**CS670/470 Team Project Phase 2: Online Feature Selection**

**Report**

Team Members: Vikram Venkatesh Gopalan, Emmanuel Saake, Sunil Srinivas, Hamza Shahid

**Objective**

Our goal for this phase was to implement the Scalable and Accurate OnLine Approach (SAOLA) to learn how to apply online learning to feature selection.

**Input**

**Loading the data:**

The source code for the SAOLA algorithm is as follows:

1. We first load the target data and the feature data as shown below.
2. For the feature set we select the last 4 days of data.
3. We convert the dtype of both the target and the feature data into float.
4. We save each part of the feature data into separate csv files.

CODE:

*import numpy as np*

*import pandas as pd*

*import glob*

*import re*

*target = np.load('G:/Phase2Data/target/target\_1980\_2010.npy')*

*label = pd.DataFrame(target).astype(float)*

*pathName = 'G:/Phase2Data/\*.npy'*

*def getDataFrames(pathName):*

*for npyFile in glob.glob(pathName):*

*print npyFile*

*pathSplit = re.split('data\_D\_1980\_2010\_|.npy',npyFile)*

*data = np.load(npyFile)*

*for i in range(6,10):*

*path = 'G:\Phase2Parts\\' + pathSplit[1] + 'day'+ str(i+1) + '.csv'*

*print path*

*dataPart = data[i:11300+i:]*

*np.savetxt(path, dataPart.astype(np.float), delimiter=",")*

*getDataFrames(pathName)*

**SAOLA**

The SAOLA algorithm performs pairwise comparisons to online calculate the correlations between features. For pairwise comparisons, we make use of the measure of mutual information to calculate correlations between features.

As shown in the code below, we perform correlation, then do the z- transform and finally get the z-score.

In SAOLA Method 1, we perform the test of the correlation between features and removal of irrelevant features using the threshold value i.e. (1) if *Zy\_c > Zf\_c and Zf\_y >=1.96; (2) if Zf\_c > Zy\_c and Zf\_y >=1.96*

In SAOLA Method 2, we change the above tests and use the correlation between each feature in the minimum set and the class label i.e. (1) *if Zy\_c > Zf\_c and Zf\_y >=Zf\_c ; (2) if Zf\_c > Zy\_c and Zf\_y >=Zy\_c:*

In both methods, the function SAOLA performs the SAOLA algorithm. It performs the correlation between the new feature and the class label as shown by corrC in the code. We perform the z transformation and get the z score. If the threshold condition is not satisfied, it will run the correlationF function where we perform pairwise comparison of features and check for feature redundancy.

**SAOLA Method 1:**

*def SAOLA(data, label):*

*dfList = pd.DataFrame()*

*for csvFile in glob.glob(data):*

*df = pd.read\_csv(csvFile,header=None)*

*for i in df:*

*corrC = df.iloc[:,i].corr(label.iloc[:,0])*

*zTrans = np.arctanh(corrC)*

*zScore = zTrans\*np.sqrt(11300 - 3)*

*if zScore < 1.96:*

*continue*

*else:*

*df3 = df.iloc[:,i]*

*if dfList.empty:*

*dfList =pd.concat([dfList, df3], axis=1,ignore\_index = True)*

*else:*

*df3, dfList = correlationF(df3,dfList)*

*return dfList*

*def correlationF(df3,dfList):*

*label = pd.DataFrame(target)*

*i = 0*

*while i < len(dfList.columns):*

*corr\_yc = dfList.iloc[:,i].corr(label.iloc[:,0])*

*xTrans\_yc=np.arctanh(corr\_yc)*

*Zy\_c= xTrans\_yc\*np.sqrt(11300-3)*

*corr\_fc = df3.corr(label.iloc[:,0])*

*xTrans\_fc=np.arctanh(corr\_fc)*

*Zf\_c= xTrans\_fc\*np.sqrt(11300-3)*

*corr\_fy = dfList.iloc[:,i].corr(df3)*

*xTrans\_fy=np.arctanh(corr\_fy)*

*Zf\_y= xTrans\_fy\*np.sqrt(11300-3)*

*if Zy\_c > Zf\_c and Zf\_y >=1.96:*

*return df3,dfList*

*if Zf\_c > Zy\_c and Zf\_y >=1.96:*

*dfList = dfList.drop(dfList.columns[i], axis=1)*

*i=0*

*else:*

*i= i+1*

*dfList =pd.concat([dfList, df3], axis=1,ignore\_index=True)*

*return df3,dfList*

**SAOLA Method 2**

*def SAOLA(data,label):*

*dfList = pd.DataFrame()*

*for csvFile in glob.glob(data):*

*df = pd.read\_csv(csvFile,header=None)*

*for i in df:*

*corrC = df.iloc[:,i].corr(label.iloc[:,0])*

*zTrans = np.arctanh(corrC)*

*zScore = zTrans\*np.sqrt(11300 - 3)*

*if zScore < 1.96:*

*continue*

*else:*

*df3 = df.iloc[:,i]*

*if dfList.empty:*

*dfList =pd.concat([dfList, df3], axis=1,ignore\_index = True)*

*else:*

*df3, dfList = correlationF(df3,dfList)*

*return dfList*

*def correlationF(df3,dfList):*

*label = pd.DataFrame(target)*

*i = 0*

*while i < len(dfList.columns):*

*corr\_yc = dfList.iloc[:,i].corr(label.iloc[:,0])*

*xTrans\_yc=np.arctanh(corr\_yc)*

*Zy\_c= xTrans\_yc\*np.sqrt(11300-3)*

*corr\_fc = df3.corr(label.iloc[:,0])*

*xTrans\_fc=np.arctanh(corr\_fc)*

*Zf\_c= xTrans\_fc\*np.sqrt(11300-3)*

*corr\_fy = dfList.iloc[:,i].corr(df3)*

*xTrans\_fy=np.arctanh(corr\_fy)*

*Zf\_y= xTrans\_fy\*np.sqrt(11300-3)*

*if Zy\_c > Zf\_c and Zf\_y >=Zf\_c:*

*return df3,dfList*

*if Zf\_c > Zy\_c and Zf\_y >=Zy\_c:*

*dfList = dfList.drop(dfList.columns[i], axis=1)*

*i=0*

*else:*

*i= i+1*

*dfList =pd.concat([dfList, df3], axis=1,ignore\_index=True)*

*return df3,dfList*

We run the SAOLA algorithms by running it on the data as shown below:

*dfList = pd.DataFrame()*

*data = "G:/Phase2Parts/\*.csv"*

*dfList = SAOLA(data,label)*

We also represent the features we obtain on a map:

plt.figure(figsize=(12,6))

map = Basemap(projection='robin',lon\_0=360)

map.drawcoastlines()

x, y = map([340,80],[5,27.5])

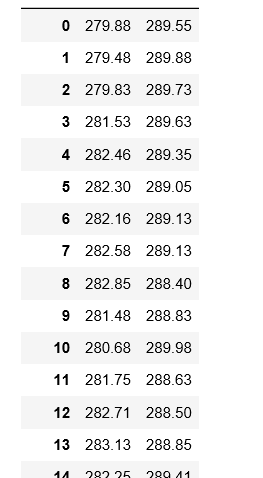
map.scatter(x, y,color = 'r',marker = '+', s = 100)

plt.show()

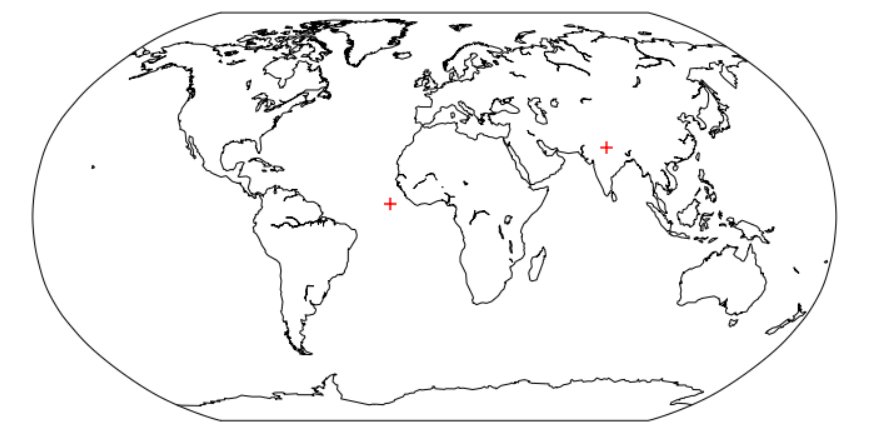
plt.clf()

**Result and Observations**

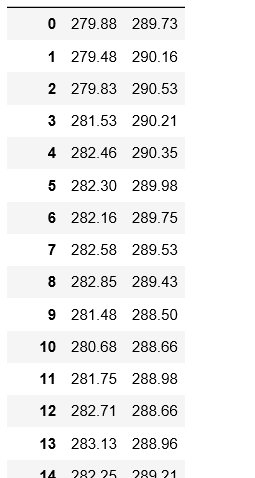
Using SAOLA Method 1, we get 2 features as output: T850 for location 1210 and T850 for location 5067. The data comes from the last day of the 4 days dataset. The sample output is as shown.



The above figure is a part of the output we get for SAOLA Method 1. We get 2 features with 11300 samples. The below figure is the representation of the data on the map. The red crosses depict the 2 features.



Using SAOLA Method 2, we get 2 features as output: T850 for location 1210 and T850 for location 4180. The data comes from the last day of the 4 days dataset. The sample output is as shown.



The above figure is a part of the output we get for SAOLA Method 2. We get 2 features with 11300 samples. The below figure is the representation of the data on the map. The red crosses depict the 2 features.

