

---

# Fast classification of newsgroup posts

---

**Bo Moon**

Princeton University  
bhmoon@princeton.edu

**Jonathan Metzman**

Princeton University  
jmetzman@princeton.edu

## Abstract

WRITE ME!!!

## 1 Introduction

WRITE ME!!!

## 2 Related Work

We worked with the Ling-Spam dataset, a collection of spam emails and non

### 2.1 Data processing

We downloaded the 20 Newsgroups data set on January 5th, 2015 from the UCI Machine Learning Archive <sup>1</sup>. We used the Python NLTK library to tokenize, convert to lower case, remove stop words, lemmatize, stem each word using the Porter stemming method, and filtering words that occurred fewer than 200 times in the corpus [?]. The resulting vocabulary contained 3,256 words. We converted these words to the bag-of-words format as features of the posts; we used the newsgroup name that each document was posted to as its label. We considered the effect of feature selection on the classifiers, where feature selection is performed using a support vector machine with a linear kernel and  $\ell_1$  penalty. After feature selection, we were left with 488 dictionary words in the bag-of-words representation, which is 15% of the original feature set size.

### 2.2 Classification methods

We use ten different classification methods from the SciKitLearn Python libraries [?]. All parameterizations are the default unless specified.

1. *K-nearest neighbors* (KNN): using ten nearest neighbors and the “KDTree” algorithm
2. *Logistic regression with  $\ell_2$  penalty* (LR): using stochastic gradient descent
3. *Perceptron with  $\ell_2$  penalty* (P2): using stochastic gradient descent
4. *Hinge loss with  $\ell_2$  penalty* (HL2): using stochastic gradient descent
5. *Naive Bayes classifier* (NB): using multinomial implementation
6. *Support vector machine with linear kernel* (SVML):
7. *Support vector machine with squared exponential kernel* (SVMS):
8. *AdaBoost* (AB): using 100 estimators
9. *Decision tree* (DT): using Gini impurity scores

---

<sup>1</sup><http://kdd.ics.uci.edu/>

## 10. *Random forest* (RF): using Gini impurity scores and 100 trees

Because this problem is one of multiclass classification, we trained and tested each of these classifiers as binary classifiers for each class using one-versus-rest classification; for this we also used the SciKit-Learn library. Because there were equal numbers of samples in each class, we averaged the evaluation metrics across the twenty classes for the values across the 20 newsgroups.

### 2.3 Evaluation

For each classification method, we performed stratified 10-fold cross validation on the 20 News-groups data. We maintained the same folds across each of the classifiers. We compared the different classifier results using precision, false discovery rate (FDR),  $F_1$  score, and wall clock time in seconds, all of which were averaged over the results from each of the 20 classes (each of which had identical numbers of posts). In particular, denoting the number of false positives (FPs), true positives (TPs), false negatives (FNs), true negatives (TNs), and false discovery rate (FDR), we can define precision and recall as

$$\text{precision} = \frac{TP}{TP + FP} = 1 - \text{FDR}, \text{recall} = \frac{TP}{TP + FN}$$

and  $F_1$ -score, which is the harmonic mean of precision and recall:

$$F_1 = 2 \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = 2 \frac{TP}{2TP + FP + FN}.$$

## 3 Results

### 3.1 Evaluation results

We found variable performance of each of the ten classifiers on the 20 Newsgroups data, particularly with respect to recall (Table 1). In particular, we see that the linear classifiers (LR, P2, HL2, NB, and SVML) tend to perform worse on this task on average relative to the non-linear classifiers (AB, SVMS, DT, RF), although logistic regression has strong performance for a linear classifier. Of the non-linear classifiers, the AdaBoost, random forest, and decision tree classifiers have stronger performance than the SVMS. Overall, the decision tree shows competitive performance, despite its simplicity relative to the other nonlinear methods. This suggests that it is a non-additive combination of the word counts, rather than either a very high dimensional basis function (SVMS) or ensemble classifiers (RF, AdaBoost), that creates the biggest gains in performance over linear classifiers.

Classifier	No feature selection				Feature selection			
	Prec	Recall	$F_1$	Time (s)	Prec	Recall	$F_1$	Time (s)
KNN	0.77	0.34	0.45	1480.4	0.74	0.40	0.54	272.1
LR	0.81	0.89	0.85	23.7	0.76	0.91	0.82	14.8
P2	0.68	0.22	0.25	5.99	0.66	0.20	0.21	0.98
HL2	0.70	0.22	0.24	6.04	0.66	0.19	0.19	0.97
NB	0.39	0.94	0.53	4.34	0.36	0.95	0.50	0.69
SVML	0.78	0.90	0.83	11.1	0.65	0.90	0.75	4.1
SVMS	0.83	0.68	0.74	929.9	0.84	0.89	0.86	150.6
AB	0.88	0.94	0.91	1424.2	0.89	0.94	0.91	178.4
DT	0.89	0.93	0.91	24.2	0.89	0.93	0.90	3.0
RF	0.87	0.79	0.83	28.5	0.89	0.89	0.89	4.5

Table 1: **Results from ten classifiers on 20 Newsgroups data.** For each classifier, we report precision, recall,  $F_1$ -scores, and wall clock time in seconds for the one-versus-rest classification task with 10-fold cross validation.

For each of the one-versus-rest classification problems, we extracted the *Gini* impurity scores from a trained random forest classifier. Gini impurity for a particular feature represents the information gain, or the average reduction in entropy of the classifier before and after a that feature is used across all of the trees in the random forest; a larger value indicates greater predictive power. We

found that the top ten words, with respect to Gini impurity, for each class (Table 2) were representative of the topics discussed in that newsgroup (e.g., `talk.religion.misc` includes *moral* and *religion*); there were also a number of words that were in the top ten word lists for a number of newsgroup classes, suggesting that they were important in discriminating one of the newsgroups with a 0 label from the newsgroup with the 1 label (e.g., *mideast* is a strong indicator of `talk.politics.mideast`).

rec.motorcycles	comp.sys.mac.hardware	talk.politics.misc	soc.religion.christian	comp.graphics	sci.med	talk.religion.misc
Austin b biblic columbia doctor ecn motif reason say speed	apollo central come handheld i3150101 me patient q sun win	access also client crabapple gateway lc mideast operation point take	apple atheist chip geb go matter rec religion run sin	3 come get govern handheld IGC mideast sun win write	digest do font mchp med path pitch say since Toronto	aft chip mideast moral package point religion sale take (quote)
comp.windows.x	comp.sys.ibm.pc.hardware	talk.politics.guns	alt.atheism	comp.os.ms-windows.misc	sci.crypt	sci.space
come engr law mideast motherboard well win write x xliv	car come handheld i3150101 ID me mideast patient science sun	astro baseball fan fire ground mideast point take we watson	also ATF atheism Canada internet jim king mideast religion take	come distribute drive FBI love mideast nuclear take win write	child Clinton crime ee electron kent order printer say sea	AI ask audio device larc monitor oracle say software z
misc.forsale	rec.sport.hockey	rec.sport.baseball	sci.electronics	rec.autos	talk.politics.mideast	
zoo come comp format mideast observe rate rec Rutgers sgi	b c enterpoop f g HIV reason speed tax widget	b c clock g Henry HIV picture reason speed tax	au batf ci come crime effect mchp say software vm	Austin cantaloup de eng food mideast motif reason say speed	April Arizona cso Islam Israel Michael pain point SNI tu	

Table 2: **Top 10 predictive words for each of the 20 Newsgroups.** The top ten words were identified after feature selection from a fitted random forest classifier using words ranked by their Gini impurity scores.

### 3.2 Computational speed

The variability in the time for training and testing these linear classifiers was substantial (Table 1). In particular, we found that the KNN classifier, which does not perform training, takes the largest amount of time because of the all-by-all comparison that occurs during test phase. AdaBoost takes the second longest, but here the time is spent on training the weak classifiers and the weights of the linear combination of those weak classifiers. The fastest classifiers include the NB classifier, the perceptron, and the hinge loss classifier, followed by the linear SVM and then the decision tree and random forest classifiers.

### 3.3 Feature selection

These results highlight the benefits for some of the methods of reducing the number of features before training the classifiers. In particular, we found that using feature selection improved the precision for SVMS, AB, and RF classifiers, and the recall for KNN, LR, NB, SVMS, and RF classifiers. The largest improvement was for the SVMS and RF classifiers. The effect on the RF classifier might be mitigated by increasing the number of trees in the random forest for larger numbers of features, although this would slow down the training time proportionally. Across all methods, feature selection substantially improved the average wall clock time, e.g., improving the time of AdaBoost by 87.5%.

## 4 Discussion and Conclusion

In this work, we compared ten different classifiers to predict the newsgroup for a particular newsgroup post using bag-of-words features. We found that, considering precision, recall, and time, the decision tree and random forest classifiers showed superior performance on this task. The effect of feature selection was mostly on the time, although the improvement in performance was substantial for the random forest classifier on this task.

There are a number of directions to go that would improve these results. First, we could expand our data set using available data from these and other related newsgroups. Second, we could consider more sophisticated features for the newsgroup posts than dictionary word counts; bi-grams, post length, or punctuation may be useful in this classification task. Third, we could use the most promising models in a more problem-tailored way. In particular, because the random forest classifier showed such promise in this task, we could consider applying it to this problem using multi class labels instead of one-versus-rest class labels, and reducing the dimension of the feature space using supervised latent Dirichlet allocation based methods [?, ?].