Principal Component Analysis (PCA)

1分

Informally, if you have a bunch of scattered data, PCA will let you determine a vector for the direction of greatest spread and the direction of least spread. These directions can be found by finding the eigenvectors and eigenvalues of the covariance matrix for the data. In this problem, you are given a scattered 2D point set with the x coordinates stored in `x` and the y coordinates in `y`.

In 2D, the sample covariance matrix for a data set of n points (x_i, y_i) can be formed in the following manner: $\frac{1}{n-1}\begin{bmatrix} (x_i-\mu_x)^2 & (x_i-\mu_x)(y_i-\mu_y) \\ (y_i-\mu_y)(x_i-\mu_x) & (y_i-\mu_y)^2 \end{bmatrix}$ where μ_x is the average of the x coordinates and similarly μ_y for y.

The sample covariance matrix is symmetric, so it will have n orthogonal eigenvectors. The dominant eigenvector (the one associated with the eigenvalue with the greatest absolute value) will be the principal direction of the data set and the eigenvector asscoiated with the eigenvalue of least magnitude will be the direction of least spread. These eigenvectors form the principal directions of the data.

One can compute an object-aligned bounding box by projecting the data points onto the normalized versions of the two principal eigenvectors. Specifically, use the centroid as the origin of the box with the principal eigenvectors forming the local coordinate system. Project the vectors from the centroid to the scattered points onto the normalized eigenvectors and keep track of the maximum and minimum extents. Use those extents to find the corners of the bounding box. Put the corners of the bounding box into a numpy array named `corners`. The order of the corners is important (purely for grading reasons). If e_1 is the dominant eigenvector and e_2 the other eigenvector, order the points so that the first point is: the minimum in directions e_1 and e_2, the second is the minimum e_1 and maximum e_2, the thir is maximum e_1 and maximum e_2, and the last is maximum e_1 and minimum e_2.

Hints:

- The eigenvectors can be extracted using the Python function `numpy.linalg.eig`.

INPUTS: `x` and `y`

OUTPUT: `corners`

评分代码 (点击查看)

起始代码 (点击查看)

回答*

```
import numpy as np
import numpy.linalg as la
import matplotlib.pyplot as pt
import matplotlib.lines as lines

# Compute the centroid of the points
xm = x.mean()
ym = y.mean()

# Create the covariance matrix
cv_mat = np.zeros((2,2))
for i in range(n):
     cv_mat[0][0]+=((x[i]-xm)**2)
     cv_mat[1][1]+=((y[i]-ym)**2)
     cv_mat[0][1]+=((y[i]-ym)*(x[i]-xm))
```

按F9以打开/关闭全屏模式. 在 用户信息 (/profile/) 中设置编辑器模式.

保存回答 | 提交用于评分的回答

(您仍然可以在提交**本问题**后修改回答)