Data Mining and Matrices (FSS18)
Assignment 4: Spectral Clustering

Steffen Schmitz
University of Mannheim
stefschm@mail.uni-mannheim.de

1. CUTS

We start by computing the values of cut (cut), ratio cut (rcut) and normalized cut (ncut) for a given clustering of the $m = 500$ digits into $k = 10$ clusters.

1.c Magnitude of different cuts

Task. Run your cut function on the cluster.test clustering provided to you. Observe that cut $\gg$ rcut $\gg$ ncut. Why is this the case? Is this always true?

Considering a non-overlapping, complete clustering $C = \{C_1, \dots, C_K\}$ of the vertices in V. Define

$$\text{cut}(C) = \frac{1}{2}\sum_{k=1}^{K} W(C_k, V\backslash C_k) = \sum_{k=1}^{K} \frac{W(C_k, V\backslash C_k)}{2}$$

$$\text{rcut}(C) = \sum_{k=1}^{K} \frac{W(C_k, V\backslash C_k)}{|C_k|}$$

$$\text{ncut}(C) = \sum_{k=1}^{K} \frac{W(C_k, V\backslash C_k)}{\text{vol}(C_k)}.$$

Now, we can argue that

$$2 \ll |C_k| \ll \text{vol}(C_k) \tag{1}$$

should hold for almost all values $C_k \in C$. In our example all $|C_k|$ have a magnitude of 10^1 and all $\text{vol}(C_k)$ have a magnitude of 10^4, which means that the assumption in Equation 1 holds.

Looking at $\sum_{k=1}^{K} 1/|C_k|$ and $\sum_{k=1}^{K} 1/\text{vol}(C_k)$ we see that they take their minimum, if all $|C_k|$ or all $\text{vol}(C_k)$ coincide, respectively [1, cf.p.9]. On the other hand, this means that the sums are large, if the clusters are skewed. Then, the assumption in Equation (1) is violated and the cut function takes a high value for all three variants.

In our example, the different clusters have approximately equal size and this is also represented in the resulting cut values.

2. SIMILARITY GRAPHS

To construct a similarity graph, we make use of the Gaussian kernel (and vary parameter σ) as well as the various neighborhood graphs we discussed in the lecture.

2.a Varying Sigma

Task. Compute the full similarity graph using $\sigma = 50$. Study the distribution of the resulting similarities. Is $\sigma = 50$ a good choice? Try to find a good setting for σ by trying both smaller and larger values. Discuss!

In our notebook we try the following four settings: $\sigma \in \{10, 30, 50, 70\}$. We can directly see that a value of $\sigma = 10$ is far too small, because there is almost no similarity between any numbers. Furthermore, the histogram shows us that there are almost no values exceeding a similarity of 0.2.

For $\sigma = 30$ we get the matrix shown in Figure 1.

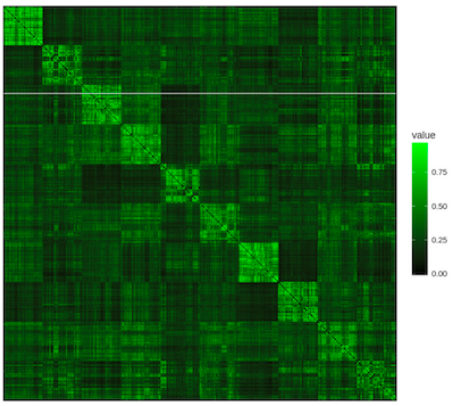

Figure 1: Matrix for $\sigma = 30$

The inherent structure of the matrix is clearly visible and we can directly identify the different clusters of numbers between which the similarity is large. The bell curve in the similarity histogram centers around 0.2 and most values are in the range from 0 to 0.5. For the moment, the resulting matrix seems like a useful result.

Next, we will look at $\sigma = 50$. The matrix is shown in Figure 2 and the corresponding histogram in Figure 3.
We can still distinguish the different clusters in the matrix, but all other regions are also brightly green. The histogram centers around 0.6, which tells us that the expected similarity between two vertices is rather high.
Compared to the resulting matrix for $\sigma = 30$, $\sigma = 50$ does not lead to an improvement. Thus, the calculated similarity between two randomly selected vertices is likely to be high, even if they are dissimilar.

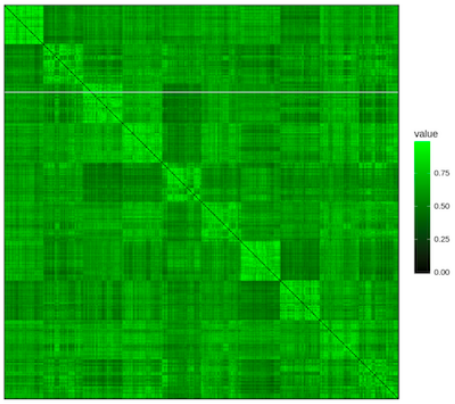

Figure 2: Matrix for $\sigma = 50$

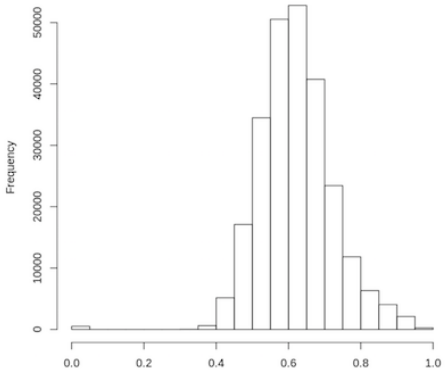

Figure 3: Histogram for $\sigma = 50$

Increasing the value for σ further, makes the result even worse as we can infer from Figure 4.

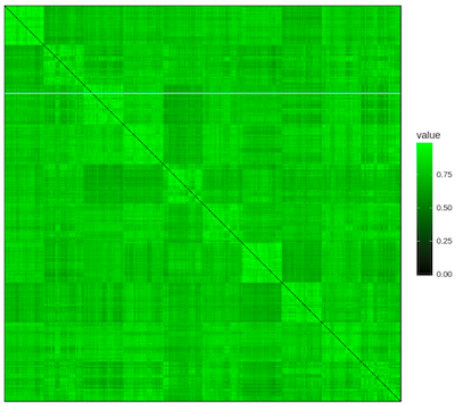

Figure 4: Histogram for $\sigma = 70$

Overall, the best result in our opinion provides the similarity matrix for $\sigma = 30$, as it clearly shows the clusters per number without assigning a high similarity to points that belong to different clusters.

2.b Neighborhood Models

Task. For $\sigma = 50$, find the smallest ϵ such that the ϵ-neighborhood graph is connected. Note that you can use the magnitudes of the smallest eigenvalues of the Laplacian to judge whether or not the graph is connected. Now find the smallest k such that the symmetric kNN graph is connected, and the smallest k such that the mutual kNN graph is connected. Plot the resulting similarity matrices. Are they different? If so, why? Discuss!

"A graph [...] is connected [...] [if] there is a path from any point to any other point in the graph" [2]. If we arrange all the eigenvalues of the Laplacian in non-increasing order, i.e.:

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{n-1} \geq \lambda_n = 0$$

we can apply the matrix-tree theorem and say that the graph is connected, iff $\lambda_{n-1} > 0$ [3, cf.p.3].

In our notebook we pick some parameters for ϵ and k and increase or decrease them until we find the lowest value for which λ_{n-1} is greater than zero.

The resulting similarity matrices are shown in Figure 5, 6 and 7. The ϵ-neighborhood and mutual kNN graphs look

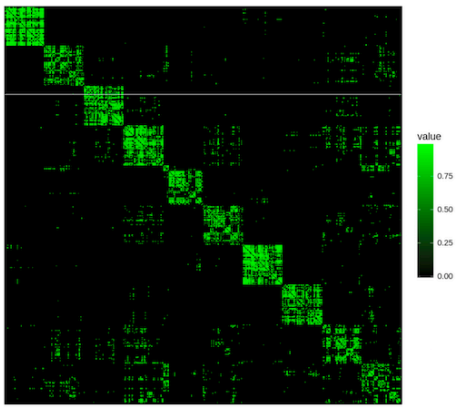

Figure 5: ϵ-neighborhood

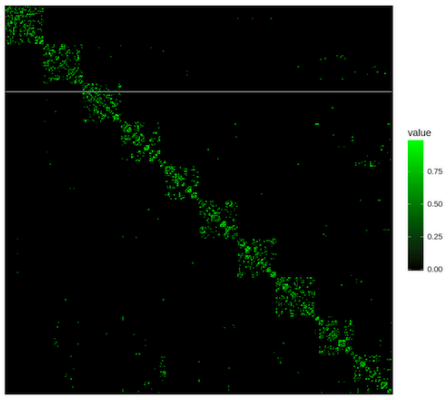

Figure 6: Symmetric kNN

similar and have approximately the same shapes and densities. Their plot contains ten obvious clusters, each representing one number, and some noise in the top right and bottom left.

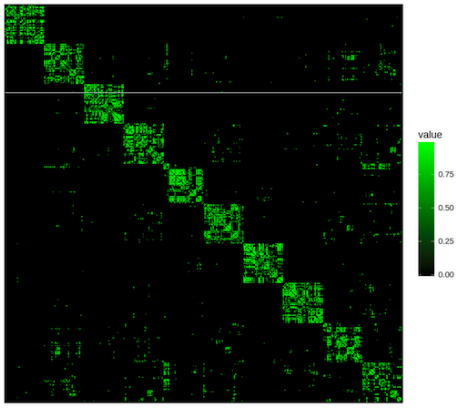

Figure 7: Mutual kNN

On the other hand, the symmetric kNN graph has very little green regions, except for the clusters we already recognized in the other two plots.

Our assumption is that the symmetric kNN graph is more likely to jump to other clusters and, therefore, connects the whole graph more quickly, while the mutual kNN and ϵ-neighborhood graph completely connect a cluster, before jumping to another one. This explains why the similarity matrices for mutual kNN and ϵ-neighborhood have a higher density around the actual clusters and why they show more noise.

2.c Parameters for Symmetric KNN

Task. For the symmetric k-nearest neighbor graph, manually determine values for σ and k that appear suitable to you.

In Section 2.a we found that $\sigma = 30$ is a good choice and a general recommendation for a first pick of k is $\log(n)$, which is 2.69 or roughly 3 in our case. Using this as our first pick we obtain the similarity matrix in Figure 8. From the eigen-

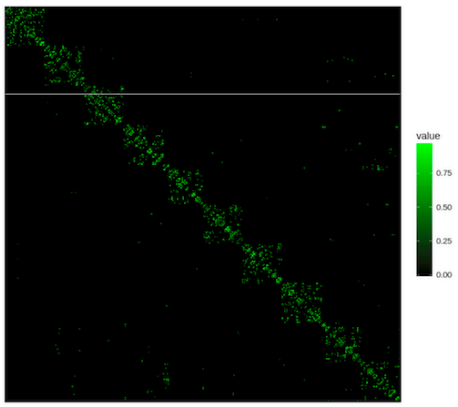

Figure 8: Symmetric kNN with s=30 and k=3

values of the Laplacian we can directly see that the graph is not connected, because λ_{n-1} is zero. The resulting matrix looks even more sparse than the parameter configuration that we used for Figure 6.

"In general, if the similarity graph contains more connected components than the number of clusters we ask the algorithm to detect, then spectral clustering will trivially return connected components as clusters. Unless one is perfectly sure that those connected components are the correct clusters, one should make sure that the similarity graph is connected" [1, p.21]. Although we are convinced that each connected component belongs into a cluster, we want to be absolutely sure. Following the insights from von Luxburg, we, therefore, should aim for a connected graph that is more densely populated than the previously computed ones.

We will now try to increase the number of clusters until the graph is fully connected and find that $k = 4$ leads to a connected graph and that the graph looks even more similar to the one in Figure 6. Increasing σ to 80 also results in a similar matrix and the graph is still connected.

Hence, we will continue to increase k, while keeping $\sigma = 30$, because an increasement of σ had no visible effect. With increasing k the resulting matrix approximates the results that we got for ϵ-neighborhood and mutual kNN in Section 2.b. The clusters become more dense, but the noise also increases. In our opinion, $k = 25$ and $\sigma = 30$ is a good parameter combination that captures the clusters well, while keeping the noise low. The result is shown in Figure 9.

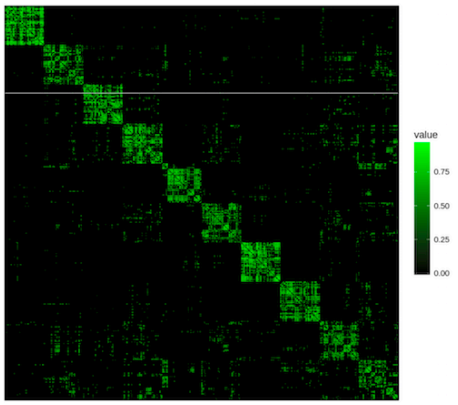

Figure 9: Symmetric kNN with s=30 and k=25

2.d Influence of sigma

Task. Consider any dataset in Euclidean space. Suppose that we use the Gaussian kernel with parameter σ to obtain similarities and subsequently construct a symmetric k-nearest neighbor graph. Describe what changes to expect in the so-obtained graph when we increase or decrease σ. Is there anything that does not change?

The Gaussian kernel is defined as

$$w_{ij} = \exp\left(\frac{-\delta_{ij}^2}{2\sigma^2}\right)$$

and the parameter σ controls what is considered local. This function assigns a value of 1 if the distance is 0 and approximates 0 asymptotically for even larger values. We can control the steepness of the decline with σ.

The Gaussian kernel function is approximately linear around its inflection point and flattens out at the edges. We assume that a change in the value of σ mostly affects points that are very close.

We suppose that x_1 and x_2 are part of a dataset in Eucledian space with x_0. If both are far away, both weights w_{01} and w_{02} will be small for any σ and negligible, because they are unlikely to be neighbors. In the case that the distance between x_1 and x_2 is around the inflection point for any σ the resulting weight will reflect the distance between them and x_0 almost linearly and, therefore, will not have a big influence. We only recognize a significant influence of σ on the similarity, if both points are very close to x_0. In this case, they will seem far apart for small σs and the distance is amplified due to the steep decline. Alternatively, they may appear closer together for large values of σ, because the Gaussian kernel is approximately flat for small distances.

With symmetric kNN it is guaranteed that each vertex has at least k neighbors. We assume that we have far more connections for a large σ, because nearby matrices seem even closer due to the flat behavior of the kernel function. On the other hand, we expect the number of neighbors to drop for very small σ. The only thing that does not change is the similarity that we expect for vertices that are far apart. We should always assign a small similarity value to them.

3. SPECTRAL CLUSTERING

In this experiment, we try to cluster the digit data into 10 different clusters; the "optimal" clustering assigns the same digits to the same cluster, and different digits to different clusters.

3.a k-means

Task. First cluster the digits data using k-means on both the raw data and the first 10 principal component scores. Visualize the result and compute the accuracy. Are the results good? Which "errors" are made?

Running k-means on the raw input data with $k = 10$ clusters results in an accuracy of 82.4%. From the confusion matrix we see that many 9s are incorrectly labeled as 3s, 1s are often confused for 8s and k-means often predicts a 4 for a 9. Although the input dataset is balanced, k-means labels more than 80 images as 3 and less than 40 images as either 1 or 4.

k-means works best, if all clusters are of globular shape and have a strict separation. Yet, the results indicate that this is not the case for the digits dataset. Especially the clusters for 9 and 3, 1 and 8 and 4 and 9 seem to overlap.

We observe the same behavior for the k-means clustering on the first 10 PCA scores. The accuracy is down to 80% and the confusion matrix is similar to the previous one. We conclude that the effect of overlapping clusters is only amplified by the reduction to ten dimensions.

3.b First run of Spectral Clustering

Task. Use your parameter settings of task 2c) and run spectral clustering. Compare the result with the results obtained above. Which method worked best? Did your parameter settings produce good results? Which "errors" are made?

Using spectral clustering we obtain an accuracy of 83.6% and, therefore, see an improvement on both of the results we obtained in the previous subsection. The combinations of problematic pairs that we identified in Section 3.a are critical, again. Nevertheless, the improvement shows that spectral clustering is an improvement over k-means on the raw data.

Looking again at Figure 9, we also see that most of the noise is in the 4th, 9th and 10th row. This may indicate that those regions still are very similar and that the graph is not sparse enough. Hence, we draw the conclusion that our selection for k was too big and that we should try a smaller k for our optimization attempt that follows in Section 3.d.

3.c Eigengap Heuristic

Task. In practice, we may not know the optimal number of clusters. Use the eigengap heuristic to estimate a good choice for the number of clusters. Discuss!

For this task, we will again use the parameter combination from Section 2.c, $k = 25$ and $\sigma = 30$. The 100 smallest eigenvalues are plotted in Figure 10. According to the eigengap-

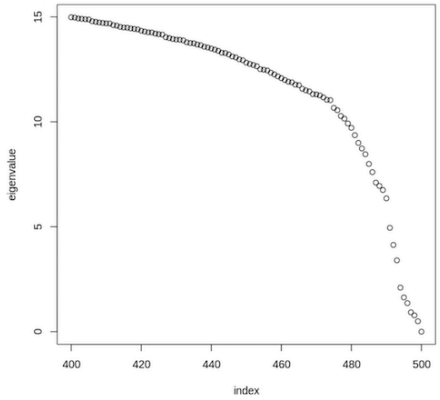

Figure 10: Last 100 eigenvalues

heuristic we should select a number of clusters k such that $\lambda_1, \ldots, \lambda_{n-k}$ are large and $\lambda_{n-k+1}, \ldots, \lambda_n$ are small. In the eigenvalue chart, we see that there is a gap after the last 7 and after the last 10 eigenvalues.

Taking the first larger gap as our guidance, we would select $k = 7$ for the clustering algorithm. This also fits our observation from the previous subsection that 9 and 3, 1 and 8 and 4 and 9 seem to overlap. Applying the eigengap heuristic, we effectively find about seven clusters with the current parameter setting.

3.d Parameter Tuning

Task. Now "tune" the parameters of spectral clustering with 10 clusters to obtain an accuracy above 0.88. Why do you think that the so-obtained parameters work well?

First, we will follow our intuition from Section 3.b and decrease the number of k that we use for our symmetric kNN graph. We already found that the graph is connected for $k = 4$ and while trying values around $k = 4$, we find that $k = 5$ yields a good result with an accuracy of 89.6%. Hence, we draw the conclusion that spectral clustering works best if the graph is very sparse, but connected.

Starting from $\sigma = 30$ we first go up to 50, 100 and 200 without seeing any change in the accuracy and afterwards down to 5 and 10. For $\sigma = 5$ the accuracy is very low and an indication that σ is too small to reflect the similarities accurately. With this parameter setting every point seems dissimilar to every other. On the other hand, we improve our accuracy to 91.4% with $\sigma = 10$. We assume that this parameter setting reflects the locality in the graph in the most accurate way. Lower values of σ lead to a low similarity for too many points, while a higher value of σ probably returns a high similarity for points, even if they belong into other, nearby clusters.

The resulting matrix with our optimized parameters is shown in Figure 11. Surprisingly, this matrix is very sparse. This

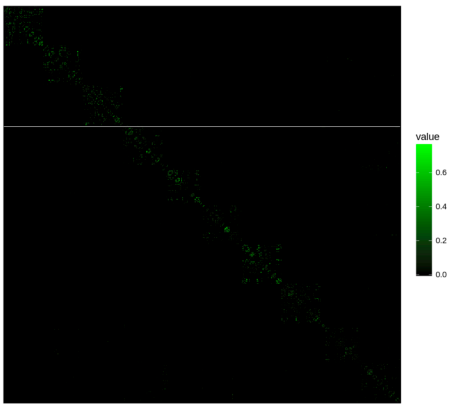

Figure 11: Symmetric kNN with s=10 and k=5

shows that our assumptions from Section 2.a are not correct and it is an advantage if the clusters are not densely connected.

Overall, a small k that connects the full graph, while maintaining sparseness, seems to be a good choice and σ should be selected in such a way that it allows neighbors be close, while returning a low similarity for points that are further away.

4. REFERENCES

[1] U. von Luxburg. A tutorial on spectral clustering. *CoRR*, abs/0711.0189, 2007.

[2] E. W. Weisssteín. Connected graph. From MathWorld—A Wolfram Web Resource. Last visited on 12/5/2018.

[3] X.-D. Zang. The Laplacian eigenvalues of graphs: a survey. *ArXiv e-prints*, Nov. 2011.