No Cell-phones in the class.

If you need to use a laptop, please sit in the back row.
I will ask you to turn off your laptop if it seems to be distracting to others.

**FAQs**

- PAD has been posted
  - August 31, 5:00 PM via Canvas
  - Individual submission (No team submission)
- Accommodation request, honor student
  - Contact me by August 31 2018
- Readings
  - Reading research papers
  - Keshav’s “How to read a paper
  - “How to Read and Understand a Scientific Paper: A Step-by-Step Guide for Non-Scientists”

**Part 0. Introduction**

Big Data Analytics - Data Collection, Sampling, and Preprocessing

---

**This Material is Built Based on,**


**Part 0. Introduction**

Big Data Analytics - Data Collection, Sampling, and Preprocessing

---

**Part 0. Introduction**

Big Data Analytics - Data Collection, Sampling, and Preprocessing

---

**Analytics Process Model**

The most time-consuming step is the data selection and preprocessing step.
This is usually around 80% of the total time needed to build an analytical model.
Types of Analytics

- Analytics is a term that is often used interchangeably with:
  - Data science
  - Data mining
  - Knowledge discovery

- Predictive analytics
  - A target variable is typically available
    - E.g. linear/logistic regression, decision trees, neural networks, support vector machines

- Descriptive analytics
  - No target variable
    - E.g. Clustering, association rules

Types of Data Sources

- Transactions
  - Structured, low-level, detailed information
    - Customer transactions
      - Purchase, claim, cash transfer, credit card payment
    - Storied in massive online transaction processing (OLTP) relational database
      - Can be summarized over longer time horizons (e.g. averages, relative trends, Max/Min values)

- Unstructured data embedded in text documents
  - E.g. Emails, web pages, claim forms
    - Requires extensive preprocessing

- Qualitative, expert-based data
  - Requires subject matter expert’s (SME) analysis
    - Scientific data

Types of Data Consumers

- New types of data consumers
  - Gather data in a particular setting (credit risk, marketing)
  - Build models
  - Sell outputs
    - E.g. Dun & Bradstreet, Bureau Van Dijk, Thomson Reuters

Sampling

- Taking a subset of data for analytics
  - Generating hypothesis
  - Model selection
  - Feature selection
  - Speculative process
  - Building analytics model

- Stratified sampling
  - Taking samples according to predefined strata
    - E.g. Fraud detection with very skewed (99 percent non-fraud customers, 1 percent fraud customers) records
    - Sample should contain the same percentage of fraud customers as in the original data

Types of Data Elements

- Continuous
  - Data elements that are defined on an interval that can be limited or unlimited
    - E.g. Income, salary, temperature

- Categorical Nominal
  - Data elements that can only take on a limited set of values with no meaningful ordering between them
    - E.g. Marital status, profession, purpose of loan

- Ordinal
  - Data elements that can only take on a limited set of values with a meaningful ordering between them
    - E.g. Credit rating, age coded as young, middle age and old

- Binary
  - Data elements that can only take on two values
    - E.g. Having child, allowed to drive

Missing Values

- Missing values can occur because of various reasons
  - The information can be non-applicable
  - The information can be undisclosed
  - The information can be unavailable
Missing Values --continued

• **Replace (Impute)**
  - Replaces the missing values with a computed/selected value
  - Imputation algorithm examples
    - Hot deck: replaces with randomly selected similar records
    - Cold deck: replaces with values from another dataset
    - Mean substitution: replaces with the mean of that variable for all other cases
  - Regression: predicts missing values of a variable based on other variables.

• **Delete**
  - Deletes observations with lots of missing values
  - This assumes that information is missing at random and has no meaningful interpretation
  - Relationships to the target
  - Missing values can be meaningful
  - e.g., a customer did not disclose the income for current condition

Outliers of Dataset

• **Outliers** are extreme observations that are very dissimilar to the rest of the population
  - **Valid observation**
    - Salary of boss
  - **Invalid observation**
    - Age is 100

• **Multivariate outliers**
  - Observations that are outlying in multiple dimensions
  - e.g. Temperature in Fort Collins is 100 degrees but on a midnight in December

Identifying Outliers using Box Plots

• A box plot represents three key quartiles of the data
  - $Q_1$: 25% of the observations have a lower value
  - $Q_3$: 75% of the observations have a lower value
  - The minimum and maximum values are added

• **Too far away** is now quantified as more than $1.5 \times \text{Interquartile Range} (IQR = (Q_3 - Q_1))$

\[ \text{Outliers} = \text{values} > Q_3 + 1.5 \times IQR \]

Identifying Outliers using Z Score

• Measuring how many standard deviations an observation is away from the mean
  - $z_i = \frac{x_i - \mu}{\sigma}$
  - where $\mu$ represents the average of the variable and $\sigma$ its standard deviation

• A practical rule of thumb then defines outliers when the absolute value of the $z$-score $|z|$ is bigger than 3

\[ z_i > 3 \]

Dealing with Outliers

• **Treat outliers as missing values**

  • **Popular schemes**
    - Truncation: taking only values that are within the limits
    - Winsorizing: limiting extreme values to reduce the effect of possible spurious outliers
    - Recentering: adjusting the mean to better represent the rest of the distribution
    - The data below the 5th percentile and above 95th percentile are replaced with the neighboring values

\[ (0.05, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100) \text{ (N = 10, mean = 55.65)} \]

\[ (0.05, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100) \text{ (N = 20, mean = 101.5)} \]

Using the Z Scores for truncation
In a nutshell

Data Layer
Apache HDFS, Amazon AWS's S3, IBM GPFS, Microsoft Azure

Data Processing Layer
Apache Hadoop MapReduce, Pig, Apache Spark, Cassandra, Storm, Mahout, MLlib

Data Integration Layer
Apache Flume, Apache Kafka, Apache Sqoop

Operations and Scheduling Layer
Apache Oozie, Apache Airflow

Security and Governance
Apache Kibana, Apache Solr

This material is developed based on,

- Download this chapter from the CS435 schedule page


- MapReduce Design Patterns, Donald Miner and Adam Shook, O’Reilly, 2013
What is MapReduce?

MapReduce [1/2]

• **MapReduce** is inspired by the concepts of map and reduce in Lisp.

• **“Modern” MapReduce**
  - Developed within Google as a mechanism for processing large amounts of raw data.
  - Crawled documents or web request logs
  - Distributes these data across thousands of machines
  - Same computations are performed on each CPU with different dataset

MapReduce [2/2]

• **MapReduce** provides an abstraction that allows engineers to perform simple computations while hiding the details of parallelization, data distribution, load balancing and fault tolerance

Mapper

• **Mapper maps input key/value pairs to a set of intermediate key/value pairs**
  - Maps are the individual tasks that transform input records into intermediate records
  - The transformed intermediate records do not need to be of the same type as the input records
  - A given input pair may map to zero or many output pairs
  - The Hadoop MapReduce framework spawns one map task for each InputSplit generated by the InputFormat for the job

Reducer

• **Reducer reduces a set of intermediate values which share a key to a smaller set of values**

• **Reducer has 3 primary phases**
  - Shuffle, sort and reduce

• **Shuffle**
  - Input to the reducer is the sorted output of the mappers
  - The framework fetches the relevant partition of the output of all the mappers via HTTP

• **Sort**
  - The framework inputs to the reducer by keys

MapReduce Example 1
Example 1: WordCount

- For text files stored under /usr/joe/wordcount/input, count the number of occurrences of each word
- How do files and directory look?

```bash
bin/hadoop dfs -ls /usr/joe/wordcount/input
/bin/hadoop dfs -cat /usr/joe/wordcount/input/file01
Hello World, Bye World!
/bin/hadoop dfs -cat /usr/joe/wordcount/input/file02
Hello Hadoop, Goodbye to hadoop.
```

Example 1: WordCount

Run the MapReduce application

```bash
bin/hadoop jar /usr/joe/wordcount.jar org.myorg.WordCount /usr/joe/wordcount/input /usr/joe/wordcount/output
```

What do you have to pass from the Mappers?

```java
public static class Map extends Mapper<LongWritable, Text, Text, IntWritable> {
  private final static IntWritable one = new IntWritable(1);
  private Text word = new Text();
  public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
    String line = value.toString();
    StringTokenizer tokenizer = new StringTokenizer(line);
    while (tokenizer.hasMoreTokens()) {
      word.set(tokenizer.nextToken());
      context.write(word, one);
    }
  }
}
```

Questions?