0_prepare_datatable

February 6, 2020

1 Prepare datatable used to benchmark the Balance Faces in the Wild (BFW) dataset.

Load table in `data/bfw-datatable.pkl` to extract all features and store in the datatable. Overwrites the table to `data/bfw-datatable.pkl`.

1.1 Add project code to PYTHONPATH, if not already there

Check that *path_package* is set to *code* directory on respective system

[1]:

```
%matplotlib inline
import pandas as pd
import pathlib
from sklearn.preprocessing import LabelEncoder
```

1.2 Load list of pairs

Load lists of pairs, with *p1* and *p2* representing samples for either pair, *label* is set as 1 if *genuine*; else, 0 for *imposter*, and the fold. Note there is no overlap in subjects between folds. Finally, the experiments are 5-fold (hence, *fold* $\in \{1, 2, 3, 4, 5\}$

[2]:

```
data = pd.read_csv('../../data/bfw-pairs.csv')
data.head()
```

[2]:

```
                                  p1                                  p2  \
0  asian_females/n000009/0010_01.jpg  asian_females/n000009/0043_01.jpg
1  asian_females/n000009/0010_01.jpg  asian_females/n000009/0120_01.jpg
2  asian_females/n000009/0010_01.jpg  asian_females/n000009/0122_02.jpg
3  asian_females/n000009/0010_01.jpg  asian_females/n000009/0188_01.jpg
4  asian_females/n000009/0010_01.jpg  asian_females/n000009/0205_01.jpg

   label  fold
0      1     1
1      1     1
2      1     1
3      1     1
4      1     1
```

1.3 Display stats

For each fold, how many *genuine* (i.e., 1) and *imposter* (i.e., 0)

[3]:
```
dfcounts = pd.DataFrame(data.groupby(by=['fold', 'label']).count()[['p1']])
dfcounts.columns = ['stats']
dfcounts.head(len(dfcounts))
```

[3]:
```
             stats
fold label
1    0      136194
     1       48514
2    0      136185
     1       48469
3    0      136284
     1       48527
4    0      136242
     1       48512
5    0      136474
     1       48497
```

[4]:
```
del dfcounts
```

1.4 Add metadata to table, set format appropriately each step

There is rich information in the pairs list, for which preparing the datatable will be convenient later

Set attributes, IDs (str), and abbreviated variants of attribute, gender, and ethnicity for both *p1* and *p2*

[5]:
```
data['att1'] = data.p1.apply(lambda x: x.split('/')[0]).astype('category')
data['att2'] = data.p2.apply(lambda x: x.split('/')[0]).astype('category')


data['e1'] = data.att1.apply(lambda x: x.split('_')[0][0].upper())
data['e2'] = data.att2.apply(lambda x: x.split('_')[0][0].upper())

data['g1'] = data.att1.apply(lambda x: x.split('_')[1][0].upper())
data['g2'] = data.att2.apply(lambda x: x.split('_')[1][0].upper())

data['a1'] = (data['e1'] + data['g1']).astype('category')
data['a2'] = (data['e2'] + data['g2']).astype('category')

data['e1'] = data['e1'].astype('category')
data['e2'] = data['e2'].astype('category')
data['g1'] = data['g1'].astype('category')
data['g2'] = data['g2'].astype('category')
data['score'] = pd.np.nan
data.head()
```

[5]:

```
                                  p1                                 p2  \
0  asian_females/n000009/0010_01.jpg  asian_females/n000009/0043_01.jpg   
1  asian_females/n000009/0010_01.jpg  asian_females/n000009/0120_01.jpg   
2  asian_females/n000009/0010_01.jpg  asian_females/n000009/0122_02.jpg   
3  asian_females/n000009/0010_01.jpg  asian_females/n000009/0188_01.jpg   
4  asian_females/n000009/0010_01.jpg  asian_females/n000009/0205_01.jpg   

   label  fold           att1           att2 e1 e2 g1 g2  a1  a2  score  
0      1     1  asian_females  asian_females  A  A  F  F  AF  AF    NaN  
1      1     1  asian_females  asian_females  A  A  F  F  AF  AF    NaN  
2      1     1  asian_females  asian_females  A  A  F  F  AF  AF    NaN  
3      1     1  asian_females  asian_females  A  A  F  F  AF  AF    NaN  
4      1     1  asian_females  asian_females  A  A  F  F  AF  AF    NaN  
```

1.4.1 assign unique ID tags per subject

Encode N subjects as 0, 1, …., N - 1

[6]:

```
le = LabelEncoder()



subject_names = list(set(["/".join(p1.split('/')[:-1]) for p1 in data['p1'].
 ↪unique()] + ["/".join(p2.split('/')[:-1]) for p2 in data['p2'].unique()]))
le.fit(subject_names)

data['ids1'] = le.transform(data['p1'].apply(lambda x: "/".join(x.split('/')[:
 ↪-1])))
data['ids2'] = le.transform(data['p2'].apply(lambda x: "/".join(x.split('/')[:
 ↪-1])))
data.head()
```

[6]:

```
                                  p1                                 p2  \
0  asian_females/n000009/0010_01.jpg  asian_females/n000009/0043_01.jpg   
1  asian_females/n000009/0010_01.jpg  asian_females/n000009/0120_01.jpg   
2  asian_females/n000009/0010_01.jpg  asian_females/n000009/0122_02.jpg   
3  asian_females/n000009/0010_01.jpg  asian_females/n000009/0188_01.jpg   
4  asian_females/n000009/0010_01.jpg  asian_females/n000009/0205_01.jpg   

   label  fold           att1           att2 e1 e2 g1 g2  a1  a2  score  ids1  \
0      1     1  asian_females  asian_females  A  A  F  F  AF  AF    NaN     0   
1      1     1  asian_females  asian_females  A  A  F  F  AF  AF    NaN     0   
2      1     1  asian_females  asian_females  A  A  F  F  AF  AF    NaN     0   
3      1     1  asian_females  asian_females  A  A  F  F  AF  AF    NaN     0   
4      1     1  asian_females  asian_females  A  A  F  F  AF  AF    NaN     0   

   ids2  
0     0  
```

```
1      0
2      0
3      0
4      0
```

[7]:
```
data.sample(10)
```

[7]:
```
                                        p1                                  p2  \
681566    black_males/n006208/0497_02.jpg  indian_females/n007046/0295_02.jpg
15927   asian_females/n006736/0058_01.jpg   asian_females/n006736/0001_01.jpg
476199  asian_females/n008401/0023_01.jpg     asian_males/n003156/0086_01.jpg
672831    black_males/n000950/0228_01.jpg     white_males/n001355/0042_01.jpg
866179  white_females/n008333/0257_02.jpg   asian_females/n006634/0078_01.jpg
590696  black_females/n003819/0479_01.jpg     white_males/n008954/0297_01.jpg
648859    black_males/n008908/0347_01.jpg     white_males/n009033/0003_01.jpg
21767   asian_females/n005216/0119_01.jpg   asian_females/n005216/0112_01.jpg
648285    black_males/n008730/0001_01.jpg     asian_males/n003635/0449_01.jpg
875115    white_males/n002878/0338_01.jpg     black_males/n000931/0063_01.jpg

        label  fold           att1            att2 e1 e2 g1 g2  a1  a2  score  \
681566      0     3    black_males  indian_females  B  I  M  F  BM  IF    NaN
15927       1     3  asian_females   asian_females  A  A  F  F  AF  AF    NaN
476199      0     2  asian_females     asian_males  A  A  F  M  AF  AM    NaN
672831      0     2    black_males     white_males  B  W  M  M  BM  WM    NaN
866179      0     5  white_females   asian_females  W  A  F  F  WF  AF    NaN
590696      0     2  black_females     white_males  B  W  F  M  BF  WM    NaN
648859      0     2    black_males     white_males  B  W  M  M  BM  WM    NaN
21767       1     4  asian_females   asian_females  A  A  F  F  AF  AF    NaN
648285      0     2    black_males     asian_males  B  A  M  M  BM  AM    NaN
875115      0     2    white_males     black_males  W  B  M  M  WM  BM    NaN

        ids1  ids2
681566   363   454
15927     65    65
476199    81   119
672831   310   719
866179   693    64
590696   272   790
648859   390   799
21767     46    46
648285   387   130
875115   732   309
```

1.5 Save datatable

if file does not exist, write to disc

[8]:

```
if not pathlib.Path('../../data/bfw-my-datatable.pkl').is_file():
    data.to_pickle('../../data/bfw-my-datatable.pkl')
```