Introduction to Machine Learning
Work 2
Principal Component Analysis Exercise

Contents

1 Description of the work

The aim of the exercise is to analyze with Principal Component Analysis algorithms several data sets from the UCI repository. To this end, first of all you will implement the principal component analysis algorithm using **Python 2.7**.

1.1 Methodology of the analysis

You will analyze the behavior of the principal component analysis algorithm in well-known data sets from the UCI repository. These data sets are defined in **.arff** format. Use the parser implemented in previous work to read the .arff file in Python and save the information in a matrix.

1. Implement your own Principal Component Analysis algorithm in Python. You **cannot** use the PCA code that exists in `sklearn` library. You will implement it in several steps:
   - Step 1. Read the .arff file and take the whole data set consisting of d-dimensional samples ignoring the class labels. Save the information in a matrix.
   - Step 2. Plot the original data set (choose two or three of its features to visualize it).
   - Step 3. Compute the d-dimensional mean vector (i.e., the means of every dimension of the whole data set).
   - Step 4. Compute the covariance matrix of the whole data set. Show this information.
   - Step 5. Calculate eigenvectors (e_1, e_2, …, e_d) and their corresponding eigenvalues of the covariance matrix. Write them in console.
   - Step 6. Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a new d x k dimensional matrix (where every column represents an eigenvector). Write the sorted eigenvectors and eigenvalues in console.
   - Step 7. Derive the new data set. Use this d x k eigenvector matrix to transform the samples onto the new subspace.
   - Step 8. Plot the new subspace (choose the largest eigenvectors to plot the matrix).
   - Step 9. Reconstruct the data set back to the original one. Additionally, plot the data set.

2. Analyze your PCA algorithm in three data sets (see Section 2). At least two of them should be large enough to be able to extract conclusions. You will continue with the datasets chosen in the previous work.

3. Compare and analyze your results to the ones obtained using `sklearn.PCA` library.

1.2 Work to deliver

In this work, you will implement and analyze PCA algorithm. You may select 3 data sets (the ones chosen in Work 1) for your analysis. At the end, you will find a list of the data sets available (see Section 2).

The idea is that you implement **your own code in Python 2.7** and you will use it to produce the results of the analysis. Implement all the code in one python file and add detailed comments into the code.

Once you have obtained the results, you will show them in several ways:

1. Plot the figures of the data sets (original data set and transformed data set).
2. Show the information of the original matrix, the covariance matrix, eigenvectors, eigenvalues, and the transformed data set onto the new subspace.

From the outputs and plots, you will reason and extract conclusions about the results obtained. For example, some questions that may help you to comment your results:

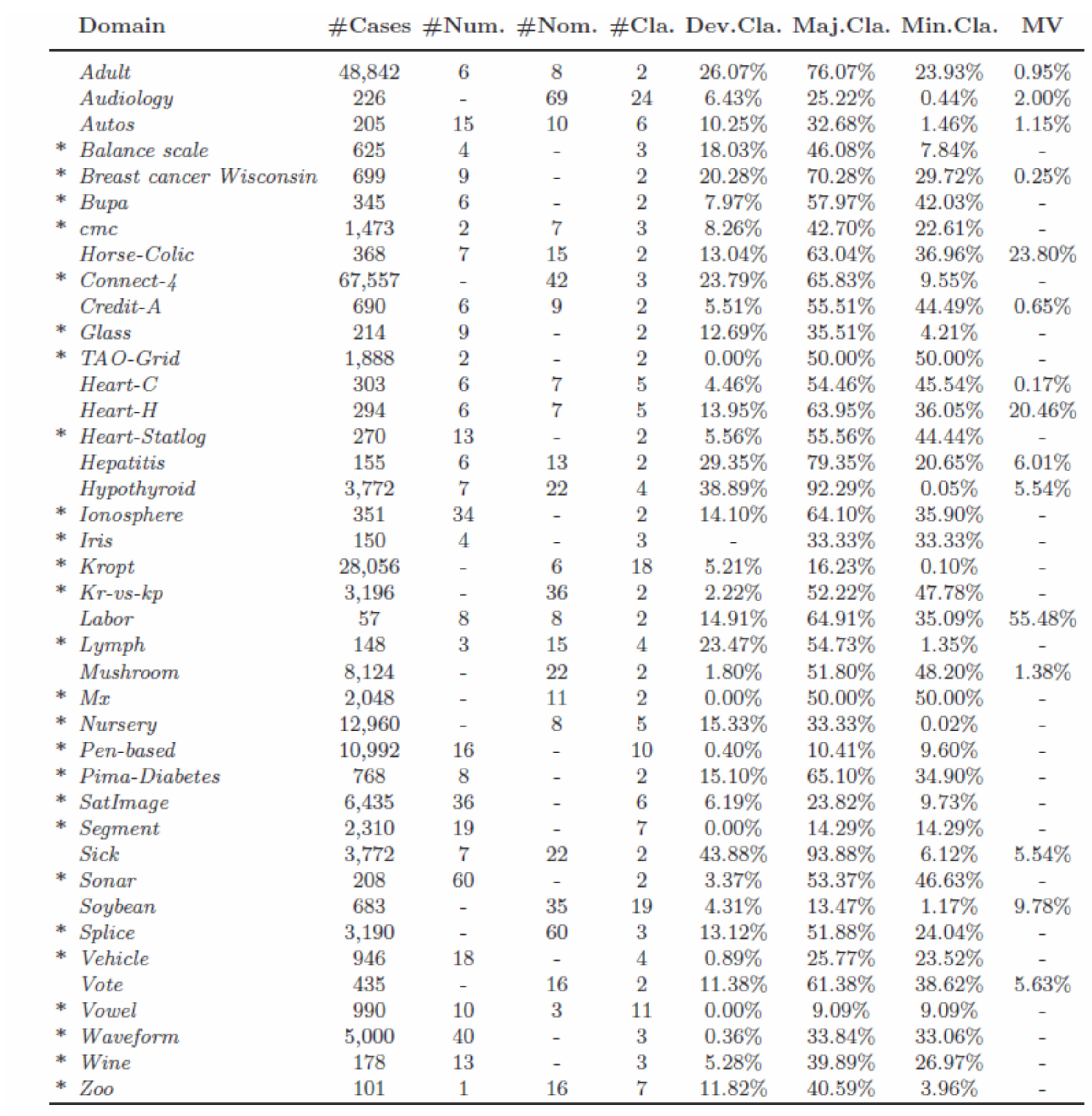
- Is PCA giving you more advice for knowing the underlying information in the data set?
- Can you explain the setup that you have used for PCA algorithm?
- Can you reduce the dimensionality of the data set? In case of an affirmative answer, detail how do you do.
- Do you obtain the same results from your code to the code in sklearn? Explain the similarities and the differences among the two implementations.

You should deliver a report as well as the code in Python 2.7 in Racó in a zip file by November, 19th 2017. The name of the zip file must contain name and surname of every member of the group.

The report will contain:
- Details about the implementation of your algorithms, including the decisions made during the implementation and the setup of the different parameters.
- The evaluation of the algorithm, including outputs in console and graphs that show your results with comments about them.
- Reason each one of the questions defined above in your evaluation and add any comment or observation that you consider important from your results.
- **Additionally, you should explain how to execute your code.**

2 Data sets

Below, you will find a table that shows in detail the data sets that you can use in this work. All these data sets are obtained from the UCI machine learning repository. First column describes the name of the domain or data set. Next columns show #Cases = Number of cases or instances in the data set, #Num. = Number of numeric attributes, #Nom = Number of nominal attributes, #Cla. = Number of classes, Dev.Cla. = Deviation of class distribution, Maj.Cla. = Percentage of instances belonging to the majority class, Min.Cla. = Percentage of instances belonging to the minority class, MV = Percentage of values with missing values (it means the percentage of unknown values in the data set). When the columns contain a '-', it means a 0. For example, the Glass data set contains 0 nominal attributes and it is complete as it does not contain missing values.



Domain	#Cases	#Num.	#Nom.	#Cla.	Dev.Cla.	Maj.Cla.	Min.Cla.	MV
Adult	48,842	6	8	2	26.07%	76.07%	23.93%	0.95%
Audiology	226	-	69	24	6.43%	25.22%	0.44%	2.00%
Autos	205	15	10	6	10.25%	32.68%	1.46%	1.15%
* *Balance scale*	625	4	-	3	18.03%	46.08%	7.84%	-
* *Breast cancer Wisconsin*	699	9	-	2	20.28%	70.28%	29.72%	0.25%
* *Bupa*	345	6	-	2	7.97%	57.97%	42.03%	-
* *cmc*	1,473	2	7	3	8.26%	42.70%	22.61%	-
Horse-Colic	368	7	15	2	13.04%	63.04%	36.96%	23.80%
* *Connect-4*	67,557	-	42	3	23.79%	65.83%	9.55%	-
Credit-A	690	6	9	2	5.51%	55.51%	44.49%	0.65%
* *Glass*	214	9	-	2	12.69%	35.51%	4.21%	-
* *TAO-Grid*	1,888	2	-	2	0.00%	50.00%	50.00%	-
Heart-C	303	6	7	5	4.46%	54.46%	45.54%	0.17%
Heart-H	294	6	7	5	13.95%	63.95%	36.05%	20.46%
* *Heart-Statlog*	270	13	-	2	5.56%	55.56%	44.44%	-
Hepatitis	155	6	13	2	29.35%	79.35%	20.65%	6.01%
Hypothyroid	3,772	7	22	4	38.89%	92.29%	0.05%	5.54%
* *Ionosphere*	351	34	-	2	14.10%	64.10%	35.90%	-
* *Iris*	150	4	-	3	-	33.33%	33.33%	-
* *Kropt*	28,056	-	6	18	5.21%	16.23%	0.10%	-
* *Kr-vs-kp*	3,196	-	36	2	2.22%	52.22%	47.78%	-
Labor	57	8	8	2	14.91%	64.91%	35.09%	55.48%
* *Lymph*	148	3	15	4	23.47%	54.73%	1.35%	-
Mushroom	8,124	-	22	2	1.80%	51.80%	48.20%	1.38%
* *Mx*	2,048	-	11	2	0.00%	50.00%	50.00%	-
* *Nursery*	12,960	-	8	5	15.33%	33.33%	0.02%	-
* *Pen-based*	10,992	16	-	10	0.40%	10.41%	9.60%	-
* *Pima-Diabetes*	768	8	-	2	15.10%	65.10%	34.90%	-
* *SatImage*	6,435	36	-	6	6.19%	23.82%	9.73%	-
* *Segment*	2,310	19	-	7	0.00%	14.29%	14.29%	-
Sick	3,772	7	22	2	43.88%	93.88%	6.12%	5.54%
* *Sonar*	208	60	-	2	3.37%	53.37%	46.63%	-
Soybean	683	-	35	19	4.31%	13.47%	1.17%	9.78%
* *Splice*	3,190	-	60	3	13.12%	51.88%	24.04%	-
* *Vehicle*	946	18	-	4	0.89%	25.77%	23.52%	-
Vote	435	-	16	2	11.38%	61.38%	38.62%	5.63%
* *Vowel*	990	10	3	11	0.00%	9.09%	9.09%	-
* *Waveform*	5,000	40	-	3	0.36%	33.84%	33.06%	-
* *Wine*	178	13	-	3	5.28%	39.89%	26.97%	-
* *Zoo*	101	1	16	7	11.82%	40.59%	3.96%	-