Aprendizaje Automático

Práctica 2: Obtención y Validación de resultados

22 de octubre de 2018

La práctica desarrollada en esta unidad está dedicada a un tema transversal a todo el aprendizaje automático. Los métodos que vamos a ver aquí se pueden aplicar a cualquiera de los algoritmos que veremos a lo largo del curso y que nos van a ayudar a obtener mejores resultados e incluso validar la bondad de esos mismos.

Para la realización de la práctica podremos usar cualquiera de los dos algoritmos implementados en las prácticas anteriores con los datasets que deseemos, de forma que se puedan aplicar los métodos aquí descritos.

Aunque aquí vamos a ver los principios y a implementar algunos de ellos, también existen librerías en diferentes lenguajes de programación que las tienen realizadas y que pueden sernos útiles en un futuro.

1. Introducción

Cuando abordamos un problema (ideal) de aprendizaje automático deberíamos de tener 3 conjuntos de datos bien diferenciados:

- Entrenamiento (Training): se usa para extraer el modelo
- Prueba (Validation): Estima el error del modelo
- Generalización (Test): Estima la generalidad del modelo.

Aunque, generalmente, lo que se tiene es 1 único conjunto de datos del que queremos aprender.

Antes de comenzar a afrontar el problema del error, debemos de distinguir 2 tipos de errores:

- In-sample error: es la media de la predicción del error sobre el conjunto de entrenamiento
- Extra-sample error: es la media de predicción del error sobre ejemplos nuevos

Nunca se debe de usar el "In-sample" error para definir la bondad de los resultados, ya que podríamos incurrir en una reducción demasiado grande de este, perdiendo generalidad. Esto es lo que se conoce como "Overfitting" o "Sobrejuste" y lo podemos ver en la figura 1.

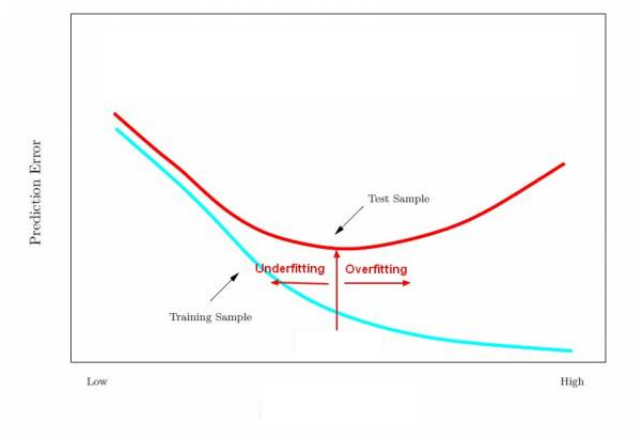

Figura 1

A partir de un número de ejemplos usados para aprender, el error sobre el training set (In-sample) sigue descendiendo, pero sin empbargo, el error frente a nuevas instancias comienza a subir. Ese sería el punto ideal donde el sistema debería "dejar de aprender".

Otra característica importante de la curva de aprendizaje es, como vemos en la figura 2, es que, salvo casos excepcionales, no llegamos a eliminar el error totalmente del sistema. En la figura tenemos representado "1-$\mathcal{E}$" (error relativo) y el límite superior de la curva está en 0.8 y no lo llega a alcanzar.

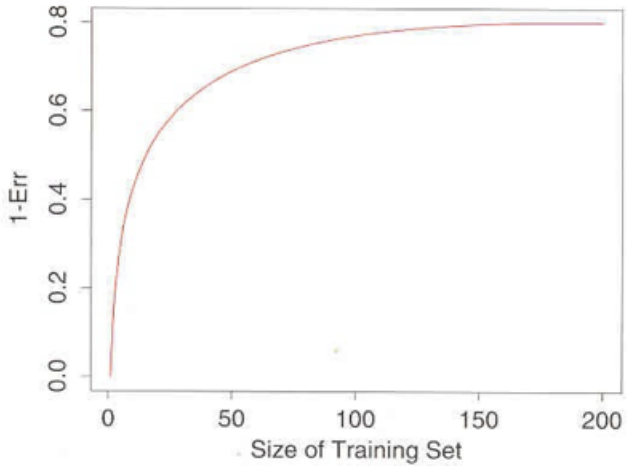

Figura 2

2. Métodos de generación y validación

Hay muchos métodos para la validación de resultados y, la mayoría, pertenecen al dominio de la Estadística, y que por tanto se salen del objetivo de esta práctica y este curso.

Nosotros vamos a ver algunos métodos interesantes y muy usados dentro del ámbito de la Inteligencia Artificial.

2.1. Jackknife o Leave-one-out

Es un método de validación muy sencillo y que consiste en entrenar con todos los ejemplos menos uno, y luego validar con ése. Este procedimiento se puede repetir para todos (n) los ejemplos del dataset, dejando uno fuera cada vez que se realiza. Los resultados de las n evaluaciones se agregan para obtener un mejor resultado y determinar la proporción de error. Este método es bueno cuando los datos son dispersos o valores muy extremos. La agregación de los resultados suele hacerse mediante la media de los resultados.

Al utilizarse la mayor cantidad de ejemplos posibles para el entrenamiento se incrementa la posibilidad de que el clasificador sea correcto ya que, en cierta medida, evalúa todas las posibilidades.

El procedimiento es determinístico ya que no se parten los datos al azar y no tiene sentido

repetir el procedimiento varias veces, ya que siempre se obtendrá el mismo resultado.

Como principal desventaja de este método es el gran coste computacional que conlleva y que lo hace poco efectivo en grandes volúmenes de datos.

2.2. Hold-out

Cuando existe una cantidad limitada de datos de entrenamiento se puede aplicar el método de retención (holdout) para estimar la proporción de error y obtener una solución agregada.

Este método reserva una cierta cantidad de datos al azar para prueba y utiliza el resto para el entrenamiento. En general, se reserva un tercio para prueba y se utilizan dos tercios como datos de entrenamiento.

Una manera de evitar la tendencia introducida por los datos retenidos, es repetir el proceso completo (entrenamiento y prueba) varias veces con distintas divisiones de los datos. Las proporciones de error obtenidas en las múltiples iteraciones se promedian para obtener una proporción de error general. Esta variante del método se conoce como retención repetida (repeated holdout).

2.3. Cross-Validation

Una de las técnicas más populares y con más éxito es la denominada "Cross-Validation".

La idea es dividir el dataset en k grupos (generalmente a partes iguales) y utilizar k-1 grupos como dataset de entrenamiento y el que queda como validación. Este procedimiento se repita k veces usando un conjunto para validación distinto cada vez, como podemos ver en la figura 3

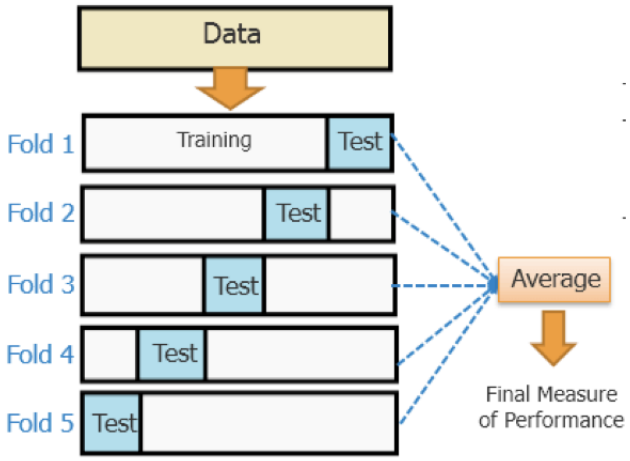

Figura 3

El resultado es la media de los resultados de cada vez.

2.4. Bootstrap

Este método está basado en el procedimiento estadístico de obtener muestras con sustitución. La idea del Bootstrap es tomar muestras del conjunto de datos con remplazo para formar un conjunto de entrenamiento.

Para ello, un conjunto de n instancias se muestrea n veces, con reemplazo, y se obtiene otro conjunto de datos de n instancias. Como algunas instancias del segundo conjunto estarán repetidas, deben existir algunas instancias del conjunto original que no fueron seleccionadas, esas serán las que utilizaremos para el conjunto de prueba.

La probabilidad de que una instancia particular sea elegida para el conjunto de entrenamiento es de $1/n$, y, por lo tanto, hay un $1 - 1/n$ de probabilidad de que no sea elegida. Si multiplicamos esto según las n oportunidades de ser elegida, obtenemos la siguiente probabilidad de que no sea escogida: un 63.2 % de entrenamiento. Esta es la razón por la cual este método se conoce como el 0.632 bootstrap.

Bootstrapping es muy útil cuando la muestra es muy pequeña para técnicas como crossvalidation o leave-one out, ya que grandes varianzas en conjuntos pequeños puede distorsionar.

2.5. Bagging

"Bagging" es el acrónimo de "Bootstrap aggregation" y es una técnica de agregación de Bootstrap.

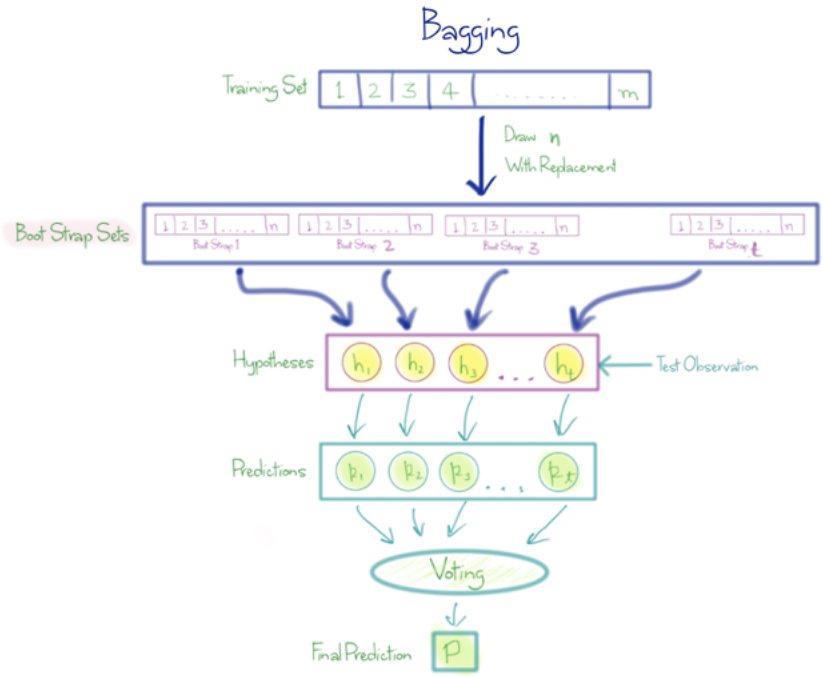

Figura 4

Como se ve en la figura 4, de un dataset de m instancias, construye n instancias de Bootstrap, obteniendo n hipótesis y n errores de predicción.

El resultado de la clasificación es una media ponderada de las predicciones mediante el error obtenido.

3. Errores en clasificación

La mayoría de los métodos anteriores han sido pensados para el problema de regresión, aunque son fácilmente extensibles a la clasificación.

Una de las herramientas más comunes en los problemas de clasificación es la *matriz de confusión*.

3.1. Matriz de confusión

Para hacernos una idea clara de la matriz vamos a restringirnos inicialmente al caso binario: los ejemplos pertenecen o no a una clase.

Realizamos la clasificación de un conjunto de datos y le pasamos el conjunto de test, representando el número de aciertos y fallos como indica la siguiente matriz:

		Clase predicha	
		Si	No
Clase real	Si	N. Verdaderos Positivo	N. Falsos Negativo
	No	N. Falsos Positivo	N. Verdaderos negativo

En verde, el número de ejemplos que pertenecían a la clase y fueron detectados como tales (Verdaderos positivos), o que no pertenecían y el clasificador ha dicho que no (Verdaderos negativos). En rojo los ejemplos erróneamente clasificados, es decir, los que eran de la clase y ha predicho que no y viceversa.

Por último, se puede extender la matriz a un conjunto de multiclasificación, haciendo una matriz de $n \times n$ (n clases), donde cada celda es el número de ejemplos de una clase que han sido clasificados de alguna forma. Los aciertos son la diagonal principal y los errores el resto. Ver figura 5

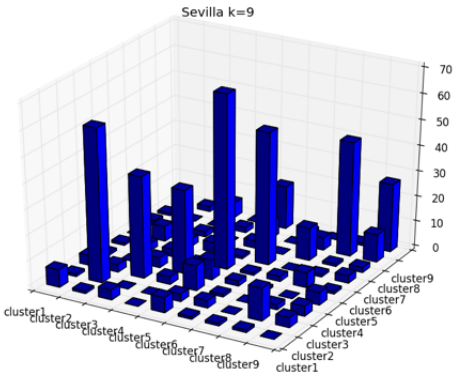

Figura 5

4. Ejercicio

Para realizar estos ejercicios utilizar los mismos datos de la práctica 1.

1. Implementar el método de Bagging donde se pueda elegir el dataset y el número de Bootstrap como parámetros.
2. Para un conjunto de test, calcular la matriz de confusión del mismo.

5. Bibliografía

- Yu, Chong Ho (2003): Resampling methods: concepts, applications and justification. Practical Assessment, Research & Evaluation, 8(19).
- `http://www.uvm.edu/~dhowell/StatPages/Resampling/Resampling.html`
- Weiss & Kulikowski, 1991, Chapter 2