**빅데이터 분석**

**Introduction to Analyzing Big Data**

Ji-Hyeong Han

## ( [jhhan@seoultech.ac.kr](mailto:jhhan@seoultech.ac.kr) )

Dept. of Computer Science and Engineering



**Today’s Contents**

* Notice
* Syllabus
  + Course overview
  + Course objectives
  + Lecture material
  + Lecture schedule
  + Grade policy
  + Term project

## Introduction of analyzing big data

1. The challenges of data science
2. Introduction to Apache Spark
3. Summary

2

**Notice**

## Update the personal information

* + Cellphone number, e-mail address, picture
  + If you don’t update the personal information on the system, you can not receive the important notice and all the responsibility is yours

## ABEEK survey

* + Please answer ABEEK survey through SUIS system

## Student counselling

* + Any subjects are ok, i.e. about this course, career, any troubles
  + If you want to have a counselling, you must arrange the time schedule first by e-mail

3

**Syllabus (1/4)**

## Course overview

* + Give an introduction to analyze the big data using Spark
  + Cover the basics of data processing in Spark and Scala
  + Provide the students some machine learning methods with Spark

## By the end of this course, the students should be able to:

* + Understand the concepts of big data and big data analyzation
  + Understand the basics of Spark and Scala
  + Apply machine learning methods to analyze the big data in Spark

## Lecture material

* + Sandy Ryza, “Advanced Analytics with Spark”, 2nd edition, O’REILLY, 2017
  + Lecture ppts will be provided at e-class

4

**Syllabus (2/4)**

## Lecture schedule

|  |  |  |
| --- | --- | --- |
| 주별 | 강의내용 | 강의방법,과제,평가 |
| 1 | Introduction of Analyzing Big Data | 온라인 강의 (동영상) |
| 2 | Introduction to Data Analysis with Scala and Spark | 온라인 강의 및 실습 (동영상&zoom, 자세한 일정은 추후 공지) |
| 3 | Collaborating Filtering & Alternating Least Squares | 온라인 강의 및 실습 (동영상&zoom, 자세한 일정은 추후 공지) |
| 4 | Predicting Forest Cover with Decision Trees | 온라인 강의 및 실습 (동영상&zoom, 자세한 일정은 추후 공지) |
| 5 | Anomaly Detection in Network Traffic with K-means Clustering | 온라인 강의 및 실습 (동영상&zoom, 자세한 일정은 추후 공지) |
| 6 | Understanding Wikipedia with Latent Semantic Analysis | 온라인 강의 및 실습 (동영상&zoom, 자세한 일정은 추후 공지) |
| 7 | Analyzing Co-Occurrence Networks with GraphX | 온라인 강의 및 실습 (동영상&zoom, 자세한 일정은 추후 공지) |
| 8 | Midterm exam | Offline Midterm exam |
| 9 | Analyzing Neuroimaging Data with PySpark and Thunder | 온라인 or 오프라인 강의 및 실습 (사회적 거리두기 단계에 따라 가변) |
| 10 | Deep Learning (Convolutional Neural Network) | 온라인 or 오프라인 강의 및 실습 (사회적 거리두기 단계에 따라 가변) |
| 11 | Term Project Proposal Presentation | 온라인 or 오프라인 발표 (사회적 거리두기 단계에 따라 가변) |
| 12 | Term Project Development | 온라인 or 오프라인 토론 (사회적 거리두기 단계에 따라 가변) |
| 13 | Term Project Progress Presentation | 온라인 or 오프라인 발표 (사회적 거리두기 단계에 따라 가변) |
| 14 | Term Project Development | 온라인 or 오프라인 토론 (사회적 거리두기 단계에 따라 가변) |
| 15 | Term Project Final Presentation | 온라인 or 오프라인 발표 (사회적 거리두기 단계에 따라 가변) |

5

**Syllabus (3/4)**

## Grade policy

* + Midterm exam (offline exam, 35%), Term Project (online or offline, 35%), Quiz (online quiz two times, 10%+10%=20%),

Attendance (online or offline lecture attendance, 10%)

## Term project

* + After the midterm exam, you should do the term project
  + You have to make a team (up to 4 people) before midterm exam
  + It is free topic, so you can pick any topics you like
  + You can consider the topics and datasets from Kaggle ([www.kaggle.com](http://www.kaggle.com/))
    - A platform for predictive modelling and analytics competitions in which companies and researchers post data
    - Many data science problems and opened data sets
  + This course is based on Spark

But if you want to use other big data platforms or machine learning libraries for the term project, it is OK

* + The detailed schedule and evaluation policy will be announced later

6

**Syllabus (4/4)**

## 동영상 강의 및 실시간 zoom 실습 관련

* + 이론 강의는 기본적으로 동영상 강의
  + 이론 강의에 실습해야 할 코드 부분들이 포함되는데, 실제 실습을 진행하면서 막히는 부분이나 궁금했던 부분들을 질의 응답하는 실시간 zoom 도 함께 진행 예정
  + 실시간 zoom 진행 일정은 추후 공지 예정이며, 오프라인 강의시간 중

1시간 가량 진행 예정

* + - 매주 진행할지, 정확히 어느 시간대에 진행할지 등 자세한 사항은 강의를 진행 하며 결정하여 공지
  + 실습 질의응답을 위한 실시간 zoom에는 HCIR Lab 연구실 대학원생들이

TA로 도움을 줄 예정

7

# Introduction of Analyzing Big Data

### The Challenges of Data Science

1. **Introduction to Apache Spark**
2. **Summary**

8

# The Challenges of Data Science

9

**What is big data analyzation?**

## Big data analyzation examples

* + Build a model to detect credit card fraud using thousands of features and billions of transactions
  + Intelligently recommend millions of products to millions of users
  + Estimate financial risk through simulations of portfolios that include millions of instruments
  + Easily manipulate data from thousands of human genomes to detect genetic associations with disease

## Live in an age of big data

* + We have tools for:
    - Collecting, storing, and processing information at a scale previously unheard of
  + Thanks to ecosystem of open source software
    - Leverage clusters of commodity computers to store and process massive amounts of data
    - Ex: Apache Hadoop

10

**What is big data analyzation?**

## Data science

* + A gap between having access to tools and all this data and doing something useful with it
  + Practice of turning raw data into something useful that non-data scientists might care about

## Small data set analyzation

* + Open source frameworks
    - R, PyData stack, Octave
    - Rapid analysis and model building over small data sets

## What should an equivalent process for small data set that can clusters of computers to achieve the same outcomes on huge data set?

* + Simply extend frameworks for small data sets to run on multiple machines?

11

**What is big data analyzation?**

## The challenges of distributed computing

* + Rethink many of the basic assumptions that we rely on single-node systems
  1. Network transfer rates
     + Data must be partitioned across many nodes on a cluster
     + Network transfer rates are slower than memory access
     + So, algorithms that have wide data dependencies will suffer
  2. Node failure
     + As the number of machines working on a problem increases, the probability of a failure increases

## Require a programming paradigm that is sensitive to the characteristics of distributed computing

12

**The Challenges of Data Science**

1. Preprocessing of data
   * The vast majority of work that goes into conducting successful analysis
   * Data is messy
     + Need to cleansing, munging, fusing, mushing, and etc.
   * Large data sets
     + Not amenable to direct examination by humans
     + Require computational methods to even discover what preprocessing steps are needed
     + To optimize model performance, require far more time in feature engineering and selection than in choosing and writing algorithms
   * Ex: detect fraudulent purchases on a website
     + First, choose from a wide variety of potential features

(purchase order, IP location info, login times, click logs, navigation logs)

* + - Then, converting each of these chosen features to vectors fit for machine learning algorithms

13

**The Challenges of Data Science**

1. Iteration
   * A fundamental part of data science
   * Require multiple passes over the same data to model and analysis the data
     + Because of characteristics of machine learning algorithms and statistical procedures
     + Repeated scans over the input data to reach convergence
   * If a framework requires reading the same data set from disk at each time
     + Slow down the process because of disk IO delay

14

**The Challenges of Data Science**

1. Data application
   * The point of data science
     + To make data useful to non-data scientists
     + So, the task isn’t over when a well-performing model has been built
   * Ex: recommendation engine, fraud detection system
   * The model is a part of production service
     + Need to be rebuilt periodically or even in real time
   * Distinction between analytics in the lab and analytics in the factory
     + Lab: exploratory analytics (understand the data, test theories, experiments)
     + Factory: operational analytics (package models, improve accuracy, care about SLA and uptime)
     + In general, using R or Python for exploratory analytics and using Java or C++ for operational analytics
   * Require a framework that makes modeling easy but also a good fit for production system to save time

15

# Introduction to Apache Spark

16

**Apache Spark**

## An open source framework that combines an engine for distributing programs across clusters of machines with a model for writing programs a top of it.

* History
  + Originated at the UC Berkeley AMPLab
  + Now managed by Apache Software Foundation

## Provide high-level APIs

* + Java, Scala, Python, R

## Support higher-level tools

* + Spark SQL, MLlib, GraphX, Spark Streaming

17

**Compared to Apache Hadoop’s MapReduce**

## Apache Hadoop’s MapReduce

* + Revolutionized computation over huge data set by executing in parallel across hundreds to thousands of machines
  + Achieve near linear scalability
    - As the data size increases, we can throw more computers at it and see jobs complete in the same amount of time
  + Break up work into small tasks and can accommodate task failures without compromising the job

## Similarity between MapReduce and Spark

* + Maintain MapReduce’s linear scalability and fault tolerance

18

**Compared to Apache Hadoop’s MapReduce**

## Improvement of Spark compared to MapReduce

* 1. Rather than relying on a rigid “map-then-reduce” format,

Spark can execute a more general directed acyclic graph (DAG)

* + - MapReduce must write out intermediate results to the distributed filesystem
    - Spark can pass them directly to the next step in the pipeline
  1. Rich set of transformations that enable users to express computation more naturally
     + A strong developer focus and streamlined API that can represent complex pipelines in a few lines of code
  2. In-memory processing
     + “Dataset” & “DataFrame” abstractions enable developers to materialize any point in a processing pipeline into memory across the cluster
     + When want to deal with the same data set, Spark need not recompute it or reload it from disk
     + Well suited for highly iterative algorithms and reactive applications that quickly respond to user queries

19

# Summary

20

**Summary**

* Data science
  + Turning raw data into something useful that non-data scientists might care about
* Analyzing big data
  + Collecting, storing, and processing information at a scale previously unheard of
  + Need distributed computing
* Challenge of data science
  + Data preprocessing
  + Iteration
  + Data application
* Apache Spark
  + A fast and general-purpose cluster computing system
  + Linear scalability and fault tolerance
* Advantages of Spark
  + More general directed acyclic graph (DAG) rather than “map-then-reduce”
  + Developer focus and streamlined APIs
  + In-memory processing

21