Algorithms Implemented

The following algorithms were implemented from scratch:

1. **Naive Bayes** (NB) The vanilla NB was implemented (as discussed in class) except one small variation. To tackle the zero variance problem, a small correction parameter (ϵ) was added to the variance of every class.

   $$\epsilon = \kappa \times max_i(\texttt{variance}(x_i))$$

   where x_i is the ith feature and κ is set to 10^{-8}. [1]

2. **K-Nearest Neighbours** (KNN) Two versions of KNN were implemented by using *(1) Euclidean distance (ED)* and *(2) Cosine similarity (CS)* as the distance measure.

3. **Ensemble** (ENS) An ensemble of the two best performing model versions of NB and KNN was implemented. Let the final scores of the 20 classes computed by NB and KNN be s_{NB} and s_{KNN}. Then the final score is calculated as,

   $$s_{ENS} = \texttt{z-score-norm}(s_{KNN}) + \lambda \texttt{z-score-norm}(s_{NB})$$

   The final class (c) was then predicted as,

   $$c = \texttt{argmax}(s_{ENS})$$

Experiments

Experiments were conducted with three kinds of features,

1. TF-IDF
2. Bag of words (BOW) - From homework-2.
3. TF-IDF weighted word co-occurrence feature (WCO) - Frequency of occurrence of all bi-grams within a 2-word context window was computed using the training set. Thus, each word is represented as a vector of its co-occurrence frequencies with all other words in the vocabulary.

The only hyper-parameter tuned for NB was the vocabulary size (see 2). For KNN, nearest neighbours k was also tuned. The optimal value for k was found to be 1. For ENS, λ is a hyper-parameter tuned on the validation set. Optimal value of λ was found to be 0.6.

	Training accuracy	Validation accuracy	Online efficiency	Offline efficiency
NB+BOW	100%	84.44%	0.012 s	15.18 s
NB+TF-IDF	100%	84.14%	0.016 s	14.99 s
KNN+BOW(CS)	100%	69.77%	0.0021 s	0.025 s
KNN+BOW(ED)	100%	51.89%	0.0043 s	0.019 s
KNN+TF-IDF(CS)	100%	85.59%	0.0025 s	0.019 s
KNN+TF-IDF(ED)	100%	63.77%	0.0035 s	0.023 s
ENS	100%	**87.39**%	-	-

Table 1: Results

	NB	KNN
Vocab. size	40,000	20,000

Table 2: Vocabulary sizes for algorithms

	Validation accuracy
NB+WCO	63.45%
KNN+WCO	78.11%

Table 3: Performance of WCO features

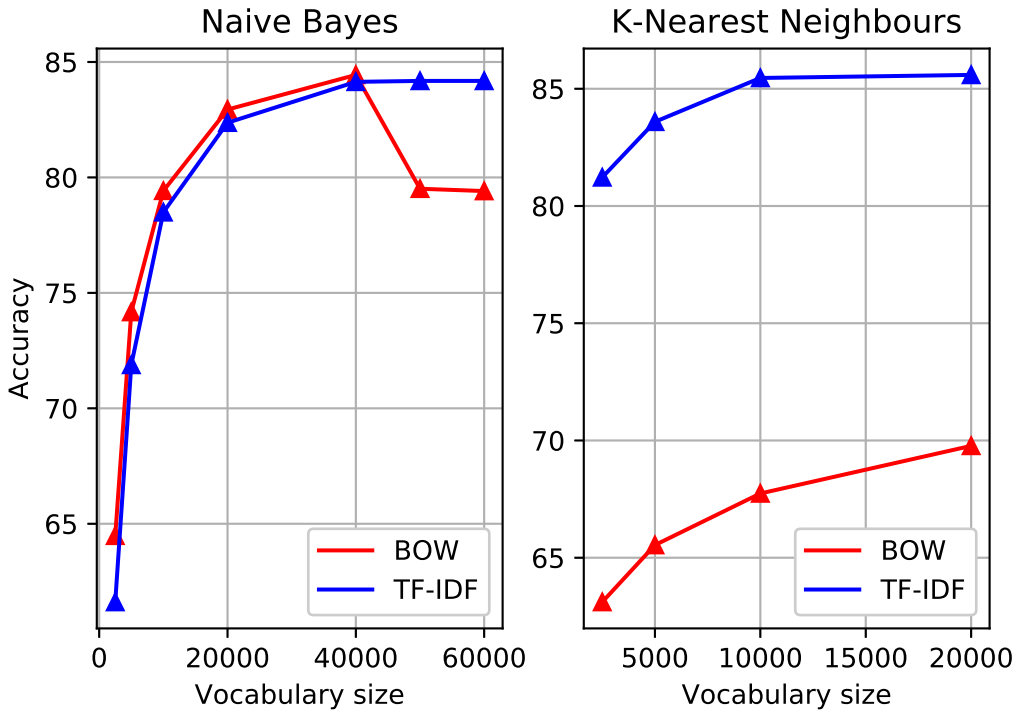

Figure 1: Effect of Vocabulary size

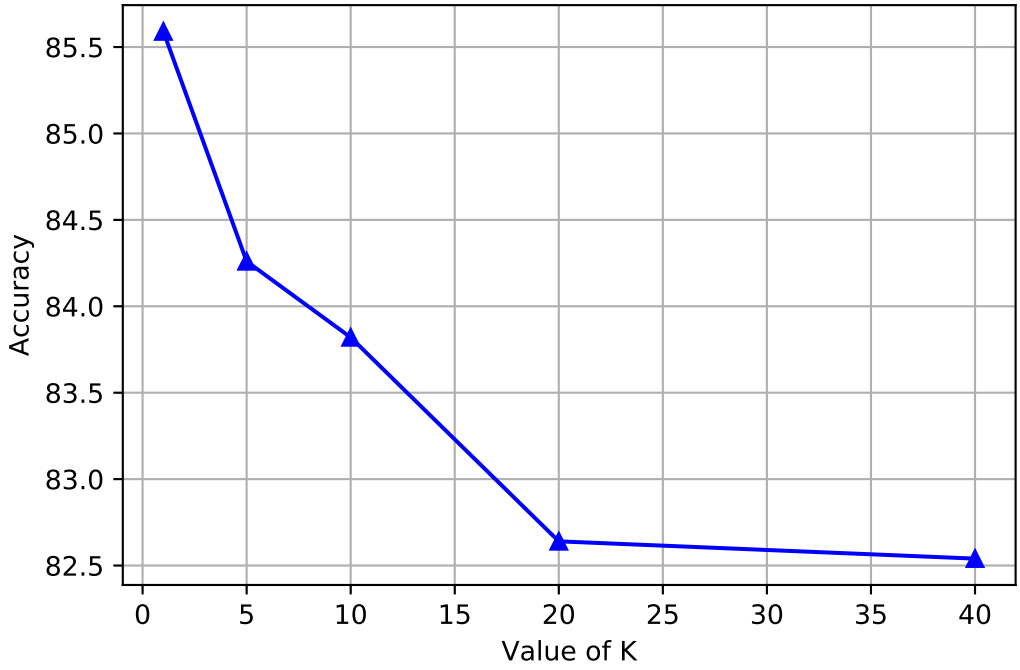

Figure 2: Effect of K in KNN

Results and Analysis

All results are reported in table 1. From the results, we can clearly see that ensemble method performs better than individual methods. This suggests that the results of KNN and NB complement each other.

I also report the results of using WCO features separately in table 3. The results of using this feature is not that impressive. The reason for this might be the lack of richness in the training documents. Word co-occurrence should work relatively better if formed from a rather large corpus. In our case, 20-news dataset contains only 9045 documents in the training set which result in a rather sparse and non-rich feature for words.

I show the effect of vocabulary size on accuracy of NB and KNN classifiers through figure 1. We can see that for NB classifier, the accuracy when using BOW features is slightly higher than when using TF-IDF features. Another observation is that when using TF-IDF with NB, the accuracy increases as we increase the vocabulary size but it's rate decreases. In contrast, accuracy when using BOW with NB starts to decrease after a vocabulary size of $40,000$.

For KNN, we can observe from figure 1 that there is a huge gap in accuracy when using TF-IDF versus BOW features. TD-IDF does exceptionally better with KNN.

[1]This is the same as variance smoothing implemented in `sklearn`.

In figure 2, I show the variation of accuracy with the value of K in KNN classifier. Lower values of K seem to work much better for 20-newsgroup dataset.