How Computers Discover How Computers Discover
(A mini-review of algorithmic meta-discovery)

Filip Železný

Czech Technical University in Prague
Technická 2, 166 27 Prague 6, Czech Republic
zelezny@fel.cvut.cz

Abstract. In traditional exerimental sciences one forms hypotheses explaining an empirical sample of some natural phenomenon. The hypothesis discovery process can in certain cases be automated using machine learning algorithms. I discuss some consequences of viewing scientific discovery, primarily computer based, itself as a natural phenomenon, whose empirical sample may be analyzed or hypothesized about. This viewpoint frames some recent findings about the statistical properties of computationally intensive discovery processes (phase transitions in complexity of learning, heavy-tailed runtime distributions of hypothesis search), empirical assessments of the Occam's razor principle, as well as the issue of meta learning (such as 'discovering how to discover best'). Here I review these points and conclude with a few speculative thoughts.

1 Introduction

Advances in the field of machine learning have made it possible to automate the traditional course of scientific discovery (Fig. 1, left) by computer algorithms able to propose hypotheses explaining a sample of empirical observations (Fig. 1, right). In this paper's context, the terms 'machine learning' and 'hypothesis discovery' coincide. Recent progress in 'closed-loop machine learning' even lead to the development of a *robot-scientist* in functional genomics (see eg. the paper [15] in Nature). The robot, besides searching for suitable first-order logic hypotheses about gene functions and interactions, also proposes and physically realizes in-vitro experiments for the purpose of hypothesis testing and their subsequent refinement. In a way, the human scientists who established the initial experimental setup are eliminated from the basic discovery loop and positioned into the mere role of meta scientists reasoning about the robot-scientist.

Here I leave the interesting point of physical experimentation by robots aside and rather focus on another aspect of computer based discovery. The massive scale on which various hypotheses can be inferred by computers can in many fields provide an empirical sample of discovery processes sufficient to be analyzed statistically on its own, just as other natural phenomena (Fig. 2 provides an illustration). Such analysis may clarify a great deal not only about computerized discovery, but potentially about experimental science in general.

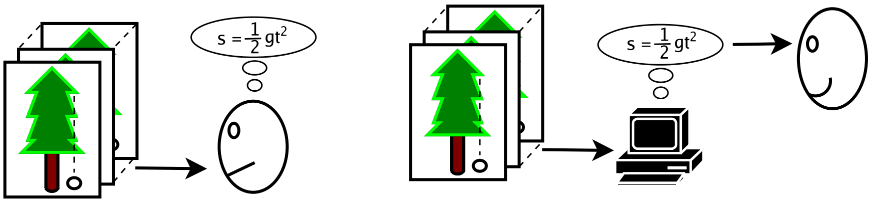

Fig. 1. Left: Traditional scientific discovery: a human forming a hypothesis explaining observations of some natural phenomena. **Right:** Computer-based scientific discovery, usually employing machine learning algorithms.

Recently, a stream emerged of research into the links between experimental philosophy of science on one hand and machine learning on the other [3, 17, 26]. In this review, though, I mostly seek to avoid the philosophical aspects of the junction and rather identify some of its fruits of instant practical impact. For example, findings originating from this 'meta' research include the detection of phase transitions in complexity of machine learning algorithms, their often peculiar runtime distributions as well as new insights into the usefulness of the popular Occam's razor principle.

Just as the basic discovery process can be computerized, so can be the meta discovery activity (Fig. 6). Here a meta learning algorithm proposes hypotheses about other learning algorithms. Practical implementations of meta learning have mainly focused on developing hypotheses predicting which learning algorithm would be 'most competent' for a given data sample. Although competency classification can be based on such mundane attributes as eg. the size of the sample and variable types involved therein, some more advanced meta learners take into account interesting information resulting from the already mentioned phase transition research.

In the following I review the phase transition and heavy-tailedness phenomena in machine learning (Sections 2 and 3), empirical assessments of the Occam's razor (Section 4) and briefly touch on the meta learning issue (Section 5). I conclude (Section 6) with a few speculative remarks concerning the concept of an infinite tower of meta learning processes and the possibility of translating results of the analysis of algorithmic discovery processes into ordinary scientific work.

2 Phase Transitions

Consider now the experimental setting exemplified in Fig. 2, where a human studies computer scientific discovery as an empirical phenomenon. In fact, statistical approaches to analyzing a sample of computer reasoning processes have originated in the artificial intelligence field of *problem solving*, an area more general than machine learning. The seminal work [16] investigated statistical properties of computational cost invested by a computer solving the NP-complete problem known as SAT, where one seeks a truth assignment to propositional

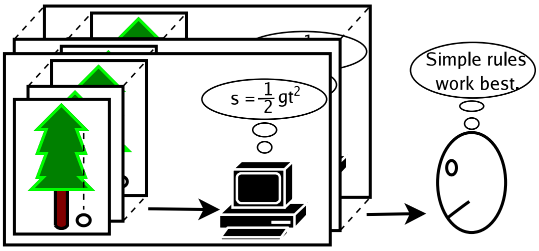

Fig. 2. A human viewing computer-based scientific discovery as an empirical phenomenon and inferring a hypothesis thereabout. Human scientific discovery (Fig. 1, left) could, in principle, be the subject of investigation as well (ie. be inside the boxes), but the researcher (outside the boxes) would likely suffer from insufficient volume of the empirical sample available for the study as well as low controllability of the experiments. The depicted experimental paradigm has lead to interesting discoveries about inductive learning processes, such as 'phase transitions' in computational complexity of hypothesis testing, heavy-tailed statistical distributions of hypothesis search run times, or some new insights into the usefulness of the Occam's razor principle; see text for details.

variables satisfying a set of propositional clauses. [16] considered a simple deterministic heuristic algorithm (Davis-Putnam) for the search, while the particular SAT problem instances were generated randomly. The experiment is reproduced in Fig. 3. The figure reveals an interesting property: problem instances requiring high computational cost are distributed very non-uniformly along the *constrainedness* parameter c, defined as the ratio between the number of clauses contained in the SAT instance and the number of propositional variables there involved. Indeed, most of the real hard instances are concentrated in a rather narrow neighborhood of $c = 4.5$. The strikingly uneven distribution of truly hard problem instances and the fact that indicators (such as the c value) derivable instantly from the instance description can be used to statistically estimate its expected hardness have had a large impact onto artificial intelligence research (see eg. the review [12]).

Interestingly, phase transition manifests itself as well in computer based scientific discovery. A family of advanced algorithms for hypothesis formation, collectively termed *inductive logic programming* (ILP), use the rich language of first-order predicate logic to express the proposed theories, in order to be able to capture complex, relational properties. This representational paradigm is for instance used by the robot-scientist mentioned in the Introduction. Another example–somewhat dated, yet well illustrative–is represented by the study [25] in the area of biochemistry. Here a learning algorithm is provided with a set of descriptions of chemical compounds. Each description is a set of *logic facts*, for example, the fact *atm(*$d1, d1_1, c, 22, -0.117$*)* represents that in the compound denoted as $d1$, there is a carbon (c) atom $d1_1$, of type 22 (which actually stands for "aromatic type"), with partial charge -0.117. Another example is

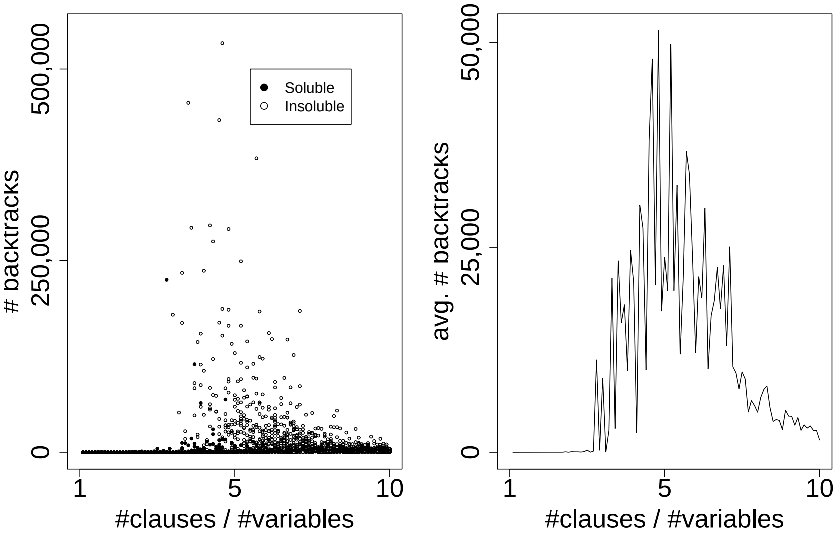

Fig. 3. The phase transition phenomenon. Each tick corresponds to solving a randomly generated combinatorial problem (SAT) by a heuristic algorithm (Davis-Putnam). The color indicates the solubility of the random problem, the vertical coordinate represents the run time, the horizontal coordinate corresponds to the problem constrainedness (left: underconstrained, right: overconstrained, middle: phase transition region with a sharp spike of average run time; see text for details). A restricted form of first-order logic hypothesis testing is equivalent to the SAT problem making this kind of statistical analysis suitable for machine learning algorithms. Interestingly [4, 23] show that they typically generate hypothesis in the phase transition region.

bond($d1, d1_1, d1_2, 7$*)*, which represents the fact that there is an aromatic (7) bond between atoms $d1_1$ and $d1_2$ in compound d1. Numerous further kinds of facts are of course empoyed, describing, amongst all, the three-dimensional structure of each compound. But more importantly, each compound in the data set is known to be either mutagenically active, or inactive, and the algorithm is asked to propose a hypothesis which would, roughly said, link the mutagenic activity with particular structural properties. A highly trivialized example of such a hypothesis is represented by the following first-order logic rule, in which capital letters denote universally quantified variables.

$$\mathit{active(A)} \leftarrow \mathit{atm(A,\ B,\ c,\ 10,\ C)} \wedge \mathit{atm(A,\ D,\ c,\ 10,\ C)} \wedge \mathit{bond(A,\ B,\ D,\ 1)}$$

This hypothesis suggests that a compound is active if it contains two aromatic carbon atoms connected by a single bond. Most machine learning algorithms proceed by searching through a large space of hypotheses, testing each one as to how well it fits the available empirical data. It is interesting to note that the problem of verifying whether a hypothesis such as the above holds for a particular data instance (ie. a chemical compound description in our example) can be translated, under rather general assumptions, onto a *contraint satisfaction* problem (CSP), a slightly generalized version of the SAT problem I described earlier, retaining all principle complexity properties, namely NP-completeness and the presence of the phase transition phenomenon. One would hence anticipate to observe some form of phase transition in machine learning algorithms.

Indeed, the recent studies [23, 4] have confirmed the expectation. They studied the behavior of ILP algorithms applied on a large collection a learning problems. By viewing each pair hypothesis – learning example as an instance of CSP, it was possible to determine which hypotheses are located in, or near, the intriguing phase transition region. Here is the noteworthy finding: the heuristic search conducted by an ILP algorithm, whichever hypothesis space point it commences at, typically terminates in the phase transition area. In other words, the phase transition region forms an attractor for trajectories pursued by heuristic ILP system searching in the space of hypotheses. But the adventure does not end here.

3 Heavy-Tailed Runtime Distributions

The phase transition region is not only an attractor for ILP systems but as well for researchers trying to figure out some concise statistical description of the events in, or close to the tricky region. The pioneering work [8] subjected some state-of-the-art heuristic CSP algorithms (such as *conflict-directed backjumping*) applied on a collection of hard CSP instances configured as to lie in the phase transition area to a statistical survey, in order to model the cumulative distribution function (cdf) $F(t)$ governing the time it takes for the respective algorithms to arrive at a solution, ie. $F(t) = P(T \leq t)$ is the probability that the algorithm in question finds a solution to a CSP instance randomly drawn from the problem collection in time T not exceeding t (measured eg. in the number of backtracks made by the algorithm). In statistical studies of this kind, researchers usually anticipate events obeying some *standard* distribution, which in this particular case would imply that $1 - F(t)$ would decay with an exponential rate with growing t. Quite surprisingly, it was found that $1 - F(t)$ typically decays significantly slower. This has some interesting consequences. For example, consider the quantity $h(t) = f(t)/(1 - F(t))$. If a problem instance is not solved at time t, then $h(t)\Delta t$ is the probability of finding a solution between t and $t + \Delta t$. While $h(t)$ is constant for the normal distribution, it decreases with t in several problem families in the study [8]. This indicates that it may be actually beneficial here to *give up* the incremental building of a solution if it has not been succesfully constructed in a short time, and restart the search (solution construction) from scratch.

This idea was elaborated in the influential paper [10]. Gomes et al have established that indeed, difficult real-world backtrack search problems typically exhibit runtime distributions with very *heavy tails* (slow decay), well approx-

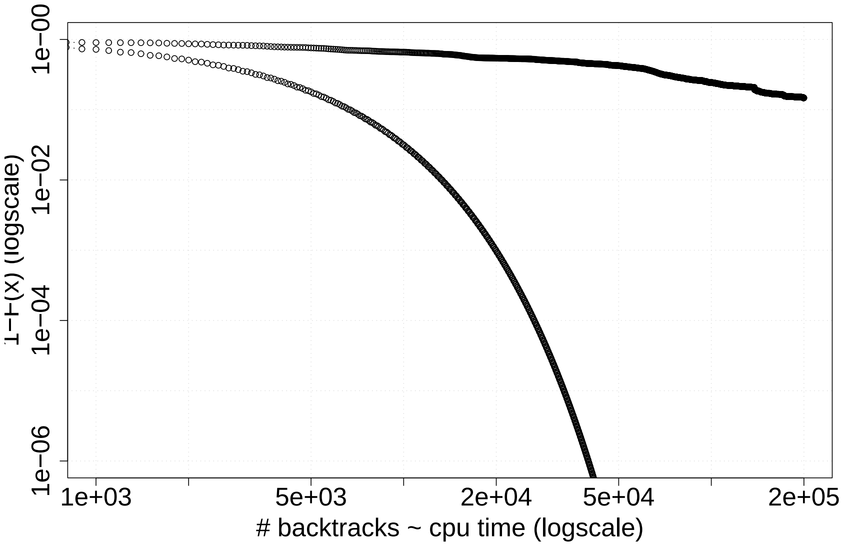

Fig. 4. The heavy-tailedness phenomenon. For standard cumulative distributions $F(x)$ (see details in text), $1-F(x)$ decays to zero with an exponential rate; the corresponding function in a log-log plot approaches a vertical line descent with growing x. Heavy-tailed distributions decay slower, in a power-law manner; their log-log plot acquires a non-vertical linear shape with growing x [10]. Some randomized hypothesis search algorithms exhibit run times governed by a heavy-tailed distribution [27], making their statistically expected run time theoretically infinite; see text for details.

imable by *power-law* distributions (Fig. 4 illustrates the difference between a normal and a power-law distribution on a log-log scale). Heavy-tailed distributions have long been known from economy. They were used by Vilfredo Pareto to model income distributions ("for an arbitrarily large income I, there is a non-negligible portion of people earning I or more," where "non-negligibility" denotes the subexponential decay of probability), but they were considered more or less probabilistic curiosities until the work of Mandelbrot [20] on modeling real-world phenomena by fractals.

The results in [10] have brought up further intriguing runtime properties of search algorithms in hard combinatorial problems; for example, certain power-law distributions have infinite moments, including the mean. In other words, one may have a significant probability that a search algorithm will find a solution in a short time, although the overall statistical runtime expectation may be theoretically infinite. These properties were empirically detected in the mentioned experimental setting of a fixed algorithm and a randomly generated problem instance, but held equally in the case of a fixed problem instance and a randomized search algorithm (introduced eg. by a non-detrimental randomization

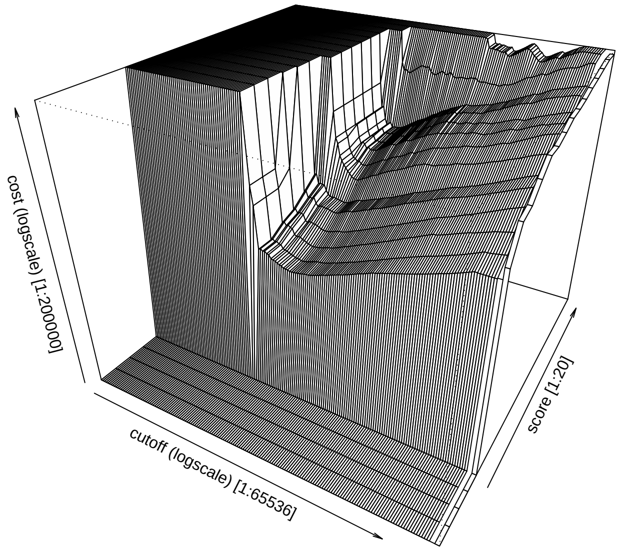

Fig. 5. The expected cost of randomized hypothesis search conducted by an ILP algorithm in a biochemistry discovery problem, as a function of (1) the required *score*, expressing the goodness of fit of the resulting hypothesis on the empirical data, and (2) the "cutoff" number of subsequently explored hypotheses after which the algorithm restarts the search 'from scratch,' with a new, randomly chosen hypothesis. The cost is measured as the total number of explored hypotheses. With a good setting of the cuttoff (around 100, corresponding to the groove parallel to the score axis), an order-of-magnitude increase of efficiency can be achieved with respect to the deterministic non-restarted algorithm described by the right-most isolated strip. The top flat area corresponds to the infinite expected cost.

of employed selection heuristics, such as random tie-breaking). The study has also confirmed the benefits of randomizing the search and restarting it repeatedly each time after a relatively short, 'cuf-off' time $\gamma > 0$ and each time with initial search conditions chosen randomly, independently and from an identical distribution. Indeed, if $F(\gamma) > 0$, then even if $F(t)$ has an infinite mean (ie. the non-restarted search has an infinite expected runtime), the cdf of the number of restarts N made by the restarted algorithm $F_\gamma(N) = 1 - (1 - F(\gamma))^N$ is clearly exponential and as such it has a finite mean.

The detection of heavy-tails in runtimes of search algorithms bears instant consequences to computer based scientific discovery, namely to its efficiency aspects. In [27], we have shown that heavy-tailed runtime distributions actually manifest themselves in the operation of an ILP system searching for hypotheses in the biochemistry domain addressed earlier in the text. Is this a surprising observation? As we have mentioned, the ILP system repeatedly generates hypotheses by means of a search in some hypothesis space. The search basically corresponds to traversing from one hypothesis onto another by performing some heuristic changes ('refinements') in the former, obtaining the latter. Each generated hypothesis is tested for its fit on the empirical data, where the testing also has a form of search (remember the mentioned correspondence with the CSP). Thus one can view ILP as one search procedure (hypothesis testing) embedded in another search (hypothesis generation). From [4] (Section 2) it follows that the outer search tends to produce hypotheses forcing the inner search towards the phase transition region, possibly giving the hypothesis testing runtime distribution a heavy-tailed shape. The composition of a large number of the testing procedure would expectedly yield the overall heavy-tailed runtime distribution of the ILP algorithm. In the recent paper [28] we have shown that this speculation does not provide the whole explanation of the phenomenon. Consider the random variable C equaling the number of hypotheses generated before a 'good' one (ie. one exceeding some prescribed score s of hypothesis-data fit) is found. We have found that $F(c|s) = P(c \leq C|s)$, a quantity *independent* of the hardness of testing, also typically acquires a heavy tail for sufficiently large s. Thus the runtime distributions of the ILP algorithm do not (only) inherit the heavy-tailedness from the embedded hypothesis testing (or CSP) procedure, but rather result from the intrinsic structural properties of the hypothesis search space.[1] The space is defined by the subset of the first-order predicate logic language used to express the hypotheses as well as the operators used traverse the space; usually heuristically designed and in a way immitating the humanly way of forming hypotheses.

It is straigtforward to exploit the empirically found typical runtime distributional properties of the ILP system to boost its efficiency. Much like in the original approach in [10], a relatively simple restarted search strategy leads to quite dramatic improvements of the mean runtime of hypothesis search (see Fig. 5) [28]. Interestingly, the favorable value of the restart cutoff (the number of subsequently refined hypotheses after which the algorithm restarts 'from scratch' with a randomly drawn hypothesis) leading to the largest efficiency improvements appears common to different refinement operators as well as different data domains [28].

4 Occam's Razor Under Scrutiny

Up to now I have demonstrated how findings originating from the experimental setting in Fig. 2 have served to improve the *efficiency* of computerized hypothesis forming. An equally, or even more important factor determining the quality of scientific discovery is of course the *predictive performance* of the resulting hypotheses, ie. their ability to *generalize* from sample data. In the biochemical example problems addressed earlier in this text, this ability would be measured in terms of the hypothesis' accuracy in classifying mutagenicity of compounds not

[1] In [9], the heavy-tailed runtime character of search algorithms is explained by the fractal structure of large search spaces corresponding to real-world combinatorial problems.

present in the *training* data, ie. data used for hypothesis construction. The basic experimental modus from Fig. 2 has provided some insight on the relevancy of the well-known Occam's razor principle[2] to predictive performance of hypotheses.

Machine learning algorithms have traditionally adhered to a loose interpretation of the Occam's dogma according to which, if several hypotheses exhibit same or close consistency with the modeled empirical data, the simplest among them should be chosen. The simplicity bias is on one hand motivated by ease of human interpretation of the produced hypotheses, but on the other hand, it is also generally believed that, everything else being (almost) equal, a simpler hypothesis is likely to achieve a greater predictive accuracy than a more complex one.

This strong assertion appears to be justified on several fronts. Prominent advocates are the *bias-variance trade-off principle* [11] from statistical learning theory and the *probably-aproximately-correct learning* framework [21] from computational learning theory. Both of these formalisms basically conclude that over-extending the space in which the hypothesis is searched[3] has a detrimental effect on the expected predictive accuracy due to the increased risk of *overfitting* the empirical data [11]. Against the usual conjecture though, there is no neccessary connection between the size of a hypothesis space and the complexity of the hypotheses contained therein, as clearly argued in eg. [7]. Thus neither of the mentioned apparata directly supports the Occam's razor. Other theoretical advocacies, encapsulated in the Kolmogorov complexity theory [18], seem to entail the Occam's principle elegantly, yet their logic is circular: they postulate one form of simplicity preference to derive another. Namely, simpler Turing machines are considered apriori more probable, and predictions made by simpler programs (corresponding to hypotheses in our context) are considered more reliable as a consequence.

An alternative stream of argumentation uses *empirical* evidence mainly from history of natural sciences, to show that the simplest consistent theory "works best." Indeed, popular examples lend themselves here, such as the simpler Copernicus's solar system model versus that of Ptolemy [7]. But history provides way too small an empirical sample to make a reliable conjecture. On the other hand, the sample of computer-based hypothesis discovery processes we nowadays can collect (Fig. 2 again) is overwhelmingly large and can serve as a perfect empirical testbed for the Occam's razor principle.

The paper [7] gives an interesting review of empirical evidence both for and against the Occam's rule, analyzing experience with algorithms for knowledge discovery in databases. On the "pro" side, the author addresses (i) the technique

[2] *"Plurality should not be posited without necessity"* a thesis of the 14th century English logician William of Ockham, later often articulated as *"Entities should not be multiplied beyond necessity,"* and generally known as the Occam's razor.

[3] Even infinite hypothesis spaces may be "over-extended" in terms of their *capacity* known as the *Vapnik-Chevronenkis dimension* [11].

of classifier *pruning*,[4] commonly considered a measure improving predictive performance, (ii) the allegedly high generalization performance of very simple hypotheses (here rules) found by [13] on some common discovery benchmark data sets as well as (iii) typically good generalization accuracy of 'stable' discovery algorithms, ie. those with a low variance component of the bias-variance trade-off. The author of [7] points out that while the extremely simple classifiers presented by [13] are in fact admitted to be 5.7% less accurate those produced by the 'golden standard' C4.5 learner and thus hardly support the simplicity-boosts-accuracy thesis, the successes of heavily pruning or intrinsically low-variance algorithms are actually due to their strong restrictions on the *size* of the searched hypothesis space. It is only a technical matter that they shrink the space as to contain only hypotheses up to some maximal complexity. In the empirical study [27], we show that just as well, the space may be constrained to contain a rather low number of relatively complex hypotheses maintaining the generalization performance of the learning algorithm. Here, the searched hypothesis space is constructed by randomly sampling elements (along with their local neighborhood) from an originally large prior hypothesis space.

Quite surprisingly, further substantial empirical evidence has emerged contradicting the discussed interpretation of the Occam's rule. [7] reviews a collection of experiments conducted by different authors, concurring in the conclusion that simple hypotheses exhibit no superiority over complex ones in terms of generalization performance. Remarkably, [22] presents compelling evidence of excessive search often leading to models that are simultaneously simpler and less accurate. Lastly, the recent successes of *ensemble methods* [11] combining differently learned classifiers through various voting or averaging schemes represent another source of evidence: [6] converts ensembles to equivalent single models which are in the majority of cases much more complex yet more accurate than any of the base models obtained originally from the learner.

Thus empirical surveys of learning algorithms on real world tasks have illuminated a deeply rooted misconception that model simplicity itself boosts generalization performance, and helped realize that the overfitting phenomenon is a consequence of excessive hypothesis search rather than hypothesis complexity per se. In light of the mentioned results, the person in Fig. 2 will likely need to revise his/her meta hypothesis.

5 Meta Learning

We now replace the human in Fig. 2 by a computer, thereby shifting to the presently hot topic of *meta learning* (Fig. 6), where a computer algorithm learns hypotheses about other algorithms for learning hypotheses.

Traditionally, meta learning approaches have been exploited for the task of 'learning to learn'. The specific goal is usually to discover a hypothesis able to determine which classes of base learners (algorithms inside the boxes in Fig.

[4] simplifying a hypothesis by removing some of its overly complex fragments usually at the price of decreased accuracy on training data

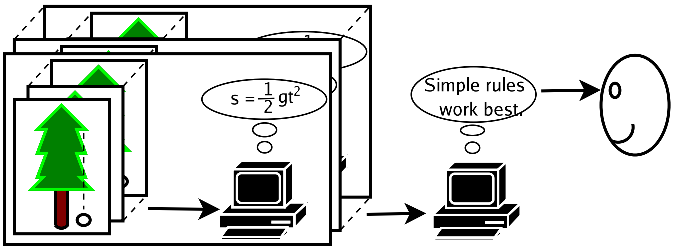

Fig. 6. Machine meta learning: a computer discovering a hypothesis about hypothesizing computers (algorithms). Meta learning approaches have generated significant contributions to data mining practice; see text for details.

6) have best chances of producing good models, or suggest a promising range of parameters of a given base learner. The input to such a hypothesis is some description of the character of the empirical data at hand, such as the number and types of quantities observed, the number of observations etc. The paper [14] is an example of the former approach: here the authors use a learning algorithm at the meta level to group base learning algorithms into clusters containing those with similar performance on data of given characteristics. Using the learned meta hypothesis, a user can then choose a learner from the most promising class of algorithms for a new discovery problem. The former approach is followed for example in [24] where the meta algorithm constructs a hypothesis that, given a data description, should determine a good value of a single parameter of a specific model.[5] The value is then used at the base level where the rest of parameters are learned.

Apart from the mentioned mainstream approaches, some interesting meta learning strategies have been researched, related to the issues discussed earlier in Sections 2 and 4 of this paper. In particular, in the paper [19] the goal is again to construct a model predicting the a suitable learning algorithm for a given base problem. The interesting point lies in how the base problem is described. Rather than utilizing usual data descriptors exemplified above, the author use with benefits certain derived parameters, which determine the problem's position in the phase transition region, showing that such information indeed has strong relevance to the competency of the algorithm selected for the learning problem. Lastly, in the work [2] a meta algorithm is devised to learn how much pruning should be applied on models produced by base learners.

6 Concluding Speculations

I have attempted to review and identify some links within recent experimental works exhibiting the common denominator of subjecting discovery processes to

[5] They namely predict a good value of the *kernel width* of a *support vector machine*.

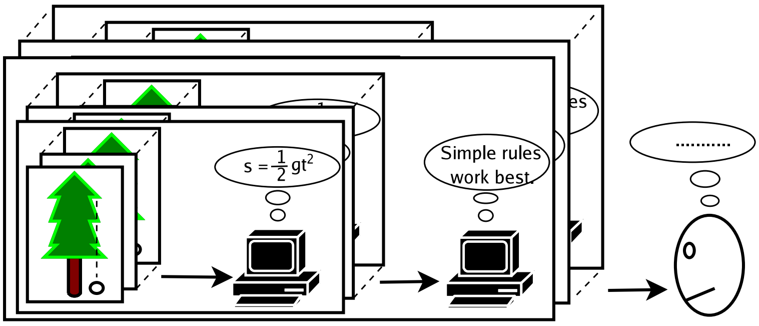

Fig. 7. Studying meta learning as an empirical phenomenon.

discovery processes, in both cases primarily computer-based. This stream of research has generated some quite significant lessons including the detection of the phase transition and heavy-tailed runtime distribution phenomena in learning, clarification of the Occam's razor's relevance to knowledge discovery, and yielded algorithms learning to learn.

The unifying viewpoint I took in this review automatically entails some questions, to my best knowledge so far unresearched. First, if meta learning about learning generated useful results, there is hardly a reason why meta meta learning about meta learning should not as well. As the state-of-the-art allows to conduct meta learning processes by computers (Fig. 6), it should be again feasible to collect a sufficient empirical sample to induce meta meta hypotheses, as illustrated in Fig. 7. The depicted person could of course be again replaced by a computer, which would ultimately allow to extend the configuration to a 'tower' of an arbitrary number n (possibly infinite) of $\underbrace{\mathit{meta}\ldots\mathit{meta}}_{n\times}$ learning processes (Fig. 8). This setting would resemble the concept of infinite towers of meta interpreters of programming languages, as known from artificial intelligence, mainly in the context of LISP [5] or Prolog [1]. Whether this kind of inquiry would yield results with impact on practice falls now of course into the realm of speculation.

The second question is whether results about algorithmic hypothesis discovery, obtained through meta discovery and applied back to improve the base discovery algorithms, could be as well exploited to improve the regime of scientific discovery performed by humans. A remarkable stream of research has recently emerged exploring the connections between philosophy of science and machine learning [3, 17, 26]. Despite its relevance, this research mostly addresses the inverse of my question. That, on the contrary, regards the potential translation of results *from* machine learning, along the following exemplary lines:

– Given that randomized restarted strategies for constructing hypotheses appear useful in numerous problems of algorithmic discovery, would they exhibit similar benefits in the scientific work of human researchers? That is,

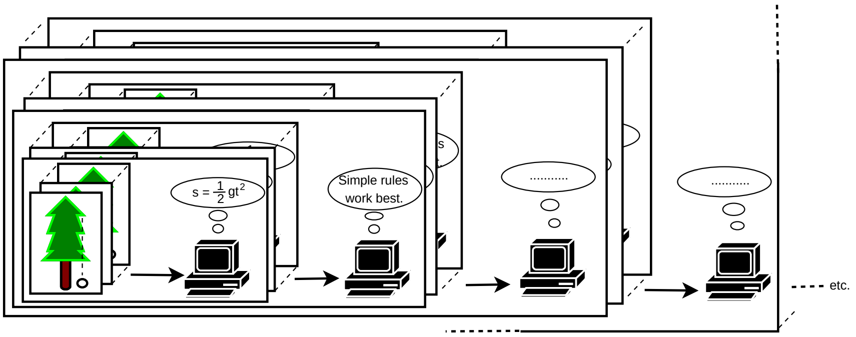

Fig. 8. An infinite 'tower' of meta learning processes, resembling eg. infinite towers of programming language meta interpreters studied mainly in the context of AI languages Prolog and Lisp. While the latter are deductive in nature, our tower is inductive; see text for details.

would it expedite the course of describing and modelling natural phenomena, if scientists deliberately abandoned their paradigms at occasions, and started anew, making each time slightly different decisions in critical phases of theory building?

- Given the successes of ensemble methods in machine learning, would it be beneficial if scientists developed, mutually independently, multiple different hypotheses for a single phenomenon, all subsequently combined to make a prediction or decision, through voting or (where appropriate) averaging?
- Given that ensemble methods require a certain degree of variance among models employed in the combined model, would traditional science profit from deliberately proposing collections of overly complex hypotheses, in order to create a larger variance thereamong?

Acknowledgement

The author is supported by the Czech Ministry of Education through the project MSM1P05ME755.

References

1. H. Abramson and M. H. Rogers, editors. *Meta-Programming in Logic Programming, Workshop on Meta-Programming in Logic.* MIT Press, 1989.
2. H. Bensusan. God doesn't always shave with Occam's razor - learning when and how to prune. In *Proceedings of the 10th European Conference on Machine Learning*, pages 119–124. Springer-Verlag, 1998.
3. H. Bensusan. Is machine learning experimental philosophy of science? In *ECAI'2000 Workshop on Scientific Reasoning in AI and Philosophy of Science*, pages 9–14, 2000.
4. M. Botta, A. Giordana, L. Saitta, and M. Sebag. Relational learning as search in a critical region. *Journal of Machine Learning Research*, 4:431–463, 2003.
5. J. des Rivieres and B. C. Smith. The implementation of procedurally reflective languages. *ACM Symp. on Lisp and Functional Programming*, pages 331–347, August 1984.
6. P. Domingos. Why does bagging work? a Bayesian account and its implications. In *3rd International Conference on Knowledge Discovery and Data Mining*, pages 155–158. AAAI Press, 1997.
7. P. Domingos. The role of Occam's razor in knowledge discovery. *Data Mining and Knowledge Discovery*, 3:409–425, 1999.
8. D. Frost, I. Rish, and L. Vila. Summarizing CSP hardness with continuous probability distributions. In S. Matwin and C. Sammut, editors, *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 327–334. AAAI Press, 1997.
9. C. P. Gomes and B. Selman. On the fine structure of large search spaces. In *11th IEEE International Conference on Tools with Artificial Intelligence*, pages 197–201. IEEE Computer Society, 1999.
10. C. P. Gomes, B. Selman, N. Crato, and H. A. Kautz. Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. *Journal of Automated Reasoning*, 24(1/2):67–100, 2000.
11. T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning, Data Mining, Inference and Prediction.* Springer, 2001.
12. B. Hayes. Can't get no satisfaction. *American Scientist*, 85(2):108–112, 1997.
13. R. C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 20:93–91, 1993.
14. A. Kalousis, J. Gama, and M. Hilario. On data and algorithms: Understanding inductive performance. *Machine Learning, Special Issue on Meta-Learning*, 54, 2004.
15. R. D. King, K. E. Whelan, F. M. Jones, P. K. G. Reiser, C. H. Bryant, S. H. Muggleton, D. B. Kell, and S. G. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427:247–252, 2004.
16. S. Kirkpatrick and B. Selman. Critical behavior in the satisfiability of random boolean expressions. *Science*, 234:1297–1301, 1994.
17. K. Korb and H. Bensusan, editors. *ECML'2001 Workshop on Machine Learning as Experimental Philosophy of Science.* 2001.
18. M. Li and P. Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications.* McGraw Hill, 1997.
19. J. Maloberti and M. Sebag. Fast theta-subsumption with constraint satisfaction algorithms. *Machine Learning*, 55(2):137–174, 2004.
20. B. Mandelbrot. *The Fractal Geometry of Nature.* Freeman, New York, 1983.

21. T. Mitchell. *Machine Learning.* McGraw Hill, 1997.
22. J. R. Quinlan and R. M. Cameron-Jones. Oversearching and layered search in empirical learning. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 1019–1024, 1995.
23. A. Serra, A. Giordana, and L. Saitta. Learning on the phase transition edge. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 921–926. Morgan Kaufmann, 2001.
24. C. Soares, P. Brazdil, and P. Kuba. A meta-learning method to select the kernel width in support vector regression. *Machine Learning, Special Issue on Meta-Learning*, 54, 2004.
25. A. Srinivasan, S. Muggleton, M. J. E. Sternberg, and R. D. King. Theories for mutagenicity: A study in first-order and feature-based induction. *Artificial Intelligence*, 85(1-2):277–299, 1996.
26. J. Williamson. Machine learning and the philosophy of science: a dynamic interaction. In *ECML'2001 Workshop on Machine Leaning as Experimental Philosophy of Science*, 2001.
27. F. Zelezny, A. Srinivasan, and D. Page. Lattice-search runtime distributions may be heavy-tailed. In *12th International Conference on Inductive Logic Programming*, Lecture Notes in Artificial Intelligence. Springer, 2003.
28. F. Zelezny, A. Srinivasan, and D. Page. A Monte Carlo study of randomised restarted search in ILP. In *14th International Conference on Inductive Logic Programming*, Lecture Notes in Artificial Intelligence. Springer, 2004.