Adv. Methods

Module 10 – Advanced Analytics - Theory and Methods Part III

EMC[2] PROVEN PROFESSIONAL

Module 10: Advanced Analytics – Theory and Methods – Part III

Upon completion of this module, you should be able to:

- Examine analytic needs and select an appropriate technique based on business objectives; initial hypotheses; and the data's structure and volume
- Apply some of the more commonly used methods in Analytics solutions
- Explain the algorithms and the technical foundations for the commonly used methods
- Explain the environment (use case) in which each technique can provide the most value
- Use appropriate diagnostic methods to validate the models created
- Use R and in-database analytical functions to fit, score and evaluate models
- Learn a bit about Text Analysis

What Kind of Problem do I Need to Solve?

How do I Solve it? ***<This module will focus on classification>***

The Problem to Solve	The Category of Techniques	Covered in this Course
I want to group items by similarity. I want to find structure (commonalities) in the data	Clustering	K-means clustering
I want to discover relationships between actions or items	Association Rules	Apriori
I want to determine the relationship between the outcome and the input variables	Regression	Linear Regression Logistic Regression
I want to assign (known) labels to objects	Classification	Naïve Bayes Decision Trees
I want to find the structure in a temporal process I want to forecast the behavior of a temporal process	Time Series Analysis	ACF, PACF, ARIMA
I want to analyze my text data	Text Analysis	Regular expressions, Document representation (Bag of Words), TF-IDF

Module 10: Advanced Analytics – Theory and Methods – Part III

Part A: Naïve Bayesian Classifiers

During this lesson the following topics are covered:

- Naïve Bayesian Classifier
- Theoretical foundations of the classifier
- Use cases
- Evaluating the effectiveness of the classifier
- The Reasons to Choose (+) and Cautions (-) with the use of the classifier

Classifiers

Where in the catalog should I place this product listing?
Is this email spam?
Is this politician Democrat/Republican/Green?

- Classification: assign labels to objects.
- Usually supervised: training set of pre-classified examples.
- Our examples:
  - Naïve Bayes,
  - Decision Trees
  - (and Logistic Regression)

Naïve Bayesian Classifier : What is it?

- Used for classification
  - Actually returns a probability score on class membership:
    - In practice, probabilities generally close to either 0 or 1
    - Not as well calibrated as Logistic Regression
- Input variables are discrete
  - **Popular for text classification**
- Output:
  - Most implementations: log probability for each class
    - You could convert it to a probability, but in practice, we stay in the log space

Naïve Bayesian Classifier - Use Cases

- Preferred method for many text classification problems.
  - Try this first; if it doesn't work, try something more complicated
- Use cases
  - Spam filtering, other text classification tasks
  - Fraud detection

Building a Training Dataset

<u>Example : Predicting Good or Bad credit</u>

Predict the credit behavior of a credit card applicant from applicant's attributes:

- personal status
- job type
- housing type
- savings account

These are all categorical variables; better suited to Naïve Bayesian classifier than to logistic regression.

personal_status	job	housing	savings_status	credit_class
male single	skilled	own	no known savings	good
female div/dep/mar	skilled	own	<100	bad
male single	unskilled resident	own	<100	good
male single	skilled	for free	<100	good
male single	skilled	for free	<100	bad
male single	unskilled resident	for free	no known savings	good
male single	skilled	own	500<=X<1000	good
male single	high qualif/self emp/mgm	rent	<100	good
male div/sep	unskilled resident	own	>=1000	good
male mar/wid	high qualif/self emp/mgm	own	<100	bad
female div/dep/mar	skilled	rent	<100	bad
female div/dep/mar	skilled	rent	<100	bad
female div/dep/mar	skilled	own	<100	good
male single	unskilled resident	own	<100	bad
female div/dep/mar	skilled	rent	<100	good
female div/dep/mar	unskilled resident	own	100<=X<500	bad
male single	skilled	own	no known savings	good
male single	skilled	own	no known savings	good
female div/dep/mar	high qualif/self emp/mgm	for free	<100	bad
male single	skilled	own	500<=X<1000	good
male single	skilled	own	<100	good
male single	skilled	rent	500<=X<1000	good
male single	unskilled resident	rent	<100	good
male single	skilled	own	100<=X<500	good
male mar/wid	skilled	own	no known savings	good
male single	unskilled resident	own	<100	good
male mar/wid	unskilled resident	own	<100	good

Technical Description - Bayes' Law

$$P(B|A) = \frac{P(A \wedge B)}{P(A)} = \frac{P(A|B)P(B)}{P(A)}$$

- B is the class label:
  - B ε $\{b_1, b_2, \dots b_n\}$
- A is the specific assignment of input variables
  - $A = (a_1, a_2, \dots a_m)$

Reverend Thomas Bayes

EMC[2] PROVEN PROFESSIONAL

The "Naïve" Assumption: Conditional Independence

$$P(A|b_j) = P(a_1, a_2, ..., a_m|b_j)$$
$$= \prod_i^m P(a_i|b_j)$$

so:

$$\boxed{P(b_j|a_1, a_2, ..., a_m) = \frac{\prod_i^m P(a_i|b_j) P(b_j)}{\cancel{P(a_1, a_2, ..., a_m)}}}$$

Independent of class – so it cancels out

Building a Naïve Bayesian Classifier

- To build a Naïve Bayesian classifier, collect the following statistics from the training data:
  - $P(b_j)$ for all the class labels.
  - $P(a_i | b_j)$ for all possible assignments of the input variables and class labels.

Credit example:

- class labels: {good, bad}
  - P(good) = 0.7
  - P(bad) = 0.3
- aggregates for housing
  - P(own|bad) = 0.62
  - P(own|good) = 0.75
  - P(rent|bad) = 0.23
  - P(rent|good) = 0.14
  - … and so on

Building a Naïve Bayesian Classifier (Continued)

- Assign the label that maximizes the value

$$\prod_{i}^{m} P(a_i|b_j)P(b_j)$$

Back to Credit Example

P(good|X) ~ (0.28*0.75*0.14*0.06)*0.7 = 0.0012

P(bad|X) ~ (0.36*0.62*0.17*0.02)*0.3 = 0.0002

Credit Example: X

- female
- owns home
- Self-employed
- savings > $1000

a_i	b_j	$P(a_i \mid b_j)$
female	good	0.28
female	bad	0.36
own	good	0.75
own	bad	0.62
self emp	good	0.14
self emp	bad	0.17
savings>1K	good	0.06
savings>1K	bad	0.02

P(good|X) > P(bad|X):

Assign X the label "good"

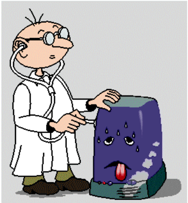
Implementation Guideline

- High-dimensional problems are prone to numerical underflow and unobserved events; it's better to calculate the log probability (with smoothing).

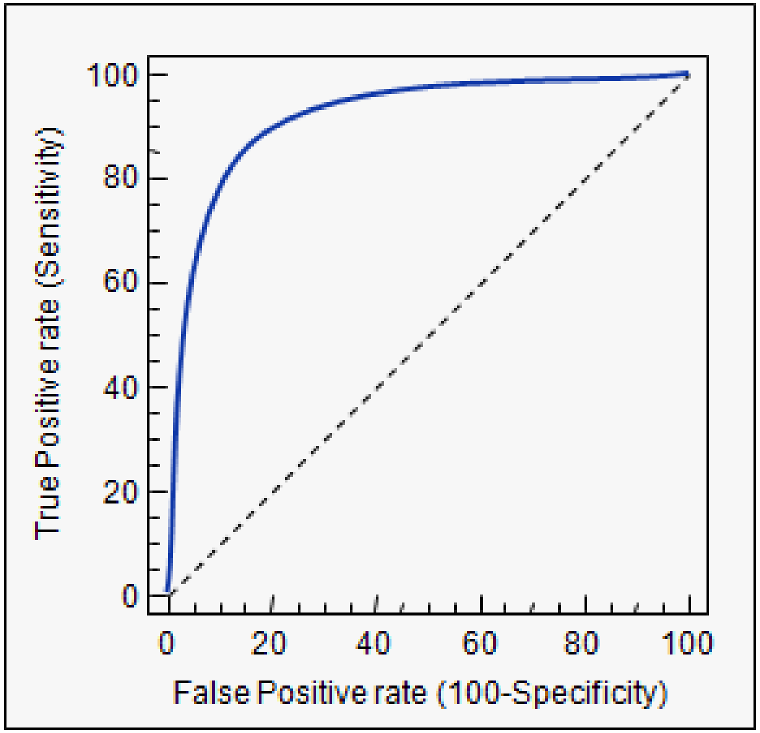
$$\sum_{i}^{m} \log(P(a_i|b_j) + \epsilon) + \log(P(b_j) + \epsilon)$$

(Smoothing technique varies with implementation)

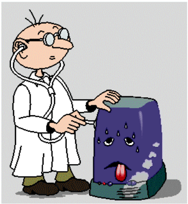
Diagnostics



- Hold-out data
  - How well does the model classify new instances?
- Cross-validation
- ROC curve/AUC



Diagnostics: Confusion Matrix



	Prediction		
True Class	bad	good	
bad	262	38	300
good	29	671	700
	291	709	1000

false positives

false negatives

accuracy: sum of diagonals / sum of table = (262+671)/1000 = 0.93

FPR: false positives / sum of first row = 38/300 = 0.13
FNR: false negatives / sum of second row = 29/700 = 0.04

Precision: true positives / sum of second column = 671/709 = 0.95
Recall: true positives / sum of second row = 671/700 = 0.96

Naïve Bayesian Classifier - Reasons to Choose (+) and Cautions (-)

Reasons to Choose (+)	Cautions (-)
Handles missing values quite well	Numeric variables have to be discrete (categorized) Intervals
Robust to irrelevant variables	Sensitive to correlated variables "Double-counting"
Easy to implement	Not good for estimating probabilities Stick to class label or yes/no
Easy to score data	
Resistant to over-fitting	
Computationally efficient Handles very high dimensional problems Handles categorical variables with a lot of levels	

Check Your Knowledge

Your Thoughts?

1. Consider the following Training Data Set:
   - Apply the Naïve Bayesian Classifier to this data set and compute

     P(y = 1|X) for X = (1,0,0)

     Show your work

Training Data Set

X1	X2	X3	Y
1	1	1	0
1	1	0	0
0	0	0	0
0	1	0	1
1	0	1	1
0	1	1	1

2. List some prominent Use Cases of the Naïve Bayesian Classifier.
3. What gives the Naïve Bayesian Classifier the advantage of being computationally inexpensive?
4. Why should we use log-likelihoods rather than pure probability values in the Naïve Bayesian Classifier?

Check Your Knowledge (Continued)

Your Thoughts?

5. What is a confusion matrix and how it is used to evaluate the effectiveness of the model?
6. Consider the following data set with two input features temperature and season
   - What is the Naïve Bayesian assumption?
   - Is the Naïve Bayesian assumption satisfied for this problem?

Temperature	Season	Electricity Usage (Class)
Below Average	Winter	High
Above Average	Winter	Low
Below Average	Summer	Low
Above Average	Summer	High

Module 10: Advanced Analytics – Theory and Methods

Part A: Naïve Bayesian Classifiers - Summary

During this lesson the following topics were covered:

- Naïve Bayesian Classifier
- Theoretical foundations of the classifier
- Use cases
- Evaluating the effectiveness of the classifier
- The Reasons to Choose (+) and Cautions (-) with the use of the classifier

Lab Exercise 7- Part A: Naïve Bayesian Classifier

This Lab is designed to investigate and practice the Naïve Bayesian Classifier analytic technique.

After completing the tasks in this lab you should be able to:

- Use R functions for Naïve Bayesian Classification
- Apply the requirements for generating appropriate training data
- Validate the effectiveness of the Naïve Bayesian Classifier with the big data

Lab Exercise 7: Naïve Bayesian Classifier PartA - Workflow

1. Set working directory and review training and test data
2. Install and load library “e1071”
3. Read in and review data
4. Build the Naïve Bayesian classifier Model from First Principles
5. Predict the Results
6. Execute the Naïve Bayesian Classifier with e1071 package
7. Predict the Outcome of “Enrolls” with the Testdata
8. Review results

Module 10: Advanced Analytics – Theory and Methods – Part III

Part B: Decision Trees

During this lesson the following topics are covered:

- Overview of Decision Tree classifier
- General algorithm for Decision Trees
- Decision Tree use cases
- Entropy, Information gain
- Reasons to Choose (+) and Cautions (-) of Decision Tree classifier
- Classifier methods and conditions in which they are best suited

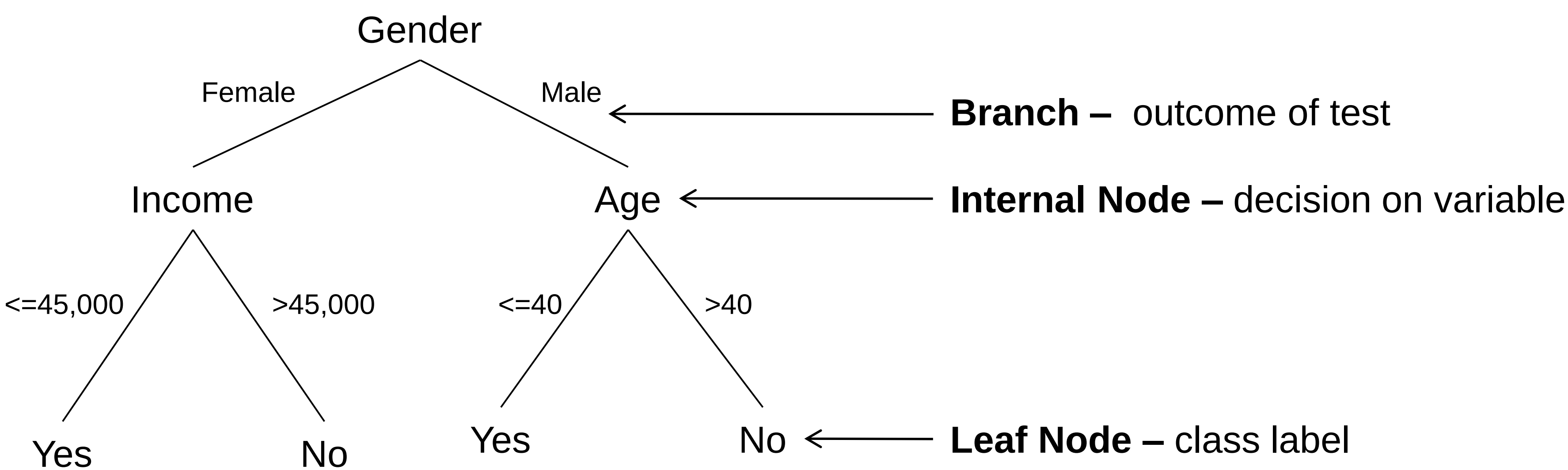
Decision Tree Classifier - What is it?

- Used for classification:
  - Returns probability scores of class membership
    - Well-calibrated, like logistic regression
    - Assigns label based on highest scoring class
    - Some Decision Tree algorithms return simply the most likely class
  - Regression Trees: a variation for regression
    - Returns average value at every node
    - Predictions can be discontinuous at the decision boundaries
- Input variables can be continuous or discrete
- Output:
  - A tree that describes the decision flow.
  - Leaf nodes return either a probability score, or simply a classification.
  - Trees can be converted to a set of "decision rules“
    - "IF income < $50,000 AND mortgage_amt > $100K THEN default=T with 75% probability“

Decision Tree – Example of Visual Structure

Female

Male



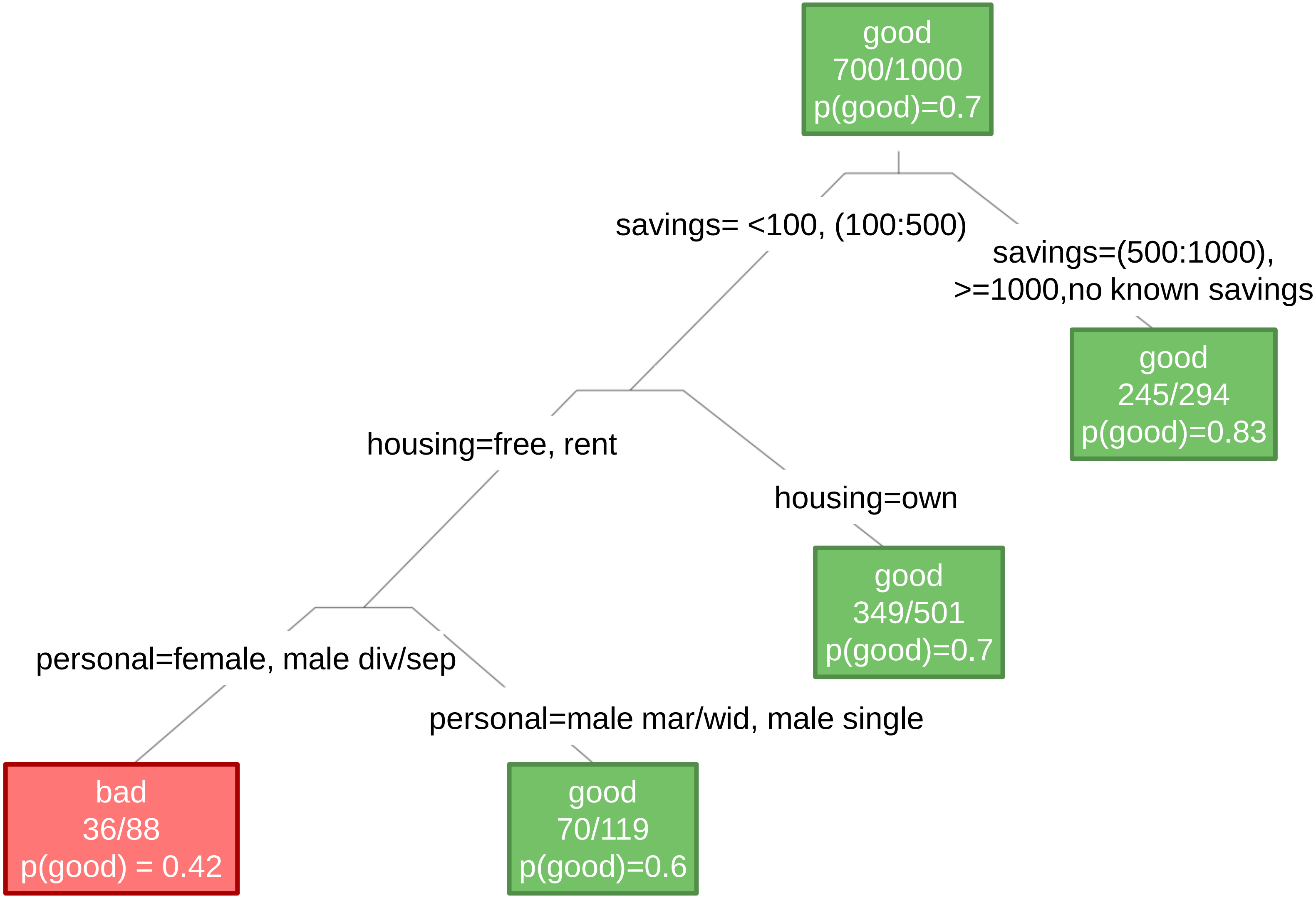
Income

Age

Decision Tree Classifier - Use Cases

- When a series of questions (yes/no) are answered to arrive at a classification
  - Biological species classification
  - Checklist of symptoms during a doctor's evaluation of a patient
- When "if-then" conditions are preferred to linear models.
  - Customer segmentation to predict response rates
  - Financial decisions such as loan approval
  - Fraud detection
- Short Decision Trees are the most popular "weak learner" in ensemble learning techniques

Example: The Credit Prediction Problem



General Algorithm

- To construct tree T from training set S
  - If all examples in S belong to some class in C, or S is sufficiently "pure", then make a leaf labeled C.
  - Otherwise:
    - select the “most informative” attribute A
    - partition S according to A’s values
    - recursively construct sub-trees T1, T2, ..., for the subsets of S
- The details vary according to the specific algorithm – CART, ID3, C4.5 – but the general idea is the same

Step 1: Pick the Most "Informative" Attribute

- Entropy-based methods are one common way

$$H = -\sum_{c} p(c) \log_2 p(c)$$

- H = 0 if p(c) = 0 or 1 for any class
  - So for binary classification, H=0 is a "pure" node
- H is maximum when all classes are equally probable
  - For binary classification, H=1 when classes are 50/50

Step 1: Pick the most "informative" attribute (Continued)

- First, we need to get the base entropy of the data

$$\boxed{\begin{aligned} H_{credit} &= -(0.7\log_2(0.7) + 0.3\log_2(0.3)) \\ &= 0.88 \end{aligned}}$$

Step 1: Pick the Most "Informative" Attribute (Continued) Conditional Entropy

$$H_{attr} = -\sum_{v} p(v) \sum_{c} p(c|v) \log_2 p(c|v)$$

- The weighted sum of the class entropies for each value of the attribute
- In English: attribute values (home owner vs. renter) give more information about class membership
  - "Home owners are more likely to have good credit than renters"
- Conditional entropy should be lower than unconditioned entropy

Conditional Entropy Example

	for free	own	rent
P(housing)	0.108	0.713	0.179
P(bad | housing)	0.407	0.261	0.391
p(good | housing)	0.592	0.739	0.601

$$\begin{aligned} H_{(housing|credit)} &= -[0.108 * (0.407 \log_2(0.407) + 0.592 \log_2(0.592)) \\ &\quad + 0.713 * (0.261 \log_2(0.261) + 0.739 \log_2(0.739)) \\ &\quad + 0.179 * (0.391 \log_2(0.391) + 0.601 \log_2(0.601))] \\ &= 0.868 \end{aligned}$$

Step 1: Pick the Most “Informative" Attribute
(Continued) Information Gain

$$\mathrm{InfoGain}_{attr} = H - Hattr$$

- The information that you gain, by knowing the value of an attribute
- So the "most informative" attribute is the attribute with the highest InfoGain

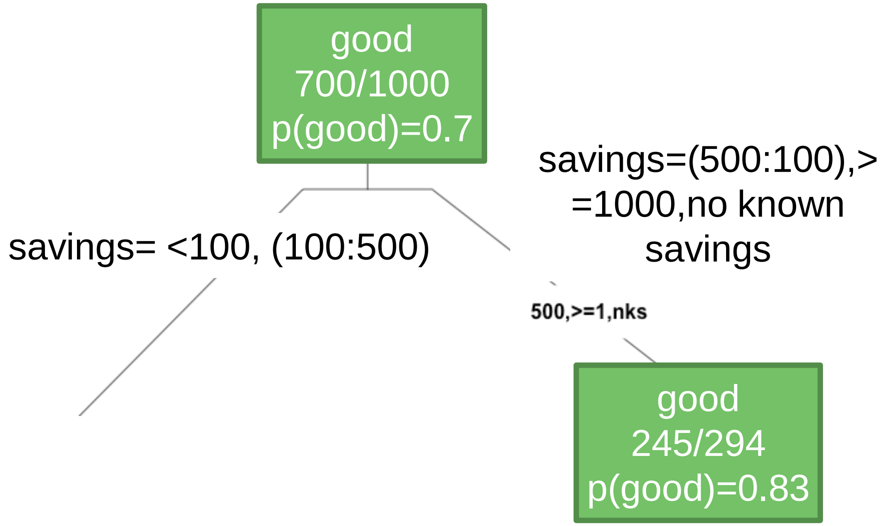
Back to the Credit Prediction Example

$$\begin{aligned} \text{InfoGain}_{credit} &= H_{credit} - H_{housing|credit} \\ &= 0.88 - 0.86 \\ &\approx 0.013 \end{aligned}$$

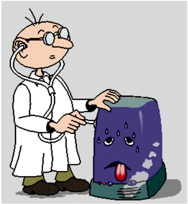
Attribute	InfoGain
job	0.001
housing	0.013
personal_status	0.006
savings_status	***0.028***

Step 2 & 3: Partition on the Selected Variable

- Step 2: Find the partition with the highest InfoGain
  - In our example the selected partition has InfoGain = 0.028
- Step 3: At each resulting node, repeat Steps 1 and 2
  - until node is "pure enough"
- Pure nodes => no information gain by splitting on other attributes



Diagnostics



- Hold-out data
- ROC/AUC
- Confusion Matrix
- FPR/FNR, Precision/Recall
- Do the splits (or the "rules") make sense?
  - What does the domain expert say?
- How deep is the tree?
  - Too many layers are prone to over-fit
- Do you get nodes with very few members?
  - Over-fit

Decision Tree Classifier - Reasons to Choose (+) & Cautions (-)

Reasons to Choose (+)	Cautions (-)
Takes any input type (numeric, categorical) In principle, can handle categorical variables with many distinct values (ZIP code)	Decision surfaces can only be axis-aligned
Robust with redundant variables, correlated variables	Tree structure is sensitive to small changes in the training data
Naturally handles variable interaction	A "deep" tree is probably over-fit Because each split reduces the training data for subsequent splits
Handles variables that have non-linear effect on outcome	Not good for outcomes that are dependent on many variables Related to over-fit problem, above
Computationally efficient to build	Doesn't naturally handle missing values; However most implementations include a method for dealing with this
Easy to score data	In practice, decision rules can be fairly complex
Many algorithms can return a measure of variable importance	
In principle, decision rules are easy to understand	

EMC^2 PROVEN PROFESSIONAL

Which Classifier Should I Try?

Typical Questions	Recommended Method
Do I want class probabilities, rather than just class labels?	Logistic regression Decision Tree
Do I want insight into how the variables affect the model?	Logistic regression Decision Tree
Is the problem high-dimensional?	Naïve Bayes
Do I suspect some of the inputs are correlated?	Decision Tree Logistic Regression
Do I suspect some of the inputs are irrelevant?	Decision Tree Naïve Bayes
Are there categorical variables with a large number of levels?	Naïve Bayes Decision Tree
Are there mixed variable types?	Decision Tree Logistic Regression
Is there non-linear data or discontinuities in the inputs that will affect the outputs?	Decision Tree

Check Your Knowledge

Your Thoughts?

1. How do you define information gain?
2. For what conditions is the value of entropy at a maximum and when is it at a minimum?
3. List three use cases of Decision Trees.
4. What are weak learners and how are they used in ensemble methods?
5. Why do we end up with an over fitted model with deep trees and in data sets when we have outcomes that are dependent on many variables?
6. What classification method would you recommend for the following cases:
   - High dimensional data
   - Data in which outputs are affected by non-linearity and discontinuity in the inputs

Module 10: Advanced Analytics – Theory and Methods – Part III

:Part B: Decision Trees - Summary

During this lesson the following topics were covered:

- Overview of Decision Tree classifier
- General algorithm for Decision Trees
- Decision Tree use cases
- Entropy, Information gain
- Reasons to Choose (+) and Cautions (-) of Decision Tree classifier
- Classifier methods and conditions in which they are best suited

Lab Exercise 7 Part B: Decision Trees

This Lab is designed to investigate and practice Decision Tree (DT) models covered in the course work.

After completing the tasks in this lab you should be able to:

- Use R functions for Decision Tree models
- Predict the outcome of an attribute based on the model

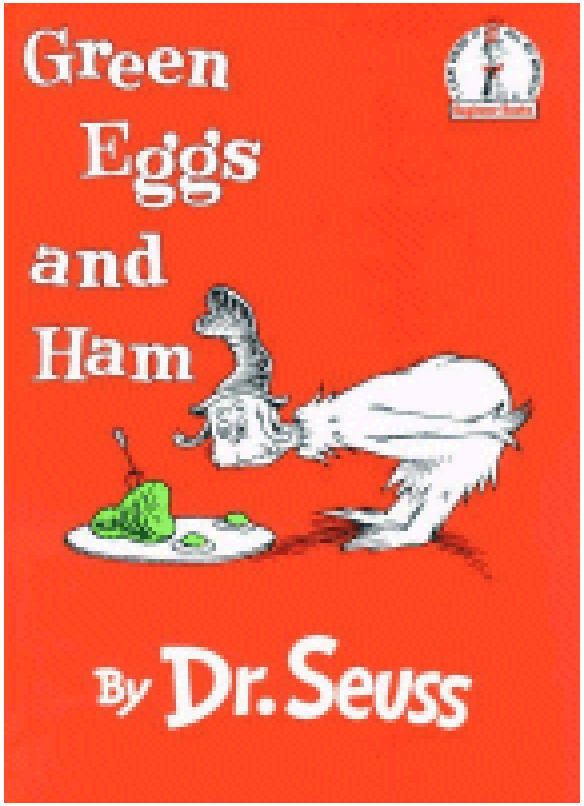
Lab Exercise 7 part B: Decision Trees - Workflow

1. Set the Working Directory
2. Read in the Data
3. Build the Decision Tree
4. Plot the Decision Tree
5. Prepare Data to Test the Fitted Model
6. Predict a Decision from the Fitted Model

Lesson: Text Analysis

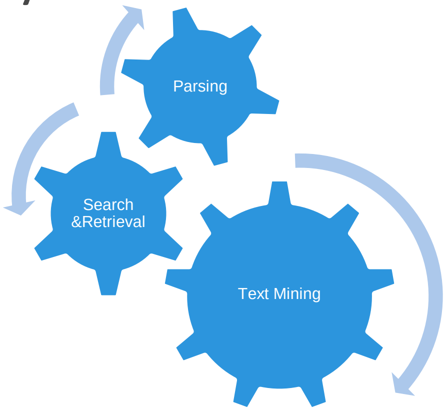
Encompasses the processing and representation of text for analysis and learning tasks

- **High-dimensionality**
  - Every distinct term is a dimension
  - *Green Eggs and Ham*: A 50-D problem!
- **Data is Un-structured**

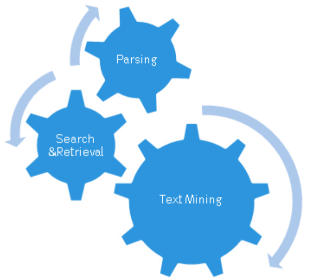


Text Analysis – Problem-solving Tasks

- Parsing
  - Impose a structure on the unstructured/semi-structured text for downstream analysis
- Search/Retrieval
  - Which documents have this word or phrase?
  - Which documents are about this topic or this entity?
- Text-mining
  - "Understand" the content
  - Clustering, classification
- Tasks are not an ordered list
  - Does not represent process
  - Set of tasks used appropriately depending on the problem addressed

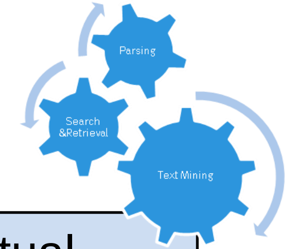


Example: Brand Management



- Acme currently makes two products
  - bPhone
  - bEbook
- They have lots of competition. They want to maintain their reputation for excellent products and keep their sales high.
- What is the buzz on Acme?
  - Search for mentions of Acme products
    - Twitter, Facebook, Review Sites, etc.
  - What do people say?
    - Positive or negative?
    - What do people think is good or bad about the products?

Buzz Tracking: The Process



1. Monitor social networks, review sites for mentions of our products.	**Parse** the data feeds to get actual content. Find and filter the raw text for product names (Use **Regular Expression**).
2. Collect the reviews.	Extract the relevant raw text. Convert the raw text into a suitable **document representation**. **Index** into our review **corpus**.
3. Sort the reviews by product.	**Classification** (or **"Topic Tagging"**)
4. Are they good reviews or bad reviews? We can keep a simple count here, for trend analysis.	**Classification** (sentiment analysis)
5. Marketing calls up and reads selected reviews in full, for greater insight.	**Search/Information Retrieval**.

Parsing the Feeds

Parsin
g

1. Monitor social networks, review sites for mentions of our products

- Impose structure on semi-structured data.
- We need to know where to look for what we are looking for.

```
<channel>
<title>All about Phones</title>
<description>My Phone Review Site</description>
<link>http://www.phones.com/link.htm</link>

<item>
<title>bPhone: The best!</title>
<description>I love LOVE my bPhone!</description>
<link>http://www.phones.com/link.htm</link>
<guid isPermaLink="false"> 1102345</guid>
<pubDate>Tue, 29 Aug 2011 09:00:00 -0400</pubDate>
</item>

</channel>
```

EMC^2 PROVEN PROFESSIONAL

Regular Expressions

Parsing

1. Monitor social networks, review sites for mentions of our products

- Regular Expressions (regexp) are a means for finding words, strings or particular patterns in text.
- A match is a Boolean response. The basic use is to ask “does this regexp match this string?”

regexp	matches	Note
b[P|p]hone	bPhone, bphone	Pipe “|” means “or”
bEb*k	bEbook, bEbk, bEback …	“*” is a wildcard, matches anything
^I love	A line starting with "I love"	“^” means start of a string
Acme$	A line ending with “Acme”	“$” means the end of a string

Extract and Represent Text

Parsin
g

2. Collect the reviews

Document Representation:

A structure for analysis

- **"Bag of words"**
  - common representation
  - A vector with one dimension for every unique term in space
    - **term-frequency (tf)**: number times a term occurs
  - Good for basic search, classification
- Reduce Dimensionality
  - Term Space – not ALL terms
    - no stop words: "the", "a"
    - often no pronouns
  - Stemming
    - "phone" = "phones"

"I love LOVE my bPhone!"

Convert this to a vector in the term space:

acme	0
bebook	0
bPhone	1
fantastic	0
love	2
slow	0
terrible	0
terrific	0

Document Representation - Other Features

Parsin
g

2. Collect the reviews

- Feature:
  - Anything about the document that is used for search or analysis.
- Title
- Keywords or tags
- Date information
- Source information
- Named entities

Representing a Corpus (Collection of Documents)

Parsin
g

2. Collect the reviews

- Reverse index
  - For every possible feature, a list of all the documents that contain that feature
- Corpus metrics
  - Volume
  - Corpus-wide term frequencies
  - Inverse Document Frequency (IDF)
    - more on this later
- Challenge: a Corpus is dynamic
  - Index, metrics must be updated continuously

Text Classification (I) - "Topic Tagging"

Text Mining

3. Sort the Reviews by Product

Not as straightforward as it seems

"The bPhone-5X has coverage everywhere. It's much less flaky than my old bPhone-4G."

"While I love Acme's bPhone series, I've been quite disappointed by the bEbook. The text is illegible, and it makes even the Kindle look blazingly fast."

"Topic Tagging"

Text Mining

3. Sort the Reviews by Product

Judicious choice of features

- Product mentioned in title?
- Tweet, or review?
- Term frequency
- Canonicalize abbreviations
  - "5X" = "bPhone-5X"

EMC^2 PROVEN PROFESSIONAL

Text Classification (II) Sentiment Analysis

Text Mining

4. Are they good reviews or bad reviews?

- Naïve Bayes is a good first attempt
- But you need tagged training data!
  - THE major bottleneck in text classification
- What to do?
  - Hand-tagging
  - Clues from review sites
    - thumbs-up or down, # of stars
  - Cluster documents, then label the clusters

Search and Information Retrieval

Search &Retrieval

5. Marketing calls up and reads selected reviews in full, for greater insight.

- Marketing calls up documents with *queries*:
  - Collection of search terms
    - "bPhone battery life"
  - Can also be represented as "bag of words"
  - Possibly restricted by other attributes
    - within the last month
    - from This Review Site

EMC² PROVEN PROFESSIONAL

Quality of Search Results

Search &Retrieval

5. Marketing calls up and reads selected reviews in full, for greater insight.

- Is this document what I wanted?
- Used to rank search results

- Precision
  - What % of documents in the result are relevant?
- Recall
  - Of all the relevant documents in the corpus, what % were returned to me?

Computing Relevance

Search &Retrieval

5. Marketing calls up and reads selected reviews in full, for greater insight.

- Call up all the documents that have any of the terms from the query, and count how many times each term occurs:

$$\text{Relevance}_{document} = \sum_{q_i} tf_{q_i}$$

Inverse Document Frequency (idf)

Search &Retrieva l

5. Marketing calls up and reads selected reviews in full, for greater insight.

$$idf_i = \log (N/tf_i)$$

- *N*: Number of documents in corpus
- tf_i: Number of documents in which term occurs in the corpus

- Measures term uniqueness in corpus
  - "phone" vs. "brick"
- Indicates the importance of the term
  - Search (relevance)
  - Classification (discriminatory power)

TF-IDF and Modified Retrieval Algorithm

Search &Retrieval

> 5. Marketing calls up and reads selected reviews in full, for greater insight.

$$tf_{document}(term) * idf(term)$$

query: *"unbrick phone"*

- Document with "unbrick" a few times more relevant than document with "phone" many times
- Measure of Relevance with tf-idf
- Call up all the documents that have any of the terms from the query, and sum up the tf-idf of each term:

$$\text{Relevance}_{document} = \sum_{q_i} tfidf_{q_i}$$

Other Relevance Metrics

Search &Retrieva l

5. Marketing calls up and reads selected reviews in full, for greater insight.

- "Authoritativeness" of source
  - PageRank is an example of this
- Recency of document
- How often the document has been retrieved by other users

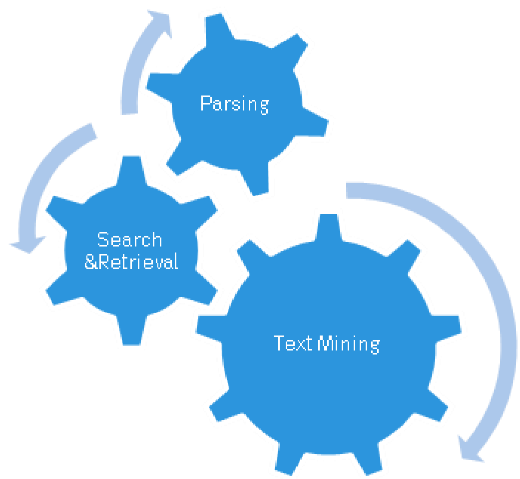
Effectiveness of Search and Retrieval

Search &Retrieval

- Relevance metric
  - important for precision, user experience
- Effective crawl, extraction, indexing
  - important for recall (and precision)
  - more important, often, than retrieval algorithm
- MapReduce
  - Reverse index, corpus term frequencies, idf

Challenges - Text Analysis

- Challenge: finding the right structure for your unstructured data
- Challenge: very high dimensionality
- Challenge: thinking about your problem the right way



Check Your Knowledge

Your Thoughts?

1. What are the two major challenges in the problem of text analysis?
2. What is a reverse index?
3. Why is the corpus metrics dynamic. Provide an example and a scenario that explains the dynamism of the corpus metrics.
4. How does tf-idf enhance the relevance of a search result?
5. List and discuss a few methods that are deployed in text analysis to reduce the dimensions.

Module 10: Advanced Analytics – Theory and Methods – Part III

Text Analysis - Summary

During this lesson the following topics were covered:

- Challenges with text analysis
- Key tasks in text analysis
- Definition of terms used in text analysis
  - Term frequency, inverse document frequency
- Representation and features of documents and corpus
- Use of regular expressions in parsing text
- Metrics used to measure the quality of search results
  - Relevance with tf-idf, precision and recall

Text Analysis: Summary

Key Topics Covered in this module	Methods Covered in this module
Algorithms and technical foundations	Categorization (unsupervised) : K-means clustering Association Rules
Key Use cases	Regression Linear Logistic
Diagnostics and validation of the model	Classification (supervised) Naïve Bayesian classifier Decision Trees
Reasons to Choose (+) and Cautions (-) of the model	Time Series Analysis
Fitting, scoring and validating model in R and in-db functions	Text Analysis