

METHODOLOGY

STUDY OBJECT

In this paper we are going to talk about selecting variables in a specific dataset and use the selecting data to do a spatially constrained clustering. The source we got is a geojson file that chose 12 ecodistricts in the St. Lawrence River Basin. A table about some features in each ecodistricts included in the source file as well. According to a paper that relate to the source file (Adams, M. D., Kanaroglou, P. S., & Coulibaly, P., 2016), we can see that the attributes are classed as physiographical attributes, climatic attributes, hydrological attributes and hydrogeological attributes. Table 1 shows the 23 attributes and their identifier, which will be indicated in the coming sections.

Identifier	Description
Shape_Leng	Length of the boundary (m)
Shape_Area	Area (m ²)
FST	Forest cover (%)
LKS	Surface water (%)
AGR	Agricultural lands (%)
HGA	Well-drained soil cover (%)
HGC	Poorly drained soil cover (%)
HGD	Very poorly drained soil cover (%)
RZD	Mean root zone depth (cm)
EVM	Mean elevation (m)
RRE	Relief ratio
EVS	Mean slope (%)
MAP	Mean annual precipitation (mm/year)
MAT	Mean annual temperature (°C)
TMX	Mean maximum monthly temperature (°C)
NDP	Annual number of precipitation days (days)
PSI	Precipitation seasonality index
MPE	Potential Evapotranspiration (mm/year)
PPE	Mean annual precipitation minus the potential Evapotranspiration (mm/year)
RSD	Snow day ratio
MXQ	Annual mean maximum discharge (m ³ /s)
DOW	Depth to subsurface water (m)
DAM	Pollution potential of the aquifer media

Table 1. Attributes and its description

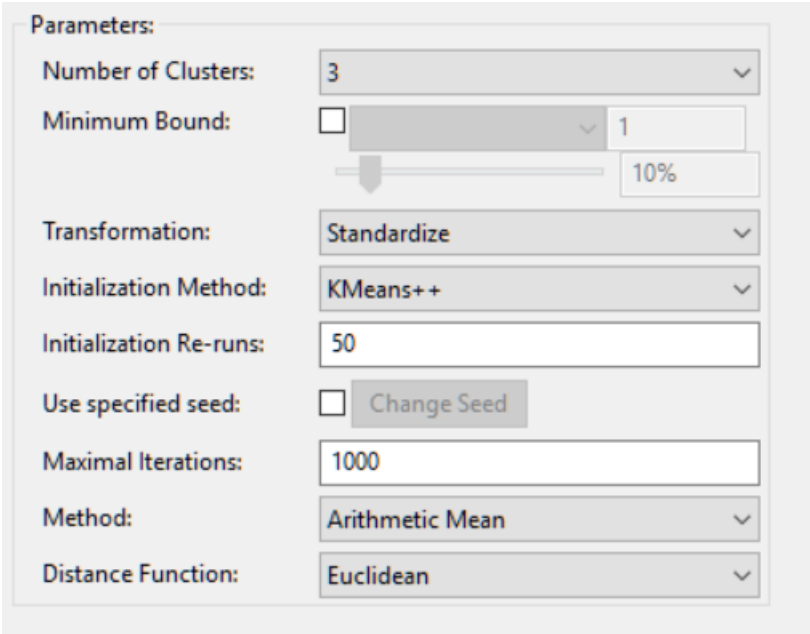
VARIABLE SELECTION

In this paper, we are only picking 6 variables to do the clustering. In order to make a proper choice, what I was trying to select the variables that are unique and important in the dataset. In our dataset, highly correlated variables should be removed to reduce variable redundancy. Perfectly correlated variables are truly redundant in the sense that no additional information is gained by adding them. But Very high variable correlation (or anti-correlation) does not mean absence of variable complementarity (Guyon, I., & Elisseeff, A. ,2003). What I was doing is to use a package caret in R language to check my variables (Appendix A – variable selection). Also, I was not going to exclude Shape_Leng and Shape_Area as my selection because they should be not considered as factors that influence stream flows. Based on these, I was approaching in two different ways. The first one is just simply set a correlation cutoff and remove the variables that has correlation greater than the cutoff. Another method is to using a recursive feature elimination (aka. RFE) function in Caret package using random forest model (Breiman, L. ,2001) to test the 6 variables that gave less errors. RFE is based on wrapper methods that evaluate multiple models using procedures that add and/or remove

predictors to find the optimal combination that maximizes model performance. In essence, wrapper methods are search algorithms that treat the predictors as the inputs and utilize model performance as the output to be optimized.

REGIONALIZATION

The spatial objects for use in regionalization were multivariate lattice data sets and required an appropriate method to minimize multivariate heterogeneity within regions and maximize heterogeneity between regions (Adams, M. D., Kanaroglou, P. S., & Coulibaly, P., 2016). I applied GeoDa as the regionalization tool. The method I applied on clustering is k-mean. Specifically, the algorithm GeoDa used is an improved version of standard k-mean, k-means++, which comes with a theoretical guarantee to find a solution that is $O(\log k)$ competitive to the optimal k-means solution (Arthur D, Vassilvitskii S, 2007). Figure 1 shows my k-mean parameter configuration in GeoDa version 1.12.1.59. Another thing need to be mentioned is the spatial weight matrix, which are used to represent the neighborhood relationship between each observations and could be used in making the clusters spatially contiguous.



The image shows the 'Parameters' dialog box for the K-means clustering algorithm in GeoDa. The settings are as follows:

Parameter	Value
Number of Clusters:	3
Minimum Bound:	<input type="checkbox"/> 1 (with a slider set to 10%)
Transformation:	Standardize
Initialization Method:	KMeans++
Initialization Re-runs:	50
Use specified seed:	<input type="checkbox"/> Change Seed
Maximal Iterations:	1000
Method:	Arithmetic Mean
Distance Function:	Euclidean

Figure 1. K-mean settings

RESULT

VARIABLES

Using the two method on getting variables, I was able to get two subsets of 6 variables. The first subset of variables I got is by using simply removing the variables that has lower correlation than a specific cutoff, which in my case is 0.55. There are only 6 variables that can have smaller correlation than 0.55 in the correlation matrix (table 2). The selecting variables in the first subset are AGR, HGA, RZD, EVS, TMX and PPE. In the second method, I used same ways to remove the variables that has correlation larger than 0.7, which is a number represent high significant correlations. Removed the variables that very likely produced

redundant, I applied RFE on the remaining 9 variables and select first 6 of them has smaller RMSE calculated from cross-validation. The variables I got this time are EVM, LKS, EVS, HGA, RZD and AGR (table 2).

	AGR	HGA	RZD	EVS	TMX	PPE
AGR	1.00000000	0.368253033	-0.30684071	-0.46347403	-0.136273851	0.01683149
HGA	0.36825303	1.000000000	-0.41966850	-0.33610521	0.003778978	0.35402267
RZD	-0.30684071	-0.419668503	1.00000000	0.06512474	-0.458595803	-0.13784933
EVS	-0.46347403	-0.336105212	0.06512474	1.00000000	0.351483343	0.30658697
TMX	-0.13627385	0.003778978	-0.45859580	0.35148334	1.000000000	-0.15847509
PPE	0.01683149	0.354022668	-0.13784933	0.30658697	-0.158475085	1.00000000

Correlation Matrix
by >cutoff 0.55

	EVM	LKS	EVS	HGA	RZD	AGR
EVM	1.0000000	-0.1378772	0.56667384	-0.3841107	0.34223142	-0.6101190
LKS	-0.1378772	1.0000000	-0.29454048	0.6267805	-0.17005466	0.2302363
EVS	0.5666738	-0.2945405	1.00000000	-0.3361052	0.06512474	-0.4634740
HGA	-0.3841107	0.6267805	-0.33610521	1.0000000	-0.41966850	0.3682530
RZD	0.3422314	-0.1700547	0.06512474	-0.4196685	1.00000000	-0.3068407
AGR	-0.6101190	0.2302363	-0.46347403	0.3682530	-0.30684071	1.0000000

Correlation Matrix
by >cutoff 0.7 and
RFE selection

Table 2. Correlation Matrix for two methods

CLUSTERS

Using the two subsets of variables in different method, two final clustered map of St. Lawrence Basin in 3 ecozones are generated by GeoDa. As the result (figure 2), the maps from two different methods are different clustered. There exists a big problem in this map that some clustered regions are not sharing same boundaries. The general k-mean clustering using the centroid of each polygon area to represent the polygon and the centroid has no boundary attribute. In this case, we want area that are contiguous to be recognized as same cluster. In order to do this, I using spdep package in r to map the neighborhood relationship (figure 3, by Appendix 2 relation.R).

Then we want to apply these relationships into clustering. Based on the spatially-contiguous property, we expect to get the final map and see its performance.

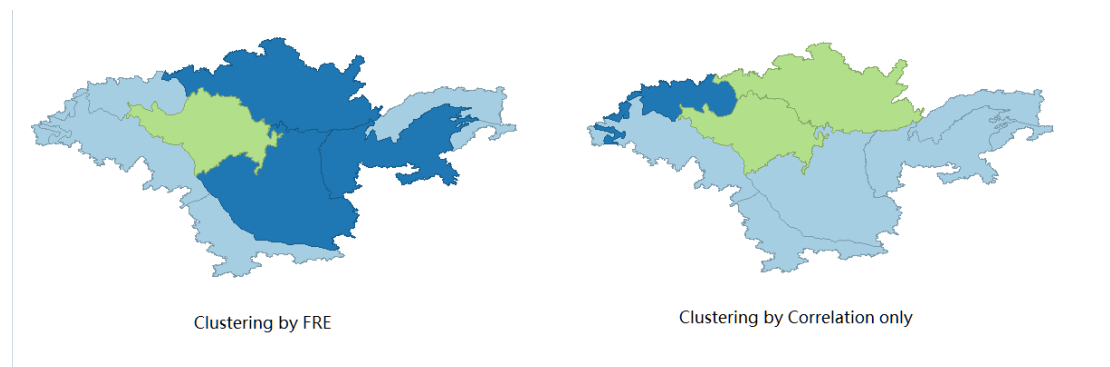


Figure 2. K-mean clustering Without spatially-contiguous

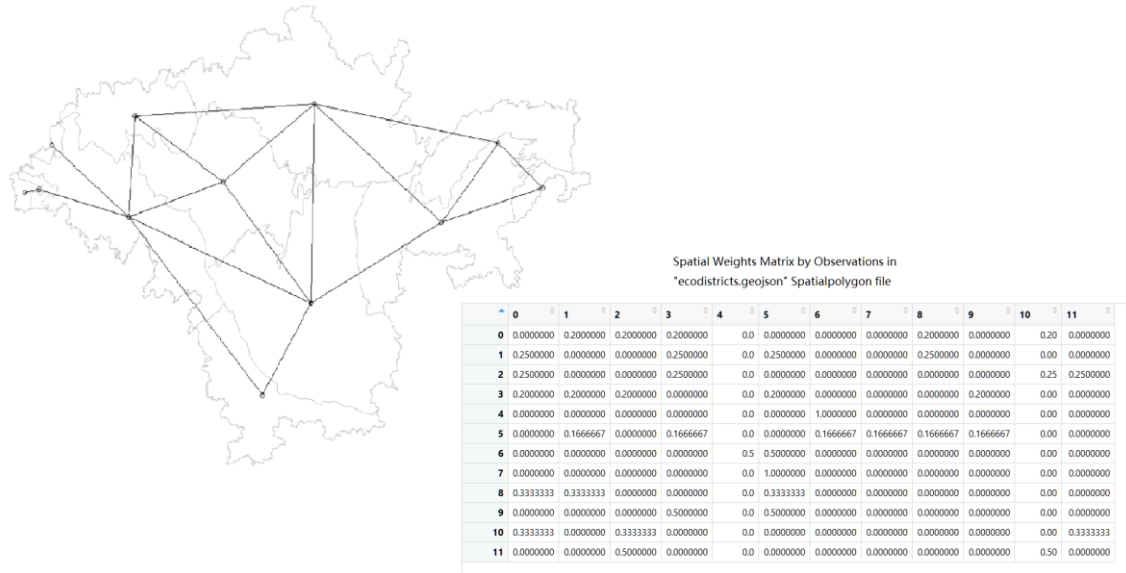


Figure 3. Spatial Contiguous Relationship Map and Table

Using k-mean cluster tool in GeoDa, I got the map of two methods with Spatially Constrained Clustering (figure 4). The result shows that they are clustered in the same way. But the summary of the two method shows that the ratio of between to total sum of squares for FRE is smaller than the one only use correlation selection. This ratio benchmarks the performance of the cluster and shows the level of internal cohesion and external separation. In case, we say selecting AGR, HGA, RZD, EVS, TMX and PPE would be better but the final result under k-mean is same for both result.

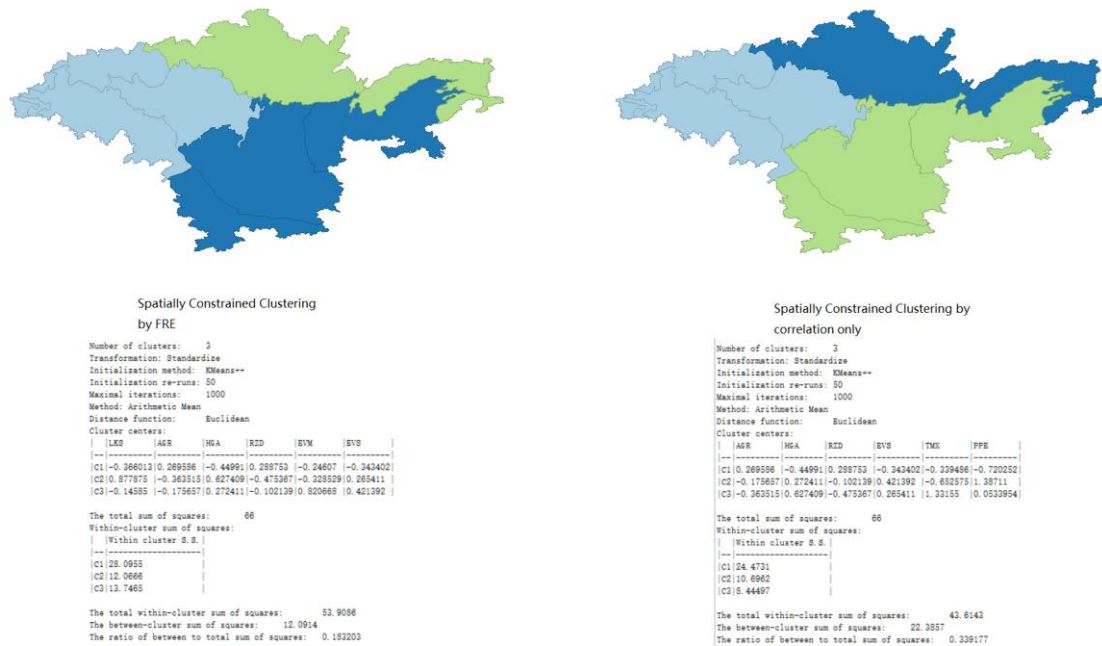


Figure 4. K-mean Spatially Constrained Clustering

In order to get a final map of the 3 spatially-contiguous clusters of Ecodistricts in the St. Lawrence River Basin, I use both GeoDa and R to handle the data. Using GeoDa's k-mean output I was able to add a column named Cluster_cor (the cluster by k-mean using variable select by correlation only). Using the new geojson file, I was able to use rgeos package to help me calculate the spatial center in for this three cluster and using ggplot package to make a map for me (Appendix C. final_map.R). As a result, my spatial center for the three cluster are (-79.85977919,47.6147317), (-77.8121813,48.01607001) and (-78.07348669,46.92191696) (figure 5).

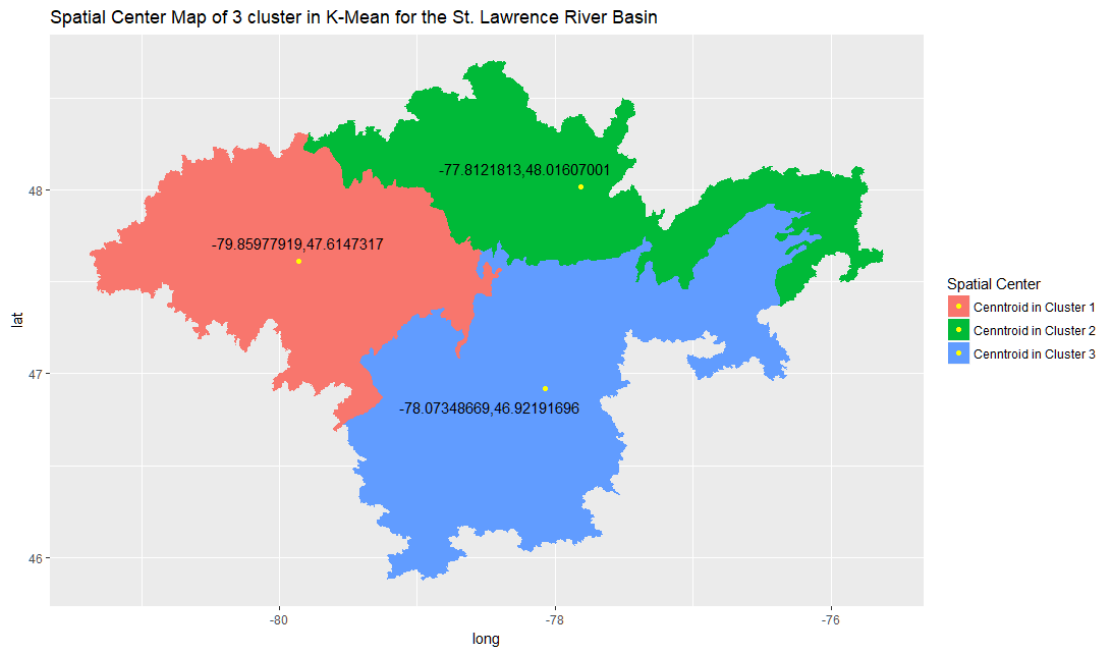


Figure 5. Map of 3 spatially-contiguous clusters of Ecodistricts in the St. Lawrence River Basin with Spatial Center

APPENDIX

APPENDIX A – VARIABLE_SELECTION.R

```
1 options(scipen=999)
2 # load the library
3 library(caret)
4 library(randomForest)
5 library(rgdal)
6 library(dplyr)
7 # load the data
8 sp_data <- readOGR("C:\\Users\\Mei\\OneDrive - University of Toronto\\ggr376\\a3\\ecodistricts.geojson")
9 raw_data <- as.data.frame(sp_data)
10 raw_data <- select(raw_data, -Shape_Area, -Shape_Leng) # remove useless variable
11
12
13 # Simple comparing correlation
14
15 # calculate correlation matrix
16 correlationMatrix <- cor(raw_data)
17 # find attributes that are highly corrected (ideally >0.75)
18 highlyCorrelated <- findCorrelation(correlationMatrix, cutoff=0.55)
19 # print indexes of highly correlated attributes
20 print(highlyCorrelated)
21 # using cor function to double check
22 fixed_data_cor <- raw_data[c(3,4,7,10,13,17)]
23 cor_mat_cor <- cor(fixed_data_cor)
24 corrpplot::corrpplot(cor_mat_cor, cl.pos = "b", tl.pos = "d")
25
26 #####
27 # Recursive Feature Elimination
28
29 # removing significant high correlation variables
30 veryHighlyCorrelated <- findCorrelation(correlationMatrix, cutoff=0.7)
31 print(veryHighlyCorrelated)
32 # select columns that not in the result
33 test_data <- raw_data[c(2,3,4,7,8,9,10,13,17)]
34 # define the control using a random forest selection function with cross-validation RMSE to check error
35 control <- rfeControl(functions=rffuncs, method="cv", number=10)
36 # run the RFE algorithm
37 results <- rfe(test_data[,1:8], test_data[,9], sizes=c(1:6), rfeControl=control)
38 # summarize the results
39 print(results)
40 # list the chosen features
41 predictors(results)
42 # plot the results
43 plot(results, type=c("g", "o"))
44 # check correaltion
45 fixed_data_fre <- select(raw_data, EVM, LKS, EVS, HGA, RZD, AGR)
46 cor_mat_fre <- cor(fixed_data_fre)
47 corrpplot::corrpplot(cor_mat_fre, cl.pos = "b", tl.pos = "d")
```

APPENDIX B – RELATION.R

```
1 # load the library
2 library(spdep)
3 library(rgdal)
4
5 # load the data
6 sp_data <- readOGR("C:\\Users\\Mei\\OneDrive - University of Toronto\\ggr376\\a3\\ecodistricts.geojson")
7 # load the coordinates in the spatial data
8 coords <- coordinates(sp_data)
9 # generate a neighbours list from polygon list
10 sp_data.FOQ <- poly2nb(sp_data, queen=TRUE, row.names=sp_data$FIPSNO)
11 # change the neighbour list to a spatial weighted matrix
12 listw<-nb2mat(sp_data.FOQ)
13 d <- as.data.frame(listw)
14 colnames(d) <- c(0,1,2,3,4,5,6,7,8,9,10,11)
15 table <- as.matrix(d)
16 # using the spatial data as basemap and neibourlist to map the neighbourhood relation ship
17 plot(sp_data, border="grey")
18 plot(sp_data.FOQ, coords, add=TRUE)
19
20
```

```

1 # load the library
2 library(rgdal)
3 library(broom)
4 library(rgeos)
5 library(ggplot2)
6 # load the data
7 sp_data <- readOGR("C:\\Users\\Mei\\OneDrive - University of Toronto\\ggr376\\a3\\modified.geojson")
8 # Change to a data frame but group by cluster
9 spatial_cor <- tidy(sp_data, region = "cluster_cor")
10 # Split the data into 3 groups that in order to calculate their center
11 clus1 <- sp_data[sp_data$cluster_cor == 1,]
12 clus2 <- sp_data[sp_data$cluster_cor == 2,]
13 clus3 <- sp_data[sp_data$cluster_cor == 3,]
14 # get the spatial center for each cluster
15 center1 <- gCentroid(clus1 ,byid=FALSE)
16 center2 <- gCentroid(clus2 ,byid=FALSE)
17 center3 <- gCentroid(clus3 ,byid=FALSE)
18 # change it to data frame so it can be used in ggplot
19 p1 <- data.frame(center1)
20 p2 <- data.frame(center2)
21 p3 <- data.frame(center3)
22 # add coordinate attribute for label
23 p1$coor <- paste(p1$x, p1$y, sep=",")
24 p2$coor <- paste(p2$x, p2$y, sep=",")
25 p3$coor <- paste(p3$x, p3$y, sep=",")
26 # plot the map
27 ggplot() +
28   geom_polygon(data = spatial_cor, aes(long,lat,fill = group))+ # base map
29   geom_point(data= p1, aes(x,y), color="yellow")+ # 3 spatial center points
30   geom_point(data= p2, aes(x,y), color="yellow")+
31   geom_point(data= p3, aes(x,y), color="yellow",show.legend = TRUE)+
32   geom_text(data = p1, aes(x,y+0.1,label = coor),size = 4)+ # 3 labels of coordinate
33   geom_text(data = p2, aes(x-0.4,y+0.1,label = coor),size = 4)+
34   geom_text(data = p3, aes(x-0.4,y-0.1,label = coor),size = 4)+
35   ggtitle("Spatial Center Map of 3 cluster in K-Mean for the St. Lawrence River Basin")+ # title
36   scale_fill_discrete(name = "Spatial Center", # legend
37     breaks=c("1.1", "2.1","3.1"),
38     labels=c("Centroid in cluster 1","Centroid in cluster 2","Centroid in cluster 3"))
39

```

REFERENCE

Adams, M. D., Kanaroglou, P. S., & Coulibaly, P. (2016). Spatially constrained clustering of ecological units to facilitate the design of integrated water monitoring networks in the St. Lawrence Basin. *International Journal of Geographical Information Science*, 30(2), 390-404.

Arthur D, Vassilvitskii S (2007) k-means++: the advantages of careful seeding. *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. pp 1027–1035

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar), 1157-1182.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.