Data Science and Scientific Computation (DSSC) Track Core Course

Bernd Bickl, Christoph Lampert, Gašper Tkacik, Chris Wojtan

July 3, 2018

1 Course Overview

The purpose of the DSSC Track Core Course is to introduce students of diverse backgrounds to the basic concepts and skills required for data analysis and numerical simulation. This should enable the students to take other advanced courses in the DSSC track, communicate across disciplinary boundaries, and do their thesis research either in data / simulation intensive fields of natural and life science, or in development of new DSSC methodologies for computer science and statistics.

More specifically, the course has the following goals:

- Provide hands-on experience and scientific insight into different DSSC problems and methodologies;
- Learn about the vocabulary and conceptual framework that different fields developed to construct, analyze, and evaluate models;
- Build a community of computational / data students by project work;
- Practice the following skills: handling data, extracting knowledge from data, creating models, running numerical simulations, identifying and understanding sources of error, working in mixed background teams, written and oral communication.

The course consists of 3 segments, taught by 3 instructors, of approximately 4-5 weeks in duration. Each year, the selection of the particular segments to be taught that year will be announced by the course instructors in advance, making sure to representatively balance the *data analysis* and *numerical simulation* aspects of the track. Typically, this balance will be achieved by combining one segment focused on data analysis, one segment focused on numerical simulation, while the third, usually the last segment, will focus on integrative and interactive project-driven work, and will specifically emphasize visualization and presentation skills. In all segments, the emphasis is on dealing with data and computations in a hands-on fashion. The The curriculum of all available segments is described in Section 4.

2 Grading

Evaluation is based on homeworks and/or written or oral (mini-)project reports at the end of each segment. Segments are weighted equally.

The two segments with data analysis and numerical simulation focus will usually be graded 50% from homeworks and 50% from a mini-project / extended homework at the end of the segment. Typically, these segments will contain around three small homeworks per segment, to ensure that students regularly apply the methodologies learned in class and build up the scripts that they need to solve the mini-projects. At the end of each segment, students present the mini-projects and/or hand in short reports, as agreed with the instructor and the TAs.

The integrative / project-driven segment will be graded based on project idea, execution, participation, and presentations.

3 Prerequisites

This course is intended primarily for students who wish to do thesis-related research in groups that participate in the Data Science and Scientific Computation (DSSC) track; typically, these will be students with computer science, machine learning, statistics, physics, bioinformatics, or applied math backgrounds. Ideally, the students should have the interest in working with real or realistic (simulated) data, and background in the following two areas:

- Undergraduate mathematics: linear algebra, calculus, probabilities.
- Basic procedural programming in a language of your choice, ability to understand C and Python code.

The above prerequisites are detailed in the Appendix. The students should be familiar with the enumerated concepts, even if they don't remember all details at this moment. By the beginning of the course, the students should be able to work with these concepts and answer (most of) the mock exam (http://pub.ist.ac.at/~gtkacik/DSSC/MockExam.pdf) without having to consult an external source (too much).

DSSC core course is not an introductory course in data analysis for life science students. This course has been successfully taught the course to students with non-formal backgrounds if they were very familiar either with the required math but did not have much prior coding experience (e.g., only on the level of the programming service courses given at IST); or to students lacking some of the required math background but fluent in coding. Lacking one of the two prerequisites requires a substantially higher time investment for the student, but is feasible. We strongly advise, however, against taking the course with insufficient background both in math *and* coding.

4 Curriculum by Segments

DSSC Track Core Course will consist of a choice of three segments from the list of four possibilities (A, B, C, D), below. For example, in academic year 2017/2018, DSSC TCC was taught in the following configuration:

Order	Focus	Segment title
1	Numerics / computation	**Segment C:** Simulation and numerics (Bickl)
2	Data analysis	**Segment A:** Working with real data (Tkačik)
3	Integrative / project work	**Segment D:** Data generation, evaluation, presentation (Wojtan)

4.1 Segment A: Working with real data (Tkačik)

Goal: Characterizing basic statistical properties of an unknown dataset.

- Assumptions about the data: IID samples, stationarity, statistical and systematic experimental errors.
- Histograms and histogram statistics.
- Error bar estimation via bootstrap / jackknife methods.
- Quantifying pairwise correlations, principal component analysis (PCA).
- Spectral estimates by Fourier transform, linear filtering and convolution.
- Beyond pairwise statistics: k-means clustering, independent component analysis (ICA).

4.2 Segment B: Predictive models (Lampert)

Goal: Understand and be able to handle predictive models.

- What are predictive models / examples.
- Linear regression and classification (least squares, logistic regression).

- Nonlinear regressio and classification (random forests, deep networks).
- Loss functions, parameter fitting by maximum likelihood.
- Judging models by predictions on held-out data, overfitting, regularization.
- Choosing between models by cross-validation or surrogates.
- Learning and testing large-scale predictive models: tricks-of-the-trade.

4.3 Segment C: Simulation and numerics (Bickel)

Goal: Understand and apply basic computational techniques for simulations in a variety of applied science problems, with focus on differential equations.

- Discretization methods for ordinary and partial differential equations.
- Comparing solvers, explicit and implicit methods.
- Stability, sensitivity, and optimizations.
- Control problems.
- Example advanced applications: chaos, software, predictive capability

4.4 Segment D: Data generation, evaluation, & presentation (Wojtan)

Goal: Build hands-on experience generating data, exploring data, and communicating the results.

The driving theme of this segment is an independent project that requires a custom computational model to generate data, an emphasis on data visualization, and an insightful quantitative data analysis that leads to a natural readjustment or enrichment of the original computational model.

- Insights into iterated computational models (limit behaviors, phase space, bifurcations, instability, etc.)
- Numerical errors, robustness, and conditioning.
- Basic model fitting (if not already covered by an earlier segment).
- Overview of data clustering methods.
- Topological data analysis.
- Scientific visualization.
- Presenting complicated data through speech and writing.

5 Teaching materials

The teaching materials in the form of lecture slides and/or notes will be made available at the beginning of the course.

Appendix: Skills list

Linear Algebra

- vectors in arbitrary dimension
- matrix-matrix/matrix-vector multiplication:
- inner product of vectors, orthogonality
- hyperplanes in Rn (lines in R2), distance of a vector to a line/hyperplane
- norms of vectors, triangular inequality
- orthonormal bases, rotations between bases
- properties of matrices: invertible, symmetric, positive definite, orthogonal
- determinant of a matrix, trace of a matrix
- eigenvalues, eigenvectors
- matrix decompositions (e.g. Cholesky decomposition, singular value decomposition) • matrix exponentiation

Calculus

- derivatives/gradients of functions
- the very basics of differential equations (e.g. finding a solution to dx/dt = ax)
- Fourier transform (not deep theorems, just familiarity with what is an "integral transform" is)
- using derivatives to find the minimum/maximum of a functions
- nice to know: Lagrangian multipliers

Probability

- basic concepts: probability distributions, random variable, sample from a distribution
- probability density function, cumulative density function
- joint distribution, marginal distribution, conditional distribution
- independence/dependence of random variables
- mean, median, mode, standard deviation, covariance, correlation, moments
- rules of probability theory: marginalization, Bayes rule
- some useful distributions: Bernoulli, uniform, (multivariate) Gaussian, Poisson, Exponential, Beta, Gamma

Programming

- being able to read C, Matlab and/or Python programs
- procedural programming in a language of your choice (that can do the things below)
- basic procedural constructs: loops, conditions/branching, subroutines, recursion
- reading and writing text files from disk
- calling numeric libraries if not built-in: matrix operations, FFT
- (pseudo-)random number generation
- making plots from data (Excel is not recommended, but would suffice if nothing else)

Numerics

- approximating a derivative by a finite difference
- approximating an integral by a sum