Ring to Rule Them All.

Mauro Brunato
Lionsolver Inc. and University of Trento
Via Sommarive 14
Povo, Trento, Italy
brunato@lionsolver.com

Roberto Battiti
Lionsolver Inc. and University of Trento
Via Sommarive 14
Povo, Trento, Italy
battiti@lionsolver.com

ABSTRACT

Almost by definition, optimization is a source of a tremendous power for automatically improving processes, decisions, products and services. But its potential is still largely unexploited in most real-world contexts. One of the main reasons blocking its widespread adoption is that standard optimization assumes the existence of a function $f(x)$ to be minimized, while in most real-world business contexts this function does not exist or is extremely difficult and costly to build by hand. *Machine learning* (ML) comes to the rescue: the function (the model) can be built by machine learning starting from abundant data. By *Learning and Intelligent Optimization* (LION) we mean this combination of learning from data and optimization which can be applied to complex, dynamic, stochastic contexts. This combination dramatically increases the automation level and puts more power directly in the hands of decision makers without resorting to intermediate layers of data scientists (LION has a huge potential for a self-service usage). Reaching this goal is a huge challenge and it will require research at the boundary between two areas, machine learning and optimization, which have been traditionally separated.

1. INTRODUCTION

One of the most common obstacles to widespread adoption of Business Intelligence and Predictive Analytics techniques, notably on big datasets, is the lack of human expertise in methodologically sound treatment of information. In particular, technically and scientifically skilled experts are very scarce. Machine Learning (ML) provides the most important algorithmic toolset for dealing with data and predictions; however, identifying the most effective ML technique requires much effort; moreover, most methods require an accurate parameter tuning phase to provide appropriate responses. Human experts are able, in most cases, to identify satisfactory solutions by theoretical or experience-based considerations; however, not many people have this kind of experience, and humans tend to have a cultural bias towards a small subset of techniques they are familiar with.

The long-term vision of our work is to develop a fully automated system so that only data and desired outputs need to be provided by the user [2]. The system will then start to determine the optimal choices by exploiting the vast amounts of cheap CPU power now available in networked (cloud) environments. We require the *anytime* property: after an initial setup period, the system will report better and better solutions as time elapses. It is up to the user to decide when to stop the system and use the best solution delivered up to that point: decision can be automated by setting a deadline or a budget. A practical usage of the system, useful to have a concrete situation in mind to guide the research, is as an automated "machine learning application builder," which will need to be fed with data and delivers a competitive application, comparable to or better than the best that can be created by expert human researchers.

2. SELF-TUNING BY LOCAL SEARCH

Software components implementing several ML methods (e.g. MLP, SVM) are available from many sources, and they can be readily applied and tuned by cross-validating them on a given dataset. We focus on optimizing the overall process by which, starting from data, an optimal solution is identified (including preprocessing/scaling, feature selection, choice of architecture, choice of critical parameters of the architecture).

Local search (LS) leading to locally optimal configurations is an effective building block for solving complex discrete and continuous optimization problems, and the local minima traps can be cured by *Reactive Search Optimization* (RSO [3, 1]). To denote a framework based on solving continuous optimization problem by a strategic use of memory and a cooperating team of self-adaptive local searchers we use the term CoRSO (Cooperating RSO). We underline that CoRSO is a methodology and not a single technique, so that different names have been used for specific techniques [4]. CoRSO represents a way of sharing the knowledge accumulated implicitly by an evolving population of members endowed with self-improvement capabilities. Each member takes care of a district (an input area), generates samples and decides when to fire local search in coordination with the other members. The CoRSO framework is motivated by the objective of sharing information accumulated by multiple local search streams, and considers direct and coordinated ways of strategically using the collected information to identify promising areas in the configuration space for local

Proceedings of the VLDB Endowment, Vol. 6, No. 11

search investigations. In particular, the objective is to coordinate a team of interacting solvers through an organized subdivision of the configuration space which is adapted in an online manner to the characteristics of the problem instance. A sociological/political paradigm can be adopted in the strategic allocation of computational resources to individual local search streams.

In many popular algorithms for continuous optimization one identifies a "local minimizer" that locates a local minimum by descending from a starting point, and a "global" component that is used to diversify the search and to reach the global optimum. We define the attraction basin of a local minimum x as the set of points that will lead to x when used as starting configurations for the local minimizer. In some cases, as we noted in our starting assumptions, an effective problem-specific local search component is available for the problem at hand, and one is therefore motivated to consider a hybrid strategy, whose local minimizer has the purpose of finding the local minimum with adequate precision, and whose combinatorial component has the duty of discovering promising attraction basins for the local minimizer to be activated.

The development of the CoRSO framework is guided by the following design principles.

General-purpose optimization: no requirements of differentiability or continuity are placed on the function f to be optimized.

Global optimization: while the local search component identifies a local optimum in a given attraction basin, the combinatorial component favors jumps between different basins, with a bias toward regions that plausibly contain good local optima.

Multi-scale search: the use of grids at different scales in a tree structure is used to spare CPU time in slowly-varying regions of the search space and to intensify the search in critical regions.

Simplicity, reaction and adaptation: the algorithmic structure of CoRSO is simple, the few parameters of the method are adapted in an automated way during the search, by using the information derived from memory. The intensification-diversification dilemma is solved by using intensification until there is evidence that diversification is needed (when too many districts are visited excessively often along the search trajectory). The tree-like discretization of the search space into districts is activated by evidence that the current district contains more than one attraction basin.

Tunable precision: the global optimum can be located with high precision both because of the local adaptation of the grid size and because of the decreasing sampling steps of the stochastic Local Searcher when it converges.

CoRSO is characterized by an efficient use of memory during the search, as advocated by the Reactive Search Optimization paradigm. In addition, simple adaptive (feedback) mechanisms are used to tune the space discretization, by growing a tree of search districts, and to adapt the main diversification parameter of RSO acting on prohibitions. This adaptation limits the amount of user intervention to the definition of an initial search region, by setting upper and lower

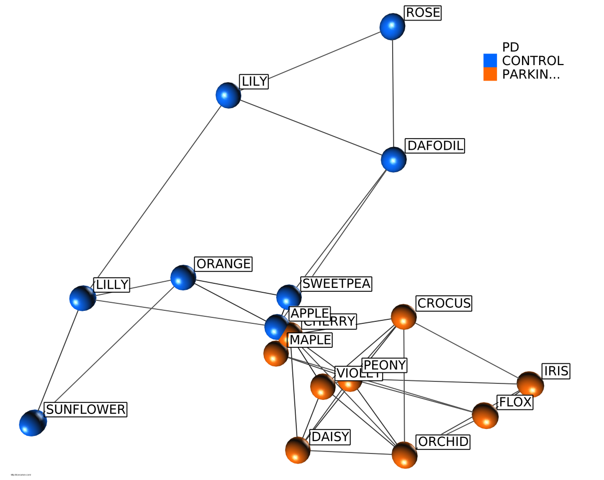

Figure 1: The LION approach for treatment of Parkinson's disease: clustering patients to optimize treatment.

bounds on each variable, no parameters need to be tuned. The CoRSO framework fuses Reactive Search Optimization with a problem-specific Local Search component.

3. LION IN HEALTHCARE

As an example of the LION "ML application builder" approach, a submission to the Michael J. Fox Foundation's Parkinson Data Challenge was awarded the first prize [5]. The application demonstrates that a ML approach is superior to conventional statistical methods for the detection, monitoring and management of Parkinson's disease. In spite of the very sparse data of this specific Parkinson's diagnosis problem, the final submission could predict the incidence and monitor the progression of the disease with an approximated 100% accuracy on the competition data. In addition to producing accurate detection, a ML approach paves the way for disruptive innovation in monitoring and managing the disease. A service for deriving insight from data by using the described approach is available at:
`http://onebutton.lionsolver.com/`

4. REFERENCES

[1] R. Battiti and M. Brunato. Reactive Search Optimization: Learning while Optimizing. *Handbook of Metaheuristics*, pages 543–571, 2010.

[2] R. Battiti and M. Brunato. *The LION Way: Machine Learning plus Intelligent Optimization.* Lionsolver Inc., USA, 2013.

[3] R. Battiti, M. Brunato, and F. Mascia. *Reactive Search and Intelligent Optimization.* Operations research/Computer Science Interfaces. Springer Verlag, 2008. ISBN: 978-0-387-09623-0.

[4] R. Battiti and G. Tecchiolli. The reactive tabu search. *ORSA Journal on Computing*, 6(2):126–140, 1994.

[5] The Michael J. Fox Foundation. "Machine Learning Approach" to Smartphone Data Garners $10,000 First Prize in The Michael J. Fox Foundation Parkinson's Data Challenge. `https://www.michaeljfox.org/foundation/publication-detail.html?id=473&category=7`.