Final.

Machine Learning: A Survey

A. Cornuéjols and M. Moulet

Laboratoire de Recherche en Informatique (L.R.I.)
Bât 490, Université Paris Sud
91405 Orsay Cedex (France)

1 Introduction

Intelligence and learning are intimately connected. They need each other to be at their peak. This is why, since its inception in the fifties, Artificial Intelligence has been preoccupied with the study of learning, as testified by the pioneering works devoted to the first cybernetic "turtles" or "mouses", or the CHECKERS program [1]. Aside from this fundamental interest in learning that, in a way, dates back to the Greek philosophers, there are more practical reasons why "intelligent" systems should be endowed with learning capacities. Numerous tasks are indeed intrinsically quite difficult to program, if only because they imply seemingly infinite numbers of unpredictable situations. For example, in the domain of pattern recognition or obstacle avoidance, it is impossible to enumerate all possible cases. Likewise, the knowledge acquisition phase has long been identified as a major problem in the development of knowledge-based systems. The prospect of a machine capable of automatically acquiring the competencies needed to face up new and unknown situations is therefore quite attractive. Furthermore, there is a need for systems that can adapt in face of changing environments, improve their performance and update their knowledge while carrying on their duties. Here too, learning is required, possibly involving longer timescale. Finally, there are activities like scientific discovery that are learning problems in themselves. For all these reasons, machine learning is an active field of research experiencing a vigorous development.

This chapter is intended to provide an introductory survey of some achievements in machine learning and of the main issues that are currently of central concern. Section 2 aims at giving a taste for what machine learning is : its historical development, the main concepts in use, typical learning methods, and some research topics. The next section reviews the relationships between machine learning and two applied fields that are also concerned with learning of some sort : knowledge acquisition and statistics. Machine learning in practice is the subject of section 4 that discusses specific concrete problems and examples of applications. Section 5 concludes on the major trends of the field and directions for future developments.

2 What is Machine Learning ?

Like intelligence, *learning* is an elusive concept that cannot be encapsulated in a definition. Simon in [2] proposed that "*learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more effectively the next time*". Such a definition, which focuses on performance improvement, limits learning to a change of behavior and excludes other learning activities like knowledge acquisition in which the effectiveness criterion is difficult to define. Short of giving a full perspective of machine learning, an ideal that is beyond the scope of this chapter, we give in the following a glimpse of its history, and define basic concepts, methods and issues.

2.1 *History and background*

Ignoring pre-computer history, the study of machine learning has developed along tracks similar to those followed by artificial intelligence. A first phase ending in the sixteen's was characterized by the invention and experimentation of numerous "paradigms": simulated evolution, neural networks, reinforcement learning or similarity based learning. During the following 1970-1980 period, the symbolic approach got preeminence (in parallel with the development of expert systems) and various learning systems were developed, often inspired by concerns for cognitive modeling (e.g. ACT [3]), and quite ambitious in that they dealt with structured representation knowledge bases (e.g. ARCH [4]) and made high claims on the nature of learning (e.g. AM [5]).

The next decade has witnessed a tremendous increase in the amount of works (and workers) aimed at investigating a wide range of basic learning methods. The basic concepts of the field have been established, and a solid theoretical basis has progressively been elaborated. Most nowadays available learning tools result from this euphoric period. It is noteworthy that it is also at that time that sub-symbolic, biologically inspired, methods, like connectionism, sprang up again and gained a wide popularity (in conjunction partly with a new interest in self-organizing systems and non-linear dynamics, and partly with the advent of massively parallel architectures). More recently, interest has focused on the integration of multiple learning strategies within larger problem-solving systems, on real-world and real-size challenges like data mining, and on making contact between the theory and the practice of machine learning.

2.2 *Characterizations of Machine Learning*

Machine learning situations can be characterized in many ways depending on which aspect is of central concern. Accordingly, there does not exist a single all-encompassing taxonomy that allows to organize all of Machine Learning, but rather several ones that provide different and complementary views on learning.

For instance, if an external point of view is chosen, one can either focus on the *learning task*, or on the types of *interactions* allowed between the learner and the environment. These aspects are generally underlined by end-users of machine learning systems. On the other hand, computer scientists that design those systems will prefer internal descriptions of learning that involve fundamental *forms of learning* and basic *mechanisms*. These different points of view are refined in turn in the following.

- **Learning tasks**

It is worth noting that the learning task is generally left implicit in a learning system. It is determined both by the type of input data (containing class description or not, in the presence of explicit background knowledge or not, and so on) and by the learning goal (e.g. classification, efficiency improvement, ...). Learning goals condition what parts of prior knowledge are relevant, what knowledge is to be acquired and in which form, how the learned knowledge is to be evaluated, and when to stop learning. This is why it is so important to accurately define what is expected from a learning system.

Learning goals include: concisely describing and/or generalizing a given observation set, discovering regularities in a collection of facts, finding a causal explanation to some identified regularity, acquiring control knowledge to perform a given activity, reformulating knowledge into a more effective form, giving confirmation to a particular piece of information, and so on. Human learners generally pursue several goals simultaneously, that sometime may be in conflict. Some planning activity may then be required in addition to the learning process. This is still not a primary concern in machine learning research even though the study of the so-called multistrategy learning (see below) has started to investigate such issues.

The following distinctions are customary in machine learning:

- *Classification* or discrimination tasks. They correspond to the induction of concepts or classes from the observation of collections of specific instances of these

concepts. For instance, in medicine, it is possible to use descriptions of patients and of their associated diagnosis in order to induce decision rules allowing to predict the diagnosis of new incoming patients. Each decision rule corresponds then to a class of symptoms related to a diagnosis. When the task is to learn a single concept (for instance "appendicitis"), that is to discriminate positive instances of one concept against negative ones (called counter-examples), one speaks naturally of a *discrimination* task.

- *Concept formation* or categorization. In contrast to classification tasks where the learner is provided with descriptions of instances that include the class, concept formation is the task of finding underlying regularities, like clusters of shared properties or dependency laws, in data that do not include any predefined class attribute. For instance, one machine learning system proposed a new, insofar unsuspected, family of galaxies from masses of observational data, and the experts agreed that this family is associated with interesting set of properties.

In *machine discovery*, concept formation is oriented towards the expression of comprehensive and understandable scientific concepts and theories. For instance, a task can be to discover the chemical composition of molecules from the description of their reactions. One sub-problem, interesting in its own right, is the discovery of numerical relationships (e.g. equations) in the data. In the past, this has been improperly called "scientific discovery" in the machine learning community.

- *Speed-up learning*. It may happen that one is not directly interested in the disclosure or acquisition of new knowledge, but rather in increasing the performance of a system through learning by experience. This is called speed-up learning. It generally involves the transformation and reformulation in more operationalized ways of the knowledge incorporated in the system.

• **Types of interactions** between the learner and the environment

Three main types of interaction are distinguished depending on the amount of information provided by the "teacher" to the learning system. *Supervised learning* occurs when the teacher provides both inputs and desired outputs (e.g. descriptions of examples and of their associated class in classification tasks). When no information is given regarding the desired output, this is called *unsupervised learning*, as is the case in concept formation. Finally, a teacher may only provide the learner with a one dimensional signal, analogous to pain or pleasure, in order to draw the system towards certain behaviors and away from others. This is called *reinforcement learning* like in psychology.

It is furthermore possible to look at the autonomy of the learning system while learning. *Passive learners* wait for data to be provided by the teacher or the environment while *active learners* can ask questions or make experiments.

Finally, one interesting dimension consists in the timescale on which learning takes place. Most learning systems to date are *one time learners* that acquire data "en masse" (they are also called one-shot or batch learners) . However, many learning situations imply incremental or *on-line learning* where data is acquired sequentially over time. Among another issues, convergence problems and the difficulty of learning in the presence of drifting concepts or conditions are then to be faced.

- **Forms of learning.**

While it is impossible in practice to disentangle one from the other, it can be profitable to distinguish between learning through the *increase of knowledge* and learning through the *transformation of the representation* of knowledge.

Increase of knowledge is measured by the variations of its deductive closure (the set of all facts that can be deduced from the knowledge base). This can be achieved either by the incorporation of *new knowledge*, through active or passive modes, or through *forgetting* of obsolete or incorrect information. So far, little research has been devoted to the latter, and most of the attention has been attracted to the former.

Alternatively, learning can be seen as a *transformation of some representation of knowledge to another one*, and it can be argued that this is all there is to learning, in that no new knowledge is ever really produced. For instance, it will be seen in section 2.4 below that inductive learning from examples does in fact *reformulate or operationalize some prior knowledge* incorporated in the learner to accommodate the observation of specific data from the environment.

The transformations of knowledge can be analyzed in terms of truth-preserving properties. A transformation is *truth-preserving* if its result must be true given that the knowledge provided as input is true. Obviously, truth-preserving transformations do not affect the deductive closure. In machine learning, they include: *explanation-based learning*, *knowledge compilation*, *construction of macro-operators*, and inversions of these (e.g. decompilation) in the case where comprehensibility is the main goal. Learning by truth-preserving transformation may serve the goal of improving the efficiency of the overall system (speed-up learning) or the comprehensibility of the represented knowledge. Conversely, *not truth-preserving transformations* produce results that go beyond the deductive consequences of the

prior knowledge and the input, and thus cannot be guaranteed to be correct. They are responsible for "increase of knowledge" in learning. Non-deductive transformations are intrinsically not-truth preserving. They include *inductive inferences*, when one is drawing general concepts from specific instances, *abductive reasoning*, when one is choosing one possible cause among several possible ones to explain some phenomenon, and *transductive inferences*, when one is inferring from some particular case properties for another particular case, like in analogy reasoning. Most learning methods like inductive *learning from examples* (e.g. classification), *concept formation*, *abductive reasoning*, *learning by analogy* and *knowledge revision* are of this kind.

2.3 *A survey of methods of interest*

• METHODS FOR INDUCTIVE GENERALIZATION

In the following, we draw a distinction between symbolic and non symbolic methods. Roughly, the former make use of explicit background knowledge and inference strategies, and produce, at least to some extent, interpretable outputs. The latter rely more heavily on numeric and statistical techniques. They are generally more robust to noise in the data and other various corruptions, but they tend to produce opaque results (e.g. an array of weights in a neural network).

Symbolic methods:

Generalization in a concept space structured by a representation language

Historically, generalization in a concept space defined by a representation language has been one of the first form of learning studied. The idea, inspired in part by research in expert systems, is to search "intelligently" a space of possible hypotheses (also called the *version space*) by defining search operators corresponding to generalization and specialization operations. The Candidate Elimination algorithm in Version Space [6], and the AQ algorithm [7] are well-known pioneering instances of these methods, but lots of subsequent works have resulted in numerous realizations.

Inductive logic programming (ILP)

ILP methods are intended to be able to deal with relational knowledge, thus overcoming the limitations of zeroth order languages. They are in fact descendants of early automatic program synthesis techniques developed in the late 70s and early 80s. The learning problem in ILP is normally stated as follows: given background knowledge B, expressed as a set of predicate definitions, positive examples $E+$, and negative examples $E-$, construct a predicate logic formula H such that all the examples in $E+$ can be logically derived from B & H, and no negative example in $E-$ can be logically derived from B & H. Typically, B, H, $E+$ and $E-$ will each be Prolog programs. The advantage of the ILP approach is to allow the use of more expressive forms of knowledge. On the other hand ILP systems are still relatively inefficient and poorly equipped to handle numerical data. ILP is presently mostly studied in Europe, and has given birth to a wide range of systems and some applications (see [8] and [9] for more information).

Non symbolic methods:

Neural networks

Neural networks, also called connectionist techniques, encompass many different learning paradigms: supervised, unsupervised, associative, competitive, and so on. The algorithm that is the most widely used and is responsible for most of the hype is the *back-propagation algorithm* or multi-layer perceptron. This can be seen as a kind of statistical parametric method with a large set of parameters: the weights of the links between the units. The idea is to learn a relation from inputs to outputs through a supervised learning phase that modifies the weights in the network. If learning is correctly taking place, the resulting network is implementing a generalized function from the input to the output space and may be used to make predictions on further instances. There are considerable development and theoretical works on this family of algorithms, and many applications have resulted (see for instance [10] for an entry point in the literature).

Induction of decision trees

The idea of tree induction is to construct a decision tree from a set of labeled examples described by attribute-based vectors. Learning algorithms (such as ID3 [11] or C4.5 [12]) usually carry out a greedy search through the space of

decision trees, typically using a statistical evaluation function to select attributes for incorporation into the knowledge structure. Most methods partition the training data recursively into disjoint sets, attempting to summarize each set (or leave in the tree) as a conjunction of logical conditions. The advantages include the fact that there exist very efficient algorithms and that the resulting decision trees can usually be interpreted by experts and be easily incorporated in rule-based systems. The simplicity of these methods are in part responsible for their popularity.

Case-Based Reasoning (CBR) instantiates a kind of transductive inference akin to analogy. We put it here since it can be considered as a generalization method that lay between symbolic and non symbolic methods.

CBR belongs to *Instance-Based Learning* methods (IBL) [13]. In these methods, knowledge is represented in terms of specific cases or experiences and one relies on flexible matching techniques to retrieve the relevant cases and apply them to the current problem or situation. One common scheme, known as nearest neighbor, simply finds the stored case nearest (according to some distance metric) to the current situation. In CBR, where cases are generally described through structures representation languages, more sophisticated indexing schemes or similarity metrics can be used, that determine in part the generalization power of the method. ([14] is a good introduction).

- **METHODS FOR UNSUPERVISED CONCEPT FORMATION**

Two main approaches are used for concept formation. *Divisive techniques* recursively split the data set into sub-parts, also called clusters, according to some measure that generally tries to maximize intra-similarity within each cluster while maximizing the dissimilarities among them, so that they correspond to well identified and coherent categories. The Cobweb system of Fisher (1987) [15] is a good example of this approach. One of its assets is to produce a hierarchy of categories instead of a flat partitioning. *Agglomerative techniques*, on the other hand, start from preselected instances that serve as seeds and then they "grow" clusters of increasing sizes around these using special similarity metrics. Because of their sensitivity on the initial choice of seeds, these methods may require several passes before convergence on a satisfactory partitioning of the data is obtained. Many variants of these two approaches exist, depending in particular on the knowledge representation language (zeroth or first order, with probabilities or not, and so on).

• METHODS FOR OPTIMIZATION GOALS

Evolutionary computing

These methods are inspired from evolutionary mechanisms in biology. Evolution seems to "produce" species closely fitted to their environment through two main mechanisms: one is a generator of diversity or experiments (mutation and cross-over operators are mostly responsible for genotype modifications), the other is a selector of the best individuals according to the requirements of the environment. By iteratively combining these two mechanisms over many generations, evolutionary methods are able to optimize populations of individuals for some task specified by a fitness criteria. *Genetic Algorithms*, for instance, evolve populations of bit-strings that represent some "solutions" to given problems. A whole research community is very actively working on these ideas. (see e.g. [16]).

Reinforcement learning

Reinforcement learning methods are mostly used in the context of agents learning to become efficient from trial-and-error experiences. The challenge here is to learn appropriate reflexes or state-action relations from experiences that consist in sequences of <state, action> pairs with occasional rewards or punishments. Various learning techniques have been proposed and experimented that tend to gradually, through back-propagation of rewards and penalties along sequences, associate to each relevant state region in the state space a suitable action. Because reinforcement learning corresponds to many realistic situations and many actual problems, it is the object of a steadily increasing research activity. (see [17]).

• METHODS FOR CHANGE OF REPRESENTATION

Explanation-Based Learning (EBL)

The purpose of EBL is to speed-up problem-solving in some specific domain area where typical sub-problems are to be met time over time (e.g. to attach firmly two mechanical parts). One solution is to cache memory of *<problem: x, solution: f(x)>* pairs. EBL does indeed maintain a cache memory, but generalizes *<x, f(x)>* pairs before storing them. Generalization employs a proof procedure that tries to "explain" why *f(x)* is a solution to *x* (hence the term "explanation-based"), and then retains the necessary conditions found in

this proof to describe families of situations comparable to *x* in that solutions adapted from *f(x)* can solve them. These new pairs *<family: X, f(X)>* are then stored for future retrieval. This type of technique (related to partial evaluation) is very interesting, but it requires a complete or nearly complete background theory to allow proof procedures. Except in some "learning apprentice" systems of academic interest, it has therefore not been applied yet to real-world problems (see [18]).

Constructive learning

It consists in a dynamic redefinition of the hypothesis representation language, particularly through the invention of new terms, according to the needs emerging during learning. There has been some pioneering works in this area, but much remains to be done.

2.4 *Fundamental issues in designing a learning situation*

As can be inferred from the above multiple points of view on learning, numerous design choices have to be made in order to define a learning situation depending on the goal pursued, the interaction with the environment, and the learning mechanism(s) adopted. This section is intended both to look deeper into fundamental issues and to focus on some specific and practical problems of concern to want-be practitioners of machine learning. We have chosen to consider particularly inductive learning from examples since this is currently the learning situation most widely encountered in the process of developing a knowledge-based system. In addition, it has been extensively studied, both on philosophical grounds ever since the Greek philosophers, and on a theoretical basis mostly during the last decade, and it brings to attention lots of questions of practical impact. Other problems of interest that deserve special attention include the choice of knowledge representation and the methods to automatically change it, the cooperation between different learning techniques, also called multistrategy learning, and, last but not least, the evaluation of the output of learning.

- **Inductive learning from examples.**

One famous and unsettling finding in the study of induction is that induction is an impossible task ! Given a set of observations, the so-called *training set*, and in the absence of any prior guess or bias, any prediction regarding future events, be it the next element in a sequence or the class of an as yet unseen pattern, is worth what a

random guess is. Put in more formal terms, this so-called "no-free-lunch-theorem" [19] has stressed the importance and necessity of prior bias or knowledge in induction. There needs to be some prior assumption about the form of the target concept (e.g. a polynomial of degree at most 5) in order that induction be possible. In a way, induction is the art of reformulating some unspecified prior information into operationalized knowledge through the observation of specific and supposedly representative instances from the environment of the learner. Therefore, apart from the problem of evaluating the resulting uncertain knowledge (see below), the main issue becomes: *what kind of prior knowledge can be put in a learning system?*

To most researchers familiar with knowledge-based systems, this question would naturally suggest explicit knowledge, either declarative, similar to metarules, or procedural. Except in part in the works on *theory revision*, this type of explicit prior knowledge has not been central to investigations concerning machine induction. Rather, theoretical studies, mostly in the last decade, have put the emphasis on implicit _in the sense that it is not directly operational_ prior knowledge expressed as constraints that restrict and structure the space of potential hypotheses (also called the *"version space"*) or that provide some preference criteria among the candidate hypotheses.

More precisely, prior knowledge is analyzed in terms of representation bias and preference bias. The *representation bias* mainly corresponds to the restriction to the hypotheses induced by the language used to define it. Thus, if the learner is restricted to use conjunctive normal forms (e.g. patterns that are `red & heavy & costly`), it cannot consider hypotheses of a disjunctive nature (e.g. patterns that are `{red & heavy} or {black & large}`). In this way, the learner can no longer entertain hypotheses corresponding to every partitions of the instance space, but only a subset of these. Induction, that is the production of some hypothesis or generalization, is therefore restricted to the exploration of a limited space. Various measures, for instance the so-called *capacity* or the *Vapnik-Chervonenkis dimension*, have been defined to evaluate the restriction imposed on the hypothesis space. They are pivotal in the study of the convergence of the learning process (see below) and are deeply related to the no-free-lunch theorem.

If only a representation bias was imposed, an inductive engine would return all hypotheses or generalizations consistent with the training instances (i.e. more general than these, and not contradicted by any negative instances). This is often not what is needed. This is why automatic learners are generally endowed with a *preference bias* whose effect is to select one or a few hypotheses among the set of correct candidate ones. There are mostly two types of preference bias. One is simply

a criterion that allows to sort candidate hypotheses. One such criterion could favor simpler hypotheses (measured according to some syntactic measure) over more complex ones. The other type of preference bias is implemented as control strategy that directly guides the search in the version space. For instance, such a strategy could be to consider first generalization operators that involve dropping conjuncts.

In general, preference biases can be analyzed as kinds of density (preference) measures imposed on the version space while declarative biases define the topology of this space. Both biases are expressions of prior knowledge *incorporated in the learner* that allow it to choose inductive hypotheses. Prior knowledge about a learning task can also be provided *from the outside* through an adequately chosen sequencing of the input data to the learner, much as a teacher does to help his or her pupils in a classroom. The study of these *sequencing effects* and their potential as an help for machine learning is still in its infancy.

Now, learning to be effective must have some ***convergence*** properties: whether measured as the number of training inputs necessary before identification of good hypotheses be possible (the so-called *sample complexity* of learning), or measured as the computational, and/or memory space, complexity of the task that should rather be reasonable (often taken as meaning polynomial in some key parameters).

Before studying convergence however, it is necessary to define the notion of distance. In the case of inductive concept learning, this means a *measure of proximity between concepts* or hypotheses. Three definitions have been proposed and investigated. The first one equates proximity with identity. Learning is thus seen as *identification in the limit* of the target concept. Lots of interesting theoretical results have been obtained in this setting, but it remains remote of realistic concerns, if only because, for instance, it leaves no space for learning from corrupted data. The second conception of proximity defines it as the probability of making a mistake on a future instance in comparison to the true target concept. When, in addition, only convergence in probability is considered _because it is unrealistic to demand that convergence be guaranteed for every sequences of data, however bad they be_, this gives rise to the celebrated *PAC (Probably Approximately Correct) learning* paradigm that has been extensively studied in *COLT (COmputational Learning Theory)*. A third definition of proximity between concepts refines the previous one by incorporating the notion of cost or risk of a mistake. This has been mostly studied within the bayesian framework (e.g. [20]).

The most important result in the theoretical study of induction relates the rate of convergence of a learning process from examples to the representation bias or the constraints on the hypothesis space. Stated qualitatively, it says that:

If one has chosen a hypothesis space H such that for any sequence of data from the environment it is possible to find in H an hypothesis that is close enough from the best possible one (e.g. one that would perfectly classify all training instances),

and if H is sufficiently restricted (as measured for instance by the VC dimension mentioned above),

then, with high probability, the hypothesis found after considering the training sequence will be good on further instances (i.e. close to the target concept),

and the convergence of the learning process will be all the more fast that H is more restricted.

This result, exploited as the *Empirical Risk Minimization (ERM) inductive principle*, justifies why it is a good inductive strategy to choose an hypothesis that minimizes errors on the training set. It also explains why it is important to have prior knowledge, in particular in the form of a well-tailored hypothesis space or of a representation bias.

Recent research efforts focus on the measure of bias imposed by various representation languages, they also study the convergence properties and complexity of various classes of learning methods and problems [19].

- **Evaluation.**

The theory of inductive learning provides (mostly worst-case) bounds on the sample complexity. However these bounds are usually much too pessimistic, and it is useful to have empirical means of measuring the proximity of the learned concepts to the underlying ones. The latter being by definition unknown, evaluation methods must rely on available data from the environment. When performance is measured through the probability of future mistake, also called the *error rate*, standard methods include the following:

1. Divide the available data into a *training set*, used by the learning system, and a *testing set*, used to measure the performance of the learner. The measures one obtains are unbiased estimates of the error rate, but they can be highly variable if the testing set is small, and having to use a test set may waste data which could

otherwise have been used for training. This is why the next method tends to be favored.

2. *Cross-validation*. The idea is to divide the available data into n subsets, and to realize n learning processes, each time evaluating the result on the ith subset that has not been used in learning. The error rate is then computed as the average of the n measured error rates. The extreme version of this strategy is to take n equal to the number of available data, and have the testing set reduced to only one instance. This is called the *leave-one-out estimator*. The advantage of cross-validation is to better use the available data for learning, but this is at the expense of more computation (n learning processes instead of one for the previous method).

3. If one is available, the *recourse to an expert* of the field is of course recommended. However, this often requires that the result of learning can be interpreted by the expert, which is not always the case, as for instance in neural networks methods.

- **Choice and change of representation.**

It is well-known that there exists a trade-off between the expressive power of one knowledge representation language and the computational complexity of basic operations within that language. An important dichotomy, in this respect, contrasts propositional logic (descriptions consisting of <attribute-value> pairs) and first order logic (relational) representation languages. Using a FOL language, many problems, such as matching or testing for subsumption, become computationally intractable, and even the very notion of "generality" may acquire more than one meaning. Moreover, the section on inductive learning has underlined the importance of the representation bias. As a consequence, it is crucial that the representation language be carefully chosen so that it allows enough expressiveness for the task at hand, while it keeps sufficiently restricted for computational and learning efficiency. One interesting idea with respect to the learning side of this issue is to explore version spaces of increasing complexity until a satisfactory result has been obtained (see for instance the Structural Risk Minimization inductive principle [21]).

It has also been proposed that the learner not only explore languages of increasing complexity, but also modify the representation language (also called *shift of representation*) so as to make it fit to the task at hand. Without entering into details (see [22]), this is the object of ideas like *constructive induction* where new terms are invented during learning to enrich the language and make it more operational, or like *abstraction*, broadly intended as a mechanism to build up a "simpler" representation language than the one in which the problem had been originally formulated. Much

remains to be done in these arrays because no efficient heuristics are known to identify the deep causes of the language insufficiency and to remedy it. This is the reason why most of the works on shift of representation focuses on the definition of a language of bias that the user may shift by himself.

- **Multistrategy learning.**

Sections 2.2 and 2.3 have provided a brief overview of basic learning or monostrategy methods befitted for single learning goals and single hypothesis description languages. However, most learning situations call for complex goal structures and interactions and multiple knowledge bases and representation languages. It has therefore become gradually apparent that learning techniques should be combined into complex performance systems that include also other problem-solving modules. This recent trend is reflected in a growing number of conferences devoted to this subject (see for instance [23]). Roughly, on one hand, theoretical efforts are made to analyze learning much in the way knowledge-based systems have been, at the level of inference types and problem-solving tasks, and, on the other hand, various associations of learning methods (called hybrid methods) are experimented (e.g. Genetic Algorithms and Neural Networks, supervised and unsupervised methods, etc.).

3 Machine Learning and related fields

3.1 *Machine Learning and Statistics*

Machine learning and statistics share a common goal : automatically finding structures in examples. Still, they are different. In which respects ? And to what extent do they complement each other ?

Each of these disciplines is driven by specific concerns. First, machine learning, descending from artificial intelligence, is mostly dealing with symbolic representations and logic-based types of inference. It seeks to be able to make use of explicit and structured background knowledge expressed in first-order logic form, to produce "high-level" and comprehensible outputs, and to perform goal-oriented reasoning. Statistics is traditionally more concerned with the design of robust and efficient methods that are able to deal with noisy data. Second, machine learning stresses the problem of small sample statistics appropriate when limited amount of data is available, statistics is oriented toward the analysis of abundant data sets,

allowing the use of theorems valid only in the limit of large numbers. This difference is at the origin of many new questions and methods. If less data is available, this must be offset by prior knowledge. Which knowledge ? How to use it ? How to evaluate the results ? These and other interrogations have led to fresh theoretical developments, discussed in part in section 2.4 above, regarding the nature of prior knowledge and convergence issues.

New inductive methods have also been studied. Classical statistical decision theory and pattern recognition had long established a distinction between parametric and non parametric methods. In the first ones, everything or so was known beforehand and learning was a matter of tuning a finite number of parameters in specific user-tailored models of the data. In the latter ones, density estimations stemmed from counting methods (e.g. k-nearest neighbor and Parzen window algorithms). machine learning has introduced new techniques for knowledge-based discovery of data models, and brought forth weakly parameterized approaches, such as neural network methods, that have stimulated new research works, particularly in statistics. On the other hand, machine learning has widely borrowed from statistics, especially in the domain of evaluation and validation procedures, and, in part, in the field of uncertain reasoning.

At the foundational level then, machine learning and statistics, while maintaining their distinct interests, have greatly benefited from their recent mutual attention. At a practical level, numerous studies and projects (such as the european ESPRIT STATLOG project [24]), after thorough comparisons, have underlined the complementarity between the two families of techniques and proposed classifications aimed at guiding the selection of the best suited method depending on key parameters of the task at hand, such as the number of examples, the number of attributes, the prevalence of binary attributes, the proportion of missing values, the noise rate, and so on.

Finally, it is worth mentioning that significant efforts in the last few years have been directed toward designing hybrid methods that could take advantage of the best of each world. This is a very attractive area, that should have significant applications Data mining (see section 4.3 below) is such an area where cooperation between machine learning and statistics is fruitful.

3.2 *Machine Learning and Knowledge Acquisition*

Knowledge acquisition for expert systems has been a major source of incentives for machine learning. Briefly, the knowledge base production implies three main stages: (1) Knowledge acquisition or elicitation, (2) Encoding of knowledge, and (3) Knowledge refinement.

Machine learning has a role to play in the first and last stages. *Knowledge elicitation* can be helped or performed completely using inductive techniques, particularly aimed at rules learning. For example, decision trees learning systems like ID3 [11] or C4.5 [12] have been widely used to induce set of decision rules in various domains (see also [25]). Likewise, concept formation systems can help in discovering categories and their taxonomies in ill-known domains. *Knowledge refinement*, often needed after shortcomings and errors have been detected at the outcome of some practice with the expert system, can benefit from theory revision methods and change of representation techniques developed in machine learning (particularly learning apprentice systems based on Explanation-Based learning, and theory refinement tools (see for instance [26])).

However, up to now, machine learning techniques are still not included as standard knowledge acquisition tools. This is mostly due to the fact that the rule base paradigm has gradually lost its prominent position in knowledge acquisition, and that machine learning by and large has not yet developed theories and methods adapted to complex and structured knowledge bases and representations that have become routine in knowledge base development. Studies in knowledge acquisition have both brought forth fundamental ideas, for instance that knowledge must be analyzed at various levels, and clarify the role that machine learning can play. Thus, inductive learning might be a primary tool for knowledge elicitation, specially when there are large amounts of data, no human expertise is available and there is no pressing need for an explanation facility, but it has been underlined that the knowledge resulting from the machine learning process generally needs be analyzed by human experts rather than automatically integrated into an expert system. Likewise, if machine learning can also offer support in structuring poorly understood knowledge, (for instance, in the MOBAL system [27], a *concept formation* module helps in reconceptualizing knowledge), it has been shown however, with ACKnowledge [28] and other experiments, that these techniques are no panacea. It has also been found that domain experts frequently exhibit poor performances in controlling machine learning systems, and that they often feel difficult to understand the output produced by these systems.

For all these reasons, progress is needed before machine learning techniques can be smoothly integrated within larger knowledge acquisition tools. In particular, the notions of models, generic tasks, problems solving methods and other knowledge categories should be incorporated in the design of machine learning methods. Beside easing integration with KA, this would enhance the capacity of learning systems to deal with multiple knowledge sources and representation languages, a major challenge to current machine learning technology.

4 Machine Learning in practice

The last fifteen years have witnessed a vigorously growing interest for knowledge-based systems. They are irreplaceable in many applications and are now in widespread use in all circles of administration and industry. This state of affair contributes a great deal to the motivation for the development and application of machine learning technology. The problems of manually engineering a knowledge are well-known, there is a vast potential for automatically discovering new knowledge in the recent explosion of available on-line database, too large for humans to manually sift through, and there is an obvious need for computers able to automatically adapt to changing expertise and environment. Nonetheless, if there are impressive success stories illustrating the huge benefits that can be gained through the use of machine learning, there is still a gap between expectations and actual realizations. This section illustrates some successful applications and gives an overview of various reasons why it may be difficult to go from academic techniques to real-world realizations.

4.1 *Applications of Machine Learning*

It is worth recalling first that there have been serious real-world machine learning applications, if only to dispel the skepticism sometime encountered when mentioning this possibility. It is not easy in fact to assess and document the spread of machine learning applications because many users either want to keep their developments confidential or do not see any interest in publishing about their experience. Various studies have nevertheless collected data about real-world testing of machine learning methods (see for instance [9], [16], [29], [30], [31], [32], [33], [34], [35], [36], [37]). We mention here briefly four examples.

• Increasing yield in chemical process control

The Westinghouse company in the US had a problem of prediction of the quality of uranium pellets. An approach using decision-tree induction resulted in rules that were used in an expert system that allowed increased and higher quality throughput, yielding gains estimated to more than ten million dollars per year.

• Diagnosis of mechanical devices

The Italian Enichem chemical company wanted to automatize preventive maintenance of motor pumps, identifying potential problems and determining the type of needed repair. A knowledge intensive machine learning technique was used, relying on causal knowledge gleaned in part from the expert. Experiments indicated that the learned knowledge base was more accurate than the handcrafted one, and the induced rules have now replaced the original ones in the diagnostic system.

• Sky survey : automatic classification of celestial objects

From about three terabytes of image data observed by the second Palomar Observatory Sky Survey in United States, nearly two billion sky objects had to be classified and catalogued. In the past, astronomers have catalogued objects in photographs manually, but the gigantic quantity required an automatic procedure. For this task, Fayyad, Smyth, Weir and Djogorvski proposed a machine learning approach. After iteratively refining the set of relevant numerical attributes, a decision tree induction system yielded 94% classification accuracy that was above the required one. All objects in Sky Survey are now classified automatically.

• Predicting the secondary structure of proteins

One largely unsolved problem in molecular biology involves predicting the secondary structure (spatial) of proteins from information about their primary amino acid sequences. Because this involves relational knowledge, and because background knowledge in the form of theories is available, an ILP approach was tried. After few iterations, each producing rules that were incorporated in the knowledge base, predictive rules of accuracy greater to any other known methods were obtained on four separate test proteins.

4.2 *Machine Learning in practice: lessons and strategies*

The wide experience gained through numerous fielded applications of machine learning has brought forth some lessons and suggested some strategies, not unlike the ones discovered in the field of knowledge-based development, or more generally in software production. It thus appears that successful large applications are not obtained by simply taking some machine learning tool out of the shelf, but require a thorough development process implying several crucial stages. There is no defined methodology as such, but several authors converge on approximately the same development process and recommendations. The main developmental phases are, according to them (e.g. [34]), as follows:

Analyzis and formulation of the problem. The first step is to reformulate the initial problem specification so that it, or some subproblems, can be handled by machine learning methods, specially inductive ones. This requires that an analyzis be carried out examining in turn the application domain (including the overall objective of the project, the prior domain knowledge availability and operational factors such as availability of training data, speed and accuracy requirements, and so on), the data factors (availability, cost, structure, size, noise characteristics, etc.), and the human factors (e.g. expertise of end-users).

Determining the representation. Once the main characteristics of the task are well understood, it becomes possible to choose an effective knowledge representation for both the training data and the knowledge to be learned. In particular, relevant features and relationships should be selected as much as possible at this stage.

Selection of the learning method. Learning methods can often be analyzed as consisting of three components : a *representation scheme* (for instance, do we need to represent the full joint density of features and classes (as in Bayes rule), or only conditional distribution of the class given the features (as in multilayers perceptrons)), an *estimation criterion* measuring the quality of the learned model (e.g. maximum likelihood criterion), and a *search strategy* that guide the exploration of the space of models (e.g. local greedy iterative, stochastic).

Collection of the training data. Collecting the training data might be straightforward in some cases, to the point of possibly being fully automatized, or demand considerable effort specially when human operators are asked to fill up forms. For instance, in the "diagnosis of mechanical devices" application mentioned above, technicians were reluctant to make measurements and collect data when the machines were working properly.

Evaluation. The evaluation process depends on the objective of the project. If classification accuracy was the main goal, then the methods described in section 2.4 are appropriate. In other cases, understandability or other desiderata demand other performance measures and may warrant further refinement of the learned knowledge (e.g. decision tree pruning). It must be noted that, following evaluation, the whole development process, or part of it, might be repeated in order to reach better performance.

Of the above analysis, one main idea must be stressed : successful applications require that machine learning expertise be available in addition to domain and user expertises. Except for limited problems, it is not possible to consider machine learning as a provider of simple and ready to use tools, but must be regarded as a set of methods and techniques in their own right that can bring much benefit but call for specific expertise.

Aside that, real-world applications entail numerous practical problems for which methods developed in academic settings are not well fitted without adaptation work. We mention some of these here.

One such problem stems from the characteristics of real-world data as compared to synthetic data. They are usually very "noisy", meaning there are lots of errors, and often missing values. The data samples are not always quite representative of the problem space. In classification tasks, the classes may overlap, there may be hidden relationships between features. Sometime, the size of the data set makes the use of some learning algorithms intractable without serious rethinking. Another significant problem concerns the integration and use of prior domain knowledge into the learning system. Some learning techniques like ILP (Inductive Learning Programming) are well-adapted for this, most (like decision tree learning systems, neural net methods, and so on) are not. Conversely, it is not always easy to extract useful knowledge from the output of the learning system. This also pleads for the development of multistrategy learning techniques (see section 2.4). Related to the prior knowledge issue is the problem of being able to use structured representations. Few learning systems in fact have this capacity, they are usually restricted to one simple (e.g. decision trees) representation scheme. These difficulties and others have been identified (see [39]) as serious obstacles to the spreading of machine learning applications.

4.3 *Data Mining as a new frontier*

One family of concerns, and hopes for new business opportunities, has gained increasing attention over the last few years. It has been popularized under many terms such as *data mining*, *knowledge discovery in databases*, *data warehousing*, and so on. Its core preoccupation is the *extraction of implicit, previously unknown, and potentially useful information from data* [40], and when closely examined, it rapidly becomes a highly demanding task as it covers many needs.

First, intended sources of information are generally *very large databases*, typically involving tens of thousands or millions of basic facts. Correspondingly, mining techniques need be efficient. Second, the databases that are so abundant everywhere have generally not been structured in view of specific discovery goals. Therefore, data mining must face *all sorts of data patterns* (relational, object-oriented, etc.) where it is often not easy to distinguish the relevant features, causes and effects, antecedents and conclusions. This difficulty is compounded by the fact that *all kinds of regularities* in the data are looked for, including generalization, classification, discrimination, characterization, clustering, statistical correlations, deviations, outliers (that may be significant), prediction rules adapted to time series, and so on. Often, the user does not know exactly what he or she is seeking. This brings us to the third aspect of the problem : data mining systems must be *much more user oriented* than typical machine learning systems. There use must be simple and intuitive, they must offer visualization tools, if they support multiple knowledge representations this is all the best, and they must be designed for an iterative trial and error approach. This again requires that discovery methods be computationally efficient. Finally, these systems must be equipped with good interfacing facilities, both to different databases standards, but also to larger computing environments.

Several techniques, of which many fall in the scope of machine learning, are potentially involved. The most obvious ones include : inductive learning techniques, clustering and concept formation methods, statistical analyzis, sequences and time series extrapolation, visualization, dialog techniques, database management methods.

The prospects are plentiful, with regards to both research and end products. For instance, progress are encouraged in the following topics : definition of new kinds of regularities or structures and thereby definition of interestingness, improvement of the efficiency of the algorithms to deal with huge amounts of data, increasing output comprehensibility and user-friendliness, supporting multistrategy learning. If these needs are met, then machine learning and related disciplines should harvest a wealth of applications and of support from interested clients.

5 Conclusions

The field of Machine Learning has undergone a tremendous development over the last fifteen years or so (see [41], [42], [43], [44] for representative collections of papers). There exists now a wide range of efficient and well-tested machine learning techniques that are available to interested users. This is especially true for inductive learning tasks such as classification and categorization. In addition, there has been a steady growth of theoretical works that bring a solid and deep understanding of induction, particularly regarding the convergence properties and complexity of various classes of learning methods and problems.

However, most of the learning systems developed so far are monostrategy systems, in that they implement only one learning method, use a single knowledge representation language and maintain a unique knowledge base. They are thus intrinsically limited to solving only certain classes of learning problems. It is now increasingly realized that there is a need for the development of multistrategy learning systems that combine several inferential strategies and knowledge bases, and for the integration of learning modules into wider, more complex problem-solving systems. This in turn is stimulating works on a kind of knowledge level analysis of learning, which offers perspectives for future integration with knowledge-based technology. Much research remains to be done in this area, but this will be a critical test for the maturation of the field. Likewise, there is a need for more studies devoted to long-time learning, that is learning taking place incrementally over long durations, with the possibility of environmental drifts and changes, all requiring techniques for theory revision and continuous adaptation.

Among many interesting directions, two new challenges and applications areas are actively pursued in the machine learning community. One is the development of the Case-Based Reasoning technology for knowledge-based systems. It is a promising method that offers an attractive alternative to the traditional knowledge intensive (therefore hard to acquire) approach. The second is related to the task of Knowledge Discovery in Databases that is increasingly recognized as a major stake for the exploitation of the formidable amounts of data stored everywhere. This represents both an opportunity and a challenge for machine learning since, on one hand, it requires the massive use of inductive techniques, something well-mastered in machine learning, but, on the other hand, it demands that integration with existing data bases tools and representation languages be realized, and moreover that the output be easily interpreted, evaluated and understood by end-users.

In summary, there is a powerful trend towards works aimed at integrating learning methods into more complex systems. This does not preclude the need for sustained studies, both theoretical and empirical, of basic methods and problems.

References

1. A.L. Samuel. 'Some Studies in Machine Learning Using the Game of Checkers', In *Computers and Thought*, E.A. Feigenbaum and J. Feldman (Eds), McGraw-Hill, New York, (1963), 71-105.
2. H. A. Simon, 'Why should machines learn ?', in [40], 25-37.
3. J.R. Anderson, *The Architecture of Cognition*, Harvard University Press, (1983).
4. P.H. Winston. 'Learning Structural Descriptions from Examples', In *The Psychology of Computer Vision*, Winston P.H. (Ed), Chapter 5, McGraw-Hill, (1975).
5. D.B. Lenat. 'EURISKO: A program that learns new heuristics and domain concepts', *Artificial Intelligence 21*, (1983), 61-98.
6. T.M. Mitchell, 'Generalization as Search', *Artificial Intelligence*, Vol.18 (2), (1982), 3-226.
7. R.S. Michalski, 'A Theory and Methodology of Inductive Learning', in *Artificial Intelligence*, vol 20, N° 2, (1983), 111-161.
8. S. Muggleton (Ed.), *Inductive Logic Programming*, Academic Press, (1992).
9. I. Bratko, S. Muggleton 'Applications of Inductive Logic Programming', In *Communications of the ACM*, Vol 38, N° 11, (1995), 65- 70.
10. M.H. Hassoun, *Fundamentals of Artificial Neural Networks*, MIT Press, (1995).
11. J. Quinlan, 'Induction of Decision Trees', in *Machine Learning* 1, (1986), 81-106.
12. J. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, (1993).
13. C. Stanfill & D. Waltz, 'Towards memory-based reasoning', *Communications of the ACM*, 29 (12), (1986), 1213-1228.
14. J.L. Kolodner, *Case-Based Reasoning*. Morgan Kaufmann (1993).
15. D. Fisher, 'Knowledge acquisition via incremental conceptual clustering', *Machine Learning*, 2, (1987), 139-172.
16. D.E. Goldberg, 'Genetic and Evolutionary Algorithms Come of Age', *Communications of the ACM*, 37, N° 3, (1994), 113-119.
17. L.P. Kaelbling, *Learning in Embedded Systems*, MIT Press, (1993).
18. T.G. Dietterich, 'Machine Learning'. In J. Traub (Ed.), *Annual Review of Computer Science*, vol.4, (1990), 255-306.
19. D.H. Wolpert (Ed.), *The Mathematics of Generalization*, Addison Wesley, (1995).

20. W. Buntine, 'Classifiers: A theoretical and empirical study', In *International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, (1991).
21. V. Vapnik , *The Nature of Statistical Learning Theory*. Springer Verlag, (1995).
22. L. Saitta, 'Representation change in machine learning', *AI Communications*, 9 (1), (1996), 14-20.
23. R.S. Michalski & G. Tecuci (Eds.), *Machine Learning. A Multistrategy Approach*. Vol. IV. Morgan Kaufmann, (1994).
24. D. Michie, D. Spiegelhalter, C. Taylor, *Machine Learning, Neural and Statistical Classification*, Prentice Hall, (1994).
25. R.S. Michalski, R.L. Chilausky, 'Learning by Being Told and Learning from Examples: An Experimental Comparison of the Two Methods of Knowledge Acquisition in the Context of Developing an Expert System for Soybean Disease Diagnosis', In *International Journal of Policy Analysis and Information Systems* 4 (2), (1980), 125-161.
26. S. Craw and D. Sleeman, 'Automating the Refinement of Knowledge-Based Systems', *Proceedings of ECAI-90*, Luigi Aiello (Ed.), Pitman, (1990), 167-172.
27. K. Morik, S. Wrobel, J.-U. Kietz, W. Emde, *Knowledge Acquisition and Machine Learning*, Academic Press, (1993).
28. A. Anjewierden, N. Shadbolt & B. Wielinga, 'Supporting knowledge acquisition: The ACKnowledge project', In *Enhancing the Knowledge Engineering Process - Contributions from ESPRIT*, Elsevier Science, (1992), 143-172.
29. B. P. Allen, 'Case-Based Reasoning: Business Applications', *Communications of the ACM*, 37, N° 3, (1994), 40-42.
30. C. Brodley & P. Smyth, 'Applying Classification Algorithms in Practice', available with brodley@ecn.purdue.edu. (1995).
31. Y. Kodratoff, 'Industrial applications of ML: Illustrations for the KAML dilemma and the CBR dream', In F. Bergadano and L. De Raedt, editors, *Proceedings of the European Conference on Machine Learning (ECML-94)*, Springer-Verlag, (1994).
32. Y. Kodratoff and P. Langley, editors, *Real-World Applications of Machine Learning*, Workshop notes ECML-93, (1993).
33. Y. Kodratoff, V. Moustakis and N. Graner, 'Can machine learning solve my problem?', *Applied Artificial Intelligence*, 8(1), (1994), 1-31.
34. P. Langley and H. A. Simon, 'Applications of machine learning and rule induction', *Communications of the ACM*, 38, (1995), 55-64.
35. MLnet. *Proceedings of the MLnet Workshop on Industrial Applications of Machine Learning*, Dourdan, France, (1994).
36. A. Rudstrom, 'Applications of machine learning', *Technical Report: 95-108*, Stockolm, Sweden: University of Stockolm, Department of Computer and Systems Sciences, (1995).