Программа учебной дисциплины «Введение в теорию статистического обучения»

Утверждена
Академическим советом ООП
Протокол № 2.3-09/3008-01
от 30 августа 2018г.

Syllabus

1. Course description.

a Title of the course : **Introduction to Statistical learning theory**.

Lecturer: Bruno Bauwens

b Pre-requisites : Linear algebra, probability theory and discrete mathematics.

c Course Type : Elective.

d Abstract : We study a theory that inspired the development of two important families of machinelearning algorithms: support vector machines and boosting. More generally, in a typical classification task we are given (1) a dataset for training and (2) a familyof classifiers, indexed by some parameters that need to be tuned. A learning algorithm uses the training set to select one of the classifiers, in other words, tune the parameters. For example, given a neural network, every choice of the weights specifies a classifier and the learning algorithm assigns values to the weights.
On one side, we want to have a large set of classifiers to model all structure in the training data. This might lead to a large number of parameters to train. On the other hand a too large set of classifiers might lead to a decrease of accuracy on unseen examples. This is called overfitting. We study a theory that quantifies overfitting theoretically. Moreover, support vector machines and boosting algorithms can be seen as algorithms that optimize the trade-off discussed above under some additional assumptions. Moreover, the theory can determine good values for meta-parameters in machine learning algorithms that otherwise need to be tuned using cross-validation. We also study some recent deep boosting algorithms that were developed using the theory. These algorithms are currently among the best for classification tasks when the number of classes is high. Finally, we study the online mistake model. This model is more general but its mathematical analysis has many similarities with the theories above.

2. Learning Objectives

- Understanding basic concepts from statistical learning theory.
- Being able to understand the connection between these models and many machine learning algorithms.

- Training of mathematical skills such as abstract thinking, formal thinking and problem solving; with emphasis on statistics.

3. Learning Outcomes

- Knowledge of several paradigms in statistical learning theory to select models (Structural risk minimization, Maximal likelihood, Minimal Description Length, etc.)
- Calculate sizes of trainingsets for several machinelearning tasks in the context of PAC-learning (and hence calculate VC-dimensions)
- Understand the link between cryptography and computational limitations of statistical learning.
- Deeper understanding of boosting algorithms and support vector machines.
- Theoretical understanding of several online learning algorithms and learning with expert advice.

4. Course Plan

1. Probably approximately correct learning

As an introduction we study binary classification of points in the plane using nearest neighbor and axis aligned rectangles. Then the general problem is formulated: for most of the course we study classification and assume that the training and data set is obtained by independent sampling from the same probability distribution. We show that in this setting, we need to make some assumptions on the set of classifiers that we are fitting. The corresponding theorems are known as «no-free-lunch theorems». We define sample complexity, PAC-learnability and some equivalent characterizations of a set of classifiers.

2. VC-dimensions

The VC-dimension of a set of classifiers is a measure for the richessness of the set. Roughly stated, the higher this dimension, the more regularities it can model. The fundamental theorem of statistical learning theory states that the VC-dimension of a set of classifiers is finite if and only if it is PAC learnable (i.e., we can learn a correct classifier with any precision using finitely many samples). A quantitative version allows us to interpret the VC-dimension as measure for the risk for overfitting. We study two versions (the realizable and the agnostic version).

3. Structural risk minimization and variants

The Bayes-variance decomposition concerns the difference between the error of a learned classifier and the best possible classifier (also known as «Bayes optimal classifier»). This error is the sum of a bias, due to the limited size of the set of classifiers, and the variance, due to the overfitting error, which typically increases as the size of classifiers becomes larger. For several

learning algorithm, this trade-off allows us to assign values to hyperparameters that otherwise need to be assigned using cross-validation. For example, we can use this to determine a good size of decision trees and neural networks.

4. The time complexity of learning and cryptography

A typical cryptographic encoding of a string is an example of a an object with a clear structure, but of which no learning algorithm can find the structure of the object in a reasonable time. In this part we discuss some learning tasks that are impossible for computational reasons. Under a plausible computational complexity assumption (which is required for secure RSA encryption) one can show that neural networks of small depth and regular languages can not be learned.

5. Boosting

A *weak learning* algorithm can generate from a train set a model that is slightly better than random guessing. We study AdaBoost and DeepBoost, two algorithms that are currently very popular in machine learning. Furthermore we prove performance guarantees and conclude that weak learning is equivalent to PAC-learnability. However, these performance bounds do not explain observed data well. This motivates the approach of the following topic.

6. Rademacher complexities

Similar to the VC-dimension of a set of classifiers, the Rademacher complexity quantifies the richessness. However, unlike VC-dimensions, the quantity does not change if we add signs of convex combinations of classifiers. Using Rademacher complexity we develop a theory where risk bounds are more closely related to experminentally observed generalization errors. We use the theory to determine risk bounds for neural networks and argue that dropout regularization is very effective for deep neural networks.

7. Support vector machines and margin theory

We review the theory of support vector machines and Kernels. We argue that they optimize risk bounds under the large margin assumption. Then we use the theory to derive a polynomial time algorithm that learns improper representations of L1-regularized neural networks with bounded depth.

8. Mutliclass classification and DeepBoost

Risk bounds using Rademacher complexities are generalized for multiclass classfication and we study the algorithms from [3] in the references below.

9. Online learning

In online learning there is initially no training set. After each prediction, the class of the label is declared and the learning algorithm can use this information to imporve the prediction model. We study 1) prediction with expert advice, 2) linear classification algorithm such as the

perceptron algorithm.

10*. Reinforcement learning

In the unlikely case there is time left, we study reinforcement learning. This part will not be a part of the exam materials (unless at least 1 student asks for a bonus question). We are interest to design an agent that succesfully picks up rewards while navigating in an environment. We model the environment as a Markov decision process and use Bellman equations to define optimal behavior. Then we discuss several planning algorithms both for the cases where the agent knows and does not know the model of the environment.

5. Reading List

a) Required:

- [1] Lecture notes available on http://wiki.cs.hse.ru/

b) Recommended:

- [2] Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar. "Foundations of machine learning (adaptive computation and machine learning series)." (2012).

6. Grading System

$$O_{accumul.} = (1/2)\cdot(O_{HW} + O_{intermediate\ exams})$$

$$O_{final} = (1/3)\cdot(O_{HW} + O_{intermediate\ exams} + O_{final\ exam})$$

The intermediate exam happens during the first lecture of the 2nd module.

$O_{intermediate\ exams} = (1/2) \cdot (O_{colloquium} + O_{problem\ exam})$.

Rounding policy. All grades are calculated exactly. Only the final grade is rounded. It will be rounded up if the score is at least 5.5/10. Otherwise, it will be rounded down.

Homework policy. There will be 3 homeworks. Homeworks should be typed in latex and submitted by email. People can collaborate, however the typing should be done individually. During the colloquium exam and final exam the teacher will interrogate the student about his submitted answers.

7. Guidelines for Knowledge Assessment

During the colloquium, the student needs to reproduce proofs from the lectures and answer additional questions to check understanding. Questions for the colloquium will be made available in advance.

Similarly, the final exam contains 2 parts with equal weight: one (written) part focusses on theory (the students need to reproduce one or more of the proofs and answer questions to check

understanding. The exercise part will be open book: students can bring the lecture notes from the wikipedia site, handwritten notes, and selected parts from [2] and [3] above.

8. Methods of Instruction

There are 13 lectures, each lecture takes 80 minutes. Each lecture is followed by a seminar, that also takes 80 minutes. The students are encouraged to ask for more explanations after the lecture and during the office hours (see the wikipage, it is recommended to email in advance).