Lecture 7:
Dealing with Missing Data

COSC 526: Introduction to Data Mining

Spring 2020

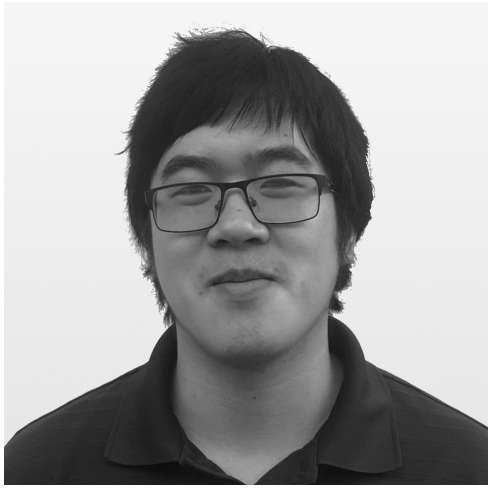
THE UNIVERSITY OF
TENNESSEE
KNOXVILLE
KNOXVILLE

BIG ORANGE. BIG IDEAS.®

Instructors:

Michela Taufer

GRA:



Nigel Tan

Today we will explore …

- Reasons for missing data
- Strategies for dealing with datasets with missing data
- Apply the strategies to real dietary data

Example of Projects

Project: Let's start ..

- Project: Poster + extended abstract (2 pages)
- Examples:
  - Dylan's project was on a dataset from United States Federal Railroad Administration Office of Safety Analysis
  - Mike's project was on National Health and Nutrition Examination Survey (NHANES) dataset

Leveraging Spark and Docker for Scalable, Reproducible Analysis of Railroad Defects

Dylan Chapp and Surya Kasturi
Advisors: Michela Taufer and Nii Attoh-Okine

Motivation

- Railroad network resiliency depends identification of defects
- Sensor-equipped monitoring cars collect rail and track-geometry data
- Can we use the data to predict defect occurrences in rail subdivisions?

Rail and Track Defects Data Sets

- Rail defects data: physical degradation
  - 26,432 20-dimensional data points
- Track geometry data: misalignment
  - 25,421 41-dimensional data points
- Mixed numerical and categorical data

Scalable Reproducible Data Analysis Workflow

Track Defect Data

Rail Defect Data

Raw data provided by collaborators

Analysis Platform: PySpark + MLLib

Preprocessing
- Parallelized with Spark, scales to larger datasets
- Extracts logically related subsets of data
- Filters out records with missing data
- Filters out irrelevant feature columns

Targeted Subset of Data

Data is now ready to be ingested by an analysis tool, or directly visualized

MLLib Analysis Tools
- Descriptive Statistics
- Classification
- Clustering

Visualization
- Generate plots specific to analysis technique

Docker Container

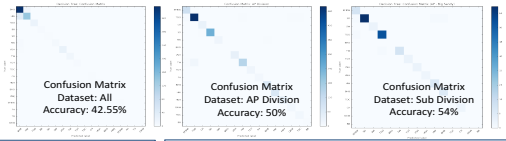
Input Data

Analysis Code

- Package data, scripts, configs as a Docker container
- Collaborators can easily reproduce our analysis
- Hosted on DockerHub: https://hub.docker.com/r/dchapp/rail

Predicting Defect Types

Can we predict defects in railroad tracks?
- Defects are classified using Decision Tree
- Defect size, accumulated tonnage, rail weight, rail section age are used as features



Defect Super Classes

T – Class: TDD, TDC, TD, TDT

B – Class: BRO, BHB, BB

W – Class: HW, HWJ, OAW, SW

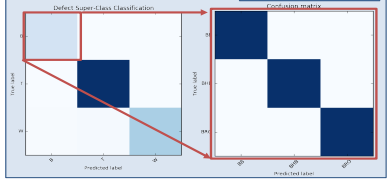
Defect Super Class T, B, W Classification

Pipeline

Defects are classified using super-class before predicting their type

Using the subset containing T, B, W defects resulted in accuracy of 89.90%

- Defect TDD, TD, TDT Classification
- Defect BRO, BHB, BB Classification
- Defect HW, HWJ, OAW, SW Classification

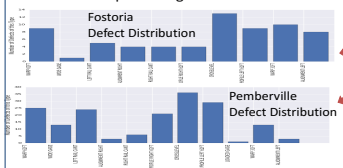


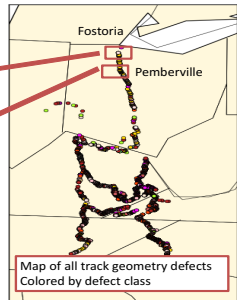
Conclusions:
- We improve prediction accuracy by a hierarchical classification scheme
- First decide membership in defect superclass, then in defect class using a second classifier

Track Region Similarity Analysis

Can track regions be grouped so that defect-type classifiers trained on region-specific data achieve better accuracy?

1: Extract each pair of regions' defect distributions

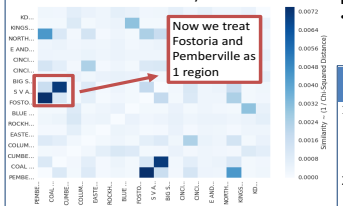




2: Compute the Chi-Squared Similarity of each pair of defect distributions

$$d(\mathbf{x_u}, \mathbf{x_v}) = \frac{1}{2}\sum_{n=1}^{N}\frac{[x_u(n) - x_v(n)]^2}{x_u(n) + x_v(n)}.$$

3: Group regions together whose defect distributions are sufficiently similar



Heatmap of regions' defect similarities

In progress:
- Training classifiers on subsets of defect data from statistically similar regions

References

1. A. Zarembski, "Some Examples of Big Data in Railroad Engineering", IEEE International Conference on Big Data, 2014
2. Track Inspector Rail Defect Reference Manual, Federal Railroad Administration, Rev. 2, 2015

Acknowledgements / funding

BIG ORANGE BIG IDEAS

Leveraging Spark and Docker for Scalable, Reproducible Analysis of Railroad Defects

Dylan Chapp
University of Delaware
dchapp@udel.edu

Surya Kasturi
University of Delaware
suryak@udel.edu

ABSTRACT

The resiliency of railroad networks depends on the ability of railroad engineers to identify and mitigate track and rail defects. As railroads modernize their defect identification measures, the volume and velocity of defect data substantially increases, necessitating adoption of techniques for "Big Data" analytics. We present a study of railroad defect prediction built atop Apache Spark and Docker to achieve scalability and reproducibility.

1. INTRODUCTION

According to the United States Federal Railroad Administration Office of Safety Analysis, track defects are the second leading cause of accidents on railways in the United States. In light of the economic significance of railway accidents [1], there is a pressing need in the railroad engineering community to adopt data-driven scalable data analysis tools from the greater "Big Data" ecosystem. [3] Track maintenance–i.e., identifying and repairing defects–is one of the primary factors that affect the service life of a rail track, but due to the severe safety implications of undetected or unprepaired defects, the ability to predict common defects is highly desirable.

In this work, we present a case study centered on the analysis of two railroad defect data sets obtained from railroad engineering researchers in the University of Delaware Department of Civil Engineering. Hereafter we will refer to these datasets as the `rail_defects` data set and the `track_geometry_defects` data set. Respectively, these data sets describe defects in the rails themselves, such as voids or internal changes in crystalline structure, and misalignment of track components, such as one rail tilting away from the other. [3] We investigate the feasibility of predicting the type of a defect based on associated data such as geographic region, mean gross tonnage (MGT) the track is subject to, and rail type. In the rest of this paper, we outline the construction of our analysis platform, present some initial results on classification accuracy, and propose extensions to our work.

© 2016 ACM. ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

2. METHODOLOGY

Both of the data sets we target have mixed categorical and numerical features and $> 99\%$ of the individual defect records have a class label indicating the type of defect. In the case of `rail_defects`, there are 20 distinct defect types. For `track_geometry_defects`, there are 25 defect types. In light of these properties, we focus on the multilabel classification task for each data set. We decompose the task into a pipeline of three parts: preprocessing, training, and testing. We implement this pipeline using the MapReduce framework Apacahe Spark [2] and its parallel machine learning library MLLib, and package the data and analysis scripts as a Docker container for ease of dissemination. In the remainder of this section, we describe the pipeline components, also display in Figure 1

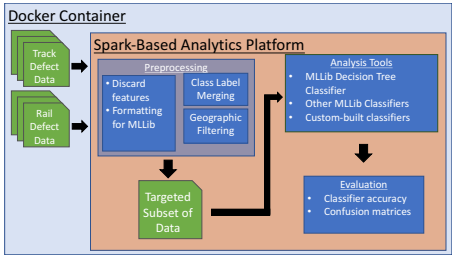


Figure 1: Block Diagram of Analytics Platform

We consider two stages of mandatory preprocessing. The first stage discards all columns except for a specified set, then discards any rows that are missing values for features from that set. The second stage maps the raw record strings to the format the MLLib API specifies, a key-value pair whose key is the type of defect and whose value is a feature vector.In addition to the above preprocessing, we implemented two optional stages: one to restrict the data to a geographically coherent region, and another to map each data point's class label to a "super- class" label indicating the general kind of defect (e.g., a welding-related defect, rather than one of the five kinds of specific welding defects). In our evaluation section, we demonstrate the usefulness of these additional preprocessing stages.

To build and evaluate our classifier, we split the subset of data remaining after preprocessing into training and testing sets consisting of, respectively, 70% and 30% of the original data. Membership in the training and testing sets is determined by uniform random sampling. We then train an instance of MLLib's decision tree classifier on the training set and test its predictions. In principle, any other MLLib classifier with a compatible API could be trained instead, but we elected to keep our classifier type fixed and investigate the effect of the "class-merging" and "geographic filtering" preprocessing steps on accuracy.

Table 1: Rail Defects Mapping

Super Class	Defect Types
T	TDD, TDC, TD, TDT
B	BRO, BHB, BB
W	HW, HWJ, OAW, SW
Others	SD, VSH, HSH, TW, CH, FH, PIPE, DR, EFBW

3. EVALUATION

To evaluate our classifier's performance, we examine the overall accuracy rate of the classifier and its associated confusion matrix. When we trained the classifier on training data drawn uniformly at random from `rail_defects` dataset with each defect type as a class label, the classifier predicted with an accuracy of 42.55%.

3.1 Class Label Merging

We propose mapping each data point's class label to a "super-class" label indicating its general kind. Out of 20 defect types in `rail_defects` dataset, 11 are mapped to 3 three "super-classes". Table 3.1 shows mapped and unmapped defect types.

With this mapping, we show that the prediction accuracy of rail defects is improved using a hierarchical classification scheme . First a classifier is trained to decide super-class of data point, then a second classifier is used to predict its defect type. When this model applied on the training data, the classifier predicted with an accuracy of 89.90%. Figure 2 shows the confusion matrix of the respective result.

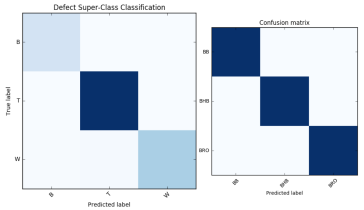


Figure 2: Confusion Matrix of the hierarchical classification scheme

3.2 Geographic Filtering

We propose that if subdivisions have similar numbers of each kind of defect, then we should group these subdivisions' data points and train a classifier with the expressed purpose of achieving good accuracy for that set of subdivisions. To determine which subdivisions to merge, we propose computing the χ-squared distance S defined below for each pair of subdivisions, then merge them based on a fixed threshold.

$$S(D_1, D_2) = \sum_{i=0}^{N} \frac{(x_i - y_i)^2}{(x_i + y_i}$$

We demonstrate the potential of grouping based on defect type distributions below. We compute $S(x, y)$ for each pair of subdivisions within the Appalachian division and display the results in the heat map in Figure 3.2

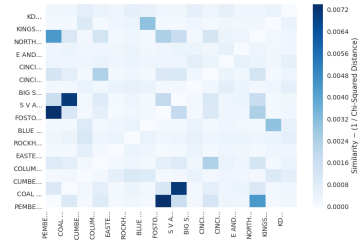


Figure 3: χ-squared distance between defect distributions for each subdivision

4. CONCLUSIONS AND CONTINUING WORK

We identified the potential of a hierarchical classification stage to improve the accuracy of defect type predictions. Additionally, we determined while that merely training classifiers on data from geographically-similar regions does not yield a significant improvement in accuracy, attempting to group together regions whose defect distributions are similar may prove useful.

Future directions for this work include evaluating classifiers beyond decision trees, and refining the similarity metric on defect distributions we use to group regions.

5. REFERENCES

[1] D. H. Schafer. A prediction model for broken rails and an analysis of their economic impact. *2008 AREMA Conference*, 2008.

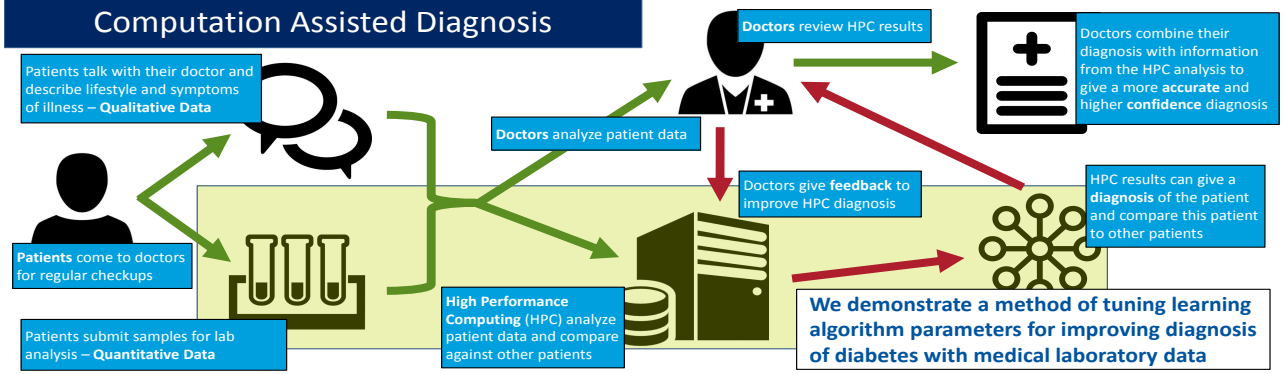
[2] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, NSDI'12, pages 2–2, Berkeley, CA, USA, 2012. USENIX Association.

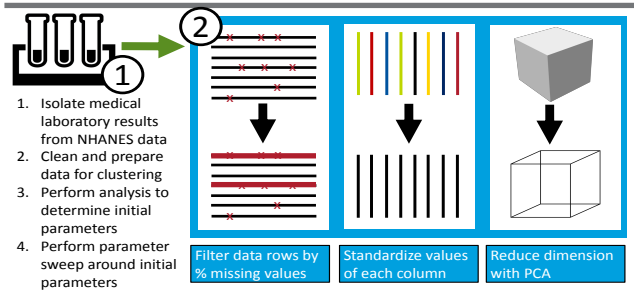
[3] A. M. Zarembski. Some examples of big data in railroad engineering. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 96–102, Oct 2014.

Parameter Tuning of DBSCAN for Medical Data and Diabetes Diagnosis

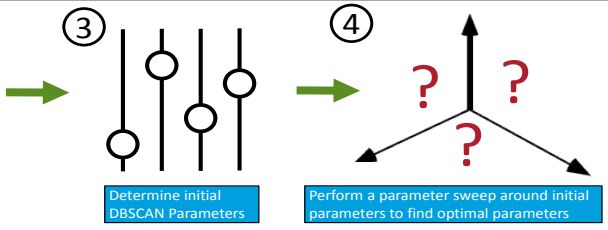
Michael Wyatt, Michela Taufer
University of Delaware: Global Computing Lab

Computation Assisted Diagnosis





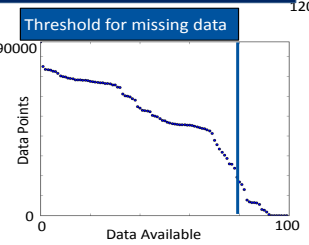
Parameter Tuning Workflow

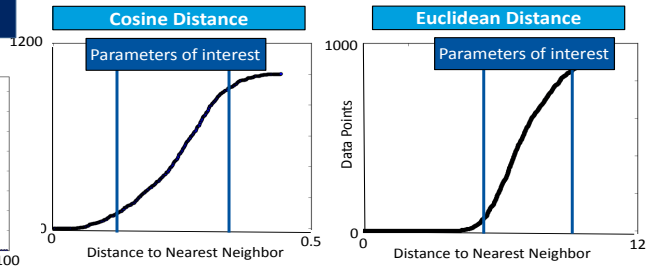


Parameter Tuning Results

- Missing data **trade-off**
  - No Missing Values → **Small Dataset**
  - Many Missing Values → **Bad Clusters**

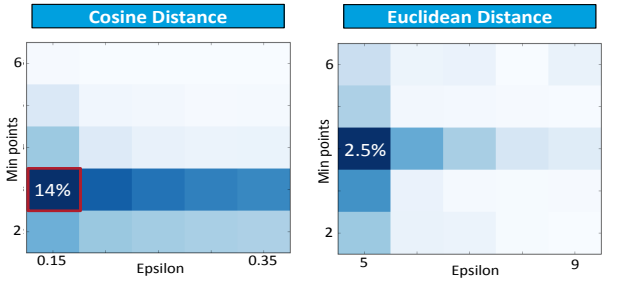
  Must find a balance between missing values and dataset size
- Patients with > 80% lab data available
  - 16,627 patients (over 1/5 original data)
  - Produces higher quality clusters





- DBSCAN parameters affect cluster quality
  - **Epsilon**: Neighborhood to search for neighbors
  - **Min_pts**: minimum neighbors to be in a cluster

  } Parameters define cluster **density**
- Distance metrics also affect cluster quality: **Euclidean** vs. **Cosine**



- Utilizing nearest neighbor analysis, we can determine the range of epsilon values which should be tested
- We cluster data with several **epsilon** and **min_pts** values around the identified optimal values
- We measure the quality of each clustering by **percentage of points clustered** and **information gained** by each clustering:

$$score = \frac{Diebetes\ Patients\ Clustered}{Total\ Diabetes\ Patients} * \frac{Informaion\ Gain}{Max\ Information\ gain}$$

- We identify an optimal parameter setting:
  - **Cosine distance, Epsilon: 0.15, Min_pts: 3**

BIG ORANGE BIG IDEAS

Parameter Tuning of DBSCAN for Medical Data and Diabetes Diagnosis

[Extended Abstract]

Michael R. Wyatt II
University of Delaware
18 Amstel Ave
Newark, DE 19701
mwyatt@udel.edu

Michela Taufer
University of Delaware
18 Amstel Ave
Newark, DE 19701
taufer@udel.edu

ABSTRACT

The increasing use of computationally assisted diagnosis in the doctor's office requires that computer diagnosis be both fast and accurate. We present a scalable method for preparing laboratory data for use with learning algorithms and a method for identifying optimal parameter settings for learning algorithms. To demonstrate our method, we predict the presence of diabetes among participants of the National Health and Nutrition Examination Survey using collected laboratory data and the DBSCAN algorithm. We performed optimization of the DBSCAN parameters for this dataset to demonstrate how diagnosis predictions can be improved.

CCS Concepts

•**Applied computing** → **Health care information systems;** *Consumer health;* •**Computing methodologies** → *MapReduce algorithms;*

Keywords

Health Informatics; Machine Learning; Optimization

1. MOTIVATION

Modern medical diagnosis is becoming increasingly computationally assisted. This means that artificial intelligence and machine learning algorithms are being used to analyze patient medical data and provide a diagnosis. Human doctors consult the computation results and a final diagnosis is made. As computationally assisted diagnosis becomes more widespread, patient diagnosis becomes more accurate [1] [3]. An important aspect of computationally assisted diagnosis is the processing of medical data by learning algorithms to produce useful results. Improving the speed and accuracy of this process will encourage the continued adoption of computationally assisted diagnosis by medical doctors, which will lead to improved population health and disease management.

2. CONTRIBUTIONS

Many efforts have been made to improve both the accuracy and processing time of computationally assisted diagnosis. These efforts focus mainly on the application of different algorithms to medical data. In this paper, we apply the clustering algorithm DBSCAN to medical data in order to diagnose patients with diabetes. We present a framework for parallel processing of medical data for learning algorithms. Additionally, we outline a method for optimizing learning algorithm parameters to achieve optimal results.

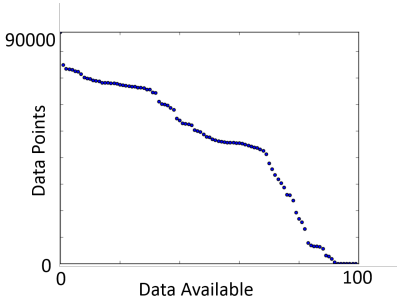


Figure 1: NHANES participants sorted by percent of laboratory data available.

3. METHODOLOGY

We isolate lab data from the National Health and Nutrition Examination Survey (NHANES) dataset for over 70,000 participants. We process this data using parallel algorithms built with the MapReduce programming paradigm via Apache Spark. The processing of data is highly scalable across many nodes. The processed data is in a form that is usable by learning algorithms like DBSCAN. We then perform parameter optimization to achieve maximum predictive capabilities.

3.1 NHANES Dataset

We obtained data from the NHANES continuous dataset collected between 1999 and 2014. We label the participants as having or not having diabetes based on their categorical response to the question, "have you ever been told by a doctor or health professional that you have diabetes or sugar diabetes?" We isolate 116 common features within the laboratory data, including urine and blood sample values, which can be used to cluster the diabetic and non-diabetic NHANES participants.

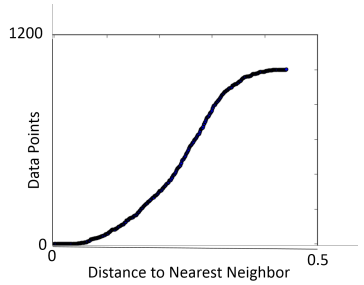


Figure 2: Number of data points (y axis) with a neighbor within distance *Epsilon* (x axis). The elbow in the plot indicates optimal values for *Epsilon*.

3.2 Data Preparation

We prepare the data by removing participants with many missing values, standardizing the data by feature, and dimensionality reduction. Figure 1 plots the participants sorted by the percent of data points (from the 116 features) available. All NHANES participants had missing values and many had most laboratory values missing. We identify a subset of 16,627 participants with more than 80% of features available for analysis. Each of the 116 features are standardized using Z-score standardization. This process makes the range of values for each feature similar to prevent one feature outweighing others (due to a large range of values). Dimensionality reduction is performed with Principal Component Analysis (PCA). The processing of NHANES data is performed with MapReduce algorithms. This allows the process to be distributed across many compute nodes and reduces result turn-around time.

3.3 Choosing Initial Parameters

There are three parameters for DBSCAN. Together, they define the density of clusters which will be found.

1. *Distance* - Metric used for calculating distance between patients
2. *Epsilon* - Distance around patients to identify neighboring patients
3. *Min_pts* - Number of neighboring patients to be "core" point

Like other learning algorithms, these parameters affect the quality of results. We chose to test two distance metrics: Euclidean and Cosine. We define cosine distance in equation 1. A range of *Min_pts* values was selected for testing. We then determined values for *Epsilon* by adapting the "elbow method" used for determining the best value of k in k-Means clustering [2]. Figure 2 shows the sorted Cosine distance to the nearest neighbor for each participant. We propose that ideal values for *Epsilon* will be around the elbow of this figure. In the case of figure 2, this range is $[0.15, 0.35]$.

$$Cosine\_distance(x, y) = 1 - \frac{x \cdot y}{\|x\|\|y\|} \quad (1)$$

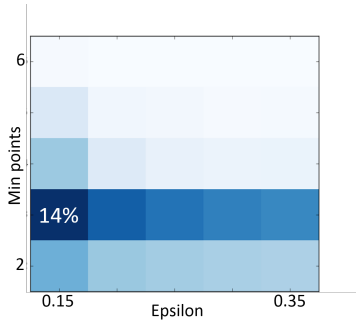


Figure 3: Information gain scoring method for Cosine distance metric across several *Epsilon* and *Min_pts* values.

3.4 Evaluation

To evaluate DBSCAN performance, we developed a scoring metric based on information gain. This scoring metric, seen in equation 2, considers the amount of information gain and size of each cluster in order to produce a value between 0 and 1. We scored each set of DBSCAN parameters and compared scores to find the best settings.

$$score = \frac{ClusteredPatients}{TotalPatients} \cdot \frac{InformationGain}{MaxInformationGain} \quad (2)$$

4. RESULTS

We observed that Cosine distance scored much higher than Euclidean distance. Figure 3 shows the scores of our parameter sweep around initial parameter selections for the Cosine distance metric. There is a clear set of parameters at $Epsilon = 0.15$ and $Min\_pts = 3$ which provides the best clustering score. The maximum observed score from our testing was 14%. Observed scores from our clustering were unexpectedly low. Despite the poor ability of DBSCAN to differentiate between patients with and without diabetes, our method of focused parameter searching was able to find optimal parameter settings.

5. REFERENCES

[1] K. Doi. Computer-aided diagnosis in medical imaging: historical review, current status and future potential. *Computerized medical imaging and graphics*, 31(4):198–211, 2007.

[2] T. M. Kodinariya and P. R. Makwana. Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95, 2013.

[3] I. Kononenko. Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine*, 23(1):89–109, 2001.

Project: Extended Abstracts

- Dylan Chapp and Surya Kasturi. Leveraging Spark and Docker for Scalable, Reproducible Analysis of Railroad Defects
- Stephen Herbein and Sean McAllister. Clustering Temporal Gene Expressions of Iron-Oxidizing Zetaproteobacteria
- Moumita Bhattacharya and Debarati Roychowdhury. Using Machine Learning to Build a Scalable Tool to support Dietitians to Fight Chronic Diseases

Project: Extended Abstracts

- Paul Soper Validation of the Short Time-series Expression Miner (STEM) on Iron Cycling in a Shallow Alluvial Aquifer
- Michael Wyatt. Parameter Tuning of DBSCAN for Medical Data and Diabetes Diagnosis

Project: Analysis of Abstracts

- Read the extended abstracts
- Identify:
  - Type of data
  - Question(s) answered
  - Methods used
  - One or two key outcomes
- Build a **general structure** that is recurrent across the **papers**
  - List your findings

Project: Analysis of Posters

- Read the posters
- Identify:
  - Type of data
  - Question(s) answered
  - Methods used
  - One or two key outcomes
- Build a **general structure** that is recurrent across the **posters**
  - List your findings

Project: Abstract vs. Poster

- What are the common components of both abstracts and posters?
- What are the key differences?
- What is the role of pictures in abstracts and posters?
- What is the role of text in abstracts and posters?

Project: Report your Findings

- Write a short report to summarize your findings (no more than 2 pages)
- Use one of the three temple for your report
  - IEEE latex or doc
  - ACM latex

Deadline: Feb 21, 2020

How to submit: Print and bring your report with you in class (it will be collected)

Missing Data: what strategy?

Data Collection

Missing data = incomplete observations

- Critical data issues:
  - Reasons for missing data
  - Scale and distribution of the values in the data

Case Study

- Study of an asthma education intervention in eight schools
- Randomly chosen set of students aged 8 to 14 with asthma
- Observations over two weeks period post-treatment
- Students complete:
  - Scale to measure self-efficacy beliefs with regard to their asthma
  - Questionnaire rating severity of their symptoms

(Velsor-Friedrich, see attached paper in GitHub)

Case Study

- Students simply forgot to visit the school clinic to fill out the form → Missing completely at random (MCAR)
  - Complete cases are representative of the originally sample
- Students missed school because of severity of their asthma symptoms and failed to complete the symptom severity rating
  - Missing variable is directly related to study
  - Example of non-ignorable missing data!!!
- Younger children missed ratings of symptom severity because they had a harder time interpreting the rating form → missing at random (MAR)
  - Values are missing for reasons related to another observable variable

Mechanisms to Deal with Missing Data

- Data are MCAR or MAR
  - Ignore the reasons for missing data in the analysis of the data
  - Simplify the model-based methods used for missing data analysis
- Use more than one method for collecting important information
  - E.g., income + years of education or type of employment

Table 1. Variable Descriptions.

Variable	Definition	Possible values	*M*	(*SD*)	*N*
Asthma belief Survey	Level of confidence in controlling asthma	Range from 1, little confidence to 5, lots of confidence	4.057	(0.713)	154
Group	Treatment or control group	0 = Treatment 1 = Control	0.558	(0.498)	154
Symsev	Severity of asthma symptoms in 2 week period post-treatment	0 = no symptoms 1 = mild symptoms 2 = moderate symptoms 3 = severe symptoms	0.235	(0.370)	141
Reading	Standardized state reading test score	Grade equivalent scores, ranging from 1.10 to 8.10	3.443	(1.636)	79
Age	Age of child in years	Range from 8 to 14	10.586	(1.605)	152
Gender	Gender of child	0 = Male 1 = Female	0.442	(0.498)	154
Allergy	Number of allergies reported	Range from 0 to 7	2.783	(1.919)	83

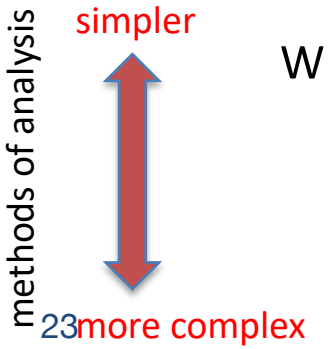
Table 1. Variable Descriptions.

POPULATION: 154

Variable	Definition	Possible values	*M*	(*SD*)	*N*
Asthma belief Survey	Level of confidence in controlling asthma	Range from 1, little confidence to 5, lots of confidence	4.057	(0.713)	154
Group	Treatment or control group	0 = Treatment 1 = Control	0.558	(0.498)	154
Symsev	Severity of asthma symptoms in 2 week period post-treatment	0 = no symptoms 1 = mild symptoms 2 = moderate symptoms 3 = severe symptoms	0.235	(0.370)	141
Reading	Standardized state reading test score	Grade equivalent scores, ranging from 1.10 to 8.10	3.443	(1.636)	79
Age	Age of child in years	Range from 8 to 14	10.586	(1.605)	152
Gender	Gender of child	0 = Male 1 = Female	0.442	(0.498)	154
Allergy	Number of allergies reported	Range from 0 to 7	2.783	(1.919)	83

Table 2. Missing Data Patterns.

Symsev	Reading	Age	Allergy	# of cases	% of cases
O	O	O	O	19	12.3
M	O	O	O	1	0.6
O	M	O	O	54	35.1
O	O	O	M	56	36.4
M	M	O	O	9	5.8
M	O	O	M	1	0.6
O	M	O	M	10	6.5
O	O	M	M	2	1.3
M	M	O	M	2	1.3
# missing	# missing	# missing	# missing	154	
13 (8.4%)	75 (48.7%)	2 (1.3%)	71 (46.1)		



What is the reasons for the missing data?

Can we accept the MCAR assumption?

Can we accept the MAR assumption?

Does missing data result from a non-ignorable response mechanism?

Commonly-Used Missing Data Methods

- Complete-Case Analysis: Cases that are missing variables in the proposed model are dropped from the analysis, leaving only complete cases
  - Assume that missing data are MCAR
  - Adequate amount of data remains for the analysis?

24

Commonly-Used Missing Data Methods

- Available Case Analysis: with X1 complete and X2 partially complete, all cases are used to estimate the mean of X1, but only the complete cases contribute to an estimate of X2, and the correlation between X1 and X2.
  - Different sets of cases are used to estimate parameters of interest in the data

Strategy 1

X1 = x11 x12 x13 x14 x15 → work on a population of 5 individuals

X2 = x22 x34 x25 → work on a population of 3 individuals

Strategy 2

X1 = x12 x14 x15 → work on a population of 3 individuals

X2 = x22 x34 x25 → work on a population of 3 individuals

25

Commonly-Used Missing Data Methods

- Single-Value Imputation: Fill in the missing value with a plausible one, e.g., mean for cases that observe the variable
  - Analyst continues with the statistical method as if the data are completely observed
  - Single value changes the distribution of that variable by decreasing the variance that is likely present
  - Bias in the estimation of variances and standard errors are compounded

Strategy 3

X1 = x11 x12 x13 x14 x15 → work on a population of 5 individuals

X2 = x_{avg} x22 x_{avg} x34 x25 → work on a population of 5 individuals

Model-Based Methods

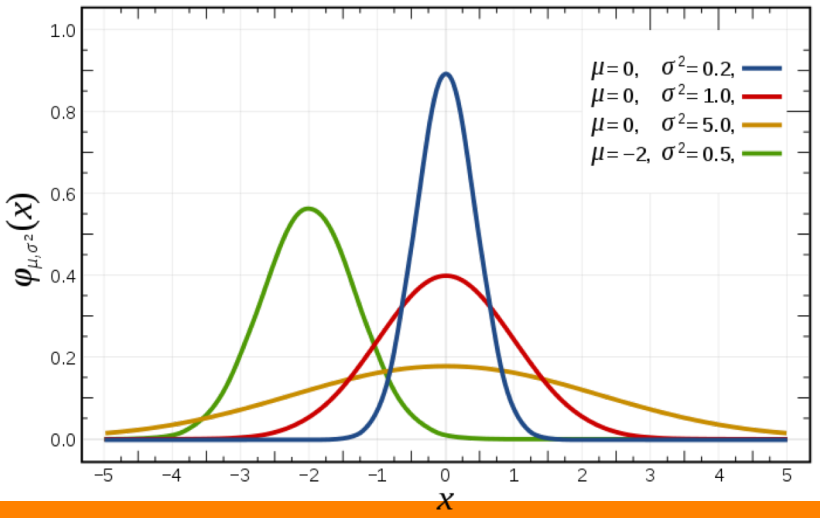
- Add assumptions about the distribution of the data and the nature of the missing data mechanism
  - Multiple imputation

Strategy 4

X1 = x11 x12 x13 x14 x15 → work on a population of 5 individuals

X2 = x_{21} x22 x_{23} x34 x25 → work on a population of 5 individuals

Assumption:



Assignment

Relevant Open-source Dataset

- Use a dataset with well-known and broadly used data format

NHANES: National Health and Nutrition Examination Survey

  - Medical, demographic, and dietary records
  - Available to the public for free
  - *Contains subjective food groups provided by USDA*

Data available at: http://www.cdc.gov/nchs/nhanes/nhanes_questionnaires.htm

NHANES Dietary Data

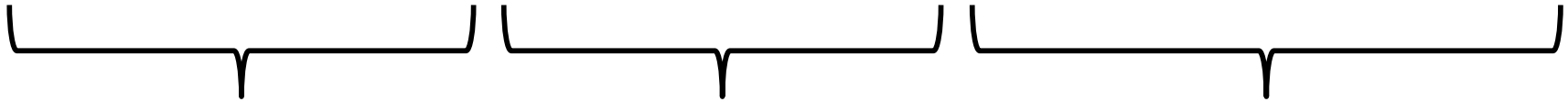
- Dietary intake of 64,653 Americans
- 7,494 unique food items
- 1,587,750 food entries
- 46 nutrient features for each food item
  - Macronutrients (e.g., fats, carbohydrates)
  - Micronutrients (e.g., vitamins, minerals)

NHANES Dietary Data

- Dietary intake of 64,653 Americans
- **7,462 unique food items**
- 1,587,750 food entries
- 46 nutrient features for each food item
  - **Macronutrients (e.g., fats, carbohydrates, proteins)**
  - Micronutrients (e.g., vitamins, minerals)

Structure of Dietary Data Item

<143672, 92510000, 3, 0, 8:15am, 7, 3, 10.1, 4, 3.45, 10, 178, …>



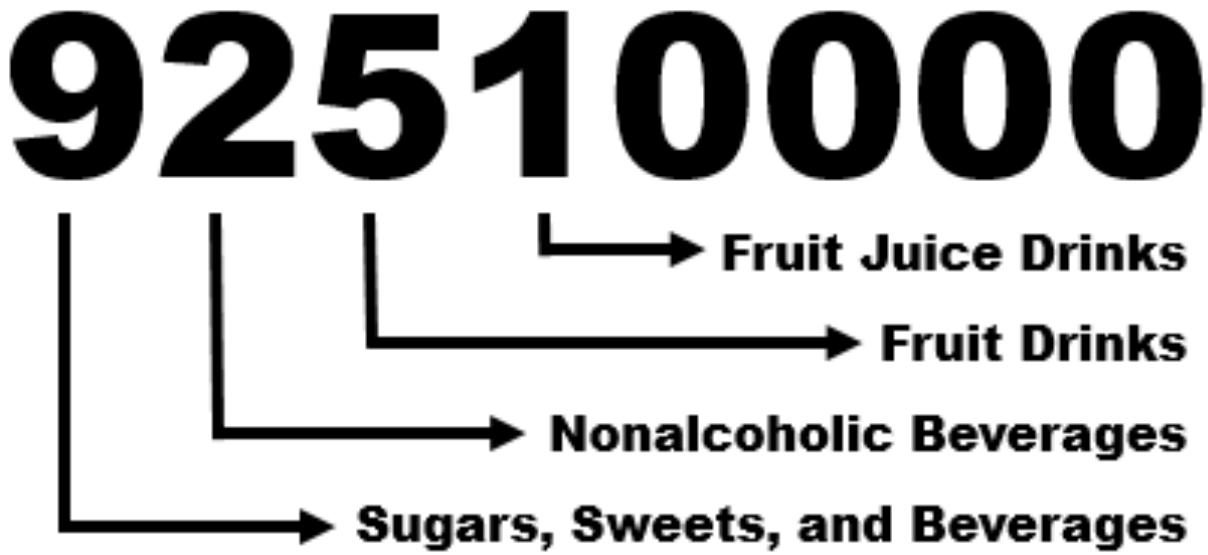
- Participant ID
- **USDA Food Code**

- Meta Data

- **Macronutrients**
- Micronutrients

USDA Food Classification

- Subjective and general
- Categorical, not nutrient-driven



Assignment 7 – Problem 1

- Data file: ./data/data-1.csv
  - Contain no missing values
- Problem: Cluster food items in the file based on carbohydrate and fat:
  - *Use the k-Means from the Spark MLlib to cluster data points*
  - *Determine the optimal value for k using the* **elbow method**
- Metric of quality: Use Within Set Sum of Squared Errors
  - This method is built into the Spark kMeans model and can be accessed with model.computeCost()
- Note: the clusters provide a ground truth for comparison to clusters we find later using data with missing values.

Assignment 7 – Problem 1

Steps:

- Define the optimal value for K
- Cluster food items (using K)
- Plot clusters by *fat* and *carbohydrate* content

Assignment 7 – Problem 2

- Data file: ./data/data-2.csv
  - Missing values for the carbohydrate content of some food items
  - The data were removed from a specific set of food items (i.e., food items with carbohydrate value near 0.5)
- Define a method to remove food items with missing any macronutrient values and apply this method to the data.
- Cluster the modified data and plot the results
  - Use the same K value as in Problem 1
- Note: we provide you with the code that loads the data and reports the percentage of values missing for each macronutrient (i.e., carbohydrates and fat)

Assignment 7 – Problem 3

- Data file: ./data/data-3.csv
  - Missing values for the fat or carbohydrate content of some food items (but not both for a single food item).
  - The data were removed from a food items randomly

PART 1:

- Define and apply a method to fill missing values with the mean of other values
  - *E.g., for missing values in fat, fill with the mean of fat values that are present*
- *Cluster the modified data and plot the results*
  - Use the same K value as in Problem 1

Assignment 7 – Problem 3

PART 2:

- *Use the code for Problem 2 to remove data with missing values rather than filling the gaps (as you did in PART 1)*
- *Cluster the modified data and plot the results*
  - Use the same K value as in Problem 1

Assignment 7 – Problem 4

- **Observe and describe:** Can you summarize your findings in each problem? Can you compare and contrast the findings across problems? How did each method for dealing with missing data (i.e., remove of filling) change the clustering outcome?
- **Impact of K:** What value did you choose for K in Problems 1-3? You based the selection of your K on the first dataset (i.e., no missing data). Do you expect a different value of K if you had used the elbow method with the second or third dataset? If yes, propose changes to your current solutions.

Assignment 7 – Problem 4

- **Building assumptions on data distributions:** Now look at the plot of clusters in Problem 1. Logically, there cannot be more than 1 gram of (carbohydrate + fat) in 1 gram of food. In your plot this can be seen in the form of a diagonal line from the top-left to bottom-right (where the sum of fat and carbohydrate content is equal to 1). How can you use this information to improve the way you fill missing values? Can you think of other methods to fill missing values? (HINT: logistic regression)

Example of Datasets

NHANES Dataset

NHANES Dataset

- National Health and Nutrition Examination Survey (NHANES) is a cross-sectional survey that is conducted every two years in the United States
- Individuals are asked to complete
  - demographics questionnaire
  - 2-day, 24 hour dietary recall data for all individuals
- Sampled population in NHANES for the years 1999 thru 2016.
  - In the early years (1999 and 2001) only 1 day of dietary information was collected
  - For years 2003 thru 2016, 2 days of dietary data were collected

NHANES Dataset

- For each year, individuals have a unique identifier called a sequence number
- All files for that year can be linked by the sequence number (variable name: SEQN)
- Every year, different participants are sampled, so you cannot track the same person over time
  - You can track the average population intakes over time.

NHANES Questions

- What do the nutrient intake profiles of individuals look like? For example, do people who consume more saturated fats consume less fiber?
- Do nutrient intake profile patterns differ by gender, age, race, education, or poverty level?
- What do time trends in nutrient patterns look like?
- How consistent are individuals in their consumption over two days? For example f they are consuming high amounts of fiber on day one, are they also consuming high amounts on day 2?

Medicaid-Vital Statistics Data

Medicaid-Vital Statistics Data

- The Medicaid and Medicare Administration in the State of Delaware (DMMA) examines medical usage and health outcomes of their clients on a regular basis
- A report came out in 2014 that stated that individuals with mental illness were twice as likely to die and that they die at a much earlier age (almost 20 years earlier) compared to those without mental illness in the United States
- DMMA wanted to know if this was true of their Medicaid population
- The DSAMH_Medicaid dataset is a subset of about 6000 individuals who have Medicaid as their primary insurance and have at least one instance of mental illness.
- These data were then linked to vital statistics to determine mortality
  - There were approximately 200 deaths

Medicaid-Vital Statistics Data

- What cause of death codes are most commonly reported in the DSAMH_Medicaid dataset?
- Do the cause of death codes cluster into groups by gender? Or by disability status?
- Are there differences in the patterns of medical care (i.e., time spent in Medicaid, number of medical claims, number of hospital claims, number of emergency department claims, total billed amounts) reported by those that die versus those that didn't die?

Project Steps

Project (I)

- Step 0: search for datasets
  - Discussion in class of datasets identified

Project (II)

- Step 1: Address (and answer) these questions
  - Do you work alone or in team? If in team, indicate who is your collaborator and what are the skills / expertise he/she is bringing to the project
  - What is the dataset you are considering?
  - What are the possible key question(s) you want to answer? Are the questions too general? Are the questions too narrow?

Project (III)

- What are the scientific questions you are answering with your tool/framework? Be as specific as possible.
- What is the tentative title of your project?
- What are the milestones you want to meet from now until the end of the semester when you will present your poster at the poster showcase? Be as specific as possible. Write the date of the deadline and the task(s) you want to achieve for that deadline.

Reading

Reading

Therese D. Pigott. A Review of Methods for Missing Data. (2001)

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

BIG ORANGE. BIG IDEAS.®

BIG ORANGE BIG IDEAS