Dimensionality reduction on the Iris dataset

In previous work, we had shown that the iris species; Setosa, Versicolor and Virginica were predictable and classifiable based on the attributes; Sepal length, sepal width, petal length and petal width. However, we have also identified that some of the dimensions in our data don’t provide meaningful information regarding the classification. In this report I’ll summarise and explain a strategy to dimensional reduction which aims to isolate the features that explain a threshold of the total energy within our data and remap our data to these features.

If we think of the attributes of the iris dataset as a matrix, we can construct an matrix of observations of attributes. We can construct a singular value decomposition representation of this matrix such that:

Where is a by unitary rotational matrix, is a diagonal amplitude vector of the eigenvalues (, in this case) of and is the final by unitary rotational matrix. For those unaware of eigenvalue decompositions see appendix 1. By removing elements from the and respectively we can construct a representation in which most of the total energy within our data is conserved but the total dimensionality of our dataset is reduced. We aim to reduce the total absolute energy of our adjusted eigenvalue vector, , to 90% of that of the eigenvalue vector . Similarly, we set the eigenvectors corresponding to the removed eigenvalues to 0. With these adjusted components and we reconstruct our original matrix as where:

For the Iris dataset as previously explored, we can produce an adjusted matrix wherein 90% of the energy of the data is conserved and only two dimensions remain as shown below:

![A close up of a map

Description automatically generated]() ![A screenshot of a cell phone

Description automatically generated]()

Fig 1a. Scatterplot of sepal width against petal length from the iris dataset 1b. scatterplot of the transformed data given feature 1 v feature 2.

Whilst transforming the data space is useful in improving the distinguishability between data classes, it becomes difficult to relate the meaning of success to the real world. Feature 1 is not a single attribute and understanding why a segmentation is successful is obfuscated by the manipulation of the data. Hence it is important to understand when and why to implement such a strategy.

**Code:**

﻿import numpy as np

from scipy.linalg import svd

from sklearn.datasets import load\_iris

import matplotlib.pyplot as plt

data, classifier = load\_iris(return\_X\_y=True)

U, S, V = svd(data)

S[2::]=0; #Adjust this to be 90% of the variance

V[:,2:3]=0;

S\_ = np.zeros((data.shape[0], data.shape[1]))

S\_[:data.shape[1], :data.shape[1]] = np.diag(S)

recon\_data = U.dot(S\_.dot(V))

np.delete(recon\_data,3,1)

np.delete(recon\_data,2,1)

species = ['Setosa','Versicolor','Virginica']

for i in range(0,3):

plt.scatter(recon\_data[np.where(classifier==i),1],recon\_data[np.where(classifier==i),0], label=species[i])

plt.legend()

plt.xlabel('Feature 1 ')

plt.ylabel('Featur 2')

plt.title('Reduction of the Iris dataset')

plt.show()

**Code 2 Eigvenvalues and Eigenvectors:**

﻿import matplotlib.pyplot as plt

import numpy as np

A = np.array([[2, 1], [1, 2]])

Ex,Ev=np.linalg.eig(A)

for i in range(0,2):

plt.polar([0,np.sin(Ev[1,i]/Ev[0,i])], [0,Ex[i]],marker='\*')

plt.title('polar plot of the eigenvectors and their magnitudes')

**Appendix 1. Eigenvalues and Eigenvectors:**

Given a matrix, , we can construct a representation such that , where are the eigenvalues of matrix and are its eigenvectors. Through rearrangement;

If we solve for and we find that and . Now we can compare out representations as shown below;

![A screenshot of a cell phone

Description automatically generated]()![A close up of a map

Description automatically generated]()

Fig Ap1. Scatterplot of the observations in matrix A. Fig Ap2.Polar plot of the eigenvectors and eigenvalues of A.

From Fig Ap2 we can see how manipulating this vector representation can change the dimensionality of our dataset. If we were to delete the orange vector, we would construct a 1D representation of our dataset wherein 75% of our amplitude is conserved.