Introduction to R & Machine Learning

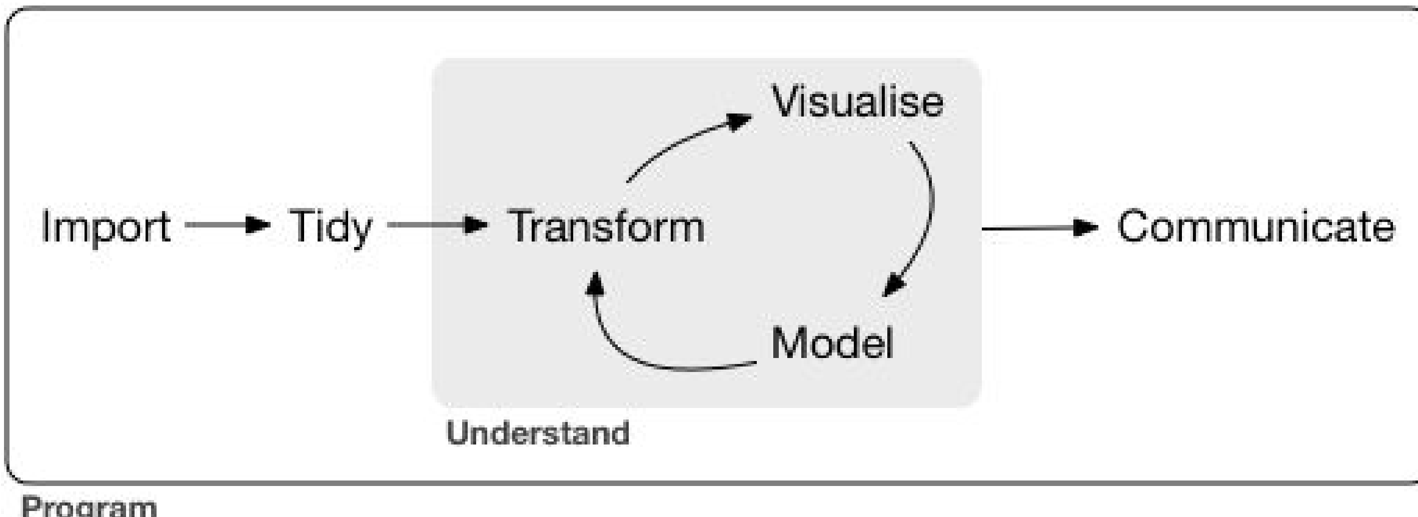
Topics

Part B:

1. Machine Learning Workflow
2. Data Wrangling
    a. Tidy Data
    b. Data Transformation
3. Machine Learning Models
    a. Supervised Algorithms
    b. Unsupervised Algorithms
    c. Cross Validation
    d. Apriori Algorithm
    e. References

Machine Learning Workflow

- Typical Data Science project workflow:



http://r4ds.had.co.nz/introduction.html

Data Wrangling-Tidy data

- Raw data:
  - Excel file with multiple worksheets
  - Missing Data (Survey)
  - Database
  - JSON/XML Data files

	country	`1999`	`2000`
*	<chr>	<int>	<int>
1	Afghanistan	19987071	20595360
2	Brazil	172006362	174504898
3	China	1272915272	1280428583

	country	`1999`	`2000`
*	<chr>	<int>	<int>
1	Afghanistan	745	2666
2	Brazil	37737	80488
3	China	212258	213766

	country	year	cases	population
	<chr>	<int>	<int>	<int>
1	Afghanistan	1999	745	19987071
2	Afghanistan	2000	2666	20595360
3	Brazil	1999	37737	172006362
4	Brazil	2000	80488	174504898
5	China	1999	212258	1272915272
6	China	2000	213766	1280428583

Data Transformation

- Why transform?
  - Select/Eliminate variables
  - Summarize variables
  - Select/Eliminate observations
- arrange()
- select()
- filter()

Let’s look at a real life example!

Some Mathematical Transformations

- Normalization:
  - Bring everyone on the same page! Same range! [0-1]
  - When multiple numerical variables have different value ranges
  - E.g. online product ratings (0-5 stars, 0-3 stars, etc.)
  - Commonly used: min-max normalization $X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$
- Discretization
  - When a variable has "continuous" values but your algorithm needs "discrete" values
  - Divide into ranges
    - E.g. 0, 3, 25, 12, 50, 35, 47, 28
    - => 0-9, 0-9, 20-29, 10-19, 50-59, 30-39, 40-49, 20-29
  - E.g. Flower petal size (continuous) and Type of flower

Machine Learning Models

- Types of Machine Learning Algorithms:
    - Supervised Algorithms: defined predictor/decision
        - Classification
        - Regression
        - Time Series Analysis
    - Unsupervised Algorithms: identify patterns
        - Frequent Pattern Mining
        - Clustering

Supervised Algorithms

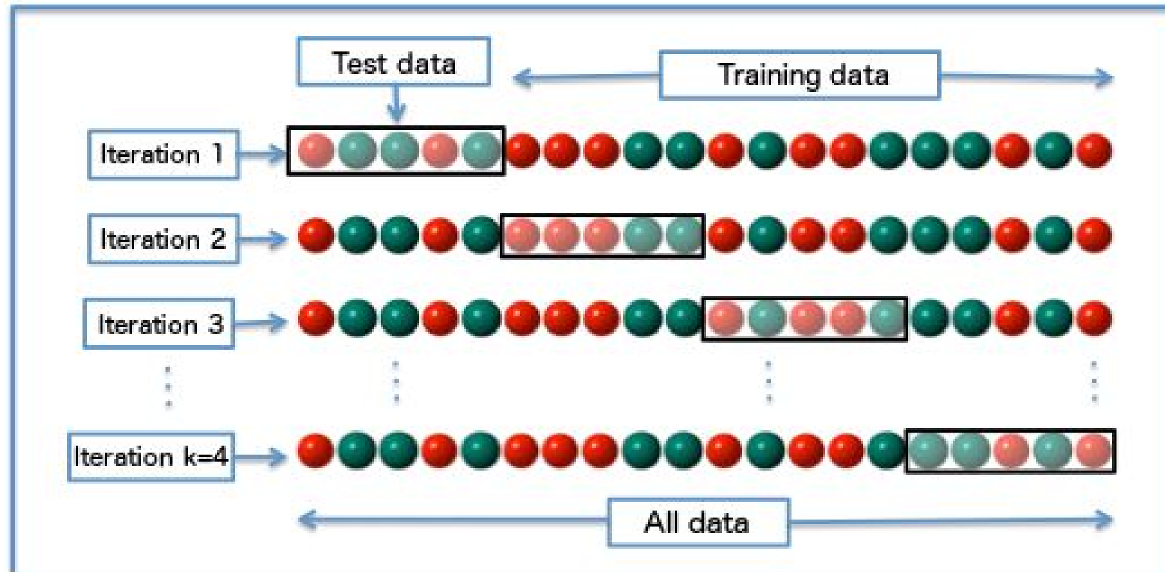
- Similar to helping a child ‘learn’
- A definitive ‘class’ to predict: e.g. is the weather good to play soccer? Is this email legitimate or spam? Is this a picture from Game of Thrones?
- Involves ‘predictors’ or ‘features’ or ‘variables’
    - Might be directly provided in the dataset or might have to be generated
- Generally involves two applications:
    - Evaluation of Algorithms
    - Prediction
- Evaluation:
    - Distribute your data into 3 sets:
    - Train, Validate, Test (50:25:25)
    - Train: Let the algorithm learn (build a model/rules)
    - Validate: Set parameters for the algorithm
    - Test: Classify unknown observations and observe the performance
- Prediction.. Any guess?

Unsupervised Algorithms

- Find patterns in the data
- Unlabeled datasets i.e. no known class
  - E.g. no labeled data for GoT and non-GoT pictures
- Mostly statistical
- Difficult to identify a ‘good’ algorithm
  - No way to identify ‘accuracy’
- Typically ‘club’ data based on their relationship/behavior with each other
- Most real-world problems use Unsupervised Algorithms
- Starts with a smaller dataset to choose parameters

Cross Validation

- Partition the dataset into k parts, use kth part for training and (k-1) parts for testing/evaluation



- The Good:
  - Not sequence constrained
- The Bad:
  - Overfitting
  - Allows multiple iterations over the data
- The Ugly:
  - Time Series

Apriori Algorithm/Association Rule Mining

- Mining frequent itemsets/Market Basket Analysis
- For transaction data recorded by Point-of-Sale systems
  - Web usage mining, bioinformatics
- Identify items frequently bought together
- Terminologies:
  - Support: Frequency of occurrence of an itemset
  - Confidence: How often a rule has been found to be true
  - Frequent Itemset: A set of items that has frequency of occurrence > minSupport
  - Apriori property: Any subset of a frequent itemset should also be frequent
  - C_k: Candidate itemset of size k
  - L_k: frequent itemset of size k
  - Self-Join Step: {1},{2},{3}: {1,2}, {1,3}, {2,3}
  - Prune Step: get rid of itemsets with frequency< minSupport

The Apriori Algorithm

C_k: Candidate itemset of size k
L_k : frequent itemset of size k

```
L_1 = {frequent items};
for (k = 1; L_k != ∅; k++) do begin
    C_{k+1} = candidates generated from L_k;
    for each transaction t in database do
        increment the count of all candidates in
        C_{k+1} that are contained in t
    L_{k+1} = candidates in C_{k+1} with min_support
    end
return ∪_k L_k;
```

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	2

Where are {1,3,5} and {1,2,3} in C3?

Generating Rules

- For each frequent itemset *'l'*, generate all nonempty subsets of *l*
- For every non-empty subset *'s'* of *l*, output the rule *'s→(l-s)'* if *supportCount(l)/supportCount(s)>=minConfidence*
- Assume minConfidence=70%
- *l={2,3,5}*
- Non-empty subsets of *l*: {2,3}, {3,5}, {2,5}, {2}, {3}, {5}
- Rule: {2,3}→{5}
  - Confidence: supportCount({2,3,5})/supportCount({2,3}) = 2/2 = 100%
- Rule: {2,5}→{3}
  - Confidence: supportCount({2,3,5})/supportCount({2,5}) = 2/3 = 22%
- Similarly for other rules

References

- ML Algorithms: http://www.dataschool.io/comparing-supervised-learning-algorithms/
- Apriori Examples: http://cse.iitkgp.ac.in/~bivasm/uc_notes/07apriori.pdf
- RStudio Cheatsheets:https://www.rstudio.com/resources/cheatsheets/
- R for Data Science: http://r4ds.had.co.nz/index.html
- http://topepo.github.io/caret/index.html
- The Art of Data Storytelling: https://www.linkedin.com/feed/update/urn:li:activity:6230324671884165120/
- Github:https://github.com/apresley1

- Email:
  - Deepti Bhatia: deeptib@timeforge.com
  - Jainam Shah: jainams@timeforge.com