

Aprendizaje Automático (2018-2019)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Cuestionario 3

Montserrat Rodríguez Zamorano

25 de mayo de 2019

Índice

1. Preguntas obligatorias	1
1.1. Ejercicio 1	1
1.2. Ejercicio 2	1
1.3. Ejercicio 3	1
1.4. Ejercicio 4	2
1.5. Ejercicio 5	3
1.6. Ejercicio 6	3
1.7. Ejercicio 7	4
1.8. Ejercicio 8	4
1.9. Ejercicio 9	4
1.10. Ejercicio 10	5

1. Preguntas obligatorias

1.1. Ejercicio 1

¿Podría considerarse Bagging como una técnica para estimar el error de predicción de un modelo de aprendizaje? Diga sí o no con argumentos. En caso afirmativo compárela con validación cruzada.

Si, *bagging* nos permite estimar el error de predicción de un modelo ajustado a partir de dicha muestra ya que se basa en *bootstrap*, que es una técnica que puede usarse para cuantificar la incertidumbre asociada con un determinado método de aprendizaje.

Bagging está basada en dos actuaciones: *bootstrapping* y promediar. Con *bootstrapping* se generan B conjuntos de entrenamiento de igual tamaño al conjunto de datos de forma aleatoria y con reemplazamiento. En promedio, cada árbol de *bootstrap* hace uso de $2/3$ de la muestra, por lo que nos queda un tercio para *test*.

Esto no ocurre en validación cruzada, donde podemos elegir los tamaños para *train* y *test*. Además, los conjuntos se generan de forma distinta: se divide la muestra en N subconjuntos, y se utilizan $N - 1$ para *train* y el subconjunto restante para *test*.

Por tanto, ambas técnicas dividen la muestra en grupos de entrenamiento y validación, lo que varía es la forma de hacer estos grupos. Una ventaja que presenta *bagging* frente a validación cruzada es que reduce la varianza y el sesgo.

Como en casi todos los casos, el uso de una u otra depende del problema. Validación cruzada puede ser buena preferible para escoger hiperparámetros, por ejemplo. Sin embargo, para un modelo inestable, será mejor usar *bagging* ya que reduce la varianza.

1.2. Ejercicio 2

Considere que dispone de un conjunto de datos linealmente separable. Recuerde que una vez establecido un orden sobre los datos, el algoritmo perceptrón encuentra un hiperplano separador iterando sobre los datos y adaptando los pesos de acuerdo al algoritmo. Modificar este pseudocódigo 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/matemáticamente el resultado, mostrando que al final del entrenamiento sólo estaremos adaptando los vectores soporte.

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

a) En la figura (1.2) pueden verse los puntos de la clase 1 en verde, los puntos de la clase 2 en azul, los márgenes óptimos en amarillo y el hiperplano óptimo en rojo.

Algorithm 1 Perceptron

```
1: Entradas:  $(\mathbf{x}_i, y_i), i = 1, \dots, n$ ,  $\mathbf{w} = 0$ ,  $k = 0$   
2: repeat  
3:    $k \leftarrow (k + 1) \bmod n$   
4:   if  $\text{sign}(y_i) \neq \text{sign}(\mathbf{w}^T \mathbf{x}_i)$  then  
5:      $\mathbf{w} \leftarrow \mathbf{w} + y_i \mathbf{x}_i$   
6:   end if  
7: until todos los puntos bien clasificados
```

Figura 1.1: Pseudocódigo algoritmo perceptrón

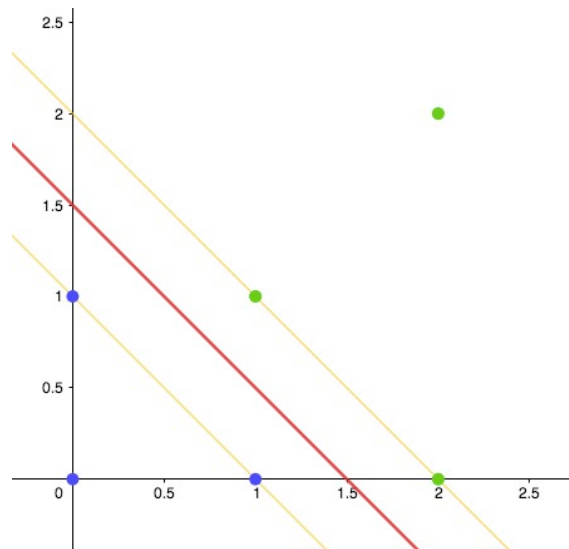


Figura 1.2: Puntos, hiperplano óptimo y el margen óptimo

- b) Los vectores soporte son $(1, 0)$ y $(0, 1)$ de la clase 1 y $(1, 1)$ y $(2, 0)$ de la clase 2.
- c)

1.4. Ejercicio 4

¿Cuál es el criterio de optimalidad en la construcción de un árbol? Analice un clasificador en árbol en términos de sesgo y varianza. ¿Qué estrategia de mejora propondría?

El problema de la optimalidad global en la construcción de un árbol de decisión es un problema NP-completo, por tanto no se puede garantizar una solución óptima para un problema determinado.

Un clasificador en árbol no está podado, por lo que presenta un bajo sesgo, pero una varianza alta. Como estrategia de mejora propondría el uso de *bagging*, ya que resuelve el problema de la alta varianza, generando B conjuntos de entrenamiento diferentes y

promediando las predicciones de los B modelos. Es el uso del promedio el que nos da esta baja varianza.

1.5. Ejercicio 5

¿Cómo influye la dimensión del vector de entrada en los modelos: SVM, RF, Boosting y NN?

Afectaría en el caso de que el modelo no manejara bien las dimensionalidades altas.

En el caso de Boosting, las dimensionalidades altas pueden provocar sobreajuste en los modelos simples que forman la combinación, por lo que un vector de entrada con una alta dimensión podría afectar al buen funcionamiento del modelo.

Por otro lado, NN, SVM y RF son modelos más estables y resistentes al *overfitting*, por lo que la dimensionalidad del vector de entrada no les afectará tanto en este caso.

1.6. Ejercicio 6

El método de Boosting representa una forma alternativa en la búsqueda del mejor clasificador respecto del enfoque tradicional implementado por los algoritmos PLA, SVM, NN, etc. a) Identifique de forma clara y concisa las novedades del enfoque; b) Diga las razones profundas por las que la técnica funciona produciendo buenos ajustes (no ponga el algoritmo); c) Identifique sus principales debilidades; d) ¿Cuál es su capacidad de generalización comparado con SVM?

a) Boosting se basa en la idea de usar diferentes clasificadores simples que actúen en diferentes partes del espacio. Así, estos clasificadores combinados nos darán un clasificador que nos de mejores predicciones.

b)

- Rapidez.
- Fácil y simple de programar.
- Sólo hace falta estimar el parámetro T , esto es, el número de subespacios en los que dividimos el espacio de entrada.
- Flexible: puede combinar cualquier algoritmo de aprendizaje.
- No se necesita conocimiento previo de los clasificadores simples.
- Versátil: se puede usar con distintos tipos de datos (numéricos, discretos,...)

c) Sus debilidades son la elección de los modelos simples y su combinación, ya que puede llevar a *overfitting* o *underfitting*. Otra debilidad es la sensibilidad al ruido que ha demostrado empíricamente.

d) Depende del problema: si se trata de un problema linealmente separable, SVM tendrá una mejor actuación que Boosting con subespacios y modelos simples inadecuados.

1.7. Ejercicio 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.

- SVM: Su principal desventaja es su ineficiencia durante el entrenamiento. Por otra parte, siempre encuentra el hiperplano óptimo, es poco sensible al ruido y puede manejar espacios de alta dimensionalidad.
- Random Forest: Algunas ventajas es que es un predictor muy eficiente, obtiene una mayor reducción de la varianza, nos da la importancia de las variables y puede manejar grandes cantidades de datos. Como desventajas tenemos la dificultad de interpretación y la posibilidad de sobreajuste en grupos de datos ruidosos.

Ambos modelos resisten bien el sobreajuste. SVM no tiene por qué ser mejor clasificador que RF, depende del problema. Por ejemplo, si el conjunto de datos es grande el elevado tiempo de entrenamiento hará que RF sea una mejor opción.

1.8. 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? ¿Cuáles son las mejoras que introduce frente a los clasificadores simples? ¿Es Random Forest óptimo en algún sentido? Justifique con precisión las contestaciones.

Al estar construido sobre la idea de *bagging*, los conjuntos de entrenamiento presentarán datos repetidos, por lo que el árbol de decisión es más simple y el aprendizaje será más eficiente.

La mejora que introduce frente a clasificadores simples es que construye árboles no correlados, lo que se traduce en una menor varianza y estabilidad para el modelo.

1.9. 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 lado. 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.

Estas conclusiones no servirán para el objetivo que se persigue. En primer lugar, se toma la muestra en una sola vez. Esto quiere decir que ha sido en momento determinado del día, en una época determinada del año, en una zona determinada del lago, con una red determinada. Puede ser que en invierno haya peces de diferente tamaño más grande que en verano, o que haya peces que sean tan pequeños que no queden atrapados en las redes.

Por tanto, no es una buena forma de obtener una muestra representativa.

1.10. Ejercicio 10

Identifique qué pasos daría y en qué orden para conseguir con el menor esfuerzo posible un buen modelo de red neuronal a partir de una muestra de datos. Justifique los pasos propuestos, el orden de los mismos y argumente que son adecuados para conseguir un buen óptimo. Considere que tiene suficientes datos tanto para el ajuste como para el test.