CS435 BIG DATA

PART 1.
LARGE SCALE DATA ANALYSIS USING MAPREDUCE

Sangmi Lee Pallickara
Computer Science, Colorado State University
http://www.cs.colostate.edu/~cs435

Today’s topics
• FAQs
• Term Project
• MapReduce design pattern
  • Filtering

FAQs
• Term project guideline is available at:
  • http://www.cs.colostate.edu/~cs435/Assignments.html
• Programming Assignment 2 help session is available at:
  • http://www.cs.colostate.edu/~cs435/Assignments.html

Term project: Proposal
• Submit via Canvas
  • Team submission
  • 1,000 – 1,200 words
  • Document only
• Components of proposal
  • Title
  • Problem formulation
  • Your strategy to solve the problem
  • Your dataset
  • Timeline
  • Bibliograph

Title
• Title should be concise and self-descriptive

Problem formulation
• The proposal should clearly identify the problem
• It should include at least one or two carefully crafted paragraph that states and highlights the problem
• The problem formulation should be able to answer following questions:
  • What is the problem you are solving?
  • What is the goal of this project?
  • Why is your project important?
  • Why is it interesting as a Big Data problem and who would use it if it were solved?
Your strategy to solve the problem

- Describe your proposed approach to solve the problem
- The description of the strategy should include,
  - The algorithms/techniques/models you plan to use in this project.
  - Regressions
  - Classifications
  - Graph analysis
  - Machine learning (if you take (or took) ML course)
  - Correlation analysis
  - The framework you plan to use in this project.

- Please note that you are also required to produce software as the final output of this project.

Your dataset

- The proposal should include a dataset to use for your project. Please include the link to the dataset and description

Project timeline (weekly plan)

- You should provide a table with a weekly plan to complete the term project
- If you have teammate, the plan should also include information about the respective roles

Bibliography

- Included a bibliography
- All references must be cited (cross-referenced) in the report

Datasets: Web data

- Web data commons
  - http://webdatacommons.org
  - Hyperlink Graph
  - 3.5 billion web pages and 128 billion hyperlinks
  - Product data corpus
  - 5.6 million product records
  - What are the example project?
  - Web analysis
  - Using in-degree, out-degree, harmonic centrality, PageRank

Datasets: Enron Email data

- Enron Email dataset
  - 1,227,255 emails with 493,384 attachments
  - https://www.cs.cmu.edu/~./enron/
  - Possible project
    - Spam filter
Dataset: Network data

- SNAP
  - Stanford Large Network Dataset Collection
  - https://snap.stanford.edu/data/index.html
- Amazon network
- Twitter messages
- Memetracker messages
- Online community (Reddit, and Flicker)
- And more

Dataset: Kaggle

- https://www.kaggle.com/
- Read Kaggle’s questions
- Pros: Many sets of interesting data
- Cons: Size of the most of datasets is compact.
- Please check the data size and make sure it is a “Big Data” problem

Dataset: etc.

- Reddit comments
  - https://www.reddit.com/r/datasets/comments/3bxlg7/i_have_every_publicly_available_reddit_comment/
  - ~1.7 billion comments (250GB compressed)
- Wikipedia dump
  - https://meta.wikimedia.org/wiki/Data_dumps

If your team has 3 members,

- Your scope must be large enough to cover activities of 3 members
- How?
  - Select bigger topic
  - Or, consider components such as,
    - Visualization
    - User interface
    - Data collection
    - Data integration/fusion

Example projects

- Spring 2016
  - Combined Fuel Economy of Medium Duty Hybrid Trucks
  - Where to Live
  - Analysis of Reddit Comments for Word Popularity and Trends Using Hadoop
  - Trend Analysis to Improve Donations
  - Movie recommendation system based on movie attributes and human-move interaction information
  - Song Genre generation
  - Predicting Wind Speeds Based on Atmospheric Data
  - Darknet Market Survival and Similarity Analysis
  - Trends in Baby Names and Predicting Popular Names
  - Predicting PM2.5
  - Predicting Stock Similarity Using K-Means
  - Analyzing Trends on Twitter

- Spring 2015
  - Similarity and Clustering on The Million Song Dataset
  - Where to Live
  - Analysis on Methods Used to Analyze Eclipse and Mozilla Bug Data
  - Movie Recommendation using Collaborative Filtering
  - A Retrospective Study Of Change In The English Language Via Textual Analysis
  - 1929 - 2009 Climate Visualization and Predictive Analysis
  - Wikipedia Link analysis
  - Google book N gram analysis
  - Analysis of stock market trends
  - Hazardous Materials Source Mapping
  - Wikipedia Page Traffic Statistics Analysis
  - Million Song Dataset Geographical Analysis
  - Trending Topics and Page Count Prediction on Wikipedia Traffic Log Data
What is a “good” proposal?

- Your problem
  - Importance
  - Challenging
  - Feasibility
  - Also, your topic should be appropriate to the academic environment

- Your approach
  - Well structured
  - Your approach should lead you to “good results”
  - Then, what is the “good results”?
    - Interesting knowledge
    - Demonstrating validity of your approach
    - Providing estimable accuracy/performance

Validation/Evaluation techniques

1. True error estimate

How do you measure the accuracy?

- Root-Mean-Square-Error (RMSE)
  - Sample standard deviation of the differences between predicted values and observed values

- Mean-Square-Error (MSE)
  - Average of squares of the errors

Background: Why validation?

1. True error estimate

What if we used a model?

1. True error estimate
   What if we used a model?

- Process for model selection and performance estimation

- Model selection (fitting the model)
  - Most of the models have one or more free parameters
    \[ h(x) = \theta_0 + \theta_1 x \]
  - You should find these values to generate (fit) your model.

- How do we select the “optimal” parameter(s) or model for a given classification problem?
Performance (Accuracy) estimation
- Once we have chosen a model or analysis, how do we estimate its accuracy?
- Accuracy is typically measured by the **true error rate**
  - the classifier's error rate on the entire population

**Background:**

Why validation? (2/2)

- This problem is more pronounced with models that have a large number of parameters
  - The error rate estimate will be overly optimistic (lower than the true error rate)
    - In fact, it is not uncommon to have 100% correct classification on training data

- A much better approach is to **split the training data** into disjoint subsets: the holdout method

**Challenges** (1/2)

- If we had access to an unlimited number of examples these questions have a straightforward answer
  - Choose the model that provides the lowest error rate on the entire population
    - Of course, that error rate is the true error rate

- In real applications we only have access to a subset of examples, usually smaller than we wanted
  - What if we use the entire available data to fit our model and estimate the error rate?
    - The final model will normally overfit the training data
    - We already used the test dataset to train the data

**Validation/Evaluation techniques**

1. The Holdout method

**Drawbacks of the holdout method**

- For a sparse dataset, we may not be able to set aside a portion of the dataset for testing
- Based on the where “split” happens, the estimate of error can be misleading
  - Sample might not be representative

- The limitations of the holdout can be overcome with a family of resampling methods
  - More computational expense
  - Stratified sampling
  - Cross Validation
    - Random subsampling
    - K-Fold cross validation
    - Leave-one-out Cross-Validation
Validation/Evaluation techniques

2. Random Subsampling

- Random Subsampling
  - Create a dataset of the dataset
  - Each split randomly selects a (fixed) number of examples without replacement
  - For each data split, retain the classifier from scratch with the training examples and estimate $E_i$ with the test examples

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Test Example</th>
<th>Total Number of Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Experiment 1

Experiment 2

Experiment 3

Experiment 4

Total number of examples

True Error Estimate

- The true error estimate is obtained as the average of the separate estimates $E_i$
- This estimate is significantly better than the holdout estimate

$$ E = \frac{1}{K} \sum_{i=1}^{K} E_i $$

Validation/Evaluation techniques

3. k-Fold Cross-validation

- Create a k-fold partition of the dataset (randomly partitioned)
  - For each of the k experiments use k - 1 folds for training
  - The remaining one for testing

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Test Example</th>
<th>Total Number of Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Experiment 1

Experiment 2

Experiment 3

Experiment 4

Total number of examples

True error estimate

- k-fold cross validation is similar to random subsampling
- The advantage of k-Fold Cross validation
  - All the examples in the dataset are eventually used for both training and testing
  - The true error is estimated as the average error rate

$$ E = \frac{1}{K} \sum_{i=1}^{K} E_i $$
Validation/Evaluation techniques

4. Leave-one-out Cross-validation

**Leave-one-out Cross Validation**

- Leave-one-out is the degenerate case of \( k \)-Fold Cross validation
- \( k \) is chosen as the total number of examples
- For a dataset with \( N \) examples, perform \( N \) experiments

<table>
<thead>
<tr>
<th>Experiment 1</th>
<th>Experiment 2</th>
<th>...</th>
<th>Experiment N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Test example</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total number of examples

**True error estimate**

- The average error rate on test examples
  \[
  E = \frac{1}{N} \sum_{i=1}^{N} E_i
  \]
- Is it suitable for large data samples?
  - No
  - Very computing intensive

**How many folds are needed?**  

(1/2)

- With a large number of folds
  - The bias of the true error rate estimator will be small
  - The estimate will be very accurate
  - The computational time will be very large
  - Many experiments

- With a small number of folds
  - The number of experiments are low
  - Computation time is reduced
  - The variance of the estimator will be small
  - The bias of the estimator will be large

**How many folds are needed?**  

(2/2)

- The choice of the number of folds depends on the size of the dataset
  - For large datasets, even 3-Fold Cross Validation will be quite accurate
  - For very sparse datasets, you may have to consider leave-one-out
    - To get maximum number of experiments

- A common choice for \( k \)-Fold Cross Validation is \( k=10 \)

**Three-way data splits**

- If model selection and true error estimates are computed **simultaneously**
  - The data needs to be divided into three disjoint sets
  - Training set
    - E.g. to find the optimal weights
  - Validation set
    - A set of examples used to tune the parameters of a model
    - To find the "optimal" number of hidden units or determine a stopping point for the back propagation algorithm
  - Test set
    - Used only to assess the performance of a fully-trained model

- **After assessing the final model with the test set, you must not further tune the model**
2. Validating/Evaluating Classifiers

Plain Accuracy
- Classifier accuracy
  - General measure of classifier performance
  
  \[
  \text{Accuracy} = \frac{\text{Number of correct decisions made}}{\text{Total number of decision made}}
  \]

  \(1 - \text{error rate}\)

  - Pros
    - Very easy to measure
  
  - Cons
    - Cannot consider realistic cases

The Confusion Matrix
- A type of contingency table
  - \(n\) classes
    - \(n \times n\) matrix
    - The columns labeled with actual classes
    - The rows with predicted classes

  - Separates out the decisions made by the classifier
    - How one class is being confused for another
    - Different sorts of errors may be dealt with separately

<table>
<thead>
<tr>
<th></th>
<th>(p) (positive)</th>
<th>(n) (negative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Y) (predicted)</td>
<td>True positive</td>
<td>False positive</td>
</tr>
<tr>
<td>(N) (predicted)</td>
<td>False negative</td>
<td>True negative</td>
</tr>
</tbody>
</table>

Problems with Unbalanced Classes
- Consider a classification problem where one class is rare
  - Sifting through a large population of normal entities to find a relatively small number of unusual ones
  - Looking for defrauded customers, or defective parts
  - The class distribution is unbalanced or skewed

Why accuracy is misleading
- \(\text{True Population}\)
  - \(50%\) True positives
  - \(50%\) True negatives

- \(\text{Balanced Population}\)
  - \(50%\) True positives
  - \(50%\) True negatives

Which model is better?
F-measure ($F_1$ score)

- Summarizes confusion matrix
- True positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN)
- True positive rate (sensitivity) = $\frac{TP}{TP+FN}$
- False negative rate (miss rate) = $\frac{FN}{TP+FN}$
- $F$-measure = $2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$
  - precision = $\frac{TP}{TP+FP}$
  - recall = $\frac{TP}{TP+FN}$
- Accuracy = $\frac{(TP + TN)}{(P + N)}$

This material is built based on,
- MapReduce Design Patterns
- Building Effective Algorithms and Analytics for Hadoop and Other Systems
  - By Donald Miner, Adam Shook
  - November, 2012

Customizing input and output
- Do we always want to load or store data the way Hadoop MR does out of the box?
  - Injecting data from original source without storing data in HDFS
  - Feeding the MapReduce output to the next process

Patterns discussed in this section
1. Generating data
2. External source input
3. Partition pruning

Modify the way data is loaded on disk
- Approach 1: Configuring how contiguous chunks of input are generated from blocks in HDFS
  - InputFormat
- Approach 2: Configuring how records appear in the map phase
  - RecordReader
Modify the way data is stored on disk

- Approach 1: Configuring how contiguous chunks of output are generated from blocks in HDFS
  - OutputFormat

- Approach 2: Configuring how records are stored after the map phase
  - RecordWriter

Roles of InputFormat in Hadoop

1. Validate the input configuration for the job (i.e., checking that the data is there).
2. Split the input blocks and files into logical chunks of type InputSplit, each of which is assigned to a map task for processing.
3. Create the RecordReader implementation to be used to create key/value pairs from the raw InputSplit. These pairs are sent one by one to their mapper

Accessing your input file in MapReduce

- STEP 1. Validates the input for the job by checking whether all of the input paths exist
- STEP 2. Splits each input file logically based on the total size of the file in bytes
  - Block size is the upper bound
  - E.g. 160MB in HDFS will generate three blocks
    - 2 x 64MB and 1x38MB
- STEP 3. Each map task will be assigned exactly one of these input splits
- STEP 4. RecordReader will generate key/value pairs for Mapper input

Methods of the InputFormat abstract

- getSplits()
  - Retrieves the configured input using the JobContext object
  - Returns a list of InputSplit objects
  - getLocation() of InputSplit returns the list of hostnames where the input split is located
  - This provides due to the system to determine where to process the map task
  - Good place to throw any necessary exceptions

- createRecordReader()
  - Called by framework and generates RecordReader

RecordReader (1/2)

- Generates key/value pairs
- Fixing boundaries
  - Input split boundary might not exactly match the record boundary
  - Eg. TextInputFormat reads text files using a LineRecordReader to create key/value pairs
  - Will the chunk of bytes for each input split be lined up with a new line character, to mark the line for the LineRecordReader?
  - Those bits that are stored on a different node are streamed from a DataNode hosting the block
    - Handled by the DataInputStream class

RecordReader (2/2)

- Reads Bytes from the input source
- Generates WritableComperable key and Writable value
  - An object-oriented way to present information to a mapper

- Example
  - TextInputFormat grabs each line
    - "<?xml version="1.0"?>" and "<quiz>" will be injected to the different Mappers
  - Customized RecordReader can read lines after the input split boundary
    - Each RecordReader should start at the beginning of an XML element