*### 代码基于python 3.7*

*## 一些函数*

*# 计算协方差矩阵的特征值和特征向量*

*def eig\_pca(data, method='cov'):*

*if method == 'corr':*

*matrix = np.corrcoef(data.T)*

*else:*

*matrix = np.cov(data.T)*

*lam, v = np.linalg.eig(matrix)*

*# 对特征值排序*

*sorted\_indices = np.argsort(-lam)*

*lam = lam[sorted\_indices]*

*v = v[:, sorted\_indices]*

*return lam, v*

*# 拐点图*

*import matplotlib.pyplot as plt*

*def screeplot(subloc, variance, xlabels, title):*

*plt.subplot(subloc)*

*plt.plot(variance, marker='o', markerfacecolor='w')*

*plt.title(title)*

*plt.xticks(range(3), labels=xlabels)*

*plt.ylabel('Variance')*

*# 配对散点图*

*import seaborn as sns*

*def pairplot(data, diag\_kind='kde', hue=None):*

*sns.pairplot(data, diag\_kind='kde', hue=hue)*

7.1

代码：

*# 生成数据*

*import numpy as np*

*np.random.seed(42)*

*data = np.random.multivariate\_normal(*

*[0, 0, 0], [[1,1,1],[1,4,1],[1,1,100]], 100)*

*# 分别计算特征值和特征向量*

*lam1, v1 = eig\_pca(data)*

*lam2, v2 = eig\_pca(data, 'corr')*

*print(lam1, '\n\n', v1, '\n\n', lam2, '\n\n', v2)*

*[68.06636876 4.11711331 0.82720024]*

*[[ 7.43724371e-04 3.57753850e-01 -9.33815629e-01]*

*[ 2.65786806e-02 9.33478921e-01 3.57646022e-01]*

*[ 9.99646448e-01 -2.50855774e-02 -8.81437429e-03]]*

*[1.52060733 0.99910462 0.48028805]*

*[[-0.69166945 -0.20599909 -0.69221221]*

*[-0.70679703 -0.00390719 0.7074056 ]*

*[-0.14842952 0.97854438 -0.14289706]]*

*# 计算 PC-scores*

*import pandas as pd*

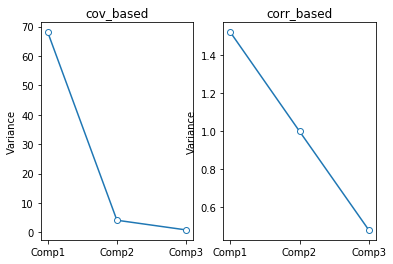
*data\_pca1 = pd.DataFrame(data.dot(v1), columns=['Comp1', 'Comp2', 'Comp3'])*

*data\_pca2 = pd.DataFrame(data.dot(v2), columns=['Comp1', 'Comp2', 'Comp3'])*

*# 绘制screeplot*

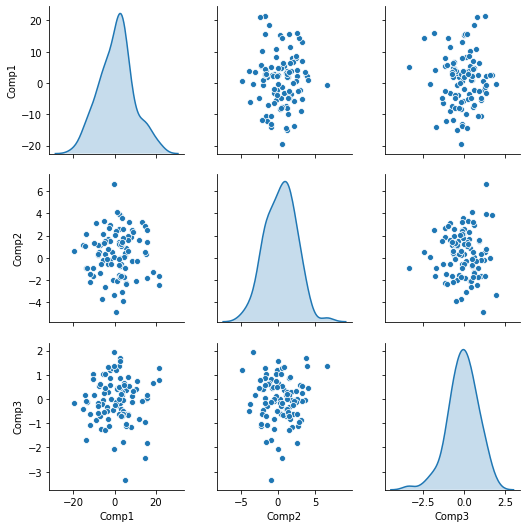
*screeplot(121, lam1, data\_pca1.columns, "cov\_based")*

*screeplot(122, lam2, data\_pca1.columns, "corr\_based")*

**

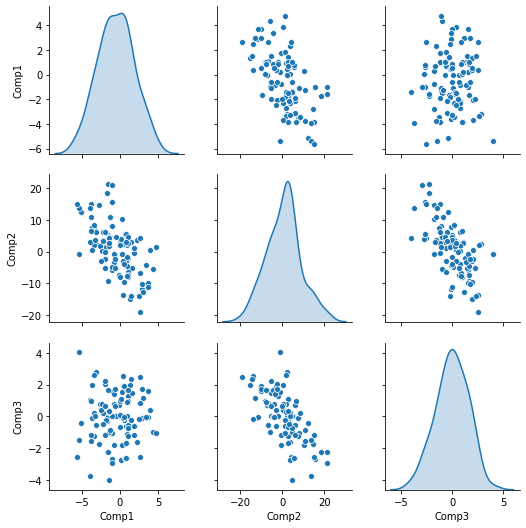
*# cov-based主成分的配对散点图*

*pairplot(data\_pca1)*

**

*­# corr-based主成分的配对散点图*

*pairplot(data\_pca2)*

**

7.4

代码：

*# 读取数据*

*file\_path = r"C:\Users\Mac\Desktop\LDR\pendigits.txt"*

*digit = pd.read\_table(file\_path, header=0, sep=' ').values*

*X = digit[:, :16]*

*# 计算变量方差*

*feature\_var = np.var(X, axis=0)*

*print(feature\_var)*

*[1173.58957462 263.02188859 693.93596968 367.24464253 1162.84033874*

*728.74685591 934.88778575 894.19485469 1164.94010702 742.63647409*

*1390.39947711 735.4633654 498.69223553 1099.00884636 1743.92616188*

*1279.63681919]*

*# 计算特征值和特征向量*

*lam, v = eig\_pca(X)*

*# 选取合适的主成分数目*

*var\_cumratio = (lam / lam.sum()).cumsum()*

*print("取前" + str(np.argmin(var\_cumratio < 0.8) + 1) + "个成分可以达到80%的方差贡献")*

*取前5个成分可以达到80%的方差贡献*

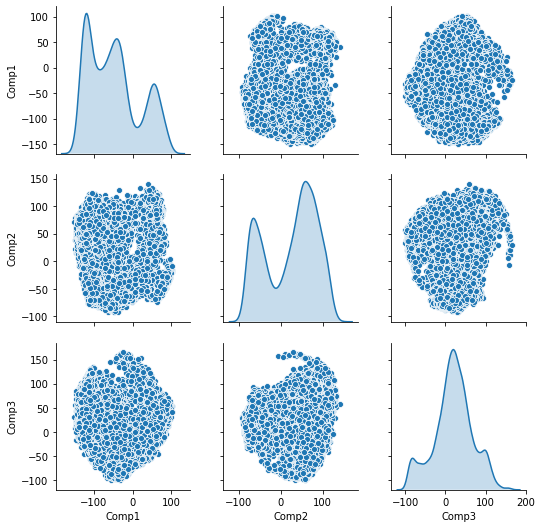
*print("取前" + str(np.argmin(var\_cumratio < 0.9) + 1) + "个成分可以达到90%的方差贡献")*

*取前7个成分可以达到90%的方差贡献*

*# 主成分的配对散点图*

*X\_pca = pd.DataFrame(X.dot(v), columns=['Comp'+str(i+1) for i in range(16)])*

*pairplot(X\_pca.iloc[:, :3])*

**

7.9

代码：

*# 读取数据（这个数据有150个样本）*

*iris = sns.load\_dataset("iris")*

*X = iris.drop(columns='species').values*

*# 计算特征值和特征向量*

*lam, v = eig\_pca(X)*

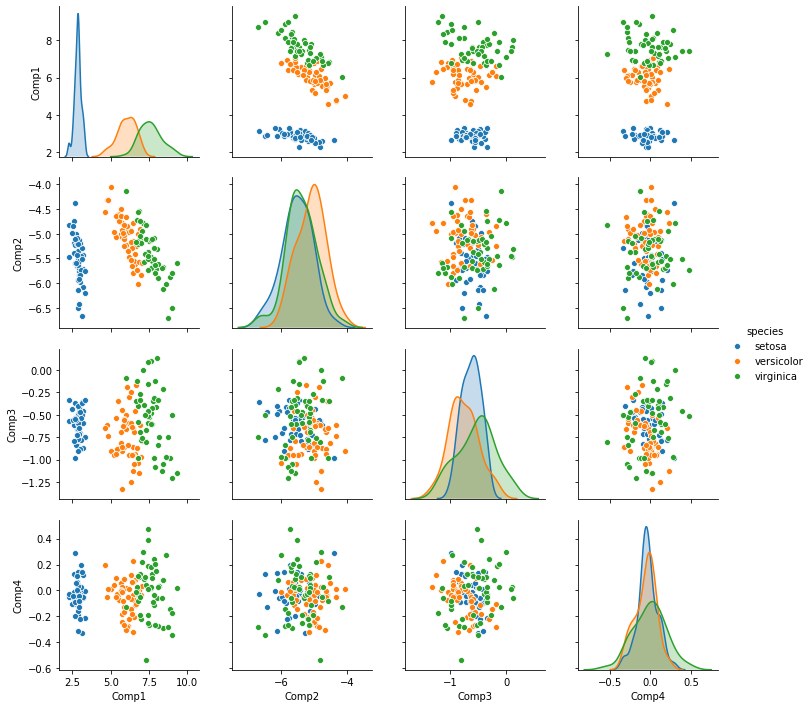
*# 计算 PC-scores*

*data = pd.DataFrame(X.dot(v), columns=['Comp'+str(i+1) for i in range(4)])*

*# 主成分的配对散点图*

*data = pd.concat([data, iris['species']], axis=1)*

*pairplot(data, hue='species')*

**

7.12

