Is learning possible without Prior Knowledge?
Do Universal Learners exist?

Shai Ben-David
with
Nati Srebro and Ruth Urner

Philosophy of Machine Learning Workshop,
NIPS, December, 2011

High level view of (Statistical) Machine Learning

"The purpose of science is

to find meaningful simplicity

in the midst of

disorderly complexity"

Herbert Simon

However, both these notions are subjective

Naive user view of machine learning

"I'll give you my data, you'll crank up your machine and return meaningful insight"

"If it does not work, I can give you more data"

"If that still doesn't work, I'll try another consultant" ....

The Basic No Free Lunch principle

No learning algorithm can be guaranteed to succeed on *all learnable* tasks.

Any learning algorithm has a limited scope of phenomena that it can capture, (an inherent *inductive bias*).

There can be no *universal learner.*

Vapnik's view

"The fundamental question of machine learning is:

What must one know *a priory* about an unknown functional dependency in order to estimate it on the basis of observations?"

Prior Knowledge (or Inductive Bias) in animal learning

The Bait Shyness phenomena in rats:

When rats encounter poisoned food, they learn very fast the causal relationship between the taste and smell of the food and sickness that follows a few hours later.

Bait shyness and inductive bias

However, Garcia et at (1989) found that:

When the stimulus preceding sickness is sound rather than taste or smell, the rats fail to detect the association and do not avoid future eating when the same warning sound occurs.

Universal learners

Can there be learners that are capable of learning ANY pattern, provided they can access large training sets?

Can the need for prior knowledge be circumvented?

Theoretical universal learners

- Universal priors for MDL type learning.

(Vitanyi, Li, Hutter, …)

Hutter: “Unfortunately, the algorithm of is incomputable. However Kolmogorov complexity can be approximated via standard compression algorithms, which may allow for a computable approximation of the classifier ”

(we will show that that is not possible)

- Universal kernels (Steinwart).

Practical universal learners

- Lipson's "robot scientists"

http://www.nytimes.com/2009/04/07/science/07robot.html?_r=1&ref=science

- Deep networks (?)

Yoshua Bengio: "Automatically learning features allows a system to learn complex functions mapping the input to the output directly from data, *without depending completely on human crafted features*."

The importance of computation

- We discuss universality in **Machine** Learning.
- Machine compute, hence emphasis on computation.
- Leaving “computational issues” to “practitioners” is dangerous!

Our formalism

- We focus on binary classification tasks with the zero-one loss.
- X is some domain set of instances, training samples are generated by some distribution D over $X \times \{0,1\}$, which is also used to determine the error of a classifier.
- *We assume that there is a class of "learners" that our algorithm is compared with*

(in particular, this may be a class of labeling functions).

What is Learnability?

There are various ways of defining the notion of "*a class of functions*, $\mathbb{F}$, *is learnable*" .

- Uniform learnability (a.k.a. PAC- learnability).
- The celebrated Vapnik-Chervonenkis theorem tells us that only classes of finite VC-dimension are learnable in this sense.

- Thus ruling out the possibility of universal PAC learning.

A weaker notion- Consistency

- A learner is *consistent* w.r.t a class of functions, $\mathbb{F}$, if for every data-generating distribution, the error of the learner converges to the minimum error over all members of $\mathbb{F}$, as the training sample size grows to infinity.

  A learner is *universally consistent* if it is consistent w.r.t. the class of *all* binary functions over the domain.

The (limited) significance of consistency

One issue with consistency is that it does not provide any finite-sample guarantees.

On a given task, aiming to achieve a certain performance guarantee, a consistent learner can keep asking for more and more samples, until, eventually, it will be able to produce a satisfactory hypothesis.

It cannot, however, estimate, given a fixed size training sample, how good will its resulting hypothesis be.

Some evidence to the weakness of consistency

Memorize is the following "learning" algorithm:

store all the training examples,

when required to label an instance,

predicts the label that is most common

for that instance in the training data

(use some default label if this is a novel instance).

- Is Memorize worthy of being called
- "*a learning algorithm*"?

A rather straightforward result

Over any countable domain set,

Memorize is a successful universal consistent algorithm.

(There are other universally consistent algorithms that are not as trivial –

e.g., some nearest-neighbor rules, learning rules with a universal kernel)

Other formulations of learnability

- PAC learnability requires the needed training-sample sizes to be independent of the underlying data distribution and the learner (or labeling function) that the algorithm's output is compared with.
- The consistency success criterion allows sample sizes to depend on both.
- One may also consider a middle ground.

Distribution-free Non-uniform learning

A learner, A, *non-uniformly learns* a class of models (or predictors) H, if there exists a function m: $(0,1)^2 \times H \rightarrow N$ such that:

For every positive ϵ and δ for every $h \in H$, if $m \geq m(\epsilon, \delta, h)$, then

$D^m [\{S \in (X \times \{0,1\})^m : L_D(A(S)) > L(h) + \epsilon\}] \leq \delta$

for every distribution, D

Characterization of DFNUL for function classes

Theorem: (For classification prediction problems with the zero-one loss).

A class of predictors is non-uniformly learnable if and only if it is a countable union of classes that have finite VC-dimension.

Proof

- If $H=\bigcup H_n$, where each H_i has a finite VC-dimension, just apply Structural Risk Minimization as a learner (Vapnik).

- For the reverse direction, assume H is non-uniformly learnable and define, for each n, $H_n=\{h \in H: m(0.1, 0.1, h) <n\}$

  (by the No-Free-Lunch theorem, each such class has a finite VC-dim).

Implications to Universal learning

Corollary:

There exists a non-uniform universal learner over some domain X, *if and only if* X is finite.

Proof: Using a diagonalization argument, one can show that the class of all functions over an infinite domain is not a countable union of classes of finite VC-dimension.

The computational perspective

Another corollary:

The family of all computable functions is non-uniformly universally learnable.

Maybe this is all that we should care about – competing with computable functions.

But, if so, we may also ask that the universal learner be computable.

A sufficient condition for computatable learners

If a class H of computable learners is *recursively enumerable*, then there exists a computable non-uniform learner.

What about the class of all computable learners (or even just functions)?

A negative result for non-uniform learnability

Theorem: There exists no computable non-uniform learner for the class of all binary-valued computable functions (over the natural numbers).

Proof idea

We set our domain set to the set of all finite binary strings. Let L be some computable learner and let D_m denote the set of all m-size binary strings. Define f_m to be a labeling function that defeats L over D_m w.r.t. the uniform distribution over D_m (the proof of the NFL theorem gives an algorithm for generating such f_m). Let $F=\cup f_m$

Can a single learning algorithm compete with all learning algorithms?

Corollary: There exists no computable learner *U*, so that for every computable learning algorithm, *L*, every ϵ>0, for some m(ϵ, L), for every data-generating distribution, on samples *S* of size >m(ϵ,L), the error of *U(S)* is no more than *L(S)*+ ϵ.

Do similar negative results hold for lower complexity classes?

Theorem: If T is a class of functions from N to N so that for every f in T, $2^m f(m)$ is also in T, then no learner with running time in T is universal with respect to al learners with running time in T.

Note that the class of all polytime learners is not of that type ☹

Polytime learners

Goldreich and Ron (1996) show that there exists a polynomial time learner that can compete with all polynomial time learners (in terms of its error on every task) *ignoring sample complexity*.

(in other words, polytime learner that is consistent w.r.t the class of all polytime learners).

The result can be extended by replacing “polytime” by “computable”.

Open question

Does there exist a Polytime learner that NUDF competes with the class of polynomial-time learners?

Conclusion

There exist computable learners that are “universal” for the class of all computable learners either with respect to running time, or with respect to sample complexity, but not with respect to both (simultaneously).

Implications for candidate universal learners

They are either not computable (like those based on MDL) or they do not have guaranteed generalization (uniformly over all data-generating distributions).

Can we come up with formal finite-sample performance guarantees for Deep Belief Networks, or MDL-based learners?