Aprendizaje Automático. Proyecto final

Manuel Herrera Ojea
53583380G

Ismael Marín Molina
AAAAAAAAB

Contents

1 Descripción del problema **1**

2 Modelos empleados **2**

3 Desarrollo de la clasificación **3**

4 Desarrollo de la regresión **4**

1 Descripción del problema

(Por ahora el texto es el de la práctica anterior)

En esta práctica el objetivo ha sido estudiar un modelo de aprendizaje automático por regresión y uno por clasificación. Para ambos nos hemos servido de las funciones y clases facilitadas por el framework scikit-learn, que tiene ya implementados los modelos de aprendizaje que hemos decidido utilizar.

Los conjuntos de aprendizaje, validación y prueba han sido tomados en ambos casos de bases de datos Machine Learning Repository alojada en `archive.ics.uci.edu`.

Para regresión hemos utilizado Airfoil Self-Noise Data Set, que consta de 1503 ejemplos con cinco atributos (frecuencia, ángulo de incidencia, longitud del alerón, velocidad y dureza) de los que nos servimos para predecir el valor en decibelios del sonido generado por el tipo de viento con el que se ha generado cada muestra.

Para clasificación hemos utilizado Optical Recognition of Handwritten Digits Data Set, que consta de 5620 ejemplos con 64 atributos cada uno. En este caso, cada atributo corresponde a la suma de píxeles encendidos de una región de 4×4 píxeles de la imagen original de cada dígito.

2 Modelos empleados

Para la clasificación nos hemos servido de la regresión logística, implementada en scikit-learn bajo la clase `LogisticRegression`. Hemos utilizado este modelo porque, pese a ser un modelo de regresión logística, nos informa de la certeza que tiene el modelo de que un dato de entrada nuevo pertenezca a cada una de las clases. Esto es conveniente porque podemos ver la decisión que ha tomado el modelo a la hora de realizar una clasificación, y si ha habido alguna otra clase bajo la que podría haber etiquetado el dato. Al final tendremos solo una clase, pero podremos ver, de forma adicional, un valor de certeza de pertenencia a esa clase.

Para la regresión hemos contrastado los modelos Lasso y Ridge, implementados bajo los homónimos `Lasso` y `Ridge`. Ambos intentan reducir la dimensionalidad del problema. Ridge hace que algunos atributos tiendan a valores pequeños, mientras que Lasso permite que algunos atributos valgan exactamente 0, eliminándose con ello por completo su participación en el valor predicho final. En ambos casos es útil este comportamiento, pues una reducción de dimensionalidad, bajo un score aceptable, arroja información sobre los datos de la muestra, en lugar de únicamente una predicción.

Como optimizador de hiperparámetros nos hemos servido de la clase `GridSearchCV`. Hemos contrastado los resultados que arrojan los modelos escogidos si modificamos como hiperparámetros la penalización y el valor de C para clasificación y el grado polinómico y el valor de α para regresión, equivalente este último a C^{-1}. Como penalizadores hemos utilizado L_1 y L_2, típicamente utilizados en, respectivamente, los modelos Lasso y Ridge.

El grado polinómico nos dice qué potencia máxima de los datos de entrada podemos utilizar en la clasificación. Aun siendo mayor que 1 este valor podemos estar frente a un modelo lineal. Al elegir un grado $n > 1$ se añaden atributos a cada dato del conjunto de aprendizaje, equivalentes a las potencias i-ésimas de sus valores, hasta llegar a sus potencias n-ésimas. Tras ello se aplica sobre el conjunto de datos resultante un modelo lineal. Aplicar tanto Lasso como Ridge a unos datos transformados de esta manera es especialmente interesante, pues eliminan las potencias de los atributos originales que no son pertinentes para la predicción.

El valor de C indica cuán suavizada será la superficie de decisión. Un valor alto de C hará que se clasifique mejor una mayor cantidad de datos del conjunto de aprendizaje, mientras que un valor más bajo simplifica el modelo, sacrificando la correcta clasificación de más datos para buscar en un espacio con menos altibajos y resultar en una mayor generalización. Produce

un efecto equivalente a la regularización de los coeficientes finales.

Tanto L_1 como L_2 añaden un término adicional al cálculo del error: el primero añade como penalización el valor absoluto de la magnitud de los coeficientes, mientras que el segundo añade su cuadrado. Si e es nuestra función de error original, entonces

$$e_{L_1} = e + \sum_i |x_i|,$$

$$e_{L_2} = e + \sum_i x_i^2.$$

Como métrica para contrastar nos hemos servido del coeficiente de determinación, que se calcula de la siguiente forma:

$$R^2 = 1 - \frac{\text{MSE}}{\sigma^2 N}$$

donde MSE es el error cuadrado medio, o por cuánto suele equivocarse nuestro modelo al predecir:

$$\frac{1}{N}\sum_i (y_{pred,i} - y_{true,i})^2$$

Esta métrica nos dice qué porcentaje de la varianza de nuestro conjunto de de aprendizaje es explicada por nuestro modelo. Nuestro objetivo es obtener una puntuación alta, cercana a 1, pero sin que llegue a él. Partimos de que parte de la varianza de los datos que estamos utilizando proviene del ruido que estos presenten, pues no podemos tener la certeza de que no se haya generado ruido al tomar los datos. No nos intenteresará, por tanto, terminar obteniendo un modelo que explique el 100 % de la varianza de nuestro conjunto, pues estaríamos pegándonos demasiado a los datos de aprendizaje y estaríamos perdiendo capacidad de generalización.

3 Desarrollo de la clasificación

Los hiperparámetros con los que hemos probado el modelo de regresión logística para la clasificación han sido $\{L_1, L_2\}$ para la penalización y $\{0.1, 0.5, 0.9\}$ para C. Tras haber probado todos los casos posibles que sus combinaciones generan, hemos visto que la mejor combinación sobre el conjunto deaprendizaje ha sido una penalización de tipo L_1 y un $C = 0.9$, resultando en un $R^2 =$

0.9634, superior al error medio de 0.94 que solemos tener las personas al clasificar dígitos escritos a mano. Que estos parámetros hayan ofrecido los mejores resultados tiene sentido, por definir un espacio de soluciones más laxo que el original pero sin eliminar demasiada varianza en la muestra.

Para una visualización de los errores que comete nuestro clasificador más clara que una tabla numérica hemos utilizado dos estructuras de representación adicionales: la matriz de confusión y la curva ROC.

Con la matriz de confusión podemos ver de forma muy clara e intuitiva qué porcentaje de datos de cada clase se han clasificado bien y qué mal. En la matriz, el elemento m_{ij} hace referencia al porcentaje de datos de la clase i de los que se ha predicho que pertenecen a la clase j. Así, un buen clasificador generará una matriz de confusión que tienda a una matriz diagonal, mientras que un mal clasificador colocará valores altos en posiciones de la matriz distintos de la diagonal. La siguiente imagen muestra la matriz de confusión que ha generado nuestro modelo.

La curva ROC muestra la proporción de ejemplos bien clasificados conforme incrementan los ejemplos mal clasificados tenidos en cuenta. Este tipo de gráfica es interesante porque el área que encierra bajo su curva es directamente proporcional al porcentaje de aciertos del modelo utilizado. Si la curva coincide con la diagonal estamos en un caso equivalente a un clasificador aleatorio, donde una predicción realizada por tal modelo no sería distinta al lanzamiento de una moneda. Cuanto mayor sea el área entre la curva y la diagonal, mayor será la media de acierto de nuestro modelo. En casos en los que la curva está por debajo, nuestro modelo habrá aprendido la función complementaria a la que buscamos, por lo que solo tendríamos que realizar un cambio de signo. La siguiente imagen muestra las curvas ROC que genera el modelo de clasificación que hemos entrenado.

4 Desarrollo de la regresión

Los hiperparámetros con los que hemos probado los modelos para la regresión logística han sido $\{1, \cdots, 7\}$ para el grado polinómico y un α que varía entre 10^{-2} y 10^{-16}. Tras haberlo probado con todas las combinaciones hemos visto que los mejores resultados los han arrojado, tanto con Lasso como con Ridge, un $\alpha = 10^{-4}$, y un grado polinómico de 5 para Lasso y de 4 para Ridge. De los dos, Lasso ha obtenido mejores resultados con esa mejor combinación de hiperparámetros, obteniendo con él una tasa de clasificación de 0.83, frente

a la de 0.79 obtenida con Ridge.

Tiene sentido que α tenga un valor así, equivalente a un $C = 10^4$, pues supone, conforme disminuye el valor de α, que se van tomando en cuenta cada vez más características de los datos originales transformados por el grado polinómico. Ha de llegar un momento en el que tomar más en cuenta sea caer en sobreajuste. La siguiente gráfica muestra cómo varía la varianza explicada conforme disminuye el valor de α.

El grado polinómico es muy dependiente del problema. En este caso, tras ver que un ajuste lineal sobre los datos originales no surtían un buen efecto, probamos a incrementar lentamente el grado de los datos de entrada. No conviene aumentar demasiado el grado, pues aumenta con ello la complejidad de la clasificación, y es una buena regla general optar por un modelo sencillo, pero no siempre existen relaciones lineales entre los datos de entrada y la predicción deseada. La siguiente imagen muestra cómo varía la varianza explicada conforme aumentamos el grado polinómico de los datos de entrada.