LING 410X: Language as Data

Semester: Spring '17

Instructor: Sowmya Vajjala

Iowa State University, USA

27 February 2018

Outline

- Assignment 3 discussion
- Text classification: Example
- Assignment 4 description
- Note: The discussion forums is how I evaluate active participation (5% of your grade). So, if you don't post when I ask, essentially, you are telling me "I don't care about that 5%".

Assignment 3 discussion

(Discussion in a short document)

Text Classification - how it is done

Steps in text classification

- Step 1: We have a classification problem, and a dataset containing examples for that
  e.g., spam classification: 100s of examples for spam, and non-spam emails.

Steps in text classification

- Step 1: We have a classification problem, and a dataset containing examples for that
  e.g., spam classification: 100s of examples for spam, and non-spam emails.
- Step 2: Read that data into R
- Step 3: Create a document term matrix (DTM), Also, keep track of what each document's category is

Steps in text classification

- Step 1: We have a classification problem, and a dataset containing examples for that
  e.g., spam classification: 100s of examples for spam, and non-spam emails.
- Step 2: Read that data into R
- Step 3: Create a document term matrix (DTM), Also, keep track of what each document's category is
- Step 4: Use an existing classification algorithm - give DTM as input. (training)
- Step 5: Use the classifier (output of Step 4) on new texts, to check how it is doing (testing)

Steps in text classification

- Step 1: We have a classification problem, and a dataset containing examples for that
  e.g., spam classification: 100s of examples for spam, and non-spam emails.
- Step 2: Read that data into R
- Step 3: Create a document term matrix (DTM), Also, keep track of what each document's category is
- Step 4: Use an existing classification algorithm - give DTM as input. (training)
- Step 5: Use the classifier (output of Step 4) on new texts, to check how it is doing (testing)
- Step 6: Repeat these two steps until you are happy, changing different settings. Once you are convinced, you can stop, and start actually using it.

Sentiment classification - my dataset

Movie reviews dataset.
source:`http://ai.stanford.edu/~amaas/data/sentiment/`

tm - library

- used for doing various text mining tasks in R
- can use it for just exploring data, do text classification, topic modeling, clustering, visualization etc.
- we will continue using this for the rest of this class.
- install.packages("tm")
- install.packages("SnowballC")
- install.packages("e1071") - to continue with the svm algorithm the author used.

Working with tm: loading files from a directory and pre-processing

tm has a lot of built in pre-processing options

```
library(tm)
cleanCorpus <- function(folder)
{
  corpus <- VCorpus(DirSource(folder))
  corpus <- tm_map(corpus, removeNumbers)
  corpus <- tm_map(corpus, content_transformer(tolower))
  corpus <- tm_map(corpus, removePunctuation)
  corpus <- tm_map(corpus, removeWords, stopwords("english"))
  #corpus <- tm_map(corpus,stemDocument)
  return(corpus)
}
```

(there are more)

Corpus organization

- This dataset has a folder structure as follows:
  1. a folder called train, in which there is a folder called pos, and another called neg.
  2. a folder called test, with same structure as above
  3. the pos and neg folders contain examples of positive reviews and negative reviews.

Corpus pre-processing

Because of the way that data is organized, I have to call cleanCorpus function for each folder, and combine them.

```
#Create training corpus
corpus_train_pos <- cleanCorpus("data/train/pos")
corpus_train_neg <- cleanCorpus("data/train/neg")
corpus_train <- c(corpus_train_pos,corpus_train_neg)
training_size <- length(corpus_train_pos) + length(corpus_train_neg)

#Create testing corpus
corpus_test_pos <- cleanCorpus("data/test/pos")
corpus_test_neg <- cleanCorpus("data/test/neg")
corpus_test <- c(corpus_test_pos,corpus_test_neg)
```

Building a Document-Term matrix (DTM)

- Since we here have both training data, and some testing data to validate our work, let us build a DTM combining all data.

```
fullcorpus <- c(corpus_train,corpus_test)
dtm_together <- DocumentTermMatrix(fullcorpus)
```

-yes, only one line!! (There are several other possible arguments to this function though).

Note: In real world, we actually will not see the actual test data until we start using the classifier. So, we don't build a DTM for test data (we cannot!). I am just using this example as illustration.

Adding sentiment information to this matrix

Once you have this dtm:

- Convert the dtm to data frame (because the svm classifier we use wants a data frame.

```
dtm_together_df <- as.data.frame.matrix(dtm_together)
```

- extra column to indicate the sentiment (positive/negative) (Why? Isn't the task that of knowing this?)

Adding sentiment information to this matrix

Once you have this dtm:

- Convert the dtm to data frame (because the svm classifier we use wants a data frame.

```
dtm_together_df <- as.data.frame.matrix(dtm_together)
```

- extra column to indicate the sentiment (positive/negative) (Why? Isn't the task that of knowing this?)

```
labels <-  c(rep("positive",length(corpus_train_pos)),
             rep("negative",length(corpus_train_neg)),
             rep("positive",length(corpus_test_pos)),
             rep("negative",length(corpus_test_neg)))
dtm_together_df <- cbind(dtm_together_df,labels)
```

Building a sentiment classifier

- Split the entire data back into training and testing again (Why?)

```
max_col <- ncol(dtm_together_df)
train <- dtm_together_df[1:training_size,1:max_col-1]
#last column is the category
class_train <- dtm_together_df[1:training_size,max_col]

test <- dtm_together_df[training_size:length(fullcorpus),1:max_col-1]
class_test <- dtm_together_df[training_size:length(fullcorpus),max_col]
```

- use svm() function like in the textbook, to "train" to classify

```
model.svm <- svm(train, class_train)
```

-Yes, the two important parts of text classification (building a DTM, and building a classifier - they are just one liners!

Using the sentiment classifier

- predict with the svm function as in the textbook.

```
final.result <- predict(model.svm, test)

#compare predictions with actual values:
predictions <- cbind(as.data.frame(final.result), class_test)
predictions

#evalute how good this classifier is:
table(final.result, class_test)
```

What now?

- At this point, we have a classifier, which can be used to analyse sentiment of new movie reviews.
- However, it also seems to be pretty bad.

What now?

- At this point, we have a classifier, which can be used to analyse sentiment of new movie reviews.
- However, it also seems to be pretty bad.
- How can we improve? - 3 ways, primarily
  1. DTM: Should we consider all terms inside a DTM? How about filtering? One intuition: Words that are two infrequent are perhaps not useful.
  2. Classification (1): I used svm. There may be some options within that function, I did not explore
  3. Classification (2): svm is not the only one around, I can look for other classification algorithms in R.
  4. Increase the number of training examples.

Removing sparse terms

```
#original dtm
dtm_together <- DocumentTermMatrix(fullcorpus)

#dtm after removing sparse terms
dtm_together_2 <- removeSparseTerms(DocumentTermMatrix(fullcorpus), sparse=0.7)
#this removes 70% of the sparse terms. So, number of columns in dtm is also
#drastically reduced!

#inspect() function in tm is useful to see the differences:
inspect(dtm_together)
inspect(dtm_together_2)
```

Increasing training data

- I have a larger version of the data set with more examples (around 1000 examples per category)
- We can repeat this process, just changing the corpus folders in the beginning.
- The actual dataset has around 12000 examples per category.

Using a different classifier

- I leave that part for your explorations!

Worked out example

ClassificationWithTM.R on Canvas.

Assignment 4 description

(check on Canvas)

Next Class

- other corpus analyses tm supports (associations between words etc)
- Practicals with tm.
- Discussion on the practicals.
- TODO: go through the slides, go through the tm vignettes document online, and post your questions and comments on the forum for today!
  `https://cran.r-project.org/web/packages/tm/vignettes/tm.pdf`

Discussion about the review document I uploaded last week