A tour of Machine Learning: an AI perspective

Michèle Sebag

A tour of Machine Learning: an AI perspective

Michele Sebag [a]
[a] *TAO, CNRS − INRIA − LRI, Université Paris-Sud, 91405 Orsay Cedex, France*
E-mail: sebag@lri.fr

Machine Learning has been at the core of Artificial Intelligence since its inception. Many promises have been held, if one is to consider that Google is a living demonstration of AI. This paper presents a historical perspective on Machine Learning, describing how the emphasis was gradually shifted from logical to statistical induction, from induction to optimization, from the search of hypotheses to the search of representations. The paper concludes with a discussion about the new frontier of Machine Learning.

Keywords: Machine Learning, Statistical Learning, Unsupervised Learning, Change of Representation, Reinforcement Learning, Machine Reasoning

1. Introduction

At the dawn of Artificial Intelligence and even before the name was coined, Machine Learning was at its core. After Alan Turing, *by (...) mimicking education, we should hope to modify the machine until it could be relied on to produce definite reactions to certain commands* [Tur50]. But the steps he suggested towards AI somewhat differ from what was eventually called the Good Old Fashion vision of AI (GOFAI): one could *carry through the organization of an intelligent machine with only two interfering inputs, one for pleasure or reward, and the other for pain or punishment.* Alan Turing's vision of AI and Machine Learning was thus centered on reinforcement learning, with criteria and methodologies significantly different from those of GOFAI.

Regarding criteria, he proposed the imitation game, now known as Turing test, to circumvent the debate of whether a machine can think. The imitation game, seeing human beings as intelligence "oracles", asks the question of how far the machine is from the oracle. Thereby, the imitation game defines a regret-like criterion [LR85]: the focus is on minimizing difference between the machine and the oracle performances. This regret-like criterion induces different research priorities, compared to the criterion of the machine performances considered in isolation. The latter criterion led to a somewhat elusive intelligence pursuit in the first AI era (if your system can't solve this puzzle, it is not intelligent). It faces strong limitations *ab ovo*, e.g. concerning reasoning decidability or tractability issues in expressive representation languages. On the one hand, the imitation game criterion does not suffer from decidability or tractability limitations since same limitations are faced by human beings. On the other hand, it induces specific research priorities such as social intelligence and common sense. Along this line, the machine is acknowledged to be intelligent on the basis of its achievements more than its limitations. Still, the imitation game criterion was hardly seriously considered in the literature as it raises critical issues of subjective assessment: like beauty, intelligence is in the eye of the beholder.

Regarding methodologies, he made reinforcement learning (RL, aimed at maximizing the sum along time of whatever rewards presented to the intelligent agent by the environment [Sam60,SB98, Sze10]) the core learning mechanism. This early stress put on RL contrasts with the fact that ML has long been almost exclusively focusing on supervised and unsupervised learning [MCM83,Bis06]. More generally and as emphasized by Cristianini [Cri09], ML has mostly tackled well defined and restricted tasks in the 1980-2010 decades. Typically, both supervised and unsupervised learning

AI Communications
ISSN 0921-7126, IOS Press.

deal with some natural or artificial environment which is hardly ever modified during the course of learning, and never by the learning agent itself. According to Turing however, the proof of concept of AI is intelligence *in situ*, where the smart agent exerts some control on its environment as a robot does. Along this behaviourist perspective, intelligence is demonstrated as a means toward some end. The point of autonomy (that is, who decides about the ends) thus is part of the learning problem. The autonomy issue raises deep ethical as well as technical difficulties, which might explain why RL has been to some extent neglected compared to supervised and unsupervised learning (section 6).

This paper revisits the actual achievements of ML in the light of Alan Turing's vision and conjectures. A still burning issue concerns the interaction between the machine and its teachers/partners, and the role of the human in the loop. In a mundane perspective, this interaction has significantly evolved from Kubrick's *Space Odyssey* (1968) to Spielberg's *Artificial Intelligence* (2001). In 1968, HAL was a (quasi) omniscient, omnipotent and omnipresent system, and its self-awareness made it a threat for its human partners. In 2001, AI was more of a vulnerable and fragile entity, emotionally abused by thoughtless human beings. In the scientific perspective however, the evolution from Lenat's AM and Eurisko [Len82] to Google or Watson [Bak11] systems tells an entirely different story. In the early days of Eurisko system, its father/programmer Doug Lenat was fantasized as a scientific Pygmalion. Quite the contrary, an international army of humble common sense providers is behind the success of Google and Watson.

This paper presents a historical perspective on five decades of Machine Learning, where the emphasis was gradually shifted from inference and logics (section 2), to statistics and priors (section 3), to algorithms and optimization (section 4), to representation design (section 5), to the interaction with environment (section 6). In conclusion, we shall argue that the new frontier of ML is to become invisible, pervasive to computer science at large, and examine the implications of this new status (an ML component in every computational system) on the research criteria and priorities. But let us start with examining the position of the AI problem, the goals of the AI founding fathers, in the light of the Turing test.

2. From strong to weak AI

The Turing test – whether the machine could answer *questions in such a way that it will be extremely difficult to guess whether the answers are given by a man, or by the machine* [Tur50] – was meant to sidestep the allegedly meaningless question of whether a machine can think. On the positive side, this question blandly avoids the debate about strong *vs* weak AI, which has occupied quite a few philosophically and technologically inclined authors for several decades (see e.g. [Bod90]), with a tendency to go in infinite recursions about the notions of consciousness and awareness[1].

In the light of developmental cognition, intelligence is indeed viewed as a means of survival more than an end and a goal *per se* [PB07], making irrelevant the distinction between simulating and using intelligence. While survival would be an objective touchstone for assessing machine intelligence, this survival criterion is not operational: the intelligence of the machine could not and still cannot be directly assessed in terms of survival. Like a baby, the machine's "survival" critically depends on a nurturing environment (electricity, air conditioning, sensory/information networks, programmers).

Along this line, the negative side of the Turing test is to make AI an ill-posed problem, with no objective touchstone: *the extent we regard something as behaving in an intelligent manner is determined as much by our own state of mind and training, as by the properties of the object under consideration* [Tur50]. Specifically, if demonstrating human-like behavior is considered to be a proxy for intelligence, at least two goals can be defined: demonstrating "social intelligence", or demonstrating "academic intelligence". After [BN96], our intelligence judgments most often refer to the social intelligence of our partners, as opposed to their academic intelligence.

For some reason however, the founding fathers of AI were more interested in academic than social intelligence. The grand enterprise of building

[1] Searle's Chinese room thought experiment, proposed in 1980, pinpoints that a room "passing the Turing test" in Chinese, still does not "speak" Chinese in the sense that it does not "understand" Chinese. Most AI scientists (see e.g. [RN95]) suggest that the difference between "simulating" intelligence and "being" intelligent is irrelevant : there is nothing wrong with a behaviourist perspective in the realm of machine design.

a General Problem Solver [NSS59] pioneered and inspired many of the key features of AI, such as the distinction between data and instructions, or the formalization of a search problem in terms of i) state space, ii) navigation operators, iii) heuristic assessment of intermediate states. Meanwhile, Weizenbaum's Eliza showed that passing the Turing test could be in some cases disappointingly easy[2] [Wei66]. In this first era, the grand goal was thus to define AI as a science *on par* with, e.g. Physics. In this perspective, the stress was put on inference engines and their mathematical properties. Negative results regarding the decidability and computational complexity of logical inference in high order logics let however little hope about building *universal* inference mechanisms. As already mentioned these negative results are irrelevant in the perspective of the Turing test. This criterion, a regret-like criterion [LR85], focuses on how well the algorithm fares compared to the oracle, instead of considering the task difficulty *per se*: inference limitations that also affect human beings are thus discarded.

Part of the research toward deep AI and general inference was therefore redeployed toward the so-called shallow AI, aimed at building appropriate representations and gathering knowledge tailored to applicative problem domains. Expert Systems blossomed, manually and patiently fed by experts and knowledge engineers. But knowledge acquisition was at best an art, with rapidly diminishing returns. Machine learning then took a fresh start, aimed at overcoming the brittleness of manual knowledge engineering. At this point the promise of machine learning faced a brick wall. As put by Douglas Lenat, the promise that *the more you know the more you can learn (..) sounds fine until you think about the inverse, namely, you do not start with very much in the system already. And there is not really that much that you can hope that it will learn completely cut off from the world* [Len82]. The revenge of social intelligence then came: interacting with the world is a must-have.

The main lesson of the glorious *General Problem Solving* era was thus a bit ironical. Not only are the most praised intellectual skills (proving theorems, playing chess, and more generally demonstrating academic intelligence) seemingly more affordable to the machine than the so-called basic ones, recognizing one's grandmother, putting the clothes in the washing machine or sensibly answering the question: *Do you know what time it is ?* (which all resort to social intelligence). But overall, the acquisition of these basic skills is a pre-requisite for actually deploying AI on a non-toy scale (except in particular domains, exemplified by organic chemistry [KRO$^+$09], see below). After Thrun [TBF05], in 2005 AI achievements were very advanced in the domain of reasoning, moderately advanced in the domain of dialogue, and much less so in the domain of perception. One can but notice that the corresponding skills are acquired in the inverse order, both at the species (philogenetic) and the individual (ontogenetic) levels: with regard to both the Darwinian evolution and developmental psychology perspectives, perception precedes communication, and communication precedes reasoning.

AI priorities were thus revisited to tackle two huge challenges: perception and common sense.

3. From logics to statistics

While common sense and perception naturally raise different issues, they have quite a few features in common. Firstly, they witness the situated cognition claim [Cla93] that a continued interaction with the world is a necessary condition of cognition. Secondly, this continued interaction with the world implies that the machine should acquire and process "big data". Thirdly and as a consequence, the stress put on logics and deduction was shifted toward induction and statistics: inductive inference can establish *probable* conclusions, not *true* conclusions (the many white swans you saw do not exclude that you'll may, one day, discover a black swan).

The relation between probabilities and rationality is a major philosophical debate [Rus06], that started long before AI appeared although it took a new turn within AI. The main two positions can be summarized as follows. According to some, the world is governed by deterministic causal relationships, and probabilities are used to account for our incomplete knowledge. According to others, the world is genuinely indeterministic and probabilities measure its intrinsic indeterminism. Far from being yet another Byzantine debate, the two

[2] A nice anecdote reported by Kurzweil is that passing the Turing test can also be very hard for a human being, if the observer thinks it is a machine.

visions of probabilities lead to different modelings, depending on whether probabilities reflect our knowledge about the world (first position) or the world itself (second position).

Recall that ever since the first discussions between Pascal and Fermat in the XVII^e century, probability was meant both as a (subjective) degree of belief, and as the (objective) tendency of a stochastic system (e.g. a dice) to display stable relative frequencies. These two interpretations are at the root of the frequentist vs Bayesian approaches.

Three main branches of machine learning thus gradually appeared. A first one posits that information is available in symbolic form: the burden of forming abstractions from factual evidence can be spared to the machine in a few mature domains. In the domain of chemistry for instance, there is no point in asking the machine to rediscover the basic concepts of atoms, bonds and so forth; these can be manually provided by the machine instructor. In such domains, the machine starts with a strong "innate" or built-in knowledge and likewise produces results which are directly intelligible and operational to support standard deductive inference, as exemplified in Inductive Logic Programming [MDR94,KRO$^+$09]. The inductive leap (which one, out of the many possible explanations for the facts, is the best/most probable one) is handled by using one of the many names of the simplicity principle: Occam's razor, minimum description length [Ris78,Grü07], or structural risk minimization [Vap95]. Logical induction, close in spirit to the General Problem Solver, relies on the formalization of the concepts and theory like the emerged part of the iceberg relies on the immersed part.

A second approach is rooted in Bayesian inference [Pea91,RW06]. The initial critical mass of knowledge provided to the machine (its "innate" part) is embodied in the prior knowledge (although a good recipe is: in lack of a better guess, go for a uniform prior − non-informative priors). Bayesian inference proceeds by modifying the system prior knowledge on the basis of the factual evidence, to build posterior knowledge; in doing so, inference, probability calculus and rationality coincide with each other. The central mechanism of Bayesian inference, belief propagation, proceeds by having beliefs bumping in each other and adjusting their internal variables, very much in the way electrons do align their spin in a ferromagnetic Ising model[3]. It must be emphasized that Bayesian inference is no less powerful than logical reasoning: it supports not only associational inference (what if I see X, a.k.a. evidential or statistical reasoning), but also interventional inference (what if I do X, a.k.a. experimental or causal reasoning) and even retrospectional inference (what if *I had not done* X, a.k.a. counterfactual reasoning) [Pea00].

The third approach, the frequentist one, relies on facts only. Indeed the empirical evidence can be massaged to reflect the instructor's prior knowledge (e.g. moderate translations or rotations of a pattern do not modify its interpretation [Cun87]; lesioned sentences are neither semantically nor syntactically valid [CW08]) but the system only sees facts.

As a very rough summary of the 250-year controversy between Bayesians and Frequentists, let us suggest that these equally scientifically legitimate frames address different problems. Typically the Bayesian stance is interested in the average-case analysis, and is comfortable with the fact that the obtained results depend on external, "subjective" priors. The frequentist stance is more interested in the worst-case analysis, hardly compatible with uncontrolled external factors. A natural question is whether the huge heaps of data available everywhere eventually will make it unnecessary to provide prior knowledge, possibly biasing and spoiling the machine cognition. A back-of-the-computer experiment can make us feel the importance of prior knowledge, as a proxy for the "family education" of the machine. Let us try to build a model of the world from the Web, for instance about the colors of the fruits. Certainly the amount of data should be sufficient to establish that cherries are vastly red. Still, a surprising 20% fraction of cherries are found to be black, after the number of Google hits... Indeed cherries are red; that goes without saying; but what goes without saying is often unsaid.

4. From learning to optimization

Let us give a tour (admittedly subjective and by no way exhaustive) of the many learning settings

[3] Interestingly, Bayesian inference is more than metaphorically related to physics: belief propagation, proposed by Pearl in 1982, was independently proposed under the name of Bethe Peierl's approximation in the 30s.

and algorithms proposed in the last four decades. The history of ML has been full of surprises, acclaimed, abandoned and rediscovered approaches, covering all the range from knowledge-intensive and symbolic approaches, to data-intensive and statistical ones. This section follows the chronological order, and concludes with a short discussion about the lessons learned about the interplay of logics, statistics, and optimization.

4.1. Neural Nets

The bulk of the ML work has focused on supervised learning, training a classifier from examples labelled by the expert in order to automatically label further examples. Historically the first supervised learning approach is that of Neural Nets. The modelling of biological neurons as information processing units by [MP43] paved the way for Rosenblatt's perceptron [Ros58] and Widrow's Adaline [Wid62], followed by many others. After a brilliant start, NN studies were frozen for almost 20 years − during the so-called AI winter − as Minsky and Papert showed that perceptrons were limited to learn linearly separable patterns (failing to learn the XOR concept). NNs eventually recovered in the 80s, as i) perceptrons were extended into multi-layer perceptrons (MLP), ii) MLP were proved to be universal approximators for decent (sufficiently regular, i.e. square integrable) models; iii) a learning algorithm, the back-propagation of the gradient [Ama77,RM86,Cun87], could be used to train NNs from the available examples. Back-propagation of the gradient unfortunately comes with no optimality guarantees, making the NN calibration an art and hindering the reproducibility of the results. NNs thus entered a second phase of decline in the late 90s, as the Support Vector Machines (see below) appeared with a quadratic optimization setting, thus with optimality guarantees. Interestingly, NNs woke up again in the mid 2000s, with the inception of Deep Belief Nets [HOT06,Ben09] (section 5).

Formally, first generation-NNs provide a computational architecture, which can be manually tailored to enforce some desired properties of the domain (e.g. convolutional NNs enforce translational invariance [Cun87]). The learning stage involved statistics (defining the criterion to be optimized), optimization (searching for optimal weights) and algorithmics (defining a procedure conducive to good performances).

4.2. ML, an Artificial Intelligence approach

Independently, a grand programme of ML was proposed in the *Machine Learning, an Artificial Intelligence approach* (1983) [MCM83,MCM86, KM90], ranging from scientific discovery to analogical reasoning. The main incentive of ML at that time however remained the need to provide Expert Systems with knowledge bases, that is, rule-sets. Accordingly, the stress was put on symbolic supervised learning approaches. The intelligibility of the ML output was − and still is in most industrial applications − a key feature. The validation methodology, essentially *not* statistical in these early days, assumedly relied on the only expert's eye, with some unpleasant consequences due to over-fitting (see below). In the symbolic learning domain, the dominant approaches were the Version Space, Decision Trees and the bottom-up AQ system.

The Version Space (VS) defines *learning as search* of the complete and correct rules, covering all positive and no negative examples [Mit82]. Along this line VS was actually rooted in Constraint Satisfaction[4]. This most influential approach was limited by its computational complexity on the one hand, and the fact that constraint solving was inappropriate to deal with actual examples. Contrarily to usual constraints, examples are noisy, implying that the learning problem defines an unsatisfiable CSP in most real-world learning applications.

Decision trees (DTs) independently appeared in the machine learning and data analysis literature in the late 70s [Qui86,BFOS84]. Brute force decision trees are not infrequently winners of the current ML challenges (see also random forests, section 4.3), and they are the main ingredients behind Microsoft' Kinnect [Bis11]. DTs are rooted in a layer-wise approach, incrementally determining the optimal boolean test after a statistical criterion (information gain or entropy, Gini index, or classification error), and splitting the dataset accord-

[4]The tight relationship between machine learning and constraint satisfaction was however only marginally exploited. For instance, the key role of near-miss examples after Winston [Win75], those negative examples which minimally differ from the positive examples, can be interpreted as the near-miss derive the tightest, thus the most effective, constraints. The importance of constraint satisfaction for the ML field at large was only recognized later on [SGC11], or in relation with Data Mining [MT97,RFKM08].

ing to whether an example satisfies this boolean test, thus recursively dividing the data until forming quasi pure, positive or negative only, example subsets. Its strength and its weakness both come from this top-down strategy, very efficient on large and noisy datasets, and possibly misled by disjunctive concepts (with the XOR concept again in the role of the villain).

The AQ approach [Mic83] is a bottom up approach, aimed at iteratively finding the best rule covering a given example. While it addresses the problem of disjunctive concepts contrarily to DTs, it has been significantly less studied than the other two approaches as its components (e.g. which criterion should be used to determine the best rule, how to select the best rule-set from the set of rules covering the examples) were left to the algorithm designer. Nevertheless AQ paved the way to rule learning systems [FGL12].

Formally, logical induction was often tackled as a combinatorial optimization problem, searching among the lattice of hypotheses in the considered hypothesis language, be it propositional or relational. The learning stage likewise involves statistics (defining the criterion to be optimized) and relaxations of the combinatorial optimization problem (e.g. greedy search) or pruning of the search space (using e.g. bottom clauses), aimed at the discovery of good local optima.

4.3. PAC learning and ensemble learning

During the 80s, the theory of machine learning developed almost independently from the algorithms, centered on the Probably Approximately Correct (PAC) framework proposed by Valiant [Val84]. The PAC analysis, chiefly concerned with determining whether classes of concepts can be learned with polynomial complexity, and with arbitrary accuracy under any example distribution (strong PAC learnability), mostly delivered negative results. A more relaxed setting was thus defined, weak PAC learnability, only requiring the existence of an algorithm able to do a little bit better than random guessing under any distribution of the training examples. In 1990, a major and unexpected result was established by Schapire [Sch90]: while strong PAC learnability trivially implies weak PAC learnability, the converse also holds ! This result led to a new branch of machine learning, referred to as ensemble learning and aimed at learning hosts of hypotheses, favoring their independence through parallel (bagging) or sequential (boosting [FS96]) procedures. Many improvements were brought to the initial boosting algorithm, rooted in the proof of the strength of weak learnability theorem. The best known bagging algorithm is an extension of decision trees primarily designed to handle problem domains with huge numbers of features, such as pattern recognition [AG97,Bre01]. It proceeds by randomly selecting in each tree node a subset of the features, on which the standard node construction process applies. The resulting decision tree thus is a random variable, and a robust classification is obtained by pooling the votes of a large number of such random decision trees, thus yielding a random forest. Random forests are in 2013 among the most efficient learning approaches.

Ensemble learning, reminiscent of the wisdom of crowds [Sur04], is rooted on the fact that averaging many weak experts enables to reduce the variance of their opinions. Formally, the stress is put on enforcing the independence and diversity of the hypotheses in the ensemble.

4.4. Statistical learning

Last but not least, another branch of supervised machine learning appeared in the 90s, rooted in the statistical learning theory (SLT) pioneered by Vapnik (with the motto *There is nothing practical like a good theory* !). SLT studies how the empirical error (the mistakes done on the training data) and the generalization error (the error expectation on the whole problem domain) relate to each other: whereas the goal clearly is to minimize the generalization error, one can only control the empirical error. Indeed, the brute force minimization of the empirical error was observed by all practitioners to yield disastrous results in real-world applications − the known *over-fitting* phenomenon. It turns out that, much as for the approximation of integrals for well behaved functions, the empirical error provides a good approximation of the generalization error provided that the empirical error is measured on an identically and independently distributed training sample. Further, the difference between empirical and generalization errors can be bounded depending on the number of examples, and the complexity of the hypothesis space. The minimization of this bound, referred to as struc-

tural risk minimization, is at the core of the Support Vector Machines. The linear support vector machine first proposed by Boser, Guyon and Vapnik [BGV92] can be sketched as follows. In the separable case, there exists a single separating hyperplane maximizing its minimal distance to the training points, called margin, and this hyperplane is better than any other separating hyperplane. The importance of maximizing the margin can be intuitively understood in terms of noise: the larger the margin, the farther away the examples from the separating hyperplane, the more robust the separating hyperplane is w.r.t. to the example description noise.

The extension of SVMs beyond linear separable functions was soon proposed, notably based on the famed kernel trick [SBS98]. The kernel trick allows one to replace the input space (the description space of the example) with a so-called feature space, for free: since examples are only taken into account in the SVM formulation through their scalar products, it suffices to replace their scalar product by their scalar product in the feature space, or kernel. The kernel trick, enabling to find linear hypotheses in the feature space, thus enables SVMs to build arbitrarily complex hypotheses, to the extent permitted by the available examples, and to operate on virtually any description thereof (strings, graphs, texts) through defining appropriate kernels. Through kernels, every example is described through its relations with other examples. The kernel trick thus establishes each example simultaneously as a point in the input space, and a feature in the feature space.

Interestingly, SVMs share with Version Spaces the use of training examples as constraints; likewise, only the "near miss" examples, here the support vectors, are operational to define the hypothesis. The difference is that SVMs operate in the space of numerical functions (as opposed to, boolean functions; the *sign* operator is used to transform a numerical function into a boolean one). By using the norm of the numerical function as regularization term, SVMs thus define a quadratic, hence well-posed, optimization problem. The guarantee of global optimality was a strong incentive to research in SVMs, together with the rich possibilities offered by the reuse of convex optimization algorithms.

4.5. Partial conclusions

The first ML phase established that *overfitting* (the fact that hypotheses compliant with the data available so far, might be much less accurate on further data) was the main disease of learning. In a second phase, the overfitting phenomenon was mostly observed and controlled through e.g. cross-validation or bootstrap [Efr82]; one could see how hypotheses vary depending on the considered training sample, and adjust the learning procedure to account for this variability [Die98]. In a third phase, overfitting was studied from a theoretical standpoint; statistical learning focused on bounding the difference between the observed behavior (the training error) and the unknown general behavior (the generalization error); such an upper bound conveniently derives an optimization goal [Vap95]. The following phase is concerned with handling these optimization problems as efficiently as possible: in terms of global optimality (as in SVMs, although the SVM hyper-parameters still need be adjusted by cross-validation); in terms of convergence; in terms of computational complexity and scalability; in terms of accommodating further constraints or priors on the sought solution.

Currently, ML is at the crossroad of several disciplines. On the one hand, one must yield statistical guarantees about the ML results; specifically, every learning output should by now be mandatorily accompanied with its confidence interval [KTSJ12]. One must thus control the trade-off between the error of approximation (the distance between the "true model" and the best model you can find in the selected search space) and the error of estimation (the average distance between the best model you can find in this search space, and the one found by the algorithm, depending on the data sample). On the other hand, the computational time becomes a major limiting factor as the amount of data readily increases (the big data phenomenon [Bol10]). Besides the above errors, the error of optimization needs thus be accounted for (the algorithm stops short of finding the actual optimum of the learning criterion). ML must thus elevate its goal and take in charge the trade-off between these three types of error [BB07].

5. Learning representations

Another central ML issue is the search for appropriate features, including feature selection [GE03], constructive induction [TPB99] and dimensionality reduction [RS00,dST03,Ach01]. Roughly speaking, the dataset must sample the relevant regions of the search space, and pave the low dimensional space made of the relevant abstract or latent features, to enable learning. If these appropriate features are unknown, exponentially more examples are required in order to learn. Unsupervised learning, concerned with designing such new features and/or identifying concepts or clusters, made of subsets of examples, is long known to be a major milestone to tackle the grand AI goals.

The theory of unsupervised learning, a.k.a. exploratory data analysis, is less advanced than that of supervised learning [BDvLSTT05]. A first reason for this is that the validation of unsupervised learning, specifically that of clusters, has long relied on the only subjective assessment of the expert (the intelligibility of the clusters); a proxy for this subjective validation is whether the clusters or features facilitate discriminant learning. In this respect, unsupervised learning defines an inverse problem: find the features/clusters which will most facilitate supervised learning, whatever the labelling of the data.

A first unsupervised learning setting relies on a distance or similarity defined on the instance space[5]; clusters are formed of similar examples, and examples in distinct clusters are dissimilar. Along this line, unsupervised learning boils down to lossy compression, aimed at minimizing description-related criteria such as the Minimum Description Length criterion [Grü07]. Alternative criteria consider the clustering stability [Mei05]: the clustering result should minimally depend on the specific data sample considered; stability criteria thus transpose to unsupervised learning the statistical approach used for supervised learning. Another criterion, inspired from statistical physics, is scale invariance [FSZ10] characterizing the convergence of the clustering process along divide-and-conquer approaches (applying the same clustering algorithm on the clusters centers formed from independent data samples, must yield the same cluster centers [FD07]).

A second unsupervised learning setting, referred to as generative learning, is concerned with estimating the probability distribution of the data. Many distribution spaces and estimation algorithms have been proposed depending on the specificities of the data, ranging from parametric (e.g. Gaussian mixture models or latent Dirichlet allocation [BNJ03]), to non-parametric (e.g. one-class SVMs [VV06]) frameworks. Generative models are also trained by optimizing description-related criteria, e.g. the loglikelihood of the data under the model; the expert's priors are either explicitly modelled, or spelled out through the structure of the probability distribution model.

Interestingly, neural networks can also be used to learn an encoder/decoder mapping (using a 1-hidden layer feedforward NN architecture [TPB99]), or a generative model (using a restricted Boltzmann machine architecture [HOT06]) of the data. The architecture weights are optimized to minimize the reconstruction error in the former case, and maximize the loglikelihood (or related and more tractable criteria) in the latter case. Notably, such non-linear mappings or generative models can be learned on the top of each other, yielding the so-called Deep neural networks [BLPL06,HOT06]. The merit of such compound descriptions, as recalled by [BLPL06], is to enable very compact representations of complex structures[6]. The ability to learn gradually complex representations of the problem domain, one of AI's holy grail, has triggered the NN revival since the mid 2000s.

Yet another branch of unsupervised learning, inspired from compressed sensing [CRT06], is concerned with dictionary learning. Instead of mapping the input space (e.g. $\mathbb{R}^D$) into a low-dimensional space ($\mathbb{R}^d$, with $d << D$) defined after a few linear or non-linear features, dictionary learning finds a family of features (the dictionary) such that every example can be expressed using a few features thereof. The data are thus mapped

[5] Obviously, a relevant distance or similarity encapsulates a good deal of expert knowledge. For instance, the Euclidean distance on the input space is relevant iff the initial description is of sufficiently good quality. Metric learning, usually aimed at discriminant learning, is a rapidly developing topic [WBL05].

[6] For instance, the representation of the n-parity problem requires 2^n nodes in a 1-layer NN architecture, *versus* n nodes in a $\log n$-layer architecture [Has87]. In counterpart, NNs with many hidden layers are notoriously hard to train, due to the many local optima of the underlying optimization problem.

onto a low-dimensional manifold of the usually high-dimensional dictionary space as each example is expressed using a few words, i.e. with small L_0 norm. Reducing the dimensionality of the example description (regardless of the space dimensionality D) helps combating the data description noise and supports efficient learning. Like generative learning, dictionary learning most conveniently enables to express the expert's priors, e.g. regarding the spatio-temporal structure of the data [MBPS10].

6. Reinforcement learning

Another major ML topic is reinforcement learning (RL), concerned with gathering rewards through acting in a possibly unknown world [SB98,Sze10]. The goal of RL is to learn a policy, mapping each state onto an appropriate action, such that the cumulative reward gathered until the time horizon, is optimal. Note that the ability of finding such an optimal policy is at the core of many domains, ranging from robotics (navigate and demonstrate an appropriate behavior in some real or artificial environment) to games (with possibly partially observable information − when you don't know the cards of your partner) or economics. RL most usually and with no loss of generality assumes the environment to be Markovian, that is, the choice of the optimal decision only depends on the current state of the agent[7].

Several temporal settings are considered in RL. In the infinite time horizon setting, a discount factor is applied to favor the reaping of rewards as early as possible in the agent lifetime (otherwise, the agent could spend the first half of eternity doing nothing; or a random walk would be as good as anything). In most other settings (episodic RL), the agent reaches some terminal state (e.g., winning or losing the game) at some point, and gets a positive or negative reward at the end of each episode.

A central RL issue thus is to determine the *value* or *utility* of each state, e.g. the maximal expected cumulative reward one can collect after reaching this state. Indeed, when the state utilities are known, the optimal policy boils down to greedily moving to the next state with maximal utility. The difficulty of RL thus depends on the environment, and the agent knowledge about the environment. In a known world, the agent recursively computes the utility of each state through e.g. dynamic programming, or approximate dynamic programming if the world (state and action spaces) is high dimensional [BT96]. If the world is unknown, the agent faces the known *Exploration vs Exploitation* dilemma: on the basis of the agent current knowledge, some state-action pairs are better than others, and the agent should thus favor these (*exploitation*). But in doing so, the agent might miss the truly optimal state-action pairs; some *exploration* of the world is thus needed in order to gradually uncover the best state-action pairs. Exploration is all the more costly when the world is partially observable.

As in supervised learning, the RL theory is concerned with the consistency property (does an algorithm eventually reach the optimal policy), and the speed of convergence (how many data and training epochs are needed to yield an approximate optimal policy).

Some attempts to build a bridge between RL and supervised learning have been done, using the expert's traces as a source of labelled examples: each state is labelled with the associated expert's decision [DL08]. The limitations of such an approach can be best understood in the domain of game playing: While supervised learning is interested in making few errors in expectation, an automatic game player should make *no* error: regardless of the number of excellent moves done, an automatic player can lose on a single mistake... Equivalently, supervised learning is concerned with *independent and identically distributed examples*. But a game archive or a robotic log do not follow an iid distribution of moves; each move depends on the preceding ones; the robot state at time t certainly is not independent on its state at time $t-1$.

The agent thus needs fresh data, generated anew during each training phase and illustrating the limitations of the current tentative policy. For instance, the first backgammon program to reach a champion level in the 80s, TD-Gammon [Tes02], exploits games generated from self-play to train

[7] This assumption is not restrictive from a theoretical standpoint: if the optimal decision would depend on the past trajectory of the agent, the state space can be redesigned in such a way that the current agent state reflects its trajectory. Such an increase of the state space dimension however has dramatic consequences on the computational complexity of the RL problem.

a value function; the initial state is labelled with value 1/2, the end state is labelled with a 0 or 1, depending on whether the first player wins or loses, and a regularized regression is used to infer the value of every state. In such cases, RL is provided with a generative model of the world (the game simulator), enabling data-intensive approaches. Actually, most RL approaches are data-intensive and rely on the use of a simulator. The stress is put on estimating the optimal value function, by alternating i) the use of the current policy, combined with some exploration of the world and providing new evidence; ii) the update of the current value function through fixed point operators, particularly the celebrated Bellman equations (noting that the utility of the current state is the instant reward, plus the utility of the next best state, up to some temporal discount); iii) the computation of the greedy policy based on the current value function.

The expert's traces are also exploited in RL to learn the reward function (see inverse reinforcement learning [NR00,AN04], learning by imitation [CGB07], or learning by demonstration [KKBG10]). Another RL trend is concerned with direct policy search (see e.g. [PS08]), that is, solving the RL optimization problem in the policy space.

A key RL issue is to design a relevant and compact state space. For instance, the predictive state representations pioneered by [LSS01] describe a state through a partial forward model of the world: conditionally to its current state, the agent can make predictions about the effects of its actions. Interestingly, PSR representations are close in spirit to sensory-motor contingencies (SMC) [8] [O'R06]. In some contexts such as robotics, data-intensive and simulator-based RL approaches however raise specific difficulties. On the one hand, the optimal policy trained in simulation often does not behave properly in the real world, a phenomenon known as *Reality Gap* [LBZM06]. On the other hand, training a robot *in situ* raises the problems of experiment time and robot fatigue, due for instance to mechanical hazards.

[8] It is well known that providing a complete and correct declarative description of e.g., simple objects such as a chair, is an AI-hard problem. SMCs elegantly overcome the lack of such declarative descriptions: though I don't know what a chair is, I know that if this is a chair and I seat on it, I will be stable, seated, and at the right height.

In the standard RL setting, the training phase is clearly separated from the production or testing phase. In the former phase, the agent determines the utilities of *every* state and action; an intensive and exhaustive exploration of the state action space is required to provide optimality guarantees on the learned policy. In the latter phase, the agent greedily exploits the optimal policy discovered during the training phase. Yet another RL setting focuses on lifelong learning, making no distinction between the training and the testing phases. In this setting, inspired from game theory, and specifically in the multi-armed bandit (MAB) framework [LR85], the agent wants to simultaneously estimate the rewards and get an optimal cumulative reward. The criterion to minimize is the *regret*, measuring the gap between the agent performance and that of the oracle[9]. The MAB setting, analogous to single-state RL, aims at identifying the action with best reward. It has been extended to standard RL, considering *sequence of actions* through the Monte-Carlo Tree Search (MCTS) framework [KS06]. Based on MCTS, computer-Go players have jumped from the beginner to the professional level since the mid 2000s [GS07], − a notable advance of AI.

7. Discussion and perspectives

Along this brief and by no way exhaustive tour of ML, some notable achievements on the AI research agenda have been mentioned, ranging from autonomous vehicle driving [TBF05] to professional Chess [Hsu02] and Go [GS07] playing, from information retrieval and question answering [Bak11] to machine translation [GCDF09], to robot scientists [KRO+09]. Let us conclude by proposing and discussing three ML/AI research objectives and priorities.

A first objective regards the **maturity of ML**. Quite a few significant problems have been solved

[9] The optimal regret is logarithmic in the number of time steps [LR85], to be compared with the linear regret of an ϵ-greedy approach (greedily selecting the best empirical action with probability $1 - \epsilon$ and randomly exploring other action with probability ϵ). This logarithmic regret rate is reached by the UCB algorithm proposed by [ACBF02], which can be sketched as *Optimism in front of the unknown!*: select the action with maximal upper confidence bound on the empirical reward.

successfully on a grand scale. However, new types of problems usually require designing new algorithms; and new instances of known problems always require selecting an algorithm among the existing algorithm portfolios[10] and adjusting their hyper-parameters along a tedious and time-consuming process.

As advocated by Langford et al. [BDHL04], a *reduction* methodology is needed to go beyond the many faces of ML problems, and assess whether two types of problems are equivalent up to some reformulation. Establishing an appropriate framework to deal with such reformulations − and their statistical and algorithmic impacts − is a key step toward a mature ML theory.

On the practitioner side, a principled approach to select appropriate algorithms depending on the problem domain and available data is needed. Algorithm selection has been viewed as an ML bottleneck since the 80s, referred to as Meta-learning [VGCB10]. Interestingly, algorithm selection also is a key issue for neighbor disciplines such as constraint satisfaction (SAT); the difference is that a comprehensive set of features has been designed to efficiently characterize SAT problem instances [XHHLB08], supporting SAT solver selection through learning or optimization (see e.g. [Hoo12]). Formulating hyper-parameter optimization as yet another ML problem (e.g. [BBKS13]) is also a promising step toward the pervasive use of ML algorithms.

A second objective regards the **interaction between ML systems and human experts**. As already mentioned, an army of human experts is behind some major achievements of ML systems, particularly those related to natural language processing such as Watson [Bak11] or NEL [CBK+10]. In the domain of NLP indeed human help is needed to overcome the lack of understanding of the machine. Recall that in the early days of Lenat's AM discovery system [Len82], the author likely had to manually select among the host of theories/hypotheses generated by the system, those which "made sense", in order to enable the discovery process to go on. In other domains, the expert's help is occasionally provided through priors, guiding the ML search to find satisfactory results. Determining effective priors however remains an art, all the more so as negative results are rarely if ever reported in the literature.

More generally, it might be thought that in many domains the distinction between relevant and true-but-useless results is a human (and possibly individual) matter. This distinction can hardly be formally specified − just as pattern recognition eludes formal specifications − but it can be learned by interacting with the expert, through the preference learning setting [TJHA05,VB10,CC11]. This interaction would thus teach the ML system *what* is interesting to learn in the current context − user, data, moment − thus paving the way toward a better awareness of the learning means and ends. Interestingly, this research avenue echoes Alan Turing's vision of steering ML through feedback and rewards, (*carrying through the organization of an intelligent machine with only two interfering inputs, one for pleasure or reward, and the other for pain or punishment*).

A third objective regards **ubiquitous ML**, i.e. the deployment of learning components in the physical and computational world. While ML algorithms are mostly used hitherto for well circumscribed tasks, we need to elevate our scope from machine learning to *machine reasoning* [Bot11]. Since computational systems tackle tasks that are increasingly harder to fully specify, they need ML primitives to palliate the lack of specifications and provide robust decision support, through e.g. approximating computationally heavy criteria, querying additional information, anticipating and preventing the system crashes and more generally enforcing some quality of service and displaying situated awareness. Such a shift from one-shot to repeated and all-purpose learning will modify in depth the requirements on learning algorithms, from expected to worst-case performance. The pervasive use of robust ML components within computational systems will announce a new era for AI: all the more effective for being invisible.

Acknowledgments

I thank the European *Pattern Analysis, Statistical Modelling and Computational Learning* (PAS-

[10] A great many open ML environments have been proposed in the last years, such as Weka (www.cs.waikato.ac.nz/ml/weka/), CLOP (http://clopinet.com/CLOP/), RapidMiner, (www.rapidminer.com) or scikit-learn (http://scikit-learn.org/) to name a few, and many toolboxes and implementations of new algorithms are available through the *Machine Learning Open Source Software* (http://jmlr.csail.mit.edu/mloss/).

CAL) Network of Excellence (FP7) and particularly the members of the PASCAL Steering Committee for sharing many thoughts about ML future.

References

[ACBF02] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002.

[Ach01] D. Achlioptas. Database-friendly random projections. *ACM Symposium on the Principles of Database Systems*, page 274281, 2001.

[AG97] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588, 1997.

[Ama77] S.-I. Amari. Neural theory of association and concept formation. *Biological Cybernetics*, 26:175–185, 1977.

[AN04] P. Abbeel and A.Y. Ng. Apprenticeship Learning via Inverse Reinforcement Learning. In C. E. Brodley, editor, *Proc. of Int. Conf. on Machine Learning*, volume 69 of *ACM Intl Conf. Proc. Series*. ACM, 2004.

[Bak11] S. Baker. *Final Jeopardy: Man vs. Machine and the Quest to Know Everything*. Houghton Mifflin Harcourt, 2011.

[BB07] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems*, 2007.

[BBKS13] R. Bardenet, M. Brendel, B. Kégl, and M. Sebag. Collaborative hyperparameter tuning. In S. Dasgupta and D. McAllester, editors, *Proc. of Int. Conf. on Machine Learning*, 2013.

[BDHL04] A. Beygelzimer, V. Dani, T. P. Hayes, and J. Langford. Reductions between classification tasks. *Electronic Colloquium on Computational Complexity (ECCC)*, (077), 2004.

[BDvLSTT05] S. Ben-David, U. von Luxburg, J. Shawe-Taylor, and N. Tishby, editors. *Theoretical Foundations of Clustering*. NIPS Workshop, 2005.

[Ben09] Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

[BFOS84] L. Breiman, J. Friedman, R. Olshen, and C. J. Stone. *Classification and regression trees*. Wadsworth Statistics/Probability Series. Wadsworth Advanced Books and Software, Belmont, CA, 1984.

[BGV92] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proc. of Computational Learning Theory Conference (COLT)*, pages 144–152, 1992.

[Bis06] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer Verlag, 2006.

[Bis11] C. M. Bishop. Embracing uncertainty: Applied machine learning comes of age. In D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, editors, *Machine Learning and Knowledge Discovery in Databases, Part I*, volume 6911 of *Lecture Notes in Computer Science*, page 4. Springer Verlag, 2011.

[BLPL06] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, pages 153–160. MIT Press, 2006.

[BN96] R. Byron and C. Nass. *The Media Equation*. Cambridge University Press, 1996.

[BNJ03] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[Bod90] M. Boden, editor. *The Philosophy of Artificial Intelligence*. Oxford University Press, 1990.

[Bol10] D. Bollier. The promise and peril of big data. Technical report, The Aspen Institute, 2010.

[Bot11] L. Bottou. From machine learning to machine reasoning. *CoRR*, abs/1102.1808, 2011.

[Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[BT96] D.P. Bertsekas and J.N. Tsitsiklis. *Neurodynamic Programming*. Athena Scientific, 1996.

[CBK+10] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr., and T. M. Mitchell. Toward an architecture for never-ending language learning. In M. Fox and D. Poole, editors, *National Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2010.

[CC11] O. Chapelle and Y. Chang. Yahoo! learning to rank challenge overview. *Journal of Machine Learning Research, Proceeding Track*, 14:1–24, 2011.

[CGB07] S. Calinon, F. Guenter, and A. Billard. On Learning, Representing and Generalizing a Task in a Humanoid Robot. *IEEE transactions on systems, man and cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, 37(2):286–298, 2007.

[Cla93] W.J. Clancey. Situated action: A neuropsychological interpretation response to Vera and Simon. *Cognitive Science*, 17:87116, 1993.

[Cri09] N. Cristianini. Are we there yet? In W. L. Buntine, M. Grobelnik, D. Mladenic, and J. Shawe-Taylor, editors, *Eur. Conf. on Machine Learning and Knowledge Discovery in Databases*, volume 5781 of *Lecture Notes in Computer Science*. Springer Verlag, 2009.

[CRT06] E. J. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59:1207–1223, 2006.

[Cun87] Y. Le Cun. *Modèles connexionnistes de l'apprentissage*. PhD thesis, Université Pierre et Marie Curie, Paris VI, 1987.

[CW08] R. Collobert and J. Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In W. W. Cohen, A. McCallum, and S. T. Roweis, editors, *Proc. of Int. Conf. on Machine Learning*, volume 307 of *ACM International Conference Proceeding Series*, pages 160–167. ACM, 2008.

[Die98] T.G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 1998.

[DL08] C. Dimitrakakis and M. G. Lagoudakis. Rollout sampling approximate policy iteration. *Machine Learning*, 72(3):157–171, 2008.

[dST03] V. de Silva and J. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, pages 721–728, 2003.

[Efr82] B. Efron. The jackknife, the bootstrap, and other resampling plans. *CBMS-NSF Regional Conf. Series in Applied Mathematics*, 38, 1982.

[FD07] B.J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[FGL12] J. Fürnkranz, D. Gamberger, and N. Lavrac. *Foundations of Rule Learning*. Springer Verlag, 2012.

[FS96] Y. Freund and R.E. Shapire. Experiments with a new boosting algorithm. In L. Saitta, editor, *Proc. of Int. Conf. on Machine Learning*, pages 148–156. Morgan Kaufmann, 1996.

[FSZ10] C. Furtlehner, M. Sebag, and X. Zhang. Scaling analysis of affinity propagation. *Phys. Rev. E*, 81(6), 2010.

[GCDF09] C. Goutte, N. Cancedda, M. Dymetman, and G. Foster. *Learning machine translation*. MIT Press, 2009.

[GE03] I. Guyon and A. Elisseeff, editors. *Special issue on Variable and feature selection*. Journal of Machine Learning Research, 2003.

[Grü07] P.D. Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

[GS07] S. Gelly and D. Silver. Combining online and offline knowledge in UCT. In Z. Ghahramani, editor, *Proc. of Int. Conf. on Machine Learning*, pages 273–280. ACM, 2007.

[Has87] J. Hastad. *Computational limitations of small depth circuits*. PhD thesis, Massachusetts Institute of Technology, 1987.

[Hoo12] H. H. Hoos. Programming by optimization. *Commun. ACM*, 55(2):70–80, 2012.

[HOT06] G.E. Hinton, S. Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Conputation*, 18:1527–1554, 2006.

[Hsu02] F.-H. Hsu. *Behind Deep Blue: Building the Computer that Defeated the World Chess Champion*. Princeton University Press, 2002.

[KKBG10] G. Konidaris, S. Kuindersma, A. Barto, and R. Grupen. Constructing skill trees for reinforcement learning agents from demonstration trajectories. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, pages 1162–1170, 2010.

[KM90] Y. Kodratoff and R.S. Michalski. *Machine Learning: an artificial intelligence approach*, volume 3. Morgan Kaufmann, 1990.

[KRO+09] R.D. King, J. Rowland, S. Oliver, M. Young, W. Aubrey, E. Byrne, M. Liakata, and M. Markham. Make way for robot scientists. *Science*, 325(5943):945945, 2009.

[KS06] L. Kocsis and C. Szepesvári. Bandit based Monte-Carlo planning. In J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, editors, *Eur. Conf. on Machine Learning*, pages 282–293. Springer Verlag, 2006.

[KTSJ12] A. Kleiner, A. Talwalkar, P. Sarkar, and M. I. Jordan. The big data bootstrap. In J. Langford and J. Pineau, editors, *Proc. of Int. Conf. on Machine Learning*, 2012.

[LBZM06] H. Lipson, J. C. Bongard, V. Zykov, and E. Malone. Evolutionary Robotics for Legged Machines: From Simulation to Physical Reality. In *IAS*, pages 11–18, 2006.

[Len82] Douglas B. Lenat. The nature of heuristics. *Artificial Intelligence*, 19(2):189–249, 1982.

[LR85] T. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6:4–22, 1985.

[LSS01] M. L. Littman, R. S. Sutton, and S. P. Singh. Predictive representations of state. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, pages 1555–1561. MIT Press, 2001.

[MBPS10] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

[MCM83] R.S Michalski, J.G. Carbonell, and T.M. Mitchell. *Machine Learning: an artificial intelligence approach*, volume 1. Morgan Kaufmann, 1983.

[MCM86] R.S Michalski, J.G. Carbonell, and T.M. Mitchell. *Machine Learning: an artificial intelligence approach*, volume 2. Morgan Kaufmann, 1986.

[MDR94] S. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19:629–679, 1994.

[Mei05] M. Meila. Comparing clustering - an axiomatic view. In L. De Raedt and S. Wrobel, editors, *Proc. of Int. Conf. on Machine Learning*, pages 577–584, 2005.

[Mic83] R.S. Michalski. A theory and methodology of inductive learning. In R.S Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine Learning: an artificial intelligence approach*, volume 1, pages 83–134. Morgan Kaufmann, 1983.

[Mit82] T.M. Mitchell. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.

[MP43] W. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 7:115 – 133, 1943.

[MT97] H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, 1997.

[NR00] A.Y. Ng and S. Russell. Algorithms for inverse reinforcement learning. In P. Langley, editor, *Proc. of Int. Conf. on Machine Learning*, pages 663–670. Morgan Kaufmann, 2000.

[NSS59] A. Newell, J.C. Shaw, and H.A. Simon. Report on a general problem-solving program. In *Proc. of the Int. Conf. on Information Processing*, page 256264, 1959.

[O'R06] J. Kevin O'Regan. How to build consciousness into a robot: The sensorimotor approach. In M. Lungarella, F. Iida, J. C. Bongard, and R. Pfeifer, editors, *50 Years of Artificial Intelligence*, volume 4850 of *Lecture Notes in Computer Science*, pages 332–346. Springer Verlag, 2006.

[PB07] Rolf Pfeiffer and Josh Bongard. *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT Press, 2007.

[Pea91] J. Pearl. *Probabilistic reasoning in Intelligent Systems: Networks of plausible inference*. Morgan Kaufmann, 1991.

[Pea00] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2000.

[PS08] J. Peters and S. Schaal. Reinforcement Learning of Motor Skills with Policy Gradients. *Neural Networks*, 21(4):682–697, 2008.

[Qui86] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[RFKM08] L. De Raedt, P. Frasconi, K. Kersting, and S. Muggleton, editors. *Probabilistic Inductive Logic Programming - Theory and Applications*, volume 4911 of *Lecture Notes in Computer Science*. Springer Verlag, 2008.

[Ris78] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[RM86] D.E. Rumelhart and J.L. McClelland. *Parallel Distributed Processing*. MIT Press, 1986.

[RN95] S. Russell and A. Norwig. *Artificial Intelligence, a modern approach*. Prentice Hall, 1995.

[Ros58] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386408, 1958.

[RS00] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[Rus06] F. Russo. Frequency-driven probabilities in quantitative causal analysis. *Philosophical Writings*, 32:32–49, 2006.

[RW06] C. E. Rasmussen and C. K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 01 2006.

[Sam60] Arthur L. Samuel. Programming computers to play games. *Advances in Computers*, 1:165–192, 1960.

[SB98] R.S. Sutton and A. G. Barto. *Reinforcement learning*. MIT Press, 1998.

[SBS98] B. Schölkopf, C. Burges, and A. Smola. *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1998.

[Sch90] R.E. Schapire. The strength of weak learnability. *Machine Learning*, 5:197, 1990.

[SGC11] L. Saitta, A. Giordana, and A. Cornuéjols. *Phase transitions in Machine Learning*. Cambridge University Press, 2011.

[Sur04] J. Surowiecki. *The Wisdom of Crowds*. Random House, 2004.

[Sze10] C. Szepesvári. *Algorithms for Reinforcement Learning*. Morgan & Claypool Publishers, 2010.

[TBF05] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[Tes02] G. Tesauro. Programming backgammon using self-teaching neural nets. *Artificial Intelligence*, 134(1-2):181–199, 2002.

[TJHA05] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

[TPB99] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[Tur50] A.M. Turing. Computing machinery and intelligence. *Mind*, 59, 1950.

[Val84] L.G. Valiant. A theory of the learnable. *Communication of the ACM*, 27:1134–1142, 1984.

[Vap95] V. N. Vapnik. *The Nature of Statistical Learning*. Springer Verlag, 1995.

[VB10] P. Viappiani and C. Boutilier. Optimal Bayesian recommendation sets and myopically optimal choice query sets. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, pages 2352–2360, 2010.

[VGCB10] R. Vilalta, C. G. Giraud-Carrier, and P. Brazdil. Meta-learning - concepts and techniques. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 717–731. Springer Verlag, 2010.

[VV06] R. Vert and J.-P. Vert. Consistency and convergence rates of one-class SVMs and related algorithms. *Journal of Machine Learning Research*, 7:817–854, 2006.

[WBL05] K. Weinberger, J. Blitzer, and S. Lawrence. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems*, pages 1473–1480. MIT Press, 2005.

[Wei66] J. Weizenbaum. Eliza a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

[Wid62] B. Widrow. Generalization and information storage in networks of Adaline neurons. In M.C. Yovits, G.T. Jacobi, and G.D. Goldstein, editors, *Self-Organizing Systems*. Spartan Books, 1962.

[Win75] P.H. Winston. Learning structural descriptions from examples. In P.H. Winston, editor, *The Psychology of Computer Vision*, pages 157–209. Mc Graw Hill, New York, 1975.

[XHHLB08] L. Xu, F. Hutter, H. H. Hoos, and K. Leyton-Brown. SATzilla: Portfolio-based algorithm selection for SAT. *J. Artif. Intell. Res.*, 32:565–606, 2008.