Feature Engineering

Papangkorn Inkeaw, Ph.D.

Feature Selection

Lab 9

Load dataset

Dataset: Hand Sign Images Dataset (https://www.kaggle.com/datasets/ash2703/handsignimages)

The data set includes 27,455 gray-scale images of size 28*28 pixels.

- Import libraries

```
import cv2
import numpy as np
from skimage import feature
import glob
```

- List all JPEG files in all subfolder in the corpus

```
filenames = []
y_train = []  #class labels list
for dirName in glob.glob("Train/*/"):                    #List all subfolders in the folder Train
     tmp = dirName.split("/")
     class_name = tmp[-1]
     for imgFile in glob.glob(dirName+"*.jpg"):
               filenames.append(imgFile)
               y.append(class_name ). #append class name to list y_train
y_train = np.array(y_train)
```

Extract Features

- Retrieve each image and process it

```
x_train = np.empty((0,feature_len), dtype=float)
for imgFile in filenames:
    img = cv2.imread(imgFile)
    # extract feature vector here
    feature_vector = feature.hog(img, orientations=9, pixels_per_cell=(8, 8),
            cells_per_block=(2, 2), block_norm="L1")

    # append the vector to x_train
    x_train = np.append(x_train, feature_vector, axis=0)

print(x_train.shape)
print(y_train.shape)
```

Feature Selection using Filter Methods

```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif
from sklearn.feature_selection import mutual_info_classif
```

- Perform ANOVA F-value to the samples to retrieve only the 50 best features

```
sel_anova = SelectKBest(f_classif, k=50)
anova_selector = sel_anova.fit(x_train, y_train)
x_selected_f = anova_selector.transform(x_train)
print(x_selected_f.shape)
```

- Perform ANOVA F-value test to the samples and select features based on p-value

```
f_statistic, p_value = f_classif(x_train, y_train)
x_selected_f2 = x_train[:, p_value<=0.05]
print(x_selected_f2.shape)
```

Feature Selection using Filter Methods

- Estimate mutual information to retrieve only the 50 best features

```
sel_mut = SelectKBest(mutual_info_classif, k=50)
mut_selector = sel_mut.fit(x_train, y_train)
x_selected_mut = mut_selector.transform(x_train)
print(x_selected_mut.shape)
```

Feature Selection using Wrapper Methods

```
from sklearn.feature_selection import SequentialFeatureSelector
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neighbors import RFE
from sklearn.svm import SVC
```

- Using kNN as a classifier, perform a sequential feature selection method to select feature

```
knn = KNeighborsClassifier(n_neighbors=5)
sfs = SequentialFeatureSelector(knn, n_features_to_select="auto", tol=0.05, direction='forward')
sfs.fit(x_train, y_train)
sfs.get_support()
x_selected_fsfs = sfs.transform(x_train)
print(x_selected_fsfs.shape)
```

Feature Selection using Wrapper Methods

- Ranking features using SVM and perform a recursive feature elimination method to select feature

```
estimator = SVC(kernel=”rbf")
rfes = RFE(estimator, n_features_to_select=50, step=1)
rfes = rfes.fit(x_train, y_train)
rfes.support_
x_selected_rfes = selector.transform(x_trian)
print(x_selected_rfes.shape)
```

Your work!

1. Use features you extracted from Lab 6, 7 or 8
2. Split the dataset into training and test dataset (Hint: use *train_test_split* method in sklearn library)
3. Perform one feature selection method on the features using the training dataset.
4. With using the selected features, construct a classifier
5. Evaluate the performance of the classifier on the test set
6. Submit your program to the assignment submission system (http://hw.cs.science.cmu.ac.th/).

Note:

- Put your name and student ID in the first cell using comment tag.
- Name your python notebook file with the pattern Lab_09_XXXXXXXXX.py (XXXXXXXXX is your student ID)

References & Study Resources

- https://www.kaggle.com/datasets/ash2703/handsignimages
- https://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_selection