CS445/545 Fundamentals of Digital Archeology

WHO SHOULD TAKE THIS COURSE

Students who want to build competence in data science and would like to gain a complete set of skills: how to ask a question, discover the relevant digital traces, extract and explore them, and produce a useful tool or insight. Students who can work independently and are highly motivated to learn how to use Python and other tools necessary to solve practical data science problems.

EXPECTED OUTCOMES

Upon completion, students will be able to discover, gather, and analyze digital traces, will learn how to avoid mistakes common in the analysis of low-quality data, and will have produced a working analytics application.

WHY TAKE THIS COURSE?

Great job prospects.

Hal Varian, Chief Economist at Google, said that: "The sexy job in the next ten years will be statisticians. . . The ability to take data — to be able to understand it, to process it, to extract value from it, to visualize it, to communicate it — that's going to be a hugely important skill."

Report on North Carolina State University "The Master of Science in Analytics (MSA)" Class of 2014: "All 75 candidates searching for new employment secured positions prior to graduation. Average starting salaries were in the mid-$90's, with solid gains for candidates without prior work experience. Students had an average of 13 initial job interviews and 3 offers."

OVERVIEW

A great volume of complex data is generated as a result of human activities, including both work and play. To exploit that data for decision making it is necessary to create software that discovers, collects, and integrates the data.

Digital archeology relies on traces that are left over in the course of ordinary activities, for example the logs generated by sensors in mobile phones, the commits in version control systems, or the email sent and the documents edited by a knowledge worker. Understanding such traces is complicated in contrast to data collected using traditional measurement approaches.

Traditional approaches rely on a highly controlled and well-designed measurement system. In meteorology, for example, the temperature is taken in specially designed and carefully selected locations to avoid direct sunlight and to be at a fixed distance from the ground. Such measurement can then be trusted to represent these controlled conditions and the analysis of such data is, consequently, fairly straightforward.

The measurements from geolocation or other sensors in mobile phones are affected by numerous (yet not recorded) factors: was the phone kept in the pocket, was it indoors or outside? The devices are not calibrated or may not work properly, so the corresponding measurements would be inaccurate. Locations (without mobile phones) may not have any measurement, yet may be of the greatest interest. This lack of context and inaccurate or missing data necessitates fundamentally new approaches that rely on patterns of behavior to correct the data, to fill in missing observations, and to elucidate unrecorded context factors. These steps are needed to obtain meaningful results from a subsequent analysis.

The course will cover basic principles and effective practices to increase the integrity of the results obtained from voluminous but highly unreliable sources.

OBJECTIVES

The course will combine theoretical underpinning of big data with intense practice. In particular, approaches to ethical concerns, reproducibility of the results, absence of context, missing data, and incorrect

data will be both discussed and practiced by writing programs to discover the data in the cloud, to retrieve it by scraping the deep web, and by structuring, storing, and sampling it in a way suitable for subsequent decision making. At the end of the course students will be able to discover, collect, and clean digital traces, to use such traces to construct meaningful measures, and to create tools that help with decision making.

In particular, students will acquire the following skills:

Use Python and other tools to discover, retrieve, and process data.

Use data management techniques to store data locally and in the cloud.

Use data analysis methods to explore data and to make predictions.

COURSE DETAILS

Classes are held MWF 10:10-11:00 in MK524

Contact Instructor: Audris Mockus

The following themes will be covered:

Ethics: legal aspects, privacy, confidentiality, governance

Reproducibility: version control, ipython notebook

Fundamentals of big data analysis: extreme distributions, transformations, quantiles, sampling strategies, and logistic regression

The nature of digital traces: lack of context, missing values, and incorrect data

During the course students will be given assignments (mini-projects) that involve a direct application of the lecture material to discovery, collection, cleaning, and analysis of digital traces from, for example, github. In particular, there will be assignments on:

Discovery of data sources in the cloud

Scraping the deep web, e.g., issue trackers, version control, mobile, reputation mining

Efficient storage and analysis

The final project will involve building a functioning system to address a real practical need or to answer an interesting research question that students choose.

COURSE REQUIREMENTS

Students are expected to have basic programming skills, in particular, be able to use regular expressions, programming concepts such as variables, functions, loops, and data structures like lists and dictionaries (for example, COSC 365)

Being familiar with version control systems (e.g., COSC 340), Python (e.g., COSC 370), and introductory level probability (e.g., ECE 313) and statistics, such as, random variables, distributions and regression would be beneficial but is not expected. Everyone is expected, however, to be willing and highly motivated to catch up in the areas where they have gaps in the relevant skills.

Students will be evaluated using the following criteria:

1. Classroom participation (15%): students are expected to read all material covered in a week and come to class prepared to take part in the classroom discussions.

2. Assignments (40%): Each assignment will involve writing (or modifying a template of) a small Python program that accomplishes one of the tasks listed above.

3. Project (45%): one original project done alone or in a group of 2 or 3 students. The project will explore one or more of the themes covered in the course that students find particularly compelling. The group needs to submit a project proposal (2 pages IEEE format) approximately 1.5 months before the end of term. The proposal should provide a brief motivation of the project, detailed discussion of the data that will be obtained or used in the project, along with a time-line of milestones, and expected outcome.