

Cuestionario 3 de Aprendizaje Automático



UNIVERSIDAD DE GRANADA

Antonio Manuel Fresneda Rodríguez

antoniomfr@correo.ugr.es

Jueves, 31 de mayo de 2018

Ejercicio 1

Tanto “bagging” como validación-cruzada cuando se aplican sobre una muestra de datos nos permiten dar una estimación del error de un modelo ajustado a partir de dicha muestra. Enuncie las diferencias y semejanzas entre ambas técnicas. Diga cual de ellas considera que nos proporcionará una mejor estimación del error en cada caso concreto y por qué.

Semejanzas

- Ambas dividen el conjunto de entrenamiento (Bagging muestrea el conjunto de entrenamiento y CV coge un porcentaje para verificar con ese como de bueno es el modelo).

Diferencias

- Validación cruzada proporciona una estimación del error fuera de la muestra, mientras que bagging muestrea el conjunto de entrenamiento en k subconjuntos, entrena el mismo modelo con los k subconjuntos y finalmente realiza un promedio del resultado de los k modelos.

Si lo que queremos es una estimación del error de la muestra lo más seguro para usar es validación cruzada, ya que con

Ejercicio 2

Considere que dispone de un conjunto de datos linealmente separable. Recuerde que una vez establecido un orden sobre los datos, el algoritmo perceptron encuentra un hiperplano separador iterando sobre los datos y adaptando los pesos de acuerdo al algoritmo.

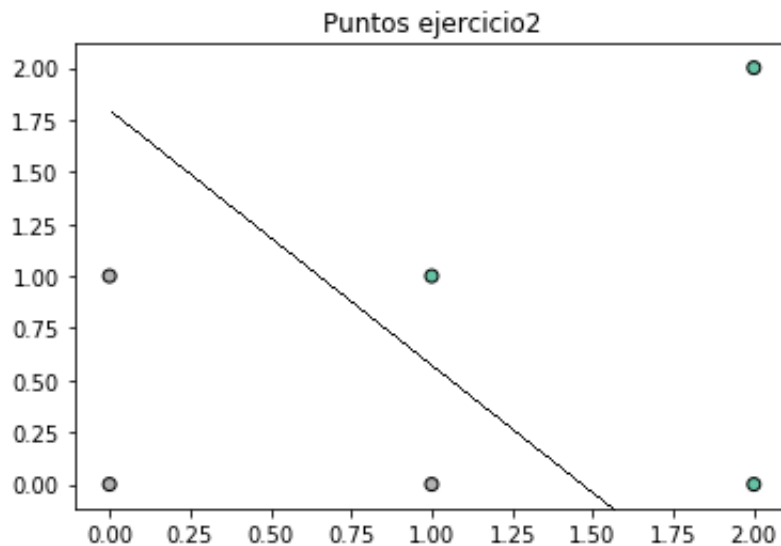
Modificar este pseudo-código para adaptarlo a un algoritmo simple de SVM, considerando que en cada iteración adaptamos los pesos de acuerdo al caso peor clasificado de toda la muestra. Justificar adecuadamente/matematicamente el resultado, mostrando que al final del entrenamiento solo estaremos adaptando los vectores soporte.

La diferencia entre el Perceptron y SVM es que perceptron termina cuando encuentra un hiperplano separador y SVM no termina hasta que encuentre el hiperplano que tenga una distancia a los vectores soporte que supere un umbral establecido.

Ejercicio 3

Considerar un modelo SVM y los siguientes datos de entrenamiento: Clase-1: (1,1), (2,2), (2,0), Clase-2: (0,0), (1,0), (0,1)

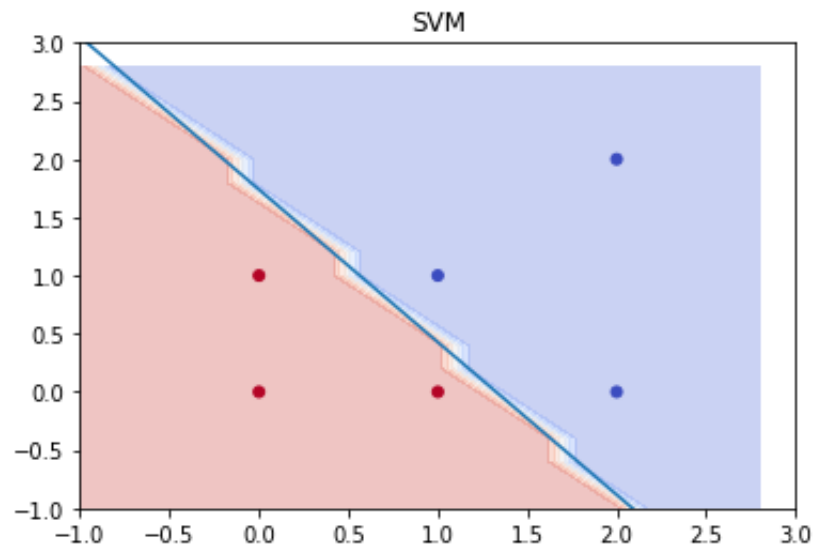
- a) Dibujar los puntos y construir por inspección el vector de pesos para el hiperplano óptimo y el margen óptimo.



- b) ¿Cuáles son los vectores soporte?

Los vectores soporte son: (1,0), (0,1) y (1,1)

- c) Construir la solución en el espacio dual. Comparar la solución con la del apartado (a)



Mas o menos podemos ver que el hiperplano obtenido es más o menos el que estimábamos. El gráfico se ha obtenido con el siguiente script de Python:

```

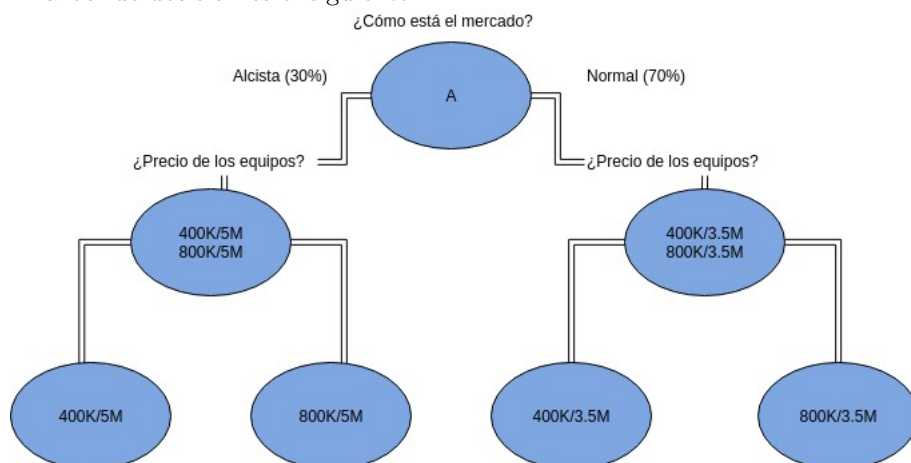
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from sklearn import svm
4
5
6 X=np.asarray([[1,1],[2,2],[2,0],[0,0],[1,0],[0,1]])
7 Y=np.asarray([0,0,0,1,1,1])
8
9 fig, ax = plt.subplots()
10 clf2 = svm.LinearSVC(C=3).fit(X, Y)
11
12 # get the separating hyperplane
13 w = clf2.coef_[0]
14 a = -w[0] / w[1]
15 xx = np.linspace(-5, 5)
16 yy = a * xx - (clf2.intercept_[0]) / w[1]
17
18 # create a mesh to plot in
19 x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
20 y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
21 xx2, yy2 = np.meshgrid(np.arange(x_min, x_max, .2),
22                         np.arange(y_min, y_max, .2))
23 Z = clf2.predict(np.c_[xx2.ravel(), yy2.ravel()])
24
25 Z = Z.reshape(xx2.shape)
26 ax.contourf(xx2, yy2, Z, cmap=plt.cm.coolwarm, alpha=0.3)
27 ax.scatter(X[:, 0], X[:, 1], c=Y, cmap=plt.cm.coolwarm, s=25)
28 ax.plot(xx,yy)
29
30 ax.axis([x_min, x_max,y_min, y_max])
31 plt.title("SVM")
32 plt.show()

```

Ejercicio 4

Una empresa está valorando cambiar su sistema de proceso de datos, para ello dispone de dos opciones, la primera es adquirir un nuevo sistema compuesto por dos sistemas idénticos al actual a 200.000 euros cada uno, y la segunda consiste en adquirir un nuevo sistema mucho mayor por 800.000 euros. Las ventas que la empresa estima que tendrá a lo largo de la vida útil de cualquiera de sus nuevos equipos es de 5.000.000 de euros en el caso de un mercado alcista, a lo que la empresa le asigna una probabilidad de que suceda del 30%, en caso contrario, las ventas esperadas son de 3.500.000 euros. Construir el árbol de decisiones y decir que opción es la más ventajosa para la empresa

El árbol de decisión es el siguiente:



No tenemos información suficiente del desempeño de las dos opciones por lo que he decidido escoger la que mayor precio/beneficio según el estado del mercado. Si el mercado es alcista escogeríamos la opción de los dos equipos informáticos y si el mercado está normal también escogeríamos la opción de los dos equipos (ya que es más barata esta opción).

Ejercicio 5

¿Que algoritmos de aprendizaje no se afectan por la dimensionalidad del vector de características?. Diga cuáles y por qué.

Los árboles de decisión. Estos no se ven afectados por la dimensionalidad del vector puesto que tienen una dimensión d_{vc} infinita, siempre podemos dividir todos los datos que tengamos creando tantas ramas como datos tenemos.

Ejercicio 6

Considere la siguiente aproximación al aprendizaje. Mirando los datos, parece que los datos son linealmente separables, por tanto decidimos usar un simple perceptron y obtenemos un error de entrenamiento cero con los pesos óptimos encontrados. Ahora deseamos obtener algunas conclusiones sobre generalización, por tanto miramos el valor d_{vc} de nuestro modelo y vemos que es $d + 1$. Usamos dicho valor de d_{vc} para obtener una cota del error de test. Argumente a favor o en contra de esta forma de proceder identificando los posibles fallos si los hubiera y en su caso cual hubiera sido la forma correcta de actuación

Yo estoy en contra. El primer error grave que salta a la vista es que estamos mirando los datos **antes** de escoger el modelo con el que vamos a aprender. No podemos ver los datos ya que nadie nos asegura que la muestra sea representativa.

Otro error es que si usamos la cota calculada a partir de la d_{vc} esta cota va a ser muy laxa y vamos a tener una aproximación pobre del error fuera de la muestra. Yo lo que haría sería realizar una validación cruzada que siempre nos va a dar una cota más ajustada.

Ejercicio 7

Discuta pros y contras de los clasificadores SVM y Random Forest (RF). Considere que SVM por su construcción a través de un problema de optimización debería ser un mejor clasificador que RF. Justificar las respuestas.

- **SVM**

Pros:

- Encuentra el hiperplano que proporciona una clasificación óptima.
- Tiene pocos parámetros.
- El concepto de kernel.
- Funciona bien con muchos datos (solo se tienen en cuenta los vectores soporte.)

Contras:

- No funcionan bien con variables categóricas.

- **RF**

Pros:

- Tienen una varianza baja.
- Árboles que no están correlados.

Contras:

- Pierden expresividad (respecto a árboles).

No tiene por que ser mejor, sabemos que los SVM encuentran el hiperplano que da una clasificación óptima, pero RF puede comportarse también muy bien puesto que son capaces de entrenar varios árboles no correlados y conseguir reducir la varianza.

Ejercicio 8

¿Cuál es a su criterio lo que permite a clasificadores como Random Forest basados en un conjunto de clasificadores simples aprender de forma más eficiente? ¿Cuales son las mejoras que introduce frente a los clasificadores simples? ¿Es Random Forest óptimo en algún sentido? Justifique con precisión las contestaciones.

La forma en la que Random Forest aprende de forma más óptima es mediante bagging y construyendo árboles no correlados. Esta técnica consiste en realizar un muestreo al conjunto de entrenamiento y entrenar varios árboles con estos subconjuntos para luego hacer un promedio de estos predictores escogidos de forma aleatoria. Puesto que los árboles no están correlados, esto provoca que la varianza que tengamos disminuya.

La mejora que introduce RF es que como tenemos menor varianza y como el sesgo podemos reducirlo (creando más ramas), podríamos llegar a tener un predictor con bajo sesgo y baja varianza. Por esto podríamos decir que RF podría en algún caso ser un predictor óptimo.

Ejercicio 9

En un experimento para determinar la distribución del tamaño de los peces en un lago, se decide echar una red para capturar una muestra representativa. Así se hace y se obtiene una muestra suficientemente grande de la que se pueden obtener conclusiones estadísticas sobre los peces del lago. Se obtiene la distribución de peces por tamaño y se entregan las conclusiones. Discuta si las conclusiones obtenidas servirán para el objetivo que se persigue e identifique si hay algo que lo impida.

Puede ocurrir que haya peces que no los pueda atrapar la red y estos no van a estar en nuestra muestra pero sí en la población. Esto va a provocar que discriminemos estos peces en nuestro intento de determinar el tamaño de los peces.

También puede ocurrir que los peces vayan en un banco y hayamos atrapado solos a los de ese banco y tengamos algunos que no hayamos capturado.

Resumiendo, no podemos asegurar que una muestra sea representativa de una población, aunque esto ocurre con una probabilidad baja.

Ejercicio 10

Identifique dos razones de peso por las que el ajuste de un modelo de red neuronal a un conjunto de datos puede fallar o equivalentemente obtener resultados muy pobres. Justifique la importancia de las razones expuestas.

Las dos razones por las que una red neuronal puede no aprender son las siguientes:

- **Función lineal:** Si tenemos una función no lineal como la sigmoide o la tangente hiperbólica vamos a correr el riesgo de que en alguna capa tengamos algún peso con un valor alto. Esto va a provocar que cuando hagamos backpropagation, la derivada en ese punto sea 0 y arrastremos ese cero hasta el final, lo que provocaría que la red no aprendiese nada.
- **Pesos:** Si iniciamos los pesos a cero nos puede ocurrir algo parecido a lo anterior, vamos a estar propagando pesos nulos hacia adelante y la red no va a aprender nada. Pero si estos valores son cercanos a 1 también es un problema puesto que estaríamos saturando la función y al hacer backpropagation estaríamos en el mismo caso que el anterior.