CS410 MP5 – Text Categorization

Due: April 30, 2017 at 11:59pm

Text categorization is a very important technique for text data mining and analytics. It is closely related with the discovery of various different kinds of knowledge, including topic mining, sentiment analysis, text-based prediction, etc. There're many models and algorithms for text categorization, including the generative probabilistic model Naïve Bayes Classifier, and deterministic models such as Logistic Regression and Support Vector Machine.

In this assignment, you will have an opportunity to experiment with some classifiers. Specifically, you will use MeTA to run Naïve Bayes and SVM on a dataset and explore the results.

1. Setup

Installation: Make sure you have the most up-to-date version of MeTA.

```
# update meta
```

```
pip install --upgrade pip
pip install --upgrade metapy
```

```
# add --user argument if you are running on EWS
```

Download the dataset and place it in directory for MP5 (along with the provided *python scripts* and *stop-words file*):

```
# download datasets
```

```
wget http://sifaka.cs.uiuc.edu/ir/textdatabook/icnale.tar.gz
tar -xzvf icnale.tar.gz
```

Setting the config file: You should edit `config.toml` and change the dataset:

```
dataset = “icnale”
inverted-index = “icnale-inv”
forward-index = “icnale-fwd”
```

Note: please type these lines instead of copying and pasting them to your config file. Sometimes the encoding format of the quotes (“”) raise error in MeTA.

2. Running a classifier

Classifiers:

In this assignment, we will work with two different classifiers: Naïve Bayes and SVM.

MeTA's implementation of SVM is actually an approximation using stochastic gradient descent on hinge loss. It's implemented as a `BinaryClassifier`, so we will need to adapt it before it can be used to solve our multi-class classification problem. MeTA provides two different adapters for this scenario: `One-vs-All` and `One-vs-One`.

Command:

The python scripts (Naïve Bayes `cls_nb.py` and SVM `cls_svm.py`) are provided in the zip file. Make sure you keep the downloaded dataset, the config file `config.toml` and the scripts in a same directory.

To run a Naïve Bayes classifier with *5*-fold cross validation, you can simply run the following command:

```
python cls_nb.py config.toml 5
```

where the number `5` is the value of parameter `k` for ***k***-fold cross validation.

Example Output:

Here's some example output on a toy dataset with 3 classes (Yours should have similar structure for `ICNALE` dataset).

This dataset is a small collection of essays written with different first languages. We have in total 1008 data points, with 3 labels of categories (`japanese`, `chinese` and `english`).

```
#instances: 1008
#labels: 3
labels: set([u'japanese', u'chinese', u'english'])
```

The `cross_validate()` method in MeTA returns a `Confusion Matrix`. Each row shows how the prediction for that row label was distributed across all the other labels.

For example, take a look at the first row. The `confusion matrix` shows that 87% of the true `chinese` labels were predicted correctly. It also shows that 3.26% of the `chinese` labels were miscategorized as `english`. Thus, the `confusion matrix` allows us to exactly see where the classifier has made mistakes.

```
             chinese    english    japanese
           ------------------------------
   chinese | 0.87       0.0326     0.0978
   english | 0.00694    0.958      0.0347
  japanese | 0.0065     0.0091     0.984
```

Finally, MeTA shows another summary of the classification results. You should know what the three measures below are based on our discussions in class.

```
-----------------------------------------------------------
Class       F1 Score    Precision   Recall      Class Dist
-----------------------------------------------------------
chinese     0.899       0.93        0.87        0.0915
english     0.942       0.927       0.959       0.144
japanese    0.984       0.983       0.984       0.764
-----------------------------------------------------------
Total       0.97        0.97        0.97
-----------------------------------------------------------
1005 predictions attempted, overall accuracy: 0.97
```

3. Tasks & Questions

Q1. (30pt) Run Naïve Bayes on ICNALE dataset with 5-fold cross validation.

1) **(20pt)** Report the overall accuracy (from performance summary), and accuracy for each class (from the confusion matrix).

Report format:

```
          format                       example
Q1.1:                          Q1.1:
accuracy=                      accuracy=0.900
CHN:                           CHN:0.800
ENS:                           ENS:0.850
……                             ……
```

2) **(10pt)** Explain how to read a confusion matrix in order to determine two classes that are often mistaken for each other by the classifier. For example, in the ICNALE dataset, what are two classes that are often confused with each other?

Report format:

```
Q1.2:
[your answer]
```

Q2. (40pt) Run SVM on ICNALE dataset with k = 5, 10.

1) **(30pt)** Report the overall accuracy (from performance summary), and accuracy for each class (from the confusion matrix).

Report format:

```
Q2.1:
#k=5
accuracy=
CHN:
ENS:
……
#k=10
accuracy=
CHN:
ENS:
……
```

2) **(10pt)** We evaluate the classifier with 5-fold cross validation and 10-fold cross validation. Which (if either) do you think gives a higher accuracy? Is it consistent with your results? Why?

Report format:

```
Q2.2:
[your answer]
```

Q3. (30pt) Run different config settings with SVM (k=5) on ICNALE dataset.

1) **(10pt)** Modify settings in `config.toml`, to also use bigrams instead of purely unigram words. Observe and report the overall performance.

***Notice:** If you change an analyzer section, you'll have to **delete** the existing index folder (or change the value for `index` in config file) in order to have your changes take effect!

2) **(20pt, general question)** Do bigram features improve performance? What would be the advantages and problems with bigrams compared with unigram features? (***Hint***: think about accuracy, efficiency, example cases where the feature works/has limitations) What do you think is a better strategy to obtain better performance? (***Hint***: propose other features? selection? combination?)

Report format:

```
Q3.1:
accuracy=
Q3.2:
##evaluation:
[advantages and problems]
##comment:
[your strategy]
```

4. Submission

Please submit a plain text file `mp5-results.txt`. We will use an auto-grader to parse your answers, so you **MUST** structure your answers strictly following the format described in section 3.

Please submit the file to compass by the due day **(April 30, 2017 at 11:59pm)**