STAT665 Final Project: Comparison of Machine Learning Algorithms for Language Classification

Rose Brewin, Emily Dodwell, Samantha Emanuele

May 5, 2015

Contents

1 Introduction

In this paper, we explore various machine learning classification algorithms to determine the most effective and computationally efficient method to identify the language in which a sentence has been written (as English or other), by recognition of letter frequency. Such a classification mechanism has many relevant applications, including web services that identify when text on a webpage must be translated to the user's default language. Machine learning methods are advantageous for such a task due to their lack of reliance on a language dictionary. That is, an algorithm's success does not depend on prior knowledge of the given language, including definitions, grammar structures, and correct use of accents and punctuation.

Given the considered algorithms' lack of reliance on a single "dictionary", our choice of a best method may then be extended to consider how platforms that operate in different contexts may use language differently. For example, *The New York Times* maintains an extensive language styleguide that has the potential to differentiate the formality of its articles from more colloquial interactions, such as social media posts on Twitter.

2 Methodology for Data Collection and Model Preparation

2.1 Data Collection

We first consider the observed frequency of twenty-seven characters; specifically the twenty-six letters of the Roman alphabet and a blank space.

To train the various classfication algorithms considered, we found from the online Christian Classics Ethereal Library plain text documents of the Bible in various languages: English, French, Spanish, Latin, German, Dutch, and Italian[1]. Our decision to use the Bible was based on its status as a universal text, and it therefore acts as a sort of control among the languages considered. The accessibility of plain text Bibles written in these languages informed their choosing, and because the goal of our model is to identify English sentences from non-English sentences, we were interested in a mixture of languages that are Latin derivatives.

2.2 Data Cleaning

For each of these text documents, we cleaned them in the following manner:

- Scanned the text document into R and saved each new line of text as an individual observation in a list.
- Converted all letters to lowercase, split each observation into a vector of individual characters, and removed any character (including any accented letter) that is not a space or letter in the Roman alphabet.
- Created a 1200×28 data frame of training data, where the empirical distribution of the 27 characters (26 letters and a space) in each observation is a row. The 28th column is a language indicator, where 0 represents that the observation was originally written in English and 1 denotes another language. The first 600 rows are randomly chosen lines from the English language Bible, and the remaining rows consist of 100 randomly chosen lines from each of the other six Bibles.

We demonstrate our methodology with a simple example:

[1] *The Holy Bible* (various lanuage versions)

```
example <- "Machine Learning is ~*SuPeR FuN*~ and we learned so much :)
We hope you enjoy our project!"

myletters <- c(letters, " ")

clean <- function(string){
  vec <- tolower(string)
  vec <- unlist(strsplit(vec, split=""))
  vec <- vec[vec %in% myletters]
  return(vec)
}

cleaned <- clean(example); print(cleaned)
```

```
##  [1] "m" "a" "c" "h" "i" "n" "e" " " "l" "e" "a" "r" "n" "i" "n" "g" " "
## [18] "i" "s" " " "s" "u" "p" "e" "r" " " "f" "u" "n" " " "a" "n" "d" " "
## [35] "w" "e" " " "l" "e" "a" "r" "n" "e" "d" " " "s" "o" " " "m" "u" "c"
## [52] "h" " " " " "w" "e" " " "h" "o" "p" "e" " " "y" "o" "u" " " "e" "n"
## [69] "j" "o" "y" " " "o" "u" "r" " " "p" "r" "o" "j" "e" "c" "t"
```

```
features <- function(vec) {
  vec <- as.numeric(sapply(myletters, FUN = function(x){round(mean(vec==x),2)}))
  vec[is.na(vec)] <- 0
  dat <- as.data.frame(t(vec))
  names(dat) <- myletters
  return(dat)
}
```

Note that the `clean` function removed characters *, ~, :,), and ! from the original sentence. Applying the function `features` to the vector `cleaned` returns the following relative frequency distribution:

a	b	c	d	e	f	g	h	i	j	k	l	m
0.05	0	0.04	0.02	0.12	0.01	0.01	0.04	0.04	0.02	0	0.02	0.02

n	o	p	q	r	s	t	u	v	w	x	y	z	
0.08	0.07	0.04	0	0.06	0.04	0.01	0.06	0	0.02	0	0.02	0	0.19

We see that the blank space character appears most often; specifically there are 16 spaces in the cleaned vector that contains 83 characters total. The quotient of these values returns the observed relative frequency of 0.19 in the table above.

2.3 Models Considered

To accomplish our stated purpose, we consider four classification algorithms:

- Logistic Regression

- Support Vector Machine (SVM) with each of the following kernels:
  - Linear
  - Gaussian
  - Kullback-Leibler

To assess which of these models is the “best” at correctly classifying Bible sentences as having been written in English or another language, we evaluate the predictive ability of each model by its misclassification rate when applied to a test set of 300 randomly-selected English Bible sentences and 300 randomly-selected sentences from the other Bibles (constructed in a manner similar to that described for the training data). We also make note of the amount of time required to train each model as a measure of its computational efficiency.

3 Logistic Regression

We first consider a logistic regression model as a baseline against which to compare other machine learning algorithms due to its simplicity and ubiquity as a classification device for a binary categorical reponse variable. Because the fitted values of a logitistic regression model are easily interpretable probabilities, this model also enables us to identify those lines of test data that the model incorrectly classifies or otherwise considers somewhat ambiguous.

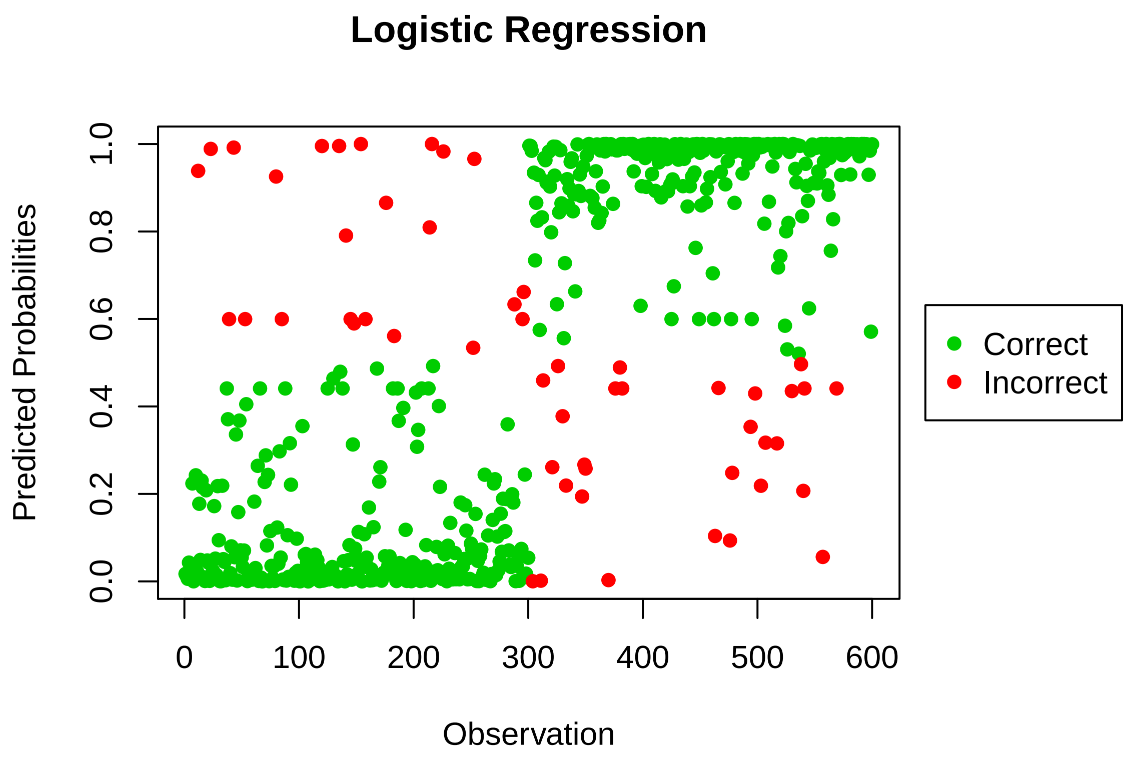


Figure 1: Predictions of Logistic Regression Model

The logistic regression model is quickly trained and has misclassification rate of 8.83% on the test set; we note that this provides a very good prediction of a sentence’s language.

Figure 1 above shows the predictions for each observation from the logisitic regression model. Each red point represents a misclassified observation, while each green point represents a correctly classified observation (this color convention will be used throughout the remainder of our paper). Keep in mind when studying the plot of predicted probabilities above that our test set is structured such that the first 300 observations are

English. This figure therefore demonstrates the logistic regression model's success at correctly classifying a majority of the observations.

The logistic regression model incorrectly classifies 25 lines from the English Bible; the complete list of original observations appears in Appendix A. It is understandable that the model stumbled when presented with observations that consisted of only a single word, such as "die", which is also German for "the", or "violence", which has the same spelling and meaning in French. Some of the other misclassified sentences contain phonetic spelling with Roman letters of Hebrew words and names, such as "Hazezontamar", "Japheth", and "Mizraim"; they are therefore not recognized as English.

4 Support Vector Machine

4.1 Linear Kernel

We first consider a simple Support Vector Machine with a linear kernel:

$$k(x, y) = \sum_{i=1}^{n} x_i y_i$$

The use of a linear kernel substantially restricts the form of the resulting decision boundary, but the reduction in complexity has the potential to provide better predictive results on a test set than a more complicated model, such as a Gaussian SVM.

We present our code for this SVM to demonstrate our use of the `ksvm` function in the `kernlab` package:

```r
# Model and prediction
then <- Sys.time()
sv.l <- ksvm(language ~ ., data=x, kernel="vanilladot", scaled=F, type="C-svc")
now <- Sys.time()
pr.svl <- predict(sv.l, newdata = y, type = "response")

# Misclassification rate
mis.svl <- round(mean(pr.svl != y$language)*100, 2)

# Computation time
time.svl <- round(as.numeric(now-then), 2)
```

Figure 2 shows the predicted languages for each of the 600 test observations using the SVM with a linear kernel (jittered for visual clarity). Its misclassification rate of 11.67% is worse than that of the logistic regression model, and we see more red points, which again represent misclassified sentences.

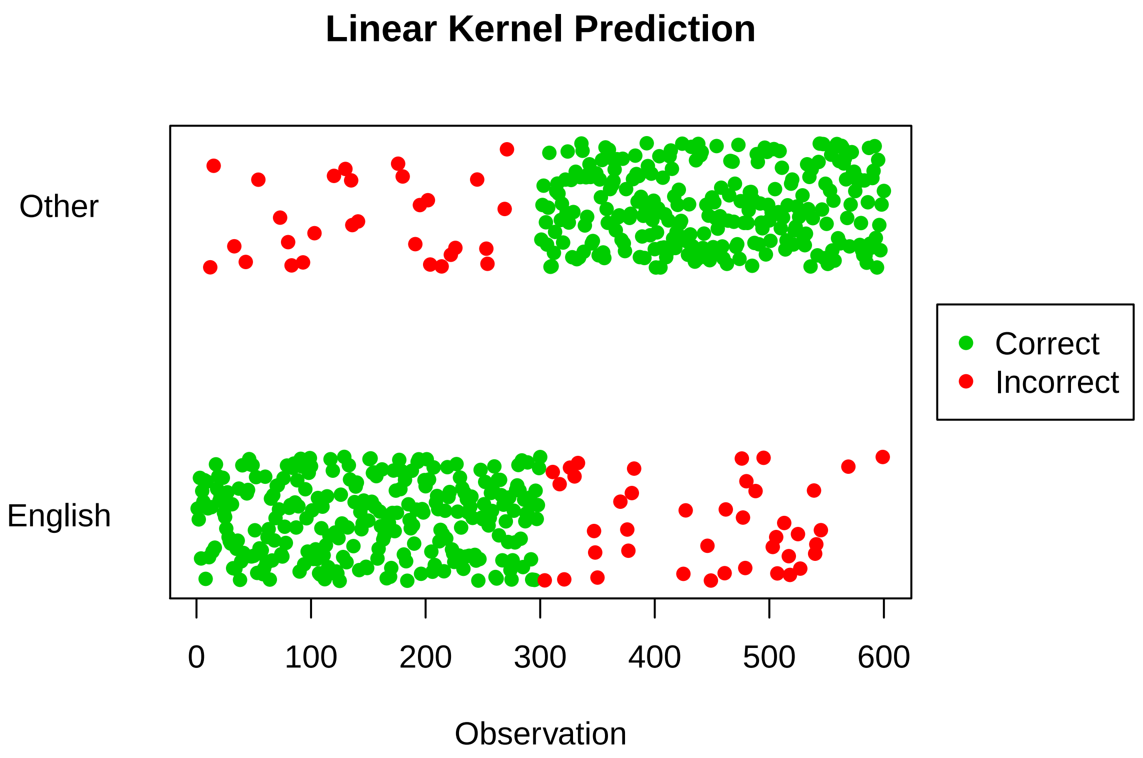


Figure 2: Predicted Outcomes from SVM Linear Kernel

4.2 Gaussian Kernel

We next consider an SVM with a Gaussian kernel:

$$k(x, y) = \exp\left(\frac{\|x - y\|_2^2}{2\sigma^2}\right)$$

This is a very popular kernel in machine learning due to its flexibility in approximating a large variety of decision boundaries. However, it is known to be susceptible to overfitting, particularly when the training data set is small[2].

Indeed, Figure 3 above shows that the Gaussian kernel predicts a sentence's language more accurately than the linear kernel, however this SVM's misclassification rate is 9.17%, which is still worse than that of the logistic regression model.

[2]Negahban, Sahand. Data Mining and Machine Learning: Lecture 15.

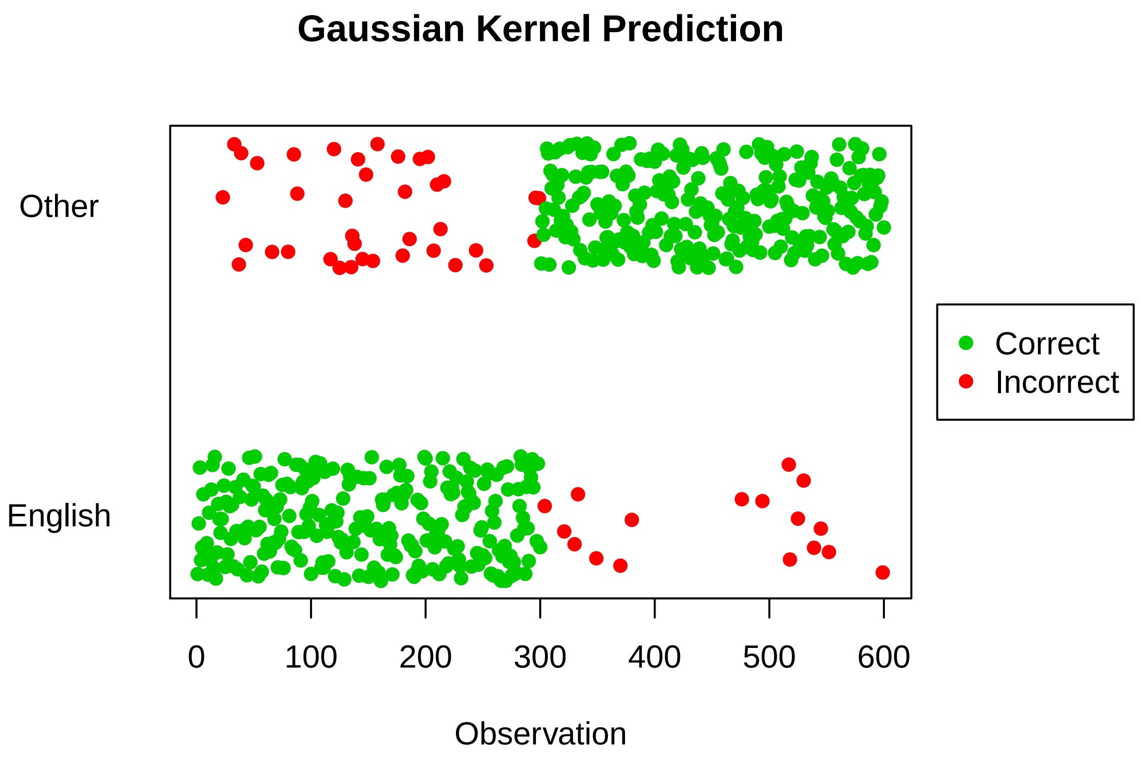


Figure 3: Predicted Outcomes from SVM Gaussian Kernel

4.3 Kullback-Leibler Kernel

The feature vectors in our model are in the form of relative frequencies. We can take advantage of this structure by treating them as empirical probability distributions. It is plausible that languages have an underlying distribution on the frequency of letters, and that from this distribution various sentences have been randomly created.

To determine from which underlying distribution a given sentence comes, we can therefore use techniques from statistical theory. We use a kernel based on the Kullback-Leibler Divergence between two distributions:

$$D(p\|q) = \sum_{i=1}^{n} p_i \log\left(\frac{p_i}{q_i}\right)$$

In order to use this as a kernel in a support vector machine, we must ensure it is symmetric and positive semidefinite. Moreno *et al.* suggest the following transformation as a kernel:

$$k(x, y) = \exp(-a(D(x\|y) + D(y\|x)))$$

where a is a parameter to be chosen.[3]

Using the `kernlab` package, we implement this SVM on our language classification data set. It runs far slower than any of the alternative algorithms, which we suspect is, to some extent, because the package is not optimized for our kernel.

Following experimentation with different values of the parameter a, we found that a value of 0.001 minimizes the misclassification rate on the test set, which still only did somewhat better than guessing with a misclassification rate of 40.33%.

Because this is a custom fuction, we present the code below for clarity:

[3]Moreno, P.J., *et al.*, p. 3

```r
# Add very small value to all values so 0 is not an issue
xnew <- x; ynew <- y
xnew[,-1] <- xnew[,-1] + .0000000000000000001
ynew[,-1] <- ynew[,-1] + .0000000000000000001

KL <- function(p, q) {
  vec <- p*log(p/q)
  return(sum(vec, na.rm=T))
}

mykernel <- function(p, q) {
  exp(0.001*(KL(p, q) + KL(q, p)))
}

class(mykernel) <- "kernel"
```

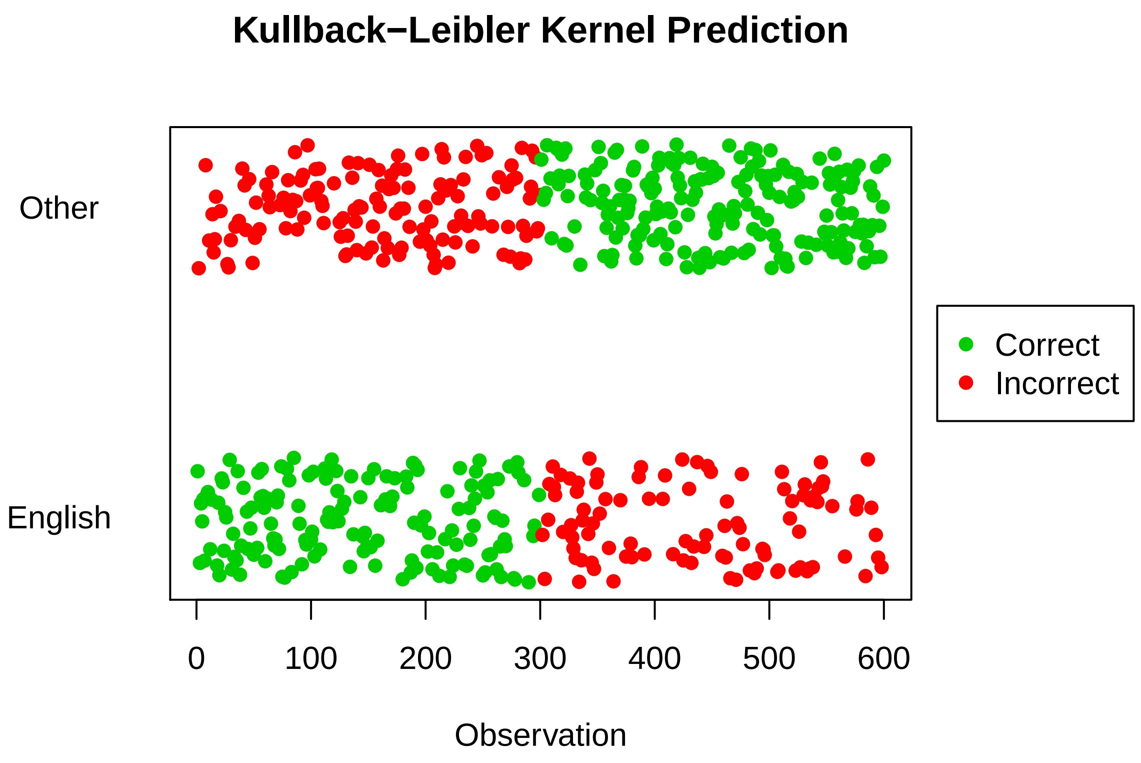


Figure 4: Predicted Outcomes from SVM Kullback-Leibler Kernel

5 Discussion of Models and Choice of a "Best"

We present in the table below a summary of each model's misclassification rate (%) on the test set, as well as the amount of time in seconds elapsed during each model's training.

Model	Misclassification	Time
Logistic	8.83	0.10
Linear SVM	11.67	0.24
Gaussian SVM	9.17	0.23
Kullback-Leibler SVM	40.33	3.87

The logistic regression model performs better than any of the SVM algorithms in both misclassification rate and computational efficiency. The SVM with Kullback-Leibler kernel is by far the worst predictor of sentence language. Although our consideration of choice of the a-parameter was thorough, perhaps further testing is required.

6 Two-letter Combinations

We saw that our logistic regression model performed well when trained on the empirical distributions of single letter frequencies, and we hypothesized that we could reduce the misclassification rate even further by considering the empirical distributions of pairs of adjacent characters as they appeared in our original observations. We suspected that greater specificity in our data set with $27^2 = 729$ columns would improve our best model's predictive accuracy because certain letter combinations are unique to specific languages. For example, "pf" appears often in German (such as in "kopf", which means "head"), but rarely in English. Similarly, "wh" appears often in English, but rarely in German.

We thus trained our two previously-identified top performing models, logistic regression and SVM with Gaussian kernel, on such a data set.

When we compare the frequency of red points in plots of Figure 5, we observe that when trained on pairs of letters, the SVM with Gaussian kernel is a clear winner. As confirmed in the summary table of results below, the SVM with Gaussian kernel takes less time to be trained than the logistic regression model, and the use of pairs of letters reduces its misclassification rate to 5.83%.

Model	Misclassification	Time
Logistic	11.50	22.2
Gaussian SVM	5.83	0.9

We know from class[4] that SVMs with a Gaussian kernel perform best when there is a great deal of information available, and we suspect that the increase of information available to the model enabled it to surpass the predictive ability of the logistic regression model.

[4]Negahban, Sahand. Data Mining and Machine Learning: Lecture 15.

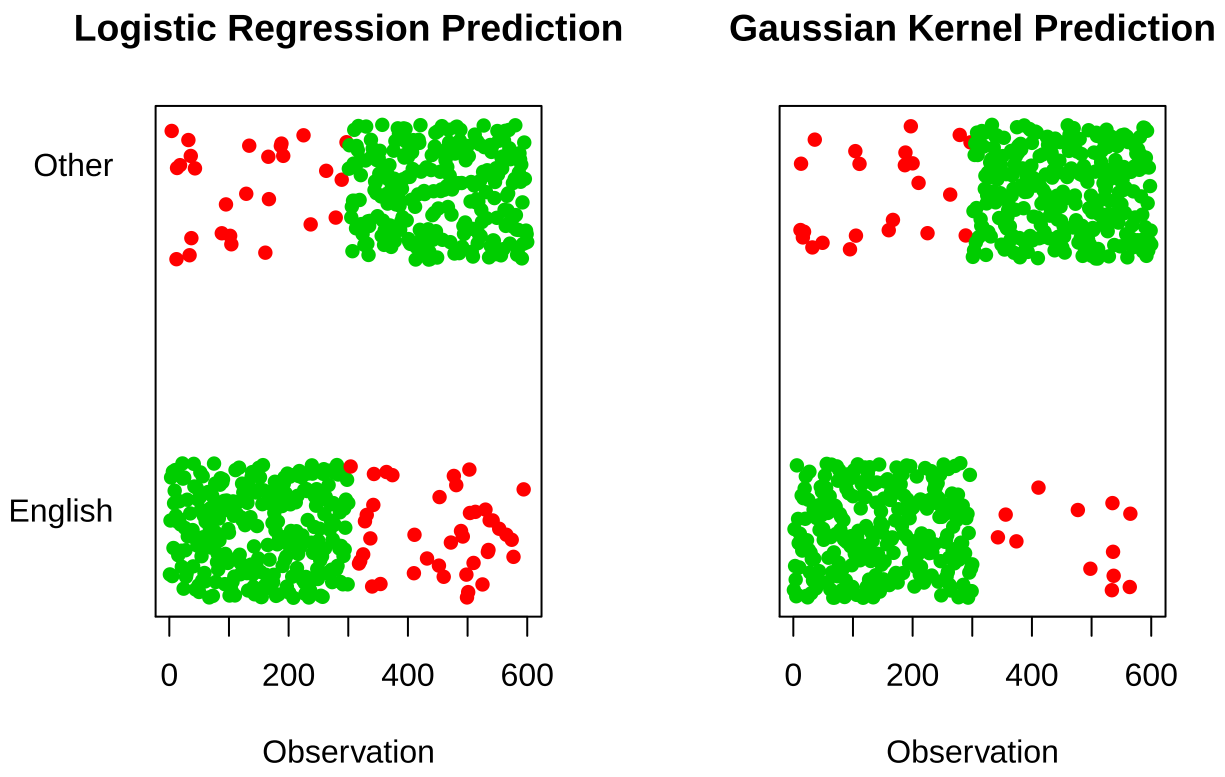


Figure 5: Predicted Outcomes from Logistic Regression and Gaussian Kernel (Double Letter Combination)

7 Application of Model: Twitter versus *The New York Times*

We have demonstrated that a SVM with Gaussian kernel has great success at differentiating sentences written in the English language from sentences written in other languages when trained with a good amount of data (in the form of relative frequencies for each of the possible 729 letter pairs per observation). We are also interested in the extension of this finding to different uses of the English language. Formal writing differs dramatically in style, punctuation, sentence structure, and vocabulary from more casual interactions, such as those that take place through social media platforms and make frequent use of abbreviations and slang. All editors and writers for *The New York Times* are expected to uphold the regulations set forth in *The New York Times Manual of Style and Usage*.

We are curious as to whether or not a SVM with Gaussian kernel is still effective when challenged to distinguish between formal and colloquial uses of the English language. Specifically, we try to correctly classify observations from social media messages on Twitter and sentences from *New York Times* articles. Figure 6[5] is an example of a sentence that appears in a *New York Times* article and Figure 7[6] is a public tweet:

7.1 Collection of *New York Times* and Twitter Data

We trained a logistic regression model and our “best” model, the SVM with Gaussian kernel, using relative frequencies of pairs of letters that appear in sentences from each of *The New York Times* and public postings from Twitter. Using the Twitter API, we extracted 1000 of the most recent tweets from the site (as of 8:00 pm on Tuesday, April 28, 2015). Similarly, we randomly chose 1000 sentences extracted from the top fifteen “Most Viewed” articles on www.nytimes.com (as of Friday, May 1st at 3:30 pm) using the *Times* API.

[5]Blinder, A., *et al.*
[6]cotiti

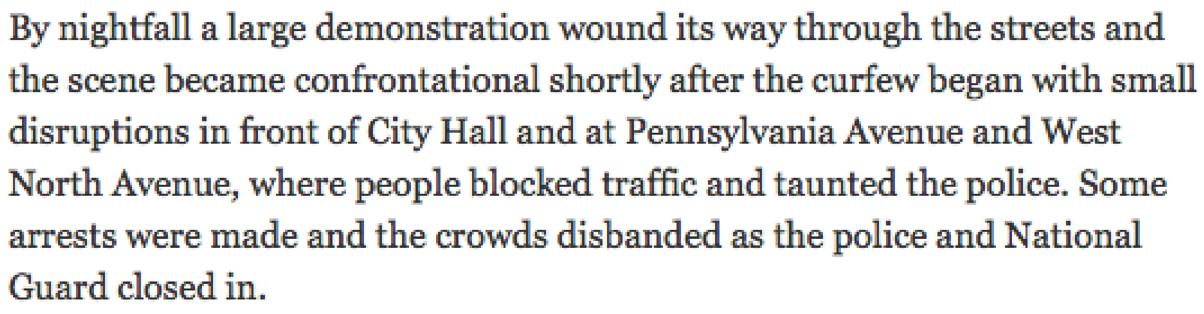
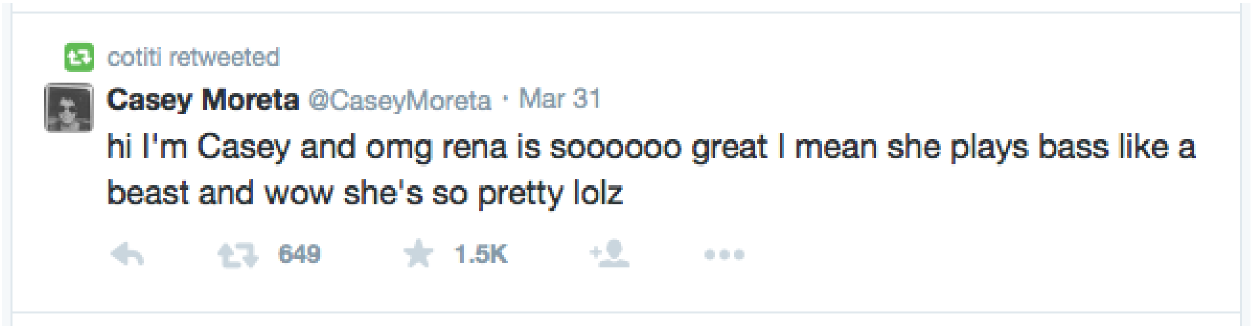

By nightfall a large demonstration wound its way through the streets and the scene became confrontational shortly after the curfew began with small disruptions in front of City Hall and at Pennsylvania Avenue and West North Avenue, where people blocked traffic and taunted the police. Some arrests were made and the crowds disbanded as the police and National Guard closed in.

Figure 6: Sentence from *NYT* article: “6 Baltimore Police Officers Charged in Freddie Gray Death”


cotiti retweeted

Casey Moreta @CaseyMoreta · Mar 31

hi I'm Casey and omg rena is soooooo great I mean she plays bass like a beast and wow she's so pretty lolz

649 1.5K

Figure 7: Retweet by user “cotiti”

Each tweet from Twitter and each sentence from *The New York Times* was cleaned using the same methodology described in Section 2.2, and the empirical distribution of frequencies of pairs of letters were created. Our training data set consists of 700 tweets and 700 *New York Times* sentences.

Figure 8 compares the predicted sentence sources from the two models, logistic regression and SVM with Gaussian kernel. Note that the first 300 observations of the test set are tweets, while observations 301-600 are sentences from the *New York Times*. The fewer number of red points suggests that the SVM with Gaussian kernel misclassifies the sources of sentences less frequently than does the logistic regression model. Furthermore, it takes less time to train than the logistic regression model when presented with an abundance of data.

Model	Misclassification	Time
Logistic	23.33	16.14
Gaussian SVM	11.33	1.97

7.2 Model Limitations Specific to Classification of *New York Times* versus Twitter

We recognize that our data cleaning function presents obvious limitations that may specifically influence our model’s ability to classify sentences as being either from the *New York Times* or Twitter. As previously discussed in Section 2.2, the function removes all capitalization of letters, as well as unique symbols. Tweets make frequent use of the hashtag (#) and at (@) symbols, and this syntax is immediately lost. Inclusion of such characters would likely increase the predictive ability of the SVM with Gaussian kernel. Similarly, we know that punctuation differs between sentences appearing in *The New York Times* and tweets. While tweets sometimes provide emphasis with multiple exclamation points, this is a pattern that would occur rarely (if ever) in a formal newspaper.

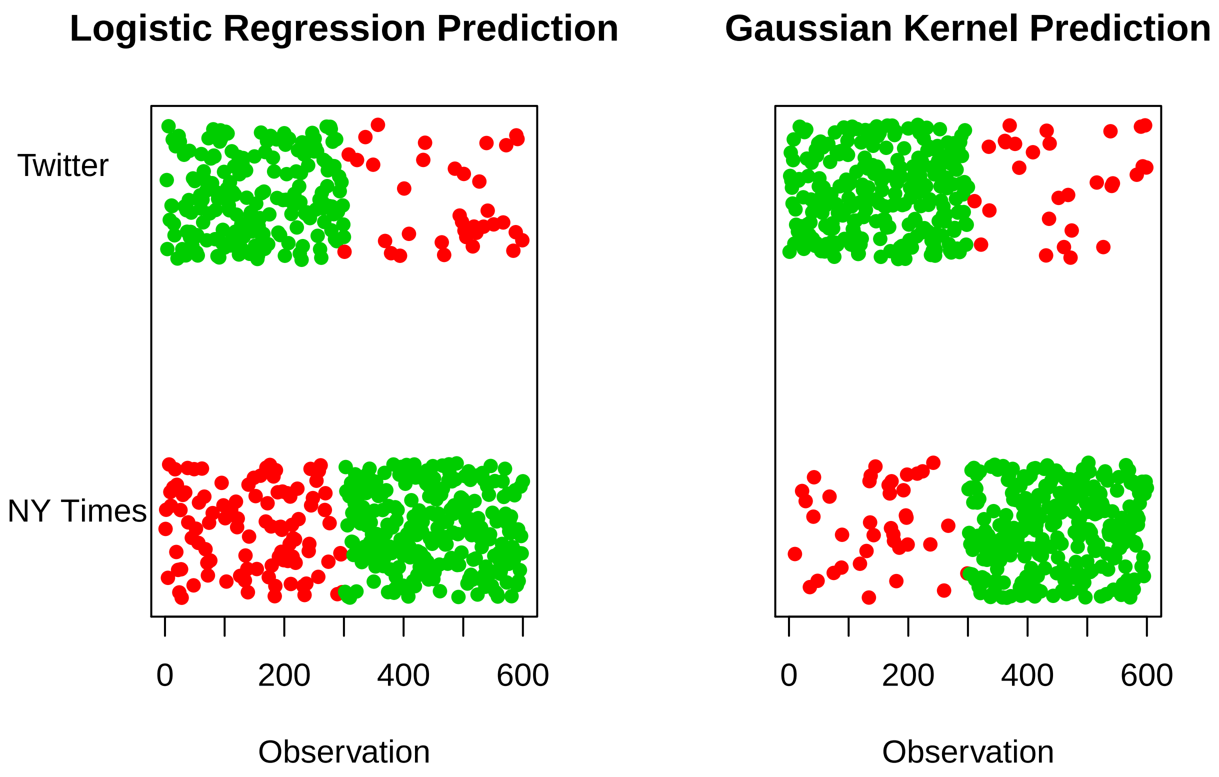


Figure 8: Predicted Outcomes from Logistic Regression and Gaussian Kernel (NY Times Application)

Furthermore, because our function split sentences from *New York Times* articles along periods, there were observations that consisted only of "mr" or "ms" that occurred when a person was referred to in an article as, for example, "Mr. [last name]".

8 Conclusion

8.1 Findings

Our analysis revealed that the Gaussian Kernel SVM with two-letter pairs was the best classifier of languages among the algorithms considered. The model was able to achieve a small predictive error, despite being given a limited amount of training data. Furthermore, no prior understanding of the languages involved was required to implement the algorithm. This flexibility enabled us to use the same algorithm to detect differences within a single language, as shown by the *The New York Times* and Twitter application.

The other models were able to classify languages with reasonable accuracy, with the notable exception of the Kullback-Leibler Kernel SVM. When considering single letter frequencies, the logistic regression slightly outperformed the Gaussian SVM. However, when the additional complexity of two-letter pairs was added to the model, Gaussian SVM surpassed logistic regression. This is consistent with the known limitations of logistic regression on sparse matrices and the necessity of large data sets for Gaussian SVM.

8.2 General Model Limitations

We recognize that our sole use of the Bible for classification algorithm training and testing is a limitation of our research. Further research may consider text from a range of sources, in order to ensure full generalizability.

Furthermore, our data collection and cleaning methodology from this source resulted in sentences that were essentially nonsense; either they consisted of too little information (a single word) or words that were not specifically English. This has led to a conservative estimate of misclassification rate, which may in truth be even lower than stated in this paper.

8.3 Possible Extensions of Our Work

Future research may include multinomial language classification so that a specific language can be returned as opposed to a binary distinction between English and non-English. Additionally, we could increase the complexity further to consider not just pairs of letters, but strings of length three or more. This may enable even subtler distinctions to be made between different sources, such as author or time period of a text.

9 References

Blinder, A., & Perez-Pena, R. (2015, May 1). 6 Baltimore Police Officers Charged in Freddie Gray Death. *The New York Times.* Retrieved May 4, 2015, from http://www.nytimes.com/2015/05/02/us/freddie-gray-autopsy-report-given-to-baltimore-prosecutors.html?_r=0

cotiti. [CotiCN21]. (2015, March 31). RT [CaseyMoreta] hi I'm Casey and omg rena is soooooo great I mean she plays bass like a beast and wow she's so pretty lolz [Tweet]. Retrieved from https://twitter.com/CaseyMoreta/status/582764574388068354

The Holy Bible: Dutch Statenvertaling. (n.d.). Christian Classics Ethereal Library. Retrieved April 24, 2015, from http://www.ccel.org/ccel/bible/nls.txt

The Holy Bible: French Louis Segond Translation. (n.d.). Christian Classics Ethereal Library. Retrieved April 24, 2015, from http://www.ccel.org/ccel/bible/frls.txt

The Holy Bible: German Luther Translation. (n.d.). Christian Classics Ethereal Library. Retrieved April 24, 2015, from http://www.ccel.org/ccel/bible/delut.txt

The Holy Bible: Italian Translation. (n.d.). Christian Classics Ethereal Library. Retrieved April 24, 2015, from http://www.ccel.org/ccel/bible/it.txt

The Holy Bible: Latin Vulgate Translation. (n.d.). Christian Classics Ethereal Library. Retrieved April 24, 2015, from http://www.ccel.org/ccel/bible/vul.txt

The Holy Bible: Spanish Reina Valera. (n.d.). Christian Classics Ethereal Library. Retrieved April 24, 2015, from http://www.ccel.org/ccel/bible/esrv.txt

The Holy Bible: Webster's Bible. (n.d.). Christian Classics Ethereal Library. Retrieved April 24, 2015, from http://www.ccel.org/ccel/bible/webster.txt

Moreno, P.J., Purdy. H.P., & Vasconcelos, N. (2003). A Kullback-Leibler Divergence Based Kernel for SVM Classification in Multimedia Applications. In *Advances in Neural Information Processing Systems 16.* S. Thrun, L.K. Saul, and B. Schölkopf, Eds. Retrieved April 29, 2015, from http://machinelearning.wustl.edu/mlpapers/paper_files/NIPS2003_SP02.pdf

Negahban, S. (2015, March 4). Data Mining and Machine Learning: Lecture 15 [.pdf Notes]. Retrieved April 29, 2015, from https://classesv2.yale.edu/access/content/group/stat365_s15/Lectures/lec15.pdf

10 Appendices

10.1 Appendix A: Logistic Regression Model's Misclassified English Bible sentences

Misclassified.Lines
die.
thou comest to Zoar.
violence.
mortar.
Damascus.
me?
portion.
eighteen, and pursued them to Dan.
Joktan.
righteousness.
ˆ2The sons of Japheth; Gomer, and Magog, and Madai, and Javan, and
ˆ13And Mizraim begat Ludim, and Anamim, and Lehabim, and Naphtuhim.
Hazezontamar.
covered.
ˆ3And they said one to another, come, let us make brick, and burn them
quickly three measures of fine meal, knead it, and make cakes upon the
ˆ26And Joktan begat Almodad, and Sheleph, and Hazarmaveth, and Jerah,
begat Methusael: and Methusael begat Lamech.