Assignment 3

Language Identification

Verena Blaschke

June 27, 2018

Assignment 3

I: Gathering the data

II: Feature extraction

III: Logistic regression

IV: Precision, recall, F-score

V: K-fold cross validation

VI: Model selection

VII: Challenge

I: Gathering the data

getting the file names from the command line (UNIX):

```
python3 download-tweets.py train/*.id
```

I: Gathering the data

```
auth = tweepy.OAuthHandler(CONSUMER_KEY,
                           CONSUMER_SECRET)
api = tweepy.API(auth,
                 wait_on_rate_limit=True,
                 wait_on_rate_limit_notify=True)
...
tweet = api.get_status(tweet_id)
```

I: Gathering the data

```
auth = tweepy.OAuthHandler(CONSUMER_KEY,
                           CONSUMER_SECRET)
api = tweepy.API(auth,
                 wait_on_rate_limit=True,
                 wait_on_rate_limit_notify=True)
...
tweet = api.get_status(tweet_id)
```

- Catch errors:
  - invalid IDs
  - deleted/private tweets

I: Gathering the data

```
auth = tweepy.OAuthHandler(CONSUMER_KEY,
                           CONSUMER_SECRET)
api = tweepy.API(auth,
                 wait_on_rate_limit=True,
                 wait_on_rate_limit_notify=True)
...
tweet = api.get_status(tweet_id)
```

- Catch errors:
  - invalid IDs
  - deleted/private tweets
- Everyone's corpora might be slightly different (depending on when you downloaded the tweets).

I: Gathering the data

```
writer = csv.writer(filename,
                    delimiter=',', quotechar='"')
...
writer.writerow([lang, tweet_id, tweet.text])
```

- Add the CSV file to your repo!

II: Feature extraction

Use exactly those bigrams that are in the tweet or...

- Change the case?
- Remove whitespace/special characters/URLs?
- Add padding?
  `<BOS>my tweet<EOS>`
  → `<BOS>m, my, y␣, ␣t, tw, we, ee, et, t<EOS>`

II: Feature extraction

- Get the bigram tallies for all tweets.
- Decide on an order of bigrams/columns.
- Fill the matrix.

II: Feature extraction

- Get the bigram tallies for all tweets.
- Decide on an order of bigrams/columns.
- Fill the matrix.

```
mat = scipy.sparse.dok_matrix((n_samples, n_bigrams),
                              dtype=np.int16)
...
mat[i, j] = count_of_bigram_j
```

II: Feature extraction

Make sure that...

- the samples and language labels still correspond to one another afterwards.
  - ⚠shuffling the samples
  - ⚠dictionaries/sets

II: Feature extraction

Make sure that...

- the samples and language labels still correspond to one another afterwards.
  - ⚠shuffling the samples
  - ⚠dictionaries/sets
- features extracted from a test set use the same bigram-to-column index mapping.
  - Even if the training set includes bigrams that are not in the test set,
  - or vice versa.

III: Logistic regression

```
clf = sklearn.linear_model.LogisticRegression()
clf.fit(features, labels)
clf.score(features, labels)
```

IV: Precision, recall, F-score

$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{recall} = \frac{TP}{TP + FN}$$

$$\text{F1-score} = \frac{2 \times precision \times recall}{precision + recall}$$

IV: Precision, recall, F-score

$$\text{prec} = \frac{TP}{TP + FP} \qquad \text{rec} = \frac{TP}{TP + FN} \qquad \text{F1-score} = \frac{2 \times prec \times rec}{prec + rec}$$

How to extend this from a binary measure to a multi-class measure?

IV: Precision, recall, F-score

$$\text{prec} = \frac{TP}{TP + FP} \qquad \text{rec} = \frac{TP}{TP + FN} \qquad \text{F1-score} = \frac{2 \times prec \times rec}{prec + rec}$$

How to extend this from a binary measure to a multi-class measure?

- Let one language label be the 'positive' class (all other languages forming the 'negative' class).
- Compute precision, recall, F1-score.
- Repeat this for all language labels and average over the scores.

IV: Precision, recall, F-score

$$\text{prec} = \frac{TP}{TP + FP} \qquad \text{rec} = \frac{TP}{TP + FN} \qquad \text{F1-score} = \frac{2 \times prec \times rec}{prec + rec}$$

How to extend this from a binary measure to a multi-class measure?

- Let one language label be the 'positive' class (all other languages forming the 'negative' class).
- Compute precision, recall, F1-score.
- Repeat this for all language labels and average over the scores.

$$\text{precision}_\text{M} = \frac{\sum_i^C \text{precision}_\text{i}}{C} \qquad \text{recall}_\text{M} = \frac{\sum_i^C \text{recall}_\text{i}}{C}$$

$$\text{F1-score}_\text{M} = \frac{\sum_i^C \text{F1-score}_\text{i}}{C}$$

IV: Precision, recall, F-score

$$\text{prec} = \frac{TP}{TP + FP} \qquad \text{rec} = \frac{TP}{TP + FN} \qquad \text{F1-score} = \frac{2 \times prec \times rec}{prec + rec}$$

Ambiguities:

- What if $TP + FP = 0$ ($TP + FP = 0$; $prec + rec = 0$)?

IV: Precision, recall, F-score

$$\text{prec} = \frac{TP}{TP + FP} \qquad \text{rec} = \frac{TP}{TP + FN} \qquad \text{F1-score} = \frac{2 \times prec \times rec}{prec + rec}$$

Ambiguities:

- What if $TP + FP = 0$ $(TP + FP = 0;\ prec + rec = 0)$?
  - Typically: precision $= 0$ (recall $= 0$; F1-score $= 0$)

IV: Precision, recall, F-score

$$\text{prec} = \frac{TP}{TP + FP} \qquad \text{rec} = \frac{TP}{TP + FN} \qquad \text{F1-score} = \frac{2 \times prec \times rec}{prec + rec}$$

Ambiguities:

- What if $TP + FP = 0$ ($TP + FP = 0$; $prec + rec = 0$)?
  - Typically: precision $= 0$ (recall $= 0$; F1-score $= 0$)
- What set of labels to use if the predicted label sequence contains classes that do not appear in the gold-standard sequence?

IV: Precision, recall, F-score

$$\text{prec} = \frac{TP}{TP + FP} \qquad \text{rec} = \frac{TP}{TP + FN} \qquad \text{F1-score} = \frac{2 \times prec \times rec}{prec + rec}$$

Ambiguities:

- What if $TP + FP = 0$ ($TP + FP = 0$; $prec + rec = 0$)?
  - Typically: precision $= 0$ (recall $= 0$; F1-score $= 0$)
- What set of labels to use if the predicted label sequence contains classes that do not appear in the gold-standard sequence?
  - Only the classes from the gold-standard sequence?
  - The union of the classes from both sequences?

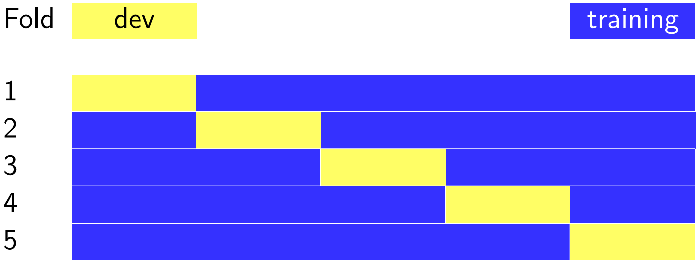
IV: Precision, recall, F-score

$$\text{prec} = \frac{TP}{TP + FP} \qquad \text{rec} = \frac{TP}{TP + FN} \qquad \text{F1-score} = \frac{2 \times prec \times rec}{prec + rec}$$

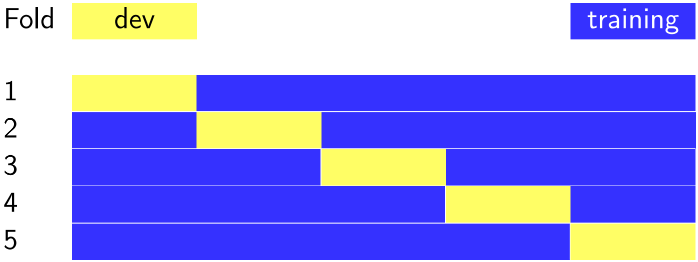
Ambiguities:

- What if $TP + FP = 0$ $\left(TP + FP = 0;\ prec + rec = 0\right)$?
  - Typically: precision $= 0$ (recall $= 0$; F1-score $= 0$)
- What set of labels to use if the predicted label sequence contains classes that do not appear in the gold-standard sequence?
  - Only the classes from the gold-standard sequence?
  - The union of the classes from both sequences?
- What set of labels to use if the gold-standard sequence for the test set does not contain all labels from the training set?

V: K-fold cross validation

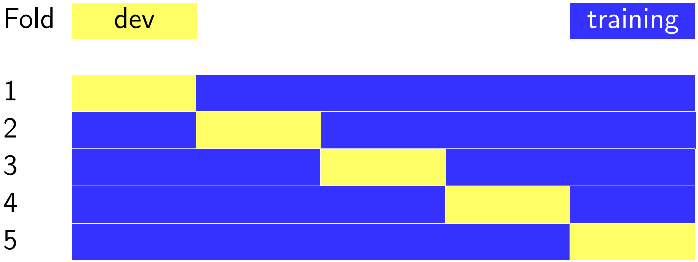


V: K-fold cross validation



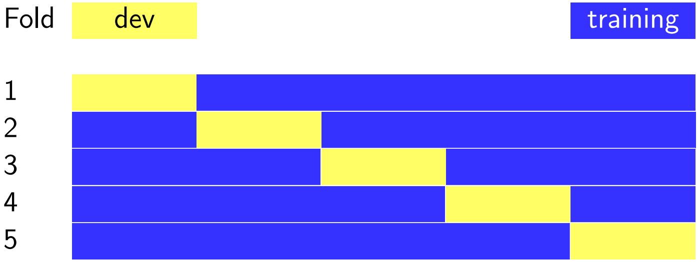
- There should be no overlap between the development partitions across different folds.

V: K-fold cross validation



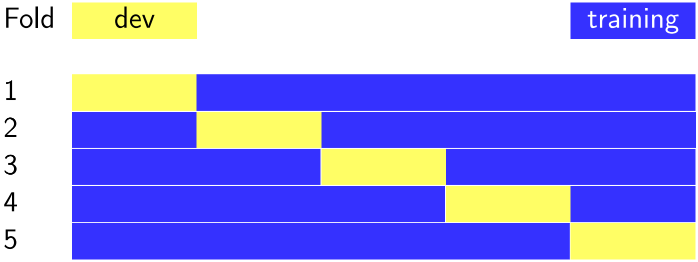
- There should be no overlap between the development partitions across different folds.
- What if the number of samples is **not** divisible by the number of folds (without remainder)?
  - Easiest solutions: Add the remainder to the last partition or exclude it completely.

V: K-fold cross validation



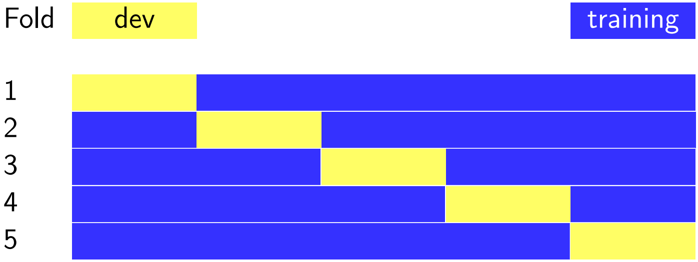
- Should we shuffle the order of the samples prior (once) prior to partitioning the data set, or manually make sure the partitions contain similar proportions of labels?

V: K-fold cross validation



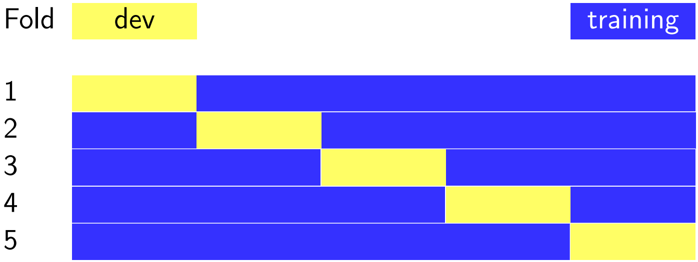
- Should we shuffle the order of the samples prior (once) prior to partitioning the data set, or manually make sure the partitions contain similar proportions of labels? (Yes!)

V: K-fold cross validation



- Should we shuffle the order of the samples prior (once) prior to partitioning the data set, or manually make sure the partitions contain similar proportions of labels? (Yes!)
- For each fold, exclude bigrams/columns that do **not** appear in the reduced training set?

V: K-fold cross validation



For each fold:

- Train the model on the training partition.
- Get predictions for the development partition.
- Calculate the macro-averaged scores for these predictions.

Then, calculate the mean of the 5 macro-averaged precision/recall/F1 scores.

VI: Model selection

Get a parameter value that lets the model perform well on unseen data.

VI: Model selection

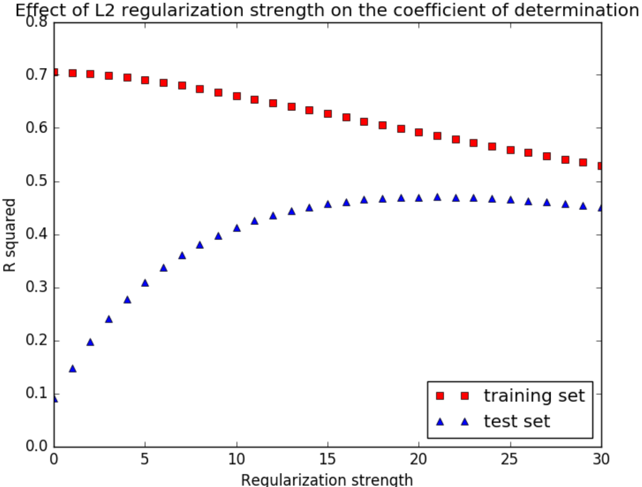
Get a **parameter value** that lets the model perform well on unseen data.

- Try out different values for `C` (inverse of regularization strength).
- Each time, get the mean of the macro-averaged F1 score for k-fold cross validation.
- Within the code, keep track of the best parameter value.

VI: Model selection

Get a parameter value that lets the model perform well on **unseen data**.

- Use a development set or cross-validation for tuning (do **not** just get performance scores for the training set!)



VII: Challenge

⚠The features (= bigrams/columns) of the test set need to correspond to the features of the training set! (see ex. II)

VII: Challenge

What kind of feature engineering/choice of classifier/etc. proved to be more successful than our simple baseline logit model while tuning?