The sample Complexity of Learning the Kernel

Shai Ben-David

Based on work with Nati Srebro

NIPS Kernel *Learning* Workshop 2008

The modeling of prior knowledge

- No learning is possible without applying prior knowledge, or learning bias (this is the “no free lunch” phenomena).
- Kernels are a common tool for introducing learning bias. They are supposed to express prior knowledge regarding how likely are two domain element to have the same label.

What happens in real-life?

In many real-life cases, a choice of a task-suitable kernel requires more detailed prior knowledge about the task at hand than is usually available to the designer of the learning algorithm.

Often, a learner uses its training data to choose a kernel for SVM learning, to tune up the kernel parameters ***and*** to learn under that kernel.

“Automated selection of kernels”

The workshop’s title, “Automated Selection of Kernels”, seems to refer to the kernel selection aspects that are not carried out on the basis of domain expertise – that cannot be automated.

The ‘automatization’ that the title refers to is probably based the use of training data for selecting kernels and for tuning up the kernel parameters.

While there is a lot of research effort is devoted to the computational efficiency of kernel learning, the sample-complexity implications of such learning paradigms remains largely outside the focus of attention of this community.

Raising awareness to the sample costs of kernel learning

The use of training data for selecting and tuning learning algorithms is commonly done “under the table” from the point of view of the statistical theory of generalization performance guarantees.

One potential merit of this workshop is raising awareness to the sampling complexity costs of that practice. (While we may take at face value researcher’s practices, we tend to require more accountability from machines ...).

The sample complexity question

The question we wish to understand is to what extent can the γ simultaneous search for a kernel and a hypothesis with respect to that kernel lead to overfitting.

Given a family of kernels, $\mathcal{K}$, and a margin value, γ, what sample size is needed to guarantee that with high probability, for every h that is a γ-margin hyperplane with respect to any kernel K in $\mathcal{K}$, its empirical error is close to its true error.

The “richness” of a family of kernels

First, one has to note, that if we do not restrict the family of candidate kernels, then we are doomed to overfitting.

Namely, for any possible sample labelling there exist a kernel relative to which there is a large-margin hyperplane that induces that labelling.

A combinatorial measure of that richness

The ‘pseudo-dimension’ is a straightforward generalization f the VC-dimension to classes of real-valued functions.

In the work that Nati will talk about, we show that the pesudo-dimension of the family of kernels can be used to provide such generalization error bounds. The smaller the dimension of a class of kernels, the smaller is the sample size one needs to avoid overfitting when searching for a kernel (and a label predictor) over that class.

Bounds for specific kernel families

We then go on to compute that dimension for some natural parameterized families of kernels, and obtain error generalization bounds for algorithms that use the training data to search for a kernel within such a family.

Other types of data for learning kernels

There are at least three other potential sources of data that can be used to guide the search for a good kernel.

1. Unlabeled data, in the Semi-Supervised Learning (SSL) setting.
2. Data from different related tasks, in the Multi-Task Learning (MTL) setting.
3. Data from different views of the same task – the Multi-View setting.

Prior-Knowledge Expression – A major (under researched?) Challenge

In all three settings the first challenge is to find suitable formalisms for expressing prior knowledge about the relationships between the external source of data and the target classification task.

Such formalisms should, on one hand, allow natural expression of domain-expert beliefs and, at the same time, allow derivation of provably significant merits of such knowledge.

Expressing SSL prior beliefs

The cluster assumption is a popular high-level (or “soft”) type of prior belief. Hardly ever explicitly defined, it asserts that data clusters tend to have homogeneous labels. Roughly speaking, we wish to say

“separators that pass through low-density areas of the unlabeled distribution are more likely to predict well”.

An SSL Kernel learning challenge

Find formal tools for expressing such cluster assumptions and kernel learning algorithms that, under such assumptions, utilize unlabeled data to find good kernels for classification.

The multi-task learning setting

Consider the setting in which a learner is faced with a collection of classification tasks, that are related in some way.

How can availability of labeled samples for each of these tasks, help find a good kernel for learning one target task in that collection?

Note that we wish to succeed on a specific target task, rather than on average over randomly drawn tasks.

Kernel learning in MTL

One way of modelling the relationships between the different tasks is to fix a family of potential kernels, $\mathcal{K}$, and assume that there exist some kernel K in $\mathcal{K}$ that works well for all of the tasks.

By restricting the richness of $\mathcal{K}$, one can demonstrate the utility of multi-tasking for the *average* learning performance.
However, not for any specific target task.

Additional task-relatedness assumptions.

In [BD-Schuller 07] we propose a notion of task-relatedness that models situations in which there is some family of 'task transformations', such that all the tasks in the MTL collection are such transformations of each other.

Under such assumptions, we can strengthen "learning a kernel that is good on the average" results, to learning a kernel that works well for any *specific target task*.

Conclusions

- The sample complexity of learning a kernel should be taken into account when considering automated kernel learning.
- We analyse this sample complexity for searches within some types of kernel families (Nati's talk).
- For learning the kernel from "auxiliary" data, the formalization of prior knowledge is a challenge of prior significance.
- Such formalization should be both "user friendly" and allowing derivation of performance guarantees.