CSE 5243: Introduction to Data Mining Assignment 2

Vaibhav Devekar  
Akshay Nikam

# Goal of the Assignment

This assignment aims at demonstrating various techniques to classifying the dataset based on feature vectors generated by Assignment 1.

# Classification Techniques

We implemented the following classification algorithms:

### Naïve Bayesian Classification:

The Naïve Bayesian classifier is based on applying the Bayes Theorem with independence assumptions. In this case we assume that the appearance of a word given topic is independent for the articles.

We apply the Bayes Theorem as:

*Where P(W|T) specifies the probability of appearance of W and T together in an article,  
 P(T) is probability of appearance of topic in an article.*

We need to compute this probability for every topic for a test instance. Since this is a multi-label classification, we take a look at number of actual topics, say t. We sort the probabilities in decreasing order and chose the first t topics to be our predicted topics. Based on predicted and actual topics, we update the metrics.

To compute P(W|T) , we maintain a count of each word and topic combination appearing together in the articles. The denominator is simply aggregate of all word and topic counts where the T is the topic. For optimization, we can ignore the P(W1 … Wn) term since they are common for each topic prediction.

Laplace correction: Some of the P(W|T) terms could be zero leading to skewed probabilities. To deal with this we perform the Laplace correction for all P(W|T) terms i.e. add 1 to numerator and size of vocabulary to denominator

P(W|T) = *where, V is the vocabulary in training set*

### K-Nearest Neighbor:

abc

# Naïve Bayesian Classifier logic

*Read the transaction feature vector  
Maintain a 2D dictionary for count to compute (W|T)   
Maintain a dictionary for topic count  
For each article:  
 For each topic:  
 Update topic count  
 For each word in document:  
 Update the (W|T) count*

*For each topic:  
 Precompute denominator of P(W|T)   
  
Apply Laplace correction to all (W|T) counts*

*For each test instance:  
 For each topic:  
 Maintain a variable to store P(T|W)  
 For each word in instance:  
 If known word, multiply with ((W|T) count/Precomputed denominator)  
 else multiply with (1/Precomputed denominator)  
 Multiply with P(T)*

*Reverse sort all the probabilities  
Given number of actual topics in test instance, pick that number of topics from the sorted list  
Based on matches, update metrics*

# Metrics

#### Accuracy

#### Precision

#### Recall

#### F-Measure

#### G – Mean

# Individual Contributions

Akshay implemented the K-Nearest Neighbor classifier while Vaibhav implemented the Naïve Bayesian classifier