Aprendizaje Automático (2016-2017)
Grado en Ingeniería Informática
Universidad de Granada

Cuestionario de teoría 1

David Criado Ramón

26 de marzo de 2017

1. Identificar, para cada una de las siguientes tareas, qué tipo de aprendizaje es el adecuado (supervisado, no supervisado, por refuerzo) y los datos de aprendizaje que deberíamos utilizar en cada caso. Si una tarea se ajusta a más de un tipo, explicar cómo y describir los datos para cada tipo.

1.1. Dada una colección de fotos de caras de distintas razas establecer cuantas razas distintas hay representadas en la colección.

En este caso nos encontramos en un caso de aprendizaje no supervisado puesto que queremos identificar el número de razas distintas que puede haber y no en el número de individuos que pertenecen a ciertas razas en concreto (en el que optaríamos por aprendizaje supervisado). Podemos utilizar datos que podamos deducir de alguna manera a partir de las imágenes, como la intensidad del color, la simetría de la cara, o la forma de los ojos.

1.2. Clasificación automática de cartas por distrito postal.

En este caso estamos ante un problema de aprendizaje supervisado. Nuestro objetivo es identificar los caracteres que conforman el distrito postal en la carta para luego juntando los números obtenidos saber a qué zona corresponde. Este problema es similar al realizado en el trabajo de prácticas y podríamos utilizar como datos de entrada la intensidad del color y la simetría vertical junto al etiquetado de las muestras de entrenamiento (puesto que es supervisado).

1.3. Determinar si un determinado índice del mercado de valores subirá o bajará dentro de un período de tiempo determinado.

En este caso puesto que la información previa que tengamos puede no ser relevante en la actualidad optaría por aprendizaje por refuerzo, que intentará adaptar sus predicciones en base a la función objetivo que vendrá determinada por la diferencia entre que el predictor acierte o falle.

1.4. Aprender un algoritmo que permita a un robot rodear un obstáculo.

Este sería otro caso de aprendizaje por refuerzo, en este caso, es muy fácil comparárlo con el ejemplo básico de aprender a montar en la bicicleta. Nuestro robot (en un entorno simulado), cuya función objetivo es alcanzar un punto al otro lado del obstáculo (probablemente sea interesante que lo haga de la manera más eficiente posible) sin producirse ninguna colisión (valorando la colisión con un valor negativo elevado en la función objetivo).

2. ¿Cuáles de los siguientes problemas son más adecuados para una aproximación por aprendizaje y cuáles más adecuados para una aproximación por diseño? Justificar la decisión.

2.1. Definir los grupos de animales vertebrados en pájaros, mamíferos, reptiles, aves y anfibios.

Puesto que la clasificación de los animales está determinada por una modelo de características físicas bien definido creo que lo correcto es una aproximación por diseño, en la que, aparte de ese modelo de características hemos de tener en cuenta las pocas excepciones que haya para la regla general de clasificación.

2.2. Determinar si se debe aplicar una campaña de vacunación contra una enfermedad.

Este problema requiere de una aproximación por aprendizaje. Basándonos en los resultados de años anteriores (porcentaje de población enferma, porcentaje de población sobre la que se aplicó la vacuna, coste de la vacuna y si se llevó a cabo la campaña o no durante ese año), intentamos deducir a partir de los datos del año actual si ha de aplicarse la campaña.

2.3. Determinar si un correo electrónico es de propaganda o no.

Este problema requiere de una aproximación por aprendizaje puesto que aun habiendo características que sean indicativos de que el correo es de propaganda, no sigue un modelo constante. Algunas de estas características pueden basarse en dominios para el correo electrónico extraños, la cantidad de contenido multimedia (imágenes, etc) presentes en el mismo o la frecuencia de ciertas palabras propias de los correos electrónicos de este tipo.

2.4. Determinar el estado de ánimo de una persona a partir de una foto de su cara.

Similarmente este es otro problema que requiere de una aproximación por aprendizaje, puesto que los rasgos que pueden definir el estado de ánimo de una persona no sigue un modelo exacto y pueden variar de unas personas a otras.

3. Construir un problema de *aprendizaje desde datos* para un problema de clasificación de frutas en una explotación agraria que produce mangos, papayas y guayabas. Identificar y definir los elementos formales del problema χ, Y, f de manera que puedan ser usados por un computador. ¿Considera que en este problema estamos en un caso de etiquetas con ruido o sin ruido?

χ es el dominio del vector de características, el espacio de entrada. Es el conjunto de datos de entrada del problema del que intentamos aprender. Elementos como el color, la forma, el peso, la textura o el tamaño podrían forma parte este espacio n-dimensional (*donde n es el número de características de entrada*)
Y es el dominio de la salida de nuestro problema, el espacio de salida. Como nos encontramos en un problema de clasificación podemos suponer que devuelve 1 para mango, 2 para papaya y 3 para guayaba siendo $Y = \{1, 2, 3\}$
f es la función que resuelve el problema y que para cada elemento del espacio χ devuelve la salida correcta del espacio Y. En el ámbito del aprendizaje automático f es siempre desconocida, puesto que si la conociésemos una aproximación por aprendizaje no nos daría ninguna ventaja frente a una aproximación por diseño basada en f.
Bajo mi punto de vista estamos ante un caso de etiquetas con ruido puesto que, aunque no sea un experto en el tema, es muy probable que un ejemplar de una fruta bajo ciertas condiciones tenga un gran parecido con un ejemplar de otras de las frutas bajo condiciones similares.

4. La regla de adaptación de los pesos del Perceptron ($w_{new} = w_{old} + yx$) tiene la interesante propiedad de que los mueve en la dirección adecuada para clasificar x de forma correcta. Suponga el vector de pesos w de un modelo y un dato x(t) mal clasificado respecto de dicho modelo. Probar que la regla de adaptación de pesos siempre produce un movimiento en la dirección correcta para clasificar bien x(t).

Partimos de la expresión $y(t)w^T(t)x(t)$, puesto que debería ocurrir que $y(t) = signo(w^T(t)x(t)$ la primera expresión será estrictamente menor que 0 en el caso de un mal clasificado. Partiendo de ésto intentemos demostrar que en la proxima iteración la clasificación será mejor que en la iteración actual, esto quiere decir que $y(t)w^T(t+1)x(t) > y(t)w^T(t)x(t)$. Vamos a expandir el primer miembro de la inecuación aplicando la regla del algoritmo PLA. $y(t)w^T(t+1)x(t) = y(t)(w(t) + y(t)x(t))^T x(t) = y(t)w^T(t)x(t) + x^2(t)$. Es este último

término al estar elevado al cuadrado el que asegura que en cada iteración la primera expresión que era menor que 0 si estaba mal clasificado nos va a indicar que en cada iteración nos iremos acercado más a que la expresión sea mayor que 0 y, por tanto, que el Perceptron clasifique correctamente el problema.

5. La desigualdad de Hoeffding modificada nos da una forma de caracterizar el error de generalización con una cota probabilística $\mathbb{P}[|E_{out}(g) - E_{in}(g)| > \epsilon] \leq 2Me^{-2N\epsilon^2}$ para cualquier $\epsilon > 0$. Si fijamos $\epsilon = 0{,}05$ y queremos que la cota probabilística $2Me^{-2N\epsilon^2}$ sea como máximo 0.03, ¿cuál será el valor más pequeño de N que verifique estas condiciones para M = 1? Repetir para M = 10 y M = 100.

Primero vamos a despejar la N de la expresión proporcionada.

$2Me^{-2N\epsilon^2} = 0{,}03$
$e^{-2N\epsilon^2} = 0{,}015/M$
$-2N\epsilon^2 = ln(0{,}015/M)$
$N = -ln(0{,}015/M)/2\epsilon^2$

Aplicamos la fórmula despejada para los distintos M y nos quedamos con el número natural inmediatamente superior si no fuese un natural el resultado.
Para $M = 1$ => $N = 839{,}94 = 840$
Para $M = 10$ => $N = 1300{,}45 = 1301$
Para $M = 100$ => $N = 1760{,}98 = 1761$

6. El jefe de investigación de una empresa con mucha experiencia en problemas de predicción de datos tras analizar los resultados de los muchos algoritmos de aprendizaje usados sobre todos los problemas en los que la empresa ha trabajado a lo largo de su muy dilatada existencia, decide que para facilitar el mantenimiento del código van a seleccionar un único algoritmo y una única clase de funciones con la que aproximar todas las soluciones a sus problemas presentes y futuros. ¿Considera que dicha decisión es correcta y beneficiará a la empresa? Argumenta la respuesta usando los resultados teóricos estudiados.

El enunciado nos permite saber que utiliza una única clase de funciones y un único algoritmo como llevamos hasta ahora estudiado en teoría, por tanto, sabemos que la diferencia entre el error fuera de la muestra y el error dentro de la muestra viene acotada por al desigualdad de Hoeffding, en concreto el factor que va a limitar su error va a ser la dimensión de Vapnik-Chervonenkis de la clase de funciones escogidas, por lo que a priori podría saber el tamaño muestral necesario para obtener una diferencia de error objetivo. No obstante, esto pone en riesgo que si la clase de funciones seleccionada no se ajusta bien al problema en la fase de entrenamiento y la empresa disponía de otras funciones que pudieran ajustarse mejor en entrenamiento perdería posibles soluciones que actuasen mucho mejor que la seleccionada por lo que concluyo que, aunque en algunos problemas te pueda permitir tener más garantías, va a haber casos que antes se podían haber resuelto y ahora no y, por tanto, que no beneficiaría a la empresa tomar esa decisión.

7. Para un conjunto H con $d_{vc} = 10$, ¿qué tamaño muestral se necesita (según la cota de generalización) para tener un 95 % de confianza de que el error de generalización sea como mucho 0.05

Para calcularlo, aplicamos la fórmula explicada en teoría para determinar el valor de N. Dicha fórmula es $N \geq 8ln(4 * [(2N)^{d_{vc}} + 1]/\delta)/\epsilon^2$. Para realizar los cálculos utilizo el siguiente progama en C++

```
#include <cmath>
#include <iostream>

using namespace std;
```

```
double aplicarFormula(double epsilon, double delta, int d_vc, double N) {
    return 8* log(4*(pow(2*N, d_vc) + 1)/ delta) / pow(epsilon,2);
}


int main(int argc, char *argv[])
{

    double epsilon = 0.05;
    double delta = 0.05;
    int d_vc = 10;
    double N = 1000;
    double error = 0.000001;

    double N_nuevo = aplicarFormula(epsilon,delta,d_vc,N);

    // Seguimos iterando hasta que haya poca diferencia (1e-6)
    while (abs(N_nuevo - N) > error) {
        N = N_nuevo;
        N_nuevo = aplicarFormula(epsilon,delta,d_vc,N);
    }

    cout << "El tamaño muestral es " << ceil(N) << '\n';
}
```

Obteniendo que el tamaño muestral ha de ser 452957.

8. Identificar de forma precisa las dos condiciones que garantizan que un problema de predicción puede ser aproximado por inducción desde una muestra de datos y una clase de funciones. Justificar la respuesta usando los resultados teóricos estudiados.

- Los datos del conjunto de datos D son muestras independientes e idénticamente distribuidas en una distribución de probabilidad P desconocida, puesto que si no ocurriése podríamos acabar en un caso de aprendizaje sobre una región concreta de la función, haciendo que luego cuando se pruebe en test los errores para las regiones que no han sido sobreaprendidas sean mucho mayores.
- El término M de la desigualdad de Hoeffding, esto según concluíamos al aplicarlo sobre el SRM quiere decir que la dimensión de Vapnik-Chervonenkis de la clase de hipótesis seleccionada ha de ser finita porque, en caso contrario, lo único que

podemos afirmar es que la probabilidad de que la diferencia fuera y dentro de la muestra sea menor que un delta es menor o igual que infinito (lo cual es evidente).

9. Considere que le dan una muestra de tamaño N de datos etiquetados {-1,+1} y le piden que encuentre la función que mejor ajuste dicho datos. Dado que desconoce la verdadera función f, discuta los pros y contras de utilizar ERM y SRM para lograr el objetivo. Valore las consecuencias de aplicar cada uno de ellos.

Utilizar ERM implica fijar un conjunto de hipótesis concreto, por tanto, si en nuestro conjunto de hipótesis no tenemos una lo suficientemente compleja no podremos ajustar bien E_{in} o en el caso de tener funciones demasiado complejas podremos encontrarnos con problemas de sobreaprendizaje. Por el otro lado SRM nos permite movernos por distintas clases de funciones delimitadas por su dimensión de Vapnik-Chervonenkis por lo que podemos ir iterando sobre ellas hasta encontrar la menos compleja que nos de el error deseado. SRM es mejor que ERM, especialmente en los casos en los que la relación tamaño muestra - dimensión Vapnik-Chervonenkis es bastante bajo (menor que 20).