Business Analytics for Unstructured Data

Carla Bonato Marcolin

Performance Augmentation Lab
Oxford Brookes University

October 2017

Agenda

1. Introduction
2. Data
3. Classifier
4. Next Steps

Introduction

Text data is expected to become more and more present:

- Social Media
- Eletronic Word-of-Mouth (eWOM) (Tang and Guo, 2015)
- Text as a sensor for measuring perception (Zhao, 2013)
- Mobile Technologies

Objective

Although in principle many models have been developed for the task of analyze text data, it still too hard to use the results in practice for decision-making process inside organizations.

- Propose a classifier, based on Service Quality model, for hotels, having as input travelers comments from TripAdvisor
- Help managers to analyze customer's perception
- Understand strong and weak points

Measuring Quality

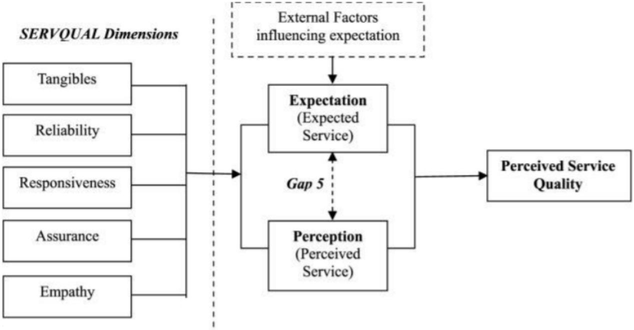

SERVQUAL Dimensions
Tangibles
Reliability
Responsiveness
Assurance
Empathy
External Factors influencing expectation
Expectation (Expected Service)
Gap 5
Perception (Perceived Service)
Perceived Service Quality

Figure: SERVQUAL Quality Model Adapted from Kumar et al. (2009)

Advantages

If there is a scale, why not simple ask people?

- Low response rate in questionnaires
- Honesty
- Willingness to share data: with the company or with other customers?
- Cost to apply, cost to analyze
- *Why* the customer is (or is not) satisfied is more important then *How much*

Data Collection

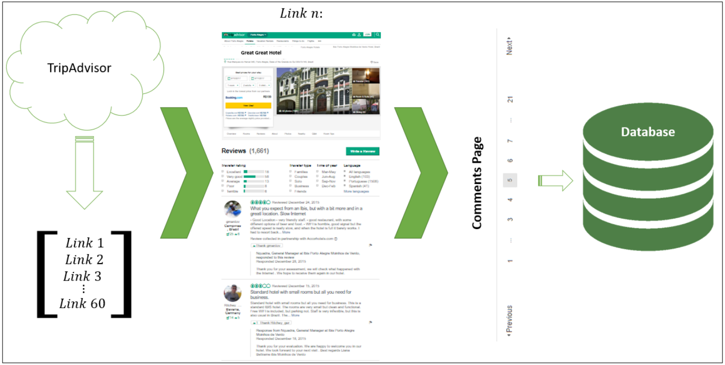


Figure: WebScrapper

Pre-Processing

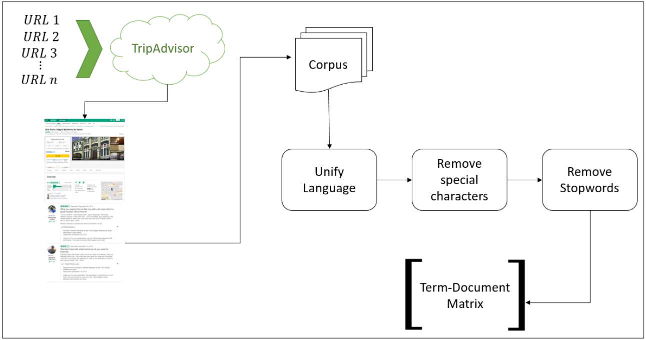


Figure: Pre-Processing

Text Classification

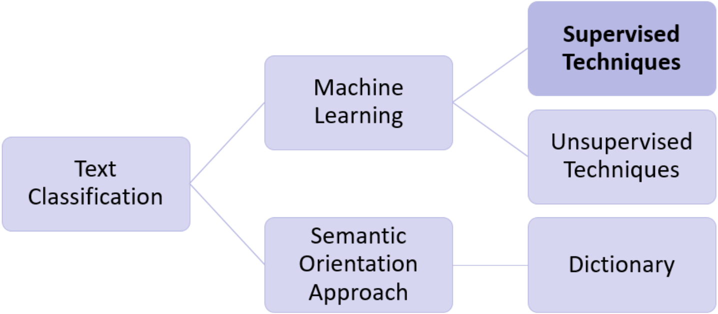


Figure: Text Classification Methods

Process

As supervised methods, we need labeled data. For that:

1. Two human classifiers labeled the data
2. One-round discussion to compare
3. Another independent classification round
4. **Second-round discussion**
5. Develop a protocol to train other classifiers
6. Use the data to train and test a classifier
7. SVM and Naive Bayes: evidence of good performance (Choi and Lee, 2017; Collingwood et al., 2013)

SVM

What is Support Vector Machine?

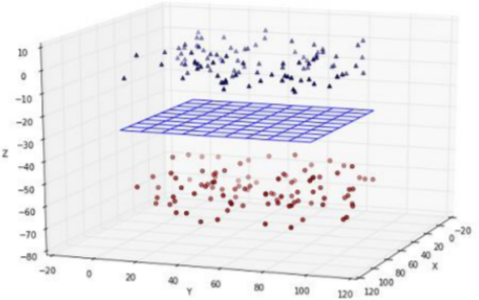


Figure: SVM (Kowalczyk, 2017)

Perceptron

How to separate data? Perceptron Algorithm!

Simple, easy algorithm, dating back from 60s, works with a simple hypothesis:

$$h(\mathbf{x}_i) = \begin{cases} +1 & \text{if } \mathbf{w} \cdot \mathbf{x_i} + b \geq 0 \\ -1 & \text{if } \mathbf{w} \cdot \mathbf{x_i} + b < 0 \end{cases}$$

Perceptron

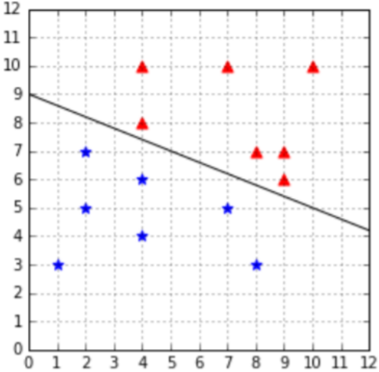


Figure: Perceptron (Kowalczyk, 2017)

Perceptron

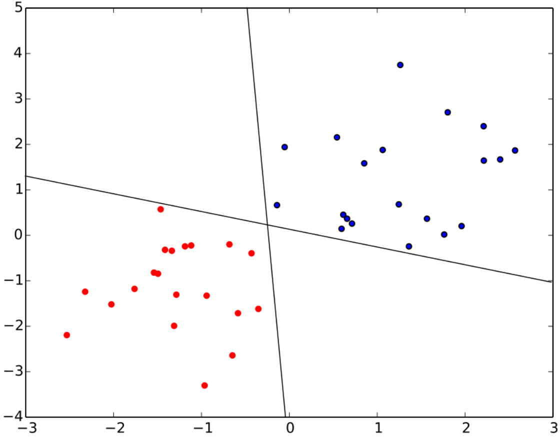


Figure: Perceptron (2017)

SVM

If the hyperplane exists, Perceptron may find different solutions for the same dataset. This is a problem since it will tend to generalize poorly when given new data (our objective).

SVM can be seen as an optimization problem among all hyperplanes that correctly separates the data, with the largest margin (as far as possible from data points from each category).

SVM

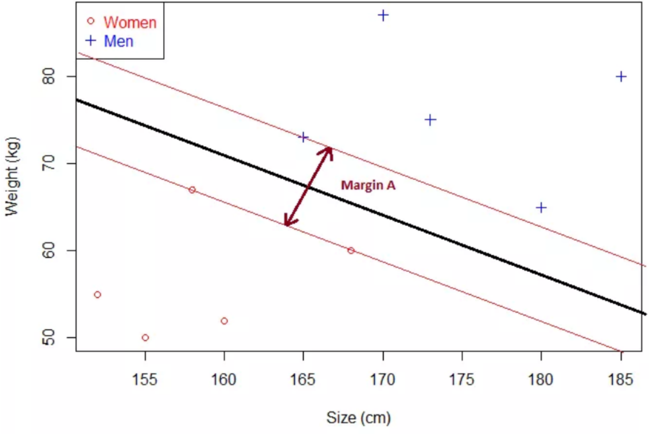
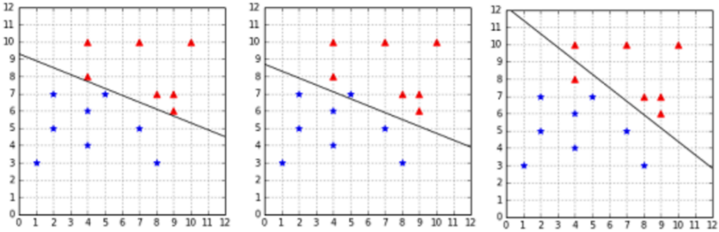


Figure: Margin in SVM (Kowalczyk, 2017)

SVM

Soft Margin SVM adds a new variable to the problem allowing some error in classification:



$C = +\infty$ $C = 1$ $C = 0.01$

Figure: C Variable Effect (Kowalczyk, 2017)

SVM trick: Kernel

Kernel methods allow to compute the dot product without having to transform the vector. The most popular are:

- Linear Kernel (indicated for text classification (Choi and Lee, 2017))
- Polynomial Kernel
- Gaussian Kernel
- Graph Kernel

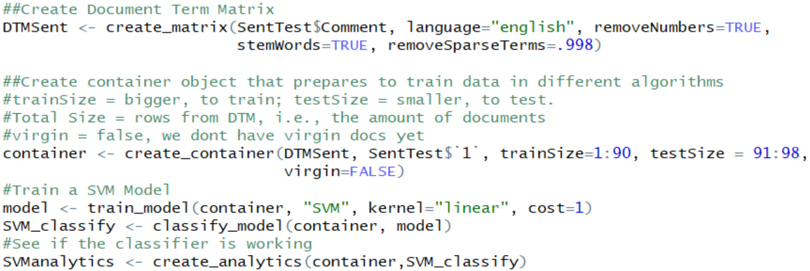
SVM in R

The most popular SVM implementation in R is from *e1071* library. For text, the package *RTextTools* can be very useful.

Some interesting features:

- Easy to work with *simple_triplet_matrix* object
- Four different kernel implementation
- Cost variable (C)
- Analytics!

SVM in R



```
##Create Document Term Matrix
DTMSent <- create_matrix(SentTest$Comment, language="english", removeNumbers=TRUE,
                         stemWords=TRUE, removeSparseTerms=.998)

##Create container object that prepares to train data in different algorithms
#trainSize = bigger, to train; testSize = smaller, to test.
#Total Size = rows from DTM, i.e., the amount of documents
#virgin = false, we dont have virgin docs yet
container <- create_container(DTMSent, SentTest$`1`, trainSize=1:90, testSize = 91:98,
                              virgin=FALSE)
#Train a SVM Model
model <- train_model(container, "SVM", kernel="linear", cost=1)
SVM_classify <- classify_model(container, model)
#See if the classifier is working
SVManalytics <- create_analytics(container,SVM_classify)
```

Naive Bayes

Naive Bayes is a probabilistic classifier that assumes independence between features, as well as data completeness.
The goal is to find the most likely class for a document.

$$P(c|d) = P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

Where:
- $P(t_k|c)$ is the conditional probability of term t_k occurring in a document of class c.
- $P(c)$ is the prior probability of document occurring in class c.
- For each document we use the terms $t1, t2, \ldots, t_{n_d}$ that have a higher prior probability.

Naive Bayes

As for each position $1 \leq k \leq n_d$ many conditional probabilities are multiplied, the computation is performed with logarithms of probabilities:

$$P(c|d) = argmax[logP(c) + \sum_{1 \leq k \leq n_d} P(t_k|c)]$$

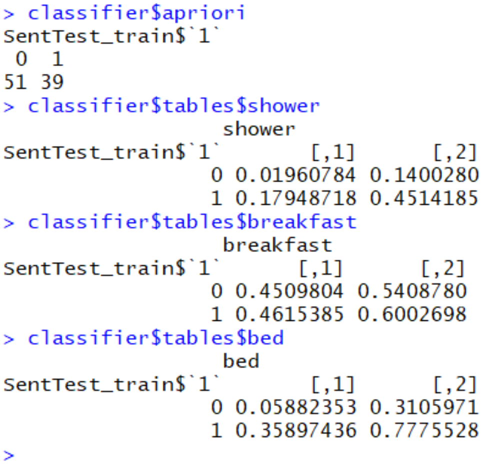
Naive Bayes in R

There are different implementations in R (and growing).
Most used are *naiveBayes* from *e1071* package and *NaiveBayes* from *klaR* package.

Some interesting features are:

- *apriori* shows the class distribution among the data
- *tables* are Gaussian Distributions for each predictor variable (each word)

Naive Bayes in R



```
> classifier$apriori
SentTest_train$`1`
 0  1 
51 39 
> classifier$tables$shower
                  shower
SentTest_train$`1`       [,1]      [,2]
                 0 0.01960784 0.1400280
                 1 0.17948718 0.4514185
> classifier$tables$breakfast
                  breakfast
SentTest_train$`1`      [,1]      [,2]
                 0 0.4509804 0.5408780
                 1 0.4615385 0.6002698
> classifier$tables$bed
                  bed
SentTest_train$`1`       [,1]      [,2]
                 0 0.05882353 0.3105971
                 1 0.35897436 0.7775528
>
```

Naive Bayes in R

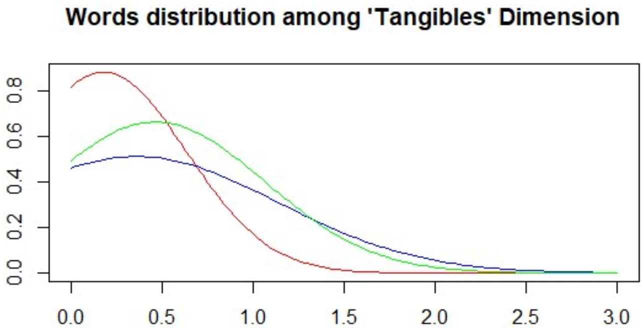


Figure: Shower, Breakfast and Bed word distribution

Next Steps

After labeling process, next steps are:

- Train and test the classifiers
- Understand main topics from each dimension with **Topic Modeling**
- Apply **Sentiment Analysis** to rate dimensions

Topic Modeling

Topic modeling works with a main idea that there exist a structure, that is non-observable, behind documents and terms.

In addition, that this structure is capable to **better represent** the main connexions among text data.

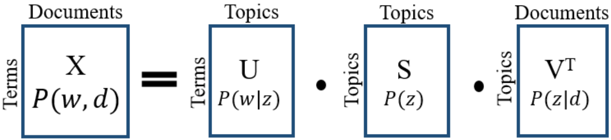


Figure: Topic Modeling Intuition

Sentiment Analysis

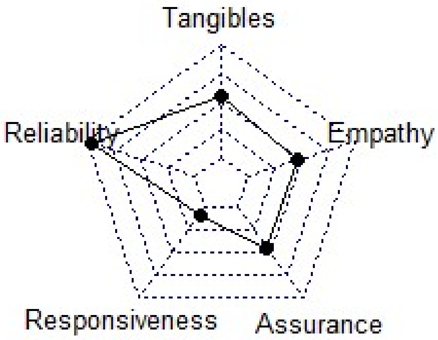


Figure: Sentiment Analysis

Future?

Some aspirations:

- Build a tool to allow managers to import their comments just with TripAdvisor URL
- Analyze service quality perception from other public and private organizations (touristic points, restaurants, museums) from the same data source (TripAdvisor)
- SERVQUAL have been adapted to different fields (EDUQUAL, HEALTHQUAL,ARTSQUAL,...). The same methodology can be used to represent quality perception in these fields.

Thank You!

Questions? Comments?

Business Analytics for Unstructured Data

Carla Bonato Marcolin
cbmarcolin@gmail.com

Choi, Y. and Lee, H. (2017). Data properties and the performance of sentiment classification for electronic commerce applications. *Information Systems Frontiers*, pages 1–20.

Collingwood, L., Jurka, T., Boydstun, A. E., Grossman, E., van Atteveldt, W., et al. (2013). Rtexttools: A supervised learning package for text classification.

Kowalczyk, A. (2017). *Support Vector Machines Succinctly*. Syncfusion.

Kumar, M., Tat Kee, F., and Taap Manshor, A. (2009). Determining the relative importance of critical factors in delivering service quality of banks: an application of dominance analysis in servqual model. *Managing Service Quality: An International Journal*, 19(2):211–228.

Perceptron (2017). Perceptron — Wikipedia, the free encyclopedia. [Online; accessed 27-October-2017].

Tang, C. and Guo, L. (2015). Digging for gold with a simple tool: Validating text mining in studying electronic word-of-mouth (ewom) communication. *Marketing Letters*, 26(1):67–80.

Zhao, Y. (2013). *R and data mining: Examples and case studies*. Academic Press.