Machine Learning and Feature Based Approaches to Gender Classification of Facebook Statuses

Jeremy Keeshin, jkeeshin@stanford.edu
Zach Galant zgalant@stanford.edu
David Kravitz kravitzd@stanford.edu

June 2, 2010

1 Abstract

The goal of this project was to predict the genders of Facebook users based on individual status updates. We did this by using several different machine learning techniques and training on a corpus of over 170,000 different status updates from males and females. This research is related to authorship classification and gender classification which has been done many times before, but it seems that this is some of the initial research into classification of Facebook status updates, which presents several fascinating and compelling new challenges. We found that our classifiers performed just as well as humans on this task, which was in the range of 60-65% correct. We also created a web interface for the project at `http://www.thekeesh.com/cs224n/`.

2 Introduction

At first it seems like this should be a relatively straightforward problem in binary classification, but after further research into the subject, there are many difficulties. Many statuses are simply very short and may not contain all of the stylisticly rich information that previous email and chat classification studies have had. In addition, many of the statuses are ambiguous even to humans, and this will be explored in greater depth later in the paper.

We first collected data using the Facebook Open Graph API and stored text files with $gender - status$ pairs. Then we used this data in several different methods of classification. We used a modified version of the Maximum Entropy Classifier from earlier in the year, which was mainly concerned with finding useful features of the data. We then looked at several linear classifier algorithms including the perceptron and closely related Winnow algorithm. We also created a simple Naive Bayes classifier, and a bigram status generative model for both males and females. Early on in the project, we attempted to create a pair of language models, one for male and one for female, and then score the statuses on each model to see which returned the higher score. Unfortunately, this method was unsuccessful.

Another important part of the project was to conduct a survey to create a baseline. We took random statuses from our data and created a survey where we had participants try to predict the gender of the user. The task appeared to be extremely difficult as the average score was only 65.5%.

3 Background Research

When we explored the research that has been done previously in this area, we found lots of helpful hints about which direction we should take our project. We first read research on human prediction of gender in electronic communication. It has been found that readers can predict gender with a high accuracy of up to 91.4% in email message exchanges [4]. At a high level it was found that females use higher frequencies of emotion words, questions, compliments, and apologies than men. It was found that men use more opinions, adjectives and insults [4]. These features map over in an interesting way to our research. While we were not able to extract these high level features, these sorts of insult words are definitely more prevalent in males, and such emotion messages like emoticons were more prevalent in females.

In another email authorship classification study, Corney et al. use machine learning algorithms to classify up to 70% correct with a combination of character attributes, word based attributes, structural attributes and function words.[1] An interesting feature they use is also common word suffixes. However many of the features have a very small impact.

We looked at papers from last year's CS224N class where there was a student who did gender classification from *Second Life*. Ricciardi uses features that take into account emoticons, slang, typos, and specific high frequency words. [3] He consistently gets accuracies in the mid 80s.

In our research, although it is related, there are many important differences that explain our lower results. Humans have been able to correctly predict upwards of 90% of email messages, but in our survey results for Facebook status classification, users can only get 65%. The main important differences are that chats and emails are generally much longer and have certain meta-characteristics that are not present in plaintext Facebook statuses. For example, although an individual instant message may be short, an entire chat consists of a series of back and forth messages. Likewise emails are much longer, and can have distinct signatures that may give away the gender. Although these attributes are present in some form in Facebook, they are much less pronounced.

4 Dataset

We used the Facebook Open Graph API to collect public statuses and gender information. We kept one large data file which we later parsed into a female and male file. We were able to gather over 170,000 statuses from males and females. We also collected smaller validation sets. There were some interesting peculiarities about the dataset. First, in every set we collected, there was a preponderance of female data. When we got around 200,000 statuses, there were 124,458 female and 70,631 male. What this means is that for any arbitrary status it was almost twice as likely to be female than male. When we trained on unequal amounts of data, we got results that were extremely skewed towards guessing female with up to 96% specificity and 7% sensitivity (where male was the positive class and female was the negative class). This achieved comparable overall accuracies of 61%, but was clearly a worse classifier than one that was able to handle male and female statuses.

There are a few possibilities here about the bias of the data. One possibility is simply that females write many more statuses than males, and the number of statuses in our training set is just a reflection on that. Another possibility is that we had an error in collecting data, but we were just using public search results, so this is unlikely. Additionally, this effect was removed because in our final tests we trained on equal amounts of male and female data.

5 Analysis of Human Classification

We sent out a survey with these status updates and asked people to guess what the gender was of the message. The answers here are posted below.

1. nothing more irritataing than seeing your bro wave to you on the bus >.< and later have a non stop lecture on how i should listen to school rules :S (Female)

2. Hahahaha what a joke!! (Female)

3. 4 guys on a bed= no homo (Male)

4. Playin are you smarter than a 5th grader with shawn aNd hannah :)..... I think imma lose :/ lol. (Female)

5. One was surpose to be a stone lighter 4 pride, opps stone fatter instead. Lol (Female)

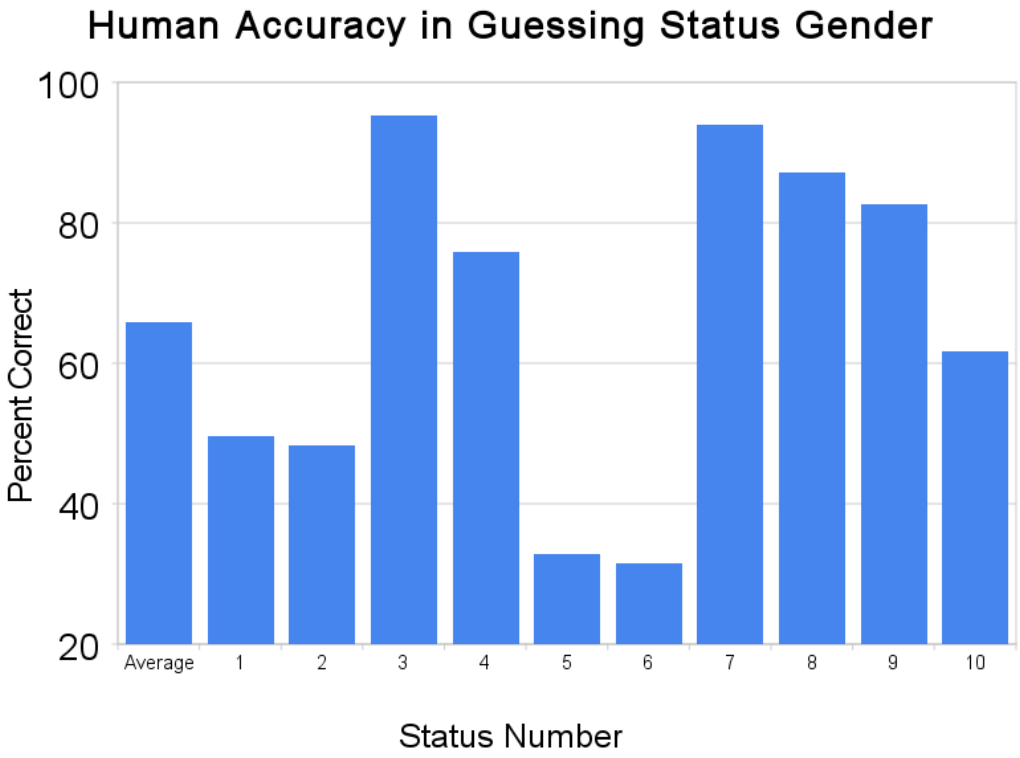
6. Sometimes I close my eyes and imagine i'm a ninja, other times I just close my eyes and then I wake up! (Female)

7. Chuck Norris does not wear a condom. Cuz there is no such thing as protection from Chuck Norris. (Male)

8. I must be tiered, for a second, I thought the MSN on the hotmail page stood for Master of Science in Nursing LOL!!! Been up writing nursing papers, and revising group projects too long! (Female)

9. cant sleep again and this is starting to be a pisser (Male)

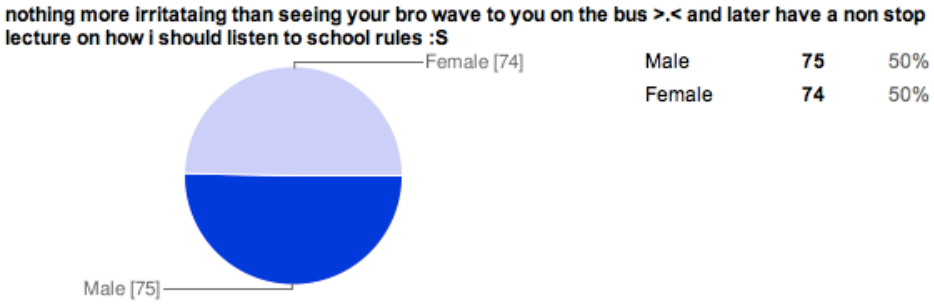
10. I just got pulled over by a cop while I was riding my bicycle haha (Male)



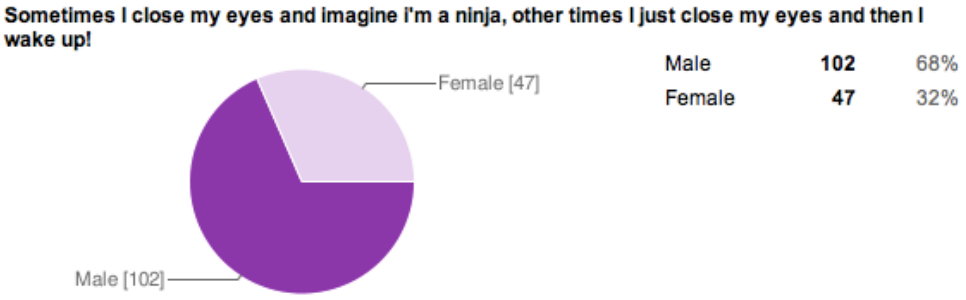
What we found was that people were not very good at predicting the status. Some of them are "easy" examples, and people get over 90% correct. Several of them, however, are extremely ambiguous, and others are even skewed strongly in for the wrong gender. Scores on individual questions ranged from as low as 31.54% to as high as 95.3%.

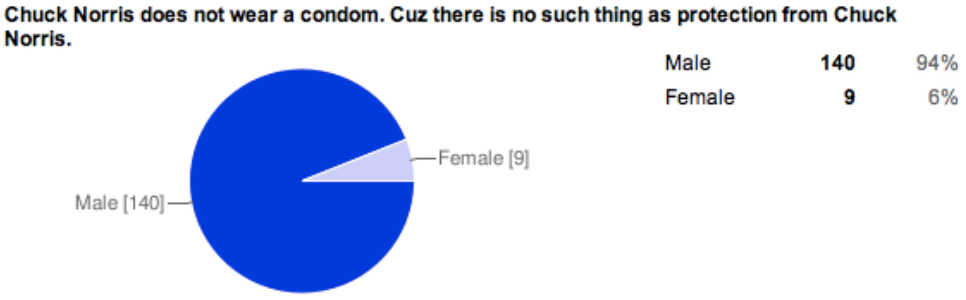
We had 149 people take this survey, and the graph shows the results. The average score was 65.5% and some questions are clearly very difficult even for humans because they simply don't contain enough information, or the information they do contain indicates the opposite gender.

Some questions were split 50-50 exactly.



On some questions people were very wrong in their guesses, guessing predominantly male while it was actually a female.





But everyone knows that only a guy would write that "Chuck Norris does not wear a condom."

What is important here is that some statuses are too ambiguous for humans to figure out, and that this is a tough problem. There are strange features that make some examples easy, and this is often the topic, like Chuck Norris, or an overall attitude as was mentioned in earlier research. In these examples the punctuation and emoticons were correct in determining female, but a problem is that although these patterns hold in general, there are many counterexamples in a dataset this large.

6 Maximum Entropy Classifier

6.1 Overview

We took the MaxEnt Classifier we made for Programming Assignment 3 and modified it to classify full statuses rather than just words. It gets trained on a dataset of over 120,000 Facebook statuses labeled either Male or Female and runs for 60 iterations. We extracted features from the statuses to train the classifier. Every feature was a feature related to the entire status or parts of the status.

6.2 Features

We designed word based, structure based, and count based features. We looked through our training data to see common features that we could utilize for feature creation.

6.2.1 Word Based

For every status, we created features of the form "HAS_WORD_"+word for each word (separated by spaces) in the status. This was a very helpful feature, since it helped determine which common words are used by either gender. This is a similar approach to a unigram bag of words model. We have multiple lists of stop words that we used. The best performance was achieved using a short list of stop words, having a slight improvement over a more comprehensive list and not using stop words at all.

On Facebook, people often make typos, especially on purpose typos, which often involve the last letter of a word being repeated multiple times in a row at the end. We implemented stemming to group all of these words together. For example, "happyyyyy," "happy," and "happyy" would all be stemmed to be the same word. We activated a feature HAS_MULTIPLE_REPEATED_LETTERS when we stemmed these words. This was helpful, since females use the repeated letters attribute much more than males. We also did stemming for plurals, so any word ending in "s," except for words ending in "ss" would remove the last letter to stem it to its lexical root.

We also added more specific features that took into account certain very common words or text strings. We had features for profanity, abbreviations like "omg" and "lol," common potentially more gender specific words like "love," and specific phrases like "haha" and "xo." Profanity, for example was almost twice as common in males than females. It was in over 7% of male statuses and in only 4% of female statuses.

We also looked for text signifying links and made features for having a link and more specific links like youtube.com and links that use url shortening services like bit.ly and tinyurl.com, which occur more in the male data. Given a male status, it was twice as likely to have a youtube link than a female status (.8% to .3%). This is a small percentage, but still quite telling if the status does have a youtube link.

These features were very helpful for classifying male posts. With these features removed, we got a .01 F1 Score for Male, though we got a .74 F1 Score for Female posts. This classifier basically guessed Female for almost everything.

6.2.2 Structure Based

We created many features around structure, including use of punctuation, use of capital letters, and use of emoticons. We have a feature called HAS_SMILEY that gets activated if the user uses any of the text-based emoticons in their post. This is telling because 12% of the female training statuses had smileys, but only 7% of the male statuses had smileys. That makes $P(F|HAS\_SMILEY) = .73$ while$P(M|HAS_SMILEY) = .27$. Also, one of the more helpful ones was the HAS_HEART feature for statuses that have "<3" in them.

We also have features regarding the use of excessive punctuation and excessive capital letters. Having a number of exclamation points or question marks in a row activate another feature. We also activated features if the entire post was in capitals, or if a large portion of it was capital letters.

6.2.3 Count Based

We created features based on counts of certain attributes. We calculated the percentage of the status that was letters, digits, punctuation, whitespace, upper case, and lower case. We then bucketed them into 10 buckets based on the percentage: 0-9% in one bucket, 10-19% in the next, and so on. Depending on the bucket, we activated a different feature.

These were not as helpful as expected, as they seemed to overload the information. We ended up only using the punctuation and case features, which improved classification, while the rest did not. The others slightly lowered the accuracy because they just made the feature list overly complicated, and it turned out that knowing the percentage of characters that are letters just doesnt help that much, as it doesnt distinguish between the genders much.

6.2.4 Overall

Word based features as well as some of the structure-based features were the most helpful. Certain words like profanity were particularly helpful, and word choice tends to separate the genders quite a bit. Website links were very informative as well, as males were more likely to share links, especially youtube links.

6.3 Performance

The MaxEnt classifier managed to classify with as high as 62.2% accuracy. It did significantly better classifying Female statuses than classifying Male statuses. We used F1 Score to evaluate the individual genders classification performance. Our Female F1 was .71, but our Male F1 was .43. Female recall was quite high, but its precision was less accurate.

Without using stop words, our accuracy stayed very close at around 62.1%, but the fluctuation was in the F1 Score. Male score was .38 and Female score was .77. The most change in modifying certain features was the difference in performance between the genders, while the overall accuracy stayed fairly similar. Depending on which features we activated, we got overall accuracy in the range of 59% to 62.2%.

$FeatureRemoved$	$Accuracy$	$MaleF1$	$FemaleF1$
Baseline (Nothing Removed)	62.2	0.43	0.71
Stop Words	62.1	0.38	0.77
Stemming	61.3	0.47	0.69
Word Based Features	59.8	0.01	0.74
Structure Based Features	61	0.42	0.7
Count Features	61.8	0.45	0.7

The table above shows that the baseline not only has the best accuracy, but also some of the more well rounded F1 Scores.

We had different amounts of training data as well. We started off using around 35,000 statuses, but when we increased to 120,000 statuses, we saw about a 1% jump in overall accuracy.

We had uneven amounts of data for male and female statuses, but this did not make a difference at test time. We got the same scores whether we used equal amounts of data for each gender or uneven amounts.

6.4 Errors and Hard Classifications

6.4.1 Female Looking Statuses

Our classifier had a tendency to classify more statuses as female than male by a significant margin. For this reason, our female recall was high, but its precision was lower. Also, the male recall was much lower. We can attribute this to most features being positive features for female writers. Many of the emoticons, capitalization patterns, and letter repetition sequences were more specific to female writers, but males still sometimes had these features. That is, male writers write like females on Facebook, making it more difficult to classify them as male.

Examples: "NeEd a VaCATIOn" was written by a male, but has female capitalization attributes, so it was incorrectly classified. Similarly, "0h what a day this is g0na be..:-) yeah! <3"was also written by a male, though it looks like a female wrote it. It even had "<3" in it, which is predominantly used by female authors. One more example that looks very female, but was written by a male: "OH CRAP gotta go to work... GLEE night yayyyyyyyyyyyyyyy, get to see Tina @ work YAYYYYYYYYYYYYY should be a good night." It uses many capitals and repeated letters at the end of words, so it incorrectly thinks it was female.

A possible explanation is that males have different posting patterns when their posts are directed towards males versus females. For example, males may pick up female writing characteristics when directing posts to females. Females, on the other hand, are less likely to change their speech patterns based on which gender at which their post is directed. One more possibility is that friends often log into their friends profile and post things that do not sound like the person as a joke. There is no way to distinguish actual posts from joke posts by friends.

6.4.2 Short Statuses

Many statuses are just a few words, making it much more difficult to classify, since so few of the features apply. This makes their classification more of a random guess than longer ones. For example, "What a boring night!" was incorrectly classified as female by our classifier, but it would be hard even for a human to guess the gender of the author.

6.4.3 Possible Improvements

One improvement would be to implement a much smarter lexical stemming system. If we could accurately determine what word the writer meant, disregarding typos, purposeful misspellings, and capitalization rules, we could much more accurately pick up patterns. This is very hard because of the relaxed, informal nature of Facebook posts, where people dont use spell check.

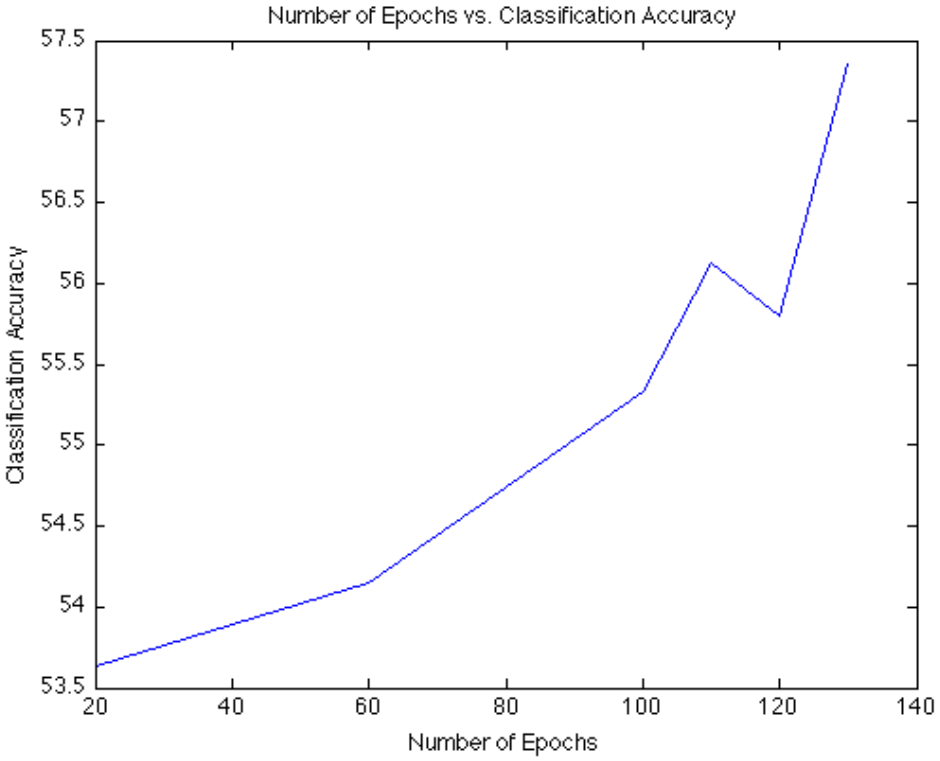
6.5 Conclusion

Classifying gender based on only a short snippet of text proves to be a difficult task, as we only managed to get 62% correct, which is just a bit better than a random guess, which would get around 50%. Using features seems to be a good strategy, but it proves to be much more difficult, because attributes change based on audience, and there are a lot of anomalies.

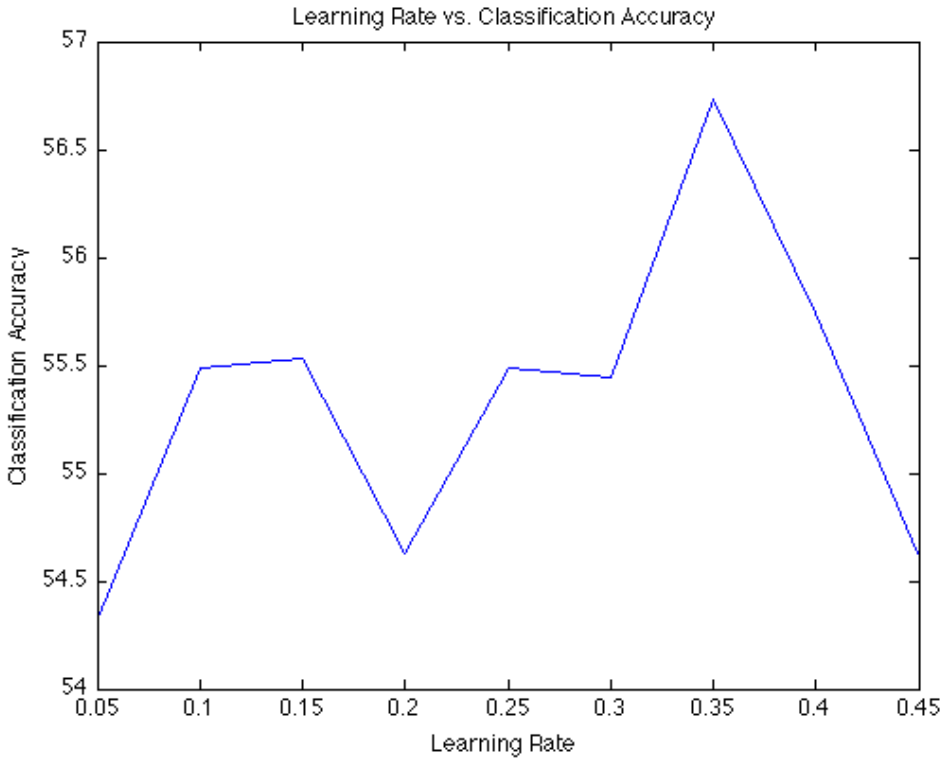
7 Perceptron Algorithm

We also implemented the perceptron algorithm for text classification. The algorithm is an iterative, data-driven, gradient descent based technique for learning a decision hyperplane. Since the algorithm is iterative, we were able to input a very large amount of data. The algorithm also gives a reasonable bound on the number of errors. See Ventura's paper for further details.[5]

After coding up the system, we devoted our efforts to improving our baseline model, which had a 53.2% classification accuracy. First, we optimized the total number of epochs.



We saw no improvement after we ran the algorithm for 130 epochs. Next, we optimized the learning rate.



The optimal learning rate was 0.35.

With an optimized number of epochs and learning rate, we began to look through the output weights for feature information. We wanted to see exactly what words were key indicators of the status gender. The idea was to then use these features in our maximum entropy model. The results were fascinating. In the perceptron algorithm, a positive weight means that the word tended to be associated with male statuses, while a negative weight means that the word tended to be associated with female statuses. The closer a weight is to 0, the less meaningful that word is in determining the gender of a status.

Examples of important female words and their weights

Feature	Weight
♡	-18.20
< 3	-52.85
fun	-9.45
we	-10.15

Examples of important male words and their weights

Feature	Weight
take	4.54
man	5.24
bad	1.75
ever	17.14

There were a number of words, like "<3", that appeared often in female statuses, but very rarely in male statuses. Male statuses tended to use words that were also used in female statuses. In other words, male statuses simply did not contain many words unique to the gender. You can see above, that the weights of even the most important male words were significantly smaller than the weights of the most important female words. As a result, the algorithm was more accurate when classifying female statuses. We actually increased performance slightly by artificially boosting (or reducing) the weights of important words.

After all optimizations, the perceptron algorithm (with learning rate = 0.35 and epochs = 130) correctly classifies 56.6% of the total sample, 40.0% of the male cases, and 67.7% of the female cases.

7.1 Winnow Algorithm

The winnow algorithm algorithm is closely related to the perceptron algorithm, but uses a different update rule. Like the perceptron algorithm, it attempts to construct a linear separator for each class. In fact, Winnow is guaranteed to find a linear separator if it exists. Winnow converges more quickly than the perceptron algorithm and is also quite effective at removing extraneous features. This means it scales well to incredibly high-dimensional spaces. See Littlestone's paper for more details.[2]

After all optimizations, the Winnow algorithm (with alpha = 2 and epochs = 130) correctly classifies 59.1% of the total sample, 19.6% of the male cases, and 85.5% of the female cases. This was a slight improvement in overall performance compared to the perceptron algorithm. However, there was a huge decrease in classification accuracy in the male cases, and a huge increase in classification accuracy in the female cases. But remember that when there is a mistake in training, the Winnow algorithm increases and decreases its weights multiplicatively. In other words, feature importance is magnified and feature unimportance is diminished compared to the perceptron algorithm. This idea, coupled with the fact that there are a large number of unique features that appear in female statuses, but not in male statuses, explains the high female classification accuracy and the low male classification accuracy.

8 Naive Bayes Classifier

We created a simple Naive Bayes classifier with add-one smoothing which performed surprisingly well. We achieved a highest score of 67.7%. We use a HashMap which stores a word count for each class and then using Bayes rule, the classification is done by taking the gender that maximizes the score on the status:

$$P(G|S) = P(G|w_1, w_2..w_n)$$
$$= \frac{P(w_1, w_2, ..w_n|G)P(G)}{P(w_1, w_2..w_n)}$$

But since the denominator is the same for both examples.

$$= P(w_1, w_2, ..w_n|G)P(G)$$

Then we choose the gender by maximizing this quantity and using the naive independence assumptions:

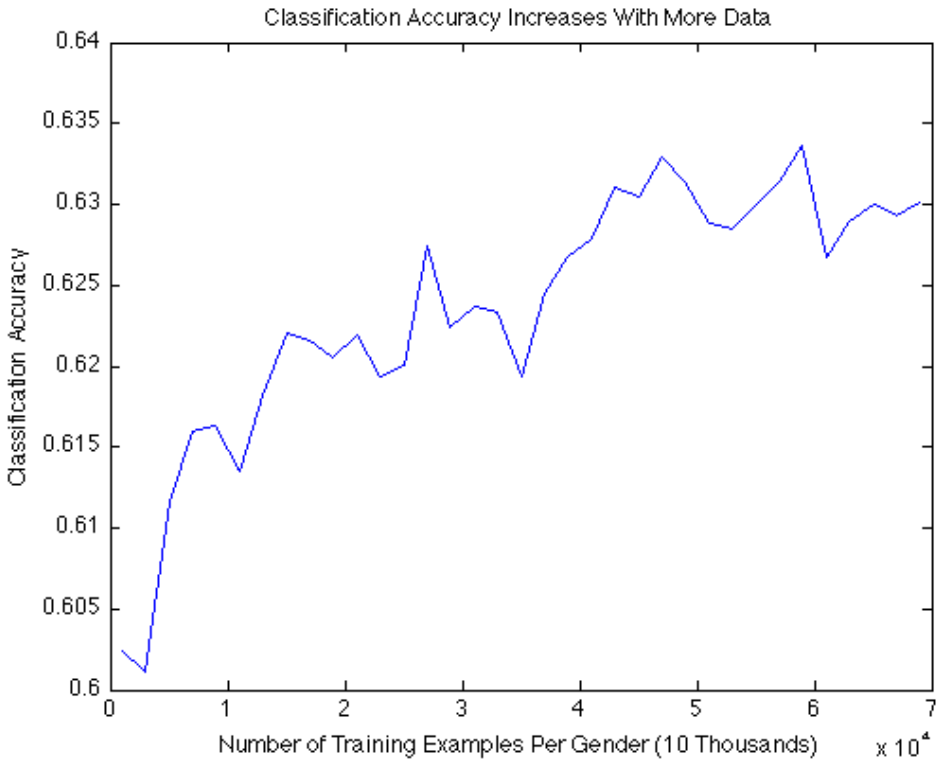
$$gender = \arg\max_{g} P(g) \prod_{i=1}^{n} P(w_i|g)$$

But since we are training on equal amounts of data the prior probability $P(g)$ is the same for both.

We tried preprocessing the data to see the effect that it had on the classifier, but the effect was minimal. We tried several techniques in preprocessing, mostly to reduce the number of types. An idea we had was to group related urls into simply their domain name, for example with YouTube urls. We also tried converting to lowercase and removing special characters and numbers. Most of these did not have that large of an impact, still staying between 61% and 64% accuracy. The best explanation we have for this is that with datasets so large, even if you find a meaningful pattern in the words or their orthographic structure, it is dominated so much by the words themselves. If you encode a special rule for urls, the problem is there just aren't enough urls compared to all other words to have a large impact.

8.1 Classification Accuracy with More Data

The amount of data was what has the largest consistent effect on performance. Below I have a graph showing how the amount of data affected the score, and we see a general trend of increasing accuracy as number of statuses moves from 1,000 to 70,000 examples.

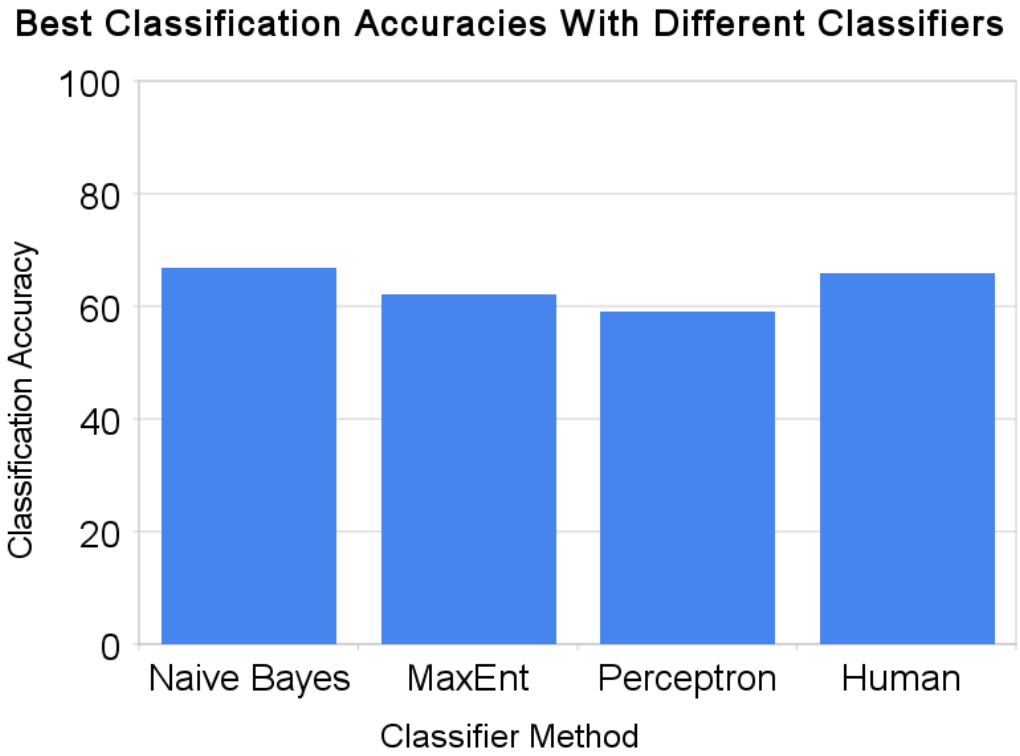


9 Bigram Status Generator

We created a bigram counter for the female and male training data and used it as the basis of a generative model. The statuses that are generated are not very good, but there are some important reasons why. First, this is using non-formatted strings. We are not using case-folding or removing formatting. Because of this you get statuses that keep the typos and formatting, but are not as sensible. If you did case folding and greatly lowered the number of types, you would generate statuses that don't look as genuine because they would be too uniform in orthographic structure. There are still important differences that you can see with the use of negative words and repeated letters and emoticons as mentioned before.

```
MALE:
just watched it again.. its rare breed and saw this again.

Had a bitch.....lol

Is finally able to sit back to chicago athletic program, after placing 5th with friends.

FEMALE:
So please stay focus... and was up!when I <3

Had a great time. We Know when your hell don't think that later
with so much to feed your memory of me....ahhhhhhh

We both been to my website and a poet. Plato Ogni cuore a dub dubby. rubber band! :]
```

10 Comparison of Different Classifiers



This graph shows a comparison of best scores on different classifiers. Naive Bayes got 67.7%, Maximum Entropy had a score of 62.2%, the perceptron algorithm got a score of 59% and humans got a score of 65.5%. Our conclusions are that this is a very difficult task. Often, statuses are too short to contain enough disambiguating information. There still are trends, and that is why we see scores near the 60% marker. However our results are extremely competitive with human prediction, and our best Naive Bayes results even exceed humans. If we wanted to improve the classifiers we would try to get more data, as we are hardly even scratching the surface of available data. We could also try to combine the classifiers into one classifier that takes input from all of them.

11 Web Interface

I build the web interface using PHP which provides a simple user interface for two parts of our project. It allows you to generate male and female statuses from our bigram model and it also allows you to classify a status using the Naive Bayes model. The web app is at `http://www.thekeesh.com/cs224n/`.

12 Partner Contributions

Each of us took a different approach to the task of gender classification of Facebook statuses. David worked on the perceptron and Winnow learning algorithm. Zach modified the MaxEnt classifier to

work with status updates and in creating specific feature patterns. Jeremy worked on collecting the Facebook data, implementing the Naive Bayes algorithm, making a bigram generative model, and also the web interface.

References

[1] Corney, Malcolm, de Vel, Olivier, Anderson, Alison, & Mohay, George. *Gender Preferential Text Mining of E-mail Discourse*. Las Vegas, NV 2002.

[2] Littlestone, Nick *Learning Quickly When Irrelevant Attributes Abound: A New Linear-threshold Algorithm*. `http://www.springerlink.com/content/j0k7t38567325716/fulltext.pdf` Santa Cruz, CA 1987.

[3] Ricciardi, Antonio *Classifying Second Life Player Gender Using Chat Data*. Stanford, CA 2009.

[4] Thomson, Rob & Murachver, Tamar. *Predicting gender from electronic discourse* University of Otago, New Zealand 2002.

[5] Ventura, Dan `http://axon.cs.byu.edu/Dan/478/misc/perceptron.pdf` 2009