Aprendizaje Automático: Cuestionario 2

David Cabezas Berrido

20 de mayo de 2020

1. 1

No podemos calcular una cota del error de generalización, ya que hemos basado la elección del modelo en información referente a los datos y la clase de funciones que podríamos haber considerado si no nos hubiésemos basado en información de los datos tiene dimensión VC potencialmente infinita.
Para mi razonamiento, me he guiado del Ejercicio 5.4 de Learning From Data. Es prácticamente el mismo problema pero dentro de contexto.
Si asumiesemos que la información que nos proporcionó la otra persona sobre la separabilidad de los datos es debida a la naturaleza del problema y no a la muestra en sí, sí que podríamos calcular una cota usando $d_{VC} = 3 + 1 = 4$ (perceptrón 3D). Obteniendo
$E_{out}(h) \leq \sqrt{\frac{31{,}79 - \ln \delta}{125}}$ con probabilidad $1 - \delta$

2. 2

$E_{out}(h) \leq E_{in}(h) + 0{,}38875375$ con probabilidad 0,95.
Utilizamos la desigualdad (denotando h a la hipótesis obtenida al entrenar el modelo con la muestra):
$E_{out}(h) \leq E_{in}(h) + \sqrt{\frac{8}{N} \ln \frac{4m_{\mathcal{H}}(2N)}{\delta}}$ con probabilidad $1 - \delta$
Queremos que la desigualdad se cumpla con (al menos) probabilidad 0,95 luego $\delta = 0{,}05$. También sustituimos $N = 1000$ obteniendo
$E_{out}(h) \leq E_{in}(h) + \sqrt{\frac{8}{1000} \ln \frac{4m_{\mathcal{H}}(2000)}{0{,}05}} = E_{in}(h) + \sqrt{\frac{1}{125} \ln \frac{4(1 + 2000 + \binom{2000}{2})}{0{,}05}} = E_{in}(h) + 0{,}38875375$ con probabilidad 0,95.

3. 3

$d_{VC}(\mathcal{H}) \leq K(d + 2) - 1$
Sabemos que $m_{\mathcal{H}_1 \cup \mathcal{H}_2}(N) < 2^N$ para todo N tal que $d_{VC}(\mathcal{H}_1) + d_{VC}(\mathcal{H}_2) + 2 \leq N$. Usando la definición de d_{VC} concluimos que $d_{VC}(\mathcal{H}_1 \cup \mathcal{H}_2) \leq d_{VC}(\mathcal{H}_1) + d_{VC}(\mathcal{H}_2) + 2 - 1$
Hagamos inducción con esta desigualdad para generalizarla:
Si $d_{VC}\left(\bigcup_{i=1}^{Q-1} \mathcal{H}_i\right) \leq \sum_{i=1}^{Q-1} d_{VC}(\mathcal{H}_i) + (Q - 1) - 1$, tenemos por el caso $Q = 2$:
$d_{VC}\left(\bigcup_{i=1}^{Q} \mathcal{H}_i\right) = d_{VC}\left(\left(\bigcup_{i=1}^{Q-1} \mathcal{H}_i\right) \cup \mathcal{H}_Q\right) \leq d_{VC}\left(\bigcup_{i=1}^{Q-1} \mathcal{H}_i\right) + d_{VC}(\mathcal{H}_Q) + 1$
ahora aplicando la hipótesis de inducción para $Q - 1$, concluimos para Q:
$d_{VC}\left(\bigcup_{i=1}^{Q} \mathcal{H}_i\right) \leq \sum_{i=1}^{Q-1} d_{VC}(\mathcal{H}_i) + (Q - 1) - 1 + d_{VC}(\mathcal{H}_Q) + 1 = \sum_{i=1}^{Q} d_{VC}(\mathcal{H}_i) + Q - 1$
Si tenemos las variables $x_1, \ldots, x_d$ en las muestras de $\mathcal{X}$, cada $\mathcal{H}_k$ es una clase de funciones lineales que trabaja con las variables $x_1^k, \ldots, x_d^k$, que son d variables independientemente del valor de k. Por tanto $d_{VC}(\mathcal{H}_k) = d + 1 \quad \forall k = 1, \ldots, K$.
Usando la cota que hemos probado concluimos
$d_{VC}(\mathcal{H}) = d_{VC}\left(\bigcup_{k=1}^{K} \mathcal{H}_k\right) \leq \sum_{k=1}^{K} d_{VC}(\mathcal{H}_k) + K - 1 = K(d + 1) + K - 1 = K(d + 2) - 1$

4. 4

Fijado $N \in \mathbb{N}$, consideramos una muestra cualquiera de $2N$ elementos. Dividimos dicha muestra en dos submuestras de tamaño N, la clase de funciones $\mathcal{H}$ será capaz de generar como mucho $m_{\mathcal{H}}(N)$ dicotomías sobre cada una de estas submuestras. Por tanto, sobre la muestra completa $\mathcal{H}$ no podrá generar más de $m_{\mathcal{H}}(N) \cdot m_{\mathcal{H}}(N)$ dicotomías diferentes, luego concluimos $m_{\mathcal{H}}(2N) \leq m_{\mathcal{H}}(N)^2$.

5. 5

Aparte.

6. 6

ID3 es un algoritmo recursivo para la construcción de árboles, la idea es elegir para cada nodo el atributo que más ganancia de información (menos entropía) proporcione, intuitivamente equivale a que el valor de ese atributo (o rango de valor para el caso continuo) ayude a separar lo más posible (por etiqueta) los elementos del conjunto de entrenamiento. Se dividen los datos según el valor de ese atributo y se crean de forma recursiva árboles con el resto de atributos partiendo de los hijos, en cada hijo todos los datos presentarán el mismo (o cercano en el caso continuo) valor del atributo elegido para el nodo padre. Se realiza este proceso hasta que todos los hijos tengan la misma (o cercana en el caso continuo) etiqueta o hasta que se agoten las características.

La clase de funciones que ajustan son árboles, funciones escalonadas de la forma $f(\mathbf{x}) = \sum_{i=1}^{M} c_i \mathbb{I}[x \in R_i]$ donde las R_i son regiones del espacio separadas por hiperplanos paralelos a los ejes. Esta clase de funciones tiene dimensión VC infinita, lo que hace que los árboles tengan poco sesgo pero mucha varianza. Son propensos al sobreajuste y se debe aplicar alguna técnica para reducir su variabilidad.
La complejidad de un árbol viene determinada por el número de nodos: a mayor número de nodos menor será el sesgo y mayor la varianza, por tanto hay que buscar un equilibrio entre ambos. Para ello hay que intentar reducir el número de nodos sin dejar de ajustar razonablemente bien los datos de entrenamiento, ya que cuanto mayor sea el número de nodos más propenso será el árbol al sobreajuste. Esto puede hacerse mientras se construye el árbol: en la fase de calcular la ganancia de información de cada atributo, si ninguno proporciona una ganancia significativa se deja de desarrollar el árbol (**early stopping**). También puede realizarse después de construir el árbol: se revisa la entropía de cada nodo y se eliminan los nodos que no aportan una ganancia de información significativa (**post-pruning**). Puede ser útil usar validación para determinar qué nodos contribuyen al sobreajuste y pueden ser podados.

7. 7

La primera ventaja que tiene SVM-Hard frente a separadores lineales como los que hemos estudiado anteriormente es su menor complejidad y por tanto menor dimensión VC. Si el radio del separador es ρ se tiene:
$d_{VC}(\rho) \leq \text{mín}\left(\left\lceil \frac{R^2}{\rho^2} \right\rceil, d\right) + 1$
y la de los separadores lineales que conocíamos a continuación $d+1$. Donde d es el número de atributos (la dimensión de los puntos de la muestra) y R es el máximo de las normas de los puntos de la muestra. Esta menor dimensión VC implica una cota de generalización más fina.
La segunda ventaja que presenta se manifiesta a la hora de acotar el error de validación cruzada. Tenemos
$E_{CV} = \frac{1}{N} \sum_{n=1}^{N} e_n$
Pero $e_n = 0$ para los puntos que no son de soporte, y (en clasificación binaria) se tiene $e_n \leq 1$ para los puntos de soporte. Por tanto SVM-Hard tiene una cota del error de validación cruzada más baja (sólo se tienen en cuenta los puntos de soporte, el resto no influyen en la solución)
$E_{CV}(SVM) \leq \frac{\#\text{ support vectors}}{N}$

SVM-Hard tiene una gran desventaja, el algoritmo requiere la resolución de un problema de optimización cuadrática en el que la matriz de los términos cuadráticos es de orden $N \times N$ y está formada por productos escalares de cada par de puntos de la muestra. Estos productos escalares son bastante costosos computacionalmente cuando el número de variables a considerar (d) es alto. Luego el algoritmo no es adecuado para problemas de alta dimensionalidad.
SVM-Hard requiere que los datos sean separables y añadir clases de funciones más complejas para separar los datos puede producir sobreajuste. Ante la existencia de ruido en los datos, SVM-Soft permite asumir cierto margen de error sin complicar demasiado las hipótesis, permitiendo una mejor generalización. Funciona de forma similar a una regularización.

8. 8

Un clasificador Random Forest ajusta varios árboles y predice la clase más votada con el objetivo de reducir la variabilidad respecto a la muestra a cambio de un ligero aumento en el sesgo. Para conseguir esto los árboles deben estar lo menos correlados posible, para ello se combinan dos técnicas:
Para ajustar cada árbol se utiliza sólo una submuestra de los datos. Estas submuestras se eligen con reemplazamiento. (**Bagging**)
Cada árbol sólo utiliza un subconjunto de las características elegido aleatoriamente.
Esta construcción provoca que RF presente una baja variabilidad y generalice de forma más precisa que otros clasificadores ante problemas con ruido.
Además, por ajustar árboles su dimensión VC es infinita y permite conseguir un menor sesgo que otros clasificadores lineales más simples como SVM. También presenta la ventaja de no necesitar adaptación para afrontar problemas de clasificación no binaria.
RF es un clasificador muy eficaz en la práctica, pero el teorema No-Free-Lunch nos asegura que no es óptimo.

9. 9

Si disponemos de medios para estimar el número total de peces como https://en.wikipedia.org/wiki/Mark_and_recapture, entonces basta con estimar el peso medio de cada pez. Para esto habría que tomar un número suficientemente grande de muestras independientes, identicamente distribuidas y representativas de la población.
El problema es que probablemente exista una relación entre el tamaño de los peces y lo propensos que sean a caer en la red, ya que sólo contamos con las redes de uso comercial, que tienen las medidas para capturar peces adultos. Por lo que la muestra que tomaríamos estaría sesgada y no podríamos garantizar resultados correctos.
En el caso de no disponer de un método para estimar el número total de peces, es imposible que sepamos el número de kilos de pescado de la piscifactoría, pues desconocemos el porcentaje de la población que representa cada muestra.
En conclusión, no podemos garantizar una estimación de los kilos de pescado presentes y futuros a partir de muestras. Nuestra estimación estaría sesgada por la relación entre el tamaño de los peces y su propensidad a ser capturados.

10. 10

Un punto del conjunto (x_n, y_n) está mejor clasificado por el hiperplano dado por los pesos $\mathbf{w}$ cuanto mayor sea el valor de $y_n \mathbf{w}^T x_n$ (la distancia del punto al hiperplano es esa expresión dividida por la norma de $\mathbf{w}$). El nuevo algoritmo busca el punto peor clasificado, que es en el que la expresión $y_n \mathbf{w}^T x_n$ alcanza su mínimo, y modifica los pesos con el fin de aumentar el valor de esa expresión: (llamando $w^{(t)}$ al vector de pesos en la etapa t)
$y_n(\mathbf{w}^{(t+1)})^T x_n = y_n((\mathbf{w}^{(t)})^T + y_n x_n^T)x_n = y_n(\mathbf{w}^{(t)})^T x_n + y_n^2 x_n^T x_n > y_n(\mathbf{w}^{(t)})^T x_n$
Por tanto el objetivo del algoritmo es separar el hiperplano de los puntos más próximos, al igual que SVM (aunque lo haga de forma diferente). Se obtendrá una solución que maximice la separación del hiperplano a los puntos más cercanos. Por lo que en general presentará menor variabilidad que la

proporcionada por el Perceptron. Pero para encontrarla se debe cambiar el criterio de parada, ya que el algoritmo seguirá mejorando la solución después de que todos los puntos estén bien clasificados.
Tras varias iteraciones, el algoritmo acabará iterando sólo sobre los puntos más próximos al hiperplano, los puntos de soporte.

11. 11

La solución del problema Dual consta del vector óptimo de multiplicadores de Lagrange $(\alpha_1^*, \ldots, \alpha_N^*)$. A partir de estos valores podemos recuperar la solución del problema primal: el vector de pesos óptimo $\mathbf{w}^*$:
$\mathbf{w}^* = \sum_{n=1}^{N} \alpha_n y_n x_n$
y la condición KKT de Complementary slackness nos permite despejar b^*, ya que para $\alpha_m^* > 0$ se verifica $y_m(x_m \mathbf{w}^* + b^*) = 1$
Además, nos proporciona información sobre la posición de las muestras respecto al hiperplano, ya que si $\alpha_n > 0$ podemos asegurar que x_n, y_n está a distancia mínima del hiperplano (se trata de un punto de soporte).

12. 12

Supongamos que participan k corredores en cada carrera. El objetivo de la persona que nos envió el e-mail es ganarse nuestra confianza acertando el ganador durante 7 carreras seguidas, el número de posibles combinaciones de ganadores es k^7.
Es perfectamente posible que la persona seleccionase k^7 hipótesis distintas, una con cada combinación de ganadores, y enviase cada hipótesis a una persona diferente, de hecho sólo tendría que seguir enviando correos a las personas que hayan recibido respuestas acertadas hasta esa semana.
Como hemos indicado, el número de posibles combinaciones es k^7, que coincide con el número de hipótesis diferentes. Por tanto $m_{\mathcal{H}}(N) = k^N$ para $N \leq 7$. Tenemos una muestra de 7 carreras y la dimensión VC del conjunto de hipótesis puede ser perfectamente $d_{VC} = 7$, luego no es de extrañar que haya clasificado bien las siete carreras y no podemos asegurar que podrá generalizar (acertar nuevas carreras) correctamente. Por tanto NO merece la pena pagar.
Me he basado en el Ejercicio 5.2 de Learning From Data.