Introduction to Statistical Machine Learning for Bioinformaticians and Medical Informaticians

SoSe 2022

Tutor: Dana Petracek

1	2	$\sum$

Marina Dittschar & Clarissa Auckenthaler

Assignment 10

(Submitted 18.07.2022)

Task 1

The similarity of two clustering results (one of which might be the ground truth) can be assessed with a measure called Rand index. It can be calculated with the *randIndex* function in the `flexclust-package`.

a) **Explain the Rand index and its adjusted variant in your own words. (7)**

- Rand index (RI)

  The Rand index is a simple way to compare the similarity of results between two different clustering methods. Instead of calculating the correctly classified items to all items RI was extended for comparing two cluster. RI counts the correctly clustered pairs of items. It is defined as:

$$RI = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}} \tag{1}$$

  With:
  a= the number of pairs of elements in the same clusters C and C^*
  b= the number of pairs in different clusters C and C^*
  c= the number of pairs in same cluster C and in different cluster C^*
  d= number of pairs in different cluster C and same cluster C^*

  Rand index for two clusters (binary classification problem) is the same as accuracy and can also be calculated as such:[4]

$$RI = \frac{TP+TN}{TP+FP+FN+TN} \tag{2}$$

  With:
  TP = true positives
  TN = true negatives
  FP = false positives
  FN = false negatives

- Adjusted Rand index (ARI)

  The Adjusted Rand Index (ARI) is the is the corrected-for-chance version of the Rand index and is defined by the following formula [1]:

$$ARI(P*, P) = \frac{\sum_{i,j}\binom{N_{ij}}{2} - \left[\sum_i \binom{N_i}{2} \sum_j \binom{N_j}{2}\right] / \binom{N}{2}}{\frac{1}{2}\left[\sum_i \binom{N_i}{2} \sum_j \binom{N_j}{2}\right] - \left[\sum_i \binom{N_i}{2} \sum_j \binom{N_j}{2}\right] / \binom{N}{2}} \tag{3}$$

With:
N= number of datapoints
N_ij = number of datapoints cluster labels $C_j^* \in P^*$ assigned to cluster $C_i \in P$
N_i = number of datapoints assigned to cluster C_i of partition P
N_j = number of datapoints assigned to cluster C_j^*
where N_ij, N_i ,N_j are values from the confusion matrix.

Both values (RI and ARI) lie between 0 and 1, the only difference is that the results of the ARI calculation can yield to negative values, in the case where the index is less than the expected index. If the partition is completely identical then the index value is 1, means higher values identify the better reconstruction of clusters. ARI should be interpreted as follows [3]:

* $ARI \geq 0.90 \longrightarrow$ excellent cluster recovery
* $ARI \geq 0.80$ & $ARI \leq 0.90 \longrightarrow$ good cluster recovery
* $ARI \geq 0.65$ & $ARI \leq 0.80 \longrightarrow$ moderate cluster recovery
* $ARI \leq 0.65 \longrightarrow$ bad cluster recovery

b) **Implement the non-adjusted Rand index in R, show your code in the pdf (properly formatted) and verify, on a reasonable selection of test examples, that it produces the same results as the corresponding function from the above-mentioned package. (8)**

```r
#RandIndex function
RandIndex=function(cluster_1, cluster_2){
  a = abs(sapply(cluster_1, function(i) i - cluster_1))
  a[a > 1] = 1
  b = abs(sapply(cluster_2, function(i) i - cluster_2))
  b[b > 1] = 1
  falses = sum(abs(a - b))/2
  no.pairs = choose(dim(a)[1], 2)
  ri = 1 - falses/ no.pairs
  return(ri)
}

```

Trying out this randindex function with a confusion matrix with 3 clusters and 6 samples and a confusion matrix with 5 clusters yielded exactly the same values, 0.6 in the first and 0.6800647 in the second case, respectively. This shows that the function works both with very small and very large data sets.

Task 2

a) **Apply agglomerative clustering with average linkage to the data set and plot the resulting dendrogram. (2)**

We can see the resulting dendrogram in figure 1.

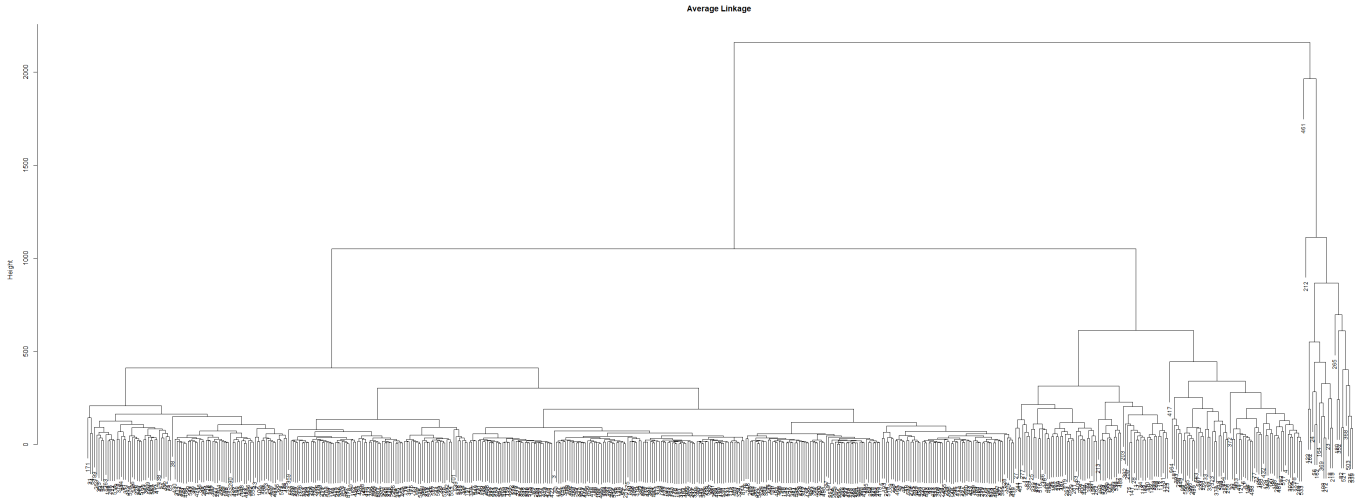


Figure 1: Resulting dendrogram for average linkage

b) **Determine cluster labels for each observation. Use different cutoffs that yield 2-6 clusters respectively. For each clustering, compare the resulting labels with the binary class assignment (column 2 of the original data) using a confusion matrix. How well does the clustering distinguish the two tumor subtypes? (3)** You can see the confusion matrices for each number of clusters in table 1. As you can see, the classification is not meaningful for two clusters, as most values, no matter the label, are clustered in cluster one. This persists for 3 and 4 clusters, with additional clusters only having minimal cluster sizes. However, the values change for 5 and 6 clusters, now the approach assigns benign and malignant clusters different labels in the majority of cases. However, there are still clusters with only a minimal sample size of 1.

c) **Further assess these clustering numerically using the Rand index and adjusted Rand index (cf. Task 1). What do you observe? (2)**

In table 1, we can find all confusion matrices for all different cutoffs (2-6). Interestingly, for two clusters, the values are very low, with a RI of 0.5564 and an ARI of 0.0617. For cluster numbers 3 and 4, the values further go down slightly, with the lowest value being 0.5561 for the RI and 0.0611 for the ARI for four clusters. For 5 and 6 clusters, the values rise dramatically, reaching a maximum value for 5 clusters with a RI value of 0.7689 and an ARI value of 0.5325. However, as described above, the ARI value is still very bad, which indicates thhat the clustering does not distinguish the two tumor subtypes well.

number of clusters ($\hat{K}$)	**2**			3			4			5			6		
		B	**M**		**B**	**M**		**B**	**M**		**B**	**M**		**B**	**M**
	1	357	188	**1**	357	188	**1**	357	188	**1**	5	124	**1**	5	124
	2	0	23	**2**	0	22	**2**	0	22	**2**	352	64	**2**	352	64
confusion matrices				**3**	0	1	**3**	0	1	**3**	0	21	**3**	0	12
							4	0	1	**4**	0	1	**4**	0	9
										5	0	1	**5**	0	1
													6	0	1
ARI	0.06166122			0.06138974			0.0611306			0.5324931			0.5311772		
RI	0.55635045			0.55621383			0.5560834			0.7689160			0.7682453		

Table 1: Confusion matrices for different number of clusters for the method cutree.

d) **Apply K-means to the same data set with K 2, 3, ..., 20. Use an appropriate amount of computational effort (iterations, restarts) to obtain stable results. Plot W_K against K and pick an appropriate value $\hat{K}$. Compare again the labels of the optimal clustering $\hat{K}$ with the binary class assignment (column 2 of the original data) by calculating a confusion matrix and the (adjusted) Rand index.(5)**

We chose a maximum iteration value of 50 iterations. The figure 4 displayed the W_K values against the different K (2-20) values. Based on this results it was a bit difficult to detect the elbow. But we decided to choose $\hat{K} = 11$. Because the elbow was hard to determine, we also compared the confusion matrices from $\hat{K} = 7$,$\hat{K} = 11$,and $\hat{K} = 13$. The results with $\hat{K} = 11$ lead to the closest recovery of the clusters. The table 2 shows the results for $\hat{K} = 11$. The calculated ARI based on the corrected confusion matrix is 0.182, this indicates a very bad recovery for this dataset. Indeed, k-means can be unsuitable for binary data [2].

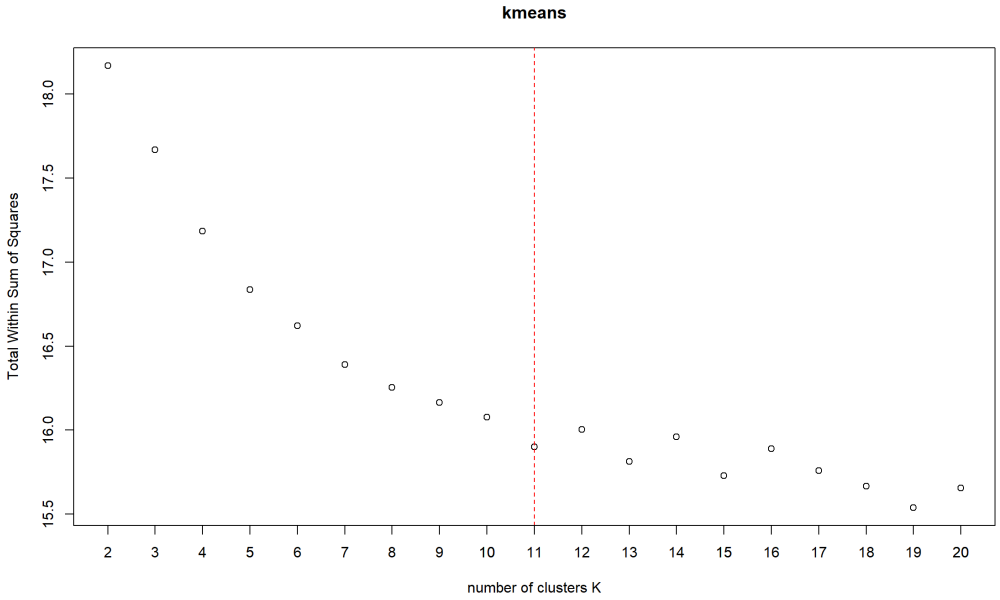


Figure 2: Plot W_K against K and pick an appropriate value $\hat{K}$ (displayed in red)

number of clusters ($\hat{K}$)	**11**	
confusion matrix	**B**	**M**
1	98	10
2	0	4
3	129	5
4	7	37
5	0	17
6	1	37
7	0	32
8	78	0
9	0	9
10	0	33
11	44	27
ARI	0.1819867	

Table 2: Results (unscaled) data of kmeans for k= 11

e) **Repeat the analysis in in a) - d) with normalized features (mean 0, stddev 1). How does the normalization affect the clustering (improvement/worsening/no effect)? (2)**

Figure 3 displays the dendogram for the scaled data. This dendrogram was even more detailed and complicated than the dendrogram on unscaled data. Judging from the performance on this data set, it would make sense to use non-normalized data for average linkage.

Although on the cutree-method the scaling has a worsening effect, Table 3 shows extremely bad recovery for all cluster numbers. The ARI values for all k-values were very close to zero. The performance of this method was clearly badly affected by scaling the data.

In the W_K against number of clusters plot (Figure 4) we again chose a value of 11 for the number of clusters (Figure 2). The performance of k-means increased by scaling the data. However, this does not seem to be represented in the ARI (see Table 4), as the value is approximately 2% lower than for non-normalized data.

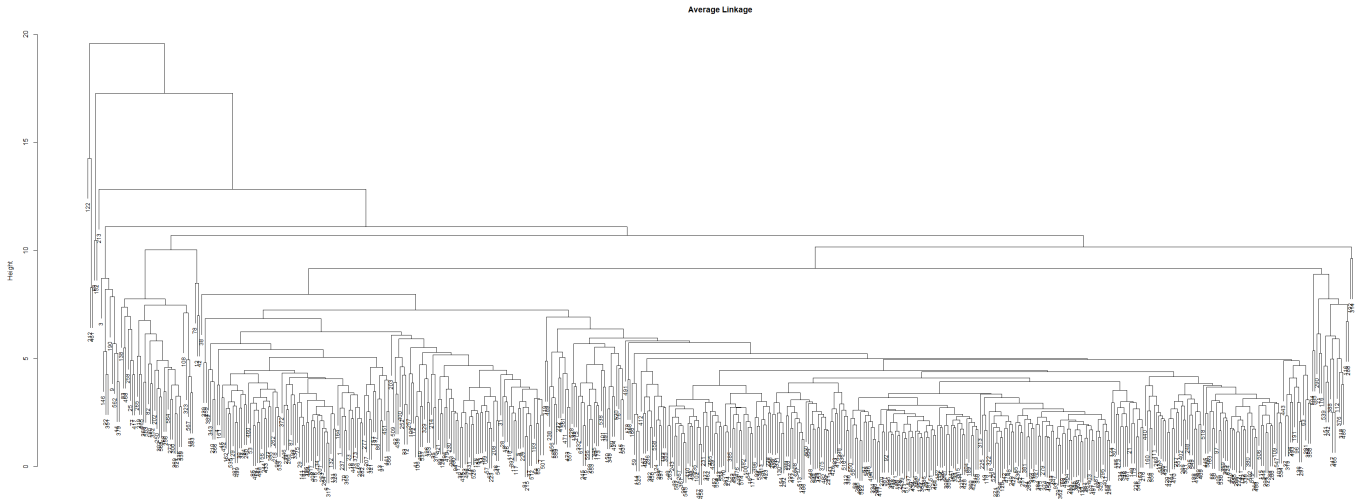


Figure 3: Resulting dendrogram for average linkage (normalized data)

number of clusters ($\hat{K}$)	**2**			**3**			**4**			**5**			**6**		
		B	**M**		**B**	**M**		**B**	**M**		**B**	**M**		**B**	**M**
	1	357	208	**1**	355	208	**1**	355	208	**1**	355	207	**1**	355	198
	2	0	3	**2**	2	0	**2**	2	0	**2**	2	0	**2**	0	9
confusion matrics				**3**	0	3	**3**	0	1	**3**	0	1	**3**	2	0
							4	0	2	**4**	0	2	**4**	0	1
										5	0	1	**5**	0	2
													6	0	1
ARI	0.007373091			0.004433115			0.004408327			0.006844574			0.02966703		
RI	0.534987704			0.533161935			0.533149514			0.534068609			0.54284348		

Table 3: Confusion matrices for different number of clusters for the method cutree (scaled data).

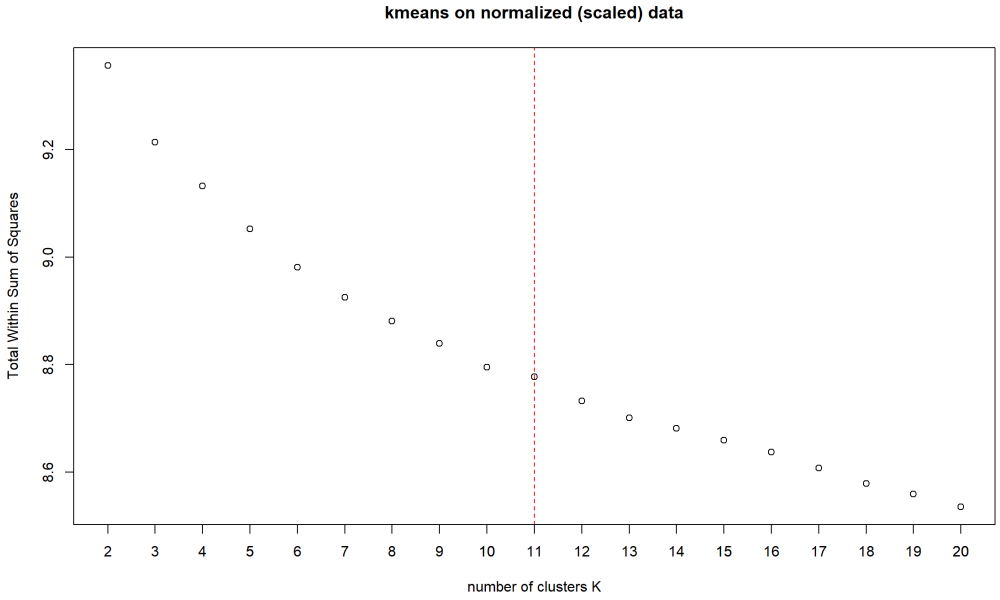


Figure 4: Plot W_K against K and pick an appropriate value $\hat{K}$ (displayed in red) for scaled data.

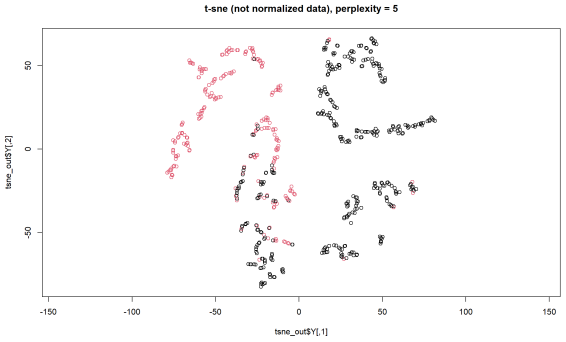
number of clusters ($\hat{K}$)	**11**	
confusion matrix	**B**	**M**
1	0	12
2	0	53
3	87	6
4	9	1
5	68	0
6	0	51
7	58	19
8	0	14
9	34	3
10	95	0
11	6	5
ARI	0.1633159	

Table 4: Results for the scaled data of kmeans with k= 11

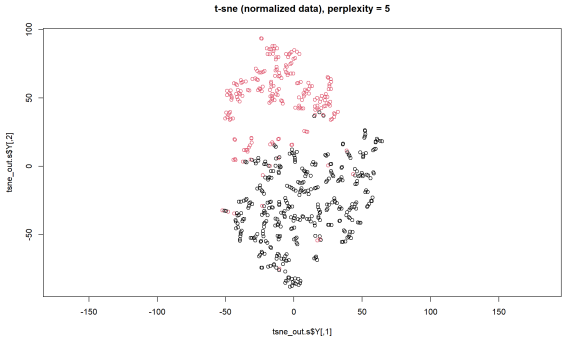
f) **Considering all results, would you prefer agglomerative clustering or K-means for this type of data? (2)´**

Based on our results we would prefer to use agglomerative clustering on unscaled data because it leads to the best recovery between the two clustering methods on scaled and unscaled data.
Our results indicate that scaling the data does not improve performance, and using k-means for binary data produced insufficient results for both scaled and unscaled data. However, it should be noted that with an ARI of 0.5312, the performance of agglomerative clustering might still not be sufficient [3].

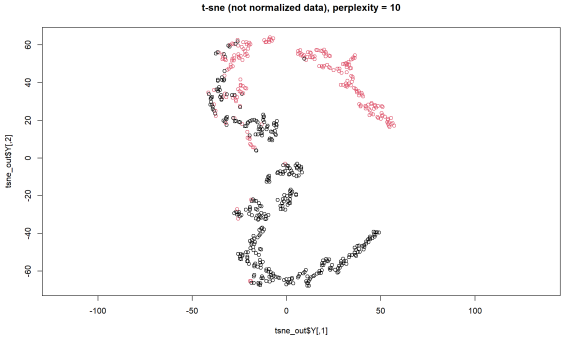
g) **Apply a dimension reduction with t-SNE down to two dimensions for four different perplexity parameter values (5,10,20,50). Compare both variants with and without feature normalization (yielding 8 plots in total). Color the data points according to their true class. What do you observe? For reproducibility, invoke set.seed(1) before each t-SNE run.(5)**



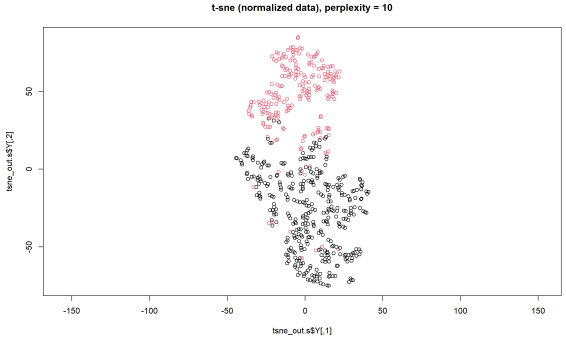
(a)



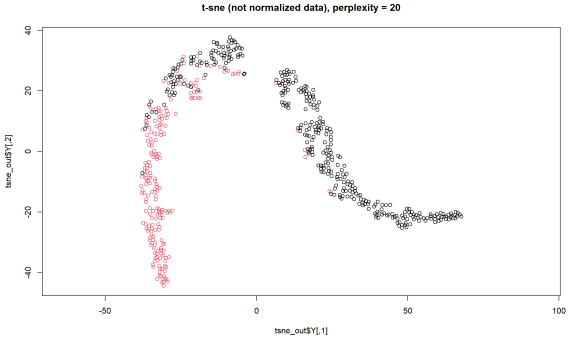
(b)



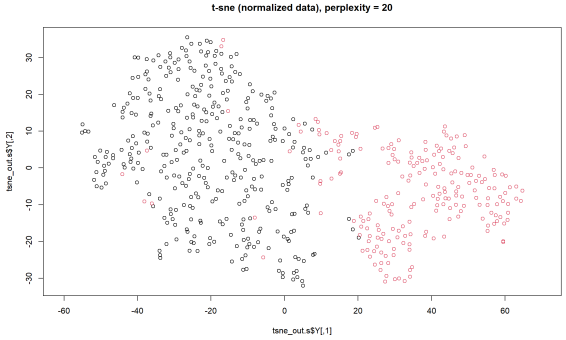
(c)



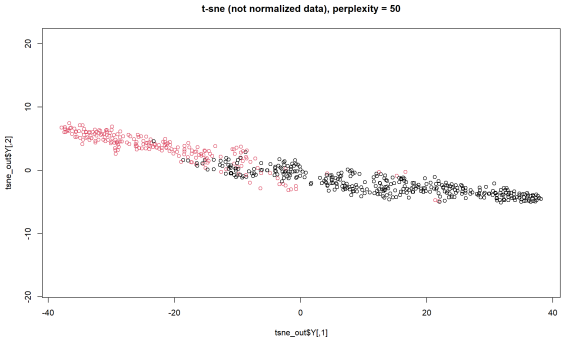
(d)



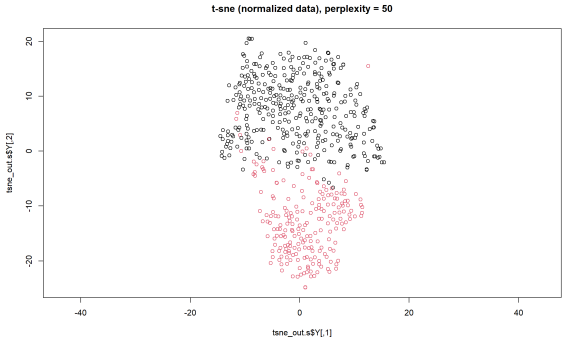
(e)



(f)



(g)



(h)

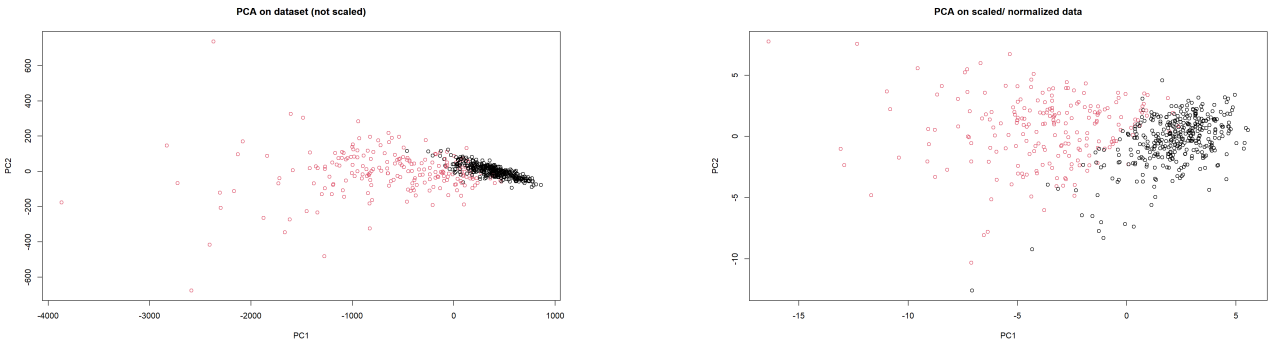
Figure 5: Output plots of t-sne for different perplexity values (5,10,20,50). On the left side are the results for the not scaled data and on the right side the results for the scaled data.

We can see that generally, normalizing the data greatly increases the performance of t-SNE, as true labels are much closer to the assigned clusters in the normalized plots than in the non-normalized plots. This is true for all tested values for perplexity between 5 and 50. The structure of the clusters on non-normalized

data becomes less "noisy" as perplexity (and therefore entropy) is increased, but classification does not improve.

In t-SNE on normalized data, classification performance is pretty robust to changes in perplexity. For all values, overall clustering is good, with some "mis-clusterings" in all cases. Visually, the clusters (corresponding to the true labels) could be identified even before coloring samples from different groups.

h) **Apply a dimension reduction with PCA down to two dimensions. Color the data points according to their true class. Once again, study the effect of feature normalization. Compare with the t-SNE results. What do you observe? (4)**



(a) (b)

Figure 6: PCA output plots, on the left side is the result for the not scaled data and on the right side the result for the scaled data.

We can see that in PCA, feature normalization also removes global geometry and in principle, improves cluster identification. Whereas without normalization, no distinct clustering could be detected between classes, there is some structure in the PCA result with normalized data. However,the clusters are not distinctive to the observer when the true labels are not coloured in.T-SNE on the other hand performs much better on this data set, as distinct clusters could be observed above.

References

[1] Alexander J Gates and Yong-Yeol Ahn. The impact of random models on clustering similarity, 2017.

[2] IBM. Clustering binary data with k-means (should be avoided). `https://www.ibm.com/support/pages/clustering-binary-data-k-means-should-be-avoided`, 2020.

[3] Paola Tellaroli , Marco Bazzi , Michele Donato , Livio Finos , Philippe Courcoux , Corrado Lanera. Computes the adjusted rand index and the confidence interval. `https://rdrr.io/cran/CrossClustering/man/ari.html`, urldate = 17/07/2022, 30/07/2018.

[4] Purdue University. Model building: Selection criteria: Stat 512 , spring 2011: Background reading knnl: Chapter 9.