[Music]

Welcome to the fifth lecture, lecture of the fourth week of the course in machine learning. In this lecture we will talk about cluster analysis. so the agenda for the lecture next is as follows, very will first talk about caster analysis in general, we say a few words about hyper parameters for this kind of algorithms, talk about distance measures and then we will choose one formal categorization of clustering algorithms, actually five categories Partitioning based, Hierarchical-based, Density based, Grid based and Model based. So there will be a short discussion about each of these categories. So cluster analysis is an important element in unsupervised concept learning, this means learning on multiple concepts from unsorted examples. Apart from being an important methodology for pre-processing of data sets in this unsupervised machine learning scenario, cluster analysis can be used as a standalone technique for particular categorization purposes. As instance are not classified in the unsupervised scenario, algorithms have to identify commonalities and structures in the data set and to group teams as a based on similarity. Then when we found these groups. the detailed concept formation can continue but then typically by using any of the techniques for supervised learning as described earlier in actually as for scenario one to ten as one of the earlier lectures in this course.

Cluster analysis has many synonyms like clustering, conceptual clustering, Clustering techniques, clustering methods etc. Cluster analysis is the task grouping a set of objects in such a way that objects in the same group all the cluster are more similar in some sense to each other than those in other groups. Cluster analysis can be achieved by various algorithms that differ significantly in our understanding of what constitutes a cluster and how to efficiently find them. There possibly over 100 published clustering algorithms. Typically clustering algorithms are dependent on several hyper parameter settings, as for all machine learning techniques. Potentially these parameter settings that can also be automated based on separate learning processes. The default is to preset them, but they can also be learned theoretically.

So let's look at a few examples of hyper parameters that may need to be specified for clustering algorithms. So of course one important parameter is the number of clusters. Several algorithms need that to be predefined. Also the number of features you to describe the instances is of course important that that's of course general to all machine learning algorithms. Type of distance measures to employ is a crucial thing. For certain measures density is a concept so and one approach there is to set up threshold for maximum distance

between instances as a criteria for being dense and also the number of instances that have to testify such a density threshold. But they can also be alternative density special measures also Also rather general for most machine learning algorithms many times you need to specify have how many sessions for expression of the data set that you would have.

Distance matrix have been described in the lecture on is the instance based learning, but as also a very crucial role in cluster analysis. A distance metric or measure or function is typically a real-valued function that quantifies the distance between the two objects. Also said in the earlier lecture but I would like to repeat it here again is that distance metric and the similarity metrics have been developed more or less independently for different purposes, but usually specific similarity measures are intuitively inverses of corresponding distances metrics and can therefore be transformed into each other. So we're then in the earlier lecture also exemplified with two categories of matrices that are pretty common, so either there are matrices in a long Euclidean vector space like the Minkovsky distance and it's its specialization, Manehattan, Euclidean and Chebyshev distances, you can also have the Cosine measure. But we can also matrices that can work on overlapping elements in other kind of data representation not necessarily feature vectors. So they have a Levenshtein distance for text, we have the Jaccard similarity for set, so we have the Hamming distances for binary strings.

So as there are so many as 100 or more published clustering algorithm, these clustering algorithms can themselves be clustered in many ways and you find in the literature very many ways of doing. So for this lecture are just choosing one of these ways of categorizing the clustering algorithms, and the categories I've chosen as most natural are partitioning based, hierarchical based, density based, grid based and model based. And as you can understand all these hundreds of clustering algorithms can then be associated with any of these categories. Many times there are borderline cases, so sometimes one kind of algorithm could depending on perspective be viewed as one category or the other, but more or less I would say that the well-known algorithms at least those I know could be reasonably well classified according to this model. Let's start with partitioning based clustering.

So partitioning algorithms are clustering techniques that subdivide the datasets into a set of k clusters. And this number k for this kind of a normally have to be preset. A majority of partitioning algorithms are based on a selection of prototypical instances or synonymously centroid instances. This algorithm may be termed Centroid clustering technique, so can say it's a sub category. In this approach the selection of Centroids are iterative optimized and instances are iteratively reallocated to the closest centroid to ultimately form the resulting

clusters. The result can be illustrated in a positioning of the data space or as we did for another purpose in instance spaced learning in a formal Voronoi diagram, you can see one such thing today. So properties of this argument, the target number of clusters really is k need to be preset and of course the setting of that parameter is it's very important. Also the initial seeds for these means or centroids have a strong impact on the outcome of the algorithm. There is some evidence that that partitioning may perform better than others like hierarchical approaches that we discussed later because the clusters they produce are kind of have a better fir or a tighter than clusters produced by other methods. There are series of algorithms of this class but the most important and also oldest approach is K-means clustering and we will discuss this in little more detail.

Partitioning based clustering is here exemplified by the approach in the k-means algorithm and actually k-means algorithm is also an instance done of the subcategory centroid based clustering. So the goal is to partition and instances into k clusters. So I'll start with selecting k instances out of the instances in the dataset and allocate these as the initial means or centroids or prototypes, terminology varies and there are some motivation for all these terms. The distance is calculated between each of these means or centroids to every other instance. Typically in the k-means algorithm or originally at least (10:24), Euclidean distance is applied. Then the next step when that has been done, you associate all the instances to the closest means according to the closes me according to the calculation. And what you got then is a division of the total set in subsets and we let these subsets that you get a base form (11:02) constitute the initial clusters.

So this you can see below in the example because this partition first purchasing of the data set (11:13) in subsets. What happens then is that we really find the means and now you can understand why we also use the term Centroid it's because in mathematics our is a way of taking a lot of points in a space and calculating the most central concepts based on a central object based on all the data points. So this is now done so if we take all the instances that happen to be in one of the original cluster and calculate from that set the centroid of those according to traditional definitions of centroid. So this means you get a new mean or new central point for each of the initial clusters, but then you also for all instances recalculate the distance from all of the instances to the new the newly computed centroids and this means that in this step it can happen that the instances change cluster membership which of course is a nice feature of this because in a way we can revoke earlier less optimal choices. So then all of this is iterated many times until we reach a stability in the sense that the centroids do not move between these iterations and then the algorithm typically stops. Let's now turn to the

second type of category of clustering techniques, Hierarchical based clustering or Hierarchical clustering techniques. So this is a kind of technique which seeks to build a hierarchy of clusters not only one layer of clusters but rather what was earlier mentioned in some uncommon mixtures like taxonomy so the results of the hierarchical clustering process is usually presented in what called the dendogram, and you can see to the upper right here on the slide you can see a dendrogram that is related to an original set of instances depicted in the claim, and as you can hopefully can infer the structure of the dendogram is based on the proximity of distance but between the instances and in the claim in the example.

So properties of hierarchical clustering it does not assume a particular value of k as needed by by the k-means clustering so you don't need to specify that. The generated tree may correspond to meaningful taxonomy concept hierarchy. So from a domain modelling point of view and it's a nice technique. What you need is a distance matrix to compute the clustering steps, or let's say you need two kinds of matrices actually you need a distance matrix between instances and you need what's called a proximity matrix between the clusters but we will come back to that in the next slide. Initial seeds have a strong impact on the final results as assignments cannot be done iteratively. So in a way this is agreements more sensitive to the kind of local decisions taking during the process. Actually also this technique is very sensitive to outliers it's another property.

This sides give you more detail very detail actually example of a dendrogram that kind of diagram that shows the hierarchical relationships between objects. Actually a dendrogram is a taxonomy, it's a graphical version or form of a taxonomy is very common that hierarchical cluster can output this kind of structures and of course the role of the diagram is to work out the best way to allocate objects to clusters. Let's look a little closer to hierarchical based clustering this kind of clustering can proceed in two ways, Agglomerative fashion which is a bottom-up approach where each other observation starts in its own cluster, so exactly in the beginning you have as many clusters as you have instances. And then through the process parents of cluster emerged as one moves up the diagram. Division fashion where is a top-down approach where all you start with all observations or instantly one cluster and then you split that cluster recursively as you move down the hierarchy. And splits and merge that typically perform based on a proximity matrix between clusters and actually here is important to understand that there are two concept of distance, here so first you have the basic distance between instances. So you can say for this kind of system and you start with an distance matrix between the instances but then it's they have it in in this kind of approach that you want to abstract a little so therefore you want to calculate also not only the distances between

in the instances but somehow the proximity or distance between clusters. It's actually very simple relation because if you have distance matrix given all the distances between the instances involve, then you can create a proximity matrix which is the term used between the clusters by taking for each cluster you take the average the two clusters you want to compare, you take the average of the distances between the distances in two clusters. So actually the proximity matrix is an abstraction based on the distance matrix. The proximity matrix is then really calculated in each step of the algorithm because the distance doesn't change, so the distance matrix is the same but because the during the process the candidate clusters shift we need to calculate the proximity matrix in every step for each new setup of candidate clusters. In general all the steps in this algorithm the merges and splits are determined in a greedy manner which has the effect that you can have the risk of not reaching a global maximum we have discussed this earlier all techniques that are greedy that take local decisions that cannot be revoked or reconsidered and then there is this risk, So to the to the right you can see an simple examples where you have a number of instances but these instances are then grouped into these five clusters and then you can see that the proximity matrix is then a matrix with them with the clusters as rows and columns and you have this calculated value of proximity based on the instance distances the elements of that matrix.

Now we turn to the third category of clustering techniques. To start with density based clustering. So density based clustering is a clustering to be which group together instances are closely packed together. Instance with many nearby neighbors, marking as outliers instances that lie in a low-density regions whose nearest neighbors are far away. So properties of this kind of algorithms are, that clusters are dense regions in the instance-based separated by regions of lower instance density. A cluster is defined as a set of connected instances with maximal density. So this kind of algorithm does not need a predefined target value for number of clusters but needs definitions for distances for each ability and density. It can discover clusters of arbitrary shape and it's pretty insensitive to noise, so there are many advantages here. We will describe this technique little more concretely in terms of the approach taken in one of these algorithms called DBSCAN.

So let's now go through one of these algorithms called the DBSCAN, the density based approach. So the instances are classified in this case as core instances, reachable instances and outlier. A core instance has a minimum number of instances within a threshold radius. An instance is density reachable from another instance if it's within a threshold radius from the core instance. An instance is density connected to another instance if both instance are density reachable from a third instance or if they are directly density reachable from each

other. All instance is not reachable from any other instance are considered outliers could possibly be noise. If *p* is a core instance then it forms a cluster together with all instances that are reachable from it. Each cluster contains at least one core instance; non-core point can be part of a cluster but they form its edge. All points within the cluster of mutually density connected. If a point is density reachable from any other point in the cluster is part of the cluster as well. So here we can see that we built up at the clusters through the core instances and then through the density reachable property. And the two important parameters here is of course the threshold radius but also the minimum number of instances that must be within the threshold radius in order for an instance to be counted as core. So in the example to the right you can see that the red instances our core instances because the area surrounding is in a epsilon radius, the threshold radius contain a specified number minimum of four points in this case. Because they are all reachable from one another they form a single cluster B and C are not core points but are in table form A and that's belong to the cluster is well. At the point N is it is a point that is neither a core point nor directly reachable and it's not therefore considered a part of the cluster.

Finally we will look a little more closely at something called grid based clustering. So grid based clustering, you quantize the instance base into a finite number of cells hyper rectangles and perform the required operation on the quantized space. So typical steps in this kind of algorithm is to find a set of grid cells, assign instances to the grid cell and compute densities of the cells. Eliminate cells that have densities below a certain threshold from clusters from adjacent cells based upon some objective optimization function. So actually if I could say that grid based clustering is density clustering of a certain kind also with this introduction of this kind of grid structure.

So finally I want to say something about model-based clustering even though it doesn't really fit into this week because as you know the title of the whole week was inductive learning in the press is in the presence of weak or absent domain theories. So model-based clustering is actually a clustering technique where we have some models or background knowledge or theory about the domain from which from the domain for which the instances of the datasets is harvested. So this model can more or less extensive back in in all cases to some extent to guide the clustering process and the basic clustering process that we start from can in principle will be any extension to one of the other clustering approaches we discussed earlier here. If the domain knowledge is some statistical information about the distributions where as kinds of instances involved one can call this kind of clustering technique distribution based learning as a category of its own. So example of a distribution based clustering scenario

where you have sample instances that the arise from a distribution that is a mixture of two more components. And two components means of course two types two types of instances. So each type of instances is described by density function and has an associated probability or weight in the mixture. The mixture is of course the law the hole the H rainiest outer set with these kinds of instances (27:01) . In principle we can adopt any probability model for the components per typically we will assume that they are p-variate normal distributions. And of course what we end up with here is that each component in the mixture becomes what we call a cluster. So this was the end of this lecture on cluster analysis thanks for your attention. The next lecture and the last one for this week will be the tutorial regarding the assignments for the week thank you bye.