Text Analytics

Ruvan Weerasinghe

University of Colombo School of Computing

Workshop at the Research & Academic Collaboration Program 2018, University of Kelaniya

Plan

- Data and Text Analytics
- Exploration and description vs inferencing
- Exploring text
- Inferencing with text
- Application

Plan

- Data and Text Analytics
- Exploration and description vs inferencing
- Exploring text
- Inferencing with text
- Application

Data Analytics

- Common types of data
  - Numerical and categorical
- Numerical data
  - Lots of descriptive, visual and inferencing techniques
- Categorical
  - Fairly well-developed set of visualizations and inferencing
  - Number of categories usually small

Data Analytics

- The problems with text data
- Usually unstructured
- Number of ‘categories’ very large = number of words
- Words are ambiguous
- Typical techniques used for categorical data unsuitable
  - Observations to categories ratio is insufficient

Plan

- Data and Text Analytics
- Exploration and description vs inferencing
- Exploring text
- Inferencing with text
- Application

Plan

- Data and Text Analytics
- Exploration and description vs inferencing
- Exploring text
- Inferencing with text
- Application

Exploring Data

- Anscombe's quartet

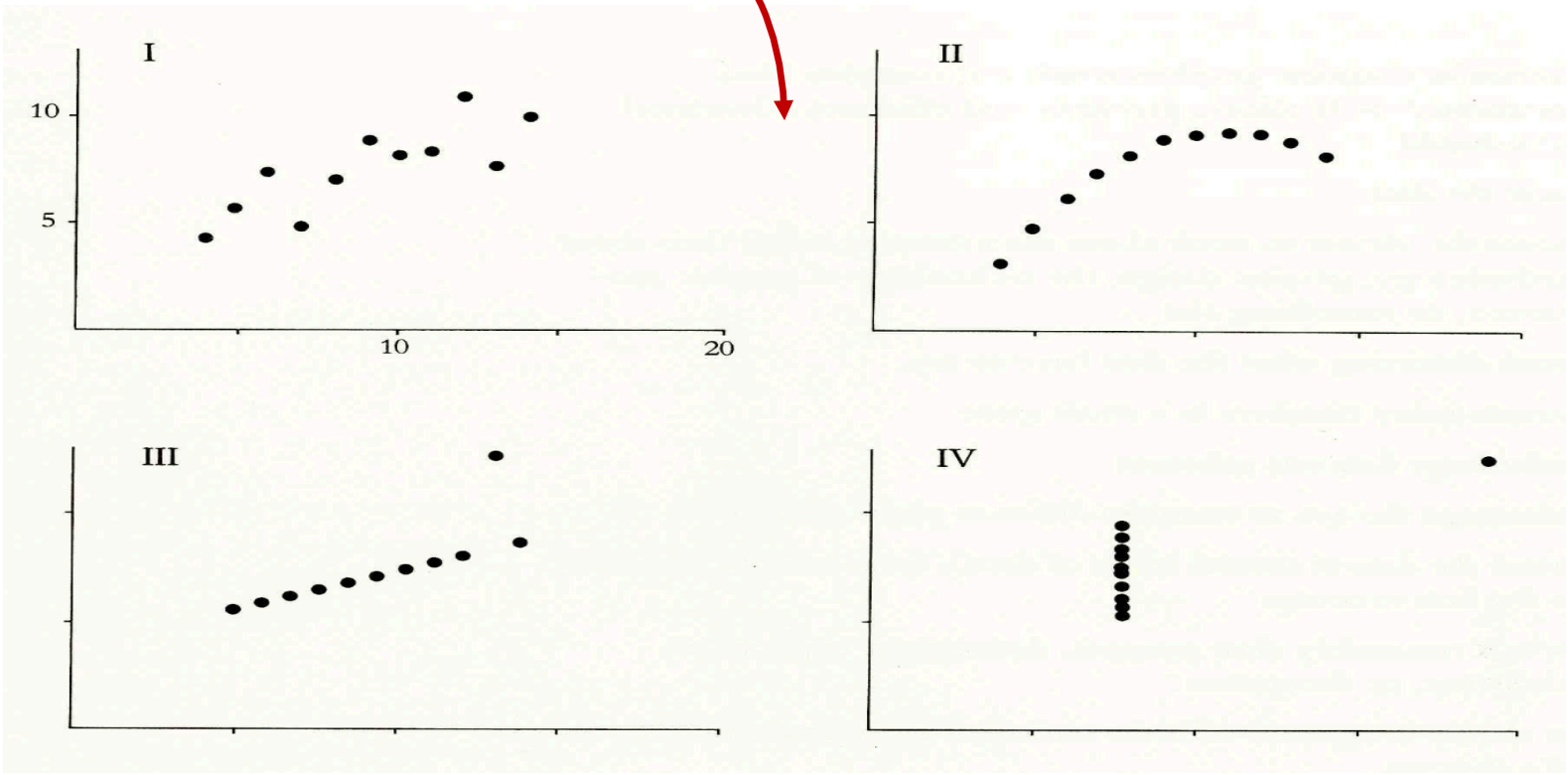
I		II		III		IV	
X	Y	X	Y	X	Y	X	Y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

description

$N = 11$
mean of X's $= 9.0$
mean of Y's $= 7.5$
equation of regression line: $Y = 3 + 0.5X$
standard error of estimate of slope $= 0.118$
$t = 4.24$
sum of squares $X - \overline{X} = 110.0$
regression sum of squares $= 27.50$
residual sum of squares of $Y = 13.75$
correlation coefficient $= .82$
$r^2 = .67$

Exploring Data

visualization



Exploring Text

- How can we describe text?

Exploring Text

- How can we describe text?
  - # of words, # of distinct words
  - # of punctuations
  - # of sentences, length of sentences
- e.g. Brown corpus
  - 'balanced': 15 text categories
  - 1,014,312 words (tokens)
  - 39,440 unique words (distinct tokens)

Exploring Text

- How can we visualize text?
  - e.g. US Presidential State of the Union addresses?

Exploring Text

- How can we visualize text?
- US Presidential SOTU address *tag clouds* compared

Exploring Text

- How can we visualize text?
- DIY tag clouds
  - https://www.wordclouds.com
  - http://tagcrowd.com
  - http://www.wordle.net
  - https://github.com/amueller/word_cloud

Plan

- Data and Text Analytics
- Exploration and description vs inferencing
- Exploring text
- Inferencing with text
- Application

Plan

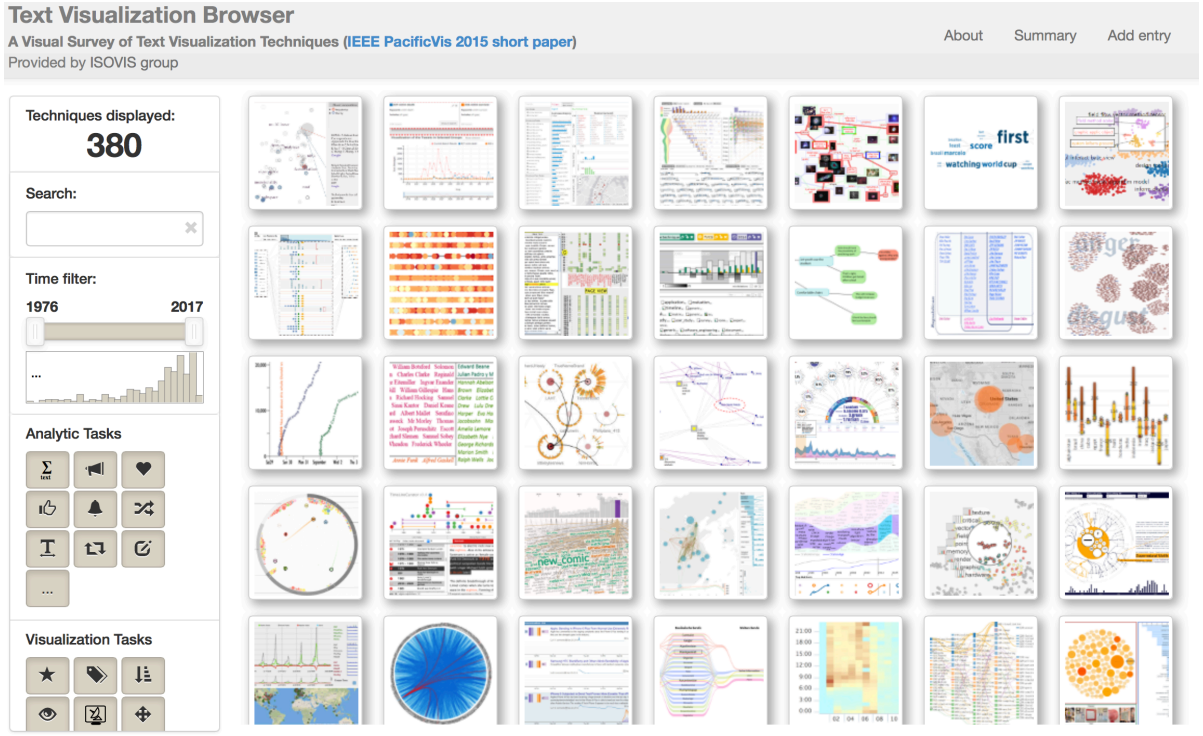
- Data and Text Analytics
- Exploration and description vs inferencing
- Exploring text
- Inferencing with text
- Application

Exploring Text

- Beyond the basics – feature extraction
- Representing text (documents)
  - Presence/absence of words in it
  - Frequency of each word
  - Other measures of importance (e.g. TF-IDF)

Exploring Text

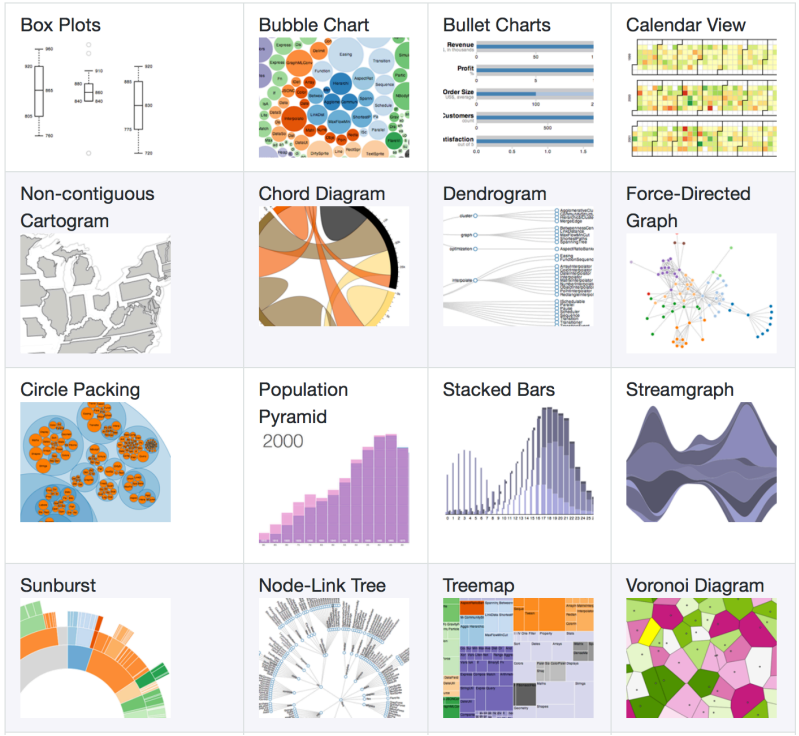
- Beyond the basics – text visualization



http://textvis.lnu.se

Exploring Text

- Beyond the basics – text visualization



http://d3js.org

Plan

- Data and Text Analytics
- Exploration and description vs inferencing
- Exploring text
- Inferencing with text
- Application

Plan

- Data and Text Analytics
- Exploration and description vs inferencing
- Exploring text
- Inferencing with text
- Application

Inferencing with Text

- Primarily done using machine learning techniques
  - Focus on corpus-based data-driven techniques

    1. Data collection
    2. Data cleaning/preparation/wrangling
    3. Feature extraction – dealing with *curse of dimensionality*
    4. Inferencing models with and without supervision

1. Data Collection

- Two primary ways for online data
  - Use API provided by service/social media (recommended)
  - Scrape from the web (more work involved)
- Most APIs have python bindings
  - e.g. Twitter, Facebook etc.
- Scraping from the web
  - Need html/xml parser

1. Data Collection

- Python has some good packages for web scraping
  - Accessing web pages: `urllib`, `requests`
  - Traversing HTML/XML: `BeautifulSoup`
  - Doing both together: `scrapy` (spider, parser and pipeline)
  - Simulating the browser: `selenium`
- General topic is very wide
  - *Mining the social web* – Matthew Russell
  - *Web scraping with Python* – Ryan Mitchell
  - *Automate the boring stuff with Python* – Al Sweigart (*Ch. 11*)

1. Data Collection

- Access `twitter`, `facebook`, `linkedin` data though API with your credentials
  - `https://github.com/ptwobrussell/Mining-the-Social-Web-2nd-Edition/tree/master/ipynb`
- Use `BeautifulSoup` to parse HTML
  - Can use `html.parser`, `lxml` or `html5lib`
  - Simple blog, multi-page listing, dynamic list, Wikipedia page, List of media files (e.g. PDF docs), Wikipedia recursive download
- Use `selenium` to simulate form-filling and mouse clicks
  - Has python bindings (`pip install selenium`)

2. Data Preparation

- Also referred to as cleaning, cleansing or wrangling
- Unlike numerical data, text data is mostly unstructured
  - e.g. in a survey form, it is that ‘Any other comments’ field!
- It is also mostly rather messy
  - Possibly contains markup, ungrammatical text, non-words, punctuation, word forms, misspellings, abbreviations, slang

2. Data Preparation

- Text *normalization* tasks include
  - cleaning text (e.g. remove markup such as
)
  - case conversion (capitalize, uppercase, titlecase, lowercase etc)
  - expanding contractions (e.g. don't → do not)
  - correcting spellings (e.g. fianly → finally)
  - removing *stopwords* and other 'unnecessary' words (non-words)
  - stemming, and lemmatization
    - running → runn vs run

3. Extracting Features

- Inferencing cannot be done even with *cleaned, normalized* text
- Text needs to be *parametrized*
- Common approach is to *vectorize* text data
  - e.g. create a list of all unique words/tokens in corpus
  - then represent each text ('document') using above list as vector

3. Extracting Features

- Assuming a corpus (of documents):
  - Doc 1: Deep learning and big data are hot topics
  - Doc 2: Data analytics is better understood than text analytics
  - Doc 3: Learning these topics can be fun
- Unique word list is:
  - [analytics, and, are, be, better, big, can, data, deep, fun, hot, is, learning, text, than, these, topics, understood]
- Documents are represented 18-element vectors as:
  - [1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1] - *doc 2 as binary*
  - [2, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1] - *doc 2 as frequency*

3. Extracting Features

- Other possible features to extract
  - Tf-idf weights instead of frequency (i.e. frequency in document adjusted for rarity in other documents)
  - Lemmas or stems instead of word tokens
  - N-grams instead of unigrams
  - POS tags instead of (or in addition to) word tokens
  - Chunk tags or constituency branch

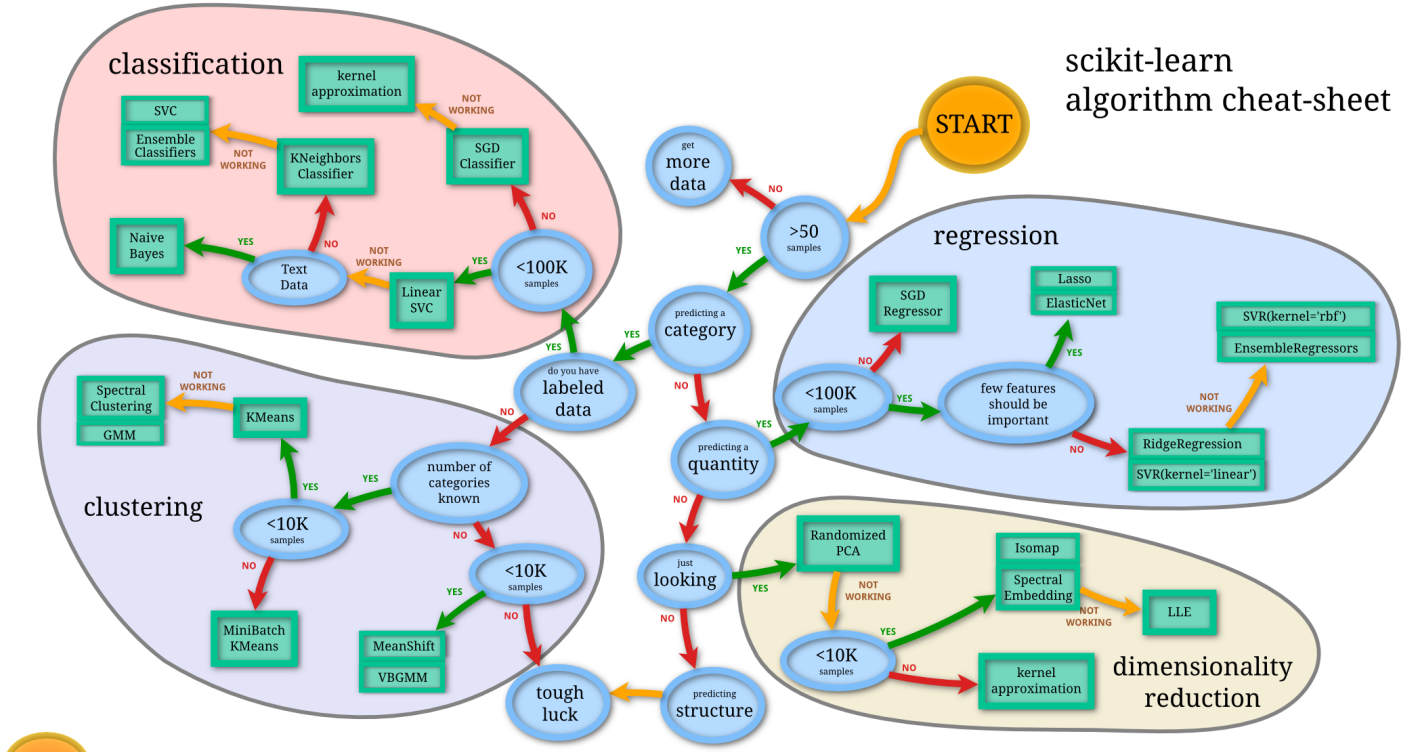
3. Extracting Features

- The *curse of dimensionality*
- If vector has 10k unique tokens, bigram feature could yield 100 million possible bigrams and 1 trillion trigrams!
  - For each ‘document’!
- Linear algebra and statistics to the rescue
  - Dimensionality reduction techniques
  - Singular value decomposition (SVD), Multidimensional scaling (MDS), Principal component analysis (PCA)

4. Model Building

- Two main situations with data availability
  - When we have the 'answer' (a teacher): supervised
  - When we don't have annotated/labeled data: unsupervised
- Supervised model building
  - Regression – when predictor is numeric (rare for text)
  - Classification – when predictor is categorical
- Unsupervised modelling
  - Clustering algorithms (and dimensionality reduction)

4. Model Building



4. Model Building

- Language processing frameworks/toolkits
  - Python: `NLTK`, `Spacy`, `pattern`, `Textblob`, `Gensim`
  - Java: `OpenNLP`, `Stanford CoreNLP`, `GATE`, `Mallet`
- Machine learning frameworks/toolkits
  - Python: `scikit-learn`
  - Java: `weka`
- Miscellaneous
  - Big data: `spark mllib`
  - Deep learning: `tensorflow`, `theano` (`keras`)

Plan

- Data and Text Analytics
- Exploration and description vs inferencing
- Exploring text
- Inferencing with text
- Application

Plan

- Data and Text Analytics
- Exploration and description vs inferencing
- Exploring text
- Inferencing with text
- Application

Applications

- POS tagging, chunking and parsing
- Text classification, document summarization and document similarity
- Unsupervised document clustering, anomaly detection
- Word embedding, word sense disambiguation and topic modelling and sentiment analysis

Applications

- Example: Sentiment Analysis
  - Can be done using a sentiment lexicon (unsupervised) or using a manually labelled dataset
  - We use a dataset of IMDb movie reviews annotated with sentiment (positive/negative) by Andrew Maas et. al. from Stanford University
  - Available at: http://ai.stanford.edu/~amaas/data/sentiment/
    - download cut-down version locally: http://192.248.22.100/icter17/
  - So we use supervised learning (classification)

Applications

- The subset of this data we use has ~33k movie reviews
  - We first divide this into training (25k) and testing (balance) sets
  - Next we store the reviews and the sentiments separately
  - Then we 'normalize' the data as discussed before
  - Finally we learn/build an SVM classifier with the training data
  - We then predict the sentiments for the test reviews
  - And then compare these with the actual sentiments of test set
  - We finally interpret the results using a confusion matrix
  - https://drive.google.com/open?id=0B2oTxCSxrA2CSW9jd0RiODVfR3c

Applications

- Applying principles to other text classification tasks
  - e.g. categorizing news articles into different classes, recognizing hate speech or fake news etc.
- What if our data has no labels?
  - Document clustering using k-means, affinity propagation, hierarchical clustering, density-based methods
  - e.g. news categorization, analyzing those ‘open ended’ questions in questionnaires!

Summary

- Increasingly larger amounts of data we have access to is in the form of text
  - e.g. Fb, twitter, blogs, reviews, news, websites…
- However this text is usually highly *unstructured*
- We can *adapt* techniques we use for data analytics to analyze textual data
- Combining such *analytics* can help us automate higher level tasks

Summary

- Example: Chatbot
- Anatomy of a chatbot
  - Automatic speech recognition (ASR)
  - Intent classification
  - Semantic parsing
  - Dialog management
  - Response generation
  - Text to speech (TTS)
  - Miscellaneous topics: personality, emotion and affect
- arw@ucsc.cmb.ac.lk