CS289A_HW05_dataprocessing

March 31, 2017

0.1 CS 289A Homework 5 - Data Processing

This script will load the data sets in need of preprocessing (census and Titanic) and perform the preprocessing for better learning results.

Start with the overhead: load necessary modules and trigger them to reload when they are modified.

```
In [1]: %load_ext autoreload
```

```
In [2]: %autoreload 2
```

```
In [3]: import csv
        import numpy as np
        import pandas
        from sklearn.feature_extraction import DictVectorizer as DV
```

Next, specify paths to data on the local machine.

You must change the path to fit your data. Also note that the distributed census and Titanic training data/test csv files have been renamed.
(among other changes, the filename now includes the suffix _raw)

```
In [4]: BASE_DIR = "/Users/mitch/Documents/Cal/2_2017_Spring/COMPSCI 289A - Intro t

        CENS_RAWPATH = "Data/hw5_census_dist/census_traindata_raw.csv"
        CENS_TSTPATH = "Data/hw5_census_dist/census_testdata_raw.csv"
        CENS_CLNTRNPATH = "Data/census_traindata.csv"
        CENS_LBLTRNPATH = "Data/census_traindata_lbl.csv"
        CENS_VECTRNPATH = "Data/census_traindata_vec.csv"
        CENS_CLNTSTPATH = "Data/census_testdata.csv"
        CENS_VECTSTPATH = "Data/census_testdata_vec.csv"

        TITA_RAWPATH = "Data/hw5_titanic_dist/titanic_traindata_raw.csv"
        TITA_TSTPATH = "Data/hw5_titanic_dist/titanic_testdata_raw.csv"
        TITA_CLNTRNPATH = "Data/titanic_traindata.csv"
        TITA_LBLTRNPATH = "Data/titanic_traindata_lbl.csv"
        TITA_VECTRNPATH = "Data/titanic_traindata_vec.csv"
        TITA_CLNTSTPATH = "Data/titanic_testdata.csv"
        TITA_VECTSTPATH = "Data/titanic_testdata_vec.csv"
```

0.2 Impute and clean the datasets

First we will impute and clean the census data: (According to the census data README, it appears that the fnlwgt feature denotes similarity of individuals in a state. The census data is not necessarily state separated, so remove that feature.)

```
In [5]: censusdf_rawtrain = pandas.read_csv(open(BASE_DIR+CENS_RAWPATH))
        censusdf_rawtest = pandas.read_csv(open(BASE_DIR+CENS_TSTPATH))

        censusdf_nantrain = censusdf_rawtrain.replace(to_replace='?',value=np.nan)
        censusdf_nantest = censusdf_rawtest.replace(to_replace='?',value=np.nan)

        censusdf_train = censusdf_nantrain.fillna(censusdf_nantrain.mode().iloc[0])
        censusdf_test = censusdf_nantest.fillna(censusdf_nantest.mode().iloc[0])

        censusdf_train.drop('fnlwgt',axis=1,inplace=True)
        censusdf_test.drop('fnlwgt',axis=1,inplace=True)

        censusdf_train.to_csv(BASE_DIR+CENS_CLNTRNPATH,index=False)
        censusdf_test.to_csv(BASE_DIR+CENS_CLNTSTPATH,index=False)
```

Since the vast majority of the unknown values, denoted with a '?', are categorical rather than continuous datapoints, replace them with the most common category-value in that feature.

Next, repeat the imputation and cleaning for the Titanic dataset: (Recognizing that the cabin feature vector is incredibly sparse–and presumably meaningless–eliminate it from the data set to be processed; similarly, due to the large variation in data types in the ticket column, remove it as well)

```
In [6]: titanicdf_rawtrain = pandas.read_csv(open(BASE_DIR+TITA_RAWPATH))
        titanicdf_rawtest = pandas.read_csv(open(BASE_DIR+TITA_TSTPATH))

        titanicdf_nantrain = titanicdf_rawtrain.replace(to_replace='',value=np.nan)
        titanicdf_nantest = titanicdf_rawtest.replace(to_replace='',value=np.nan)

        titanicdf_train = titanicdf_nantrain.fillna(titanicdf_nantrain.mode().iloc[
        titanicdf_test = titanicdf_nantest.fillna(titanicdf_nantest.mode().iloc[0])


        titanicdf_train.drop('cabin',axis=1,inplace=True)
        titanicdf_train.drop('ticket',axis=1,inplace=True)
        titanicdf_test.drop('cabin',axis=1,inplace=True)
        titanicdf_test.drop('ticket',axis=1,inplace=True)

        titanicdf_train.to_csv(BASE_DIR+'/'+TITA_CLNTRNPATH,index=False)
        titanicdf_test.to_csv(BASE_DIR+'/'+TITA_CLNTSTPATH,index=False)
```

0.3 Vectorize the cleaned and full data

First, separate the labels from the data, and save to a csv file.

```
In [7]: censuslbldf = censusdf_train['label']
        censuslbldf.to_csv(BASE_DIR+'/'+CENS_LBLTRNPATH,index=False)
        censusdf_train.drop('label',axis=1,inplace=True)

        titaniclbldf = titanicdf_train['survived']
        titaniclbldf.to_csv(BASE_DIR+'/'+TITA_LBLTRNPATH,index=False)
        titanicdf_train.drop('survived',axis=1,inplace=True)
```

Use the DictVectorizer class from sklearn to create vectors for categorical mappings

```
In [8]: #For the census dataset
        censusdict_train = censusdf_train.to_dict('records')
        censusdict_test = censusdf_test.to_dict('records')

        dv = DV(sparse=False)

        censusvec_train = dv.fit_transform(censusdict_train)
        censusvec_test = dv.fit_transform(censusdict_test)

        np.savetxt(BASE_DIR+CENS_VECTRNPATH,censusvec_train,fmt='%10d',delimiter=',
        np.savetxt(BASE_DIR+CENS_VECTSTPATH,censusvec_test,fmt='%10d',delimiter=','

In [9]: #For the Titanic dataset
        titanicdict_train = titanicdf_train.to_dict('records')
        titanicdict_test = titanicdf_test.to_dict('records')

        dv = DV(sparse=False)

        titanicvec_train = dv.fit_transform(titanicdict_train)
        titanicvec_test = dv.fit_transform(titanicdict_test)

        np.savetxt(BASE_DIR+TITA_VECTRNPATH,titanicvec_train,fmt='%10d',delimiter='
        np.savetxt(BASE_DIR+TITA_VECTSTPATH,titanicvec_test,fmt='%10d',delimiter=',
```