Assignment 5

Submitted by:
Anirban Ghosh - 3279100 - s6anghos@uni-bonn.de
Kajal Puri - 3305398 - s6kapuri@uni-bonn.de
Shilpa Chaterjee - 404788-shilpa.chatterjee@rwth-aachen.de

Course: *Visual Data Analysis (MA-INF 2222)* – Professor: *Dr. Thomas Schultz*
Submission Date: *May 25th, 2020*

Exercise 2

a) Please submit your results, along with a guess of what might have been the criterion in each of the two cases.

Answer. We guess that the first Mat is compared wrt the size of balls and the second matrix is a comparison of the size of the field for the different sports. PFA the results.

Exercise 4

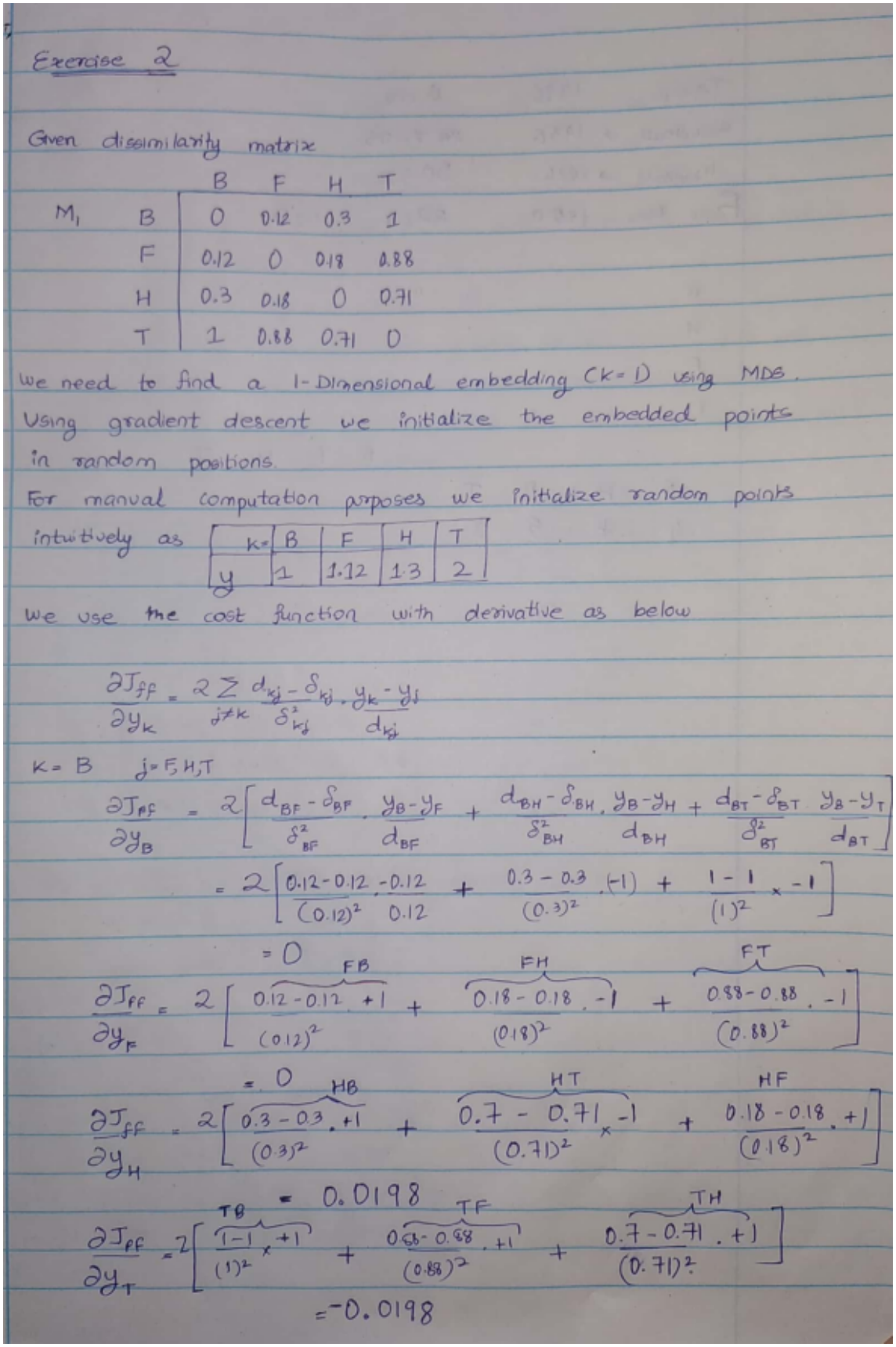
a) Pick the "three clusters with equal numbers of points" data set. Set the number of points per class to 10, and number of dimensions to 50. Once run the demo with perplexity=29, and once with perplexity=30. Explain why there is a big difference in the final 2D embedding

Answer. The big difference in the final 2D embedding illustrates to the pitfall; for the algorithm to work perfectly the perplexity really should be smaller than the number of points. Implementations can give unexpected behavior otherwise.

b) Try the example "a square grid with equal spacing between points", with 20 points per side. In the resulting plot with perplexity=100, why are distances between points in the middle of the square larger than near the boundary?

Answer. Since we tune the perplexity to 100 the changes the local aspects of the data with respect to its nearest neighbors. Since the paper does tell us that the best results are illustrated when we have perplexity 5-50

c) Pick "a square grid with equal spacing between points" data set, with 20 points per side, and perplexity=2. Run the t-SNE multiple times. You will observe that the square grid sometimes breaks down into separate smaller clusters. Why?



Exercise 2

Given dissimilarity matrix

M_1		B	F	H	T
	B	0	0.12	0.3	1
	F	0.12	0	0.18	0.88
	H	0.3	0.18	0	0.71
	T	1	0.88	0.71	0

We need to find a 1-Dimensional embedding (k=1) using MDS.
Using gradient descent we initialize the embedded points in random positions.
For manual computation purposes we initialize random points intuitively as

k=	B	F	H	T
y	1	1.12	1.3	2

We use the cost function with derivative as below

$$\frac{\partial J_{ff}}{\partial y_k} = 2 \sum_{j \neq k} \frac{d_{kj} - \delta_{kj}}{\delta_{kj}^2} \cdot \frac{y_k - y_j}{d_{kj}}$$

K = B j = F, H, T

$$\frac{\partial J_{ff}}{\partial y_B} = 2\left[\frac{d_{BF} - \delta_{BF}}{\delta_{BF}^2} \cdot \frac{y_B - y_F}{d_{BF}} + \frac{d_{BH} - \delta_{BH}}{\delta_{BH}^2} \cdot \frac{y_B - y_H}{d_{BH}} + \frac{d_{BT} - \delta_{BT}}{\delta_{BT}^2} \cdot \frac{y_B - y_T}{d_{BT}} \right]$$

$$= 2\left[\frac{0.12 - 0.12}{(0.12)^2} \cdot \frac{-0.12}{0.12} + \frac{0.3 - 0.3}{(0.3)^2} \cdot (-1) + \frac{1 - 1}{(1)^2} \times -1 \right]$$

$$= 0$$

$$\frac{\partial J_{ff}}{\partial y_F} = 2\left[\overbrace{\frac{0.12 - 0.12}{(0.12)^2} \cdot +1}^{FB} + \overbrace{\frac{0.18 - 0.18}{(0.18)^2} \cdot -1}^{FH} + \overbrace{\frac{0.88 - 0.88}{(0.88)^2} \cdot -1}^{FT} \right]$$

$$= 0$$

$$\frac{\partial J_{ff}}{\partial y_H} = 2\left[\overbrace{\frac{0.3 - 0.3}{(0.3)^2} \cdot +1}^{HB} + \overbrace{\frac{0.7 - 0.71}{(0.71)^2} \times -1}^{HT} + \overbrace{\frac{0.18 - 0.18}{(0.18)^2} \cdot +1}^{HF} \right]$$

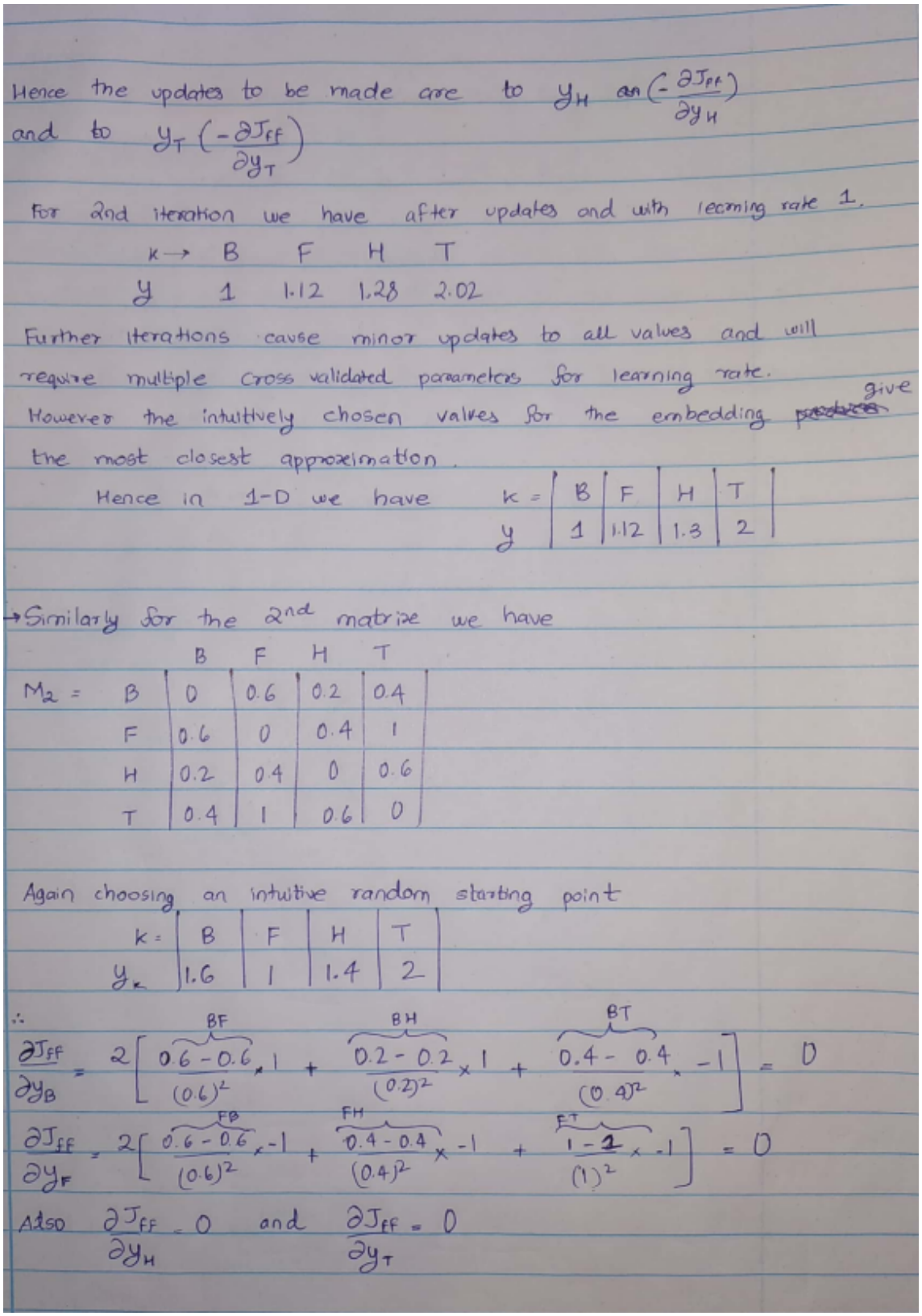
$$= 0.0198$$

$$\frac{\partial J_{ff}}{\partial y_T} = 2\left[\overbrace{\frac{1 - 1}{(1)^2} \times +1}^{TB} + \overbrace{\frac{0.88 - 0.88}{(0.88)^2} \cdot +1}^{TF} + \overbrace{\frac{0.7 - 0.71}{(0.71)^2} \cdot +1}^{TH} \right]$$

$$= -0.0198$$

Figure 1: Q2

Anirban Ghosh - 3279100 - s6anghos@uni-bonn.de
Kajal Puri - 3305398 - s6kapuri@uni-bonn.de
Shilpa Chaterjee - 404788-shilpa.chatterjee@rwth-aachen.de



Hence the updates to be made are to y_H as $\left(-\frac{\partial J_{FF}}{\partial y_H}\right)$ and to $y_T\left(-\frac{\partial J_{FF}}{\partial y_T}\right)$

For 2nd iteration we have after updates and with learning rate 1.

k →	B	F	H	T
y	1	1.12	1.28	2.02

Further iterations cause minor updates to all values and will require multiple cross validated parameters for learning rate.
However the intuitively chosen values for the embedding give the most closest approximation.

Hence in 1-D we have

k =	B	F	H	T
y	1	1.12	1.3	2

→ Similarly for the 2nd matrix we have

M_2 =

	B	F	H	T
B	0	0.6	0.2	0.4
F	0.6	0	0.4	1
H	0.2	0.4	0	0.6
T	0.4	1	0.6	0

Again choosing an intuitive random starting point

k =	B	F	H	T
y_k	1.6	1	1.4	2

$$\therefore \frac{\partial J_{FF}}{\partial y_B} = 2\left[\overbrace{\frac{0.6-0.6}{(0.6)^2}}^{BF} \times 1 + \overbrace{\frac{0.2-0.2}{(0.2)^2}}^{BH} \times 1 + \overbrace{\frac{0.4-0.4}{(0.4)^2}}^{BT} \times -1\right] = 0$$

$$\frac{\partial J_{FF}}{\partial y_F} = 2\left[\overbrace{\frac{0.6-0.6}{(0.6)^2}}^{FB} \times -1 + \overbrace{\frac{0.4-0.4}{(0.4)^2}}^{FH} \times -1 + \overbrace{\frac{1-1}{(1)^2}}^{FT} \times -1\right] = 0$$

Also $\frac{\partial J_{FF}}{\partial y_H} = 0$ and $\frac{\partial J_{FF}}{\partial y_T} = 0$

Figure 2: Q2

Anirban Ghosh - 3279100 - s6anghos@uni-bonn.de
Kajal Puri - 3305398 - s6kapuri@uni-bonn.de
Shilpa Chaterjee - 404788-shilpa.chatterjee@rwth-aachen.de

Answer. Distance between the clusters may get compressed by the KL divergence which causes to form square grids at times.

d) Use different perplexities for "points randomly distributed in a circle" with 100 points. Around what perplexity value does the resulting visualization start to resemble the input data set? Explain why the perplexity has to be large enough for the result to look like the input.

Answer. Since the number of points are high enough it requires higher perplexity value to show the local data points similar to its global response. We observe the result closer to the input when the perplexity is about greater than 75.

Exercise 5

a) Briefly mention the steps involved in the tSNE algorithm. At which point does A-tSNE introduce an approximation?

Answer. tSNE interprets the the distances in high dimensional data-points as a symmetric joint probability distribution.

- Let us consider the mapping of data points in high dimensional space as symmetric distribution as P.
- Now Q is computed which is a joint probability distribution that describes the similarity between data points in low dimensional space.
- In order for Q to accurately represent P, we we optimize the positions of points in the lower dimension to minimize the cost function C given by the Kullback-Leibler (KL) divergence between P and Q
- The KL divergence uses the relative similarity measure between neighbours.

The A-tSNE improves the similarity computation step. Specifically, it approximates this computation using Approximated K-Nearest Neighborhood (KNN) queries. This allowes to begin the iterative PVA minimization almost immediately after the data is loaded.

b) The paper frequently refers to the concept of Progressive Visual Analytics. What is meant by this? In which sense could the original tSNE algorithm be used for Progressive Visual Analytics? How has this been extended by A-tSNE?

Answer. The idea of Progressive Data Analytics is to provide the user with meaningful intermediate results which can be helpful in the analysis process instead of providing the final result since it could me more expensive.

The t-SNE can fit the PVA method because it can provide intermediate results because it is based on the iterative gradient descent technique so it can provide input for computing visualization before each iteration.

Anirban Ghosh - 3279100 - s6anghos@uni-bonn.de
Kajal Puri - 3305398 - s6kapuri@uni-bonn.de
Shilpa Chaterjee - 404788-shilpa.chatterjee@rwth-aachen.de

The A-tSNE uses the approximates this computation using Approximated K-Nearest Neighborhood (KNN) queries. This allows analysts to begin the iterative computation immediately after loading the data.

c) In the first case study (Allen Mouse Brain atlas), the authors propose to pre-process the data with PCA before running A-tSNE on it. Briefly explain why.

Answer. The A-tSNE graph shows small clusters in the embedding due to the values in the data. This was removed by using PCA, selecting the first 10 components as the high-dimensional space. Usually when high-dimensional data is modified, the tSNE embedding needs to be computed from scratch. Since we directly modify the high dimensional space, only the current embedding is modified.

Anirban Ghosh - 3279100 - s6anghos@uni-bonn.de
Kajal Puri - 3305398 - s6kapuri@uni-bonn.de
Shilpa Chaterjee - 404788-shilpa.chatterjee@rwth-aachen.de