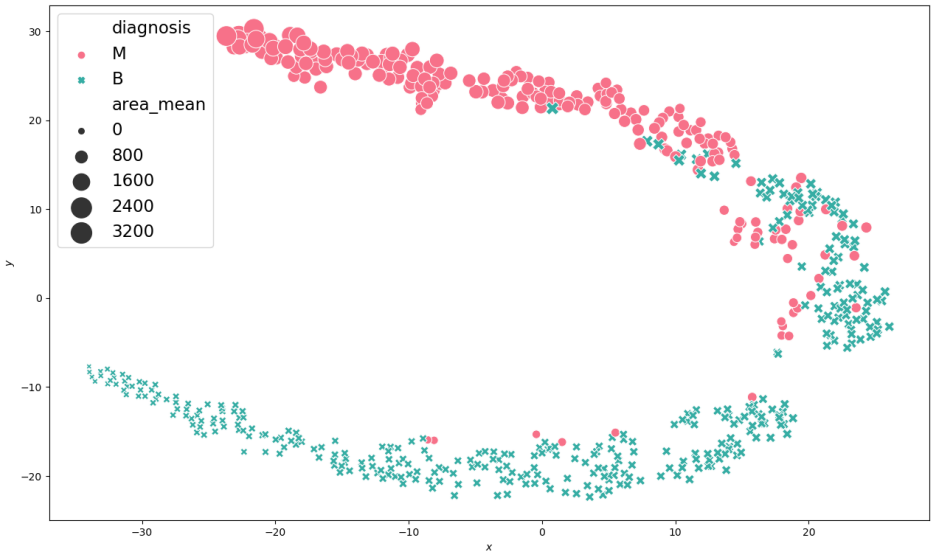
UNIVERSITÄT ZU LÜBECK - INSTITUTE OF MEDICAL INFORMATICS

Report
Population Medicine - Clustering

Jesse Kruse - 675710
Konrad von Kügelgen - 676609
Falco Lentzsch - 685454

February 10, 2020



General idea of Scenario 3

In scenario 3 the given task was to cluster data to distinguish between malignant and benignant breast tumors. Breast cancer is the most common cancer among women in Germany. In the last decades the amount of affected people are constantly increasing, therefore also increasing risk and death rates. Recognising breast cancer in an early state is crucial for the outcome of the patient. Easy and cost efficient methods to do this are valuable. Being able to do this automatically is what motivates the idea of scenario 3. To achieve this we first need to recognize patterns in our data that could predict cancer. In vast amounts of unlabeled data we could try to structure the data first. One approach is to cluster the data automatically, the key task here. In general that means to partition the data in a given or computed amount of subsets of most equal data points and optimally be able to identify features that could predict a disease like breast cancer. So in the following part we describe the used algorithms and techniques.

1 K-means

K-Means is a clustering algorithm. Ideally, it separates the data into distinguishable clusters that share certain similarities within a cluster. The k in the name is a parameter and describes the amount of clusters that should be determined by the algorithm. We are dealing with a batch learning algorithm which operates on a clear data set. "Clear" in this context means that all data point should have definite values/coordinates upon starting the algorithm. K-Means-Clustering runs in iterations and follows certain steps.

1. Choose k cluster centers (e.g. randomly from existing data points)
2. Assign each point the closest cluster center (using a distance function)
3. Determine new cluster centers by calculating the "middle" of all assigned data points
4. Stop if iterations reach a set threshold or if no cluster centers change (compared to the previous round)

The K-Means-Clustering will go through steps $2, 3 and 4$ in each iteration until the end (defined in 4) is reached.
In this exercise we used the implemented KMeans from the package *sklearn.cluster*. To get familiar with the usage we started with random data which the KMeans was fitted to. Later on, in the second part of this scenario, we applied the K-Means-Clustering to breast cancer data. The description will follow after the *BSAS* section.

Elbow plot

The elbow plot is a method to determine the amount of clusters to be used in a clustering algorithm. It is said that this method is not the most reliable as it is quite ambiguous. Different approaches like the silhouette methods are more common. Though, in this scenario we used and implemented the elbow method. In Figure 1.1 you can see one of the before mentioned plots. The y-axis shows the total distortion which represents the average distance between each point and its cluster center. (Here, the *Euclidean distance* was used). The x-axis shows the amount of cluster centers used. In the plot we can see, that from $k = 3$ upwards there is not much of a change in distortion. The ideal k for this data set would be 3. (Which is at the elbow of the plot, if one imagines the line to be an arm) This way we will probably receive a good separation/clustering of the data instead of having too many clusters. Adding more clusters will not result in a beneficial gain of information, as the distortion does not vary much from $k = 3$ upwards.

This method calculated the outcome of K-Means for 9 different values of k. This method obviously is harder to perform if the data set is very large and the amount of k's that should be tested is high as well.

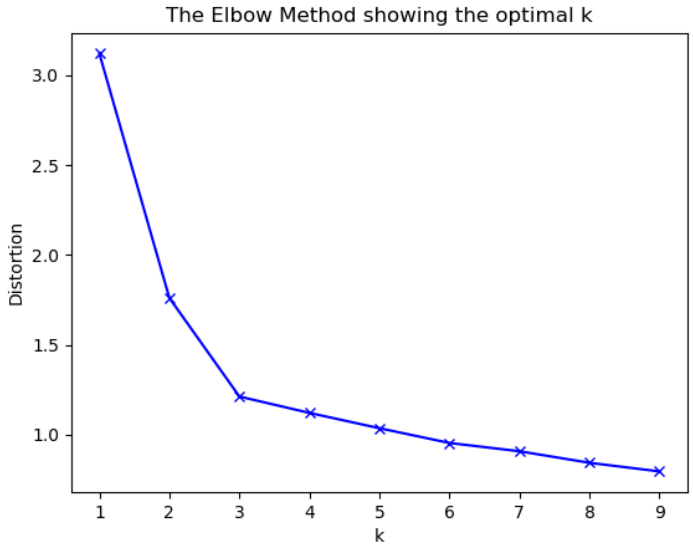


Figure 1.1: Elbow plot with 9 different k's

2 BSAS

In the first session of scenario 3 we also had to implement the **Basic Sequential Algorithmic Scheme (BSAS)**. The input is a data set, in our case an array of 2D points, Theta, a distance and q, the maximal number of clusters. If the distance between a point an a cluster center (cl) is bigger then Theta the algorithm uses a new cl for that point as long as the amount of the clusters has not already reached q. So every point of the data set is presented to the algorithm one by one. In our implementation we used the first point for the first cl. We stored the indexes of the chosen cl for each data point in an additional array of the length equal to the amount of points in the input data. Afterwards we iterate over our input array and for each point we compute the distance to every cl and take the one with the minimal distance. If the distance is smaller than Theta or the amount of cl is equal to q we assign the point to this cl. If this distance (to the closest cluster) is bigger than Theta and the amount of existing cluster center is less than q we will create a new cl center - the center being the current point. When the point is assigned to an existing cl we have to update the coordinates of that cl. This is done by the formula from the lecture:

$$m_{c_k}^{new} = \frac{((n_{c_k}^{new} - 1) * m_{c_k}^{old} + x)}{(n_{c_k}^{new})} \tag{2.1}$$

$m_{c_k}^{new}$ are the coordinates of the new cl. To compute it without having to compute every distance again we first take the old amount of assigned points $(n_{c_k}^{new} - 1)$ as a weight for the old cl $m_{c_k}^{old}$ because it already represents $(n_{c_k}^{new} - 1)$ of the points in the cluster. Then we add the coordinates of our new point x and divide the whole term by the new amount of assigned points $n_{c_k}^{new}$ to have the new mean vector.

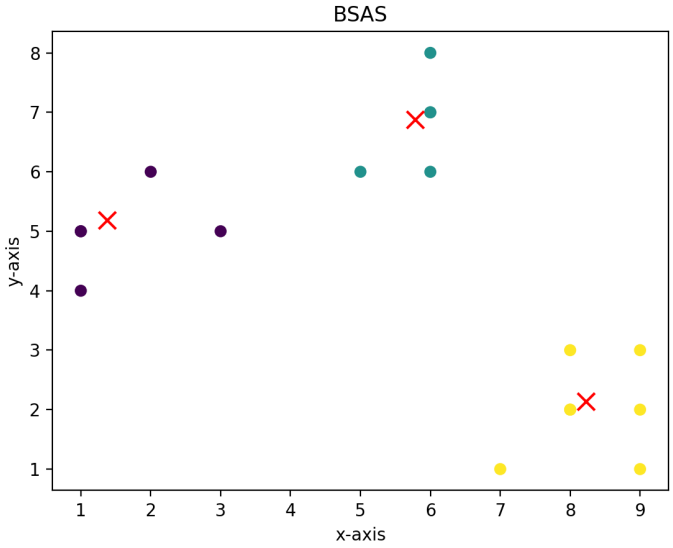


Figure 2.1: BSAS Plot

3 Operating on breast cancer data

Data set description

For the next part of this scenario we received breast cancer data from a machine learning database. This data is comprised of 32 columns which represent the features and 569 rows which shows the values for each column. Important is the second column and its value. The second column is the diagnosis, being either *benignant* (*B*) or *malignant* (*M*). The first column is the ID and the rest of the columns represent features describing the breast cancer tumor. Examples are *smoothness*, *concavity*, *texture* and *radius* which are all float values.

3.1 Pair plot

To get a first visual impression of the data one powerful tool is the pair plot provided by *seaborn*. It plots the different dimensions of a given data frame against each other to reveal potential correlations between them. The pair plot makes use of the scatter plot and histogram. On the diagonal where each feature is plotted "against itself" we can see the distribution of a single variable in a histogram. The rest of the plots are scatter plots visualizing how the features look when plotted against every other feature. Because the plots are ordered in a matrix, the plots are mirrored along the diagonal, apart from swapped axes. We can also alter the hue of the plots by one feature. In our case, we used the *diagnosis* to see weather malignant and benignant tumors are distributed differently. In fact in the data one can discover some patterns. In the plot where the perimeter mean and the area mean are plotted against each other we see a nearly linear increase which seems logical because with an increase of the area the perimeter also has to increase. But if we take a look at the hue we see that with an area higher than 1000 and a perimeter more than 130 nearly all the tumors are malignant. So these two plus the concavity mean could be useful to distinguish between malignant and benignant. Perhaps not so great are smoothness and symmetry. Here the histograms show that the distributions of the two diagnosis are overlapping heavily and also plotted against each other they show a high overlap of both of them.

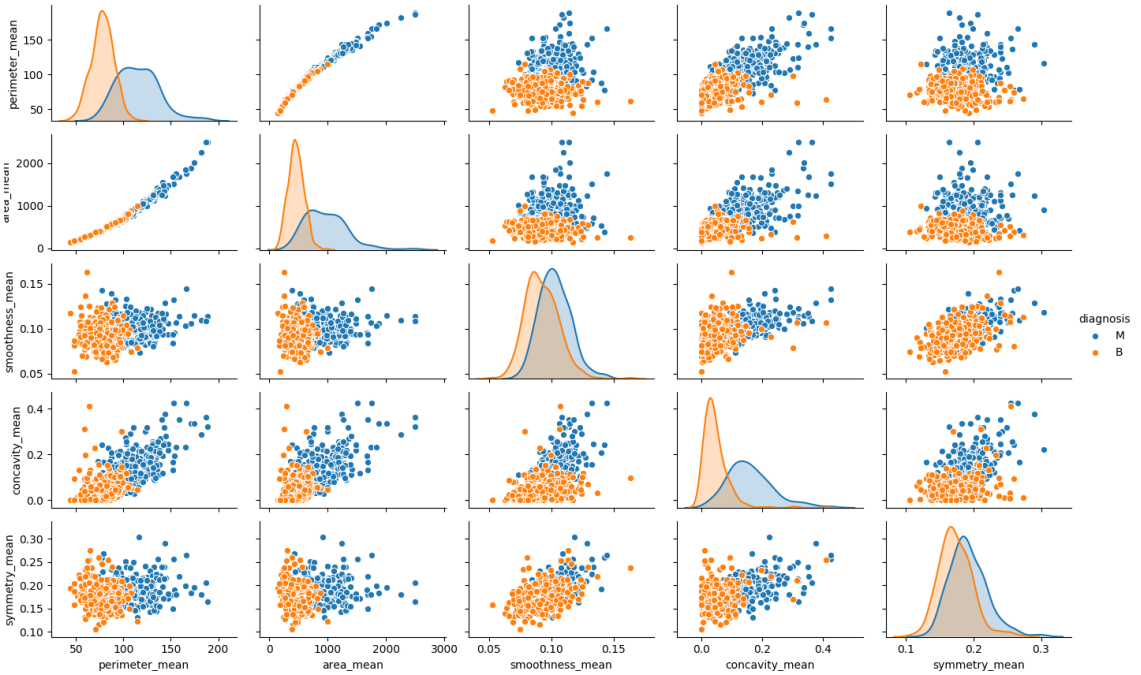


Figure 3.1: pair plot of five features

3.2 Projection by t-SNE

To be able to visualize the multidimensional data we used the ***t-distributed Stochastic Neighbor Embedding*** tool. Like the *Principal Component Analysis* which we mentioned in the previous scenario this tool aims to reduce the dimensionality of data. The t-SNE projects multidimensional objects which share similarities on a two-dimensional point which then represents certain multidimensional objects. With further algorithms, probabilistic distributions and computational methods, whose explanation would go beyond the scope of this project, the multidimensional data is projected to an X- and a Y-coordinate. We are now able to visualize the data as seen in the title figure. In that figure we see redish/pink points (malignant) and turquoise crosses (benignant) as tumor representations varying in size. The bigger the visual data point appears the greater is (in this case) the mean area of the tumor. We can see a trend of increasing size of the the benignant tumor from left to right and in the same direction a decreasing size of the malignant tumors. At the very right section of the figure pink and turquoise points mix in with each other. We can say, that bigger benignant tumors rather get mistaken with malignant tumors than if they are smaller. (By this t-SNE dimensionality reduction!)

What can we take from this representation?

It is easy to see that there is some kind of cluster separation. Surely one could draw a horizontal line at a height of $y = 0$ which would separate the clusters quite well - though not ideal. How many false classifications do we allow?

By plotting the data with different features one can estimate the effect of a clustering by that particular feature.

3.3 Multidimensional K-Means

For clustering of the data, we used the multidimensional dataset and fitted the K-Means classificator to the data. We used $k = 2$ and $k = 8$.

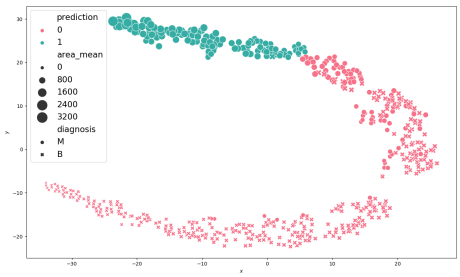


Figure 3.2: K-Means with 2 centroids

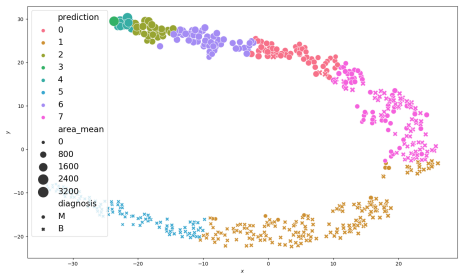


Figure 3.3: K-Means with 8 centroids

3.4 Metrics

Evaluation metrics can identify the performance of a classifier. We used the *completeness score*, the *homogeneity score* as well as a heatmap. The ground truth was always given by the actual diagnosis. B is mapped to 0, M to 1. The prediction was made by a K-Means calssificator initialized with two centroids. The values of the two scores each can lie between 0 and 1. Homogeneity is given (value is 1) when all points/objects of one cluster are members of the same class. $(B, M \vee 0, 1)$ Completeness is given (value is 1) when all points/objects of one class are in one cluster. The heatmap is a visualization of a confusion matrix which was explained in the previous report.