Which Variables Contribute to Air Quality? Evidence From California

Introduction

The World Health Organization (WHO) estimates 7 million that air pollution contributes to the death of 7 million people, anually (WHO 2020). For policymakers, it is thus impportant to understand which factors contribute to air quality, in order to target their efforts to improve it. This study aims to pinpoint the factors that contribute to air quality, using multiple regression. The factors considered in this study are rainfall, population density, income per capita, added value of companies and adjacency to coast. To find which combination of these variables best described their relationship on air quality, we use the better subset selection algorithm (Xiong 2014). Previous work showed this algorithm yields a better fit than a subset without optimization as the result of its monotonicity (Xiong 2014). Our data revealed that whether or not an area is coastal was the only statistically significant variable.

Data

Previous work has suggested the relation between the five chosen variables and air quality. Both natural and anthropogenic events attribute to air quality in the atmosphere. The distribution of air pollution mainly depends on the wind field (Leelőssy et al. 2014), which is quantified by the variable of adjacency to coast and rainfall in this study as they both reflect the wind field's condition. Air quality is also influenced by the production and consumption from society, leading to emmissions (Baklanov, Molina, and Gauss 2016). \

To account for this, our study includes variables on population density, income per capita and added values from companies. We use the econometrics dataset on air quality in California for 1972 (r-project 2020). The dependent variable is an indicator of air quality , the lower the better. The independent variables under study are rainfall(inch), population density(per square mile), income per capita (\$), added value of companies (\$) and adjacency to coast(binary).The dependent variable is an air quality index, consistent of an weighted score for several pollutant concentrations. The lower the index, the better the air quality. The dataset has 30 set of observations, each being a different metropolitan area in California. As the unit of each variable is heterogeneous, we scaled all independent variables to have a mean of 0 and a variance of 1, to ensure fair interpretation of the model coefficents. Scaling also ensures the MM algorithm converges faster.

Looking at the correlations between the variables, the only one that stands out is median income and value added per company with a correlation of 0.89 (see table below). This means we need to be careful with models which include both of these variables, since multicollinearity might lead to a wrong interpretation of the coefficients.

Method

Multiple regression makes a linear combination of several explanatory predictors to predict the outcome of a response variable Y:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon, \tag{1}$$

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon, \tag{2}$$

	Value Added	Rain	Coastal Area	Population Density	Median Income
Value Added	1	−0.149	0.010	0.158	0.890
Rain		1	0.185	0.009	-0.086
Coastal Area			1	0.005	0.170
Population Density				1	0.195
Median Income					1

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
Air Quality	30	104.700	28.028	59	81	126.2	165
Value Added	30	4,188.460	4,630.194	992.900	1,535.775	4,141.375	19,733.800
Rain	30	36.078	13.488	12.630	31.017	42.697	68.130
Coastal Area	30	0.700	0.466	0	0	1	1
Population Density	30	1,728.583	2,827.786	271.590	365.188	1,635.152	12,957.500
Median Income	30	9,476.667	12,499.020	853	3,339.8	8,715	59,460

In (1), Y denotes a random response variable, X denotes random vectors of p predictor variables. This can also be written in matrix form in (2), where $\boldsymbol{y}$ denotes an nX1 column vector, $\mathbf{X}$an include n×(p+1) matrix of thepredictor variables with a first column of 1 for the intercept, and $\boldsymbol{\beta}$ denotes (p+1) vector of weights$[\beta_0, \beta_1, \beta_2, ..., \beta_p]^\top$. In order to find the set of $\boldsymbol{\beta}$ that best fits the model, we need to solve the following minimization problem:

$$RSS(\boldsymbol{\beta}) = \mathbf{e}^\top \mathbf{e} = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}), \tag{3}$$

$$= \mathbf{y}^\top \mathbf{y} + \boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{X}\boldsymbol{\beta} - 2\boldsymbol{\beta}\mathbf{X}^\top \mathbf{y}. \tag{4}$$

The difficult part of minimizing $RSS(\boldsymbol{\beta})$ lies in $\boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{X}\boldsymbol{\beta}$, since computing $\mathbf{X}^\top \mathbf{X}$ is computationally expensive with a large sample. One workaround for this problem is to minimize $RSS(\boldsymbol{\beta})$ using the majorization in minimization algorithm (MM). For this algorithm, we need to find a majorizing function of the form $\lambda\boldsymbol{\beta}^\top \boldsymbol{\beta} - 2\boldsymbol{\beta}^\top \beta$. For this majorizing function, we need $\mathbf{X}^\top \mathbf{X} - \lambda \mathbf{I}$ to be negative semidefinite (nsd). If this condition is unmet, the majorizing function is not convex, and thus we cannot find a minimum. We know that $\mathbf{X}^\top \mathbf{X} - \lambda \mathbf{I}$ is nsd for $\lambda \geq \lambda_{max}$ with λ_{max} is the largest eigenvalue of $\mathbf{X}^\top \mathbf{X}$. \

If we substitute the majorizing function into $RSS(\boldsymbol{\beta})$, and minimize this new function (Ruoying adds notation here)

We try to find the $\boldsymbol{\beta}$ for which RSS is minimized. We do this by choosing with an random initial $\beta_0 \in \mathbb{R}^p$. We improve $\boldsymbol{\beta}$ through iteration k iterations. We define the improvement at step k is defined as the step score:

$$\text{step score} = \frac{RSS(\boldsymbol{\beta_{k-1}}) - RSS(\boldsymbol{\beta_k})}{RSS(\boldsymbol{\beta_{k-1}})} \tag{5}$$

If the step score falls below ϵ, we stop the iteration to minimize $RSS(\boldsymbol{\beta})$. We set the ϵ to a very small number, to ensure that the function is minimized until only very minor improvements can be made. This brings our final solution closer to the optimal solution. \

To judge which model best explains the relationship between the independent and dependent variables, we examine the adjusted Rˆ2. To explain why we use this statistic, let's first examine R^2, which iS defined as $R^2 = 1 - \frac{SSE}{SSTO}$, with SSTO and SSE denoting total sum of squares and error sum of squares correspondingly. This is a measure of the % of variance in the dependent variable that can be explained by the model. When more variables are added, R^2 inherently will increase, but this might due to random chance. To account for this, we look at adjusted R^2, defined as $R^2_\alpha = 1 - (\frac{n-1}{n-p})(\frac{SSE}{SSTO})$, with n denoting our sample size, and p the number of independent variables minus the intercept. \

To find which models have a high R^2_α, we use the better subset algorithm as proposed by (Xiong 2014). In this algorithm, we again use MM to minimize $RSS(\boldsymbol{\beta})$, but only allow models with a number of m variables, as a subset of the original independent variables. This goes as follows: we set m initial random values, with $\beta_0 \in \mathbb{R}^p$. Once we calculated β_k, we sort $|\beta_k|$, and only leave the m largest coefficients as non-zero. We then continue iterating on $|\beta_k|$ with this new set of coefficients. In our case, we let $m \in [1, 5]$, as there were 5 independent variables in our dataset. We then computed for each m the best model according to better subset selection.

Result

The model that best describes the relationship (e.g. with the highest adjusted R^2) is the one with just coastal area, and value added per company. This model does not suffer from multicollinearity, since the value added variable and the coastal area variable are not correlated. Across all the models, the only variable that is statistically significant is the coastal area variable. The table below shows the results of the standardized independent variables on the dependent variable of Y quality. The coefficient in this case tells us that one standard deviation of change in the independent variable, leads to a certain change in the dependent variable.

	Dependent variable:				
	Air Quality				
	(1)	(2)	(3)	(4)	(5)
Coastal Area (1 = coastal, 0 = non-coastal)	-13.73^{***}	-13.83^{***}	-16^{***}	-14.67^{***}	-15.54^{***}
	(-2.54)	(-2.93)	(-3.37)	(-3.00)	(-3.19)
Value Added	-	9.26	-	10.19	4.24
		(0.92)		(0.98)	(0.41)
Median Income	-	-	10.0	-	6.9
			(0.969)		(0.64)
Population Density	-	-	-	-2.66	-3.04
				(-0.58)	(-0.66)
Rain	-	-	-	-	3.38
					0.73
Intercept	104.700^{***}	104.700^{***}	104.700^{***}	104.700^{***}	104.700^{***}
	(21.34)	(23.075)	(23.24)	(23.43)	(23.69)
Observations	30	30	30	30	30
R^2	0.240	0.349	0.359	0.369	0.383
Adjusted R^2	0.21	0.30	0.29	0.27	0.25
Note:					*p<0.1; **p<0.05; ***p<0.01

To make the result on coastal areas a bit more interpretable, If we rescale the coefficient of the coastal area (by adding the mean, and dividing by the standard deviation), we can infer that being an coastal area leads to change of-28.17 points in the air quality index, which means that coastal areas have higher air quality.

Limitations and Future Research

Apart from the limitations to our dataset, we should note several other limitations to our applied method. First, there are other variables that affect air polution that we did not consider. Second, while our method is appropriate for estimating linear relationships, previous research has shown that the effects of certain variables on air pollution is non-linear - one such example is environmental regulation (Liu, Luo, and Wu 2019). Future research should aim to add more areas, years, and variables to our analysis to consider other possibilities.

Conclusion

Our research shows that in California, whether or not a metropolitan area is coastal is a contributor to air quality. It does not indicate other variables as statistically significant. This suggests that coastal areas experience much better air quality than non-coastal areas. This has implications for policymakers. It provides evidence for initiatives that move residents to coastal areas to expose them to enhanced air quality. One such initiative was recently announced in Taiwan (Writer 2020). On the other hand, our results suggest that policies which aim to limit population density or economic activity might be less effective than previously thought, since we found no statistically significant relationship between these variables and air pollution in California.

Reference

Functions

```
# calcRSS: Calculates the residual squared errors for a multiple regression of the form Y = XBeta + e
#
# Parameters:
#   mX: Matrix of n x p (n = observations, p = independent variables)
#   mY: Column matrix of n x 1 dependent variables (n = observations)
#   mBeta: Column Matrix of p x 1 coefficients
#
# Output:
#   ESquared: double, residual squared errors

calcRSS <- function(mX, mY,mBeta){

  # calculate the errors
  mE <-  mY - mX %*% mBeta

  # get errors squared
  ESquared <- t(mE) %*% mE

  # return the residual sum of squared errors
  return(ESquared[1,1])

}

# calcCovar: Calculates the covariance matrix
#
# Parameters:
#   RSS: Residual squared errors
```

```r
#   mXtX: pxp matrix, created from independent variables (X), multiplied with itself
#   n: double, number of observations
#   p: double, number of variables
#
# Output:
#   Covar: matrix, covariance matrix

calcCovar <- function(RSS, mXtX,n, p){

  # est. for sigma squared
  SigmaSquared <- (RSS) / (n - p -1)

  Covar <- SigmaSquared * as.matrix(inv(mXtX))

  return(Covar)

}

# calcSignificance: Calculates the statistical significance of a set of beta's
#
# Parameters:
#   RSS: Residual squared errors
#   mXtX: pxp matrix, created from independent variables (X), multiplied with itself
#   n: double, number of observations
#   p: double, number of variables
#   mBetaEst: matrix of estimated Beta's
#
# Output:
#   dfSignificance: dataframe, containing the results on statistical signficance

calcSignificance <- function(RSS, mXtX, n,p, mBetaEst){

  # get covariance matrix
  mCovar <- calcCovar(RSS,mXtX,n,p)

  # calculate the standard deviations
  stdev <- sqrt(diag(mCovar))

  # define t, which is t-distributed with n-p-1 degrees of freedom
  t <- mBetaEst/stdev
  pval <- 2*pt(-abs(t),df=n-p-1)

  dfSignificance <- data.frame(BetaEst = mBetaEst,
                               stdev = stdev,
                               t = t,
                               pval = pval)

  return(dfSignificance)
}

# calcLargestEigen: Calculates the largest eigenvalue of an matrix of independent variables
#
# Parameters:
```

```
#   mX: Dataframe of n x p (n = observations, p = independent variables)
#
# Output:
#   LargestEigenval: float, largest eigenvalue of said matrix

calcLargestEigen <- function(mX){

  # get the eigenvalues of X
  EigenValX <- eigen(mX)$values

  # from these eigenvalues, get the largest one
  LargestEigenVal <- max(EigenValX, na.rm = TRUE)

  return(LargestEigenVal)

}

# CalcStepScore: Calculates the % improvement between the k-1th and kth set of beta's
#
# Parameters:
#   prevBeta: double, k-1th beta
#   currbeta: double, kth beta
#   mX: Dataframe of n x p (n = observations, p = independent variables)
#
# Output:
#   StepScore; double, % improvement between the RSS of the two sets of beta's

calcStepScore <- function(mX,mY, prevBeta, currBeta){

  # difference in RSS between previous and current set of beta's
  diffRSS <- (calcRSS(mX,mY,prevBeta) - calcRSS(mX,mY,currBeta))

  # divide difference with previous score to get % change
  StepScore <- diffRSS /calcRSS(mX,mY,prevBeta)

  return(StepScore)

}

# calcRsquared
#
# Calculates the r-squared
#
# Parameters:
#   Y: matrix, the true dependent variable
#   Yest: matrix, the predicted dependent variable
#   (optional) adjusted: if True, return adjusted r squared
#   (optional) p: if adjusted is calculated, add number of variables
#
# Output:
#   Rsquared: double, the Rsquared or adjusted Rsquared for a linear model

calcRsquared <- function(mY, mYest, adjusted = FALSE, p=0, n=0){
```

```r
  # standardize Y, and Yest (mean of 0)
  mStandY = mY - mean(mY)
  mStandYest = mYest - mean(mYest)

  # calculate Rsquared
  numerator <- (t(mStandY) %*% mStandYest)^2
  denominator <- (t(mStandY) %*% mY) %*% (t(mStandYest) %*% mStandYest)
  resultRsquared <- (numerator/denominator)

  # if want adjusted R squared,
  if(adjusted){

    adjRsquared = 1 - (((1-resultRsquared)*(n - 1))/(n-p-1))
    resultRsquared <- adjRsquared
  }

  return(resultRsquared)

}

# calcModelMM
#
# Calculates a linear model, using the majorization in minimization (MM) algorithm
#
# Parameters:
#   X: Dataframe of n x p (n = observations, p = independent variables)
#   Y: Dataframe of n x 1 dependent variables (n = observations)
#   e: epsilon, parameter for threshold of improvement after which the algorithm should halt
#   nBeta: number of variables one wants to use
#
# Output:
#   result: dataframe with attributes of the model:
#       - Beta: dataframe, the calculated Beta's
#       - RSS: double, Sum of squared residuals
#       - Yest: dataframe, the predicted Y
#       - Rsquared: double, R^2 for the predicted Y
#       - AdjRsquared: Adjusted Rsquared
#       - Significance results: dataframe with significance results on the beta's
#       - Residuals: dataframe, Y - Yest.
#

calcModelMM <- function(mX,mY,e, nBeta){

  # get number of observations, and number of variables minues the intercept
  n <- nrow(mX)
  p <- ncol(mX) - 1

  # check the user has filled in an appropriate amount of beta's
  if(nBeta > p + 1){
    stop("You want to use more variables than there are in the dataset of independent variables")
  }

  # set the previous beta variable to initial, random beta's
```

```r
prevBeta <- runif(ncol(mX), min=0, max=1)

# calculate X'X
mXtX <- t(mX) %*% mX

# get largest eigenvalue for the square of independent variables
Lambda <- calcLargestEigen(mXtX)

# set initial stepscore to 0, k to 1.
StepScore <- 0
k <- 1

# run while, either if k is equal to 1, or the improvement between k-1th and kth set of beta's is sma
while (k == 1 | StepScore > e ){

  # step to next k
  k <- k + 1

  # calculate beta's for this k
  BetaK <- prevBeta - ((1/Lambda) *  mXtX %*% prevBeta) + ((1/Lambda) * t(mX) %*% mY )

  # sort the beta's based on absolute value, remove the smallest ones to keep m
  absBetaKOrdered <- order(abs(BetaK[,1]), decreasing = T)
  BetaK[!BetaK %in% BetaK[absBetaKOrdered,][1:nBeta]] <- 0

  # new stepscore, % difference in RSS between new Beta's and previous beta's
  StepScore <- calcStepScore(mX,mY,prevBeta,BetaK)

  # assign current beta's to prevBeta variable for next iteration
  prevBeta <- BetaK


}

## Calculate several attributes of the linear model, put in dataframes or doubles

# final Beta's
BetaFinal <- as.matrix(BetaK)

# calculate the RSS of this final est.
RSSBetaK <- calcRSS(mX,mY, BetaK)

# get the est. dependent variables
mYest <- mX %*% BetaFinal

# get the r2 and adjusted r2
Rsquared <- calcRsquared(mY, mYest)
adjRsquared <- calcRsquared(mY,mYest, adjusted = T,  p, n)

# get the residuals
Resi <- mY - mYest

# get the results on significance
```

```
  dfSignificance <- calcSignificance(RSSBetaK, mXtX, n, p, BetaFinal)

  # add these attributes together as a list to make it easily accessible
  result <- list(Beta = BetaFinal,
                 RSS = RSSBetaK,
                 Yest = mYest,
                 Rsquared = Rsquared,
                 adjRsquared = adjRsquared,
                 SignificanceResults = dfSignificance,
                 Residuals = Resi,
                 n = n,
                 p = p)


  return(result)

}

# findModelMM
#
# finds the best linear model, using the MM algorithm, by testing model with 1, 2...up to all variables
#
# Parameters:
#   mX: Matrix of n x p (n = observations, p = independent variables)
#   mY: Matrix of n x 1 dependent variables (n = observations)
#
# Output:
#   results: list with the results for each model version

findModelMM <- function(mX, mY, e){

  # get the number of independent variables used
  nIndVar = ncol(mX) - 1

  # start at m = 1, create empty list to be filled with results
  M = 1
  results <- list()

  # for each m, check the best model and save the results
  while(M <= nIndVar){

    M <- M + 1

    resultM <- calcModelMM(mX, mY, e, M)

    strSave <- paste0("Model with ", M-1, " variable(s)")
    results[[strSave]] <- resultM

  }

  return(results)

}
```

Analysis

```
# load necessary packages
library(matlib)
library(stargazer)
```

```
## Warning: package 'stargazer' was built under R version 4.0.3
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```
library(sjPlot)
```

```
## Warning: package 'sjPlot' was built under R version 4.0.3
```

```
## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car
```

```
library(multiColl)
```

```
## Warning: package 'multiColl' was built under R version 4.0.3
```

```
# load the air quality data
load("Data/Airq_numeric.Rdata")

# set to dataframe
dfAirQ <- data.frame(Airq)

# select dependent variable of air quality
Yair = dfAirQ$airq

# select all other variables as independent variables
Xair = dfAirQ[,-1]

# scale the independent variables, and add an intercept to these
XairScaled <- scale(Xair)
XairIntercept <- cbind(intercept = 1, XairScaled)

# set the data to matrix format
mYair <- as.matrix(Yair)
mXairIntercept <- as.matrix(XairIntercept)
```

```r
# set seed to ensure stability of results
set.seed(0)

# set e small
e <- 0.0000000001


# calculate the model with MM, for 1-5 variables. This contains all the values shown in the paper
compareModelMM <- findModelMM(mXairIntercept, mYair, e)
```


Baklanov, Alexander, Luisa T Molina, and Michael Gauss. 2016. "Megacities, Air Quality and Climate." *Atmospheric Environment* 126: 235–49.

Leelőssy, Ádám, Ferenc Molnár, Ferenc Izsák, Ágnes Havasi, István Lagzi, and Róbert Mészáros. 2014. "Dispersion Modeling of Air Pollutants in the Atmosphere: A Review." *Central European Journal of Geosciences* 6 (3): 257–78.

Liu, Yuncai, Nengsheng Luo, and Shusheng Wu. 2019. "Nonlinear Effects of Environmental Regulation on Environmental Pollution." *Discrete Dynamics in Nature and Society* 2019.

r-project. 2020. "Ecdat: Data Sets for Econometrics." World Health Organization. https://cran.r-project.org/web/packages/Ecdat/index.html.

WHO. 2020. "Air Pollution." World Health Organization. https://www.who.int/health-topics/air-pollution#tab=tab_1.

Writer, Staff. 2020. "Taiwan Mayor Unveils Plan to Move 20,000 Pollution-Hit Residents." *Nikkei Asia*. Nikkei Asia. https://asia.nikkei.com/Spotlight/Environment/Taiwan-mayor-unveils-plan-to-move-20-000-pollution-hit-residents.

Xiong, Shifeng. 2014. "Better Subset Regression." *Biometrika* 101 (1): 71–84.