

Project report 8

Incremental Clustering: Intrusion Detection by Visual Surveillance

Name :DHIVAKAR.R

COURSE: AI and ML(Aug 2020)

Question: In today's world, the data is dynamic and hence, it is not always feasible to use Non - incremental clustering techniques which rely on the complete dataset for forming the clusters. Thus, we need an incremental clustering algorithm that automatically adapts to itself as the data points increase. Implement a basic incremental K Means algorithm using the iris dataset (available in scikitlearn module or can download the csv file). Once done, try using the same algorithm for intrusion detection using any video of your choice (Only one such video is required as our algorithm will learn as the new frames are introduced automatically)

Prerequisites

What things you need to install the software and how to install them:

Python 3.6 This setup requires that your machine has latest version of python. The following url <https://www.python.org/downloads/> can be referred to download python. Once you have python downloaded and installed, you will need to setup PATH variables (if you want to run python program directly, detail instructions are below in how to run software section). To do that check this: <https://www.pythoncentral.io/add-python-to-path-python-is-not-recognized-as-an-internal-or-externalcommand/> . Setting up PATH variable is optional as you can also run program without it and more instruction are given below on this topic.

Second and easier option is to download anaconda and use its anaconda prompt to run the commands. To install anaconda check this url <https://www.anaconda.com/download/> You will also need to download and install below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.6 then run below commands in command prompt/terminal to install these packages pip install -U scikit-learn pip install numpy pip install scipy if you have chosen to install anaconda then run below commands in anaconda prompt to install these packages conda install -c scikit-learn conda install -c anaconda numpy conda install -c anaconda scipy

Dataset used: The dataset used is IRIS dataset which is an in-built dataset available in scikit-learn library

Importing the requires libraries

```
Launcher X Incremental Clustering Intru X AIMLProject8IncrementalCl X
Code v

[28]: from sklearn.cluster import MiniBatchKMeans
      from sklearn.datasets import load_iris
      import numpy as np
      import cv2
      import pandas as pd
      from sklearn.datasets import load_iris
```

Testing on Iris Dataset

```
Testing on Iris Dataset

[30]: data = load_iris()
      iris = load_iris()
      df = pd.DataFrame(data=np.c_[iris['data']],
                        columns=iris['feature_names'])

[31]: df

[31]:   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
0              5.1             3.5             1.4             0.2
1              4.9             3.0             1.4             0.2
2              4.7             3.2             1.3             0.2
3              4.6             3.1             1.5             0.2
4              5.0             3.6             1.4             0.2
...           ...             ...             ...             ...
145             6.7             3.0             5.2             2.3
146             6.3             2.5             5.0             1.9
147             6.5             3.0             5.2             2.0
148             6.2             3.4             5.4             2.3
149             5.9             3.0             5.1             1.8

150 rows x 4 columns

[32]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 4 columns):
sepal length (cm)    150 non-null float64
sepal width (cm)     150 non-null float64
petal length (cm)    150 non-null float64
petal width (cm)     150 non-null float64
dtypes: float64(4)
memory usage: 4.8 KB
```

Getting the output

```
[8]: y_pred = minibatchKmeans.predict(X)

[9]: (1-(np.count_nonzero(y_pred == y) / len(y)))

[9]: 0.9133333333333333

[18]: font = cv2.FONT_HERSHEY_SIMPLEX
cam = cv2.VideoCapture(0)
cam.set(3, 640)
cam.set(4, 480)
image_list = []
count = 0

[19]: while(True):
    ret, img = cam.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    if count >= 1:
        temp = res_list
        minibatchKmeans = MiniBatchKMeans(n_clusters=3, random_state=0)
        res_list = minibatchKmeans.fit(gray)
        if count < 1:
            temp = res_list
        count+=1
        pred = temp.predict(gray)

        if False in (temp.labels_ == pred):
            cv2.putText(img, "Intrusion Detected ", (320, 240), font, 1, (255,255,255), 2)
        else:
            cv2.putText(img, "no one around ", (320, 240), font, 1, (255,255,255), 2)

        cv2.imshow('image', img)
        k = cv2.waitKey(100) & 0xff
        if k == 27:
            break

[21]: cam.release()
cv2.destroyAllWindows()
```

otuput



