Sparsity in Adaptive Control

Philippe Preux, Sertan Girgin*

January 15, 2009

Current trends in machine learning tend to equate sparsity with kernel methods along with l_1 regularization. We do share this point of view quite much, as these are principled ways to reach sparsity. However, we would like to discuss less theoretically grounded methods that also lead to sparse solutions to represent real functions. Our experience here is focused on adaptive control tasks, namely approximate dynamic programming, and reinforcement learning. These tasks share some idiosyncracies that we mention below, and are in some sense, fundamentally different from traditional supervised learning.

Actually, sparsity may be met with probably any non parametric function approximator (NPFA), that is, any function approximator which structure adapts to training data. In particular, we have investigated cascade-correlation networks (CCN) to learn optimal control. In a few words, either approximate dynamic programming, or reinforcement learning is considered, given some function V_k at step k, and given some data (either collected using V_k, or collected once for all at the beginning of the process), we learn a new V_{k+1}. We are interested in the fixed point, or limit point when k goes to ∞ (in theory, both exist, and are equal).

Beyond the details, the central point here is that the data from which we learn are not examples with the usual meaning in supervised learning. Furthermore, we deal with a closed-loop system, which means that the current estimate of the function being learned influence the data on which the function will be estimated at the next step. So, we may see the problem as a sequence of regression problems.

The traditional approach is to use parametric function approximators to represent the learned functions in these settings (mostly multi-layer perceptrons). Recently, we have investigated the use of CCN. This shows:

- very striking improvements in the learning curve,
- since the CCN has a structure that adapts to the complexity of the function to learn, CCN leads to the possibility to obtain information about the problem after the learning by examining the features that have been automatically extracted,
- much sparser function representations, than with the usual parametric approach (by orders of magnitude),
- this leads to severe CPU-time and space downsizing,
- furthermore, CCN are very easy to implement, in an efficient way.

The shortcoming is the current lack of formal results when NPFA are used (*e.g.* regarding convergence). Furthermore, we consider here the problem of learning faster, not the problem of being able to learn. A formal representation of learning curves would probably provide a very significant help to obtain the best we can expect to get from NPFA in such learning setting.

As a last remark, also investigating with LARS-like algorithm, we have opposite experimental results in supervised learning, and adaptive control: while kernel methods perform better than CCN in supervised learning, CCN outperforms kernel methods in optimal control. This observation requires further work.

More details may be found in our papers at EWRL'2008, ECAI ERLARS-workshop, ICML&A'2008, and ADPRL'2009, available *via* `http://www.grappa.univ-lille3.fr/~ppreux/papers`.

*SequeL team-project, INRIA Lille Nord-Europe, & Computer Science Lab. of the University of Lille, Villeneuve d'Ascq, France. Sertan Girgin is on leave. `philippe.preux@inria.fr`