# Assignment 1: Naive Bayes Classification

Meg Kobashi

September 2017

## 1 Logic

The multinomial model is trained according to the following formula:

$$y = \underset{y}{\text{argmax}}(\log P(y) + \sum_{n=1}^{\infty} \log P(x_i|y))$$

where

$$P(y) = \frac{\text{counts instances with label y}}{\text{counts instances}}$$

and

$$P(x_i|y) = \frac{\text{counts}(x_i,\, y)}{\text{counts}(x,\, y)} = \frac{\text{counts feature } x_i \text{ in instances with label y}}{\text{counts features in instances with label y}}$$

Oftentimes, an instance may have multiple features that are inherently different. Such a feature may be *bag of words* and *word length*, and the feature values will be a word or a count. It does not make sense to perform any aggregation across these different feature values, so conditional probabilities of the possible values are calculated within the category. To comply with this logic, the provided features were modified to be in the form:

{bag-of-words: {*dog*, ..., *cat*}, word-length: {*2*, ..., *4*}}

Generated from instances with these features, the multinomial model is a simple mapping of the features and their logarithmic conditional probabilities given the label:

{Male:{bag-of-words:{*dog*:-12.87, *cat*:-11.76}, word-len:{*2*:-11.89, *4*:-10.49 }}}
{Female:{word-len:{*dog*:-10.65, *cat*:-11.43}, word-length:{*2*:-11.80, *4*:-12.33 }}}

# 2 Improvements

## 2.1 Models

Using a binomial model, which records only the existence of features, yielded bad results compared to the multinomial model, which records the number of appearances of features. For example, the classification of the balanced blogs corpus had an accuracy of 45.69% for binomial, and 72.41% for multinomial.

## 2.2 Feature Engineering

Surprisingly, the highest scores were achieved when only simple pre-processing was applied to the raw documents. For the blogs corpus, the following operations improved the accuracy significantly.

- Lower case
- Removal of non-alphabetic symbols

On the other hand, the following preprocessing and features were attempted but proved to only decrease the accuracy by $\tilde{5}\%$.

- Removal of stopwords using NLTK
- Bi-grams
- Number of punctuations
- Word length

## 2.3 Smoothing

### 2.3.1 Motivation

Features that appear in the test set but are not present in the training set were not properly penalized. For example, the current even-odd classification model yields the following distribution:

$$\text{P(n \% 2 == 0 is True} \mid \text{Even)} = 1$$
$$\text{P(n \%2 == 0 is False} \mid \text{Even)} = 0$$

However, once we take the logarithm, the two probabilities will equal 0. (The logic defines $\log(0) = 0$.) Therefore, given the task of classifying say the number 2 with feature $n\%2 == 0$ is $True$, the labels $Even$ and $Odd$ are equally likely, which clearly is incorrect. Therefore, some probability greater than 0 but less than 1 must be assigned to unknown features in the model, which would yield a negative logarithm and serve as a penalty in classification.

#### 2.3.2 Laplacian Smoothing

$$P(x_i|y) = \frac{\text{counts}(x_i,\ \text{y}) + 1}{\text{counts}(x,\ y)\ + |V| + 1}$$

where $|V|$ is the range of feature values We add a new unknown word to the feature set, which is assigned a conditional probability:

$$P(UNKNOWN|y) = \frac{1}{\text{count(c)}+|V|\ +1}$$

## 3 Results

Here are the tabulated results with some metrics:

Table 1: Performance on Numbers

| | | Precision | Recall | F-Measure | Accuracy |
|---|---|---|---|---|---|
| Numbers | Even | 100.00 | 100.00 | 100.00 | 100.00 |
| | Odd | 100.00 | 100.00 | 100.00 | |

Table 2: Performance on List of Names

| | | Precision | Recall | F-Measure | Accuracy |
|---|---|---|---|---|---|
| Names | Male | 74.41 | 65.52 | 69.69 | 78.65 |
| | Female | 80.74 | 86.51 | 83.53 | |

Table 3: Performance on Blogs Corpus

| | | Precision | Recall | F-Measure | Accuracy |
|---|---|---|---|---|---|
| Balanced | Male | 73.33 | 68.14 | 70.64 | 72.41 |
| | Female | 71.65 | 77.12 | 74.29 | |
| Imbalanced | Male | 89.96 | 96.85 | 93.28 | 87.40 |
| | Female | 0.00 | 0.00 | 0.00 | |