

Homework #4 (due in 3 weeks)

There will be four more homework assignments (including this one). The other three assignments will cover internet search, multilingual retrieval, and applications of NLP to IR. These assignments should involve less work (i.e., less programming) compared to the first three assignments to give you more time to devote to your independent project. Also, you only need to hand in three of the remaining assignments; you can skip the assignment of your choice.

Text Classification (40 points)

(20 pts) Explain the following concepts and what they mean in terms of text classification: (1) *bias-variance tradeoff*; and (2) *kernel trick*.

(20 pts) Compute how a naïve Bayes classifier would classify the following 'documents' using the binomial model. The two classes are 'Good' and 'Spam'. Recall in the binomial model that estimates of $P(\text{word}|\text{class})$ are based on the percentage of documents of the class containing the word. Estimates of $P(\text{class})$ and $P(\text{word}|\text{class})$ are given in the tables below.

Document 1: "free drugs william"

Words	$P(w \text{Good})$	$P(w \text{Spam})$
drugs	0.03	0.15
free	0.01	0.25
please	0.03	0.025
william	0.05	0.005

Document 2: "free william please"

$P(\text{Good}) = 0.7$
$P(\text{Spam}) = 0.3$

Binary classification using Reuters 21578 dataset (60 points)

(60 pts) Download from the course web site training documents for three Reuters categories (coffee, ship, and wheat) and build a classifier for each. You can use any method (e.g., kNN, naïve Bayes, decision trees, or SVMs). I suggest using the SVMlight tool which is available from <http://svmlight.joachims.org/> (binaries are available for modern operating systems). Generally you should process the text files, which are similarly formatted to HWs 1-3, and write out files of vectors, one document vector per line. For example:

```
+1 5:1 13:1 78:1 ... 15008:1
+1 5:1 45:1 78:1 15000:1
-1 3:1 13:1 87:1 12000:1
```

“+1” in the leftmost column indicates that the vector is positive for the class and “-1” indicates it is negative. Each *termid:value* element in this example uses binary weights; ‘1’ indicates the presence of a term in the document, and terms not in the document (i.e., the zeros) are not written out. Having created these vectors, train a classifier for each class and then run the classifier on the test sets.

The example above is in the format SVMlight expects. To train a model with SVMlight:

```
% svm_learn coffee.train coffee.mod
```

To run a test set against a trained model:

```
% svm_classify coffee.test coffee.mod coffee.out
```

Using the output predictions (+1/-1) compute recall, precision, and F_1 scores for each class. Recall is the percentage of +1s in the test file that were correctly predicted to belong to the class; precision is the percentage of +1s in the output file that are correct according to the test file labels. $F_1 = 2 * P * R / (P + R)$. Finally, repeat your work under a different experimental condition. For example, compare the use of stemming and stopword removal against no stemming and no stopword removal. Or use a different learning method. Or use *tf/idf* instead of binary weights.

Describe the methods and tools you used and the differences observed in your two experimental conditions. Also hand in any source code that you write for the assignment.