

# LIN570: HW7 – mallet (100pts)

YOUR NAME (UW NetID)

Due date: 11pm on Nov 19, 2019 (Tuesday)

The goal of this assignment is to use the Mallet package for the text classification task. All the examples are under `~/dropbox/19-20/570/hw7/`. Let `$dataDir` be `hw7/20_newsgroups`, and `$exDir` be `hw7/examples/`.

Note:

- When you type the commands, you need to replace `$dataDir` with `/dropbox/19-20/570/hw7/20_newsgroups` and `$exDir` with `/dropbox/19-20/570/hw7/examples`.
- All the options of Mallet commands (e.g., `--input`) start with two `-s`, not one `-`.
- Use the Mallet package on `patas`, which is the correct version for this assignment.

Questions:

## 1. Q1 (10 points): Learning the Mallet commands

- 1 point:** Check out Mallet website at <http://mallet.cs.umass.edu/> and focus on the classification part. Go over the mallet slides and set up your `PATH` and `CLASSPATH` on `patas` properly.
- 1 point:** Run the following command to create a data vector, `politics.vectors`, using the data from the three `talk.politics.* newsgroups`:  

```
mallet import-dir --input $dataDir/talk.politics.* --skip-header --output politics.vectors
```
- 1 point:** Run the following command to convert `politics.vectors` to the text format `politics.vectors.txt`.  

```
vectors2info --input politics.vectors --print-matrix siw > politics.vectors.txt
```
- 1 point:** Run the following command to split `politics.vectors` into training (90% of the data) and testing files (10% of the data):  

```
vectors2vectors --input politics.vectors --training-portion 0.9 --training-file train1.vectors --testing-file test1.vectors
```
- 1 point:** Run the following command to train and test. The training and test accuracy is at the end of `dt.stdout`.  

```
vectors2classify --training-file train1.vectors --testing-file test1.vectors --trainer DecisionTree > dt.stdout 2>dt.stderr
```

Table 1: Classification results for Q1(e)

	Training accuracy	Test accuracy
NaiveBayes		
MaxEnt		
DecisionTree		
Winnow		
BalancedWinnow		

(f) **5 points:** Run `vectors2classify` to classify the data with five learners and complete Table 1.

- Use the `train.vectors` and `test.vectors` under `$exDir` for this classification task.
- The names of the five learners are: NaiveBayes, MaxEnt, DecisionTree, Winnow, and BalancedWinnow.
- The command for classification is:  
`vectors2classify --training-file $exDir/train.vectors --testing-file $exDir/test.vectors --trainer $zz > $zz.stdout 2>$zz.stderr`

whereas `$zz` is the name of a learner (e.g., MaxEnt).

2. **Q2 (25 points):** Write a script, `proc_file.sh`, that processes a document and prints out the feature vectors.

- The command line is: `proc_file.sh input_file targetLabel output_file`
- The `input_file` is a text file (e.g., `input_ex`).
- The `output_file` has only one line with the format (e.g., `output_ex`):  
`instanceName targetLabel f1 v1 f2 v2 ...`
  - The `instanceName` is the filename of the `input_file`.
  - The `targetLabel` is the second argument of the command line.
- To generate the feature vector, the code should do the following:
  - First, skip the header; that is, the text before the first blank line should be ignored.
  - Next, replace all the chars that are not `[a-zA-Z]` with whitespace, and lowercase all the remaining chars.
  - Finally, break the text into token by whitespace, and each token will become a feature.
  - The value of a feature is the number of occurrences of the token in `input_file`.
  - The `(featname, value)` pairs in the feature vector are ordered by the spelling of the `featname`.
- For instance, running `proc_file.sh $exDir/input_ex c1 output_ex` will produce `output_ex` as the one under the `$exDir`.

3. **Q3 (25 points):** Write a script, `create_vectors.sh`, that creates training and test vectors from several directories of documents. This script has the same function as `mallet`

`import-dir`, except that the vectors produced by this script are in the text format and the training/test split is not random.

- The command line is: `create_vectors.sh train_vector_file test_vector_file ratio dir1 dir2 ...`.  
That is, the command line should include one or more directories.
- `ratio` is the portion of the training data. For instance, if the ratio is 0.9, then the FIRST 90% of the FILES in EACH directory should be treated as the training data, and the remaining 10% should be treated as the test data. By the first x%, we mean the top x% when one runs `ls dir`.
- `train_vector_file` and `test_vector_file` are the output files and they are the training and test vectors in the text format (the same format as the `output_file` in Q2).
- The class label is the basename of an input directory. For instance, if a directory is `hw7/20_newsgroups/talk.politics.misc`, the class label for every file under that directory should be `talk.politics.misc`.

4. **Q4 (15 points):** Classify the documents in the `talk.politics.*` groups under `$dataDir`.

- Run `create_vectors.sh` from Q3 with the ratio being 0.9, and the directories being `talk.politics.guns`, `talk.politics.mideast`, and `talk.politics.misc`.
  - The `train_vector_file` and `test_vector_file` should be called `train.vectors.txt` and `test.vectors.txt`, respectively.
- Run `mallet import-file` to convert the training and test vectors from the text format to the binary format.
  - The binary vector files should be called `train.vectors` and `test.vectors`, respectively.
  - Suppose you run `mallet import-file` first on `train_vector_file` and create `train.vectors`. When you run `mallet import-file` next on the `test_vector_file`, remember to use the option `--use-pipe-from train.vectors`. That way, the two vector files will use the same mapping to map feature names to feature indexes.
- Run `vectors2classify` for training (with MaxEnt trainer) and for testing.
  - The MaxEnt model file should be called `me-model`
  - Redirect stdout to a file called `me.stdout` and stderr to a file called `me.stderr`.
- What are the training and test accuracy?

The submission should include:

- The `readme.[txt|pdf]` file that includes Table 1 (no need to submit anything else for Q1) and training and test accuracy in Q4.
- `hw.tar.gz` that includes `proc_file.sh`, `create_vectors.sh`, and the files created in Q4 (see the complete list in `submit-file-list`).