**Overview about Machine Learning (ISL chapter 1,10):**

Statistical learning, is the whole set of tools, helps in understanding data. These tools can be categorized to supervised, and unsupervised. Supervised statistical learning, uses given data (inputs), and predict, estimate their output, this method is broadly used. However, this project will primarily focus on supervised statistical learning, which is recognised as having variety of inputs, without having their corresponding supervised outputs.

Through unsupervised learning, by inspecting each element characteristics, we can reach the capacity of grouping similar data, within a certain dataset, to the same cluster(group). Its primary goal, is to discover significant aspects about the measurements. Principle Components Analysis, a type of unsupervised learning, is a useful tool for visualising data, or pre-processing. Moreover, Clustering, is another valuable technique, that simplifies the process of determining subgroups in dataset.

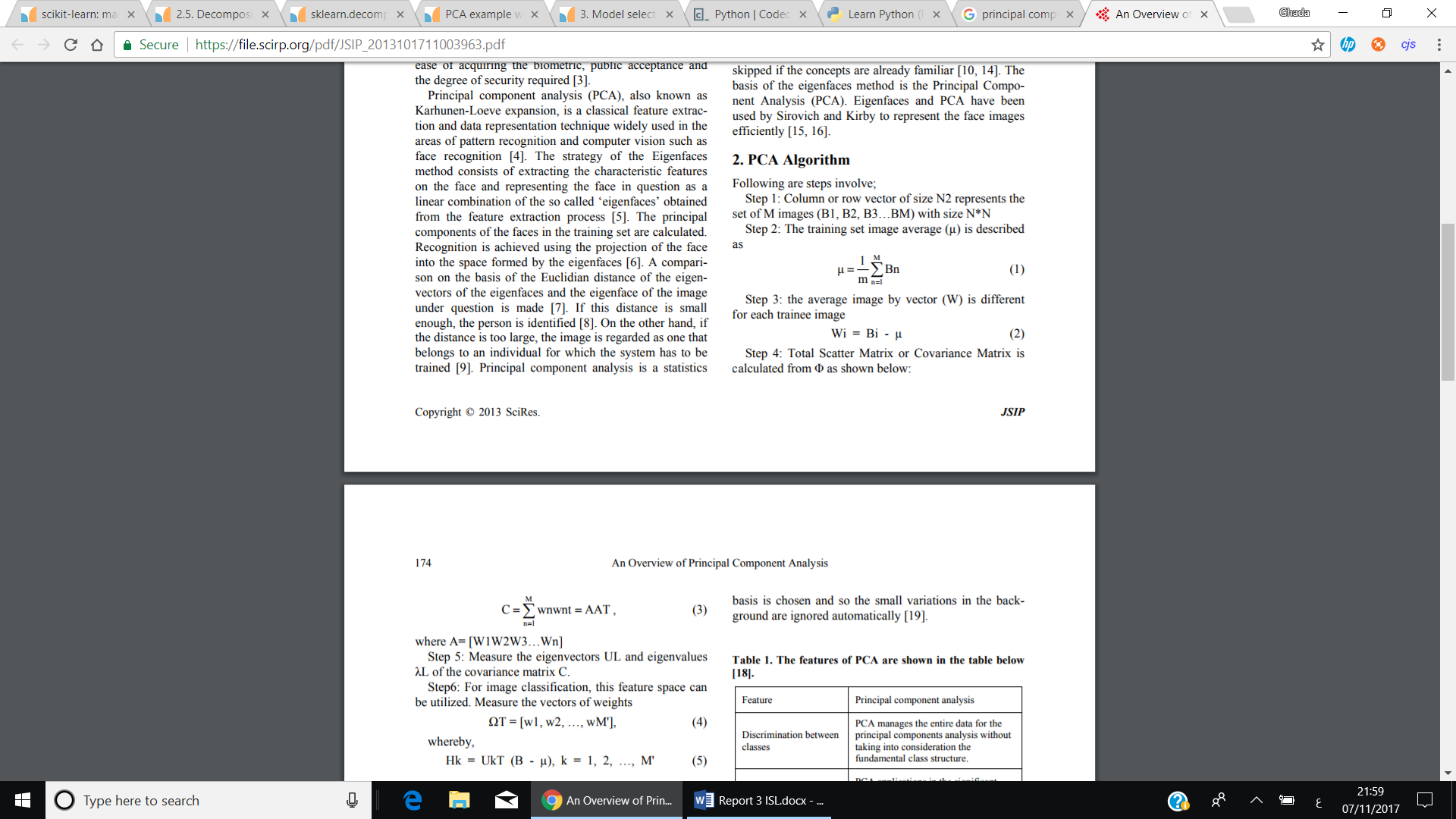
* Unsupervised Statistical Learning Tools:
  + Principal Components Analysis (PCA):

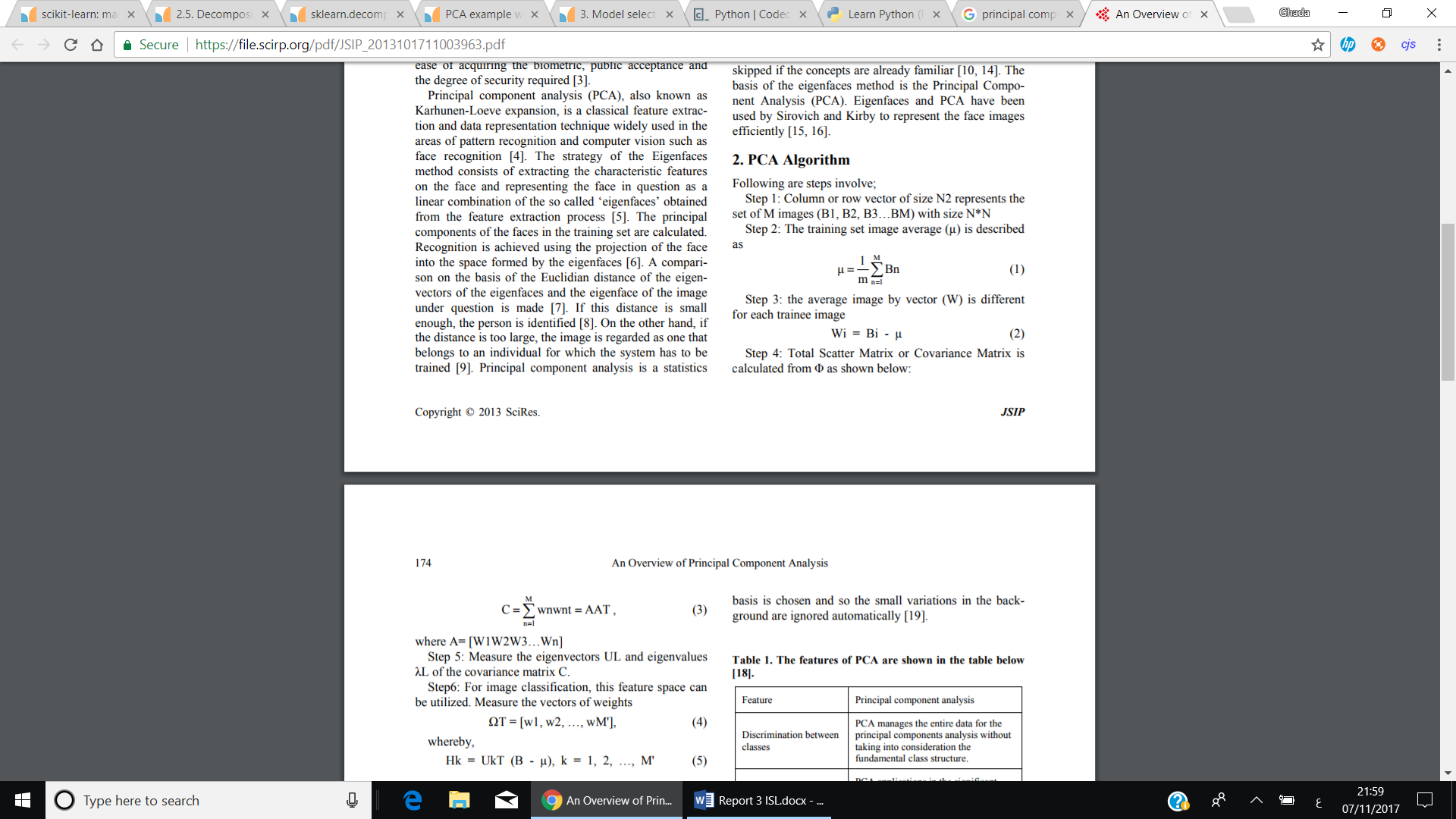
PCA Is a largely known technique, that’s used for reducing dimensionality, and decomposing a large dataset, that has a huge number of dimensions (to a lower dimensional space). The data points will be displayed, as orthogonal components, which describes the maximum amount of the variance. In general, instead of using all parameters of a dataset, using PCA will result in using, for example, 2 values at most that will eventually provide a precise representation of data.

Some of the benefits of using PCA can be summarised in few points,

* + - it clearly decreases the requirement of capacity and memory used.
    - Decrease complexity of system.
    - less sensitive to noise than other algorithms.

However, it can be argued that, evaluation of the covariance matrix result in less accuracy. Which can be viewed as a disadvantage of PCA. <https://file.scirp.org/pdf/JSIP_2013101711003963.pdf>





* 1. Scaling the variables.

Before performing PCA on observations, all data, should be scaled (using a certain method, such as standard deviation = 1).

It is important since; different variables are measured in different units, so when calculating the **variance** of each variable, it will result in first principal component loading vector to be for example, large loading, since it had the largest variance. Mostly, each variable is scaled to have unit standard deviation when the possibility of them having different units is high. Hence, scaling has a substantial impact on obtained results, and should be performed beforehand.

* 1. The minimum number of Principal Components to use is calculated via, **min(n-1, p),** where n is the number of observations and p is the number of variables. However, it is crucial that we use the very first PC’s to help in visualising data. By examining a **SCREE PLOT …**
  + Clustering:

Clustering can be viewed as the set of algorithms, which used in order to find similar subgroups within dataset. Each cluster, contains number of observations in which they’re found similar to each other, whereas observations in other groups are found dissimilar. A partitioning rule, is used to determine what observations can be included at an exact cluster, in other words, what it means for two items to be similar or different.

Nonetheless, Clustering is unsupervised dilemma(problem), because it’s applied to determine different clusters, in a set of unlabelled data.

(<https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/>)

There are quite a few requirements that each clustering algorithm, should satisfy, such as, ability to deal with noise and outliers, high dimensionality, insensitivity to order of data provided and scalability.

Some of issues faced when using clustering include, the difficulty in specifying a distance measurement, which is used to indicate the partition between clusters. Time complexity is also considered, since datasets have high dimensionality, and the results of clustering algorithms can be viewed in different ways.

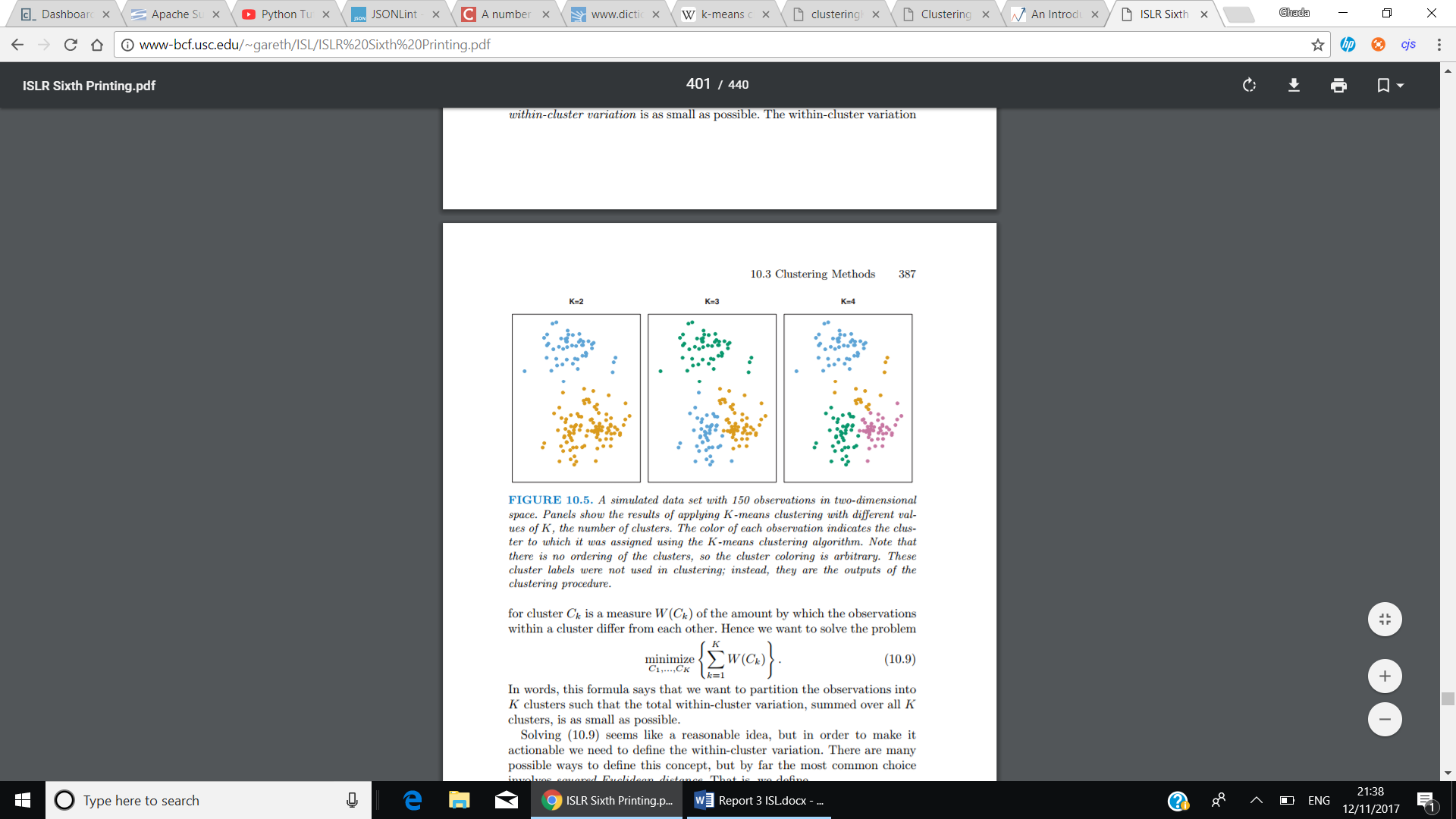
However, since every clustering type follows a different set of rules for determining the similarity measure between points. It’ll be mentioned, 3 of the most popular algorithms used, which are, K-means, Hierarchal and Density-based Algorithms.

(include a bit on the differen types of algorithms? )

(bit on hard and soft clusters? <http://www.ims.uni-stuttgart.de/institut/mitarbeiter/schulte/theses/phd/algorithm.pdf>)

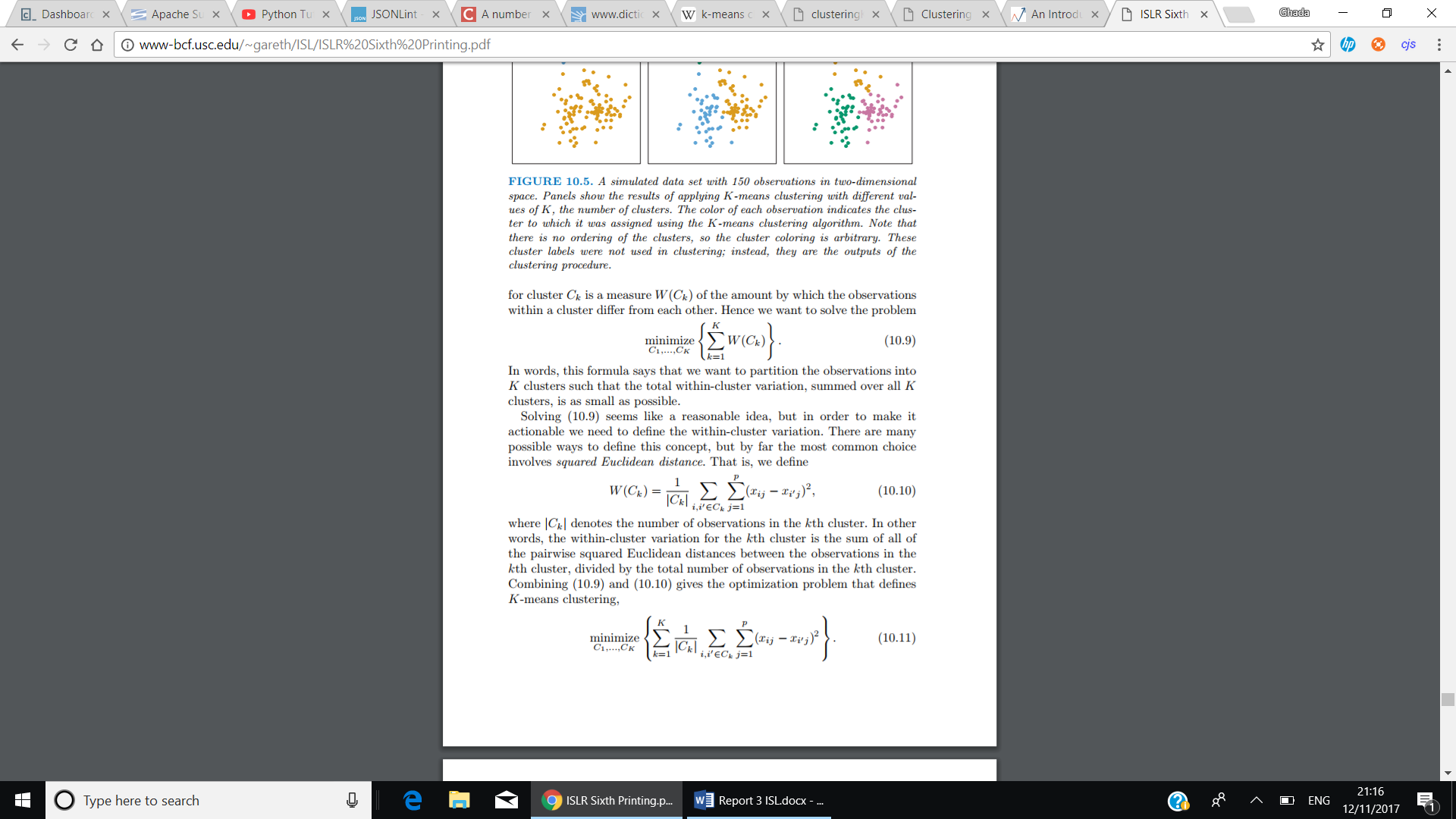
* **K-Means Clustering:**

This algorithm uses a set value of K, which specifies the number of distinct clusters dataset should divided to. Choosing value of K is significant, since the algorithm assign each observation to only one of K-clusters.



Moreover, K-means successfully achieve two main conditions, that are, each data point should belong to 1 cluster, and there is no overlapping found in clusters.

An important point that should be as well considered is, minimizing the within-cluster variation. The within-cluster variation, measures how tightly grouped the clusters are and can in fact be determined using several options, such as, Euclidian’s distance … etc.



The above method, describes the optimisation problem which defines k-means clustering. The within-cluster variation of the kth cluster can be described as, the sum of all pairwise squared Euclidean distances between data points in the kth cluster, divided by |Ck| the number of data points in the kth cluster.

An abstract way to describe K-means clustering algorithms, used from ISLR book, is as follows:

1. For each observation in dataset, assign a random number from 1 to K.
2. Iterate until clusters, stop changing.
   1. For each cluster K, compute cluster centroid.
   2. Allocate each observation to the cluster, that has the nearest centroid.

Centroid, is the mean of observations assigned to each cluster.



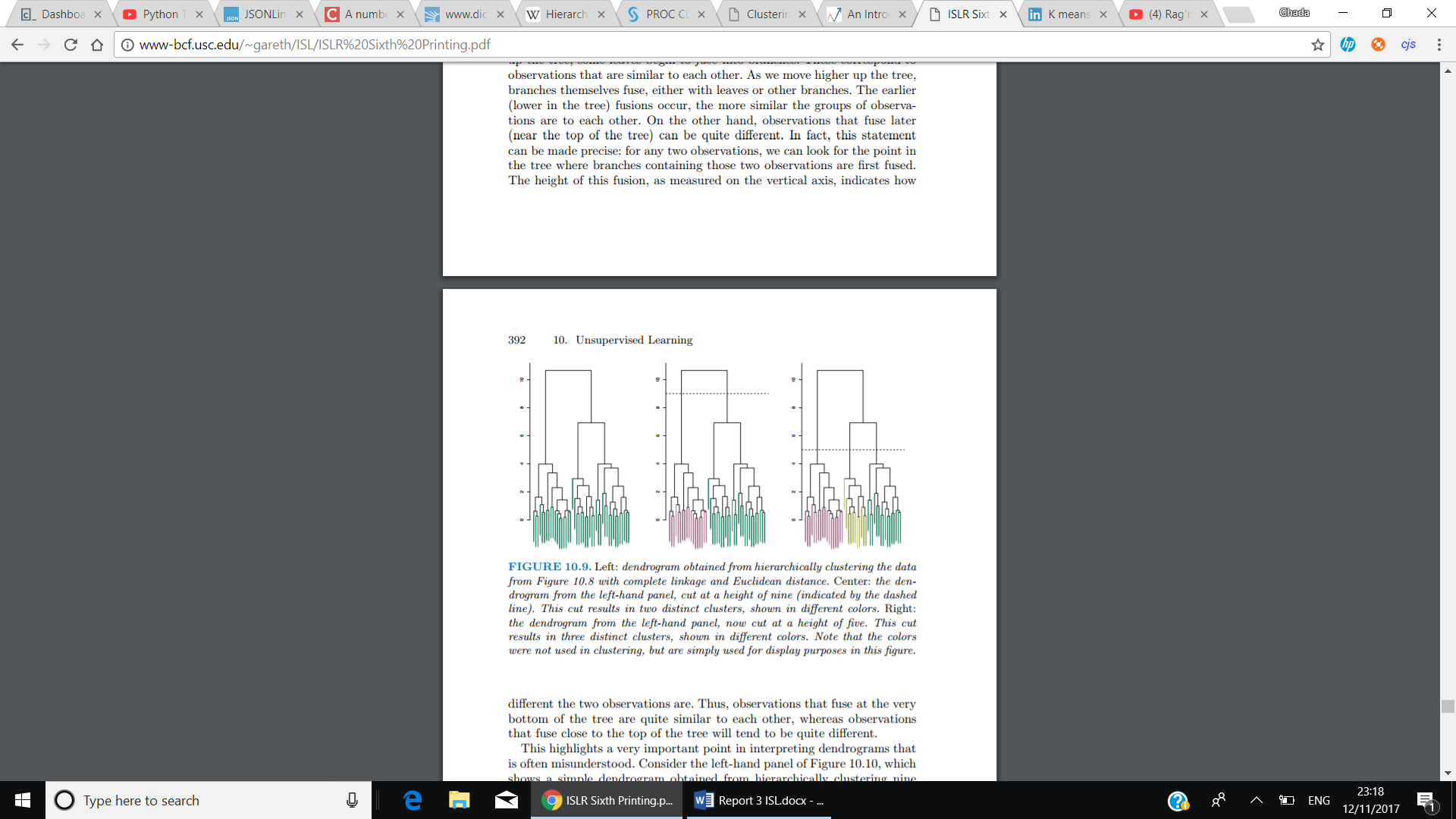
There are several difficulties encountered when using K-means, which include, choosing the value of K (number of clusters) beforehand, portioning data into K clusters while keeping the within-cluster variation minimized, since there are Kn different ways to portion data. However, using K-means is found to be efficient with datasets of high dimensionality, and considered to be easy to implement, and has high performance(fast).Very understandable problem( fs).

<https://pdfs.semanticscholar.org/b0a2/0ecaf9398990bf552c0bd02e1799235a986f.pdf>

* **Hierarchal clustering:**

Another algorithm that doesn’t require the user’s prior knowledge to specify the number of clusters beforehand. Instead it creates a tree-based representation of data observations (builds hierarchy of clusters), which is known as *dendrogram*. It has two types, Agglomerative, and Divisive.

Agglomerative, which is known as “bottom-up, the way it works is that each observation initiates its own cluster, and merging pairs of clusters as moving upwards (the top). Whereas, Divisive “top-down”, all clusters are in the same cluster, and it starts splitting them up, while moving down the hierarchy. However, using Agglomerative is less problematic, since Divisive needs to consider all possible divisions. (<http://www.ims.uni-stuttgart.de/institut/mitarbeiter/schulte/theses/phd/algorithm.pdf>)



The height in the dendrogram at which two clusters are merged represents the distance between two clusters in the data space. In addition, mainly observations that fuse near the bottom in dendrogram, tend to be similar, whereas, observations that fuse near the top of tree are quite different.

To discover number of clusters a dendrogram contain, a horizontal cut at the relevant depth, and depending on where the horizontal line is, the below sets (major branches), are the clusters made (see above).

Different values of the depth of horizontal line would indicate a different number of clusters, i.e. diagram () in middle () the horizontal line is at height = 9, and divides dataset to 2 distinct clusters, while on the diagram in left hand side (), height=5, thus, clusters are set to 3 in that case.

* + Dissimilarity measure: is calculated using variety of methods. i.e. Euclidean distance. Where it considers the distance between two data points. Computed as the square root of the sum of the squared differences of distance between two variables. Another example, is correlation-based distance, which computes the correlation between features of 2 observations to be high, if they’re found similar. (<https://arxiv.org/ftp/arxiv/papers/1205/1205.1117.pdf>)
  + Linkage: on the other hand, describe the dissimilarity between two sets of data points(groups) (two groups of observations), by using one of the following four different types to achieve, the required output, which are complete, average, single and centroid.

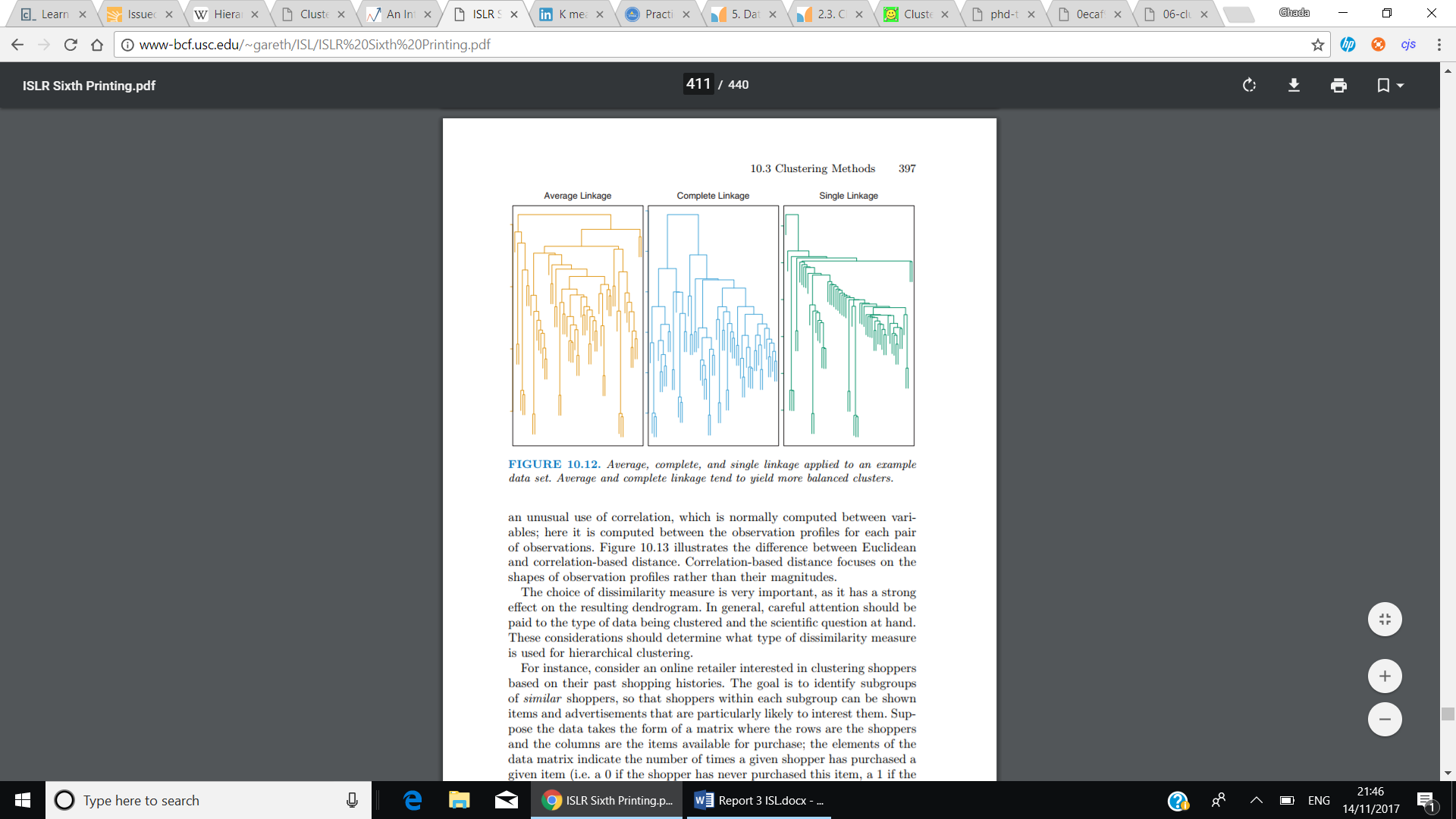
Single linkage: measures the closest pair of points.

Complete linkage: measures the farthest pair of points

Average linkage: measures the average dissimilarity over all pairs

Centroid linkage, measures the distance between the group centroids (i.e., group averages).

<http://www.stat.cmu.edu/~ryantibs/datamining/lectures/06-clus3.pdf>



According to introduction to statistical learning book, a way to describe Agglomerative hierarchal clustering algorithms, is as follows:

* + 1. Starting with n observations, and a dissimilarity measure, such as Euclidean distance, from n to 2 = n(n − 1)/2 pairwise dissimilarities, where each data point is a cluster.
    2. Iterate from i=n to 2:
       1. Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.
       2. Compute the new pairwise inter-cluster dissimilarities among the i − 1 remaining clusters

Advantage?

It’s not required to specify the value of K before starting the algorithm. When using Agglomerative clustering, there is some flexibility given to the user, to choose the number of clusters. On the other hand, using Divisive clustering, result in best possible solution, since the user has access to all data. <https://pdfs.semanticscholar.org/b0a2/0ecaf9398990bf552c0bd02e1799235a986f.pdf>

Disadvantage?

Where to cut the dendrogram counts as a disadvantage, as it is in many cases, seem to be uncertain. When using Divisive, several computational difficulties emerge particularly when splitting the clusters. <https://pdfs.semanticscholar.org/b0a2/0ecaf9398990bf552c0bd02e1799235a986f.pdf>

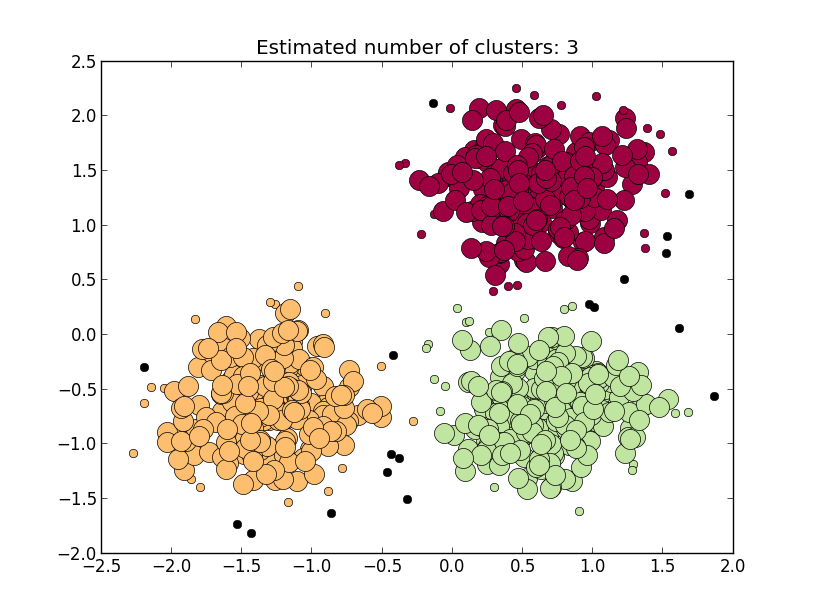
Using Centroid linkage, can cause an inversion, where , which will result in more difficulties, especially in visualising and analysing data.

Sometimes can lead to less accuracy rather than K-means (ISLR,394). Depending on the distance metric used, the results would vary. Furthermore, if the data grouping is incorrectly done in earlier stages, the results cannot be altered after. <https://pdfs.semanticscholar.org/b0a2/0ecaf9398990bf552c0bd02e1799235a986f.pdf>

Comparison between K-means and hierarchal?

Similarities, the horizontal line that cuts through dendrogram in hierarchal algorithm, is equivalent to the K value set in K-means, which both specify the number of clusters included.

* **Density-Based(DBSCAN) Clustering:** (2 paragraphs from <http://onlinelibrary.wiley.com/wol1/doi/10.1002/widm.30/full>)

According to Kreigel et el. explanation of Density-based clustering,” in density-based clustering, a cluster is a set of data objects spread in the dataspace over a contiguous region of high density of objects. Density-based clusters are separated from each other by contiguous regions of low density of objects.”. The resulting clusters, unlike previously mentioned algorithms, would have an arbitrary-shape, that’s determined by high density areas, where number of data points exceeds threshold value. Moreover, data points that exist in low-density areas, are considered either noise, or outlier clusters.

(pic:<https://www.google.co.uk/search?q=DBSCAN&rlz=1C1DSGZ_enGB609GB609&source=lnms&tbm=isch&sa=X&ved=0ahUKEwju6LfDn7_XAhUEDMAKHSd_DKkQ_AUICygC&biw=1536&bih=759#imgdii=_AL2Yk1_e0U9WM:&imgrc=TGZdwFQpGnJe_M>:)

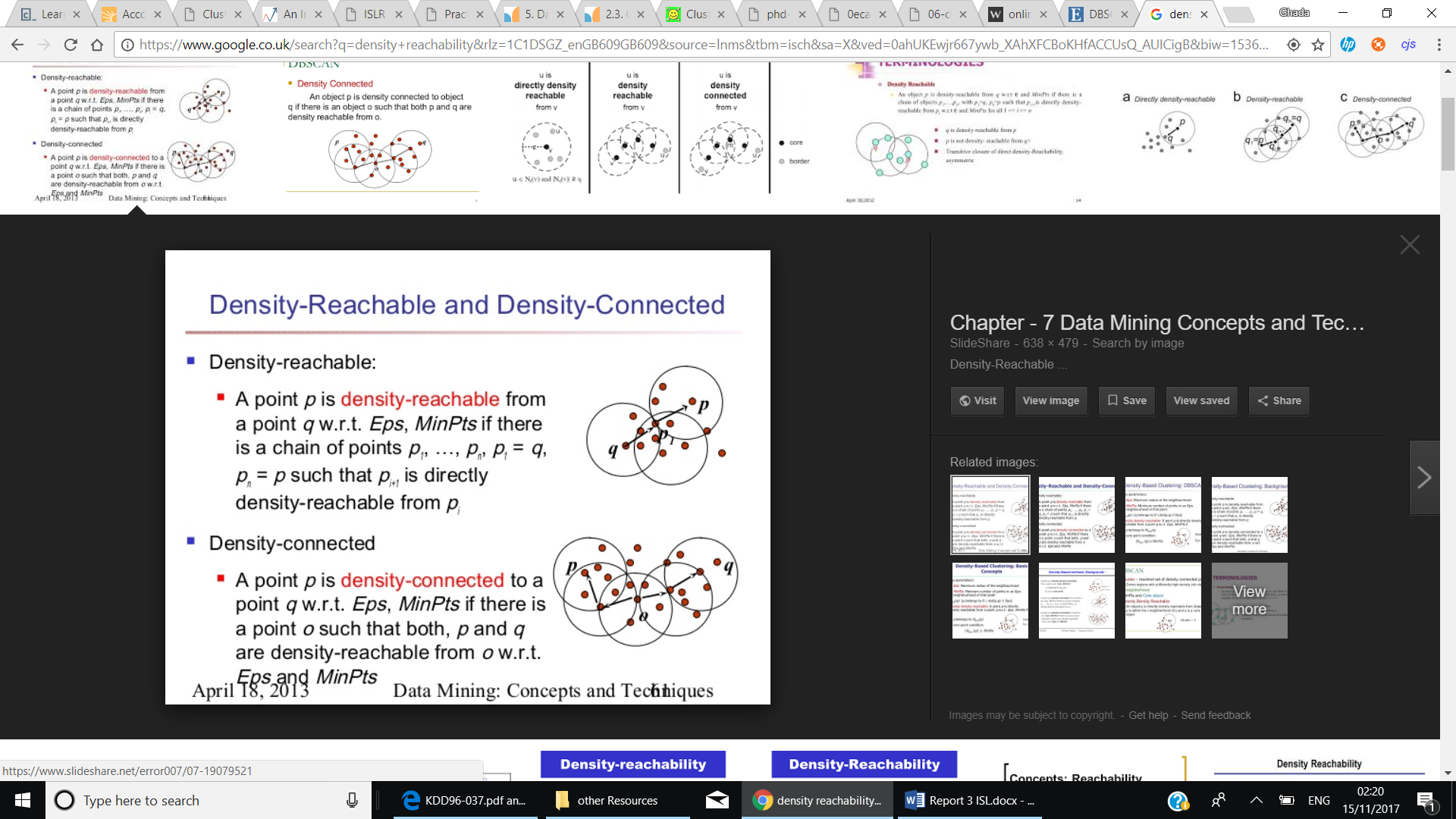
The DBSCAN Algorithm 3 parameters to preform clustering, which are:

<https://arxiv.org/ftp/arxiv/papers/1205/1205.1117.pdf>

* 1. Eps (ε), that describes the radius of neighbourhood; serves as distance threshold.
  2. MinPts, which defines, the minimum number of points that must exist in eps radius, serves as density threshold.
  3. dist(), distance function, i.e. Euclidean distance.

Nevertheless, using the definition of density-reachability, which states that in order for 2 data points , a core and border point, p and q, to be density-reachable, the following criteria should be met:

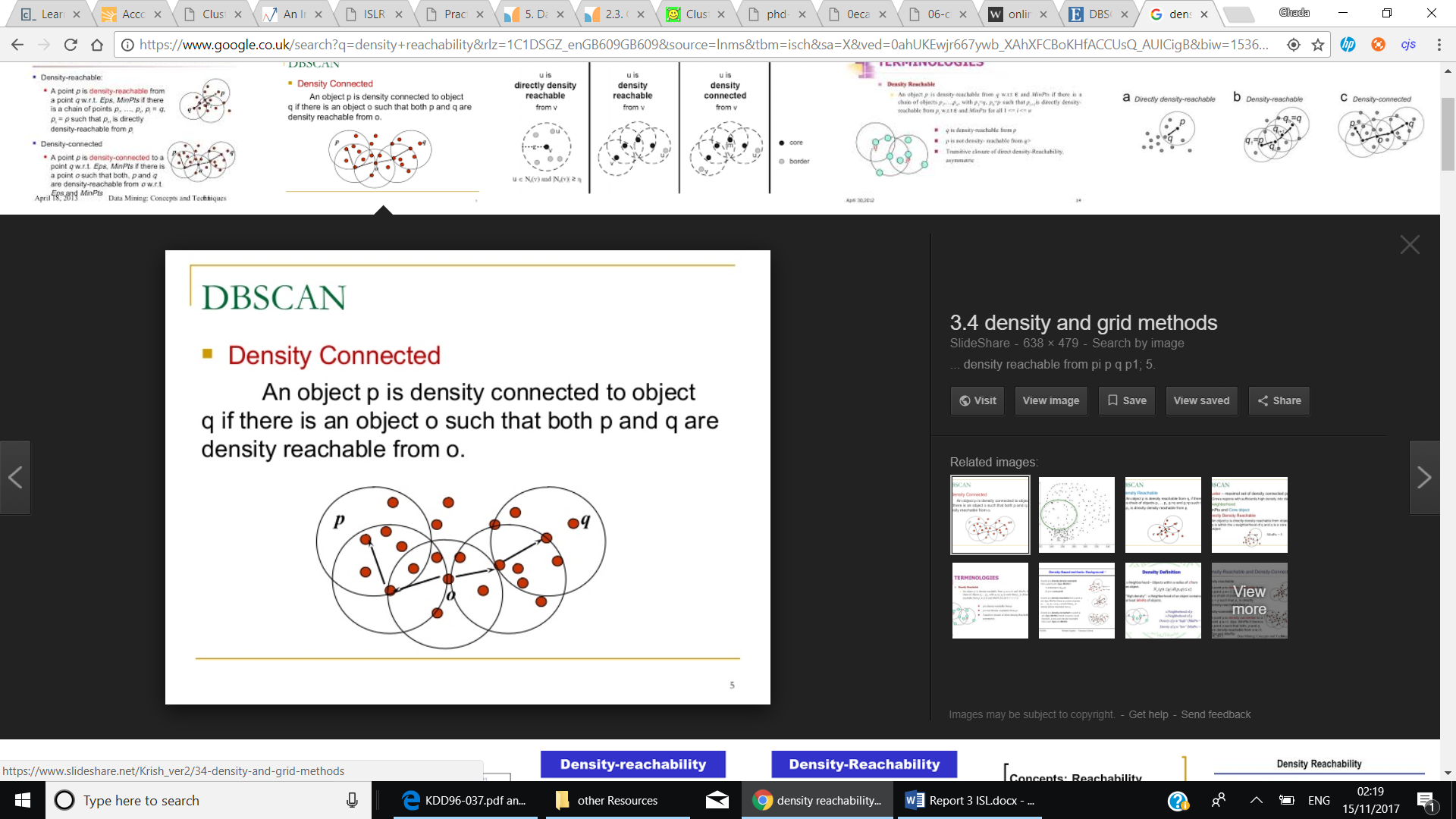
1. , point p is within Eps threshold of q
2. , the number of points in q’s neighbourhood is greater than or equal to MinPts(Core point condition).



Density-Connectivity, is another definition, that implies:

For points p, q and o, to determine if p is density-connected to q, the following condition apply:

1. There exists a point o, where p and q are both density-reachable from o, with respect to Eps and MinPts. ( if they have a distance of less than r)

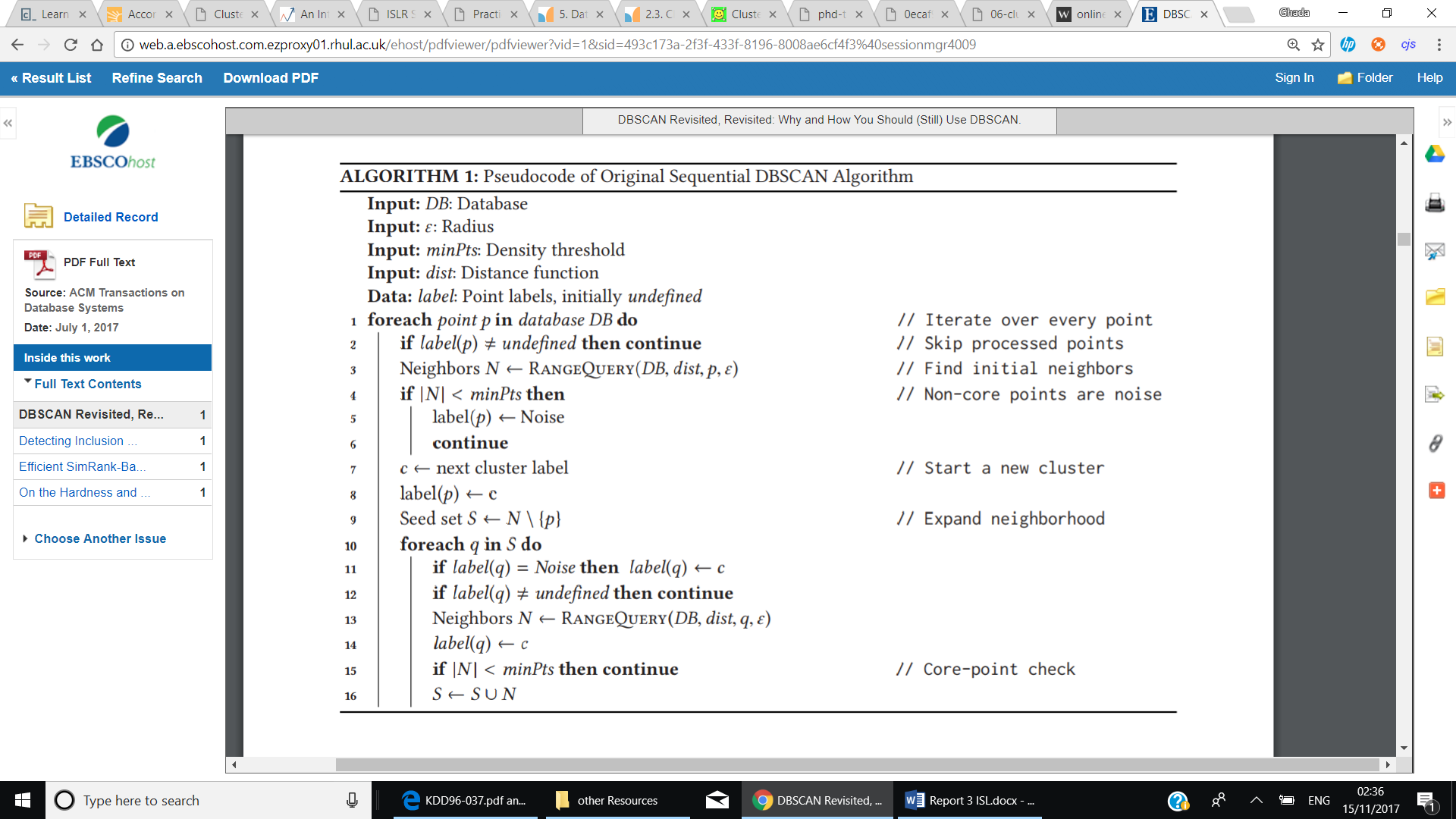


(pic:<https://www.google.co.uk/search?q=density+reachability&rlz=1C1DSGZ_enGB609GB609&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjr667ywb_XAhXFCBoKHfACCUsQ_AUICigB&biw=1536&bih=759#imgrc=08yB0_Rinqs51M>:

An abstract way to illustrate DBSCAN clustering algorithm:

1. Compute neighbours of each point and identify core points // Identify core points
2. Join neighbouring core points into clusters // Assign core points
3. foreach non-core point do:
   1. Add to a neighboring core point if possible // Assign border points
   2. Otherwise, add to noise

At first, distance measure is calculated, that determines shape of cluster. For example, for each observation the distance between its k-nearest neighbours and farthest neighbours is computed, the average of this distance, hence, is the distance measure (can use Euclidian distance). After that, each observation could be categorized, as a core point, border point or an outlier (noise point), using value of Eps threshold parameter.



Advantage?

Disadvantage?

Why Machine Learning?

Due to the significant growth of malware on devices in general, antimalware programs are trying to keep up to date with changes made. Some Antimalware are built to compare signatures to the database, to decide whether a code is considered benign or malicious, this method is now becoming less helpful with time. Using machine learning algorithms to better detect malicious code, as it has some traits that helps in deciding. “So, code that’s determined to be benign might have some traits that the software considers to be a possible indication of malware.”

Machine learning, assist devices to learn new things about data give, without the need of explicitly programming it.

For example, AI-Squared analyses large amounts of data and uses a recurrent neural network and machine learning techniques (also known as unsupervised learning) to find anomalies. Once something abnormal is found, a human analyst is then alerted to confirm whether the activity is a hacker or a genuine user. After the intent of the activity is decided, artificial intelligence takes these findings and puts them into the equation for future reference. This security process is called supervised learning.

<https://www.ccsinet.com/blog/what-to-expect-from-ai-and-cyber-security-roles-in-the-future/>

The only things that unsupervised learning methods have to work with are the observed input patterns xi, which are often assumed to be independent samples from an underlying unknown probability distribution PI [x], and some explicit or implicit a priori information as to what is important. One key notion is that input, such as the image of 1 a scene, has distal independent causes, such as objects at given locations illuminated by particular lighting. Since it is on those independent causes that we normally must act, the best representation for an input is in their terms. Two classes of method have been suggested for unsupervised learning. Density estimation techniques explicitly build statistical models (such as BAYESIAN NETWORKS) of how underlying causes could create the input. Feature extraction techniques try to extract statistical regularities (or sometimes irregularities) directly from the inputs. Unsupervised learning in general has a long and

<http://www.gatsby.ucl.ac.uk/~dayan/papers/dun99b.pdf>

Bibliography:

<http://web.a.ebscohost.com.ezproxy01.rhul.ac.uk/ehost/pdfviewer/pdfviewer?vid=1&sid=493c173a-2f3f-433f-8196-8008ae6cf4f3%40sessionmgr4009>

foreach point p in database DB do // Iterate over every point

if label(p) undefined then continue // Skip processed points

Neighbors N ← RangeQuery(DB, dist,p,ε) // Find initial neighbors

if |N | < minPts then // Non-core points are noise

label(p) ← Noise

continue

c ← next cluster label // Start a new cluster

label(p) ← c

Seed set S ← N \ {p} // Expand neighborhood

foreach q in S do

if label(q) = Noise then label(q) ← c

if label(q) undefined then continue

Neighbors N ← RangeQuery(DB, dist,q,ε)

label(q) ← c

if |N | < minPts then continue // Core-point check

S ← S ∪ N