Reflections on Niyogi's book "The Informational Complexity of Learning"*

Márk Jelasity
Research Group of Artificial Intelligence
József Attila University,
Szeged H-6720, Aradi vértanúk tere 1., Hungary,
jelasity@inf.u-szeged.hu

Abstract

Niyogi's book entitled "The Informational Complexity of Learning" [6] addresses the problem of learning from examples. He considers two distant fields: artificial neural networks and natural language. In both areas he gives a theoretical analysis of informational complexity, i.e. the effects of the size of the leaning set and the number of model parameters on the accuracy of learning depending on the target function class. After outlining the main ideas, this work discusses the usability of such results in practice and the relevance of the book in linguistic research, and also raises a philosophical question about the possibility of error prediction.

1 Outline of the Main Ideas

Niyogi's book entitled "The Informational Complexity of Learning" [6] addresses the problem of learning from examples. He considers two distant fields: artificial neural networks and natural language. In this first section I will introduce the theoretical framework needed to understand the basic points of the book. Due to the lack of space I will avoid unnecessary technical details while trying to remain as clear and exact as possible.

The problem of learning from examples is usually stated as the following function approximation problem. There is an unknown function $f : X \to Y$ from a function space $\mathcal{F}$. Unlike f, $\mathcal{F}$ is supposed to be known. The problem is to find

*in *Akusztikai Szemle* IV(1–2) pp19–21, 2002, book review

a function g such that g is as close to f (in some fixed sense) as possible. Studies on informational complexity of learning attempt to describe the relationships between the amount of information used to find g and the accuracy of g (i.e. its distance from f). What kinds of sources of information are available? There are basicly two sources. The first is a set of l examples $\{(x_1, y_1), \ldots, (x_l, y_l)\}$ where $f(x_i) = y_i$ $(i = 1, \ldots, l)$. One may think that the more examples we are given the closer g can be to f. In general it is true, but only under a number of assumptions about the distribution of the examples. The second source is a fixed function class $\mathcal{H} \subset \mathcal{F}$ from which we decide to choose g. This is not really new information since it is fixed *a priori*, we will get back to this problem in Section 4. This class is usually referred to as the hypothesis class. In this case the smaller $\mathcal{H}$ is the less degree of freedom we have when choosing g, or – equivalently – the fewer examples we need to find a suitable g. However, when choosing a hypothesis space which is too small, we may loose too much expressive power, i.e. we may not be able to find a suitable g in it. As is should be clear by know, the size of $\mathcal{H}$ is of major importance. For being able to control this size, the hypothesis space is usually chosen to contain functions that can be described by a set of parameters. An example for such a function could be a polynomial of degree n, where the n parameters are the coefficients. This parameterization is often chosen to allow $\mathcal{H}_1 \subset \ldots \subset \mathcal{H}_n$ where $\mathcal{H}_i$ contains the functions that can be given by at most i parameters. Thus, increasing n means increasing the expressive power but decreasing n allows using a smaller amount of training data. This makes choosing the hypothesis space a hard problem. (Choosing the size of the training set is not a problem: the larger the better.)

After this short introduction we can turn to Niyogi's results. In the first half of the book he points out that there are two kinds of learning errors that are handled separately in the literature. The first is the *approximation error* which is caused by the fact that $\mathcal{F}$ is (usually much) bigger than $\mathcal{H}_n$. If $f \notin \mathcal{H}_n$ then even the best g will introduce an error which is called the approximation error. At this point one could think that in this case we should chose a hypothesis space with as much expressive power as possible. The situation is not that simple because of another source of error: the *prediction error*. It arises because even inside the hypothesis space it is generally impossible to find the best approximation since we have only a finite amount of examples of f. Given a fixed number of examples increasing the expressive power increases the prediction error while decreases the approximation error.

Niyogi attempts to integrate these sources of error in a common framework. He derives a scheme to give error bounds with two terms corresponding to these two kinds of errors. He also gives an example using a neural network class with fixed architecture as the hypothesis class. He is able to predict the qualitative effects of choosing the number of model parameters as a function of the size of the

learning set. He shows that for a given size there is an optimal model complexity: simpler or more difficult models result in higher error with high probability.

In the second part of the book Niyogi addresses the question of natural language learning and development, again from the viewpoint of information complexity. How much samples do we need to learn a language, what kinds of errors occur and how do these errors affect the evolution of language? As a linguistic theory he accepts the principles and parameters framework of Chomsky [2]. In this framework natural language is described as a set of well-formed sentences just like in the theory of formal languages. The syntax of a language is given by general production rules (principles) having a couple of parameters making it possible to reduce language learning to setting these values. The mathematical tools used here are different. Niyogi formalizes a memoryless language learning algorithm based on [4] as a Markovian process. With the help of this formalization he can estimate the time and other properties of convergence of learning using the mathematics of Markovian chains. In the same framework he can address some phenomena of language change as well in a rigorous way. Though his simulations do not fit the empirical data (evolution of French) he offers a way to handle such questions in a controlled manner.

In the following sections I will concentrate on some problems connected to the book. I must be emphasized that these problems are the problems of the framework the author chose. This does not change the fact that the book is a valuable contribution to machine learning and Chomskian linguistics.

2 Machine Learning Theory in Practice

Niyogi's work and in general the work done in the theory of machine learning raises an interesting question: in what degree are these results useful from the point of view of practical applications? Of course, mathematics and computer science is essential when analyzing the time and storage complexity and the representational power of learning algorithms. However, one has to look at the assumptions of theoretical results closely to judge practical relevance. To put it simple, the problems arise when one wants to predict future based on insufficient – or *a priori* – information. In the following let us mention a couple of factors that lie in the way of practical applications of the results of the book (though the comments apply to many other similar approaches, as well). This is *not* a criticism since the author is aware of these problems as he mentions them at several points in the book.

The function class $\mathcal{F}$. When illustrating his method using a concrete example (a class of neural networks) the author uses a very restricted class $\mathcal{F}$. This allows

him to show that the approximation error does not depend on dimension – a very unusual situation in approximation theory. This assumption that turns out to be very useful when deriving formulas also restricts the applicability of the result to this function class.

Using *a priori* information. In machine learning, the function class $\mathcal{F}$ is usually not known. Even if the function to be approximated belongs to a certain class for which we have a nice theory it is impossible to apply it (i.e. to make *a priori* estimations of the expected error) because we don not know this property of the target function. Of course it is possible to assume such a property but this becomes an assumption about an assumption which is not necessarily easier to handle than the simple assumption that the chosen algorithm will work. Practical experience shows that, when facing a real-world problem, researchers try all available algorithms and use the one which seems to work best. The problem is that in non-trivial domains such as pattern recognition it is also non-trivial to make assumptions (especially very restrictive assumptions) about the domain.

The role of the learning algorithm The considerations about the informational complexity necessarily involves restrictive target function classes since otherwise it is impossible to derive meaningful upper bounds on the error of inductive learning from a finite set of examples. In the context of concept learning [1] mentions some of these problems. In fact a lot of methods exist that make no assumptions about the domain at all, e.g. decision-tree learning algorithms. However, in these cases the algorithm has some preferences towards some regions of the space, e.g. towards smaller trees. In these cases the *bias* is introduced by the algorithm [5].

3 Natural Language Learning

In the part on natural language Niyogi uses very simple and nice mathematics to analyze the behavior of memoryless learning algorithms in Chomsky's principles and parameters framework. Turning to his assumptions again we can conclude that they are unfortunately highly unrealistic. A lot of empirical data is available in the field of child language acquisition which clearly contradicts to memoryless learning [7]. To the contrary: in the early stages children seem to memorize a lot of (even nonsense) sentence fragments trying to figure out how to use them and what they mean. For instance they might learn the syntax of the different verbs separately first (verb islands) and they integrate them only at a later stage. There is certainly more in their heads than a set of hypothesis about parameters. It is likely that in this case the discussion of natural language learning remains a pleasant mathematical brain-exercise.

Since the author's theory about language change is completely built on the theory of learning we have no more grounds to accept it as relevant. Language change is affected by a lot of factors, which are well known for a very long time described e.g. by Saussure back in [3]. Among these are the tradeoff between intelligibility and effectiveness, different social factors, e.t.c. But there is a more serious error the author makes. He seems to indicate that languages converge to some stable state trough their evolution (though not necessarily trough an S-shaped curve). This idea is outdated as described by any introductory book to linguistics: languages are subject to a very difficult dynamics determined by the factors I mentioned above. Some languages loose certain properties while others develop them. There are no "optimal languages" nor stable states.

The idea to model language development through some explicit dynamics is great and considering a simplistic setting used in the book may be the first step towards understanding some phenomena. But it would be unrealistic to expect that this model in itself can model anything real.

4 Concluding Notes

After reading the book we can conclude that the author did a great job and made a valuable contribution to theory of machine learning and Chomskian linguistics. However, the book shows the same picture as many other books: theory is behind application. The assumptions that make mathematical analysis possible are unrealistic. It probably will not change the practice that engineers use those tools that work and not those that are predicted to work by theory. It is not very surprising since saying something relevant about the performance of an inductive learning method on an interesting unseen domain would be something like solving the problem of proving the "correctness of induction" in science, a philosophical research program the Vienna Circle failed to complete. Scientific research is very similar to inductive learning: the example set (or the set of "facts") comes from experiments, the target function is Nature (in a restricted set of situations) and the process of approximation is theory forming by scientists.[1] It turned out that the only way to check any kind of extrapolation in general is to see how it works. To put it another way, it is impossible to use the knowledge one would like to obtain for obtaining the very same knowledge.

I have to point out that the above comment is relevant only when nothing is known about the target function except the examples. In these cases the very goal is to gather this knowledge trough experimentation. It seems that in many applica-

[1] This is a rude simplification, there are other important issues like the origin of concepts used in the theories and even when collecting data (the facts are not theory independent) and so on. Anyway, the basic structure is still the same.

tions this is the case. Maybe machine learning is more like making little theories of abstract undiscovered worlds. This is what makes the field so interesting.

References

[1] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[2] Noam Chomsky. *Lectures on Government and Binding*. Number 9 in Studies in generative grammar. Foris Publications, Dordrecht, Holland, 1981.

[3] Ferdinand de Saussure. *Cours de linguistique générale*. Payot, Paris, 1939.

[4] Edward Gibson and Kenneth Wexler. Triggers. *Linguistic Inquiry*, 25:407–454, Summer 1994.

[5] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[6] Partha Niyogi. *The Informational Complexity of Learning*. Kluwer Academic Publishers, 1998.

[7] Michael Tomasello and Patricia J. Brooks. Early syntactic development: A construction grammar approach. In Martyn Barrett, editor, *The Development of Language*. Psychology Press, 1999.