Prof. Dr. Thomas Schultz
Mohammad Khatami (khatami@cs.uni-bonn.de)
Ikram Jumakulyyev (ijumakulyyev@cs.uni-bonn.de)

University of Bonn
B-IT/Computer Science II
May 15, 2020

Summer term 2020

Visual Data Analysis

Assignment Sheet 5

Solution has to be uploaded by May 25, 2020, 8:00 a.m.
to https://uni-bonn.sciebo.de/s/EIyXBwx0SksLr9n with password `vda.2020`

Please bundle the results (as PDF) and scripts (*.py/*.ipynb files) in a single ZIP file. Submit each solution only once, but include names and email addresses of all team members in the PDF and each script. Name the file `vda-2020-xx-names.zip`, where `xx` is the assignment sheet number, and `names` are your last names.

If you have questions concerning the exercises, please write to our mailing list:
vl-scivis@lists.iai.uni-bonn.de.

Exercise 1 (ISOMAP and t-SNE, *14 Points*)

In the previous assignment sheet, we used PCA for linear dimensionality reduction. Now, we will try out ISOMAP and t-SNE, compare them, and find out how their hyper parameters affect their results. You are free to use the implementations provided in the Python package scikit-learn.

a) Again read the `breast-cancer-wisconsin.xlsx` file from the previous sheet. Interpolate missing values as before, and keep all variables. We will now explore the results of t-SNE with different settings.
   - Run t-SNE with a random initial distribution of points and different perplexities, i.e., 5, 10, 20, 30, 40, and 50. Visualize the 2D data set in a scatter plot using different colors for cases from benign and malignant classes. (2P)
   - Repeat this experiment, except this time use PCA to create the initial distribution of points. Note that the implementation in scikit-learn allows you to select the initialization using a keyword parameter. (1P)
   - Compare the diagrams from random initialization to the ones with PCA initialization. For which settings did the 2D embeddings fail to nicely separate the two data clusters? Does it depend on perplexity, initialization, or both? Why? (3P)

b) We will now compare PCA to ISOMAP on a Mice Protein Expression Dataset, available as `Data_Cortex_Nuclear.xls` on our lecture webpage. It contains expression levels of 77 proteins, measured in the cerebral cortex of 8 classes of mice. The classes result from two genotypes (Ts65Dn, which serves as a mouse model of human down syndrome, vs. normal controls), two treatments (injection of the drug memantine vs. a saline solution as a control), and two experimental conditions related to context fear conditioning (context-shock, which should lead to learning, vs. shock-context, in which no learning takes place). Counting all repeated measurements, there are 1080 samples overall, some with missing data. You can find more information on the data in the corresponding publication.
   - Read the data set and interpolate missing values in a reasonable way. Extract subgroups c-SC-s and t-SC-s. How many mice were measured for each of these groups? (2P)

- Only for the mice from c-SC-s and t-SC-s classes, use PCA to reduce the 77 dimensional data set to two dimensions. Make sure not to include the remaining columns, which contain meta information. Visualize the 2D data set in a scatter plot using different colors for instances from each class. (2P)
- Produce a corresponding plot with ISOMAP. Try different values for the number of neighbors when constructing the neighborhood graph. Can you find a setting that separates the groups more clearly than PCA? (2P)
- Finally, try t-SNE with different settings. Does this allow you to separate the groups better than with PCA or ISOMAP? (2P)

Exercise 2 (A Sports Game Using Multidimensional Scaling, *6 Points*)

Two players A and B play the following game: Each of four ball sports (Basketball, Association Football, Handball, and Tennis) can be characterised by different numbers, such as the size of the ball, size of the field, the number of players, or the first year in which it was part of the Olympics. A and B each select such a number, compute (normalized) differences between them, and give the resulting dissimilarity matrix to their opponent. The other player has to recover the correct ranking and should guess what the underlying measure might have been.

You will find the two dissimilarity matrices on our lecture webpage, `dissMat1.xlsx` and `dissMat2.xlsx`. For each of them, please (manually) find a one-dimensional embedding that matches the given dissimilarities (2×2P). Please submit your results, along with a guess of what might have been the criterion in each of the two cases. (2P)

Exercise 3 (Shrinkage in Linear Discriminant Analysis, *9 Points*)

The algorithm for solving multi-class LDA that was introduced in the lecture is based on computing eigenvectors of the matrix $S_W^{-1} S_B$. This assumes that S_W is invertible.

a) When dealing with data from K classes in a p-dimensional space, what is the minimum number n of samples $\mathbf{x}_i \in \mathbb{R}^p$ for which S_W can be invertible? Briefly justify your answer. (3P)

b) Load the data in the file `LDA-input.csv` (from the lecture webpage) into Python. Use the implementation of `LinearDiscriminantAnalysis` from `sklearn` to perform an LDA based on the given class information. When you instantiate the `LinearDiscriminantAnalysis`, select `solver='eigen'` to use the algorithm described in the lecture. Submit your code and a plot of the result. (2P)

c) Repeat the analysis in b), but this time set `shrinkage='auto'` in addition to `solver='eigen'`. Submit the resulting plot. Do you prefer it over the one you obtained previously? Briefly justify your answer. (2P)

d) Setting `shrinkage='auto'` regularizes the estimation of S_W, which is useful especially when dealing with a limited number of samples in high dimension. For the given data, specify the condition number of S_W with and without shrinkage. What do you observe? (2P)

Exercise 4 (Pitfalls in t-SNE, *12 Points*)

t-SNE is a powerful technique for visualizing clusters in high-dimensional spaces, but correctly interpreting its results requires a good understanding of how it works. In this respect, the article "How to Use t-SNE Effectively", which can be found at https://distill.pub/2016/misread-tsne/, is a worthwhile read. It also includes several interactive experiments that you can try for yourself. Based on them, please answer the following questions:

- 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? (3P)
- 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? (3P)
- 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? (3P)
- 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. (3P)

Exercise 5 (Approximated tSNE, *9 Points*)

In an interactive data analysis, one might be willing to sacrifice some amount of accuracy if it substantially decreases computation times. To this end, Pezzotti et al. introduced approximated tSNE (A-tSNE) in the paper `pezzotti-a-tsne-2017.pdf` that you can find on the lecture webpage. Please read it and answer the following questions in your own words. As always, we will not grant even partial credit for copy-pasted text.

a) Briefly mention the steps involved in the tSNE algorithm. At which point does A-tSNE introduce an approximation? (3P)

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? (4P)

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. (2P)

Good Luck!