Informe

Carmen Calvo Olivera

29 de mayo de 2020

Resumen

En este documento se recogen los resultados obtenidos de la aplicación de diferentes técnicas de aprendizaje automático a nuestro conjunto de datos, cuyo fin es conseguir un modelo capaz de... obtener en tiempo real el índice de verosimilitud de una predicción meteorológica respecto a los valores reales de las mediciones realizadas dentro de la CHE (Cuenca Hidrográfica del Ebro).

Para ello, se ha llevado a cabo la creación de un conjunto de datos a partir de predicciones obtenidas a partir del modelo WRF (Weather Research and Forecasting).

1. Introducción

En la actualidad, ...

2. Dataset

La preparación de los datos supone una parte esencial a la hora de trabajar con aprendizaje automático. En esta sección de describre brevemente tanto la obtención como el tratamiento (o preprocesamiento) de las predicciones a partir de las cuales se obtienen los conjuntos de datos utilizados para el entrenamiento y validación de los distintos clasificadores.

En primer lugar, cabe destacar uso de dos grandes conjuntos de datos, uno para *train* y el otro para *validation*.

- *train_dataset*: este conjunto de datos abarca desde febrero (a la expera de añadir enero) de 2015 hasta diciembre de 2015.
- *validation_dataset*: que abarca desde enero de 2016 hasta diciembre de 2016.

2.1. Obtención

Los datos de las predicciones se obtinen a partir del modelo WRF, un modelo meteorológico numérico de mesoescala no hidrostático, utilizado para finalidades tanto de predicciones operativas en tiempo real como para investigación atmosférica.

Los datos de inicialización del modelo WRF provienen de análisis y predicciones de otros NWP cuyo formato debe ser GRIB1 o GRIB2. En nuestro caso, todos los ficheros fuente han sifo obtenido del Research Data Archive del NCAR (Centro Nacional de Investigación Atmosférica).

Posteriormete se llevan a cabo todos los pasos incluidos dentro del flujo de procesamiento y, a continuación, comienza la simulación meterorológica. Tras todo esto, obtenemos una serie de ficheros del tipo "*wrfout_d0X_yyyy-mm-dd_hh:mm:ss*"donde X es el número del dominio, y la cadena yyyy-mm-dd_hh:mm:ss representa la fecha y hora de la primera salida guardada en el fichero. Todos estos ficheros son obtenidos en formato NetCDF, un formato de archivo destinado a almacenar datos científicos multidimensionales (variables) como la temperatura, la humedad ola presión.

En nuestro caso, se hará uso de los ficheros horarios para el dominio 2 representado en la figura XX y definido previamente en los ficheros de configuración para 24h, es decir, para el día 1 de enero de 2016 se hace uso de 24 ficheros que van desde "*wrfout_d02_2016-01-02_01:00:00*" hasta "*wrfout_d02_2016-02-03_00:00:00*".

2.2. Procesamiento

Tras la obtención de las predicciones, se llevan a cabo una serie de pasos para el tratamiento y finalmente obtención de nuestro dataset:

- **Filtración de características**: el primer paso realizado es filtrar las variables que obtenemos de las predicciones del WRF y quedarnos con una seleccion de variables detalladas en la tabla Tabla 1. Mediante el uso de la libreria de python *wrf-python* te hace un tratamiento de los datos para, a partir de una prediccion horaria (por qué horaria (?)), obtener un único fichero .nc diario con toda la información que necesitamos. A continuación se muestra la cabecera de unos de los ficheros filtrados:

```
[ccalvo@frontend1 nc]$ ncdump -h 2016-04-12.nc
netcdf \2016-04-12 {
dimensions:
        south_north = 78 ;
        west_east = 123 ;
        time = 24 ;
variables:
        float XLAT(time, south_north, west_east) ;
        float XLONG(time, south_north, west_east) ;
        float HGT(time, south_north, west_east) ;
        float RAINC(time, south_north, west_east) ;
        float RAINNC(time, south_north, west_east) ;
        string DATE(time) ;
        float TIMESTAMP(time) ;
        float QVAPOR_500(time, south_north, west_east) ;
```

```
float QVAPOR_700(time, south_north, west_east) ;
float QVAPOR_850(time, south_north, west_east) ;
float QCLOUD_500(time, south_north, west_east) ;
float QCLOUD_700(time, south_north, west_east) ;
float QCLOUD_850(time, south_north, west_east) ;
float QRAIN_500(time, south_north, west_east) ;
float QRAIN_700(time, south_north, west_east) ;
float QRAIN_850(time, south_north, west_east) ;
float QICE_500(time, south_north, west_east) ;
float QICE_700(time, south_north, west_east) ;
float QICE_850(time, south_north, west_east) ;
float QSNOW_500(time, south_north, west_east) ;
float QSNOW_700(time, south_north, west_east) ;
float QSNOW_850(time, south_north, west_east) ;
float QGRAUP_500(time, south_north, west_east) ;
float QGRAUP_700(time, south_north, west_east) ;
float QGRAUP_850(time, south_north, west_east) ;
float T_500(time, south_north, west_east) ;
float T_700(time, south_north, west_east) ;
```

Tabla 1. Variables utilizadas para la creación de un dataset

Variable	**Descripción**
DATE	Fecha de la predicción
TIMESTAMP	
XLAT	
XLONG	
HGT	
RAINC	Precipitación convectiva
RAINNC	Precipitación no convectiva
T_500hPa	Temperatura a diferentes presiones
T_700hPa	
T_850hPa	
QVAPOR_500	Razón de mezcla
QVAPOR_700	
QVAPOR_850	
QCLOUD_500	
QCLOUD_700	
QCLOUD_850	
QRAIN_500	
QRAIN_700	

. . . continúa en la siguiente página

Variable	Descripción
QRAIN_850	
QICE_500	
QICE_700	
QICE_850	
QSNOW_500	
QSNOW_700	
QSNOW_850	
QGRAUP_500	
QGRAUP_700	
QGRAUP_850	

- **Creación de los csv**: el siguiente paso es la creación de un dataset, para el cuál convertiremos los ficheros NetCDF a formato CSV y añadiremos algunas variables (o etiquetados de los datos). En la tabla Tabla 2 se recogen las variables añadidas a nuestros ficheros.

Tabla 2. Variables añadidas a los datasets

Variable	Descripción
PRECIPITACION_WRF	Precipitación acumulada de la predicción del WRF (RAINC + RAINNC)
PRECIPITACION	Precipitación acumulada real (obtenida a partir pluviómetros de la CHE)
LLUVIA_WRF	Variable binaria para la predicción del WRF (0 → No precipitación y 1 → precipitación)
LLUVIA	Variable binaria para la CHE (0 → No precipitación y 1 → precipitación)
RANGO_WRF	Rango para la predicción del WRF ([0 - 14]*)
RANGO	Rango para la CHE ([0 - 14]*)

* Rangos (mm): 0.1,1.,1.5,2.5,5.,10.,15.,20.,25.,30.,40.,50.,80. Rangos (representación): [0 - 14]

- **Creación de los csv**: como último paso, y previo al entrenamiento de todos los modelos, se crea un csv único con los días que deseamos incluir en nuestros datasets.

3. Resultados

A continuación, se detallana los resultados obtenidos, los cuales se han organizado de la siguiente manera. Todos ellos cuentan con la tasa de acierto obtenida en el conjunto de datos destinado a la validación tanto en el train como en el test y posteriormente con el error cuadrático medio (ECM):

- Binaria: en primer lugar se muestran los resultados obtenido para la predicción de la variable binaria con 3 escenarios distintos.
  - Todas las variables incluidas en el CSV a excepción de aquellas consideradas como etiquetas. Ver Tabla 2. Resultados: Tabla 3
  - Todas las variables del punto anterior eliminando RAINC y RAINNC. Resultados: Tabla 4
  - Únicamente con las coordenadas y las variables RAINC y RAINNC. Resultados: Tabla 5
- Rango: en este caso se realiza la predicción con los rangos definidos con los 3 escenarios anteriores.
  - Todas las variables incluidas en el CSV a excepción de aquellas consideradas como etiquetas. Ver Tabla 2. Resultados: Tabla 6
  - Todas las variables del punto anterior eliminando RAINC y RAINNC. Resultados: Tabla 7
  - Únicamente con las coordenadas y las variables RAINC y RAINNC. Resultados: Tabla 8

Error cuadrático medio (CHE - WRF)= 0.34860156086592764

Tabla 3. Binaria. Todas características

Variable	Train	Test	ECM (CHE - model)
MPL	0.736403	**0.756870**	0.243130
LogisticRegression	0.728701	**0.738289**	0.261711
QDA	0.703161	**0.731465**	0.268535
NeuralNetwork	0.743430	0.731454	0.268546
LDA	0.740250	0.731177	0.268823
OVR	0.688272	0.727229	0.272771
AdaBoost	0.770249	0.721126	0.278874
RandomForest	0.696645	0.712761	0.287239

. . . continúa en la siguiente página

Variable	Train	Test	ECM (CHE - model)
DecisionTree	0.773973	0.711559	0.288441
NaiveBayes	0.732275	0.705945	0.294055
KNeighbors	0.892685	0.645533	0.354467
SGDClassifier	0.393367	0.358522	0.641478

Tabla 4. Binaria. Todas características sin RAIN

Variable	Train	Test	Error cuadrático medio
SGDClassifier	0.686392	**0.715450**	0.284550
LDA	0.734469	**0.711946**	0.288054
QDA	0.688710	**0.704254**	0.295746
LogisticRegression	0.709877	0.698761	0.301239
DecisionTree	0.742524	0.698344	0.301656
NeuralNetwork	0.715451	0.694780	0.305220
RandomForest	0.669924	0.690709	0.309291
OVR	0.668298	0.690013	0.309987
MPL	0.718232	0.675204	0.324796
AdaBoost	0.730053	0.672329	0.327671
NaiveBayes	0.709254	0.630211	0.369789
KNeighbors	0.862658	0.617016	0.382984

Tabla 5. Binaria. Solo RAIN

Variable	Train	Test	Error cuadrático medio
MPL	0.741252	**0.753171**	0.246829
NeuralNetwork	0.744667	**0.750569**	0.249431
LogisticRegression	0.714578	**0.749263**	0.250737
NaiveBayes	0.712043	0.748209	0.251791
LDA	0.703919	0.741426	0.258574
DecisionTree	0.764922	0.719928	0.280072
AdaBoost	0.762199	0.717789	0.282211
RandomForest	0.753338	0.715067	0.284933
QDA	0.667406	0.687748	11.073562
KNeighbors	0.813913	0.678051	0.321949
OVR	0.426253	0.407584	0.592416

. . . continúa en la siguiente página

Variable	Train	Test	Error cuadrático medio
SGDClassifier	0.333523	0.313059	0.686941

Tabla 6. Rango. Todas features

Variable	Train	Test	Error cuadrático medio
OVR	0.669560	0.689178	9.145259
MPL	0.669085	0.688222	9.102521
NeuralNetwork	0.669087	0.688164	9.079000
RandomForest	0.667468	0.687758	11.070078
LogisticRegression	0.668037	0.684587	8.619534
SGDClassifier	0.664522	0.677533	8.401055
LDA	0.664978	0.675906	9.561050
AdaBoost	0.666912	0.655973	9.656310
DecisionTree	0.680278	0.651697	8.835721
NaiveBayes	0.651177	0.620479	8.792356
KNeighbors	0.850602	0.541724	10.216860
QDA	0.414040	0.394458	64.326082

Tabla 7. Rango. Todas features sin RAIN

Variable	Train	Test	Error cuadrático medio
NeuralNetwork	0.667406	0.687749	11.072704
MPL	0.667406	0.687748	11.073562
RandomForest	0.667415	0.687747	11.073196
LogisticRegression	0.665773	0.685467	10.581673
AdaBoost	0.667699	0.677845	11.002875
LDA	0.662489	0.675684	10.928471
DecisionTree	0.673827	0.662918	10.344793
SGDClassifier	0.654264	0.653064	10.036488
NaiveBayes	0.636050	0.564641	10.302585
KNeighbors	0.809269	0.535653	11.650298
QDA	0.303942	0.293877	83.531754
OVR	0.262102	0.229779	11.982337

Tabla 8. Rango. Solo RAIN

Variable	Train	Test	Error cuadrático medio
RandomForest	0.667406	0.687748	11.073562
QDA	0.667406	0.687748	11.073562
OVR	0.668867	0.687748	9.380225
NeuralNetwork	0.668598	0.687300	9.158468
MPL	0.668488	0.686898	9.082758
LogisticRegression	0.667769	0.683597	8.929119
LDA	0.664579	0.681965	10.134818
AdaBoost	0.668623	0.677344	9.771984
DecisionTree	0.670085	0.674760	9.498254
NaiveBayes	0.665110	0.673433	8.316886
KNeighbors	0.786454	0.592601	9.774854
SGDClassifier	0.148744	0.135874	10.248119

Por último, y como prueba de concepto, se ha llevado a cabo el entrenamiento de un único modelos, en este caso "DecissionTree", con diferente número de características para así intentar obtener mejores resultados. La Tabla 9 recoge los resultados obtenidos.

En este caso, se han considerado características todas exceptuando las consideradas etiquetas, haciendo un total de 26 características y se ha utilizado la variable del rango como nuestra variable a predecir:

Tabla 9. Resultados para k características

k	Train	Test	Error cuadrático medio
20	0.773973	0.713114	0.286886
21	0.773973	0.713114	0.286886
19	0.773356	0.713104	0.286896
22	0.773973	0.711559	0.288441
23	0.773973	0.711559	0.288441
4	0.771234	0.707060	0.292940
15	0.772258	0.695624	0.304376
16	0.772258	0.695624	0.304376
17	0.772258	0.695624	0.304376
18	0.773356	0.695624	0.304376
5	0.771229	0.695305	0.304695
6	0.771229	0.695305	0.304695
7	0.771229	0.695305	0.304695

. . . continúa en la siguiente página

k	Train	Test	Error cuadrático medio
8	0.771229	0.695305	0.304695
11	0.772258	0.690686	0.309314
3	0.727233	0.687748	0.312252
9	0.772258	0.312252	0.687748
10	0.772258	0.312252	0.687748
12	0.772258	0.312252	0.687748
13	0.772258	0.312252	0.687748
14	0.772258	0.312252	0.687748

4. Trabajo futuro

Como trabajo inmediato posterior, se pretende complementar los resultados con la siguiente información:

- Incluir enero de 2015 en el *dataset* de entrenamiento.
- Realizar predicciones con lo mm.
- Hacer pruebas para obtener el mejor score para distintos KBest.

5. Referencias