CLUSTER-MEDIAN PROBLEM:
IRIS

Joaquim Girbau Xalabarder

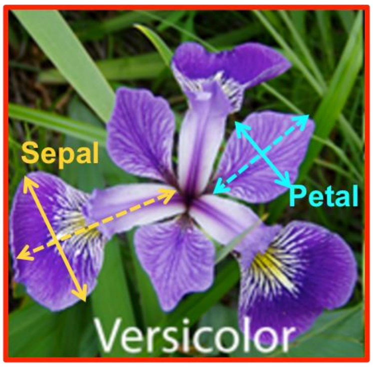
Ignasi Mañé Bosch

18-03-2019

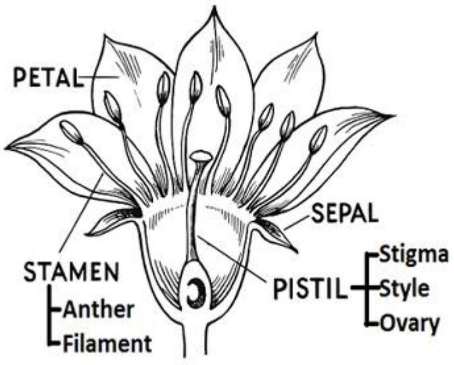
CONTENTS

INTRODUCTION

This project aims to classify three different plants from the taxonomy family *Iridaceae*: *Versicolor*, *Virginica* and *Setosa*.



All of them are similar concerning to their appearance, however they can be differentiated looking the stem, the leaves' layout, the petals, the sepals, the color, the dimensions, etc. During this work, petal length and width, and sepal length and width will be considered to measure to classify these plants.



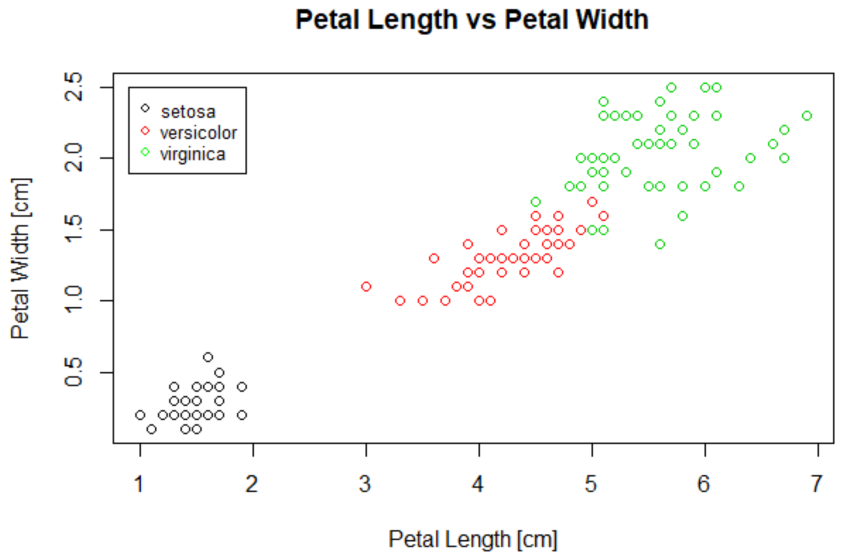
In order to see what are the Sepal and Petal (and other parts of the plant), an image of parts of a flower is provided. The Sepal is always beneath the Petals. The length of the Sepal (and Petal) is the measure from the extreme of the Sepal to the center of the flower (previous image). The width measure will be orthogonal to the fictitious line of the length measure, and will be measured from an extreme of the Sepal to the other (the most separated distance points).

The Data Set has been downloaded in: https://archive.ics.uci.edu/ml/datasets/iris. This data consists on 150 (sorted by type) samples (50 samples for each type of flower), 4 continuous variables (the Length and Width of the sepal and petal of the flowers in centimeters) and the type of the plant (*Versicolor*, *Virginica* and *Setosa*). The last factor, will not be considered to clustering because it will be useful to test if the clustering is good or bad. The first 5 rows of the data set are the following:

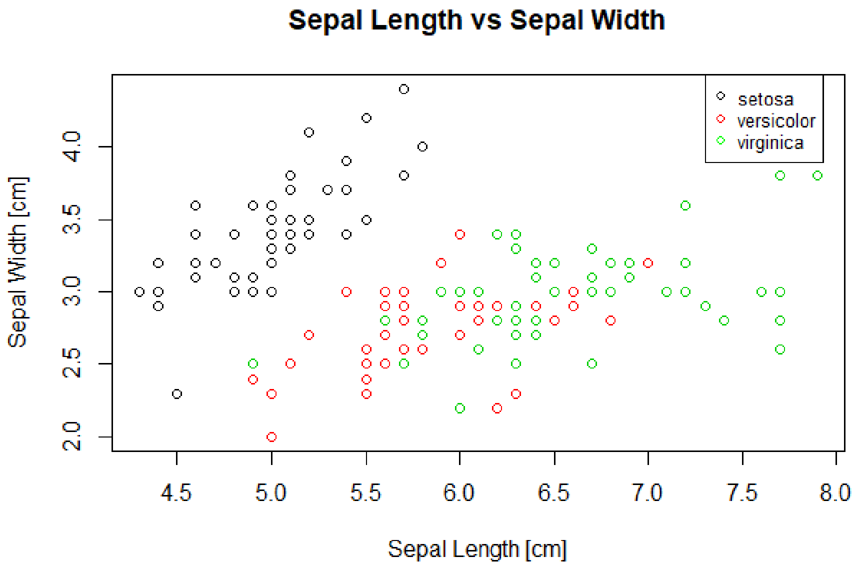
	Sepal.Length [cm]	Sepal.Width [cm]	Petal.Length [cm]	Petal.Width [cm]	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa

EXPLORATORY ANALYSIS

First of all, an exploratory analysis has to be performed to know approximately which variables can classify the flowers. To do it, we could not plot in a 5D graph using the 5 variables, however, we can try to plot 2 continuous variables in a 2D plot and change the color of the points depending on which set of flowers do they belong.



If we take a look in the previous graph, we can see that the different types of flowers seems to be classified with these 2 continuous variables. It can be observed that there could be a positive linear relation between the Petal Width and Length. If we take a look on the sepal measures:



We can see that the Versicolor flower could be classified correctly using Sepal measurements, however, using only these variables it is impossible to classify the Virfinica and Setosa flowers. That is why we will try to classify using 4 variables and 2 variables (petal measurements).

CLUSTERING

Using AMPL

Given a data matrix of m points (number of rows) and n variables (number of columns), we want to group them in k clusters. Even though only k cluster are needed, m clusters will be considered, such that m-k clusters will be empty. The cluster which has as median the element j, will be denoted “cluster-j”.

Using a distance matrix between pairs of points (it can be done using R), we can try to formulate the problem using optimization.

Variables

For all i,j = 1, …, m   Where m is the number of observations

$$x_{ij} = \begin{cases} 1 & \textit{if the element i belongs to cluster j} \\ 0 & \textit{otherwise} \end{cases}$$

In AMPL:

1) Define the set M
   set M := 1..m **by** 1

2) Define the variable x
   var x {M, M} **binary**

Objective function

The minimum total distance between points and the median has to be analysed to make k clusters. So, we have to minimize the distance of all points to their medians:

$$\min \sum_{\mathrm{i}=1}^{\mathrm{m}} \sum_{\mathrm{j}=1}^{\mathrm{m}} \mathrm{d}_{\mathrm{ij}} \mathrm{x}_{\mathrm{ij}}$$

In AMPL:

```
minimize DistMedian: sum {i in M} sum {j in M} d[i,j]*x[i,j];
```

Where d[i,j] is the element i,j from the distance matrix (performed using R)

Constraints

We have to make sure that every point belongs to one cluster-j, i.e.:

$$\sum_{\mathrm{j}=1}^{\mathrm{m}} x_{ij} = 1 \quad i = 1, \dots, m$$

In AMPL:

```
subject to onecluster {i in M}: sum {j in M} x[i,j]=1;
```

We also have to make sure that there are exactly k clusters, so:

$$\sum_{j=1}^{m} x_{jj} = k$$

In AMPL:

```
subject to kclusters: sum {j in M} x[j,j]=k;
```

And finally, we have to ensure that a point belongs to cluster-j if the cluster-j exists (if xjj is 0, then xij is also 0), i.e.:

$$x_{jj} \geq x_{ij} \quad i, j = 1, \dots, m$$

To simplify the number of constraints, it can be written as:

$$m \cdot x_{jj} \geq \sum_{i=1}^{m} x_{ij} \quad j = 1, \dots, m$$

In AMPL:

```
subject to existscluster {j in M}: m*x[j,j] >= sum {i in M} x[i,j];
```

The *.dat* file contains the parameter k (number of clusters) and the distance matrix of the data. The *.run* file contains the execution of the program, the display of the variables x (when x = 1) and the display of the solving time required.

For this work, we will classify the data using 150 observations of 4 variables and 2 variables. When we use 2 variables, we will classify the data using 15, 30, 60, 120 and 150 observations to see how the solving time increases. The results and observations are provided in the section *Results using AMPL*.

Using KRUSKAL heuristic

Since the computational cost for solving the clustering problem is very high, some heuristics are commonly used to find an approximate solution in a reasonable amount of time.

We will apply a heuristic solution for our problem that relies on minimum spanning trees. The idea is the following: On the basis of a minimum spanning tree among all the possible existing trees for our problem, we will erase as many largest edges as clusters we would like to classify the data into minus one, obtaining different isolated groups of points that will define a cluster. However, the problem now is how to find a feasible minimum spanning tree for our problem. One possible solution may be find using Kruskal's algorithm, for instance.

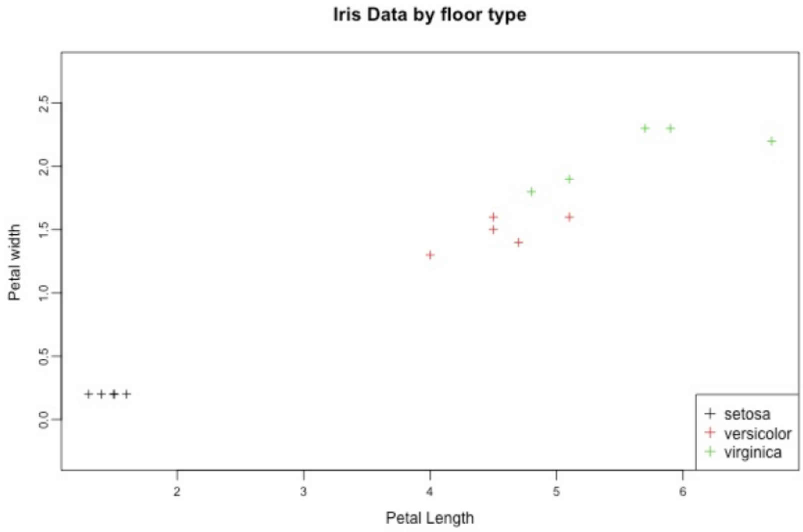
Kruskal's algorithm finds a minimum spanning tree for a connected weighted graph adding increasing cost arcs at each iteration:

- First it creates a forest F made of n sets of trees, one for each vertex
- Second, it creates a set S that contains all the existing arcs between vertexes
- While S is nonempty or F is not a spanning tree
    i. Remove the arc with the least cost form S
    ii. If the removed arc connects two different sets of trees, then add it to the forest and combine both trees in a single tree, else discard the arc.

At the termination of the algorithm, if the graph is connected the forest F forms a minimum spanning tree.

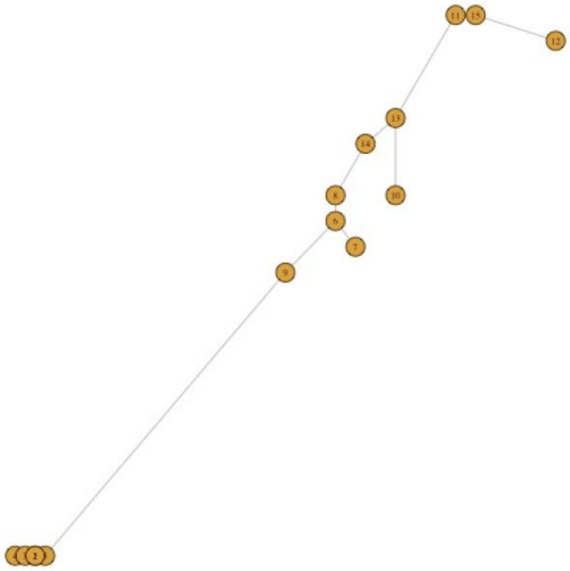
To illustrate how the algorithm works and for simplicity, we will implement this algorithm in the subset of N=15 observations used in the previous part of the document.

If we represent the data of this subset we can find three clusters. Setosa, again, is very particular and its traits differ from those of versicolor and virginica. However, as in the previous case, some observations of versicolor and viriginica are very similar in terms of petal length and petal width, making them difficult to distinguish.



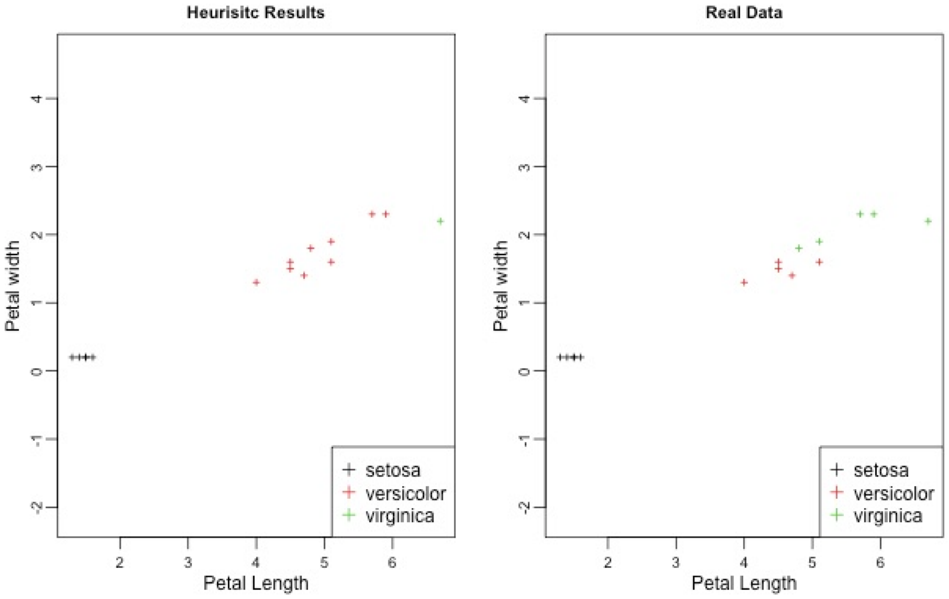
To implement the algorithm, first, we calculate the distance matrix of the data and define all the nodes or, in this case, observations that will be part of our network or graph.

Second, we find, using the Kruskal algorithm implemented in R, the minimum spanning tree that connects all the nodes, or points, of the graph following the steps described above. The final result is presented in the following chart and table:



As we mentioned, to identify the clusters based on this result we have to cut the more expensive or larger edges of the graph. Thus, based on the results obtained in the R, we have to erase the arcs marked in red color {9, 5} and {15, 2}. Notice that, although the last plot seems to contradict our decision, it is not true that arc {13, 11} is larger than {15, 2}. Probably, this can be explained by the relative scale of the axes used by the library "igraph" that has been used for plotting the nodes on the screen.

Once the two largest arcs are erased from the minimum spanning tree, we relaunch the Kruskal algorithm using in the remaining arcs. Since the nodes are no longer connected, in this case, we obtain three different forests, each containing the nodes that form a cluster.



As we see in the chart, contrary to the exact solution, the approximate solution based on the heuristic does not correctly identify the virginica cluster, although it does detect that setosa points form group with similar characteristics by means of petal length and petal width.

RESULTS

Results using AMPL

All the raw results AMPL provides, are attached in the zip file. The result provides us which observations are the median of the clusters (represent the clusters) and which observations belong to the cluster 1 (*Setosa*), 2 (*Versicolor*) or 3 (*Virginica*).

We have classified in 3 clusters the data using all the variables (4) and all the observations (150), and using 2 variables (Petal Length and Petal Width) and 15, 30, 60, 120 and 150 observations to analyse the computational time.

Performing all the classifications we can summarize them in a table:

	Observations used	REAL			PREDICTED	Accuracy	Real Time [sec]
		Setosa	Versicolor	Virginica			
2 Variables	15	5	0	0	Setosa	86,7%	0,188
		0	5	0	Versicolor		
		0	2	3	Virginica		
	30	10	0	0	Setosa	96,7%	0,312
		0	9	1	Versicolor		
		0	0	10	Virginica		
	60	20	0	0	Setosa	95,0%	2,532
		0	18	2	Versicolor		
		0	1	19	Virginica		
	120	40	0	0	Setosa	96,7%	27,954
		0	37	3	Versicolor		
		0	1	39	Virginica		
	150	50	0	0	Setosa	94,7%	82,078
		0	49	1	Versicolor		
		0	7	43	Virginica		
4 Variables	150	50	0	0	Setosa	89,3%	37,062
		0	48	2	Versicolor		
		0	14	36	Virginica		

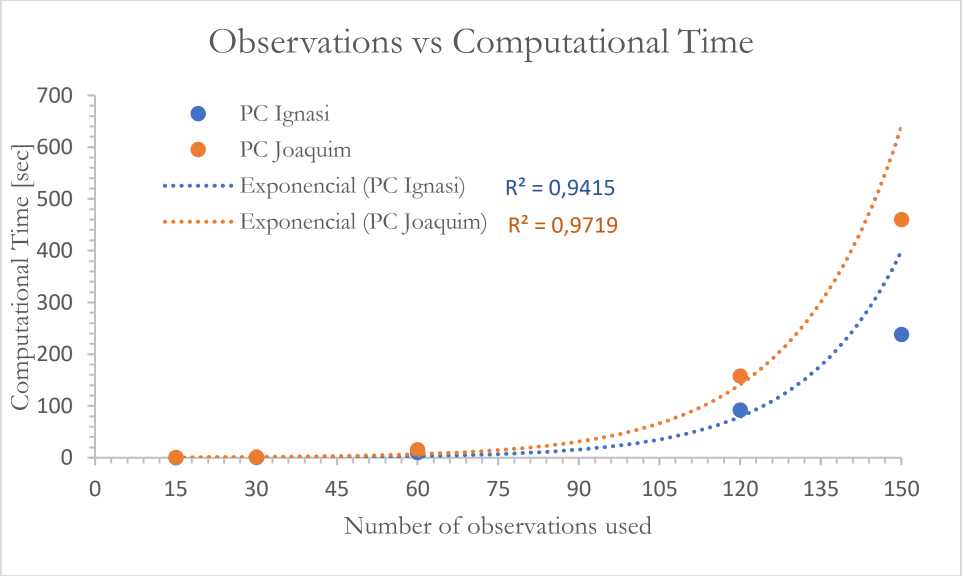
Bad accuracy (beside the others) can be observed using 15 observations and 2 variables and 150 observations and 4 variables (<90%). If we remember the image and the exploratory graphs, the flowers can be distinguished more or less by the petals, however, the sepals are quite similar.

We have decided to work using only 2 variables for all the analysis because, looking at the previous table and the exploratory graphs, it can be observed that the accuracy using 2 variables is better than 4 variables, even if the solving time with 4 variables are less than the solving time taken using 2 variables.

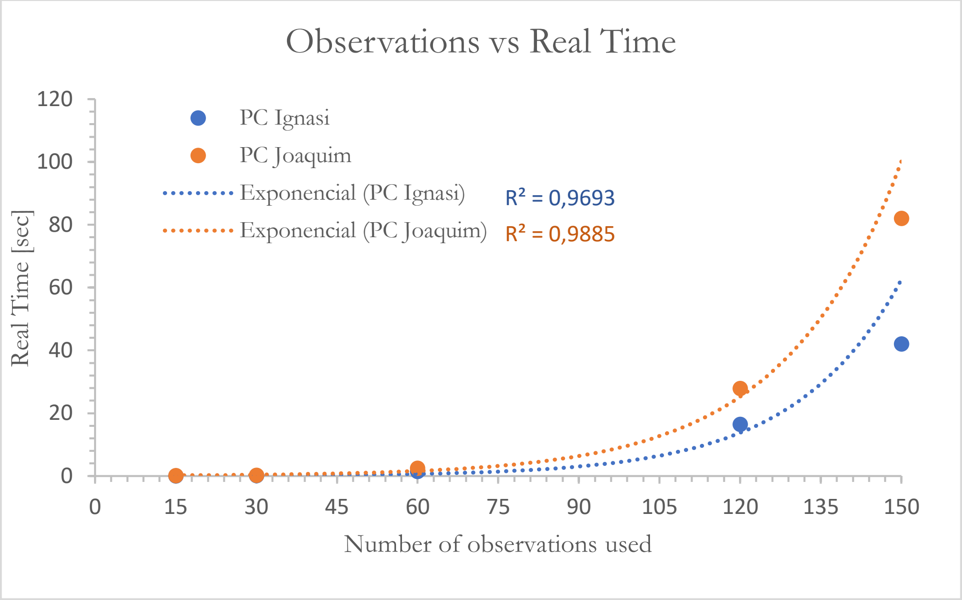
The computational time (total time used by CPUs) and the real time (total time that the user waits until AMPL gives its solution) can be computed in AMPL. It allows us to compare the time using the computer of Joaquim vs using Ignasi's computer. It is interesting to see which behaviour follows the solving time (real and computational) using solver CPLEX.

PC JOAQUIM			**PC IGNASI**		
N	Real Time [sec]	Computational Time [sec]	N	Real Time [sec]	Computational Time [sec]
15	0,19	0,516	15	0,04	0,114
30	0,31	1,156	30	0,14	0,678
60	2,53	15,766	60	1,47	10,016
120	27,95	157,516	120	16,46	92,369
150	82,08	459,844	150	42,06	238,124

To compare the two computers and the behaviour of the solving time depending on the number of observations (N), two plots (one for Real time, another for Computational Time) and the computation of the regression exponential line with its coefficient of determination (R^2) have been made.

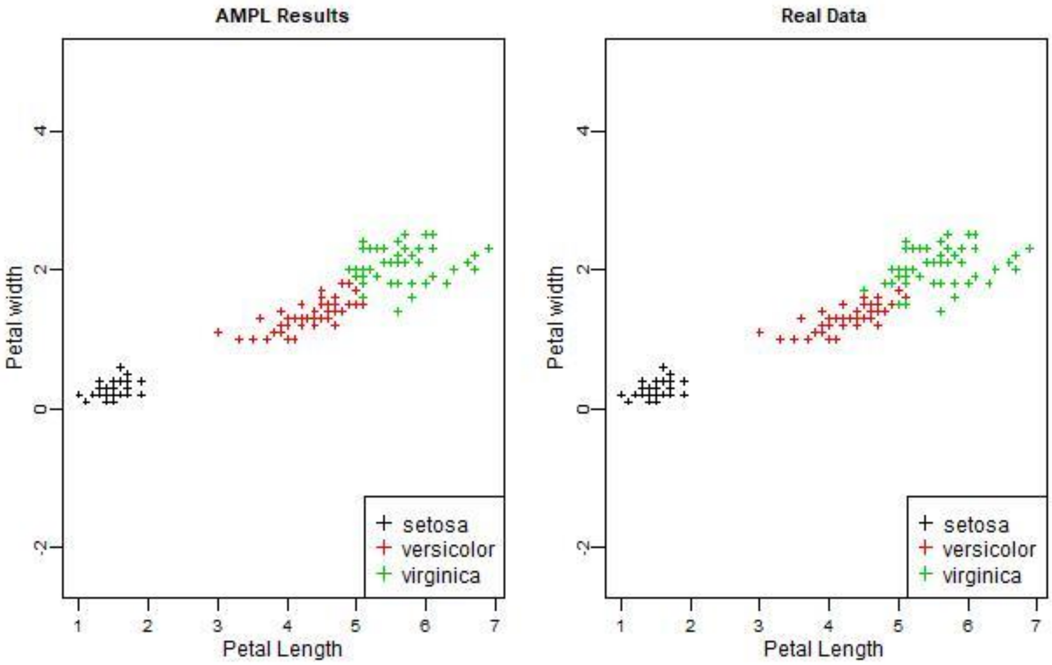


The computational time is the time only one processor would need to execute the AMPL commands. To fit an exponential, it would be better to use large amount of observations to obtain large computational time, however we did not have more than 150 observations in the data set.



As we can observe, the PC of Ignasi is faster than the PC of Joaquim. We tried to fit the points in a polynomial line with 2 degrees and it does not fit so well (using polynomials with 3 or 4 degrees it fits well, but because of overfitting: we only have 5 points to fit). Finally, we tried to fit the points into an exponential line and it seems that the line fits well (its coefficient of determination (R^2) is 0.97).

Another thing to analyse is the comparison between the real data and the solution using optimization. It can be seen in the table which contains the confusion matrix, however, graphically its better to understand what is happening. The next plots are the Petal Length vs Petal Width with the optimization result and the real data. It allows us to compare visually the observations which fail or success.



As we can observe in these graphs, the AMPL results give us a good approximation of the reality. This solution only fails in the intersection of the two clusters. (For all plots, it can be generated using the *04_Analysis_V2*.R file).

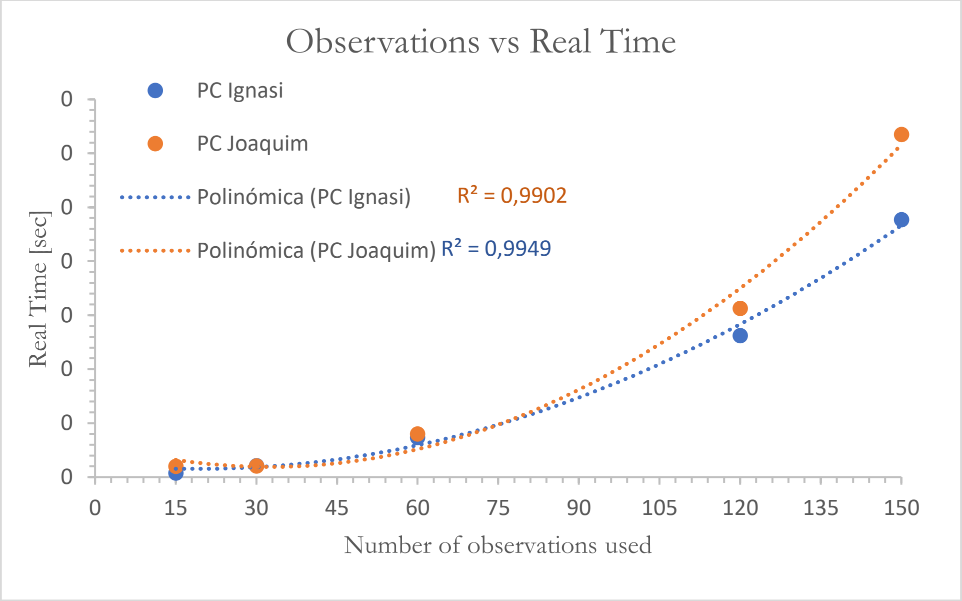
Results using KRUSKAL

We apply the heuristic described above to cluster the whole points belonging each of the data sets created using the data generator R script. As we already saw with the sub set N=15, the misclassification rate of the virginica plants is very high. No matter the size of the moisture, the heuristic is not able to correctly identify three differentiated clusters of points. As before, we have summarized the results in the following table:

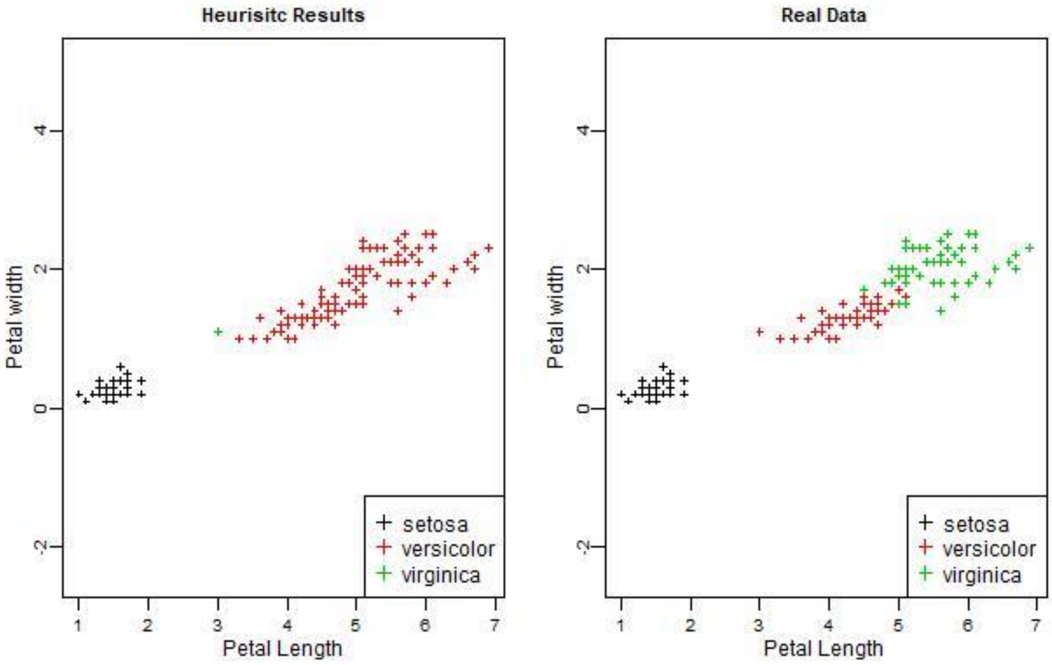
	Observations used	REAL			PREDICTED	Accuracy	Real Time [sec]
		Setosa	Versicolor	Virginica			
2 Variables	15	5	0	0	Setosa	73,3%	0,00100
		0	5	4	Versicolor		
		0	0	1	Virginica		
	30	10	0	0	Setosa	70,0%	0,00103
		0	10	9	Versicolor		
		0	0	1	Virginica		
	60	20	0	0	Setosa	73,3%	0,00401
		0	20	16	Versicolor		
		0	0	4	Virginica		
	120	40	0	0	Setosa	67,5%	0,01563
		0	40	39	Versicolor		
		0	0	1	Virginica		
	150	50	0	0	Setosa	66,0%	0,03176
		0	49	50	Versicolor		
		0	1	0	Virginica		

Despite not being accurate enough to correctly detect the three clusters, the heuristic solved the problem in far less time than AMPL and cplex required to find the exact solution of the k-median clustering problem. Theory results, tell us that the computational time needed to find a minimum spanning tree using kruskal's algorithm is polynomic and, as the data recollected seems to confirm in the following table and char.

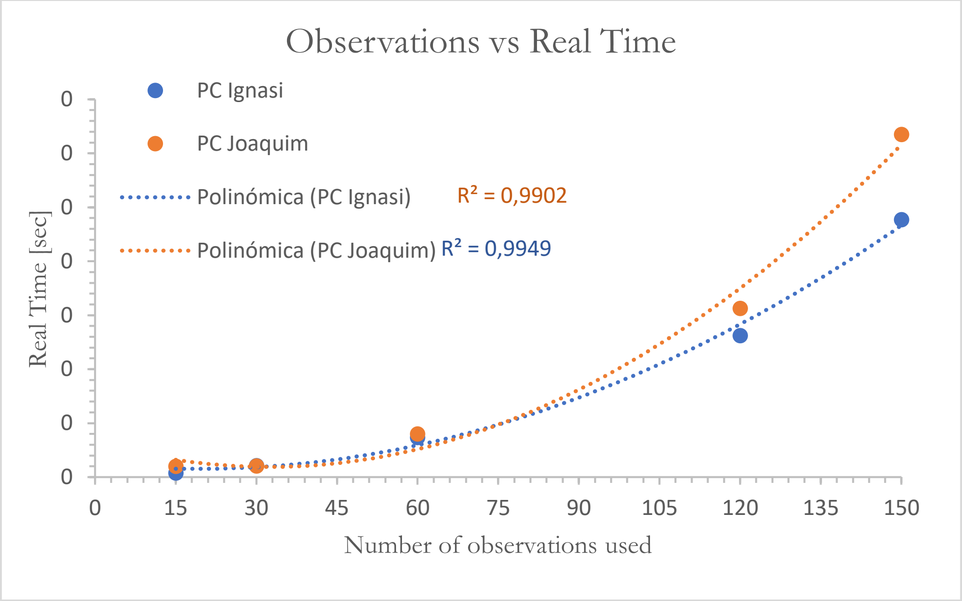
PC JOAQUIM		PC IGNASI	
N	Real Time	N	Real Time
15	0,00100	15	0,00039
30	0,00103	30	0,00107
60	0,00401	60	0,00368
120	0,01563	120	0,01313
150	0,03176	150	0,02385



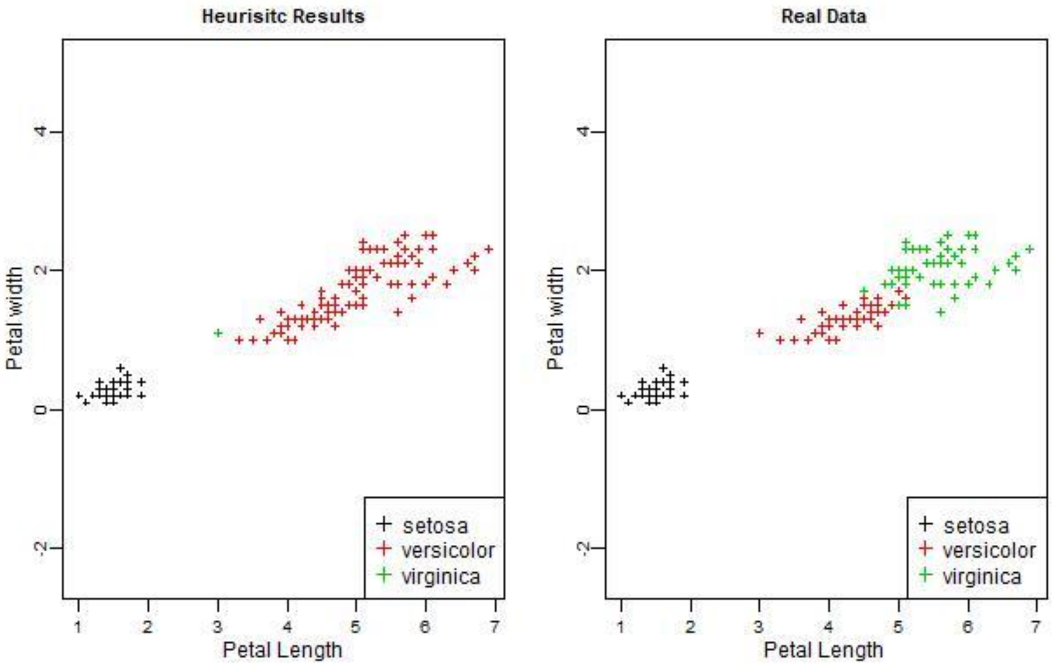
Notice that, in this case, the time needed to find a solution does not growth exponentially as N gets higher. Rather it seems to grow following a polynomial of grade two.



As we see in the chart, again, the approximate solution based on the heuristic does not correctly identify the virginica cluster, although it does detect that setosa points form group with similar characteristics.



Notice that, in this case, the time needed to find a solution does not growth exponentially as N gets higher. Rather it seems to grow following a polynomial of grade two.



As we see in the chart, again, the approximate solution based on the heuristic does not correctly identify the virginica cluster, although it does detect that setosa points form group with similar characteristics. (For all plots, it can be generated using the *05_KruskalV2.R* file).

CONCLUSIONS

Accuracy

On one hand, it is clear that AMPL returns a better solution of the problem allowing us to cluster the data precisely, returning accuracies around 95%. However, using KRUSKAL, we only obtain accuracies in the order of 70%.

It seems the KRUSKAL algorithm fails to distinguish clusters whose observations are pretty close. So KRUSKAL is a bad heuristic to classify overlapping clusters of points.

Polynomial vs exponential time

On the other hand, although the solution provided by AMPL is better than the solution obtained using KRUSKAL heuristic, it requires much more computational time. Actually, the computational time needed to obtain the global optimal solution growth exponentially as the number of observations increases. Therefore, for large data sets it might be impossible to find a solution in a reasonable amount of time.

However, KRUSKAL's heuristics solution, despite not being as good as the global optimal solution, is computationally cheaper to compute, allowing us finding approximate solutions in a polynomial time (for 150 observations the solving time of KRUSKAL is 0,03 seconds and the solving time using AMPL is 82 seconds)

Final remarks

We can conclude that there exists a trade of between computational time and accuracy, when trying to solve the k-median-clustering problem. So, before solving this kind of problem we need to consider if weather it is worth to try to find the global optimal solution at an expensive computational time.