Aprendizaje Automático (2017-2018)
Doble Grado en Ingeniería Informática y Matemáticas
Universidad de Granada

Cuestionario 3

Simón López Vico

31 de mayo de 2018

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.

   Solución:

   En ambos casos tomamos muestras del conjunto de entrenamiento, entrenamos nuestro clasificador con ellos y calculamos el error *Out-of-Bag* (en caso de "bagging") o el error del validación (en caso de validación cruzada) con los datos que no participan en la muestra de entrenamiento.

   La principal diferencia entre los dos métodos reside en la manera de obtener las muestras para el conjunto de entrenamiento. Con "bagging", se generan B conjuntos de entrenamiento mediante bootstrapping, remuestreando de forma aleatoria y con reemplazo la muestra original; algunos datos en el conjunto pueden repetirse. Por otra parte, en validación cruzada se separa la muestra de entrenamiento dada en K pliegues, eligiendo cada vez 1 pliegue para validación y $K-1$ pliegues como conjunto de entrenamiento, a través de los cuales obtendremos K modelos y K estimaciones del error de fuera de la muestra.

   Otra diferencia es como tratan los resultados: mientras que la validación cruzada se quedará con el modelo que mejores resultados dé (es decir, menor E_{out}), "bagging" generará un modelo obtenido mediante el promedio de todas las predicciones de los B-modelos (para regresión) o escogerá el voto mayoritario de todos los B-modelos (para clasificación).

   El "bagging" será útil cuando tengamos un conjunto de datos con una varianza muy elevada (ya que ésta técnica la disminuye) mientras que la validación cruzada se usará típicamente como una manera de elegir empíricamente buenos hiperparámetros sin contaminar el conjunto de prueba.

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.

   Algorithm 1 Perceptron
   1: **Entradas**: (x_i, y_i), $i = 1, ..., n, w = 0, k = 0$
   2: **repeat**
   3: $\quad k \leftarrow (k+1) \bmod n$
   4: $\quad$ **if** $\text{sign}(y_i) \neq \text{sign}(w^T x_i)$ **then**
   5: $\quad\quad w \leftarrow w + y_i x_i$
   6: $\quad$ **end if**
   7: **until** todos los puntos bien clasificados

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.

Solución:

3. Considerar un modelo SVM y los siguientes datos de entrenamiento: Clase-1: $\{(1,1),(2,2),(2,0)\}$, Clase-2: $\{(0,0),(1,0),(0,1)\}$:

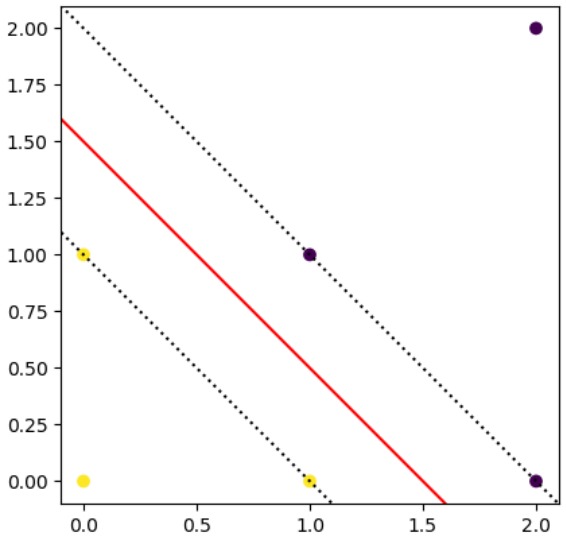
   a) Dibujar los puntos y construir por inspección el vector de pesos para el hiperplano óptimo y el margen óptimo.
   Solución: Dibujaremos en una gráfica dichos puntos con el hiperplano óptimo, usando el siguiente código:

```
import matplotlib.pyplot as plt
import numpy as np

X=np.array([[0,0],[0,1],[1,0],[1,1],[2,2],[2,0]])
y=np.array([1,1,1,-1,-1,-1])

plt.scatter(X[:, 0], X[:, 1], c=y)
t = np.arange(-5, 5, 0.1)
plt.plot(t, -t+1.5, 'r-')
plt.plot(t, -t+2, 'k:')
plt.plot(t, -t+1,'k:')
plt.axis([-0.1, 2.1, -0.1, 2.1])
plt.gca().set_aspect('equal', adjustable='box')
plt.show()
```

   El resultado obtenido será el siguiente:



   El vector de pesos para el hiperplano será el vector ortogonal al hiperplano representado mediante la recta roja.

El margen óptimo será la distancia entre las dos líneas de puntos que hay en la gráfica, las cuales representarán las rectas $r = \{x \in \mathbb{R}/x = -t+1, t \in \mathbb{R}\}$ y $s = \{x \in \mathbb{R}/x = -t+2, t \in \mathbb{R}\}$. Por tanto, el margen óptimo valdrá $d(r,s) = \frac{1}{\sqrt{2}}$.

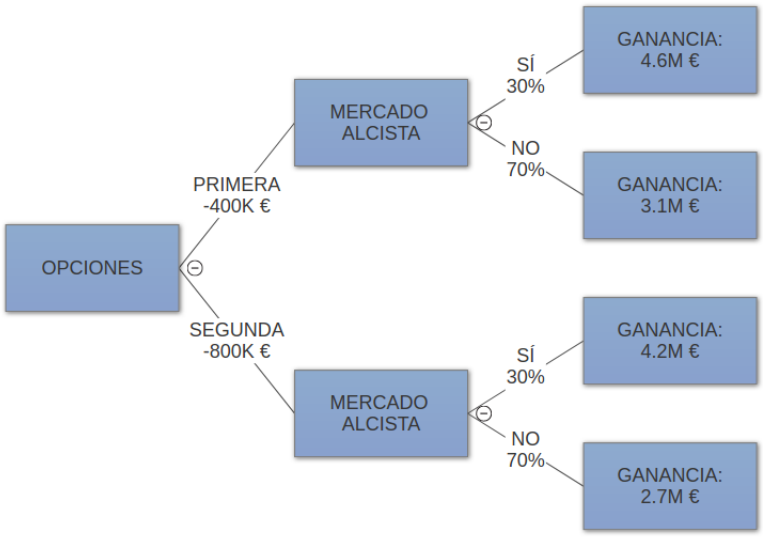
b) ¿Cuáles son los vectores soporte?
Solución: Los vectores de soporte serán los formados por los puntos más cercanos al hiperplano y que determinan el margen óptimo; por tanto, $VS_{Clase\text{-}1} = \{(1,1),(2,0)\}$ y $VS_{Clase\text{-}2} = \{(0,1),(1,0)\}$.

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

4. Una empresas esta 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.

   Solución:

   El árbol de decisión construido será el siguiente:



   Con dicho árbol, la opción más ventajosa para la empresa sería comprar los dos sistemas idénticos de 200.000 euros, pues en el peor de los casos obtendría unas ganancias de 3.100.000 euros. Además, si la empresa dispone de dos sistemas idénticos y uno de ellos comienza a fallar o se estropea, siempre puede seguir usando el otro mientras solucionan los problemas de este, cosa que no puede hacer si compra el sistema único de 800.000 euros.

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

   Solución:

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 posible fallos si los hubiera y en su caso cual hubiera sido la forma correcta de actuación.

   Solución:

   Para empezar, el hecho de mirar los datos es un fallo en la manera de proceder; al haber visto la muestra, hemos obtenido información del problema a resolver, sesgando así el conocimiento que podemos obtener de ellos. Por tanto, perdemos capacidad de generalización del modelo y todos las conclusiones obtenidas a continuación pierden su validez; hemos contaminado el aprendizaje.

   La forma correcta de actuación hubiera sido tomar distintos modelos e ir analizando los errores de los conjuntos train y test. Después, con d_{vc} ya estimada, calculamos una cota de error de generalización. Si aplicamos este procedimiento sobre distintos modelos podremos obtener una estimación más fiable del modelo óptimo para resolver este problema.

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

   Solución:

   - Support Vector Machine:
     - Pros:
       - Se comporta de forma similar a la regresión logística cuando hay separación lineal.
       - Funciona bien con límites no lineales, dependiendo del kernel utilizado.
       - Maneja bien los datos de alta dimensión.
       - Es útil tanto para datos separables linealmente (*hard margin*) como para datos no separables linealmente (*soft margin*).
     - Contras:
       - Sensible a problemas de entrenamiento/overfitting dependiendo del kernel.
       - Menos eficaz en conjuntos de datos más ruidosos con clases superpuestas.
   - Random Forest:
     - Pros:
       - Es un predictor muy eficiente.
       - Construye árboles no correlados, obteniendo así una mayor reducción en la varianza.
       - Da estimados sobre qué variables son importantes en la clasificación.
     - Contras:
       - No es tan fácil de interpretar visualmente.
       - La calidad de la solución en cada "bolsa" debe ser relativamente alta para que "bagging" tenga un rendimiento prometedor.

   Respecto a si SVM es un mejor clasificador que RF, podemos aplicar el teorema de no free lunch, el cual dice que cuando se comparan los algoritmos de aprendizaje máquina a través de infinitamente muchos conjuntos de datos no habrá ninguno mejor que otro. Por tanto, el hecho de que SVM provenga de un problema de optimización no tiene por qué determinar que sea mejor clasificador que RF, ya que, además, RF es un predictor muy eficiente.

   En términos prácticos, ambos tienen sus ventajas y desventajas. Ninguno de los dos métodos es mejor que el otro, pues cada uno tiene su finalidad y manera de actuar.

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.

   Solución:

   Random Forest está construido sobre la idea de "bagging", un clasificador simple. Una de las mejoras que introduce frente a este clasificador es que descorrela los árboles de bagging para obtener una mayor reducción en varianza.

   Usando Random Forest en un problema de clasificación, la dimensionalidad subespacial es $\sqrt{p}$ por defecto (bastante pequeña; p es el número total de predictores), pero los árboles contendrán muchos nodos. Por otra parte, en un problema de regresión, la dimensionalidad subespacial es $p/3$ por defecto (lo suficientemente grande), aunque los árboles contendrán menos nodos. Por lo tanto, el número óptimo de árboles en Random Forest dependerá del número de predictores sólo en casos extremos.

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.

   Solución:

   Nos encontramos ante un caso de sesgo muestral, pues el tamaño de la malla de red será importante. Si usamos una red con tamaño de malla de 1 cm, entonces los peces que sean más estrechos de 1 cm no se encontrarán en las muestra obtenida. Los datos estadísticos obtenidos no cumplirán (totalmente) el objetivo que se persigue, pues no hay manera de saber si hay peces de tamaño inferior a 1 cm tras realizar el experimento.

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.

    Solución: