

Learning

David Danks

Department of Philosophy, Carnegie Mellon University; and
Institute for Human & Machine Cognition

Danks, D. (2014). Learning. In K. Frankish & W. M. Ramsey (Eds.), *Cambridge handbook to artificial intelligence* (pp. 151-167). Cambridge: Cambridge University Press.

Contact info:

Department of Philosophy
135 Baker Hall
Carnegie Mellon University
Pittsburgh, PA 15213
ddanks@cmu.edu

Learning by artificial intelligence systems—what I will typically call machine learning—has a distinguished history, and the field has experienced something of a renaissance in the past twenty years. Machine learning consists principally of a diverse set of algorithms and techniques that have been applied to problems in a wide range of domains. Any overview of the methods and applications will inevitably be incomplete, at least at the level of specific algorithms and techniques. There are many excellent introductions to the formal and statistical details of machine learning algorithms and techniques available elsewhere (e.g., Bishop, 1995; Duda, Hart, & Stork, 2000; Hastie, Tibshirani, & Friedman, 2001; Mitchell, 1997). The present chapter focuses on machine learning as a general way of “thinking about the world,” and provides a high-level characterization of the major goals of machine learning. There are a number of philosophical concerns that have been raised about machine learning, but upon closer examination, it is not always clear whether the objections really speak against machine learning specifically. Many seem rather to be directed towards machine learning as a particular instantiation of some more general phenomenon or process. One of the general morals of this chapter is that machine learning is, in many ways, less unusual or peculiar than is sometimes thought.

Three Broad Classes of Inference

At a very high level, one can distinguish between three different, not necessarily exhaustive, inferential strategies: analogical, domain-specific, and structural. As an example of the generality of this taxonomy, both deductive and inductive logics are types of structural inference. Analogical inference aims to map some situation or problem onto salient historical examples, whether well-known or personal; inferences are then made by using the analogical mapping to translate the historical outcomes onto the present problem. The problems and

outcomes need not be large or significant: if one has previous experiences with light switches and light bulbs, then one can use analogical inference to make a decision about how to turn on the lights when entering a new room. If the analogies are suitable, then analogical inference can support inferences for very rare situations or phenomena, or inferences from very limited data. The reliability of analogical inference is, however, highly dependent on the analogical cases and mappings, and there is little known about search for such cases, or development of suitable mappings. With regards to the focus of this chapter, analogical inference is rarely done by machine; usually, we do analogical inference in our minds.

Domain-specific inference uses techniques that are specifically tailored to knowledge about the particular problems, environments, and responses that occur in a domain. By using specialized algorithms and constraints, one can often make quite powerful inferences, even given only limited amounts of data. Domain-specific methods, however, can only be developed and used with substantial prior domain knowledge, which may preclude the widespread use of such methods. Domain-specific machine learning and inference can often be understood as part of the particular domain, rather than as a distinctive and novel inference strategy. Also, since any technique must use some domain-specific information (e.g., the possible values of a variable), it is unclear whether any sharp line can be drawn to delimit exactly the “domain-specific” methods, though there are clearly many inference algorithms that are applicable only for highly specific situations.

Structural inference uses (relatively) domain-general algorithms whose success depends on the internal structure of the data, rather than features of the semantic content of the data. That is, structural inference focuses on the relationships among the variables, objects, or predicates, rather than on any intrinsic properties of them. This type of inference is necessarily domain-

general, as such methods are explicitly designed not to use any domain information except “structural” information about the objects of inference (e.g., number of variable values, whether spatial location of objects is relevant, and so on). The advantage of structural inference is obvious: the methods are applicable for any domain in which the appropriate structural features hold and can be discovered from data. These methods are thus not restricted to domains in which we happen to have substantial prior knowledge, nor do we need to have any significant experience with situations of this type. The disadvantages of such inferences are equally obvious: one cannot infer domain-specific mechanisms (since domain-specific information is excluded), and inference from small datasets can be quite difficult.

Structural inference is the basis of many, and arguably most, machine learning frameworks and methods (including many well-known ones such as various forms of regression, neural network learning algorithms such as back-propagation, and causal learning algorithms using Bayesian networks). In all of these methods, the algorithm works by extracting—and exploiting—structural relationships among the variables without regard to the meaning or domain of the variables. For example, if doing classification using an artificial neural network, one might be provided with a dataset containing measurements of various features of widgets, as well as some target category. The neural net learning algorithm (e.g., standard back-propagation) then uses only the statistical regularities in the dataset to learn the relevant inter-variable structure, which can then be used to predict the target category (“functional” vs. “defective”) for future widgets. The precise “meaning” of the variables is irrelevant to the learning algorithm. For all of these methods, one need not know much about the underlying domain in order to apply the methods, though domain-specific information (e.g., variable X takes on a value before variable Y) can typically be incorporated in various ways. The domain-generality of machine learning

methods partly explains their popularity in relatively novel scientific domains, such as bioinformatics, in which there is substantial uncertainty about what models or methods are appropriate.

There is a natural division among structural inference methods between logical and statistical methods. Logical methods typically aim to model the structure in terms of deductive relationships, perhaps supplemented with various representations of one's lack of precise knowledge about a situation. The methods often use various types of modal logic to help represent and infer uncertain possibilities. The most common use of logical machine learning methods is for inference from prior knowledge, where that prior knowledge encodes structural information about the particular domain. This chapter will focus more closely on statistical methods, which use larger amounts of data to infer structural relationships. Most of these methods use data to determine which variables are informationally relevant for which other ones, and then use the absence of such informational connections to develop simple-but-accurate models with significant predictive power.

There is an obvious difference between the learning algorithms and the learned model. For example, some particular neural network (with connection weights, etc.) is a learned model; back-propagation is the algorithm by which the model is learned. Machine learning algorithms are a type of structural inference because the *learning* makes not intrinsic reference to the domain under study; in particular, the learning algorithm does not use (significant) semantic information about the variables. This observation leaves open the question of whether the learned *model* does have interesting semantic content. We will return to that question later in this chapter. In the meantime, however, it is important to bear the “learning algorithm vs. learned model” distinction in mind when thinking about these processes.

A Rough Taxonomy of Machine Learning

Suppose one has a dataset D : a collection of datapoints, each of which has measurements of the values of variables V for a particular individual or unit. There might be many complications with the dataset: the variable values might be imputed or inferred; it might not be obvious how to specify the variables or individuals; the datapoints might not be independent (e.g., if one has time series data); there might be unmeasured factors that influence variables in the dataset; and so on. In terms of developing a rough taxonomy of machine learning methods, these subtleties are largely irrelevant. That being said, essentially all machine learning methods assume that the situation is “well-posed” in various ways, such as using well-defined variables.

At the coarsest level, machine learning algorithms can be divided into two classes (with a small middle ground) based on whether the algorithm requires the specification of a target variable in the dataset. *Supervised learning* algorithms assume that some variable X is designated as the target for prediction, explanation, or inference, and that the values of X in the dataset constitute the “ground truth” values for learning. That is, supervised learning algorithms use the known values of X to determine what should be learned. The most common type of supervised learning algorithm aims to develop a classification or categorization model: given information about various individuals and the categories to which they belong, the algorithm produces a learned model that can be used to predict the category membership of new individuals. For example, one might want to predict which widgets being produced in a factory are most likely to fail. If one has data on the performance of many different widgets as well as measurements of other relevant features, then one can use a machine learning algorithm to learn a model that will predict the performance of future widgets. Under the right conditions, classification algorithms can yield models that can make novel, warranted generalizations about the groups based on the

inter-feature relationships. Classification algorithms can also be used for recognition or identification by classifying into a “category” with exactly one member. Examples of supervised learning algorithms include: learning algorithms for artificial neural networks, decision trees, and support vector machines; the many forms of regression; and most reinforcement learning methods.

Unsupervised learning algorithms do not single out any particular variables as a target or focus, and so aim to provide a general characterization of the full dataset. Probably the most common use of unsupervised learning is in clustering algorithms: separating the various individuals into “natural” groups according to one or another metric. These algorithms will sometimes draw relatively arbitrary lines between individuals, but they can be quite effective at discovering groups when they actually exist. For example, one might measure people’s attitudes about various political issues, and then want to determine whether there are natural groups that can be defined by those beliefs. The output of a clustering algorithm can, in certain conditions, subsequently serve as the target variable for a supervised learning algorithm. It is typically quite difficult to validate the output or model from an unsupervised learning algorithm, precisely because one usually has no “ground truth” against which to compare the performance of the model. There is rarely a single correct, true way to cluster various individuals, only better and worse ways. Various methods for probability distribution or density estimation also fall under the heading of unsupervised learning.

The standard view of learned models is that their semantic content is entirely statistical: connections between variables provide information that can be used for prediction, but no further semantic content—for example, causal structure—is thought to be attributable to these models. In recent years, however, there has been a significant surge of interest in recent years in machine

learning algorithms that avoid the use of domain-specific assumptions, but produce learned models with rich semantic content. In particular, the learned models can be used to predict the future behavior or features given interventions or manipulations from outside of the system. These machine learning algorithms are typically unsupervised learning methods, although one often wants to learn the causal structure in order to affect or bring about a change in some particular variable. One might wonder how such causal inference is possible, given the completely standard maxim in the sciences and philosophy that “correlation is not causation.” Machine learning methods for causal discovery must make assumptions with some causal content, but they typically use only domain-general assumptions about the ways in which causation and correlation are connected. For example, the widely-discussed causal Markov assumption (e.g., Cartwright, 2002; Hausman & Woodward, 1999, 2004) asserts that a variable provides no information about its non-effects, if one already knows the values of the variable’s direct causes. This assumption has causal content, but at a very high level of generality.

This type of causal learning is more difficult than purely statistical learning (e.g., clustering, classification, density estimation, function approximation) in which one is simply trying to find informational connections between the variables. Except in highly unusual circumstances, the set of causal relationships among some features will be a strict subset of the set of informational relationships. That is, (almost all) causal relationships are informational relationships, but not all statistical relationships are causal. Given this asymmetry in learnability for the different types of models, one might hope that statistical models could suffice for all interesting applications. Purely statistical information, though, is insufficient for prediction when the system changes, whether because of one’s actions or policies, or perhaps because the causal structure breaks in various way. One needs causal information to predict the likely effects of

most interventions, policy decisions, or other exogenous changes in the system. We must sometimes tackle the harder learning problem.

Machine learning algorithms must always try to find a balance between (i) complexity of the learned model, which provides increased accuracy in representing the input dataset; (ii) ability of the learned model to generalize to new data, which makes them informative about the world; and (iii) computational tractability of learning and using the model, which makes them practically useful. The complexity of the world will sometimes be greater than the complexity of the models that are practical for a particular situation, which suggests that no single model will be sufficient. One might hope that there are “sub-problems” that are not more complex than the available models. If that is the case, then one could plausibly learn more about the world by (a) determining the scope of the sub-problems; (b) learning an appropriate model (or models) for each sub-problem; and then (c) integrating the model outputs in a principled manner. Various “meta-learning” techniques implement this three-step proposal. As an illustration, consider the case of boosting (Freund, 1995; Schapire, 1990; and subsequent work) for simple binary classification: e.g., is this widget defective? Rather than trying to learn a complete model in one step, a boosting algorithm first learns a simple classification model that works reasonably well, though typically not as the user wants or requires. The system then extracts all of the cases for which this simple model makes an *incorrect* prediction, and learns a second classification model just for those cases. The outputs of those two models can be integrated in various ways to get a classifier for all of the cases. That unified classifier will make incorrect predictions for other cases, and so one can learn a third classifier for those misclassified cases, integrate the new classifier

into the unified one, and iterate.¹ In this way, boosting builds a unified classifier consisting of a number of “weak” classifiers, each of which focuses accurate classification of a reduced subset of the data. Hierarchical models such as mixtures of experts (e.g., Jordan & Jacobs, 1994) function similarly.

No overview of machine learning would be complete without a discussion of Bayesian learning. A Bayesian learning algorithm requires specification of a (possibly infinite) set of possible hypotheses or models, as well as a probability distribution—the “prior probability distribution”—over those hypotheses. When provided with data, the learning algorithm then uses Bayes’ Rule to determine the correct (by the probability calculus) probability distribution given that data. Bayesian reasoning captures the intuition that beliefs after observing some data should be given by the probability of each possible explanation given that data. Expressed in ordinary language, Bayes’ Rule states: the probability of a hypothesis after observing some data [$P(H | D)$] is equal to (i) the prior probability of the hypothesis [$P(H)$], multiplied by (ii) the likelihood of seeing data like that if the hypothesis actually were true [$P(D | H)$], divided by (iii) the probability of seeing that data in the first place [$P(D)$]. The idea that Bayesian learning is rational has a long philosophical history. Bayesianism had relatively little practical impact for many years, however, because of a simple fact: except in toy examples, the computations required for Bayesian learning quickly become too difficult to do analytically or by hand. The development of modern digital computers has made it possible to carry out or approximate Bayesian learning for more realistic situations, and so Bayesianism has (re-)emerged as a dominant theme in learning. Many machine learning algorithms can be viewed as implementing or approximating

¹ For technically minded readers, boosting techniques rarely focus on *only* the misclassified datapoints at each stage. Rather, the currently misclassified datapoints are simply weighted more heavily for training of the next classifier.

Bayesian learning under various assumptions or constraints on the hypothesis space, prior probability distribution, likelihood functions, and so on.

Scope and Limits of Machine Learning

As with human learning, the value of machine learning is less in the output, and more in the way that the output can be used for future tasks: prediction, planning, classification, recognition, and so on. As a community, we know how to do quite a lot with machine learning. Machine learning is a large part of present-day computer science, and there are many different algorithms and techniques that are suitable for a wide range of conditions. For clustering, classification, and causal learning, there are of course algorithms for the simple situations: datasets containing all relevant variables, clean measurements, simple relationships (e.g., linear), and no missing datapoints. But there are also algorithms that are robust to variations along all of these dimensions: noisy data, unmeasured variables, complex relationships, missing data, sample selection bias, and so on. There are numerous success stories for each of these algorithms in terms of real-world applications. There are also algorithms for handling time series data, and in particular, for conducting systems monitoring and fault detection. There are quite powerful text and image classification algorithms that are highly specialized for these purposes (though they typically still fall short of human performance in accuracy). Information fusion—the integration of information from multiple distinct sources—has emerged more recently as a central component of real-world machine learning.

At the same time, there are known theoretical limits to machine learning, many of which mirror the limits on human learning. For example, if the data are too noisy—if they are essentially random—then learning will be nearly impossible. Machine learning algorithms employ structural inference, and so if there are no patterns in the data, then there is nothing that

can be inferred. Learning also requires some variation in the world, either between individuals, or between times, or between places. Machine learning algorithms cannot learn anything about a constant-valued feature, since there is nothing to learn: the constant feature is always the same. And although some situations are clearly easier for learning than others, learning is almost always difficult in the worst case. More precisely, essentially all interesting machine learning problems are sufficiently hard that they require (we think) an algorithm with exponentially many computational steps in the worst case.

A more interesting constraint on machine learning is the inability, even under seemingly easy conditions, to infer features of the causal or informational structure of an individual from group-level measurements. Suppose that we want to learn something about the individual (e.g., how does education influence subsequent income?), but we only measure features at the group level (e.g., the averages in various groups of education, income, and other relevant variables). Further suppose that every individual has exactly the same type of underlying relationships (though not necessarily the same values), and the group-level features are simple, deterministic functions of the individual-level features (e.g., average or total value). Even under these strong simplifying assumptions, there are many interesting cases for which the informational relationships between the group-level features are *not* the same as the relationships among the corresponding individual-level features (Chu, Glymour, Scheines, & Spirtes, 2003). That is, the learned model for the group-level features is not necessarily the same as the model for the individual, even when every individual has the same model. This possibility raises a serious methodological challenge to the use of machine learning for domains in which individuals are the primary focus, but data collection principally occurs for groups (e.g., parts of economics, other social sciences, and bioinformatics).

Philosophical Challenges to Machine Learning

Machine learning is a major area of research in computer science and statistics, and so many, and perhaps almost all, of the most prominent problems in machine learning are computational and algorithmic (e.g., “what can one learn under certain conditions?” or “can this algorithm run faster?”), rather than necessarily philosophical. Even notions from machine learning that might appear philosophical often turn out to be less philosophical than one might have thought. As just one example, consider the so-called “No Free Lunch” theorems (e.g., Wolpert, 1996; Wolpert & Macready, 1997), which are sometimes colloquially stated as: “Algorithms are successful only when they are ‘tuned’ to their domain; there are no universal learning algorithms.” This phrasing suggests various philosophical arguments, but all trade on a misunderstanding of the actual theorems. For example, one might be tempted to argue that machine learning is pointless, since one might think that the no-free-lunch theorems imply that proper algorithm choice requires that one already know the underlying truth, which would obviate the need for any learning. This suggested argument fails to understand the sense in which no algorithm has an advantage over others. The No-Free-Lunch theorems are, in many ways, just a precise statement of the ancient skeptical observation that any future is consistent with the past. If any future is possible given the past observations, then no learning algorithm has any advantage over others. But one can now straightforwardly see that almost any restriction on the possibility space suffices to defeat the no-free-lunch theorems; for example, a single weak regularity assumption can suffice to define a “domain,” and so pick out a privileged class of superior algorithms. One certainly need not *a priori* know the actual, underlying truth.

Of course, as with many bad arguments, there is a kernel of truth inside this suggested objection. Any interesting machine learning method makes assumptions about the nature of the

world, and algorithms can readily fail if those assumptions turn out to be false. An important part of machine learning is to investigate whether the assumptions of one's algorithm actually hold, at least approximately (e.g., by checking to see whether the data distribution is approximately Gaussian). Such tests are often missing from both the practice and rhetoric of machine learning. If the relevant assumptions are false, then one should turn to other methods that do not make those assumptions, even though those other methods will typically be correspondingly weaker. It is incorrect to think about machine learning as a "black box" that simply takes data as input and returns the truth. The practice of machine learning is instead much closer to the use of statistics in science: as a tool to investigate more precisely the structure of one's data. The appropriate tool (i.e., machine learning algorithm) should be chosen for a particular task, and tools can be used with varying degrees of skill (e.g., by interpreting the output of the algorithm in various ways). One might hope for a sophisticated system that could take the input, determine the best algorithm for that type of data, and then apply the algorithm, but such a meta-learner currently remains largely a hope.

One of the least-discussed "assumptions" of machine learning algorithms is that they all require one to provide well-specified variables with precise, possibly infinite, sets of values. The variables need not be numeric—they can range over various categories, such as "large" and "small"—but they must be clearly stated: in some sense, there must be some, possibly unknown, fact about the "true" value of each variable for each datapoint. Machine learning relies on structural inference, and so it must be possible to find patterns and structure within the data. It is not clear what it even means to talk about 'structure' among variables that are not well-defined. This concern is not a serious challenge in practice, as one is essentially always concerned with datasets that result from measurement processes that specify the variables; metaphysical realists

of various types will also typically be untroubled by this concern. If, however, one questions whether there is any stable underlying structure to be measured, then machine learning will seem to be a futile enterprise.

All of the observations in this section raise a natural question: if machine learning is roughly analogous to statistics, then in what sense is it “learning”? A more contentious framing would be: is the machine doing any learning, or is it really the human who uses the algorithm that learns? There are two different aspects to the “is it learning?” question: the first is a general philosophical concern, and the second points towards gaps in our understanding of cognition. The first concern was most famously presented by Searle (1980) using his Chinese Room, but has arisen in many different forms (e.g., Harnad, 1994). The argument starts with the general claim that computation involves only symbol manipulation while cognition involves something more. The ‘something more’ of cognition differs between authors, but is often some semantic notion (or “grounding” for concepts, or...). Symbol manipulation is then characterized as a purely syntactic notion: according to this argument, one can manipulate symbols correctly solely by examining features of the physical representation and without any understanding of the semantics or meaning of the symbol. The argument then concludes that computation cannot be cognition, as the former lacks any semantic content or grounding in the world while the latter necessarily has it.

This argument is offered as a general one against the idea of “cognition as computation,” and machine learning is clearly a relevant type of computation. The successes of machine learning result from structural inference; these methods use patterns or statistical regularities in the data, and are (relatively speaking) indifferent to the semantics of the input variables. A more specific version of the previous argument would conclude that machine “learning” might be

useful, but it cannot be *true* learning, at least in so far as true learning requires cognition. In other words, there might be some actual learning, but the human being who processes the machine “learning” output is the one who does it. The machine simply makes certain patterns in the data salient, though that might be a computationally non-trivial task. This is a serious objection to at least the title ‘machine learning’ for these algorithms, since this argument calls into question the use of all cognitive terms to refer to machine operations. However, this argument does not seem to provide any specific objection to machine learning in itself, but rather it applies to machine learning *qua* machine operation. That is, one’s particular response to (or acceptance of) this argument—for example, appeal to some symbol grounding process, special causal powers of the brain, or rejection of some premise—will arise from more general philosophical grounds, and not from some deeper reflection on the nature of machine learning in isolation. The overall objection is clearly relevant to machine learning, but it seems just as clear that any solution to it must take into consideration many issues that lie outside of the scope of machine learning.

There is a more specific form of the “is this learning?” objection that does speak directly to machine learning. Insight and creativity are often held up as a central feature of human learning, if not *the* central feature. Our learning seems to depend at times on crucial intuitive leaps that we do not seem to be able to explain or predict. Introspectively, there seems to be something “non-algorithmic” about creative insight. Machine learning algorithms seem to offer no such capacity for insight, as they are “just” complex sequences of simple operations. The practice of machine learning inevitably involves some human element to specify and control the algorithm, test various assumptions, and interpret the algorithm output. These observations suggest the conclusion that machine learning is (again) not true learning at all, but rather fast, useful detection of various patterns in data. On this account, the human who controls and

validates the algorithms does the “real” learning. This objection is notably different from the previous one: no claims are made here about the impossibility of machine cognition, but only about the failure of *current* machine learning algorithms to rise to the level of true learning. This objection is entirely consistent with the possibility that more sophisticated and reflective algorithms, supplemented with appropriate background knowledge, could perform real learning. The argument depends instead on the claim that none of the currently available algorithms meet that standard for true learning.

The previous paragraph used the phrases ‘human learning’ and ‘true learning’ without exposition; the reader was simply assumed to understand what was intended by it. One might wonder, however, if our understanding of the nature of human learning is sufficiently clear to provide a standard of ‘true learning’ that machine learning fails to satisfy. There is no well-established model for how people actually do learn, and so it is not clear what criteria would need to be met for a machine algorithm to be considered ‘learning.’ There is no question that—for certain situations—human learning is far superior to machine learning. Our ability to assemble disparate pieces of background knowledge and information, whether by analogy, accident, or some other process, is unmatched in machine learning (despite many attempts to build systems for common sense reasoning). That observation, however, is not sufficient to conclude that we use some wholly different process in our learning; one can only conclude that there is *something* different about our learning. A plausible alternative explanation is that we have a body of information, biases, and experiences that is quite simply unmatched by contemporary machine learning systems. A database with 10,000 datapoints is considered large in machine learning; a child who has only one experience per waking hour (roughly, sixteen per day) exceeds that number in less than two years. If she has one experience per waking minute,

then she surpasses the database in around eleven days. The products of human learning are superior (in some sense) to the products of machine learning, but the processes need not be fundamentally different in kind, given that there are enormous differences in background knowledge, accuracy of biases, temporal and semantic information, and so on.

Perhaps more importantly, there are substantial gaps in our understanding of the processes underlying human learning. We do not know enough about those processes to determine at this point their similarity—or dissimilarity—to the algorithms proposed in machine learning. One might respond that introspection on our own learning provides all of the evidence that is required. Such a response neglects the large psychological literature demonstrating the unreliability of introspection in revealing the details underlying fundamental cognitive processes (Nisbett & Wilson, 1977; Ross & Nisbett, 1991). It may well be the case that there is something qualitatively different about human learning such that machine “learning” algorithms do not deserve that name. At the current time, however, such claims are grounded largely in ignorance, rather than positive evidence of a difference.

The preceding discussions have made a potentially problematic assumption: there is value in worrying about the particular label that is attached to machine learning algorithms. The fundamental properties of those algorithms—their reliability, convergence, computational complexity, etc.—are features of them regardless of the name one uses. Moreover, the algorithms are already referred to by many different names, such as ‘data mining’, ‘applied statistics’, ‘automated search’, and so on. One might thus be willing to give up the label of ‘machine learning,’ since it does not obviously make a difference to the underlying science. One ought not give up on the label of ‘machine learning’ so easily, though, as the name points towards a number of interesting issues about the nature of cognition and learning, and the

relevance of machine methods for the study of human cognition. Machine learning methods are regularly used today to provide frameworks and inspiration for cognitive models, sometimes under the heading of ‘computational cognitive science.’ The label is also important because it establishes biases and expectations in those who hear the label. Sometimes those expectations are unreasonable, but they prompt individuals—both proponents and skeptics—to ask important questions about the nature and performance of these algorithms.

One final philosophical issue concerns the extent to which one can be a realist about the contents or intermediate processes of a learned model. That is, when can the internal structure or richer semantic content of a learned model be understood to correspond—perhaps only with some probability—to features of the world? This question is particularly pressing for research in causal learning that seeks to infer causal structure in the world from sets of passive observations. Causal inference algorithms putatively learn the set of causal structures that could have produced some given dataset, or discover the most probable such structure (Chickering, 2002; Pearl, 2000; Spirtes, *et al.*, 1993). There are many instances in which these algorithms have been applied to actual datasets, and the learned models have subsequently been successfully attributed to the world (e.g., the case studies in Glymour & Cooper, 1999). These algorithms—like all inference methods—are only reliable under particular assumptions about the nature of the world.

Moreover, the semantic content of the learned model—the fact that we can call it a *causal* model—derives from these assumptions, which provide a characterization of one (the?) way for causal structures to be “projected” into observed or experimental data. The analogy here is with assumptions such as the ray theory of light: those assumptions explain how three-dimensional objects are projected onto a two-dimensional plane (e.g., a retina), and are necessary for any visual system to make inferences about object structure from the limited, two-dimensional input.

Just as our visual system experiences optical illusions when various assumptions fail to hold (e.g., a straight stick appearing bent when placed into water), causal inference algorithms are subject to “causal inference illusions” when the assumptions are violated in particular ways.

These causal inference algorithms, and machine learning algorithms more generally, do not seem at this point to be any different from standard instances of inductive inference: no inductive inference can have any guarantees of reliability without various assumptions about the world. If those assumptions are satisfied, then the algorithms work; if they are violated, then one has no particular warrant to believe the internal structure of the algorithm outputs. The assumptions of causal inference algorithms are sometimes claimed to be different, however, because it seems that we can only test whether the assumptions are actually satisfied by having *the very same causal knowledge that we are trying to learn* (e.g., Cartwright, 1999, 2001). This argument is not focused on the bare possibility that the assumptions could be false (though that additional claim is also made in, e.g., Cartwright, 2001), since that is a risk that any inductive inference must carry. This concern is also not about the practical testability of the algorithms’ assumptions; inductive inference—whether human or machine—inevitably involves making assumptions that might not be practically testable at the particular moment. One might need, for example, orders of magnitude more data than one currently has. If the assumptions are testable in principle, though, then one could (in some sense) determine whether the algorithms are reliable for a situation like this one, and so have some warrant to regard the learned model in a realistic manner.

The fundamental worry here is that the assumptions are not even testable in principle, since it seems that the only way to know which statistical tests are relevant is to know the underlying causal structure, but that is exactly what the causal inference algorithm is supposed to

find. These algorithms might (the argument continues) occasionally find approximations to the true causal structure, but only because of pure random chance. The assumptions might be true in any particular situation, but one has no way to know that, and so no warrant to treat the algorithm outputs as anything other than representations of the observed or experimental data. The algorithms might produce a useful “shorthand” version of the data that one could use in various ways, but one is not (on this argument) learning anything substantive about the underlying structure of the world. Although principally directed at causal inference algorithms, this potential problem is not limited to them. Many clustering algorithms, for example, are reliable only if one can make certain assumptions about the underlying groups. One must thus be able to examine the groups to determine whether they have those properties, but that would require knowing the groups ahead of time, which would obviate the very need to use the clustering algorithm. Machine learning methods clearly have substantially less value if they must always be interpreted in an instrumentalist manner. Instrumentalist theories—those that make predictions about the behavior of a system without making any commitments to the underlying structure or ontology of the theory—are useful in a number of ways, but one often desires something more than mere prediction. Most notably, one must have information about the underlying mechanisms in order to make accurate predictions about what will happen when the system breaks or changes in various ways; instrumentalist theories provide no such information.

There are three natural responses to this objection. First, careful examination of the assumptions often reveals that the knowledge required to test them is weaker than is suggested by the surface framing of the assumption. In the particular case of causal inference algorithms, one must have certain causal knowledge in order to test the assumptions, but the necessary knowledge is not the same as knowledge of the causal structure being sought. For example, one

might need to know that a particular population is “causally homogenous” (i.e., all individuals have the same causal relations, though not necessarily the same values). This knowledge requires causal knowledge, but not necessarily about the causes or effects of that particular variable. The second, related, response notes that the argument frames the testability of assumptions as all-or-nothing: one knows either exactly what is required to test an assumption (though one might not actually test it), or else nothing at all. A more realistic characterization of the situation is that one often knows some-but-not-all of the tests of an assumption, as well as a number of possible avenues for future tests. One might also have reason to believe that one has tested an assumption imperfectly. If one has this type of limited knowledge, then one can have limited confirmation of an assumption, while recognizing that the algorithm output must therefore be interpreted or accepted in a limited manner. One can object to many machine learning methods on the grounds that they require some strong, not completely established, property, but one ought not reject the output of those methods simply because one is somewhat uncertain in the short run about whether the precise property holds.

The third and most general response to this family of objections is to note that the argument-schema actually speaks against most inductive methods, and not just causal inference; it objects to machine *learning*, not *machine* learning. Consider a particular inductive conclusion: “All electrons have negative charge.” Any method that conjectures this conclusion must make some assumptions about the world, such as that electrons form a coherent, stable set with respect to having some electric charge. Such an assumption can only be tested by actually determining the electric charges of all electrons, but such tests would eliminate the need to make any inference to the inductive conclusion. Notice that there was no mention of machine methods in this example, nor was there any specification beyond “inductive method.” The problem of

knowing the confirmation conditions for the assumption of a method is a general one that speaks against almost all inductive inference methods, and not machine learning methods specifically. One can rarely know *a priori* all of the confirmation or testing conditions for assumptions that are required for a particular inference method to provide reliable information about internal or universal structure.

Conclusion

Machine learning methods are often regarded with a certain degree of suspicion. They are frequently presented as “black boxes” that take data and, without any guidance, somehow learn part of the true structure of the world. These algorithms are, in practice, much less mysterious: the label of ‘automated statistics’ is frequently an apt descriptor. Machine learning methods discover and exploit structural relations among the data, and this structural inference underlies both the strengths and weaknesses of machine learning algorithms. These methods can be applied in a relatively domain-general manner, since the specific meaning of the variables is irrelevant to the functioning of the algorithm. Because of this generality, however, they cannot yield domain-specific information, such as mechanisms underlying informational relationships. Machine learning is one of the most rapidly growing areas of computer science, and many of the most prominent challenges revolve around the development and improvement of algorithms for learning novel types of models, or under various assumptions. There are philosophical concerns about machine learning, but most of those concerns center on either the ‘machine’ or the ‘learning’ part. On the one side, machine learning is an instance of complex machine computation, and so natural questions arise about whether any machine operations can be correctly described using cognitive terms. On the other side, machine learning algorithms perform complex, but clearly specified, sequences of computations, and so questions arise about

whether the methods qualify as ‘learning,’ or whether the assumptions necessary for the inductive inference can be suitably tested. In sum, machine learning methods have opened novel avenues for learning about the structure and behavior of our world. These algorithms must of course be used with appropriate awareness and testing of the underlying assumptions. When used properly, however, machine learning can exploit the structure within data to yield valuable knowledge about structure and relations in the world.

References

- Bishop, Christopher M. 1995. *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press.
- Cartwright, Nancy 1999. *The Dappled World: A Study of the Boundaries of Science*. Cambridge: Cambridge University Press.
- Cartwright, Nancy 2001. 'What Is Wrong with Bayes Nets?', *The Monist* 84: 242–264.
- Cartwright, Nancy 2002. 'Against Modularity, the Causal Markov Condition, and Any Link Between the Two: Comments on Hausman and Woodward', *British Journal for the Philosophy of Science* 53: 411-453.
- Chickering, David M. 2002. 'Optimal structure identification with greedy search', *Journal of Machine Learning Research* 3: 507-554.
- Chu, Tianjiao, Glymour, Clark, Scheines, Richard, and Spirtes, Peter 2003. 'A Statistical Problem for Inference to Regulatory Structure from Associations of Gene Expression Measurements with Microarrays', *Bioinformatics* 19: 1147-1152.
- Duda, Richard O., Hart, Peter E. and Stork, David G. 2000. *Pattern Classification* (2nd edition). New York: John Wiley & Sons.
- Freund, Yoav 1995. 'Boosting a Weak Learning Algorithm by Majority', *Information and Computation* 121: 256–285.
- Glymour, Clark, and Cooper, Gregory F. 1999. *Computation, Causation, & Discovery*. Cambridge, MA: AAAI Press & The MIT Press.
- Harnad, Stevan 1994. 'Computation is just interpretable symbol manipulation; cognition isn't', *Minds and Machines* 4: 379-390.

- Hastie, Trevor, Tibshirani, Robert and Friedman, Jerome 2001. *The Elements of Statistical Learning*. New York: Springer.
- Hausman, Daniel M., and Woodward, James 1999. 'Independence, Invariance, and the Causal Markov Condition', *British Journal for the Philosophy of Science* 50: 521-583.
- Hausman, Daniel M., and Woodward, James 2004. 'Manipulation and the Causal Markov Condition', *Philosophy of Science* 71: 846-856.
- Jordan, Michael I., and Jacobs, Robert A. 1994. 'Hierarchical Mixtures of Experts and the EM Algorithm', *Neural Computation* 6: 181-214.
- Mitchell, Tom M. 1997. *Machine Learning*. New York: McGraw-Hill.
- Nisbett, Richard E., and Wilson, Timothy D. 1977. 'Telling More than We Can Know: Verbal Reports on Mental Processes', *Psychological Review* 84: 231-259.
- Pearl, Judea 2000. *Causality: Models, Reasoning, and Inference*. Cambridge: Cambridge University Press.
- Ross, Lee, and Nisbett, Richard E. 1991. *The Person and the Situation: Perspectives of Social Psychology*. New York: McGraw-Hill.
- Schapire, Robert E. 1990. 'The Strength of Weak Learnability', *Machine Learning* 5:197-227.
- Searle, John 1980. 'Minds, Brains, and Programs', *Behavioral and Brain Sciences* 3: 417-424.
- Spirtes, Peter, Glymour, Clark, and Scheines, Richard 1993. *Causation, Prediction, and Search*. Berlin: Springer-Verlag. 2nd edition 2000. Cambridge, MA: The MIT Press.
- Wolpert, David H. 1996. 'The lack of a priori distinctions between learning algorithms', *Neural Computation* 8: 1341-1390.
- Wolpert, David H. and Macready, William G. 1997. 'No Free Lunch theorems for optimization', *IEEE Transactions on Evolutionary Computation* 1: 67-82.

Further Reading

No single machine learning text includes every standard machine learning algorithm. The following books are all excellent introductions that cover a range of the machine learning literature.

Mitchell, Tom M. 1997. *Machine Learning*. New York: McGraw-Hill.

Bishop, Christopher M. 1995. *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press.

Hastie, Trevor, Tibshirani, Robert and Friedman, Jerome 2001. *The Elements of Statistical Learning*. New York: Springer.

Duda, Richard O., Hart, Peter E. and Stork, David G. 2000. *Pattern Classification (2nd edition)*. New York: John Wiley & Sons.

For the more specific case of causal reasoning methods, as well as their applicability to traditional philosophical problems such as the nature of counterfactuals, a good introduction is: Pearl, Judea 2000. *Causality: Models, Reasoning, and Inference*. Cambridge: Cambridge University Press.