Machine Learning Thoughts

Some thoughts about philosophical, theoretical and practical aspects of Machine Learning.

Learning Theory disconnected from Practice?

Is learning theory really answering the right questions?
More precisely, does working on (statistical) learning theory bounds really help designing better algorithms?

Let me first say that this note is not against anyone in particular.
I know most of the people working on SLT bounds and I do have a lot of respect for them, and I also wrote papers that were trying to justify an algorithm from a theoretical analysis. So I am just questioning the approach but not judging anyone following it!

I have spent a lot of time thinking about what theory did really bring that could not have been obtained without it. Because this is really the question. If you consider the algorithms that people use in practice, the two important questions are whether any of those algorithms could not have been inspired by considerations of a non-SLT nature, and whether the SLT analysis brings an understanding of why they work.

Regarding the first one (inspiration), it is risky to tell what could have inspired an algorithm a posteriori. But I believe that the amount of effort spent on trying to prove bounds and then obtain a criterion to minimize for an algorithm would be better used if people were just trying to find new criteria directly. Indeed, there are surely many ideas that we (theoreticians) refrain from trying or even expressing, because we do not see how they are connected to the standard patterns of SLT analysis!

Regarding the second one (whether bounds justify an algorithm), I would be even less confident: the only thing we can infer from most learning theory bounds is that the algorithm we are studying is not meaningless, that is, it will eventually make good predictions (with enough samples), but these bounds cannot really help comparing two algorithms. Also, they rarely (if ever) can justify the use of a specific criterion.

So, I think that the theoretical analysis is mainly a way to popularize an algorithm and to raise its visibility. The effect is then that more people try it out, and streamline it. So in the end, the algorithm may be adopted, but a theoretical analysis rarely justifies the algorithm and never provides guarantees. Hence theory is a way to attract attention to an algorithm.
It should also be a way to get new insights for the development of new algorithms, but this happens much less frequently than is claimed!

June 19, 2006 in Theory | Permalink | Comments (6) | TrackBack (0)

When does sparsity occur?

Sparsity is a very useful property of some Machine Learning algorithms. Such an algorithm yields a sparse result when, among all the coefficients that describe the model, only a small number are non-zero. This is typically associated with interesting properties such as fast evaluation of the model (see the reduced set methods for obtaining sparse kernel expansions), fast optimization (e.g. in SVM, many algorithmic approaches exploit this fact), statistical robustness (sparsity is usually associated to good statistical performance), or other computational advantages (e.g. ability to compute full regularization paths, for example in LASSO-style regression).

However I have not seen a clear explanation of this phenomenon. My feeling (I have no proof but it seems intuitively reasonable) is that sparsity is related to the regularity of the criterion to be optimized.
More precisely, the less regular the optimization criterion, the more sparse the solution may end up being.

The idea is that, for sparsity to occur, the value 0 has to play a special role, hence something unusual has to happen at the value 0. This something can be a discontinuity of the criterion or of one of its derivatives.

If the criterion is discontinuous at 0 for some variables, the solutions might get "stuck" in this value (provided it is a local miminum of course). If instead, the criterion is continuous but has a derivative which is

discontinuous at 0, it means that the criterion is V-shaped at 0, so that solutions might be "trapped" at this point. If we continue the reasoning, we see that the "attraction" of the point 0 is less and less effective as the regularity increases. When the function is twice differentiable everywhere, there is not any reason for the solution to be "trapped" at 0 rather than ending up somewhere else.

This reasoning partly explains the sparsity of SVMs. Indeed, the standard L1-SVM (hinge loss) have a discontinuous criterion, while for L2-SVM (squared hinge loss), the criterion has a discontinuous derivative and finally, for the LS-SVM (squared loss), the criterion is twice differentiable. It turns out that the most sparse is the L1 version and then the L2 version, while for LS-SVM there is no sparsity at all.

The same reasoning applies when one compares penalized least squares regression: when the penalization is the L2-norm of the weights, there is no sparsity, while with the L1-norm, the sparsity occurs, and for the L0-norm there is even more sparsity.

I am wondering whether there is any mathematical treatment of these issues anywhere in the Machine Learning litterature. If anyone has a pointer, please let me know.

November 08, 2005 in Machine Learning, Theory | Permalink | Comments (32) | TrackBack (0)

Learning to solve complex problems

I would like to come back to the tight relationship between optimization and learning.
Of course, there is much to say about this, but let us focus on one specific example: randomization.
Randomization is an old and well-known method for solving difficult (e.g. combinatorial) problems. Indeed, when one is faced with a complex problem where there are many combinations to try out, and very little structure in the search space, it is often convenient to introduce some randomness in the search.
As it turns out, you can often prove that a "stupid" random method will perform as well as a sophisticated one in terms of how close you can get to the optimal solution as a function of the computation time spent.

Let me give you some simple example: consider the d-dimensional unit cube and assume you are given a set of k d-dimensional hyperrectangles defined by the coordinates of their borders (2d such borders, hence 2d numbers characterize each hyperrectangle) that lie inside the unit cube (the numbers are all between 0 and 1).
The problem is to determine the volume left out by these hyperrectangles.

Put in simple words, you have a cubic space from which you cut out (possibly overlapping) rectangular pieces and you want to determine the volume left out after you cut k such pieces out.

This seems like a simple problem but solving it exactly requires heavy computations: you essentially need to determine all intersections between hyperrectangles and the intersections between these intersections and so on... Of course this can be done, but will require exponential time in d.
This means that it is practically impossible as soon as the dimension gets bigger than say 20.

So, instead of trying to solve it exactly, one can try to approximate the solution with the following simple randomized method:
you imagine throwing a stone in the cubic space repeatedly and counting how many times it hits a hyperrectangle. The fraction of such hits will very quickly converge to the solution of the problem.

More precisely, imagine you can sample points in the d-dimensional unit cube independently and uniformly. For each such point it is easy to determine whether it is inside a hyperrectangle (it takes 2dk comparisons at most) and thus if one samples n such points, with a computation time 2dkn, one gets an estimate of the volume v which is (with large probability) at least within epsilon of v.
The size of this epsilon is roughly n^{-1/2} so that with 10000 draws you get 1% precision in your estimation, and this is **independent** of the dimension.

So you can have, in high dimensional spaces, a very cheap solution for solving the problem (which takes time linear in the dimension and not exponential, at the expense of yielding an approximate solution).
This is actually a very common phenomenon which has been largely exploited in solving many high

dimensional problems.

Now the connection to learning: the randomized method proposed above is a learning algorithm. Indeed, you are trying to "learn" the fact that a point in the unit cube is covered or not by one of the hyperrectangles. It is a very simple learning problem since one is only interested in the volume of the empty space and not in a full description of this empty space, but the techniques that are required are basically similar. More precisely, the estimation of the error in the approximation involves computations of the same type as those used in learning theory for deriving bounds on the generalization error of learning algorithms.

Of course this was a very simple use of learning and there are plenty of ways to make more sophisticated uses of learning in solving complex problems. The message is that statistical learning theory can be useful for analyzing algorithms for solving complex problems (and not only for analyzing learning algorithms as such).

October 07, 2005 in Machine Learning, Theory | Permalink | Comments (2) | TrackBack (0)

The Meaning of Probability

The formalization of the concept of probability has a long history. Probability Theory is now a well-founded and very mature part of Mathematics, mainly due to its axiomatization by Kolmogorov who grounded the concept of probability in measure theory.
It may thus seem that defining and combining probabilities can be done in a unique way, without any questions.
However, there is still a lot of disagreement on the crucial issue: the interpretation of probability. The problem here is that interpretation means connection to the real world. In that respect, the issue is not just a technical one but also a philosophical one, which explains why there can be many different points of view.

First of all, it is possible to distinguish between the *objective* and the *subjective* points of view:

- Objective probability: the objective point of view consists in postulating that probabilities do not depend on the person observing events or performing experiments. This means that there exists some absolute notion of probability for every possible event and this probability originates from Nature itself. Once this is assumed, the question becomes : how to "measure" these pre-existing probabilities, or how to confirm that the probability of a given event has a given value?
- Subjective probability: in the subjective point of view, probabilities are not something that can be measured, but something one *assumes*. The idea is that events either occur or do not occur and the probability is not a property of Nature but rather a convenient way of representing someone's uncertainty prior to the event actually occurring.

There are two classical (and opposed) ways of interpreting probabilities: the *frequency* and *Bayesian* interpretations.

- Probability as frequency: in this approach, the probability of an event is defined as the ratio of how many times the event occurs to the number of times a similar experiment is performed. For example, if you repeatedly flip a coin, the probability of this coin landing on "heads" will be defined as the percentage of trials where it does land on "heads". Of course, this will highly depend on the number of such trials and may vary from one sequence to another. However, this issue is solved by the theorem called "the law of large numbers" which essentially states that the frequency of an event in successive independent trials will converge to a fixed value (its probability). In other words, if you flip your coin again and again, the frequency will (slowly but surely) converge to a definite value. There are some issues about the definition of independent trials and about the fact that one can really perform successive experiments in exactly the same way, but we will not worry about this now.
- Bayesian probability: it is obvious that not all notions of probability (as they are used in every day life) can be properly captured by the frequency definition given above. For example, when one speaks about the probability of an event that may occur only once (hence it is not possible to perform repeated experiments) such as the probability of a politician being elected at a given election, it is clear that frequency does not make practical sense and cannot be tested. Another issue with frequency is that it

makes sense in the limit only: say we start flipping a coin and it keeps landing heads up; how many times does it need to land heads up before we decide that this is not happening with probability 1/2? Five? Ten? A thousand? A million? There is no reasonable answer to this question. Hence (subjective) Bayesians do not attempt to measure probabilities, rather they consider that a probability is a "degree of belief" that someone may have in the fact that a given event will occur. The whole point is that how you obtain your "prior" probability or initial degree of belief (before observing anything) does not matter. What matters is how these values are combined and updated when events are observed.

There is of course a lot to be said about the above two interpretations and there are many refinements or deviations from these. I hope to be able to explore this in more details in later posts.

September 25, 2005 in Machine Learning, Philosophy, Theory | Permalink | Comments (10) | TrackBack (0)

How to be a Bayesian while pretending you are a Frequentist?

Using the Bayesian modeling approach is very convenient when designing learning algorithms. You just have to encode your prior knowledge and assumptions into a prior distribution and a likelihood function and then you can simply unroll the Bayes rule mechanism. Of course there are usually sophisticated computational problems, but no conceptual ones.
So-called "frequentists" usually try to justify the algorithm they design based on a theoretical analysis in the iid framework. Hence it is common practice (I am not saying here that I like this practice) to use a probabilistic error bound as a justification for a new algorithm.
The problem is that it is usually very hard to obtain such a bound and there is no unique way or guiding principle for doing so.

Fortunately, people have recently obtained bounds that involve a prior over a class of functions. They are called "PAC-Bayesian" bounds and are quite easy to apply to several different algorithms.
In a way this gives rise to a new principled method for producing new algorithms. Here is the recipe:

- You encode your prior assumptions into a prior distributions (just like in the Bayesian case)
- You plug this distribution into a PAC-Bayesian bound
- You use the bound as an objective criterion, so that the algorithm is simply minimizing the bound

This gives something that is similar in spirit to the MAP (maximum a posteriori) method. In my opinion, this is not any better or worse than MAP and there is no reason to believe that this way of designing algorithms is better than the Bayesian way.
Unfortunately, people tend to think that because the algorithm was derived from a bound which is some sort of formal mathematical statement, then it should be a better algorithm that if this were not the case.

August 14, 2005 in Machine Learning, Theory | Permalink | Comments (0) | TrackBack (0)

Measurability and the Axiom of Choice in Learning Theory

This post is quite technical, but this question has been obsessing me for a while.
The starting point is the theorems about universal consistency of learning algorithms: a learning algorithm is universally (strongly) consistent provided, for any probability measure, as the sample size grows to infinity, the error of the learning algorithm converges (almost surely) to the Bayes error (the best possible error).

There is no issue in countable spaces, because everything in the above statement can be defined in a straightforward way.
However, if you work in an uncountable space (like the set of real numbers), things get more complicated. Of course one may argue that it is unnatural to model data processing phenomena (which necessarily deal with finite precision numbers in practice) with such an abstraction as the set of real numbers, but it is so convenient that nobody really argues...

So, in the real numbers, to define probabilities properly, one has to introduce a sigma-algebra, that is a collection of sets (with certain properties) on which the probabilities are defined. The elements of the sigma-

algebra are called "measurable" and they are the only ones for which probabilities make sense. The point is that not every subset of the reals is measurable (at least in the standard Lebesgue construction) and this has to do with the Axiom of Choice. A consequence of this construction is that the "Bayes classifier" (i.e. the best possible prediction model for classification) is a measurable function. And the existence of universally consistent learning algorithms is a consequence of the fact that every measurable function can be approximated by "simple" functions which you can learn from finite samples.

In other words, when your input space is the set of real numbers, universal consistency of a learning algorithm is related to the measurability property.
So the questions that comes to mind are:

- Can one imagine a model of learning on the real numbers where universally consistent algorithm exist but where there is no requirement of measurability for the Bayes classifier?
- Would it help to work in a context where the axiom of choice is replaced by an axiom such as "all subsets of R are measurable"?
- Should one just forget about the real numbers when dealing with learning theoretic questions and rather use another model?

July 19, 2005 in Theory | Permalink | Comments (1) | TrackBack (0)

Foundations of Learning Theory

In my opinion, the field of Machine Learning still lacks theoretical foundations. Indeed, even if there are many people working on "Learning Theory", most of them (including myself) work on theoretical problems inspired by Machine Learning.
So, what is probably missing is some kind of agreement on what exactly is learning theory.
What is called "Statistical Learning Theory" is probably the most well-defined part of learning theory, as it is based on a mathematically sound framework. But unfortunately, not all learning problems can be studied in this framework, so there is room for either a broader framework, or other complementary theories.

Of course, it is rarely the case that the framework comes before the results: usually people produce research papers proving some results in some setting and ultimately, a bunch of such papers forms a theory (after many people read them, digest them and try to get the main elements out of them).

Moreover, it is probably a waste of time to try and argue which existing results are or are not learning theory results.

So some questions arise

1. Why do we need to define learning theory?
2. What do we expect from such a theory?
3. Assuming we want to do it, what are the first steps to take?

Here are some possible answers

1. Defining learning theory would, firstly allow the gathering and unification of the existing results and secondly allow to identify the remaining open questions this theory should address.
2. Ideally, such a theory should allow to build better learning machines. In the case this is not possible, it should at least allow to understand why this is the case.
3. The first steps could be to classify existing models of learning and extract their commonalities.

Here are some constraints one should have in mind:

- A theory is merely a formal representation (or model), of some "real" phenomenon. As such, it has limitations partly due to the necessary simplifications it introduces.
- Within a theory, equally important are positive and negative results. In other words, the limits of the theory should be clear and impossibility results should be produced to investigate these limits.
- Predictions made by a theory should be testable (this in connected to Popper's falsifiability).

The last point is probably the most important, but also the one that is most subject to controversy. Indeed, testing predictions is somewhat possible for theories about natural phenomena (Physics mainly), but, in the case of learning, the goal is not necessarily to build a theory that accounts for the forms of learning (animal or human) that we observe in Nature, but rather to build some kind of "general" theory of learning. Hence testability may become an issue. In a way, similar issues are encountered with probability theory: is this a theory about natural phenomena? can its predictions be tested?
Answering these questions requires to build some kind of connection between the abstract concepts of the theory and a concrete interpretation of it. Since this is still being debated for probability theory, it is probably far to be resolved for learning theory...

July 12, 2005 in Theory | Permalink | Comments (0) | TrackBack (0)

Clustering Workshop

This week was held the Statistics and Optimization of Clustering Workshop (supported by the PASCAL network). There were many interesting talks and in particular, there was a great open discussion about foundations of clustering.
Several issues were raised in this discussion, and it seems that none of them could be settled in a satisfactory manner.
Among these issues, I noted the following:

1. Definition of clustering: what is the goal of clustering? what does it mean to extract hidden information, or hidden structure from the data? The main issue here is that there are plenty of possible definitions but there does not seem to be one that encompasses every other.
2. Difference between clustering and dimensionality reduction: can these things put together in the "exploratory data analysis" category? Does it make sense to separate them?
3. Clustering with a goal in mind: most often, clustering is used as a first step of data analysis. This means that there are other steps, and defining the goal of clustering only makes sense when one knows what is really the ultimate goal (for example, in many cases, people use clustering as a pre-processing step for supervised learning).
4. What is generalization in clustering: what is an appropriate formal setting for the clustering problem? Can we use the iid setting of supervised learning, where one is concerned with convergence of the generalization error as the sample size grows? Or should we (as proposed by Tali Tishby) be concerned with convergence as the number of features grows?
5. Evaluation of clustering algorithms: is there some way to tell two clustering algorithms apart in terms of their "generalization ability" or their ability to find structure in the data? This seems like the most important yet most unclear point. Most people seemed to agree that a desirable feature of a clustering algorithm is stability, but that this is far from being sufficient. Some people were saying that the only way to evaluate the results is to show them to an expert. Others said that there should be a theory that allows to compare clustering algorithms.

The conclusion is that there are plenty of interesting open questions here, and many (brilliant) people are willing to work on them.
So I hope that progress will be made soon, and bring unsupervised learning to the level of formalization that supervised learning has reached.

July 08, 2005 in Theory | Permalink | Comments (0) | TrackBack (0)