Practical Data Science for Data Scientists

Spotfire Dashboard
Spotfire Dashboard

Proposed GMU Course

Prerequisites
Previous Lectures
Overview of Schedule
Detailed Schedule

1/21 What is Data Science and the Data Science Process?

Purpose
My Resources
Chapter 1 Excerpts
Learning Exercise
Chapter 2 Excerpts
Hands-on Class Exercise
Chapter 2 Excerpts
Team Homework Exercise

1/28 Finding, Cleaning, Analyzing, and Visualizing Data

Purpose
My Resources
Present and Discuss Team Homework Exercise
Chapter 3 Excerpts
Hands-on Class Exercise
Chapter 4 Excerpts
Team Homework Exercises

2/4 Asking and Answering Questions About Data

Purpose
My Resources
Present and Discuss Team Homework Exercises
Chapter 5 Excerpts
Hands-on Class Exercise
Chapter 6 Excerpts
Team Homework Exercise

2/11 Specific Data Science Tools and Applications 1
Purpose
My Resources
Present and Discuss Team Homework Exercises
Chapter 7 Excerpts
Hands-on Class Exercise
Chapter 8 Excerpts
Team Homework Exercise

Chapter 7 Excerpts
Hands-on Class Exercise

Chapter 8 Excerpts
Team Homework Exercise

Chapter 9 Excerpts
Hands-on Class Exercise

Chapter 10 Excerpts
Team Homework Exercise

2/18 Specific Data Science Tools and Applications 2
Purpose
My Resources
Present and Discuss Team Homework Exercises
Chapter 9 Excerpts
Hands-on Class Exercise
Chapter 10 Excerpts
Team Homework Exercise

2/25 Specific Data Science Tools and Applications 3
Purpose
My Resources
Present and Discuss Team Homework Exercises
Chapter 11 Excerpts
Hands-on Class Exercise
Chapter 12 Excerpts
Team Homework Exercise

Chapter 11 Excerpts
Hands-on Class Exercise

Chapter 12 Excerpts
Team Homework Exercise

Chapter 13 Excerpts
Hands-on Class Exercise

Chapter 14 Excerpts
Team Homework Exercise

3/4 Specific Data Science Tools and Applications 4
Purpose
My Resources
Present and Discuss Team Homework Exercises
Chapter 13 Excerpts
Hands-on Class Exercise
Chapter 14 Excerpts
Team Homework Exercise

Chapter 13 Excerpts
Hands-on Class Exercise

Chapter 14 Excerpts
Team Homework Exercise

3/18 Data Science Students and Careers
Purpose
My Resources
Present and Discuss Team Homework Exercises
Chapter 15 Excerpts
Hands-on Class Exercise
Chapter 16 Excerpts
Team Homework Exercise

3/25 Semantic Medline on the YarcData Graph Appliance
4/1 Class Project Proposals Presentations and Feedback
4/8 Discuss March 4-5, 2014, NIST Data Science Symposium
4/15 Class Projects and Final Exam
4/22 Class Projects and Final Exam
4/29 Class Projects and Final Exam
5/7-5/14 Final Exam

Required Textbook
Technology Requirements
Course Description
Course Objectives

Spotfire Dashboard
For Internet Explorer Users and Those Wanting Full Screen Display Use: Web Player Get Spotfire for iPad App

Media, iframe, embed and object tags are not supported inside of a PDF.

Proposed GMU Course
Title: Practical Data Science for Data Scientists
Time: Tuesdays 7:20-10:00 PM
Location: TBA
Prerequisites

To be determined by GMU.

See recent discussion at http://www.meetup.com/Data-Science-DC/events/149203262/?_af_eid=149203262&a=uc1_vm&_af=event

Previous Lectures

- Visualizing Data Patterns with Micromaps, GMU Professor Dan Carr Meeting, March 9, 2011
- Using Data Science Evidence in Public Policy for Big Data and Elections, George Mason University, University Hall, November 1-2, 2012. Slides
- Spotfire Learning Network for Class Projects, George Mason University, University Hall, November 9, 2012.
- County Health Rankings, George Mason University, University Hall, November 16, 2012.

Overview of Schedule

Tue Jan 21: First day of classes
Mon Mar 10 - Sun Mar 16: Spring Break
Mon May 5: Last day of classes
Wed May 7 - Wed May 14: Exam Period (beginning at 7:30 a.m.)

Detailed Schedule

1/21 What is Data Science and the Data Science Process?

Purpose

Discuss Reading: Chapters 1 and 2, see My Profile: Slides, Hands-on Exercise on Exploratory Data Analysis, and a Team Homework Exercise (find the data, store the data, and analyze the data) to present next week.

My Resources

- http://semanticommunity.info/Data_Science
- http://semanticommunity.info/Analytics/Predictive_Analytic_World_Government_2013#Story

Chapter 1 Excerpts

For the first class, the initial thought experiment was: can we use data science to define data science?

Perhaps the most concrete approach is to define data science is by its usage—e.g., what data scientists get paid to do.
An academic data scientist is a scientist, trained in anything from social science to biology, who works with large amounts of data, and must grapple with computational problems posed by the structure, size, messiness, and the complexity and nature of the data, while simultaneously solving a real-world problem.

In industry, a data scientist is someone who knows how to extract meaning from and interpret data, which requires both tools and methods from statistics and machine learning, as well as being human. She spends a lot of time in the process of collecting, cleaning, and munging data, because data is never clean.

Once she gets the data into shape, a crucial part is exploratory data analysis, which combines visualization and data sense.

We’re done with talking about data science; let’s go ahead and do some!

**Learning Exercise**

My Profile: [Slides](#) (Used Data Science for three migrations of my data science content for Data Journalism and Spotfire)

**Chapter 2 Excerpts**

We begin this chapter with a discussion of statistical inference and statistical thinking. Next we explore what we feel every data scientist should do once they’ve gotten data in hand for any data-related project: exploratory data analysis (EDA).

**New kinds of data**

Gone are the days when data is just a bunch of numbers and categorical variables. A strong data scientist needs to be versatile and comfortable with dealing a variety of types of data, including:

- Traditional: numerical, categorical, or binary
- Text: emails, tweets, New York Times articles (see Chapter 4 or Chapter 7)
- Records: user-level data, timestamped event data, json formatted log files (see Chapter 6 or Chapter 8)
- Geo-based location data: briefly touched on in this chapter with NYC housing data
- Network (see Chapter 10)
- Sensor data (not covered in this book)
- Images (not covered in this book)

**Hands-on Class Exercise**

There are 31 datasets named nyt1.csv, nyt2.csv,…,nyt31.csv, which you can find here: [https://github.com/oreillymedia/doing_data_science](https://github.com/oreillymedia/doing_data_science).

NYT Data Set (31 CSV files, 151 MB) My Note: Un-zip [ZIP File](#) and start with NYT1 (CSV).

Each one represents one (simulated) day’s worth of ads shown and clicks recorded on the New York Times home page in May 2012. Each row represents a single user. There are five columns: age, gender (0=female, 1=male), number impressions, number clicks, and loggedin. You’ll be using R or Spotfire to handle these data. R is a programming language designed specifically for data analysis, and it’s pretty intuitive to start using. Spotfire is a leading analytics tool
that is easy to start using without coding, but it also supports the use of R! My Note: I will give extra credit to those who do it both ways in Spotfire!

Hint for doing the rest: don’t read all the datasets into memory. Once you’ve perfected your code for one day, read the datasets in one at a time, process them, output any relevant metrics and variables, and store them in a dataframe; then remove the dataset before reading in the next one. This is to get you thinking about how to handle data sharded across multiple machines. My Note: You are not limited in this way with Spotfire. I will give extra credit to those who import and analyze the entire data set in Spotfire!

Result: See Spotfire Web Player Chapter 2 EDA NYT Clickstream and Spotfire File

Chapter 2 Excerpts

The Data Science Process

Let’s put it all together into what we define as the data science process. The more examples you see of people doing data science, the more you’ll find that they fit into the general framework shown in Figure 2-2. As we go through the book, we’ll revisit stages of this process and examples of it in different ways.

A Data Scientist’s Role in This Process

The data scientist is involved in every part of this process

Connection to the Scientific Method

We can think of the data science process as an extension of or variation of the scientific method:

- Ask a question.
- Do background research.
- Construct a hypothesis.
- Test your hypothesis by doing an experiment.
- Analyze your data and draw a conclusion.
- Communicate your results.

In both the data science process and the scientific method, not every problem requires one to go through all the steps, but almost all problems can be solved with some combination of the stages. For example, if your end goal is a data visualization (which itself could be thought of as a data product), it’s possible you might not do any machine learning or statistical modeling, but you’d want to get all the way to a clean dataset, do some exploratory analysis, and then create the visualization.

Team Homework Exercise

Case Study: RealDirect

Go to the NYC (Manhattan) Housing dataset: How to Find Recent Sales Data for New York City Real Estate and Click on: Rolling Sales Update (after the fifth paragraph). You can use any or all of the datasets here—start with Manhattan August, 2012–August 2013.
My Note: Now November 2012-November 2013:
rollingsales_bronx, rollingsales_brooklyn, rollingsales_manhattan, rollingsales_queens,
and rollingsales_statenisland

Form Teams, Ask Me Questions, and Prepare to Present Next Week

Result: See Spotfire Web Player Chapter 2 NYC (Manhattan) Housing and Spotfire File

1/28 Finding, Cleaning, Analyzing, and Visualizing Data

Purpose

Discuss Reading: Chapters 3 and 4, Present and Discuss Team Homework Exercise, Hands-on Class Exercise, and Team Homework Exercise.

My Resources

Spotfire Cloud Library

Present and Discuss Team Homework Exercise

Case Study: RealDirect: Answer the questions: Where did we find the data?, Where did we store the data?, and What did we find when we analyzed the data? What is our data story and product?

Chapter 3 Excerpts

In the previous chapter we discussed in general how models are used in data science. In this chapter, we’re going to be diving into algorithms.

An algorithm is a procedure or set of steps or rules to accomplish a task. Algorithms are one of the fundamental concepts in, or building blocks of, computer science: the basis of the design of elegant and efficient code, data preparation and processing, and software engineering.

Machine learning algorithms are largely used to predict, classify, or cluster. Three Basic Algorithms: linear regression, k-nearest neighbors (k-NN), and k-means.

The rest is details for each

Summing It All Up

We’ve now introduced you to three algorithms that are the basis for the solutions to many real-world problems. If you understand these three, you’re already in good shape. If you don’t, don’t worry, it takes a while to sink in.

Thought Experiment: Automated Statistician

Rachel attended a workshop in Big Data Mining at Imperial College London in May 2013. One of the speakers, Professor Zoubin Ghahramani from Cambridge University, said that one of his long-term research projects was to build...
an “automated statistician.” What do you think that means? What do you think would go into building one?

Does the idea scare you? Should it?

**Hands-on Class Exercise**

*Exercise: Basic Machine Learning Algorithms:* Continue with the NYC (Manhattan) Housing dataset you worked with in the preceding chapter: [How to Find Recent Sales Data for New York City Real Estate](#) and [Rolling Sales Data](#).

*My Note: Now November 2012-November 2013:*
- [rollingsales_bronx](#)
- [rollingsales_brooklyn](#)
- [rollingsales_manhattan](#)
- [rollingsales_queens](#)
- [rollingsales_statenisland](#)

See Spotfire User's Guide for Data, Insert Rows, to merge these five data sets so they are 90,328 rows. This can be done for the NYT 31 CSV files as well!

**Chapter 4 Excerpts**

Let’s start by looking at a bunch of email text, whose rows seem to contain the subject and first line of an email in an inbox. You may notice that several of the rows of text look like spam. How did you figure this out? Can you write code to automate the spam filter that your brain represents?

In this chapter, we’ll use Naive Bayes to solve this problem. So are we at a loss now that two methods we’re familiar with, linear regression and k-NN, won’t work for the spam filter problem? No! Naive Bayes is another classification method at our disposal that scales well and has nice intuitive appeal.

**State of the Art for Spam Filters**

In the last five years, people have started using stochastic gradient methods to avoid the noninvertible (overfitting) matrix problem. Switching to logistic regression with stochastic gradient methods helped a lot, and can account for correlations between words. Even so, Naive Bayes is pretty impressively good considering how simple it is.

**Scraping the Web: APIs and Other Tools**

As a data scientist, you’re not always just handed some data and asked to go figure something out based on it. Often, you have to actually figure out how to go get some data you need to ask a question, solve a problem, do some research, etc. One way you can do this is with an API. For the sake of this discussion, an API (application programming interface) is something websites provide to developers so they can download data from the website easily and in standard format.

**Historical Context: Natural Language Processing**

The example in this chapter where the raw data is text is just the tip of the iceberg of a whole field of research in computer science called natural language processing (NLP). The types of problems that can be solved with NLP include machine translation, where given text in one language, the algorithm can translate the text to another language; semantic analysis; part of speech tagging; and document classification (of which spam filtering is an example). Research in these areas dates back to the 1950s.
Team Homework Exercises

Select One, But Please Present Both:

- Jake’s Exercise: Naive Bayes for Article Classification: NYT Data Set (31 CSV files, 151 MB) Already Used
- A Spam Filter for Individual Words: To do this yourself, go online and download Enron emails

- The cleansed EDRM Enron data set is 18GB in total. The data set has been divided into many smaller files for ease of download. The files are available for download here. 131 ZIP files.
- Also see: Mashing Up Structured Data and Unstructured Content: Use free text search to explore and analyze the unstructured content in Enron emails. I downloaded the Spotfire file but could not get it to open

Form Teams (Same or New), Ask Me Questions, and Prepare to Present Next Week

2/4 Asking and Answering Questions About Data

Purpose

Discuss Reading: Chapters 5 and 6, Present and Discuss Team Homework Exercise, Hands-on Class Exercise, and Team Homework Exercise.

My Resources

http://semanticommunity.info/#AOL_Government_Stories

Present and Discuss Team Homework Exercises

Select One, But Please Present Both:

- Jake’s Exercise: Naive Bayes for Article Classification: NYT Data Set (31 CSV files, 151 MB) Already Used
- A Spam Filter for Individual Words: To do this yourself, go online and download Enron emails

Chapter 5 Excerpts

In this chapter, we’re talking about logistic regression, but there’s other classification algorithms available, including decision trees (which we’ll cover in Chapter 7), random forests (Chapter 7), and support vector machines and neural networks (which we aren’t covering in this book).

The big picture is that given data, a real-world classification problem, and constraints, you need to determine:

1. Which classifier to use
2. Which optimization method to employ
3. Which loss function to minimize
4. Which features to take from the data
5. Which evaluation metric to use

Never underestimate the power of creativity—people often have a vision they can describe but no method as to how to get there. As the data scientist, you have to turn that vision into a mathematical model within certain operational...
You need to state a well-defined problem, argue for a metric, and optimize for it as well. You also have to make sure you’ve actually answered the original question.

There is art in data science—it’s in translating human problems into the mathematical context of data science and back.

But we always have more than one way of doing this translation—more than one possible model, more than one associated metric, and possibly more than one optimization. So the science in data science is—given raw data, constraints, and a problem statement—how to navigate through that maze and make the best choices. Every design choice you make can be formulated as an hypothesis, against which you will use rigorous testing and experimentation to either validate or refute.

This process, whereby one formulates a well-defined hypothesis and then tests it, might rise to the level of a science in certain cases. Specifically, the scientific method is adopted in data science as follows:

- You hold on to your existing best performer.
- Once you have a new idea to prototype, set up an experiment wherein the two best models compete.
- Rinse and repeat (while not overfitting).

Classifiers

This section focuses on the process of choosing a classifier. Classification involves mapping your data points into a finite set of labels or the probability of a given label or labels. We’ve already seen some examples of classification algorithms, such as Naive Bayes and k-nearest neighbors (k-NN), in the previous chapters. Table 5-1 shows a few examples of when you’d want to use classification:

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Will someone click on this ad?</td>
<td>0 or 1</td>
</tr>
<tr>
<td>What number is this (image recognition)?</td>
<td>0, 1, 2, etc.</td>
</tr>
<tr>
<td>What is this news article about?</td>
<td>“Sports”</td>
</tr>
<tr>
<td>Is this spam?</td>
<td>0 or 1</td>
</tr>
<tr>
<td>Is this pill good for headaches?</td>
<td>0 or 1</td>
</tr>
</tbody>
</table>

From now on we’ll talk about binary classification only (0 or 1).

M6D Logistic Regression Case Study

The team have three core problems as data scientists at M6D:

1. Feature engineering: Figuring out which features to use and how to use them.
2. User-level conversion prediction: Forecasting when someone will click.
3. Bidding: How much it is worth to show a given ad to a given user?

This case study focuses on the second problem. M6D uses logistic regression for this problem because it’s highly scalable and works great for binary outcomes like clicks.

Click Models
At M6D, they need to match clients, which represent advertising companies, to individual users. Generally speaking, the advertising companies want to target ads to users based on a user’s likelihood to click. Let’s discuss what kind of data they have available first, and then how you’d build a model using that data.

M6D keeps track of the websites users have visited, but the data scientists don’t look at the contents of the page. Instead they take the associated URL and hash it into some random string. They thus accumulate information about users, which they stash in a vector.

*Lift*
How much more people are buying or clicking because of a model (once we’ve introduced it into production).

**Hands-on Class Exercise**

[Media 6 Degrees Exercise](#)

Media 6 Degrees kindly provided a dataset that is perfect for exploring logistic regression models, and evaluating how good the models are. [dds_ch5_binary-class-dataset](#)

[See Spotfire Web Player Chapter 5 Logistic Regression Media 6 Degrees and Spotfire File](#)

**Chapter 6 Excerpts**

The main topics for this chapter—times series, financial modeling, and fancypants regression, and building a GetGlue-like recommendation system to address the problem of content discovery within the movie and TV space.

**Timestamps**

Timestamped event data is a common data type in the age of Big Data. In fact, it’s one of the causes of Big Data. The fact that computers can record all the actions a user takes means that a single user can generate thousands of data points alone in a day. When people visit a website or use an app, or interact with computers and phones, their actions can be logged, and the exact time and nature of their interaction recorded. When a new product or feature is built, engineers working on it write code to capture the events that occur as people navigate and use the product—that capturing is part of the product.

For example, imagine a user visits the New York Times home page. The website captures which news stories are rendered for that user, and which stories were clicked on. This generates event logs. Each record is an event that took place between a user and the app or website.

**Exploratory Data Analysis (EDA)**

As we described in Chapter 2, it’s best to start your analysis with EDA so you can gain intuition for the data before building models with it. Let’s delve deep into an example of EDA you can do with user data, stream-of-consciousness style. This is an illustration of a larger technique, and things we do here can be modified to other types of data, but you also might need to do something else entirely depending on circumstances.
The very first thing you should look into when dealing with user data is individual user plots over time. Make sure the data makes sense to you by investigating the narrative the data indicates from the perspective of one person.

Metrics and New Variables or Features

The intuition we gained from EDA can now help us construct metrics. For example, we can measure users by keeping tabs on the frequencies or counts of their actions, the time to first event, or simple binary variables such as “did action at least once,” and so on.

What’s Next?

We want to start moving toward modeling, algorithms, and analysis, incorporating the intuition we built from the EDA into our models and algorithms.

Historical Perspective: What’s New About This?

Timestamped data itself is not new, and time series analysis is a well-established field. There are a couple things that make this new, or at least the scale of it new. First, it’s now easy to measure human behavior throughout the day because many of us now carry around devices that can be and are used for measurement purposes and to record actions. Next, timestamps are accurate, so we’re not relying on the user to self-report, which is famously unreliable. Finally, computing power makes it possible to store large amounts of data and process it fairly quickly.

Financial Modeling

Before the term data scientist existed, there were quants working in finance. There are many overlapping aspects of the job of the quant and the job of the data scientist, and of course some that are very different. For example, as we will see in this chapter, quants are singularly obsessed with timestamps, and don’t care much about why things work, just if they do.

Preparing Financial Data

We often “prepare” the data before putting it into a model. Typically the way we prepare it has to do with the mean or the variance of the data, or sometimes the data after some transformation like the log (and then the mean or the variance of that transformed data). This ends up being a submodel of our model.

GetGlue

GetGlue is a New York-based startup whose primary goal is to address the problem of content discovery within the movie and TV space. The usual model for finding out what’s on TV is the 1950’s TV Guide schedule; that’s still how many of us find things to watch. Given that there are thousands of channels, it’s getting increasingly difficult to find out what’s good on TV.

GetGlue wants to change this model, by giving people personalized TV recommendations and personalized guides. Specifically, users “check in” to TV shows, which means they can tell other people they’re watching a show, thereby creating a timestamped data point. They can also perform other actions such as liking or commenting on the show.

We store information in triplets of data of the form {user, action, item}, where the item is a TV show (or a movie). One way to visualize this stored data is by drawing a bipartite graph as shown in Figure 6-1.
We’ll go into graphs in later chapters, but for now you should know that the dots are called “nodes” and the lines are called “edges.” This specific kind of graph, called a bipartite graph, is characterized by there being two kinds of nodes, in this case corresponding to “users” and “items.” All the edges go between a user and an item, specifically if the user in question has acted in some way on the show in question. There are never edges between different users or different shows. The graph in Figure 6-1 might display when certain users have “liked” certain TV shows.

GetGlue enhances the graph and also hires human evaluators to make connections or directional edges between shows.

We can think of a raw data point from GetGlue as being in the order of \{user, verb, object, timestamp\}.

**Team Homework Exercise**

**Exercise: GetGlue and Timestamped Event Data**

GetGlue kindly provided a dataset for us to explore their data, which contains timestamped events of users checking in and rating TV shows and movies.

The raw data (CAUTION 1.4GB TAR ZIP) is on a per-user basis, from 2007 to 2012, and only shows ratings and check-ins for TV shows and movies. It’s less than 3% of their total data, and even so it’s big, namely 11 GB once it’s uncompressed.

Downloaded and used: [http://www.7-zip.org/](http://www.7-zip.org/), which gave an 11.1 GB TAR file. But file would not import to Spotfire so posted the question to TIBCO Community: [https://tibbr.tibcommunity.com/tibbr/#!/users/15130](https://tibbr.tibcommunity.com/tibbr/#!/users/15130)

Get the Data: [Go to Yahoo! Finance](http://finance.yahoo.com/q/hp?s=%5EO...torical+Prices) (CSV) See Spotfire Web Player and File

Form Teams (Same or New), Ask Me Questions, and Prepare to Present Next Week

**2/11 Specific Data Science Tools and Applications 1**

**Purpose**

Discuss Reading: Chapters 7 and 8, Present and Discuss Team Homework Exercises, Hands-on Class Exercise, and Team Homework Exercise.

**My Resources**

[http://semanticommunity.info/Data_Science/Free_Data_Visualization_and_Analysis_Tools](http://semanticommunity.info/Data_Science/Free_Data_Visualization_and_Analysis_Tools)

[http://semanticommunity.info/Data_Science/KDD_Cup](http://semanticommunity.info/Data_Science/KDD_Cup)

**Present and Discuss Team Homework Exercises**

Get the Data: [Go to Yahoo! Finance](http://finance.yahoo.com/q/hp?s=%5EO%20torical+Prices) and download daily data from a stock that has at least eight years of data, making sure it goes from earlier to later. If you don’t know how to do it, Google it. Yahoo: [http://finance.yahoo.com/q/hp?s=%5EO%20torical+Prices](http://finance.yahoo.com/q/hp?s=%5EO%20torical+Prices) (CSV) See Spotfire Web Player and File

---

**Chapter 7 Excerpts**

How do companies extract meaning from the data they have? In this chapter we hear from two people with very different approaches to that question—namely, William Cukierski from Kaggle and David Huffaker from Google.

While William Cukierski was working on writing his dissertation, he got more and more involved in Kaggle competitions (more about Kaggle in a bit), finishing very near the top in multiple competitions, and now works for Kaggle.

After giving us some background in data science competitions and crowdsourcing, Will will explain how his company works for the participants in the platform as well as for the larger community.

Will will then focus on **feature extraction** and **feature selection**. Quickly, feature extraction refers to taking the raw dump of data you have and curating it more carefully, to avoid the “garbage in, garbage out” scenario you get if you just feed raw data into an algorithm without enough forethought. Feature selection is the process of constructing a subset of the data or functions of the data to be the predictors or variables for your models and algorithms.

**Data Science Competitions Cut Out All the Messy Stuff**

Competitions might be seen as formulaic, dry, and synthetic compared to what you’ve encountered in normal life. Competitions cut out the messy stuff before you start building models—asking good questions, collecting and cleaning the data, etc.—as well as what happens once you have your model, including visualization and communication. The team of Kaggle data scientists actually spends a lot of time creating the dataset and evaluation metrics, and figuring out what questions to ask, so the question is: while they’re doing data science, are the contestants?

Being a data scientist is when you learn more and more about more and more, until you know nothing about everything. — Will Cukierski

Kaggle is a company whose tagline is, “We’re making data science a sport.” Kaggle forms relationships with companies and with data scientists. For a fee, Kaggle hosts competitions for businesses that essentially want to crowdsource (or leverage the wider data science community) to solve their data problems. Kaggle provides the infrastructure and attracts the data science talent.

In Kaggle competitions, you are given a training set, and also a test set where the ys are hidden, but the xs are given, so you just use your model to get your predicted xs for the test set and upload them into the Kaggle system to see your evaluation score. This way you don’t share your actual code with Kaggle unless you win the prize (and Kaggle doesn’t have to worry about which version of Python you’re running). Note that even giving out just the xs is real information—in particular it tells you, for example, what sizes of xs your algorithm should optimize for. Also for the purposes of the competition, there is a third hold-out set that contestants never have access to. You don’t see the xs or the ys—that is used to determine the competition winner when the competition closes.

One reason you don’t want competitions lasting too long is that, after a while, the only way to inch up performance is to make things ridiculously complicated. For example, the original Netflix Prize lasted two years and the final winning model was too complicated for them to actually put into production.
The idea of feature selection is identifying the subset of data or transformed data that you want to put into your model. Different branches of academia use different terms to describe the same thing. Statisticians say “explanatory variables” or “dependent variables” or “predictors” when they’re describing the subset of data that is the input to a model. Computer scientists say “features.”

To improve the performance of your predictive models, you want to improve your feature selection process.

This process we just went through of brainstorming a list of features for Chasing Dragons is the process of feature generation or feature extraction. This process is as much of an art as a science. It's good to have a domain expert around for this process, but it's also good to use your imagination.

My Note: See my Knowledge Base examples in spreadsheets.

David Huffaker: Google’s Hybrid Approach to Social Research

David’s focus is on the effective marriages of both qualitative and quantitative research, and of big and little data. Large amounts of big quantitative data can be more effectively extracted if you take the time to think on a small scale carefully first, and then leverage what you learned on the small scale to the larger scale. And vice versa, you might find patterns in the large dataset that you want to investigate by digging in deeper by doing intensive usability studies with a handful of people, to add more color to the phenomenon you are seeing, or verify interpretations by connecting your exploratory data analysis on the large dataset with relevant academic literature.

Google does a good job of putting people together. They blur the lines between research and development. They even wrote about it in this July 2012 position paper: Google’s Hybrid Approach to Research. Their researchers are embedded on product teams. The work is iterative, and the engineers on the team strive to have near-production code from day 1 of a project.

David suggested that as data scientists, we consider how to move into an experimental design so as to move to a causal claim between variables rather than a descriptive relationship. In other words, our goal is to move from the descriptive to the predictive.

As data scientists, it can be helpful to think of different structures and representations of data, and once you start thinking in terms of networks, you can see them everywhere.

Other examples of networks:

- The nodes are users in Second Life, and the edges correspond to interactions between users. Note there is more than one possible way for players to interact in this game, leading to potentially different kinds of edges.
- The nodes are websites, the (directed) edges are links.
- Nodes are theorems, directed edges are dependencies.

David left us with these words of wisdom: as you move forward and have access to Big Data, you really should complement them with qualitative approaches. Use mixed methods to come to a better understanding of what’s going on. Qualitative surveys can really help.
Hands-on Class Exercise
SAS and Sas Public Data Sets

See:

SAS-Spotfire Web Player and Spotfire File,

SAS exercises-Spotfire Web Player and Spotfire File, and

SAS Public Data Sets-Spotfire Web Player and Spotfire File

Exercise: Build Your Own Recommendation System

Chapter 8 Excerpts

My Note: This is the most difficult chapter in the book for me to teach since I do not understand the Python code at the end and have never built a Recommendation Engine myself. I would welcome some help here.

Recommendation engines, also called recommendation systems, are the quintessential data product and are a good starting point when you're explaining to non-data scientists what you do or what data science really is. This is because many people have interacted with recommendation systems when they've been suggested books on Amazon.com or gotten recommended movies on Netflix. Beyond that, however, they likely have not thought much about the engineering and algorithms underlying those recommendations, nor the fact that their behavior when they buy a book or rate a movie is generating data that then feeds back into the recommendation engine and leads to (hopefully) improved recommendations for themselves and other people.

In this chapter, Matt Gattis walks us through what it took for him to build a recommendation system for Hunch.com—including why he made certain decisions, and how he thought about trade-offs between various algorithms when building a large-scale engineering system and infrastructure that powers a user-facing product.

Hunch is a website that gives you recommendations of any kind.

At first, they focused on trying to make the questions as fun as possible. Then, of course, they saw things needing to be asked that would be extremely informative as well, so they added those. Then they found that they could ask merely 20 questions and then predict the rest of them with 80% accuracy.

Eventually Hunch expanded into more of an API model where they crawl the Web for data rather than asking people direct questions. The service can also be used by third parties to personalize content for a given site—a nice business proposition that led to eBay acquiring Hunch.com.

A Real-World Recommendation Engine

We're going to show you how to do one relatively simple but compete version in this chapter.
To set up a recommendation engine, suppose you have users, which form a set $U$; and you have items to recommend, which form a set $V$. As Kyle Teague told us in Chapter 6, you can denote this as a bipartite graph (shown again in Figure 8-1) if each user and each item has a node to represent it—there are lines from a user to an item if that user has expressed an opinion about that item. Note they might not always love that item, so the edges could have weights: they could be positive, negative, or on a continuous scale (or discontinuous, but many-valued like a star system).

Next up, you have training data in the form of some preferences—you know some of the opinions of some of the users on some of the items. From those training data, you want to predict other preferences for your users. That’s essentially the output for a recommendation engine. You may also have metadata on users (i.e., they are male or female, etc.) or on items (the color of the product). For example, users come to your website and set up accounts, so you may know each user’s gender, age, and preferences for up to three items.

Beyond Nearest Neighbor: Machine Learning Classification

We’ll first walk through a simplification of the actual machine learning algorithm for this—namely we’ll build a separate linear regression model for each item. With each model, we could then predict for a given user, knowing their attributes, whether they would like the item corresponding to that model. So one model might be for predicting whether you like *Mad Men* and another model might be for predicting whether you would like Bob Dylan.

The good news: You know how to estimate the coefficients by linear algebra, optimization, and statistical inference: specifically, linear regression.

The bad news: This model only works for one item, and to be complete, you’d need to build as many models as you have items. Moreover, you’re not using other items’ information at all to create the model for a given item, so you’re not leveraging other pieces of information.

But wait, there’s more good news: This solves the “weighting of the features” problem we discussed earlier, because linear regression coefficients are weights.

Crap, more bad news: over-fitting is still a problem, and it comes in the form of having huge coefficients when you don’t have enough data (i.e., not enough opinions on given items).

The Dimensionality Problem

OK, so we’ve tackled the over-fitting problem, so now let’s think about over-dimensionality—i.e., the idea that you might have tens of thousands of items. We typically use both Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) to tackle this, and we’ll show you how shortly.

To understand how this works before we dive into the math, let’s think about how we reduce dimensions and create “latent features” internally every day. For example, people invent concepts like “coolness,” but we can’t directly measure how cool someone is. Other people exhibit different patterns of behavior, which we internally map or reduce to our one dimension of “coolness.” So coolness is an example of a latent feature in that it’s unobserved and not measurable directly, and we could think of it as reducing dimensions because perhaps it’s a combination of many “features” we’ve observed about the person and implicitly weighted in our mind.

Two things are happening here: the dimensionality is reduced into a single feature and the latent aspect of that feature.

Time to Brush Up on Your Linear Algebra if You Haven’t Already

http://semanticommunity.info/Data_Science/Practical_Data_Science_for_Data_Scientists

Updated: Wed, 23 Sep 2015 02:36:14 GMT

Powered by mindtouch
A lot of the rest of this chapter likely won’t make sense (and we want it to make sense to you!) if you don’t know linear algebra and understand the terminology and geometric interpretation of words like rank (hint: the linear algebra definition of that word has nothing to do with ranking algorithms), orthogonal, transpose, base, span, and matrix decomposition. Thinking about data in matrices as points in space, and what it would mean to transform that space or take subspaces can give you insights into your models, why they’re breaking, or how to make your code more efficient. This isn’t just a mathematical exercise for the sake of it—although there is elegance and beauty in it—it can be the difference between a star-up that fails and a startup that gets acquired by eBay. We recommend Khan Academy’s excellent free online introduction to linear algebra if you need to brush up your linear algebra skills.

Principal Component Analysis (PCA)

Let’s look at another approach for predicting preferences. With this approach, you’re still looking for U and V as before, but you don’t need S anymore.

As with any machine learning model, you should perform cross-validation for this model—leave out a bit and see how you did, which we’ve discussed throughout the book. This is a way of testing over-fitting problems.

Exercise: Build Your Own Recommendation System

In Chapter 6, we did some exploratory data analysis on the GetGlue dataset. Now’s your opportunity to build a recommendation system with that dataset.

My Note: You have to understand Python Code to understand this!

Team Homework Exercise

Read in next week’s reading: Data Visualization for the Rest of Us: See my Slides and Web Player. Start to create your own Hubway Data Visualization Challenge and eventually submit it for your class project and the challenge (now closed but still accepting submissions) if you want.

2/18 Specific Data Science Tools and Applications 2

Purpose

Discuss Reading: Chapters 9 and 10, Present and Discuss Team Homework Exercises, Hands-on Class Exercise, and Team Homework Exercise.

My Resources

http://semanticommunity.info/Data_Science/Free_Data_Visualization_and_Analysis_Tools
Present and Discuss Team Homework Exercises

Read in next week's reading: Data Visualization for the Rest of Us: See my Slides and Web Player. Start to create your own Hubway Data Visualization Challenge and eventually submit it for your class project and the challenge (now closed but still accepting submissions) if you want.

Chapter 9 Excerpts

There are two contributors for this chapter, Mark Hansen, a professor at Columbia University, and Ian Wong, an inference scientist at Square.

Mark will walk us through a series of influences and provide historical context for his data visualization projects, which he will tell us more about at the end. Mark’s projects are genuine works of art—installations appearing in museums and public spaces.

He’s been doing data visualization since before data visualization was cool, or to put it another way, we consider him to be one of the fathers of data visualization.

Mark started by telling us a bit about Gabriel Tarde, who was a sociologist who believed that the social sciences had the capacity to produce vastly more data than the physical sciences. According to Tarde, we should instead be tracking every cell.

In the social realm we can do the analog of this, if we replace cells with people. We can collect a huge amount of information about individuals, especially if they offer it up themselves through Facebook.

In 1903, Tarde even foresaw the emergence of Facebook, as a sort of “daily press”.

Mark (and Tarde) want us to newly consider both the way the structure of society changes as we observe it, and ways of thinking about the relationship of the individual to the aggregate.

In other words, the past nature of data collection methods forced one to consider aggregate statistics that one can reasonably estimate by subsample—means, for example. But now that one can actually get one’s hands on all data and work with all data, one no longer should focus only on the kinds of statistics that make sense in the aggregate, but also one’s own individually designed statistics—say, coming from graph-like interactions—that are now possible due to finer control. Don’t let the dogma that resulted from past restrictions guide your thinking when those restrictions no longer hold.

Mark started the conversation with this undated quote from our own John Tukey:

The best thing about being a statistician is that you get to play in everyone’s backyard.
— John Tukey

In Mark’s opinion, “everything else” should include fields from social science and physical science to education, design, journalism, and media art.
Bottom-line: it’s possible that the language of data science has *something* to do with social science just as it has *something* to do with math. You might not be surprised to hear that, when Mark told us about his profile as a data scientist, the term he coined was “expansionist.”

In a language for artists, you’d want to be able to specify shapes, to faithfully render whatever visual thing you had in mind, to sketch, possibly in 3D, to animate, to interact, and most importantly, to *publish*.

In other words, we don’t just go into their backyards and play; maybe instead we go in and watch *them* play, and then formalize and inform their process with our own bells and whistles. In this way they can teach us new games, games that actually expand our fundamental conceptions of data and the approaches we need to analyze them.

Here are some of Mark’s favorite visualization projects, and for each one he asks us: is this your idea of data visualization? What’s data?

Here are some of Mark’s favorite visualization projects, and for each one he asks us: is this your idea of data visualization? What’s data?

Data Science and Risk

Ian Wong, who came to tell us about doing data science on the topic of risk. Ian is an inference scientist at Square. The mission of the company is to make commerce easy for everyone. The question they set out to answer is “how do we make transactions simple and easy?”

Square wants to make it easy for sellers to sign up for their service and to accept payments. Of course, it’s also possible that somebody may sign up and try to abuse the service. They are, therefore, very careful at Square to avoid losing money on sellers with fraudulent intentions or bad business models.

But how does Square detect bad behavior efficiently? Ian explained that they do this by investing in machine learning with a healthy dose of visualization.

Let’s start by asking: what’s suspicious? If we see lots of micro transactions occurring, say, or if we see a sudden, high frequency of transactions, or an inconsistent frequency of transactions, that might raise our eyebrows.

This example crystallizes the important challenges Square faces: false positives erode customer trust, false negatives cost Square money.

To be clear, there are actually two kinds of fraud to worry about: sellerside fraud and buyer-side fraud. For the purpose of this discussion, we’ll focus on seller-side fraud.

Because Square processes millions of dollars worth of sales per day, they need to gauge the plausibility of charges systematically and automatically. They need to assess the risk level of every event and entity in the system.

Here’s the process shown in Figure 9-16: given a bunch (as in millions) of payment events and their associated date (as shown in the data schema earlier), they throw each through the risk models, and then send some iffy-looking ones on to a “manual review.” An ops team will then review the cases on an individual basis. Specifically, anything that looks rejectable gets sent to ops, who follow up with the merchants.
You can think of the model as a function from payments to labels (e.g., good or bad). Putting it that way, it kind of sounds like a straightforward supervised learning problem. And although this problem shares some properties with that, it’s certainly not that simple.

Technically we would call this a **semi-supervised learning problem**, straddling the worlds of supervised and unsupervised learning.

Now that we’ve set the stage for the problem, Ian moved on to describing the supervised learning recipe as typically taught in school:

- Get data.
- Derive features.
- Train model.
- Estimate performance.
- Publish model!

But transferring this recipe to the real-world setting is not so simple. In fact, it’s not even clear that the order is correct. Ian advocates thinking about the **objective** first and foremost, which means bringing performance estimation to the top of the list.

**Data Visualization at Square**

- Next Ian talked about ways in which the Risk team at Square uses visualization:
  - Enable efficient transaction review.
  - Reveal patterns for individual customers and across customer segments.
  - Measure business health.
  - Provide ambient analytics.

And as parting advice, Ian encourages you to:

- Play with real data.
- Build math, stats, and computer science foundations in school.
- Get an internship.
- Be literate, not just in statistics.
- Stay curious!

**Data Visualization for the Rest of Us**

Not all of us can create data visualizations considered to be works of art worthy of museums, but it’s worth building up one’s ability to use data visualization to communicate and tell stories, and convey the meaning embedded in the data. Just as data science is more than a set of tools, so is data visualization, but in order to become a master, one must first be able to master the technique. Following are some tutorials and books that we have found useful in building up our data visualization skills. See [List](http://semanticommunity.info/Data_Science/Practical_Data_Science_for_Data_Scientists) and Collaborate with an artist or graphic designer!

**Data Visualization Exercise**
More advanced students in the class were given the option to participate in the Hubway Data Visualization challenge. Hubway is Boston’s bike-sharing program, and they released a dataset and held a competition to visualize it. The dataset is still available, so why not give it a try? My Note: I did!

**Hands-on Class Exercise**

Data Visualization for the Rest of Us: See my Slides and Spotfire Web Player and Spotfire File. Start to create your own Hubway Data Visualization Challenge and eventually submit it for your class project and the challenge (now closed but still accepting submissions) if you want.

Social Networks: NodeXL CodePlex and Microsoft NodeXL

**Chapter 10 Excerpts**

In this chapter we’ll explore two topics that have started to become especially hot over the past 5 to 10 years: social networks and data journalism. However, with the emergence of online social networks such as Facebook, LinkedIn, Twitter, and Google+, we now have a new rich source of data, which opens many research problems both from a social science and quantitative/technical point of view.

We’ll hear first about how one company, Morningside Analytics, visualizes and finds meaning in social network data, as well as some of the underlying theory of social networks. From there, we look at constructing stories that can be told from social network data, which is a form of data journalism. At the heart of both (data science or data journalism) is the ability to ask good questions, to answer them with data, and to communicate one’s findings.

John Kelly from Morningside Analytics wants to understand how people come together, and when they do, what their impact is on politics and public policy. His company, Morningside Analytics, has clients like think tanks and political organizations. They typically want to know how social media affects and creates politics.

Case-Attribute Data versus Social Network Data

Kelly doesn’t model data in the standard way through case-attribute data. Case-attribute refers to how you normally see people feed models with various “cases,” which can refer to people or events—each of which have various “attributes,” which can refer to age, or operating system, or search histories.

Kelly points out that there’s been a huge bias toward modeling with case-attribute data. One explanation for this bias is that it’s easy to store case-attribute data in databases, or because it’s easy to collect this kind of data. In any case, Kelly thinks it’s missing the point of the many of the questions we are trying to answer.

In some sense, the current focus on case-attribute data is a problem of looking for something “under the streetlamp”—a kind of observational bias wherein people are used to doing things a certain (often easier) way so they keep doing it that way, even when it doesn’t answer the questions they care about.

Kelly claims that the world is a network much more than it’s a bunch of cases with attributes.

Social Network Analysis

http://semanticommunity.info/Data_Science/Practical_Data_Science_for_Data_Scientists
Updated: Wed, 23 Sep 2015 02:36:14 GMT
Powered by Mindtouch
Social network analysis comes from two places: graph theory, where Euler solved the Seven Bridges of Konigsberg problem, and sociometry. For the most part, we need to know enough about the actual social network to know who has the power and influence to bring about change.

Terminology from Social Networks

The basic units of a network are called actors or nodes. They can be people, or websites, or whatever "things" you are considering, and are often indicated as a single dot in a visualization. The relationships between the actors are referred to as relational ties or edges. For example, an instance of liking someone or being friends would be indicated by an edge. We refer to pairs of actors as dyads, and triplets of actors as triads. For example, if we have an edge between node A and node B, and an edge between node B and node C, then triadic closure would be the existence of an edge between node A and node C.

We sometimes consider subgroups, also called subnetworks, which consist of a subset of the whole set of actors, along with their relational ties. Of course this means we also consider the group itself, which means the entirety of a "network." Note that this is a relatively easy concept in the case of, say, the Twitter network, but it’s very hard in the case of “liberals.”

We refer to a relation generally as a way of having relational ties between actors. For example, liking another person is a relation, but so is living with someone. A social network is the collection of some set of actors and relations.

There are actually a few different types of social networks. For example, the simplest case is that you have a bunch of actors connected by ties. This is a construct you’d use to display a Facebook graph—any two people are either friends or aren’t, and any two people can theoretically be friends.

In bipartite graphs the connections only exist between two formally separate classes of objects. So you might have people on the one hand and companies on the other, and you might connect a person to a company if she is on the board of that company. Or you could have people and the things they’re possibly interested in, and connect them if they really are.

Finally, there are ego networks, which is typically formed as “the part of the network surrounding a single person.” For example, it could be “the subnetwork of my friends on Facebook,” who may also know one another in certain cases. Studies have shown that people with higher socio-economic status have more complicated ego networks, and you can infer someone’s level of social status by looking at their ego network.

See Social Media - Six Degrees of Separation and Now Even Less and Visualization of the Osama bin Laden Letters

Data Journalism

Our second speaker of the night was Jon Bruner, an editor at O'Reilly who previously worked as the data editor at Forbes. He is broad in his skills: he does research and writing on anything that involves data.

Data journalism has been around for a while, but until recently, computer-assisted reporting was a domain of Excel power users. (Even now, if you know how to write an Excel program, you’re an elite.)
My Note: So I am beyond elite because I use Spotfire:)

If you work for anything besides a national daily, you end up doing everything by yourself: you come up with a question, you go get the data, you do the analysis, then you write it up. (Of course, you can also help and collaborate with your colleagues when possible.)

Writing Technical Journalism: Advice from an Expert

First of all, it involved lots of data visualization, because it’s a fast way of describing the bottom line of a dataset. Computer science skills are pretty important in data journalism, too. There are tight deadlines, and the data journalist has to be good with their tools and with messy data—because even federal data is messy.

Statistics, Bruno says, informs the way you think about the world.

Bruno admits to being a novice in the field of machine learning. However, he claims domain expertise as critical in data journalism, and you need to acquire a baseline layer of expertise quickly.

Of course communications and presentations are absolutely huge for data journalists. Their fundamental skill is translation: taking complicated stories and deriving meaning that readers will understand. They also need to anticipate questions, turn them into quantitative experiments, and answer them persuasively.

Here’s advice from Jon for anyone initiating a data journalism project: don’t have a strong thesis before you interview the experts. Go in with a loose idea of what you’re searching for and be willing to change your mind and pivot if the experts lead you in a new and interesting direction. Sounds kind of like exploratory data analysis!

Team Homework Exercise

Data Journalism: Critique One of My Health Data Stories/Products (How could it be better? Is there other data I should have included? How would you present it?)

2/25 Specific Data Science Tools and Applications 3

Purpose

Discuss Reading: Chapters 11 and 12, Present and Discuss Team Homework Exercises, Hands-on Class Exercise, and Team Homework Exercise.

My Resources

http://semanticommunity.info/Data_Science/Free_Data_Visualization_and_Analysis_Tools

Present and Discuss Team Homework Exercises

Data Journalism: Critique One of My Health Data Stories/Products (How could it be better? Is there other data I should have included? How would you present it?)
Chapter 11 Excerpts

Many of the models and examples in the book so far have been focused on the fundamental problem of prediction where your goal was to build a model to predict whether or not a person would be likely to prefer a certain item—a movie or a book, for example. There may be thousands of features that go into the model, and you may use feature selection to narrow those down, but ultimately the model is getting optimized in order to get the highest accuracy. When one is optimizing for accuracy, one doesn’t necessarily worry about the meaning or interpretation of the features, and especially if there are thousands of features, it’s well-near impossible to interpret at all.

We wish to emphasize here that it’s not simply the familiar correlation-causation trade-off you’ve perhaps had drilled into your head already, but rather that your intent when building such a model or system was not even to understand causality at all, but rather to predict. And that if your intent were to build a model that helps you get at causality, you would go about that in a different way.

A whole different set of real-world problems that actually use the same statistical methods (logistic regression, linear regression) as part of the building blocks of the solution are situations where you do want to understand causality, when you want to be able to say that a certain type of behavior causes a certain outcome. In these cases your mentality or goal is not to optimize for predictive accuracy, but rather to be able to isolate causes.

This chapter will explore the topic of causality, and we have two experts in this area as guest contributors, Ori Stitelman and David Madigan. Madigan’s bio will be in the next chapter and requires this chapter as background. We’ll start instead with Ori, who is currently a data scientist at Wells Fargo. As part of his job, he needed to create stories from data for experts to testify at trial, and he thus developed what he calls “data intuition” from being exposed to tons of different datasets.

Correlation Doesn’t Imply Causation

Causal inference is the field that deals with better understanding the conditions under which association can be interpreted as causality.

Asking Causal Questions

The natural form of a causal question is: What is the effect of \( x \) on \( y \)?

Some examples are: “What is the effect of advertising on customer behavior?” or “What is the effect of drug on time until viral failure?” or in the more general case, “What is the effect of treatment on outcome?”

It turns out estimating causal parameters is hard. In fact, the effectiveness of advertising is almost always considered a moot point because it’s so hard to measure. People will typically choose metrics of success that are easy to estimate but don’t measure what they want, and everyone makes decisions based on them anyway because it’s easier. But they have real negative effects. For example, marketers end up being rewarded for selling stuff to people online who would have bought something anyway.

The Gold Standard: Randomized Clinical Trials
The gold standard for establishing causality is the randomized experiment. This is a setup whereby we randomly assign some group of people to receive a “treatment” and others to be in the “control” group—that is, they don’t receive the treatment. We then have some outcome that we want to measure, and the causal effect is simply the difference between the treatment and control group in that measurable outcome.

But there’s bad news for randomized clinical trials as well, as we pointed out earlier. If we know treating someone with a drug will be better for them than giving them nothing, we can’t randomly not give people the drug.

The other problem is that they are expensive and cumbersome. It takes a long time and lots of people to make a randomized clinical trial work. On the other hand, not doing randomized clinical trials can lead to mistaken assumptions that are extremely expensive as well.

In conclusion, when they are possible, randomized clinical trials are the gold standard for elucidating cause-and-effect relationships. It’s just that they aren’t always possible.

Randomized clinical trials measure the effect of a certain drug averaged across all people. There is a push these days toward personalized medicine with the availability of genetic data, which means we stop looking at averages because we want to make inferences about the one.

A/B Tests

In software companies, what we described as random experiments are sometimes referred to as A/B tests. In fact, we found that if we said the word “experiments” to software engineers, it implied to them “trying something new” and not necessarily the underlying statistical design of having users experience different versions of the product in order to measure the impact of that difference using metrics.

The design goals for our experiment infrastructure are therefore: more, better, faster.

Second Best: Observational Studies

While the gold standard is generally understood to be randomized experiments or A/B testing, they might not always be possible, so we sometimes go with second best, namely observational studies.

An observational study is an empirical study in which the objective is to elucidate cause-and-effect relationships in which it is not feasible to use controlled experimentation.

Most data science activity revolves around observational data, although A/B tests, as you saw earlier, are exceptions to that rule. Most of the time, the data you have is what you get. You don’t get to replay a day on the market where Romney won the presidency, for example.

There are all kinds of pitfalls with observational studies.

It’s a general problem with regression models on observational data. You have no idea what’s going on.

The fundamental problem in observational studies is: a trend that appears in different groups of data disappears when these groups are combined, or vice versa. This is sometimes called Simpson's Paradox.
The Rubin causal model is a mathematical framework for understanding what information we know and don't know in observational studies.

It's meant to investigate the confusion when someone says something like, “I got lung cancer because I smoked.” Is that true? If so, you’d have to be able to support the statement, “If I hadn’t smoked, I wouldn’t have gotten lung cancer,” but nobody knows that for sure.

This is sometimes called the fundamental problem of causal inference.

We can represent the concepts of causal modeling using what is called a causal graph.

Let’s say we have a population of 100 people that takes some drug, and we screen them for cancer. Say 30 of them get cancer, which gives them a cancer rate of 0.30. We want to ask the question, did the drug cause the cancer?

To answer that, we’d have to know what would’ve happened if they hadn’t taken the drug. Let’s play God and stipulate that, had they not taken the drug, we would have seen 20 get cancer, so a rate of 0.20. We typically measure the increased risk of cancer as the difference of these two numbers, and we call it the causal effect. So in this case, we’d say the causal effect is 10%.

But we don’t have God’s knowledge, so instead we choose another population to compare this one to, and we see whether they get cancer or not, while not taking the drug. Say they have a natural cancer rate of 0.10. Then we would conclude, using them as a proxy, that the increased cancer rate is the difference between 0.30 and 0.10, so 20%. This is of course wrong, but the problem is that the two populations have some underlying differences that we don’t account for.

If these were the “same people,” down to the chemical makeup of each others' molecules, this proxy calculation would work perfectly. But of course they’re not.

Three Pieces of Advice

Knowing exactly how the data was generated will also help you ascertain whether the assumptions you make are reasonable.

The first step in a data analysis should always be to take a step back and figure out what you want to know.

Make sure you’ve created a reasonable narrative and ways to check its validity.

Hands-on Class Exercise

Project TYCHO Data for Health See Research Notes

Result: See Spotfire Web Player and Spotfire File

Also see: http://omop.org/ and http://elmo.omop.org/

Chapter 12 Excerpts

The contributor for this chapter is David Madigan, professor and chair of statistics at Columbia. Madigan has over 100 publications in such areas as Bayesian statistics, text mining, Monte Carlo methods, pharmacovigilance, and probabilistic graphical models.
He learned about controlling graphics cards on PCs, but he still didn’t know about data.

He was program chair of the KDD conference, among other things. He learned C and Java, R, and S+. But he still wasn’t really working with data yet.

He claims he was still a typical academic statistician: he had computing skills but no idea how to work with a large-scale medical database, 50 different tables of data scattered across different databases with different formats.

He then went to an Internet startup where he and his team built a system to deliver real-time graphics on consumer activity.

Since then he’s been working in big medical data stuff.

Thought Experiment

We now have detailed, longitudinal medical data on tens of millions of patients. What can we do with it?

To be more precise, we have tons of phenomenological data: this is individual, patient-level medical record data. The largest of the databases has records on 80 million people: every prescription drug, every condition ever diagnosed, every hospital or doctor’s visit, every lab result, procedures, all timestamped.

But we still do things like we did in the Middle Ages; the vast majority of diagnosis and treatment is done in a doctor’s brain. Can we do better?

Can we harness these data to do a better job delivering medical care?

This is a hugely important clinical problem, especially as a healthcare insurer. Can we intervene to avoid hospitalizations?

So for example, there was a prize offered on Kaggle, called “Improve Healthcare, Win $3,000,000.” It challenged people to accurately predict who is going to go to the hospital next year. However, keep in mind that they’ve coarsened the data for privacy reasons.

There are a lot of sticky ethical issues surrounding this 80 million person medical record dataset. What nefarious things could we do with this data? Instead of helping people stay well, we could use such models to gouge sick people with huge premiums, or we could drop sick people from insurance altogether.

This is not a modeling question. It’s a question of what, as a society, we want to do with our models.

Modern Academic Statistics

It used to be the case, say 20 years ago, according to Madigan, that academic statisticians would either sit in their offices proving theorems with no data in sight—they wouldn’t even know how to run a t-test—or sit around in their offices and dream up a new test, or a new way of dealing with missing data, or something like that, and then they’d look around for a dataset to whack with their new method. In either case, the work of an academic statistician required no domain
expertise.

Nowadays things are different. The top stats journals are more deep in terms of application areas, the papers involve deep collaborations with people in social sciences or other applied sciences. Madigan sets an example by engaging with the medical community.

Madigan went on to make a point about the modern machine learning community, which he is or was part of: it’s a newish academic field, with conferences and journals, etc., but from his perspective, it’s characterized by what statistics was 20 years ago: invent a method, try it on datasets. In terms of domain expertise engagement, it’s a step backward instead of forward.

Not to say that statistics are perfect; very few academic statisticians have serious hacking skills, with Madigan’s colleague Mark Hansen being an unusual counterexample. In Madigan’s opinion, statisticians should not be allowed out of school unless they have such skills.

Medical Literature and Observational Studies

As you may not be surprised to hear, medical journals are full of observational studies. The results of these studies have a profound effect on medical practice, on what doctors prescribe, and on what regulators do.

For example, after reading the paper entitled “Oral bisphosphonates and risk of cancer of oesophagus, stomach, and colorectum: case-control analysis within a UK primary care cohort” (by Jane Green, et al.), Madigan concluded that we see the potential for the very same kind of confounding problem as in the earlier example with aspirin. The conclusion of the paper is that the risk of cancer increased with 10 or more prescriptions of oral bisphosphonates.

It was published on the front page of the New York Times, the study was done by a group with no apparent conflict of interest, and the drugs are taken by millions of people. But the results might well be wrong and, indeed, were contradicted by later studies.

There are thousands of examples of this. It’s a major problem and people don’t even get that it’s a problem.

Billions upon billions of dollars are spent doing medical studies, and people’s lives depend on the results and the interpretations. We should really know if they work.

Research Experiment (Observational Medical Outcomes Partnership)

To address the issues directly, or at least bring to light the limitations of current methods and results, Madigan has worked as a principal investigator on the OMOP research program, making significant contributions to the project’s methodological work including the development, implementation, and analysis of a variety of statistical methods applied to various observational databases.

About OMOP, from Its Website

In 2007, recognizing that the increased use of electronic health records (EHR) and availability of other large sets of marketplace health data provided new learning opportunities, Congress directed the FDA to create a new drug
surveillance program to more aggressively identify potential safety issues. The FDA launched several initiatives to achieve that goal, including the well-known Sentinel program to create a nationwide data network.

In partnership with PhRMA and the FDA, the Foundation for the National Institutes of Health launched the Observational Medical Outcomes Partnership (OMOP), a public-private partnership. This interdisciplinary research group has tackled a surprisingly difficult task that is critical to the research community’s broader aims: identifying the most reliable methods for analyzing huge volumes of data drawn from heterogeneous sources.

Employing a variety of approaches from the fields of epidemiology, statistics, computer science, and elsewhere, OMOP seeks to answer a critical challenge: what can medical researchers learn from assessing these new health databases, could a single approach be applied to multiple diseases, and could their findings be proven? Success would mean the opportunity for the medical research community to do more studies in less time, using fewer resources and achieving more consistent results. In the end, it would mean a better system for monitoring drugs, devices, and procedures so that the healthcare community can reliably identify risks and opportunities to improve patient care.

Madigan and his colleagues took 10 large medical databases, consisting of a mixture of claims from insurance companies and electronic health records (EHR), covering records of 200 million people in all. This is Big Data unless you talk to an astronomer.

They mapped the data to a common data model and then they implemented every method used in observational studies in healthcare. Altogether they covered 14 commonly used epidemiology designs adapted for longitudinal data. They automated everything in sight. Moreover, there were about 5,000 different “settings” on the 14 methods.

The idea was to see how well the current methods do on predicting things we actually already know.

They bought all the data, made the methods work automatically, and did a bunch of calculations in the Amazon cloud.

If you go to http://elmo.omop.org, you can see the AUC for a given database and a given method. The data used in this study was current in mid-2010. To update this, you’d have to get the latest version of the database, and rerun the analysis. Things might have changed.

In the study, 5,000 different analyses were run. Is there a good way of combining them to do better? How about incorporating weighted averages or voting methods across different strategies? The code is publicly available and might make a great PhD thesis.

**Team Homework Exercise**

All the KDD Cups, with their tasks and corresponding datasets, can be found at http://www.kdd.org/kddcup/index.php

Here’s a list:

- KDD Cup 2010: Student performance evaluation
- KDD Cup 2009: Customer relationship prediction
- KDD Cup 2008: Breast cancer
- KDD Cup 2007: Consumer recommendations
- KDD Cup 2006: Pulmonary embolisms detection from image data
- KDD Cup 2005: Internet user search query categorization
- KDD Cup 2004: Particle physics; plus protein homology prediction
- KDD Cup 2003: Network mining and usage log analysis

http://semanticommunity.info/Data_Science/Practical_Data_Science_for_Data_Scientists

Updated: Wed, 23 Sep 2015 02:36:14 GMT

Powered by mindtouch
• KDD Cup 2002: BioMed document; plus gene role classification
• KDD Cup 2001: Molecular bioactivity; plus protein locale prediction
• KDD Cup 2000: Online retailer website clickstream analysis
• KDD Cup 1999: Computer network intrusion detection
• KDD Cup 1998: Direct marketing for profit optimization
• KDD Cup 1997: Direct marketing for lift curve optimization

See my work with the [KDD Cup](http://www.kdd.org/kddcup/index.php) data sets where I have updated this to include 2011-2013.

Form Teams (Same or New), Ask Me Questions, and Prepare to Present One of These Next Week

### 3/4 Specific Data Science Tools and Applications 4

**Purpose**

Discuss Reading: Chapters 13 and 14, Present and Discuss Team Homework Exercises, Hands-on Class Exercise, and Team Homework Exercise.

**My Resources**

- [http://semanticommunity.info/Data_Science/Free_Data_Visualization_and_Analysis_Tools](http://semanticommunity.info/Data_Science/Free_Data_Visualization_and_Analysis_Tools)
- [http://semanticommunity.info/Cloudera](http://semanticommunity.info/Cloudera)

**Present and Discuss Team Homework Exercises**

All the KDD Cups, with their tasks and corresponding datasets (1997-2010), can be found at [http://www.kdd.org/kddcup/index.php](http://www.kdd.org/kddcup/index.php). See my work with the [KDD Cup](http://www.kdd.org/kddcup/index.php) data sets where I have updated this to include 2011-2013.

Present One of These

**Chapter 13 Excerpts**

The Life of a Chief Data Scientist

Claudia Perlich likes to understand something about the world by looking directly at the data. Claudia's skill set includes 15 years working with data, where she’s developed data intuition by delving into the data generating process, a crucial piece of the puzzle.

Claudia drew a distinction between different types of data mining competitions. The first is the "sterile" kind, where you’re given a clean, prepared data matrix, a standard error measure, and features that are often anonymized. This is a pure machine learning problem.

All the KDD Cups, with their tasks and corresponding datasets, can be found at [http://www.kdd.org/kddcup/index.php](http://www.kdd.org/kddcup/index.php)
On the other hand, you have the “real world” kind of data mining competition, where you’re handed raw data (which is often in lots of different tables and not easily joined), you set up the model yourself, and come up with task-specific evaluations. This kind of competition simulates real life more closely, which goes back to Rachel’s thought experiment earlier in this book about how to simulate the chaotic experience of being a data scientist in the classroom. You need practice dealing with messiness.

Examples of this second kind are KDD cup 2007, 2008, and 2010. If you’re in this kind of competition, your approach would involve understanding the domain, analyzing the data, and building the model. The winner might be the person who best understands how to tailor the model to the actual question.

Claudia prefers the second kind, because it’s closer to what you do in real life.

Claudia claims that data and domain understanding are the single most important skills you need as a data scientist. At the same time, this can’t really be taught—it can only be cultivated.

Leakage

The contestants’ best friend and the organizer and practitioners’ worst nightmare. There’s always something wrong with the data, and Claudia has made an artform of figuring out how the people preparing the competition got lazy or sloppy with the data.

My Note: I also did this for the Heritage Health Prize

We need to know what the purpose of the model is and how it is going to be used in order to decide how to build it, and whether it’s working.

Winning a competition on leakage is easier than building good models. But even if you don’t explicitly understand and game the leakage, your model will do it for you. Either way, leakage is a huge problem with data mining contests in general.

Claudia and her coauthors describe in the paper referenced earlier a suggested methodology for avoiding leakage as a two-stage process of tagging every observation with legitimacy tags during collection and then observing what they call a learn-predict separation.

Parting Thoughts

According to Claudia, humans are not meant to understand data. Data is outside of our sensory systems, and there are very few people who have a near-sensory connection to numbers. We are instead designed to understand language.

We are also not meant to understand uncertainty: we have all kinds of biases that prevent this from happening that are well documented. Hence, modeling people in the future is intrinsically harder than figuring out how to label things that have already happened.

Even so, we do our best, and this is through careful data generation, meticulous consideration of what our problem is, making sure we model it with data close to how it will be used, making sure we are optimizing to what we actually desire, and doing our homework to learn which algorithms fit which tasks.
Hands-on Class Exercise

More KDD Cups datasets

Chapter 14 Excerpts

David Crawshaw and Josh Wills worked at Google on the Google+ data science team, though the two of them never actually worked together because Josh Wills left to go to Cloudera and David Crawshaw replaced him in the role of tech lead.

Josh and David were responsible at Google for collecting data (frontend and backend logging), building the massive data pipelines to store and munge the data, and building up the engineering infrastructure to support analysis, dashboards, analytics, A/B testing, and more broadly, data science.

MapReduce, which was developed at Google, is an algorithm and framework for dealing with massive amounts of data that has recently become popular in industry. The goal of this chapter is to clear up some of the mysteriousness surrounding MapReduce. It’s become such a buzzword, and many data scientist job openings are advertised as saying “must know Hadoop” (the open source implementation of MapReduce). We suspect these ads are written by HR departments who don’t really understand what MapReduce is good for and the fact that not all data science problems require MapReduce. But as it’s become such a data science term, we want to explain clearly what it is, and where it came from. You should know what it is, but you may not have to use it—or you might, depending on your job.

MapReduce is the third category of algorithms we brought up in Chapter 3 (the other two being machine learning algorithms and optimization algorithms). As a point of comparison, given that algorithmic thinking may be new to you, we’ll also describe another data engineering algorithm and framework, Pregel, which enables large-scale graph computation (it was also developed at Google and open sourced).

David, as an engineer, revises the question we’ve asked before in this book: what is Big Data? It’s a buzzword mostly, but it can be useful. He tried this as a working definition:

You’re dealing with Big Data when you’re working with data that doesn’t fit into your computer unit.

Given this, is Big Data going to go away? Can we ignore it?

David claims we can’t, because although the capacity of a given computer is growing exponentially, those same computers also make the data. The rate of new data is also growing exponentially. So there are actually two exponential curves, and they won’t intersect any time soon.

Word Frequency Problem
Because counting and sorting are fast, this scales to ~100 million words. The limit now is computer memory—if you think about it, you need to get all the words into memory twice: once when you load in the list of all words, and again when you build a way to associate a count for each word.

Hold up, computers nowadays are many-core machines; let’s use them all! Then the bandwidth will be the problem, so let’s compress the inputs, too. That helps, and moreover there are better alternatives that get complex.
Now take a hundred computers. We can process a thousand trillion words. But then the “fan-in,” where the results are sent to the controller, will break everything because of bandwidth problem. We need a tree, where every group of 10 machines sends data to one local controller, and then they all send back to super controller. This will probably work.

It’s like this:

• The first 10 computers are easy;
• The first 100 computers are hard; and
• The first 1,000 computers are impossible.

There’s really no hope.

Or at least there wasn’t until about eight years ago. At Google now, David uses 10,000 computers regularly.

MapReduce allows us to stop thinking about fault tolerance; it is a platform that does the fault tolerance work for us. Programming 1,000 computers is now easier than programming 100. It’s a library to do fancy things.

To use MapReduce, you write two functions: a mapper function, and then a reducer function. It takes these functions and runs them on many machines that are local to your stored data. All of the fault tolerance is automatically done for you once you’ve placed the algorithm into the map/reduce framework.

The mapper takes each data point and produces an ordered pair of the form (key, value). The framework then sorts the outputs via the “shuffle,” and in particular finds all the keys that match and puts them together in a pile. Then it sends these piles to machines that process them using the reducer function. The reducer function’s outputs are of the form (key, new value), where the new value is some aggregate function of the old values.

But MapReduce isn’t ideal for, say, iterative algorithms where you take a just-computed estimation and use it as input to the next stage in the computation—this is common in various machine learning algorithms that use steepest descent convergence methods.

Pregel

Just as a point of contrast, another algorithm for processing large-scale data was developed at Google called Pregel. This is a graph-based computational algorithm, where you can imagine the data itself has a graph-based or network-based structure. The computational algorithm allows nodes to communicate with other nodes that they are connected to. There are also aggregators that are nodes that have access to the information that all the nodes have, and can, for example, sum together or average any information that all the nodes send to them.

My Note: We will learn about the Cray YarcData Graph Appliance next week with massive memory!

Josh Wills is Cloudera’s director of data science, working with customers and engineers to develop Hadoop-based solutions across a wide range of industries.

Josh is also known for pithy data science quotes, such as: “I turn data into awesome” and the one we saw way back in the start of the book: “data scientist (noun): Person who is better at statistics than any software engineer and better at
Josh had some observations about the job of a data scientist. A data scientist spends all their time doing data cleaning and preparation—a full 90% of the work is this kind of data engineering. When deciding between solving problems and finding insights, a data scientist solves problems. A bit more on that: start with a problem, and make sure you have something to optimize against. Parallelize everything you do.

It’s good to be smart, but being able to learn fast is even better: run experiments quickly to learn quickly.

Josh keeps everything. He’s a fan of reproducible research, so he wants to be able to rerun any phase of his analysis. He keeps everything. This is great for two reasons. First, when he makes a mistake, he doesn’t have to restart everything. Second, when he gets new sources of data, it’s easy to integrate them in the point of the flow where it makes sense.

My Note: I met with Josh and interviewed him for a story. He liked my story that was behind his example!

There’s also a Big Data economic law, which states that no individual record is particularly valuable, but having every record is incredibly valuable. So, for example, for a web index, recommendation system, sensor data, or online advertising, one has an enormous advantage if one has all the existing data, even though each data point on its own isn’t worth much.

There are two core components to Hadoop, which is the open source version of Google’s GFS and MapReduce. Cloudera was cofounded by Doug Cutting, one of the creators of Hadoop, and Jeff Hammerbacher, who we mentioned back in Chapter 1 because he co-coined the job title “data scientist” when he worked at Facebook and built the data science team there.

Cloudera is like Red Hat for Hadoop, by which we mean they took an open source project and built a company around it.

If you are working at a company that has a Hadoop cluster, it’s likely that your first experience will be with Apache Hive, which provides a SQL-style abstraction on top of HDFS and MapReduce. Your first MapReduce job will probably involve analyzing logs of user behavior in order to get a better understanding of how customers are using your company’s products.

If you are exploring Hadoop and MapReduce on your own for building analytical applications, there are a couple of good places to start. One option is to build a recommendation engine using Apache Mahout, a collection of machine learning libraries and command-line tools that works with Hadoop.

**Team Homework Exercise**

Your brief chapter of what you have learned so far and hope to yet learn in this class.

### 3/18 Data Science Students and Careers
Purpose

Discuss Reading: Chapters 15 and 16, Present and Discuss Team Homework Exercises, Hands-on Class Exercise, and Team Homework Exercise.

My Resources

http://semanticommunity.info/#Events_Calendar

http://semanticommunity.info/AmericasDataFest

Present and Discuss Team Homework Exercises

Your brief chapter of what you have learned so far and hope to yet learn in this class.

Chapter 15 Excerpts

We invited the students who took Introduction to Data Science version 1.0 to contribute a chapter to the book. They chose to use their chapter to reflect on the course and describe how they experienced it.

When you’re learning data science, you can’t start anywhere except the cutting edge.

The class itself became sort of an iterative definition of data science.

We often worked on industry-related problems, and our homework was to be completed in the form of a clear and thoughtful report—something we could pride ourselves in presenting to an industry professional. Most importantly we often had little choice but to reach out beyond our comfort zones to one another to complete assignments.

And as our skills improved, we were increasingly able to focus on the analysis of the data. But to actually finish our homework on time, we had to be willing to learn from the different backgrounds of the students in the class. In fact, it became crucial for us to find fellow students with complementary skills to complete our assignments.

Data science’s body of knowledge is changing and distributed, to the extent that the only way of finding out what you should know is by looking at what other people know.

There is a focus in the business world to use data science to sell advertisements. You may have access to the best dataset in the world, but if the people employing you only want you to find out how to best sell shoes with it, is it really worth it?

The philosophy that was repeatedly pushed on us was that understanding the statistical tools of data science without the context of the larger decisions and processes surrounding them strips them of much of their meaning.

The philosophy that was repeatedly pushed on us was that understanding the statistical tools of data science without the context of the larger decisions and processes surrounding them strips them of much of their meaning.

The students improved on the original data science profiles.
Hands-on Class Exercise

Start AmericasDataFest Competition

Chapter 16 Excerpts

The two main goals of this book were to communicate what it’s like to be a data scientist and to teach you how to do some of what a data scientist does.

In the industry, they say you can’t learn data science in a university or from a book, that it has to be “on the job.” But maybe that’s wrong, and maybe this book has proved that. You tell us.

Data science could be defined simply as what data scientists do, as we did earlier when we talked about profiles of data scientists.

Let's define data science beyond a set of best practices used in tech companies.

Data science happens both in industry and in academia, i.e., where or what domain data science happens in is not the issue—rather, defining it as a “problem space” with a corresponding “solution space” in algorithms and code and data is the key.

Data science is a set of best practices used in tech companies, working within a broad space of problems that could be solved with data, possibly even at times deserving the name science.

The best minds of my generation are thinking about how to make people click ads… That sucks.
— Jeff Hammerbacher

We’d like to encourage the next-gen data scientists to become problem solvers and question askers, to think deeply about appropriate design and process, and to use data responsibly and make the world better, not worse.

What Would a Next-Gen Data Scientist Do?

• Next-gen data scientists don’t try to impress with complicated algorithms and models that don’t work. They spend a lot more time trying to get data into shape than anyone cares to admit—maybe up to 90% of their time. Finally, they don’t find religion in tools, methods, or academic departments. They are versatile and interdisciplinary.

• Next-gen data scientists remain skeptical—about models themselves, how they can fail, and the way they’re used or can be misused. Next gen data scientists understand the implications and consequences of the models they’re building. They think about the feedback loops and potential gaming of their models.

• Next-gen data scientists don’t let money blind them to the point that their models are used for unethical purposes. They seek out opportunities to solve problems of social value and they try to consider the consequences of their models.

Follow Ben Shneiderman’s 8 Golden Rules of Data Science

Source: "8 Golden Rules of Data Science"

Preparation

• Choose actionable problems & appropriate theories
• Consult domain experts & generalists

Exploration
• Examine data in isolation & contextually
• Keep cleaning & add related data
• Apply visualizations & statistical patterns, clusters, gaps, outliers, missing & uncertain data

Decision
• Evaluate your efficacy, refine your theory
• Take responsibility, own your failures
• World is complex, proceed with humility

Team Homework Exercise

Study about Graph Databases, Graph Computing, and Semantic Medline

Review Wiki and View Videos: YarcData Videos (Schizo-7 minutes, Cancer-21 minutes).

Ask Me Questions and Prepare to Ask Questions Next Week

3/25 Semantic Medline on the YarcData Graph Appliance

Discuss Graph Databases and Computing: http://semanticommunity.info/Data_Science/Graph_Databases

My Resource: Wiki YarcData Videos (Schizo-7 minutes, Cancer-21 minutes).

4/1 Class Project Proposals Presentations and Feedback

Present and Discuss Proposed Class Projects: See Sign Up Schedule To Be Posted

4/8 Discuss March 4-5, 2014, NIST Data Science Symposium


My Resource: Wiki Wiki Slides

4/15 Class Projects and Final Exam

Present and Discuss Class Projects: See Sign Up Schedule To Be Posted

4/22 Class Projects and Final Exam

Present and Discuss Class Projects: See Sign Up Schedule To Be Posted

4/29 Class Projects and Final Exam

Present and Discuss Class Projects: See Sign Up Schedule To Be Posted
5/7-5/14 Final Exam

TBA

Required Textbook

Doing Data Science: http://shop.oreilly.com/product/0636920028529.do

Optional Supplemental Reading: Data Science Starter Kit: http://shop.oreilly.com/category/get/data-science-kit.do
DC Data Community: http://datacommunitydc.org/blog/about/
DC Data Community Calendar: http://datacommunitydc.org/blog/calendar/

Technology Requirements

Internet and Free Tools like Spotfire Cloud: https://spotfire.cloud.tibco.com/tsc/#/compproductrequest
NodeXL: http://nodexl.codeplex.com/

Course Description

(from GMU course catalog): (3 credits)
Learn how to do practical data science by doing hands-on exercises in class and class projects that build your portfolio from this and other classes for future employment. Read a recent book based on the Columbia University Introduction to Data Science Class and see a large number of data science stories, products, and applications produced by the instructor with public open government data sources and for practical problems. Gain experience with the leading open source, free tools, and the most advanced tools for relational and graph databases (e.g. Hadoop, Neo4j, YarcData, etc.). Participate in the local Data Science Community for further learning and networking opportunities.

Prerequisites: TBD

Course Objectives

• to learn what data science is and what a data scientist does
• to learn how to produce data science stories, products, and applications
• to learn the kinds of tools a data scientists uses from Excel to the YarcData Graph Appliance
• to learn from the work of the world's leading data scientists for government, private industry, and academic applications
• to work on real world big and small data science data sets and problems
• to participate in the local data science communities of practitioners

http://semanticommunity.info/Data_Science/Practical_Data_Science_for_Data_Scientists
Updated: Wed, 23 Sep 2015 02:36:14 GMT
Powered by mindtouch