Machine learning methods for parameter tuning in heuristics

Roberto Battiti,
Department of Computer Science and telecommunications, University of Trento, Via Sommarive 14, 38050 Povo (Trento) - Italy
http://rtm.science.unitn.it/~battiti

5th DIMACS Challenge Workshop: Experimental Methodology Day, Oct 30, 1996, Rutgers University

Abstract:

Parameter tuning is a crucial issue both in the scientific development and in the practical use of heuristics. In some cases the role of the user as an intelligent (learning) part makes the reproducibility of heuristic results difficult and the competitiveness of alternative techniques dependent in a crucial way on the user capabilities.

We argue that some simple subsymbolic machine learning methods can be profitably used in order to automate the tuning process and make it an integral (and fully documented) part of the algorithm.

If learning acts on-line, task-dependent local properties can be used by the algorithm to determine the appropriate balance between diversification and intensification. In this way a single algorithm maintains the flexibility to deal with related problems through an internal feedback loop that considers the previous history of the search.

The practical success of the approach in some cases raises the need of a sounder theoretical foundation of non-Markovian search techniques.

Introduction

Heuristics suffer from a bad reputation, citing from Papadimitriou and Steiglitz book [24]:

> *6. Heuristics* Any of the **<** five **>** approaches above without a formal guarantee of performance can be considered a ``heuristic.'' However *unsatisfactory mathematically*, such approaches are certainly valid in practical situations.

Unfortunately, because of the recent ``hardness of approximation'' results [28], the hope of finding approximation algorithms (with formal performance guarantees) must be abandoned for many relevant problems. We are condemned to live with heuristics for very long times and some effort is therefore required to make them more ``satisfactory,'' both from a theoretical and from a practical point of view.

The following points illustrate some crucial topics that must be addressed to place the research in heuristics on a more stable ground.

Parameter tuning and the role of the user

In many cases state-of-the-art (non-dominated) heuristics are characterized by a certain number of **choices and free parameters**, whose appropriate setting is a subject that raises issues of research methodology [5,16,22].

In some cases the parameters are tuned through a feedback loop that includes **the user as a crucial learning component**: depending on preliminary tests, some values are changed and different options are tested until acceptable results are obtained. The quality of results is not automatically transferred to different instances and the feedback loop can require a lengthy ``trial and error" process when the algorithm has to be tuned for a specific application.

In addition, scientific research demands **reproducibility**, and reproducibility is difficult to reach when the developer (user) becomes an integral part of the algorithm. Even if the good faith of the researcher is assumed, the appropriate documentation of the learning process is a difficult task: some seemingly obvious details can be omitted, unpromising avenues tried can be neglected in the published paper (although negative results can be in some cases as informative as positive results), assumptions about the expertise of the developer (user) during the tuning process are left implicit, etc.

Machine learning for automation and full documentation

Parameter tuning is a typical ``learning" process where experiments are designed in a focussed way, with the support of statistical estimation (parameter identification) tools.

Now, the Computer Science community is very proficient at describing processes so that they can be reproduced even by a (mechanical) computer: with algorithms, possibly realized in software. In particular, the Machine Learning community, with significant influx from Statistics, developed in the last decades a rich variety of ``design principles" that can be used to develop machine learning methods and algorithms.

It is therefore appropriate to consider whether some of these design principles can be profitably used in the area of parameter tuning for heuristics. The long-term goal is that of completely **eliminating the human intervention** in the tuning process. This does not imply higher unemployment rates in the CS community, on the contrary, the researcher is now loaded with a heavier task: the algorithm developer must aim at transferring its expertise into the algorithm itself, a task that requires the exhaustive description of the tuning phase in the algorithm.

Let us note that the algorithm ``complexity" will increase as a result of the process, but the price is worth paying if the two following objectives are reached:

- **Complete and unambiguous documentation**. The algorithm (and the research paper based on the algorithm) becomes self-contained and its quality can be judged independently from the designer or specific user. This requirement is particularly important from the scientific point of view, where objective evaluations are crucial. The recent introduction of software archives (possible related to scientific journals) further simplifies the test and **simple re-use** of heuristic algorithms.

- **Automation**. The time-consuming tuning phase is now substituted by an automated process. Let us note that only the final user will typically benefit from an automated tuning process. On the contrary, the algorithm designer faces a longer and harder development phase, with a possible preliminary phase of exploratory tests, followed by the above described exhaustive documentation of the tuning process when the algorithm is presented to the scientific community.

Although formal learning frameworks do exist in the CS community (notably, the PAC learning model [29,18]) one should not reach the conclusion that these models can be simply adapted to the new context. On the contrary, the theoretical framework of computational learning theory and machine learning is very different from that of heuristics. For example, the definition of a ``quality'' function against which the learning algorithm has to be judged is complex. In addition, the abundance of negative results in computational learning should warn about excessive hopes.

Nonetheless, as a first step, some of the **principles and methodology** used in machine learning can be used in an analogic fashion to develop ``learning heuristics.'' Some considerations follow in the next sections.

Learning as means for generalization

The purpose of learning is **generalization**. As an example, in the PAC model a number of random examples (with classification labels given by a **concept**) are extracted from a given probability distribution.

These examples are used in the training process to output a **hypothesis concept**, but the performance of the learning algorithm is judged by the error rate on **different** examples, although extracted from the same probability distribution.

The analogy with parameter training in heuristics is that, for a problem characterized by a certain probability distribution on instances, some of these instances are used for training but **different** instances are used to validate the algorithm.

Unfortunately this is not the typical case in the heuristics literature. In many cases the performance is judged on the tasks used to tune the algorithm parameters. This method is particularly dangerous if many parameter are present in the algorithm.

Occam's razor: complication must be motivated

Finding a **succinct hypothesis** consistent with the observed data is one technique used in machine learning: in some contexts the succinctness guarantees predictive power.

In a more general context, the *Occam's razor* principle indicates that overly complex scientific theories should be subjected to a simplifying knife. In the area of heuristics one should definitely aim at the simplest possible algorithm capable of producing competitive results. Unnecessary complications should be trimmed, the relevance of each parameter should be motivated by either theoretical or experimental results.

Learning by careful experimentation

Active learning algorithms should be considered for parameter tuning. For reasons of efficiency, the tuning phase should be focussed and appropriate ``questions'' should be formulated during the process.

In CLT, **query learning** where the learning algorithm can ask membership queries makes the difference between intractability and efficient learning for finite automata [4].

The tuning phase of the heuristic algorithm should formulate hypotheses to be tested by discriminating experiments.

Off-line tuning

At least two experimental setups can be imagined for parameter tuning in heuristics, that we could denote as **off-line** and **on-line** tuning.

In the first setup the algorithm runs with different parameter settings on different instances. The parameter setting does not change during the run and the results obtained are used only at the end of the run by the tuning process. Trivially, the best parameter setting could be selected at the end of the tuning phase as the setting that obtained the best results (with sufficient statistical backing). But more complex schemes could be investigated, where the results obtained in a preliminary series of runs are used to select additional experiments.

On-line tuning: use of task-dependent local properties

In the **on-line** setup, the tuning acts **while the algorithm runs** on a specific instance. In this way the algorithmic process is **history-sensitive**: preliminary results in the first phase of the run can be used to influence the subsequent phases.

The positive aspect is that now **task-dependent local properties** can be used in the tuning. In particular, different parameter settings can be appropriate for different tasks and, on a given tasks, different parameter setting can be appropriate in different regions of the search space.

For a concrete example in the area of local search with **prohibition-based diversification** [25,12,13,15], the possibility of tuning the *prohibition parameter* depending on the local structure of an attractor around a locally optimal point is an effective way to select an appropriate balance of intensification and diversification. The term **reactive search** is used in [7,6] to underline that an internal tuning scheme ``reacts'' to the occurence of certain events, like the repetition of previously visited configurations.

Wanted: a theory of history-sensitive heuristics

Randomized Markovian local search algorithm have enjoyed a long period of scientific and applicative excitement, in particular see the flourishing literature on Simulated Annealing [19,17]. SA generates a **Markov chain**: the successor of the current point is chosen stochastically, with a probability that does *not* depend on the previous history (**standard SA does not learn**). A consequence is that the ``trapping" of the search trajectory in an attractor cannot be avoided: the system has no memory and cannot detect that the search is localized. Incidentally, the often cited asymptotic convergence results of SA are unfortunately irrelevant for the application of SA to optimization. In fact, repeated local search [10], and even random search [9] has better asymptotic results. According to [1] ``approximating the asymptotic behavior of SA arbitrarily closely requires a number of transitions that for most problems is typically larger than the size of the solution space ... Thus, the SA algorithm is clearly unsuited for solving combinatorial optimization problems to optimality." Of course, SA can be used in practice with fast cooling schedules, but then the asymptotic results are not directly applicable [27].

History-sensitive techniques in local search contain an internal **feedback loop** that uses the information derived from the past history to influence the future behavior. In the cited prohibition-based diversification techniques one can, for example, decide to increase the diversification when configurations are encountered again along the search trajectory [7,6]. It is of interest that state-of-the-art versions of Simulated Annealing incorporate ``temporal memory" [11]. The non-Markovian property is a mixed blessing: it permits heuristic results that are much better in many cases, but makes the theoretical analysis of the algorithm difficult.

Therefore one has an unfortunate chasm: on one side there is an abundance of mathematical results derived from the theory of Markov processes, but their relevance to optimization is dubious, on the other side there is mounting evidence that simple **machine learning** or history-sensitive schemes can augment the performance of heuristics in a significant way, but the theoretical instruments to analyze history-sensitive heuristics are lacking.

The practical success of history-sensitive techniques should motivate new search streams in Mathematics and Computer Science for their theoretical foundation.

Related approaches

The issue of (machine) learning, designing experiments, selecting parameters, validating hypotheses, are at least as old as the scientific development and it is difficult to pinpoint a precise origin of these themes. In particular, Statistics provides a variety of techniques to derive models from experimental data. The aim of this paper was that of raising some issues not that of covering all aspects of a very wide field with many interconnections.

Related issues with emphasis on sub-symbolic learning are considered in the ``neural networks" area [2,8], while symbolic generate-and-test search methods are used in problem-solving processes in Artificial Intelligence [26]. Related approaches are also known as *metaheuristics* [23], i.e., heuristics about the design of heuristics, although precise borders between heuristics and meta-heuristics are sometimes fuzzy. In the area of machine learning for heuristics, see also the recent proposal of [14]. Last but not least, let us mention a possible connection with the recently expanding theory of *on-line algorithms*, where decisions are made based on partial on-line information [20] and with reinforcement learning methods [21].

References

1

E. H. L. Aarts, J.H.M. Korst, and P.J. Zwietering. Deterministic and randomized local search. In M. Mozer P. Smolensky and D. Rumelhart (Eds.), editors, Mathematical Perspectives on Neural Networks. Lawrence Erlbaum Publishers, Hillsdale, NJ, 1995, to appear.

2

Y. S. Abu-Mostafa. The Vapnik-Chervonenkis dimension: Information versus complexity in learning. *Neural Computation* **1(3)**:312--317 (1989).

3

A. V. Aho, D. S. Johnson, R. M. Karp, S. R. Kosaraju, C. C. McGeoch, C. H. Papadimitriou, and P. Pevzner. Theory of computing: Goals and directions. Technical Report Tech. Rep. CSE-93-03, University of Washington, March 1996. to appear in Computing Surveys.

4

D. Angluin. Learning regular sets from queries and counterexamples. *Information and Computation* **75(2)**:87--106 (1987).

5

R. S. Barr, B. L. Golden, J. P. Kelly, M. G. C. Resende, and W. Stewart, Designing and Reporting on Computational Experiments with Heuristic Methods, *Journal of Heuristics* **1(1)**:9--32 (1995).

6

R. Battiti and M. Protasi, Reactive local search for the Maximum Clique problem, Tech. Rept. TR-95-052, International Computer Science Institute, Berkeley, CA (1995).

7

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

8

E. Baum and D. Haussler. What size net gives valid generalization? *Neural Computation* **1(1)**:151--160 (1989).

9

T.-S. Chiang and Y. Chow, On the convergence rate of annealing processes. *SIAM Journal on Control and Optimization* 26(6):1455--1470 (1988).

10

A. G. Ferreira and J. Zerovnik, Bounding the probability of success of stochastic methods for global optimization. *Computer Math. Applic.* **25**:1--8 (1993).

11

B. Fox, Simulated Annealing: folklore, facts, and directions. In: *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing* (H. Niederreiter and P.J.-S. Shiue, eds.), Lecture Notes in Statistics, Springer-Verlag, (1995, to appear).

12

F. Glover, Tabu search - part I. *ORSA Journal on Computing* **1(3)**:190--260 (1989).

13

F. Glover, Tabu search - part II. *ORSA Journal on Computing* **2(1)**: 4--32 (1990).

14

M. K. Goldberg and D. L. Hollinger, Learning Better Algorithms for the Maximum Clique Problem. Tech. Report, Rensselaer Polytechnic Institute (Jan 1996).

15

P. Hansen and B. Jaumard, Algorithms for the maximum satisfiability problem, *Computing* **44**:279--303 (1990).

16

J. Hooker, Testing Heuristics: We Have It All Wrong, *Journal of Heuristics* **1(1)**:33--42 (1995).

17

D. S. Johnson, C. R. Aragon, L.A. McGeoch, and C. Schevon, Optimization by simulated annealing: an experimental evaluation; part II, graph coloring and number partitioning. *Operations Research* **39(3)**:378--406 (1991).

18

M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*, The MIT Press, Cambridge, Massachusetts (1994).

19

S. Kirkpatrick, C. D. Gelatt Jr. and M. P. Vecchi Optimization by simulated annealing. *Science* 220:671-680 (1983).

20

E. Koutsoupias and C. H. Papadimitriou, Beyond competitive analysis, 35th Annual Symposium on Foundations of Computer Science, 394--400, IEEE (1994).

21

L.-J. Lin, Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning Journal, Special Issue on Reinforcement Learning* 8:293-322 (1992).

22

C. C. McGeoch, Toward and experimental method for algorithm simulation. *INFORMS J. Computing* **8(1)** : 1--15 (1996).

23

Meta-heuristics: Theory and Applications, edited by I. H. Osman and J. P. Kelly, Kluwer, Boston (1996).

24

C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization, Algorithms and Complexity*. Prentice-Hall, NJ (1982).

25

K. Steiglitz and P. Weiner, Some improved algorithms for computer solution of the Traveling Salesman problem, in: *Proc. Sixth Allerton Conf. on Circuit and System Theory*, Urbana, Illinois (1968), 814--21.

26

B. S. Stewart, C.-F. Liaw, and C. C. White III, A bibliography of heuristic search research through 1992. *IEEE Trans. on Systems, Man, and Cybernetics* **24(2)**:268--293 (1994).

27

P. N. Strenski and S. Kirkpatrick, Analysis of Finite Length Annealing Schedules. *Algorithmica* 6:346--366 (1991).

28

Efficient Checking of Polynomials and Proofs and the Hardness of Approximation Problems. Ph. D. Thesis, Computer Science Division, University of California at Berkeley (1992).

29

L. G. Valiant, A theory of the learnable. *Communications of the ACM* **27(11)**:1134--1142 (1984).

Thu Oct 24 12:20:50 MET 1996