ex6 16/25 points

May 24, 2022

0.1 Exercise sheet 6

Cüneyt Erem 3277992 s6cuerem@uni-bonn.de

Nkeh Victor Ndiwago 3504121 s0vinkeh@uni-bonn.de

Paula Romero Jiménez 3320220 s0parome@uni-bonn.de

Exercise 1 - NMF Clustering (17 points) 9.5/17 points

1. Investigate the given gene expression (GE) dataset and you are asked to cluster the patients/samples based on their GE profile.

a. Which unsupervised ML algorithm would you suggest to tackle this task? (1 point) 1/1

- Agglomerative hierarchical clustering algorithms

b. Briefly explain the reasoning behind your suggestion. (2 points) 2/2

The objective of any agglomerative hierarchical clustering algorithm is to cluster a set of n objects (e.g., patients, genes) based on an n × n similarity matrix. The reason for choosing this clustering algorithm is due to its ability to simultaneously discover several layers of clustering structure, and visualize these layers via tree diagrams (i.e., dendrogram). Agglomerative clustering could take a lot of time on GE datasets.

2. Using the nimfa library, carry out NMF on GE data.

a. Fit the NMF model in the nimfa library to the GE data and using the cophenetic correlation, estimate the number of clusters (rank) needed to classify the samples in the GE data. 0/2

[ ]:
```
import pandas as pd
path= "data.csv"
data_f=pd.read_csv(path)
data_f
```

[ ]:
```
    Unnamed: 0  36638_at  39318_at  38514_at  266_s_at  38585_at  41266_at  \
0        01005  0.583368  0.535258  0.642984  0.891901  0.269871  0.605566
1        01010  0.505321  0.704177  0.913612  0.657634  0.402911  0.429698
2        03002  0.375805  0.073716  0.707562  0.847162  0.792428  0.819212
3        04006  1.000000  0.226960  0.119596  0.394317  0.115411  0.050117
4        04007  0.890125  0.631314  0.518785  0.880312  1.000000  0.858408
..         ...       ...       ...       ...       ...       ...       ...
123      56007  0.072759  0.068235  0.473041  0.239455  0.223252  0.280791
```

```
124      64005  0.053972  0.086147  0.000065  0.136057  0.488365  0.470852
125      65003  0.136163  0.082575  0.456977  0.120866  0.216307  0.239017
126      83001  0.022913  0.026321  0.774337  0.135344  0.379465  0.544207
127       LAL4  0.060686  0.091866  0.710210  0.218240  0.302652  0.624061

     36108_at  39389_at  31525_s_at  …  37655_at  1440_s_at  32260_at  \
0    0.069070  0.938101    0.314737  …  0.884590   0.399443  0.379796
1    0.803187  0.360471    0.718665  …  0.086277   0.432789  0.431440
2    0.644334  0.735292    0.828776  …  0.750758   0.305592  0.444996
3    0.440698  0.649291    0.498758  …  0.090901   0.365967  0.170796
4    0.638519  0.933899    0.963169  …  0.340640   0.285412  0.561193
..        …         …           …  …         …          …         …
123  0.082024  0.133053    0.768916  …  0.371955   0.343000  0.415382
124  0.053909  0.116930    0.861130  …  0.359889   0.557229  0.388962
125  0.069328  0.164105    0.565675  …  0.720453   0.632587  0.652855
126  0.049219  0.015351    0.724584  …  0.768953   0.357137  0.396453
127  0.251337  0.037572    0.452716  …  0.528329   0.787925  0.278878

     40070_at  1056_s_at  39200_s_at  36105_at  32578_at  39383_at  33718_at
0    0.272379   0.216088    0.355740  0.167203  0.438651  0.395720  0.273593
1    0.461242   0.443493    0.291054  0.102883  0.584035  0.589994  0.401375
2    0.293573   0.318939    0.259713  0.271869  0.556874  0.931118  0.301529
3    0.565408   0.456499    0.407845  0.100038  0.670769  0.541601  0.659517
4    0.470224   0.468927    0.243284  0.467118  0.571348  0.425701  0.196066
..        …          …           …         …         …         …         …
123  0.351072   0.351224    0.513185  0.136617  0.558943  0.225453  0.291071
124  0.375595   0.298626    0.584230  0.079315  0.353649  0.185538  0.461440
125  0.336831   0.153023    0.757112  0.179819  0.494060  0.277194  0.224542
126  0.674790   0.207006    0.287185  0.063400  0.327525  0.157277  0.337916
127  0.231955   0.233604    0.869236  0.174480  0.364662  0.237925  0.240359

[128 rows x 5001 columns]
```

[ ]:
```python
import numpy as np

import nimfa

V = np.random.rand(23, 200)

# Factorization will be run 3 times (n_run) and factors will be tracked for␣
 ↪computing
# cophenetic correlation. Note increased time and space complexity
bmf = nimfa.Bmf(V, max_iter=10, rank=30, n_run=3, track_factor=True)
bmf_fit = bmf()

print('K-L divergence: %5.3f' % bmf_fit.distance(metric='kl'))
```

```
sm = bmf_fit.summary()
print('Rss: %5.3f' % sm['rss'])
print('Evar: %5.3f' % sm['evar'])
print('Iterations: %d' % sm['n_iter'])
print('Cophenetic correlation: %5.3f' % sm['cophenetic'])
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-5-306dec2be5f4> in <module>
      1 import numpy as np
      2
----> 3 import nimfa
      4
      5 V = np.random.rand(23, 200)

ModuleNotFoundError: No module named 'nimfa'
```

???

---

3)) Inform yourself about sparse NMF (sNMF) and Non-Negative Matrix Tri- Factorization (NMTF)

a. What is the primary difference between NMF, sNMF and NMTF? (2 points) 2/2 points

NMF decomposes a nonnegative matrix into the product of two matrices (W, H).

sNMF additionally decomposes a nonnegative matrix into the product of two matrices (W, H) and for each cluster, it results into sparse set of features. applies sparsity constraints to both matrices

NMTF decomposes a nonnegative matrix into the product of three matrices (W, S, H).

NMF do not take into account for training labels and unlabeled test data. NMTF is more flexible and not limited for data residing in one space for tranforming training examples to test examples. So it is good for non squared data and provides more degrees of freedom than NMF.

b. Which amongst the 3 NMF would be suitable for the dataset mentioned in question 1? (1 point) 0.5/1 points

sNMF is better for sparsity with genomic data. Check model answer.

dataset consists of more than 50,000 columns which means it has a big data. NMTF uses three factorization, therefore NMTF will outperform NMF and sNMF average AC results.

c. Briefly explain the reasoning behind your suggestion in question 3b. (2 points) 0/2 points

4)) PCA and NMF are both matrix factorization methods. Write down the corresponding formulas and compare them. What is similar, what is mathematically different? Describe a situation where NMF is favored over PCA. (2 points)

formulas; 2/2 points

PCA;

mean $\sum = \frac{1}{n-1} \sum_i^n x_i * x_i^T$

variance $\frac{1}{n-1}\sum_i^n (x_i^T * v)^2 = \frac{1}{n-1}\sum_i^n \tilde{\phi}^2$

Explain whats different between both methods mathematically. Check model answers.

NMF:

$D(X||WH) = \sum_{i,j}(x_{ij} log \frac{x_{ij}}{w_{ij}h_{ij}} - x_{ij} + w_{ij} * h_{ij})$

$w_{ik} = w_{ik} \frac{\sum_{j=1}^p w_{kj} x_{ij}/(WH)_{ij}}{\sum_{j=1}^p h_{kj}}$

$h_{kj} = h_{kj} \frac{\sum_{j=1}^N w_{ik} x_{ij}/(WH)_{ij}}{\sum_{j=1}^N w_{ik}}$

To give an example, NMF divides a face into a set of features that you can interpret as "nose", "eyes" that you can combine to reconstruct the original image. PCA however gives "overall" faces sorted by how well they capture the original image.

NMF decomposes a dataset matrix into its non-negative submatrix and PCA gives new data feature as a result of the combination of the existing original example.

For linearly separable data and reducing large number of features, PCA is better. For image processing, word and vocabulary recognition NMF gives better results.

5)) Have a look at this paper: https://arxiv.org/abs/1512.07548 and explain, why k- means clustering can be understood as a matrix factorization problem as well. (2 points)

2/2 points

The paper mentions about previous papers showed k-means clustering can be understood as a constrained matrix factorization problem, however it is not understood in detail. So, this paper shows mathematical proof of hard k-means clustering function can be shown as matrix factorization problem.

$\sum = \frac{1}{n-1}\sum_i^n x_i * x_i^T$

function of hard k-means clustering;

$\sum_{i=1}^k \sum_{j=1}^n z_{ij}||x_j - \mu_i||^2 = ||X - MZ||^2 = ||X - XZ^T(ZZ^T)^{-1}Z||^2$

the objective of the paper is to prove these statements as above,

in notion and preliminary, it gives information about squared Frobenius norm of a matrix as $||X||^2 = \sum_j ||X||^2 = tr[X^T X]$

then it implements step by step derivation of the k-means clustering function and expansion of each statement,

first it shows this equation; $ *{i=1}^{k}* {j=1}^{n} z_{ij}||x_j - _i||^{2} $ as T1, T2 and T3.

then by using notions and preliminaries; it finds T1 as $tr[X^T X]$, T2 as $tr[X^T MZ]$ and T3 as $\sum_i ||\mu_i||^2 n_i$

then it expands $||X - MZ||^2$ as T4, T5, T6

T1 = T4 and T2 = T5, and need to check whether T3 = T6?

writer finds out that $tr[Z^T M^T MZ] = \sum (M^T M)_{ii}(ZZ^T)_{ii} = \sum_i ||\mu_i||^2 n_i$ so T3 also equals to T6

then it eliminates M as taking partial derivative of $ _{j}||X||^{2}$ with respect to M; and set the equation to 0, then it yields $ M = XZ^{T}(ZZ^{T})^{-1}$ that each of the k-means cluster centroids μi coincides with the mean of the corresponding Ci

so k-means clustering can be understood as constrained matrix factorization problem as;

$min_z||X - XZ^T(ZZ^T)^{-1}Z||^2$ st $z_{ij} \in 0, 1$ and $\sum_j z_{ij} = 1$

0.1.1 Exercise 2 - Machine Learning (8 points) 6.5/8

The type of machine learning (e.g. supervised learning, unsupervised learning, etc.) applied depends on the problem at hand. Assume that we have an Alzheimer's disease (AD) dataset where rows represent 500 participants and columns represent 100 different collected measurements (such as patient characteristics, MRI measurements, and cognitive tests) for each participant. We are provided with diagnoses (healthy, and AD) of participants.

1) You are asked as a data scientist to predict the diagnosis status of 20 participants based on a model trained with data from 500 participants ***a) What type of machine learning (ML) would you choose for this task? (1 point)*** 1/1

I would choose a supervised learning (predictive modeling), because we can train the model with fully labeled samples and the main goal is to predict the diagnosis.

b) What are the steps you should consider before training your ML algorithm? Hint: List the potential preprocessing steps you would carry out. (2 points)

Before training the ML algorithm we should clean and preprocess the data. 2/2

- Load the AD dataset into pandas.
- Search for missing values. If there are, we have to decide if we drop the rows or we fill them by imputation (replacing them with their values, such as the median). I would say we could drop them if they are few because we have a dataset of 500 patients, but if they are a large number we should fill them.
- Perform feature selection, for example by correlation coefficient. This way we use only relevant data and we get rid of noise.
- Splitting the data into training and testing sets. And then the training set is divided again in training and validation set (only the training set is going to be used to train the model).
- Cross-validation to figure out the best parameters for each model.

2) Assuming that we do not have any information about the diagnosis (i.e. no label) of participants, answer the following questions. ***a) What type of machine learning would you use to group the participants based on the collected measurements? (1 point)***

If we are learning from a dataset without labels, we will use an unsupervised learning. 1/1

b) Based on your answer in part (2a), suggest a model you would use to carry on with this task, and please explain how you can determine the number of groups that separates your participants? (2 points)

For an unsupervised learning we could use clustering, this way data would be groupped into clusters depending on the presence or absence of commonalities between objects. We expect a significant difference between the data of healthy and AD patients so that the model can work properly. We could use k-means or k-nearest neighbours...

2/2

To determine the number of groups that separate the participants, in this case the number of clusters, there are some techniques that can help with that. One option to find the optimal number of clusters is to calculate the BIC, other option is the silhoutte index, etc. All these methods are going to give us the number of clusters in which the data fits best.

3) Imagine that the shape of our dataset is (100, 1000), mention one pre-processing step that you would take to carry out the tasks (1) and (2)? (1 point) 0/1

4) You are asked to investigate the age distribution of healthy versus AD participants, name a visualization plot that can be used in this case, and explain why you think this plot would work best. (1 point) I think a bar histogram would work best. An histogram allows us to represent the distribution and relationship of a single variable (age) over a set of categories (age-group). Then if we want to relate that to the distrubution of healthy vs AD, we could do a bar chart so we see the distribution of age and diagnosis. 0.5/1

Bar chart does not show you a good representation of your data. In theory, you can visualize your data but plots like Violin or density would be a better choice.