**Machine Learning Assignment 1**

Liang Shuailong 1000829

**Task: Documentation classification**

1. It is a multiclass classification problem. The input X is a matrix, each row of which is a feature vector corresponding the document. The output is a column vector, and each element is the label representing the class(category) of the document. For binary classification case, the output is just 1 and -1 for two classes.

In class we talked about the Bag-of-Word approach to represent a document by the term frequencies. So for now we can use this approach to extract the feature vector for each document. The algorithm is described in the pseudocode below:

|  |
| --- |
| def feature\_extractor(X):  features = []  vocabulary = set()  for each document d in training set:  add tokens in d to vocabulary  for each document:  for each token in vocabulary:  count frequencies of token in d  feature = frequencies of each token in the vocabulary  append feature to features  return features |

1. The terminate condition is that in one loop of scanning the data points, the number of mislabeled data reaches 0. The result is shown in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Training accuracy | Test accuracy | Iterations |
| **Perceptron** | 100% | 93.7% | 80 |
| **Averaged perceptron** | 97.76% | 91.51% | 80 |

If we terminate the program after 100 loops, the result is shown in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Training accuracy | Test accuracy | Iterations |
| **Perceptron** | 100% | 93.7% | 100 |
| **Averaged perceptron** | 97.76% | 91.51% | 100 |

Since after 80 iterations, any data points in training set can not update theta anymore. So the theta will not be updated anymore, and causing no difference to the result.

From the table above we can see that averaged version of perceptron does not improve the result. It decreases the result a little bit actually. Theoretically, averaged version of perceptron should generalize better, however in here it is not the case. The reason may be…