Project for the courses Theoretical/Computational foundations of data science

Thomas Bonis & Théo Lacombe

December 12, 2024

1 General rules

A part of the evaluation will be of the form of a project, evaluated through an oral presentation (defense). There are two types of exercise: a "theoretical part" where students are expected to choose a topic and propose an introduction to it, and a "practical part" where one deal with a real-life dataset and attempt at performing a machine learning task.

Students following both courses must cover both parts during their defense; they will be graded evenly.

For students following only "theoretical foundations (...)" (resp. "computational foundations (...)") should only cover the "theoretical" (resp. "practical") part. Naturally, such students are expected to go deeper during their presentation than student following both courses.

Students must gather by group of 3, and therefore students from a same group should follow the same courses (both courses / only theoretical (...) / only computational (...)). At most one group with less than 3 students will be allowed for each profile.

The defenses will be 45 minutes long: 5 minutes to set-up, 30 minutes of presentation, 10 minutes for questions.

2 Theoretical part

2.1 Random Fourier Features

Keywords: Positive definite kernels, spectral analysis

Positive definite kernels are used in many machine learning algorithms (Kernel ridge regression, Gaussian Process Regression, SVM, ...). However, such approaches are usually computationally expensive with a computational cost scaling with the size of the training dataset used. To deal with this issue, [1] proposed to replace classical kernels by random kernels generated randomly. For this subject, we expect (at least):

- A formal presentation of the random fourier features approach.
- A proof of Bochner's Theorem, with the main arguments highlighted during the oral presentation.
- A comparison of computational complexities between this approach and a standard kernel approach for a given kernel-based algorithm.

  In addition, a practical implementation in some cases would be highly appreciated.

2.2 Gradient descent on manifold

Gradient descent on a Riemannian manifold extends classical optimization to spaces with curved geometry. Unlike Euclidean spaces, where straight lines define directions, manifolds rely on geodesics curves that generalize straight lines. In this setting, the algorithm iteratively minimizes a function f by updating along the tangent spaces of the manifold, guided by the Riemannian gradient, which incorporates the geometry through the manifold metric. After computing the gradient in a local tangent space, the update is ”projected” back onto the manifold via a retraction or exponential map.

We expect (at least) a presentation of the following concept:

- Formal definition of gradients in manifold (and related concepts),
- Situations in which such a problem appears.

In addition, a practical implementation in some cases would be highly appreciated.

2.3 Sampling through score-based generative modeling

Keywords: Stochastic differential equations, Functional analysis

Sampling from an arbitrary measure μ on $\mathbb{R}^d$ is a difficult task unless μ happens to be the d-dimensional Gaussian measure. A very recent approach to this problem consists in finding a stochastic differential equation whose solution is, after a large enough time, distributed following μ thus allowing to sample from it.

The students are expected to do the following:

- Explain precisely what score-based generative modeling is about, the relation with stochastic differential equations.
- Implement a basic score-based method to sample from 2D distribution.
- List all steps of the algorithm which would require studying in order to obtain a theoretical understanding of the behavior of the algorithm.

2.4 VC dimension

Keywords: Statistical learning theory

In order to guarantee the performance of a learning algorithm, we need to be able to provide (probabilistic) guarantees on the accuracy of trained model. The concept Vapnik–Chervonenkis dimension is one way to obtain such guarantees.

We expect (at least) a presentation of the following concept:

- Present the conception of Vapnik-Chernovenkis dimension.
- Show how it can be used to obtain guarantees on the performance of a trained model. (While a theoretical proof is expected, it can be complemented through simulations). In particular, main ingredients of the proof should be presented during the oral presentation.

2.5 Spectral clustering

Keywords: Graph Theory, clustering

Spectral Clustering proposes a way to cluster nodes of a graph based on the connectivity (adjacency matrix) of the graph. For this, it leverages the eigenvalues and eigenvectors of a graph Laplacian matrix constructed from pairwise similarities.

We expect (at least) a presentation that includes the following:

- Present the spectral clustering algorithm, detailing key steps like constructing the similarity graph and computing the graph Laplacian.
- Show how this method can be viewed as a relaxation of a graph-cut algorithm. (And why we need such a relaxation)
- Explain how this method can then be used to perform (for instance) k-means in a usual Euclidean space.
- Propose an implementation to solve a simple problem, such as clustering a synthetic dataset

2.6 Lasso Regression

Keywords: Linear regression, optimization

Linear regression in (very) high dimension is prevalent in many fields such as genomics for instance. The problem one faces is that the dimension of the data is usually much higher than the number of data points. One way to circumvent this issue is to introduce a L_1 regularization term in the loss function. The estimator obtained is called a Lasso estimator.

We expect (at least) a presentation that includes the following:

- Present the Lasso regression framework.
- Discuss the optimization issues appearing in this framework.

- Highlight the performance of Lasso estimators through simulations. In particular, show how the influence of the regularization parameter on the bias and on the variance of the obtained estimator.

2.7 Expectation-maximization applied to Gaussian mixture models

Keywords: Density estimation, parametric estimation, clustering

Gaussian Mixture Models (GMM) assume that the data is generated from a mixture of Gaussian distributions, each with its own mean, covariance, and mixing proportion. Assuming that one observes a set $(x_1, \ldots, x_n)$ sampled from a GMM, a natural question is to estimate the parameters (proportions, means and covariances) of this GMM. This is typically done using a two-step algorithm: the Expectation-Maximization (EM) scheme.

We expect (at least) a presentation that includes the following:

- Present formally GMMs, how to sample from them, etc.
- Explain how GMMs can be used to perform clustering,
- Implement the algorithm to fit a GMM to a synthetic dataset.
- Discuss the limitations of the approach.

3 Practice on a real-life dataset

Students must pick one of the following datasets (available on the elearning page of the course) and attempt to solve a supervised learning problem on it.

Each dataset contains a `train_file.csv` and a `test_file.csv`. The subtlety is that the `test_file.csv` does not contain the true labels (but the train file contains labels of course). You must submit on e-learning a file `prediction.csv` that must be a *single column csv file* giving the predictions of their model on the test set. These predictions will be confronted with the corresponding test labels the day of the defense.

Models will be evaluated using either the square root of the MSE or the accuracy, depending on the nature of the problem (therefore, pick the correct loss for your problem!).

Students will be evaluated with respect to the following:

- Presentation of the problem, the dataset, etc.,
- Choice of the model(s), presenting what did work and what didn't,
- Application of the supervised learning methodology,
- Actual performances of their model.

In particular, note that the actual performances of the model—though requested—is not the only criterion for evaluation.

Dataset 1: the Hotel Reservation Dataset. A dataset of hotel booking describing the booking (number of adults, number of children, number of nights, date, is it booked online, etc.) and the eventual status of the reservation (the label): did the customer cancel the reservation?

Dataset 2: Star dataset. A set of astronomic observations representing stellar objects that can be either Star, Galaxy, or Quasar Objects. Observations are given as a set of measurements with respect to various wavelength (visible color r,g,b, but also infrared filter, etc.) and other characteristics (angle of the lens, etc.), its redshift, etc. The goal is to predict the nature of the observation from these features.

Dataset 3: the DVF dataset. DVF is a French dataset recording real estate transactions (during year 2021 for this dataset). Observations describe the flat/house (localisation, surface, etc.), and the target variable is the price at which the transaction was done.

References

[1] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.