40.016 The Analytics Edge

Text Analytics (Part 2)

Stefano Galelli

SUTD

Term 6, 2020

Outline

- Text Analytics
- The Enron Corpus
- Modelling process
- Naive Bayes classifier

Text Analytics

- Process of deriving high-quality information from text
- Common tasks are (1) text categorization and summarization, and (2) sentiment analysis
- Text analytics build on several steps (e.g., processing the text, finding patterns, learning a classification model)

The Enron Corpus

- Enron Corporation was an American energy and services company established in 1985 and based in Houston
- Before its bankruptcy in 2001, Enron employed nearly 30,000 staff, with claimed revenues of about $100 billion (in 2000)
- By the end of 2001, it was revealed that the reported financial conditions were based on systematic accounting fraud (the *Enron scandal*)

The Enron Corpus

- Publicly-available dataset of the email messages sent or received by about 150 Enron senior managers
- Created by the Federal Energy Regulatory Commission during its investigation
- 'Rare' kind of dataset (such collections are often bond by privacy laws)

The Enron Corpus

Our goal: identify which emails are *responsive* to content related to *energy bids*. Terms like "electricity bid" or "energy schedule" may have enough predictive power to learn an accurate classifier.

To this purpose, we will use **predictive coding**.

Modelling process

Modelling workflow

- Pre-processing
  1. Convert text to lower case
  2. Remove stopwords
  3. Remove punctuation
  4. Stemming
  5. Create DTM
  6. Removing sparse terms
- Preparing the DTM for model learning
- Train and test a classifier

Naive Bayes classifier

Recall (Bayes Theorem)

Naive Bayes classifier

Recall (Bayes Theorem)

Naive Bayes classifier

Recall (Chain rule of conditional probability)

Naive Bayes classifier

Given a problem instance to be classified, represented by a vector $\{x_1, \ldots, x_p\}$ of predictors, a Naive Bayes classifier assigns to this instance probabilities $P(C_k|x_1, \ldots, x_p)$ for each of the K possible classes or outcomes C_k.

Using **Bayes' rule**, let's rewrite this as:

$$P(C_k|x_1, \ldots, x_p) = \frac{P(C_k) \cdot P(x_1, \ldots, x_p|C_k)}{P(x_1, \ldots, x_p)}.$$

Naive Bayes classifier

Let's focus on the numerator $P(C_k) \cdot P(x_1, \ldots, x_p | C_k)$, which is equal to the joint probability $P(C_k, x_1, \ldots, x_p) = P(x_1, \ldots, x_p, C_k)$. Using the **chain rule**, we can write:

Naive Bayes classifier

Now, we make a **naive hypothesis of conditional independence of the features**, that is, x_i is conditionally independent of every other feature x_j (with $i \neq j$). With this hypothesis in place, we can write:

Naive Bayes classifier

Recalling that $P(x_1, \ldots, x_p)$ is a constant Z, we can finally write:

$$P(C_k|x_1, \ldots, x_p) = \frac{1}{Z} P(C_k) \prod_{i=1}^{p} P(x_i|C_k).$$

Naive Bayes classifier

Recalling that $P(x_1, \ldots, x_p)$ is a constant Z, we can finally write:

$$P(C_k|x_1, \ldots, x_p) = \frac{1}{Z} P(C_k) \prod_{i=1}^{p} P(x_i|C_k).$$

The prediction rule is to assign a class k such that:

$$\arg \max_{k=1,\ldots,K} P(C_k) \prod_{i=1}^{p} P(x_i|C_k).$$

Naive Bayes classifier

To estimate the $P(x_i|C_k)$, one can, for example, use a Gaussian distribution. Denoting with μ_{ki} and σ^2_{ki} the mean and variance of x_i in class k (i.e., x_{ki}), we get:

$$P(x_i|C_k) = \frac{1}{\sqrt{2\pi\sigma^2_{ki}}} \; e^{-\frac{(x_{ki}-\mu_{ki})^2}{2\sigma^2_{ki}}}.$$

Naive Bayes classifier

Advantages:

- Works with binary and multi-class problems
- Computationally efficient

Disadvantages

- They are not very accurate
- Assumption of conditional independence of the features
- (If a categorical variable has a category in test data set, which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as Zero Frequency.)

References

- Teaching notes.