Using MATLAB for Sentiment Analysis and Text Analytics

By Liliana Medina

MathWorks UK – Software Engineer
MATLAB Text Analytics Toolbox
Outline

- Sentiment Analysis
- Strings in MATLAB
- Introduction to Text Analytics Toolbox
- Using the toolbox for a sentiment analysis task
  - Overview
  - Step-by-step
  - Insights
- Additional capabilities and Resources
Sentiment Analysis

- Identify sentiment expressed in documents and social media
  - Microblogging platforms, news articles, company reports, e-mails

Applications
- Volatility and Risk analytics
- Inform trading strategies
- Correlate with stock movements
- Economics research
- Litigation
Strings
The better way to work with text

- Manipulate, compare, and store text data efficiently
  
  ```
  >> "image" + (1:3) + ".png"
  1×3 string array
  "image1.png"   "image2.png"   "image3.png"
  ```

- Simplified text manipulation functions
  - Example: Check if a string is contained within another string
    - Previously:
      ```
      if ~isempty(strfind(textdata,"Dog"))
      ```
    - Now:
      ```
      if contains(textdata,"Dog")
      ```

- Performance improvement
  - Up to 50x faster using `contains` with `string` than `strfind` with `cellstr`
  - Up to 2x memory savings using `string` over `cellstr`
Text Analytics Toolbox

Extract Value from Text Data

Access and Explore Data

Preprocess Data

Model and Derive Insights

- Word Docs
- PDF’s
- Text Files
- HTML
- Tables/Spreadsheets

- Stop Words
- Stemming
- Tokenization
- Find special tokens

- Bag of Words
- Bag of N-grams
- TF-IDF
- Word Embeddings

- Latent Dirichlet Allocation (LDA)
- Latent Semantic Analysis (LSA)
- Word clouds
- Text scatter plots

RT @wsv Dry, warm and sunny for most today. #weatherReport http://bit.ly/2vcZa8w

Dry warm sunny today

<table>
<thead>
<tr>
<th></th>
<th>cat</th>
<th>dog</th>
<th>run</th>
<th>two</th>
</tr>
</thead>
<tbody>
<tr>
<td>doc1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>doc2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Statistics & Machine Learning Toolbox

Neural Network Toolbox
Text Analytics Toolbox
A Quick Example

- Explore, manipulate, and transform text data
- Convert text into input for numerical analysis
- Visualize text

**Applications**
- **Sentiment Analysis**: discover sentiment in news, reports, e-mails
- Maintenance: identify hidden groups of issues in maintenance logs
- Document Classification: automatically tag unread documents (e.g. for triaging or routing)
Compare Sentiment of Tweets with Stock Prices

Goal

- Analyze the sentiment of tweets for several tech companies and compare the sentiment scores to stock prices

Approach

- Access data from Twitter
- Preprocess to clean-up text
- Identify domain-specific sentiment terms
- Map text to numbers with a word embedding
- Build a sentiment classification model
- Compare sentiment scores to prices
Workflow Overview

Tweets

Preprocess Text

Word Embedding Model

Create Target

Positive & Negative Word List

Create Predictors

Machine Learning Model

Predict Sentiment

predict(model,newTweet)

Sentiment
(neutral,positive,negative)
Access and Import Data

- Use **Datafeed Toolbox** to search Twitter for keywords and import tweets
- Extract the text of the tweets

```matlab
load myCreds
creds = twitter(creds.ConsumerKey, creds.ConsumerSecret,...
    creds.AccessToken, creds.AccessTokenSecret);

tweets = search(creds, 'MSFT', 'count', 100,...
    'include_entities', 'true', 'lang', 'en');

msft = tweets.Body.Data.statuses;
msfttweets = msft.Tweet;

ans =
100x1 string array

"In 2Q17 $AMZN AWS controlled 34% of cloud infrastructure market, topping $MSFT $IBM $GOO
"RT @infosecnews: Microsoft patch alert: Outstanding problems with recent updates - http
"$MSFT ready for breakout? Check it out on Chartmill https://t.co/USmAQN23dI #ainasm $%
"$MSFT is Up tall on day is stop"
"The Chartmill setup rating of $MSFT is on an all-time high https://t.co/UlyVEeBjXb #Ban
"Comparing Microsoft Corporation ($MSFT) &amp; Callidus Software ($CALL) https://t.co/eQ9SYL
"The Chartmill setup rating of $MSFT is on an all-time high https://t.co/yLRRNSkVWv #IPO
"$MSFT - Schmidt P J Investment Management Inc. Increases Holdings in Microsoft Corporat
"$GOOG holders keep selling down. Maybe our puts be worth something after all! $GOO $SP
"State Of Alaska Department Of Revenue Has Lowered By $378,200 Its Microsoft $MSFT Posi
"Microsoft $MSFT Stock Price Rose While 1st Global Advisors Lifted by $420,846 Its Holdi"
```

Search for more keywords: ‘GOOG’, ‘AMZN’, ‘APPL’, …
Preprocessing Text: Clean and Tokenize

- **Remove urls and html tags**
  - `eraseTags`
  - `eraseURLs`

- **Remove tickers with a regular expression**
  ```
  torep = {'(?:\S+[/\w_.]+)'}
  strip(regexp(replace(tweettext, torep, '')))
  ```

- **Split text into tokens to better support preprocessing and analysis**
  ```
  t = tokenizedDocument(allTweets);
  t(5:10)
  ```
  ```
  ans = 6x1 tokenizedDocument:
  [(3,1) 13 tokens: mu investments co. ltd. sells 30 shares of alphabet inc.
  (2,1) 10 tokens: ad $ are going to and away from uppgy #advertising
  (3,1) 17 tokens: big bullish unusual option activity detected : , , , - big bearish : , , ,
  (4,1) 10 tokens: report: apple to build data center in iowa:
  (5,1) 13 tokens: rt @theyflynws: report: apple to build data center in iowa:
  (6,1) 13 tokens: rt @theyflynws: report: apple to build data center in iowa:
  ```
Preprocessing Text: Clean and Tokenize

- Find and remove hashtags, at-mentions, digits, punctuation, and stopwords

Stop words: “the”, “a”, “an”, ...
- English stopWords list included
- removeWords
- erasePunctuation
Preprocessing Text: Explore and Visualize

- Explore the text with **wordcloud charts**:
  - Use a **bagOfWords** model to find the most frequent words

Further cleaning:
- Remove very frequent words
- Remove very short words
Modelling: Use Sentiment Word Lists

- Import external list of words
  - Positive and negative words
  - **Specific** to finance domain ([Loughran & McDonald](https://www.jstor.org/stable/pdf/2615774.pdf))
  - Useful to create a target sentiment variable

```matlab
>> domainWords = load('domainSentimentWords.mat');
```
```
>> domainWords.pos(1:10) >> domainWords.neg(1:10)
```
```
"a+"
"able"
"abound"
"abounds"
"abundance"
"abundant"
"accessable"
"accessible"
"acclaim"
"acclaimed"
```
```
"2-faced"
"2-faces"
"abandon"
"abandoned"
"abandoning"
"abandonment"
"abandonments"
"abandons"
"abdicated"
"abdicates"
```
Modelling: Use Sentiment Word Lists

- Build the sentiment target variable
  - Subtract number of negative words from number of positive words
    $$\text{target} = \text{number of positives} - \text{number of negatives}$$
  - Three sentiment classes
    - target > 0 → tweet belongs to positive class
    - target = 0 → tweet belongs to neutral class
    - target < 0 → tweet belongs to negative class

Are Stocks **Overweight** In Active Funds At **Risk** For **Correction**?

$\text{MSFT}$ moderately **bullish** in premarket - had a **good** day yesterday - testing **important** breakout areas
Modelling: Word Embeddings

- Load or train a `wordEmbedding` model
  - Each word represented by a numeric vector
  - Embeddings preserve information about word order and context
  - Vector operations enable discovery of **word similarities** and **analogies**

**fastTextWordEmbedding** support package precomputed vectors for 16B English tokens
Modelling: Word Embeddings

- Use the word embedding model to create a vector per tweet
  - Take advantage of the word2vec function
  - Combine individual word vectors into one

- Average the individual word vectors in the tweet

```matlab
>> emb = fastTextWordEmbedding;
>> tweetWords = ["my" "example" "tweet"];
>> tweetVector = mean(word2vec(emb,tweetWords));
```
Modelling: Train a Sentiment Classification Model

Requires: Statistics & Machine Learning Toolbox

>> TrainData = array2table(tweetVectors);
>> TrainData.ClassResponse = target;
>> classificationLearner

- Cubic SVM model: 96.5%
Derive Insights: Score Tweets for Sentiment

1. Use model to **predict** sentiment scores for new tweets

   >> newTweet = "$AMZN: Here’s Amazon’s Plan to Make Students Better Writers: https://t.co/SVfi0eDHt0"

2. Preprocess new tweet

   tokenizedDocument:

   7 tokens: heres amazons plan make students better writers

3. Derive a vector with the **wordEmbedding** model

   >> tweetVector = mean(word2vec(emb,string(t)));

4. Use classification model to predict the sentiment of the new tweet

   >> pred = predict(svmModel,tweetVector)

   pred =
   1  → positive!
Intraday price data
  – **use synchronize**
• Smooth sentiment scores
• Align with price data
Key Takeaways

- Text data is everywhere, and contains valuable information
- Text Analytics Toolbox has tools to help you extract the signal from the noise
- Combine text with other data sources to take advantage of all your data
Text Analytics with Deep Neural Networks

- Use **deep learning** to model text data
  - **Neural Network Toolbox**
  - Classification and regression with **LSTMs**
  - **MATLAB Deep Learning Onramp**
    - Free two hour DL tutorial
Additional Text Analytics Toolbox Capabilities

Extract text from different file formats

- Extract text from pdf files, word documents, and HTML

- `extractFileText`
- `extractHTMLText`
- `readPDFFormData`
Additional Text Analytics Toolbox Capabilities

Bag of words, Bag of n-grams, and TF-IDF

- Discover **words** and **multi-words**

### Weather Reports: Trigrams

<table>
<thead>
<tr>
<th>Ngram</th>
<th>Count</th>
<th>NgramLength</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;inches&quot; &quot;of&quot; &quot;snow&quot;</td>
<td>2075</td>
<td>3</td>
</tr>
<tr>
<td>&quot;across&quot; &quot;the&quot; &quot;county&quot;</td>
<td>1316</td>
<td>3</td>
</tr>
<tr>
<td>&quot;were&quot; &quot;blown&quot; &quot;down&quot;</td>
<td>1189</td>
<td>3</td>
</tr>
<tr>
<td>&quot;wind&quot; &quot;gust&quot; &quot;of&quot;</td>
<td>934</td>
<td>3</td>
</tr>
<tr>
<td>&quot;A&quot; &quot;tree&quot; &quot;was&quot;</td>
<td>860</td>
<td>3</td>
</tr>
<tr>
<td>&quot;the&quot; &quot;intersection&quot; &quot;of&quot;</td>
<td>612</td>
<td>3</td>
</tr>
<tr>
<td>&quot;inches&quot; &quot;of&quot; &quot;rain&quot;</td>
<td>739</td>
<td>3</td>
</tr>
<tr>
<td>&quot;hail&quot; &quot;was&quot; &quot;reported&quot;</td>
<td>645</td>
<td>3</td>
</tr>
<tr>
<td>&quot;was&quot; &quot;blown&quot; &quot;down&quot;</td>
<td>638</td>
<td>3</td>
</tr>
<tr>
<td>&quot;and&quot; &quot;power&quot; &quot;lines&quot;</td>
<td>631</td>
<td>3</td>
</tr>
</tbody>
</table>
Additional Text Analytics Toolbox Capabilities

LSA and LDA

- **Unsupervised machine learning** techniques
- Well-suited to **wide** feature matrices
- Both can be used for **dimensionality reduction**
- LDA is also useful for **topic modeling**
  - Discover **latent topics** in the data
Learn More – MATLAB Discovery Pages for Text

- **Sentiment Analysis**: Analyze and predict sentiment with machine learning
- **Natural Language Processing**: Data analytics with human language data
- **Text Mining with MATLAB**: Deriving insight from text data

Ask the Expert: Sarah Palfreyman
Learn More – Documentation Examples

https://www.mathworks.com/help/textanalytics/examples.html
Learn More about Word Embeddings

Math with Words – Word Embeddings with MATLAB and Text Analytics Toolbox

What is President Trump tweeting about that gets our attention?

Questions?