|  |  |  |
| --- | --- | --- |
| **SIPEI** |  |  |
|  | | |
| **Origen de un episodio** | | |

**ÍNDICE**

1. Introducción 3

2. Cálculo 4

Anexo I: Código de *predESTADmin()* 6

Anexo II: Código de *origen()* 7

Anexo III: Código de *Mapa.vb* 9

# Introducción

En el presente documento se detalla el procedimiento que se utiliza para calcular el origen de un episodio en la actualidad.

En este análisis se han tomado como base las siguientes versiones de los procesos y la aplicación:

* Conexion\_2007-v.01.00
* Prediccion\_2007-v.01.09
* VPR\_INM\_2007-v.3.0.0.0

Donde 2007 hace referencia al año de la aplicación de visualización e indica en los procesos de conexión y predicción que están desarrollados para esa aplicación (aunque no necesariamente en ese año).

# Cálculo

El cálculo del origen de un episodio se hace utilizando la función ***origen()*** definida en *Fuente\_episodio.R*. Y es llamada desde la función *predESTADmin()* que hay definida en *pred\_medas\_min.R*.

Además, para el cálculo se utiliza información histórica del NOX de fondo de cada una de las estaciones de inmisión.

Esta información está almacenada en el archivo *NOx\_FONDO.RData*. donde actualmente se puede encontrar el objeto *NOx\_2006* con los datos de NOX.

Se trata de una matriz de 17 columnas (una por cada estación) y 128.540 filas (datos comprendidos entre el 02/01/2006 11:26 horas y el 29/12/2006 23:34 horas). Estos son datos del año 2006 filtrados para los que se han eliminado instantes en los que se habían producido episodios o fenómenos anormales.

Lo que se hace en *predESTADmin()* es lo siguiente:

* Se almacena en la variable *NOx\_2006* la mediana y la desviación típica de los datos de NOX de cada estación obtenidos de *NOx\_FONDO.RData*.
* Después, cada minuto se comprueba si hay datos nuevos en las estaciones y, en cada una en que los hubiese, se comprueba si existe o no alarma.
* Finalmente, en función de si hay alarma para SO2 y NOX, solo para SO2 o solo para NOX se llama a la función *origen()* para calcular la probabilidad de que el origen del episodio que motivó la alarma esté en la actividad de la central térmica.

Aunque hay que tener en cuenta que cuando hay alarma para ambos se procede de igual forma que cuando la hay solo para SO2 (se le da preferencia en la clasificación).

Lo que se hace en *origen()* es lo siguiente:

* Se simula una distribución Normal con media igual a la mediana de *NOx\_2006* y desviación típica la de *NOx\_2006*, almacenando estos valores de NOX de fondo en una nueva variable (bootstrap paramétrico).

En caso de resultar algún valor inferior a 2 se cambia y se pone 2.

* Se resta de los valores reales de NOX en inmisión este valor de fondo.

En caso de resultar algún valor inferior a 2 se cambia y se pone 2.

* Se calculan los ratios de inmisión para SO2 y NOX tanto para los datos reales actuales como sin el de fondo de NOX. Y esto en función de si lo que interesa es uno u otro (SO2/NOX o NOX/SO2).
* Se calcula el ratio de emisión (SO2/NOX o NOX/SO2).
* Se estiman de manera no paramétrica las densidades para el ratio de emisión y el de inmisión sin fondo.
* Se asigna la probabilidad de que el origen del episodio esté en la central térmica en función de lo próximas que estén estas densidades: a mayor proximidad mayor probabilidad.

Si y son, respectivamente, las funciones de densidad estimadas para los ratios de emisión de la central térmica y de inmisión sin fondo, se trata de obtener:

donde y es el número de puntos estimados para las densidades. En particular, se toma y .

Lo que se hace en la visualización es lo siguiente:

* Se mira en cada instante cual es la estación que tiene el valor más alto de predicción de SO2 y cual de NOX.
* En función de las alarmas se selecciona la estación:
  + Si hay alarma de SO2 y NOX se toma aquella en la que el valor de predicción es mayor entre ellos.
  + Si hay alarma solo de SO2 se toma la que tiene el valor más alto de predicción de SO2.
  + Si hay alarma solo de NOX se toma la que tiene el valor más alto de predicción de NOX.
* Se consulta la probabilidad de que el origen del episodio esté en la central térmica que hay para la estación que haya resultado seleccionada y éste es el valor que se muestra.
* También está contemplado mostrar una probabilidad para la central de ciclo combinado, pero actualmente no se está calculando (aunque aparece en pantalla “Prob. Ciclo:” sin que se ponga valor alguno).

# Anexo I: Código de *predESTADmin()*

…

# Datos para clasificación: NOx de fondo

load("C:\\fprimag\\Endesa\\Entornos\\2007\\Desarrollo\\Prediccion\\programas\\x\_medas\\conf\\NOx\_FONDO.RData")

NOx\_2006=rbind(apply(NOx\_2006,2,median,na.rm=T),apply(NOx\_2006,2,sd,na.rm=T))

dimnames(NOx\_2006)[[1]]=c("Median","SD")

…

#Calcular el origen de un episodio

if(length(ind)!=0){

for (i in ind){

if (mensaje[n\_long+n\_pred,1,i]>1 & mensaje[n\_long+n\_pred,2,i]>1){

porcentaje[1,1,i]=origen(i,"SO2\_MH",120,inmi\_medH[n\_long,1:2,i],350,emi\_real[(n\_long-119):n\_long,1:2,],NOx\_2006)}

else{

if (mensaje[n\_long+n\_pred,1,i]>1){

porcentaje[1,1,i]=origen(i,"SO2\_MH",120,inmi\_medH[n\_long,1:2,i],350,emi\_real[(n\_long-119):n\_long,1:2,],NOx\_2006)}

else{

if (mensaje[n\_long+n\_pred,2,i]>1){

porcentaje[1,1,i]=origen(i,"NOx\_MH",120,inmi\_medH[n\_long,1:2,i],200,emi\_real[(n\_long-119):n\_long,1:2,],NOx\_2006)}

else {porcentaje[1,1,i]=-1}}}

fporcentaje[1,1,i]=formato(porcentaje[1,1,i],9,2)}}

…

# Anexo II: Código de *origen()*

# --------------------------------------------------- #

# ---------------- Rutina para calcular ------------ #

# -------------- el origen de un episodio ----------- #

# --------------------------------------------------- #

origen=function(est,variable,ndatos,datos\_inmi,lim\_var,datos\_emi\_ct,NOx\_2006){

######################################################################

# Estación que nos interesa (onde está a ocorrer o episodio)

#est

# Variable que nos interesa (SO2\_MH se o episodio é de SO2, NOx\_MH se é de NOx)

#variable

# Número de datos cos que se van facer os cálculos (en principio 120)

#ndatos

# Datos de inmisión (vector cos valores actuales de SO2 e NOx)

#datos\_inmi

# Límites para a variable que nos interesa

#lim\_var

# Datos de emisión da CT (array con 2ª dim SO2 e NOx, e 3ª dim os catro grupos)

#datos\_emi\_ct

########################################################################

#Reducir información a cocientes

nome=ifelse(variable=="SO2\_MH","SO2/NOx","NOx/SO2")

#Inmisión

datos\_inmi[datos\_inmi<=0]=NA

#Simulación de inmisión de NOx de fondo: bootstrap paramétrico (normal)

NOxF=rnorm(ndatos,NOx\_2006[1,est],NOx\_2006[2,est])

NOxF[NOxF<2]=2

NOx\_sinF=datos\_inmi[2]-NOxF #NOx\_actual-NOx\_fondo

NOx\_sinF[NOx\_sinF<2]=2

#Ratios

if (variable=="SO2\_MH"){

ratio\_inmi=datos\_inmi[1]/datos\_inmi[2] #SO2\_actual/NOx\_actual

ratio\_inmi\_sinF=datos\_inmi[1]/NOx\_sinF} #SO2\_actual/NOx\_sin\_fondo

else{

ratio\_inmi=datos\_inmi[2]/datos\_inmi[1]

ratio\_inmi\_sinF=NOx\_sinF/datos\_inmi[1]}

#Emisión

datos\_emi\_ct[datos\_emi\_ct<=0]=NA

aux=apply(datos\_emi\_ct,c(1,2),sum,na.rm=T)

aux[aux==0]=NA

if (variable=="SO2\_MH"){

ratio\_emi\_ct=aux[,1]/aux[,2]} #SO2/NOx

else{

ratio\_emi\_ct=aux[,2]/aux[,1]} #NOx/SO2

#Densidades

auxp=seq(0,1,l=1000)

limx=range(c(ratio\_inmi,ratio\_inmi\_sinF,ratio\_emi\_ct),na.rm=T)

##Necesitamos al menos dos datos de emisión y dos de inmisión

if(length(ratio\_emi\_ct[!is.na(ratio\_emi\_ct)])>1 & length(ratio\_inmi\_sinF[!is.na(ratio\_inmi\_sinF)])>1){

# Densidad de emisión

dens\_emi\_ct=density(ratio\_emi\_ct,from=limx[1],to=limx[2],na.rm=T)$y

# Densidad de inmisión

dens\_inmi=density(ratio\_inmi\_sinF,from=limx[1],to=limx[2],na.rm=T)$y

# Cálculo de probabilidad de CT

aux=dens\_inmi

aux=apply((aux-outer(dens\_emi\_ct,auxp,"\*"))^2,2,sum)

p\_ct=auxp[min(which(min(aux)==aux))]\*100}

else{p\_ct=-1}

return(p\_ct)}

# Anexo III: Código de *Mapa.vb*

…

'Calcular las estaciones con los máximos para clasificación

'para SO2

aux = 0

For I = 0 To n\_est

If matriz\_prediccion(I, n\_min\_real + 30, 0) > aux Then

aux = matriz\_prediccion(I, n\_min\_real + 30, 0)

indice\_max\_clasificacion(0) = I

End If

Next I

'para el NOx

aux = 0

For I = 0 To n\_est

If matriz\_prediccion(I, n\_min\_real + 30, 1) > aux Then

aux = matriz\_prediccion(I, n\_min\_real + 30, 1)

indice\_max\_clasificacion(1) = I

End If

Next I

'Probabilidad de origen de episodio

If Trim(mensaje(0)) <> "No hay alarma" Then

If Trim(mensaje(1)) <> "No hay alarma" Then

If matriz\_prediccion(indice\_max\_clasificacion(0), n\_min\_pred, 0) < matriz\_prediccion(indice\_max\_clasificacion(1), n\_min\_pred, 1) Then

I = indice\_max\_clasificacion(1)

Else

I = indice\_max\_clasificacion(0)

End If

Else

I = indice\_max\_clasificacion(0)

End If

Else

If Trim(mensaje(1)) <> "No hay alarma" Then

I = indice\_max\_clasificacion(1)

Else

I = -1

End If

End If

…

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'Leer clasificación de origen de episodio

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

Private Sub lee\_clasificacion()

Dim I, J As Integer

Dim fichero As String

Dim basura As String

Dim aux As Double

fichero = directorio & "\clasificacion.dat"

'Valores por defecto para los datos y los mensajes de alarma

For I = 0 To n\_est

For J = 0 To 1

clasif(I, J) = -1

Next J

Next I

If Not IO.File.Exists(fichero) Then

MsgBox("No se ha encontrado el fichero " & fichero & ".")

Else

Try

Using sr As IO.StreamReader = New IO.StreamReader(fichero)

basura = sr.ReadLine() 'linea de títulos

'lectura estaciones

For I = 0 To n\_est

basura = sr.ReadLine()

For J = 0 To 1

aux = Val(Mid(basura, 6 + (9 \* J), 6))

If aux > -1 And aux < 101 Then

clasif(I, J) = aux

End If

Next J

Next I

sr.Close()

End Using

Catch e As Exception

MsgBox("Error en la lectura del fichero " & fichero & "." & e.Message)

End Try

End If

End Sub