Machine Learning	Prof. Matthias Hein
Exercise Sheet 5 - 3.6.2021 (12 Points)	due: 10.6.2021

Exercise 9 - Dimensionality reduction with LDA and VC dimension

a. **(3 points)** Write a function `[Z, W] = LDAFeature(X, Y)` which has the inputs

- design matrix `X` $\in \mathbb{R}^{n \times d}$, ($n$: number of points, d: dimension of input space),
- label vector `Y` (column vector of size n)

and outputs (given that we have K classes)

- the d-dimensional generalized eigenvectors w^s, $s = 1, ..., K-1$ of multiclass LDA in matrix form $W = (w^1, w^2, ..., w^{K-1})$ (thus $W \in \mathbb{R}^{d \times (K-1)}$),
- the coordinates `Z` (as an $n \times (K-1)$ matrix) of the points in `X` projected on the generalized eigenvectors $w^s (s = 1, ..., K-1)$,

$$z_{is} = \sum_{r=1}^{d} X_{ir} w_r^s, \qquad\qquad i = 1, ..., n \quad s = 1, ..., K-1 \, .$$

b. **(3 points)** In statistical learning theory, the **VC dimension** of a classifier is the largest cardinality n of a set of points such that the classifier can achieve any of the 2^n possible binary labelings. Prove the following statement: Any linear classifier in d dimensions has VC dimension $d + 1$.

c. **(4 points)** We have three classes, each having a normal distribution with zero mean and unit covariance.

- For different dimensions of the feature space, $d = 5, 30, 60, 75$, sample 25 points from each class, `X=numpy.random.randn(25,d)` and generate the embedding with LDA.
- Plot the result for each dimension d in one figure. Note that the coordinates of each data point `i` are given by `Z[i,0], Z[i,1]`, each class should have a corresponding color. Given how the three classes are distributed, what do you expect as a result of LDA? What do you observe in your experiment? What could be the reason for the different behavior of the result depending on the number of dimensions? (Think about the result from b.)

d. **(2 points)** Apply dimensionality reduction to the handwritten digits 3, 8, 9 in the USPS dataset. Load the file `digits389.npy`, which contains the training and test data, that is images (each digit is an image of 16×16, so we have 256 gray values) and the corresponding labels 3, 8, 9. Apply `[Z,W]=LDAFeature(X,Y)` and plot the data as in the previous part. Test the obtained embedding by embedding the test data using the learned projection, `Z=X*W`, and plot the obtained embedding in a separate figure. How do you judge this result ? Does the embedding generalize (written answer on paper) ?

Hints:

- Use `D, V = scipy.linalg.eig(A, B)` to solve the generalized eigenproblem $Av = \lambda Bv$. `D` contains the eigenvalues and the columns of `V` the corresponding eigenvectors. To find the largest eigenvalues you have to sort them, and correspondingly the eigenvectors (you might use the function `numpy.argsort()`).

- In part b, you might prove first that there exists a set of $d+1$ points for which every labeling can be learnt by a linear classifier, second that any set of $d+2$ points cannot be learnt. For this it might be helpful that a linear model $f_{w,b}(x) = \langle w, x \rangle + b$ can learn data $(X_i, Y_i)_{i=1}^n$, with $Y_i \in \{-1, 1\}$, if there exist w, b such that

$$f_{w,b}(X_i) = \operatorname{sign}(w^T X_i + b) = Y_i, \qquad \forall i = 1, ..., n,$$

which can be written as a linear system.