Machine Learning and Data Mining

Updated course program

Maxim Borisyak

September 19, 2016

National Research University
Higher School of Economics

Program

Outline

The course has total 18 lectures and seminars. As usual there are 3 parts:

- 'shallow' Learning: 9 weeks;
- Deep Learning: 3 weeks;
- Big Learning: 5 weeks;
- individual projects: 1 week.

Disclaimer

The following program is most likely to be
an optimistic estimate.

We will try to adopt to the situation by omitting advanced topics.

'Shallow' learning I

1. Introductory lecture:
    - logistics.
2. HSE day.
3. Philosophical note:
    - definitions;
    - no-free-lunch theorem.
4. Meta-learning I:
    - bias-variance decomposition;
    - cross-validation;
    - bagging: Random Forest, Extra Trees;
    - stacking: calibration, ensemble.

'Shallow' learning II

1. Regularization:
    - model complexity;
    - l_1 and l_2 regularization;
    - feature selection via regularization;
    - Decision Tree regularization.
2. Meta-learning II (Boosting):
    - AdaBoost;
    - Gradient Boosting;
    - XGBoost.
3. Practical session:
    - Gradient Boosting practice;
    - `homework`: Viola–Jones (Haar) cascades.

'Shallow' learning III

1. Optimization methods for ML:
   - gradient descent and stochastic gradient descent;
   - adaptive methods;
   - second order methods;
   - global optimization.
2. Notes on real-world problems:
   - imbalanced datasets;
   - reweighting;
   - semi-supervised learning;
   - invariant classifiers.

Deep Learning

Since we agreed that you all will attend 'Deep Learning' course by Alexander Panin, instead of introductory course, advanced vision will be presented.

1. Convolutional Neural Networks:
    - Conv layers;
    - VGG architecture;
    - deep CNNs.
2. Generative Models:
    - Convolutional AutoEncoders as a Generative Model;
    - Generative Networks;
    - Generative Adversarial Networks;
    - Domain Adaptation.
3. Practical session:
    - Domain Adaptation on MNIST.
    - `homework`: GAN on MNIST;

Big Data

1. Introduction to Apache Spark I:
   - intro to Functional Programming;
   - RDD abstraction;
   - distributed computations as a monad.
2. Introduction to Apache Spark II:
   - Spark 'under the hood';
3. Practical session:
   - `homework`: SGD on Apache Spark.
4. Recommendation Systems:
   - Collaborative filtering;
   - Alternating Least Squares.
5. Practical session:
   - `homework`: Alternating Least Squares.

Details

Assesments

Your final marks for the course will be calculated according to the following expression:

$$\text{final mark} \quad = \quad \left\lceil \frac{5}{4} \cdot \text{homework score} + \frac{1}{2} \cdot \text{project score} \right\rceil$$

where:

- homework score - number of accepted homeworks (0-4);
- project score - mark for the final project (0-10).

Homeworks

There will be 4 homeworks, each denoted by word `homework` in the program:

- 1 for the first part;
- 1 for the Deep Learning part;
- 2 for the Big Data part.

Projects

To successfully pass the course you need to make a 'project'. Your 'project' can be either:

- presentation on an advanced topic:
- non-trivial solution for a Machine Learning problem.

Presentation

Presentation on an advanced topic:

- topic must be related to the course;
- it can be presentation of **a technology**
- or **advanced theoretical material**;
- both should contain practical material (exercises etc);
- *you should notify me in advance with some kind of plan for the presentation.*

Examples of presentations:

- TensorFlow + exercises;
- parallelization in XGBoost library;
- your research.

Practical projects

Non-trivial solution for a Machine Learning problem:

- should include analysis;
- explanation of chosen classifiers/architecture/features/...;
- should include proofs (or at least demonstration) for each non-trivial step;
- recommended to be presented as Jupyter Notebook.

Examples:

- solution for a Kaggle competition (fined tuned XGBoost itself is a trivial solution, at least you have to show feature analysis);
- solution for a real-world problem, like content-based recommendation system for cat pictures;
- your research.

An important note

I highly recommend to notify me about your project in advance. It would be a good idea if you send me some kind of abstract or short description. Otherwise, you risk to get low mark e.g. because of projects unrelated to the course, or solved problem being too simple.