Proyecto Final: Online News Popularity

Aprendizaje Automático

Miguel Lentisco Ballesteros Antonio Coín Castro

Curso 2019-20

Índice

1. Introducción

En esta práctica perseguimos ajustar el mejor modelo lineal en dos conjuntos de datos dados, para resolver un problema de clasificación y otro de regresión. En ambos casos seguiremos una estrategia común que nos llevará finalmente a elegir un modelo y estimar su error.

1. Analizaremos la bases de datos y entenderemos el contexto del problema a resolver.
2. Preprocesaremos los datos de forma adecuada para trabajar con ellos.
3. Elegiremos una clase de hipótesis (lineal) para resolver el problema.
4. Fijaremos algunos modelos concretos y seleccionaremos el mejor según algún criterio.
5. Estimaremos el error del modelo.

Trabajamos en su mayoría con las funciones del paquete `scikit-learn`, apoyándonos cuando sea necesario en otras librerías como `numpy`, `matplotlib` ó `pandas`. El código de la práctica se ha estructurado en dos *scripts* de `Python` debidamente comentados:

- En `fit.py` se resuelve el problema de regresión.
- En `visualization.py` se recogen todas las funciones de visualización de gráficas, tanto comunes como propias de cada problema.

La ejecución de los dos programas principales está preparada para que se muestren solo algunas gráficas además del procedimiento de ajuste del modelo (aquellas que consumen menos tiempo). Este comportamiento puede cambiarse mediante el parámetro `show` de la función principal en cada caso, eliminando todas las gráficas (valor 0) o mostrándolas todas (valor 2). Además, en las operaciones de cálculo intensivo que lo permitan se utiliza el parámetro `n_jobs = -1` para paralelizar el flujo de trabajo en tantas hebras como se pueda. Por pantalla se muestra información sobre el tiempo de ejecución de los ajustes de los modelos y del programa completo.

Para que los experimentos sean reproducibles se fija una semilla aleatoria al incio del programa. Todos los resultados y gráficas que se muestran a continuación se han obtenido con el valor 2020 para la semilla, y pueden reproducirse si se ejecuta el programa tal y como se proporciona.

2. Base de datos y descripción del problema

Citar a (Fernandes, Vinagre, & Cortez, 2015) y poner link a la BBDD. Mencionar un resumen muy general de los diferentes atributos que se recogen (tipo real, entero, etc). Mostrar tabla-resumen que viene en el artículo.

2.1. Resumen estadístico de los datos (?)

Estudiar la distribución de clases si hacemos clasificación binaria (número exacto, porcentaje).

3. Selección de la clase de funciones

4. Conjuntos de entrenamiento, validación y *test*

5. Preprocesado de datos

5.1. Valores perdidos

5.2. Selección de características

5.3. Transformaciones polinómicas

5.4. Estandarización y umbral de varianza

5.5. Orden de las transformaciones

6. Métricas de error

7. Regularización

El uso de la regularización es esencial para limitar la complejidad del modelo y el *overfitting*, cosa que nos permitirá obtener una mejor capacidad de generalización. En principio se consideran dos posibilidades de regularización:

- **Regularización L2 (Ridge)**: se añade una penalización a la función de pérdida que es cuadrática en los pesos,
$$L_{reg}(w) = L(w) + \lambda ||w||_2^2.$$
- **Regularización L1 (Lasso)**: en este caso la penalización es lineal en los pesos, considerando la suma del valor absoluto de los mismos,
$$L_{reg}(w) = L(w) + \lambda \sum_i |w_i|.$$

En ambos casos el valor de $\lambda > 0$ es un hiperparámetro del modelo, que controla la intensidad de la regularización (a mayor valor, más pequeños serán los pesos). Encontrar un valor adecuado es una tarea complicada, pues si es demasiado pequeño seguiremos teniendo sobreajuste, pero si es demasiado grande podríamos caer en el fenómeno opuesto: tener *underfitting* porque el modelo sea poco flexible y no consiga ajustar bien los datos de entrenamiento.

8. Técnicas y selección de modelos

9. Análisis de resultados y estimación del error

10. Conclusiones y justificación

Anexo: Funcionamiento del código

Bibliografía

Fernandes, K., Vinagre, P., & Cortez, P. (2015). A proactive intelligent decision support system for predicting the popularity of online news. *Proceedings of the 17th portuguese conference on artificial intelligence*. Coimbra, Portugal.