UNIVERSIDAD
DE GRANADA

Aprendizaje Automático
grado en ingeniería informática

TRABAJO 1

Cuestiones de Teoría

Autor
Vladislav Nikolov Vasilev

Rama
Computación y Sistemas Inteligentes

Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación

Curso 2018-2019

Índice

Ejercicio 1

Identificar, para cada una de las siguientes tareas, cuál es el problema, qué tipo de aprendizaje es el adecuado (supervisado, no supervisado, por refuerzo) y los elementos de aprendizaje $(\mathcal{X}, f, \mathcal{Y})$ que deberíamos usar en cada caso. Si una tarea se ajusta a más de un tipo, explicar como y describir los elementos para cada tipo.

a) Clasificación automática de cartas por distrito postal.

Solución

El problema ante el que nos encontramos en este caso consiste en agrupar las cartas por una característica determinada, en este caso por código postal. Como tal, no se quiere aprender nada, si no que simplemente se quieren agrupar los datos por un criterio. Así que, para resolver este problema, podemos utilizar aprendizaje no supervisado.

El conjunto de datos de entrada $\mathcal{X}$ puede ser por ejemplo los datos del destinatario (su dirección o código postal, por ejemplo). El conjunto de etiquetas $\mathcal{Y}$ no existe como tal en el aprendizaje no supervisado, es desconocido. La función f sería alguna de función distribución de probabilidad condicional que queremos aproximar, que nos permita agrupar los datos.

b) Decidir si un determinado índice del mercado de valores subirá o bajará dentro de un período de tiempo fijado.

Solución

El problema en este caso consiste en predecir o decidir a partir de unos datos de entrada una clase (la de si subirá o bajará el índice de mercado). Por tanto este problema se puede ver como una clasificación binaria $(0, 1)$ o $(-1, 1)$.

En el caso de los datos de entrada $\mathcal{X}$ podríamos utilizar valores del mercado y el tiempo. En el caso de los datos de salida o etiquetas $\mathcal{Y}$ podríamos tener las etiquetas $(-1, 1)$, siendo -1 el caso de bajar el índice y 1 el de subir. Y por último, f sería una función que relacione a $\mathcal{X}$ y a $\mathcal{Y}$ tal que $f: \mathcal{X} \mapsto \mathcal{Y}$.

c) Hacer que un dron sea capaz de rodear un obstáculo.

Solución

El problema en este caso es hacer que un dron aprenda a esquivar un obstáculo rodeándolo. Como el objetivo no es clasificar ninguna información, ni predecir ningún valor real ni buscar características o patrones en los datos, parece que el tipo de aprendizaje más adecuado es el aprendizaje por refuerzo. Esto se podría hacer mediante un simulador en un ordenador, donde se representaría el espacio donde se quiere entrenar al dron. Una vez entrenado en este simulador, se podría transferir todo lo aprendido al dron y ver cómo se desempeña.

Como tal, el aprendizaje por refuerzo no tendría ni entradas $\mathcal{X}$, ni etiquetas de salida $\mathcal{Y}$ ni una función f para el aprendizaje, pero sí que tendría otra información que se correspondería con un Proceso de Decisión de Markov (MDP), como por ejemplo un conjunto de estados, acciones, probabilidades de transicionar de un estado a otro, una recompensa por transicionar de estado, etc.

d) Dada una colección de fotos de perros, posiblemente de distintas razas, establecer cuántas razas distintas hay representadas en la colección.

Solución

En este caso el problema consiste en encontrar patrones o características que permitan agrupar los datos (agrupar los perros según su raza, para saber cuántas hay); es decir, encontrar características que permitan dividir los perros según su raza. Por tanto, como queremos aprender esas características, deberíamos utilizar el aprendizaje supervisado, y al acabar de clasificar nuestros datos, ver cuántas categorías están presentes (cuántas razas de perros hemos obtenido de los datos).

En este caso, $\mathcal{X}$ son los datos de los que dispondríamos (las fotos de los perros), $\mathcal{Y}$ serían razas de perros, y f sería una función tal que $f : \mathcal{X} \mapsto \mathcal{Y}$.

Ejercicio 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.

a) Determinar si un vertebrado es mamífero, reptil, ave, anfibio o pez.

Solución

Este problema parece ser más adecuado para el diseño, ya que si se conocen qué características diferencian a los distintos animales, no hace falta aprender nada,

solo aplicarlas. Además, por lo general, el problema suele ser bien conocido, con lo cuál la mayoría de características son conocidas, y solo haría falta ajustar unos pocos parámetros para distinguir ciertos casos.

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

Solución

Este problema parece que puede ser aproximado mejor por diseño que por aprendizaje, ya que es un problema conocido, que es la aplicación de una campaña de vacunas, y solo queremos ajustar algún parámetro, como por ejemplo sería el umbral de personas enfermas desde el que se aplicaría. No haría falta aprender todo el modelo, solo ajustar ese dato. cumplirse una condición que se aplique.

c) Determinar perfiles de consumidor en una cadena de supermercados.

Solución

Para este porblema lo mejor es el aprendizaje, el aprendizaje no supervisado en concreto. No conocemos a priori cuántos perfiles hay y como distinguirlos, pero podemos aplicar alguna técnica de aprendizaje no supervisado con el objetivo de encontrar patrones que permitan distinguir unos perfiles de otros y ver a cuál pertenece un individuo.

d) Determinar el estado anímico de una persona a partir de una foto de su cara.

Solución

La mejor aproximación que se puede seguir en este caso es el aprendizaje, ya que como tal no conocemos exactamente qué detalles de una expresión facial determinan el estado anímico. Si las supiésemos, podríamos simplemente codificar el diseño de éstas, pero como no las sabemos, optaremos por aprender de los datos. Se puede seguir alguna técnica de aprendizaje supervisado o no supervisado para determinar dichos detalles.

e) Determinar el ciclo óptimo para las luces de los semáforos en un cruce con mucho tráfico.

Solución

En este caso, la mejor aproximación que podemos seguir es el aprendizaje, y más concretamente, el aprendizaje por refuerzo. Esto se debe a que el entorno de un semáforo es caótico, e intentar hacerlo por diseño sería demasiado difícil, ya que habría que considerar todos los posibles casos que se pueden dar. Para hacer este aprendizaje, se puede construir un simulador donde entrenar un semáforo mediante aprendizaje por refuerzo para que aprenda cuál sería el ciclo óptimo de luces para un determinado cruce con mucho tráfico. Después, se podría trasladar todo lo aprendido al semáforo.

Ejercicio 3

Construir un problema de *aprendizaje desde datos* para un problema de clasificación de fruta en una explotación agraria que produce mangos, papayas y guayabas. Identificar los siguientes elementos formales $\mathcal{X}, \mathcal{Y}, \mathcal{D}, f$ del problema. Dar una descripción de los mismos que pueda ser usada por un computador. ¿Considera que en este problema estamos ante un caso de etiquetas con ruido o sin ruido? Justificar las respuestas.

Solución

Vamos a suponer que nos encontramos ante un problema de clasificación, y por tanto, de aprendizaje supervisado. Para construir nuestro modelo podemos considerar los siguientes elementos:

- $\mathcal{X}$ sería el vector de características de las frutas. Podríamos considerar características tales como el **color**, la **forma**, el **tamaño** y la **textura**.
  - El **color** se podría codificar como una categoría, de tal forma que solo pudiese tomar un valor, como por ejemplo 0 para el verde, 1 para el amarillo y 2 para el verde.
  - La **forma**, al igual que el color, podría tomar un valor categórico, siendo 0 redonda y 1 ovalada.
  - El **tamaño** puede ser también una variable categórica, tomando los valores 0 para pequeño y 1 para grande.
  - La **textura** también puede verse como una variable categórica, pudiendo tomar los valores 0 para lisa y 1 para granulada.
- $\mathcal{Y}$ serían los valores de las etiquetas. Podríamos tener 0 para el **mango**, 1 para la **papaya** y 2 para la **guayaba**.

- $\mathcal{D}$ podría ser en este caso un conjunto de vectores de características con sus correspondientes etiquetas, es decir una muestra, con la cuál pordríamos entrenar nuestro modelo. Es muy importante que sea una muestra independiente (un elemento no condiciona a los otros) e idénticamente distribuida (cada elemento de la muestra tenga la misma probabilidad).
- f sería nuestra función objetivo, una función desconocida que permitiese asignar las etiquetas a nuestras entradas, es decir, que $f: \mathcal{X} \mapsto \mathcal{Y}$.

En este caso podríamos encontrarnos ante un caso de etiquetas con ruido. Por ejemplo, puede que debido a factores que no hayamos considerado a la hora de establecer las características usadas en $\mathcal{X}$ nos encontremos con que hayan dos frutas con las mismas características, pero que sin embargo luego se hayan clasificado en distintas clases (como puede ser que en algún caso haya habido algúna anomalía durante el crecimiento de una de las frutas y haga que tenga características similares a las de una fruta de la otra clase).

Ejercicio 4

Suponga una matriz cuadrada A que admita la descomposición $\mathrm{A} = \mathrm{X}^{\mathrm{T}}\mathrm{X}$ para alguna matriz X de números reales. Establezca una relación entre los valores singulares de las matriz A y los valores singulares de X.

Solución

Vamos a partir de que la matriz X puede descomponerse en valores singulares de la forma:

$$X = UDV^T \tag{1}$$

donde encontramos que:

- U es una matriz ortogonal, y por tanto, $U^{-1} = U^T$.
- D es una matriz diagonal que contiene los valores singulares de X en su diagonal principal ordenados de mayor a menor.
- V es una matriz ortogonal, de forma que $V^{-1} = V^T$

Al sustituir los valores de X en la descomposición original por la descomposición mostrada en (1), obtenemos que:

$$A = X^T X = (UDV^T)^T(UDV^T) = VDU^T UDV^T = VDDV^T = VD^2V^T \quad (2)$$

Por tanto, al haber supuesto que la matriz A se podía descomponer en $X^T X$, podemos suponer que el resultado obtenido en (2) se corresponde con la descomposición en valores singulares de A. Como hemos partido sustituyendo X por su descomposición en valores singulares, y sabiendo que los valores propios de la matriz X están contenidos en D, entonces podemos decir que los valores propios de A son los de X al cuadrado.

Ejercicio 5

Sean $\mathbf{x}$ e $\mathbf{y}$ dos vectores de características de dimensión $M \times 1$. La expresión

$$\text{cov}(\mathbf{x}, \mathbf{y}) = \frac{1}{M}\sum_{i=1}^{M}(x_i - \overline{x})(y_i - \overline{y}) \quad (3)$$

define la covarianza entre dichos vectores, donde $\overline{z}$ representa el valor medio de los elementos de $\mathbf{z}$. Considere ahora una matriz X cuyas columnas representan vectores de características. La matriz de covarianzas asociada a la matriz X = $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ es el conjunto de covarianzas definidas por cada dos de sus vectores columnas. Es decir,

$$\text{cov}(\mathbf{X}) = \begin{pmatrix} \text{cov}(\mathbf{x}_1, \mathbf{x}_1) & \text{cov}(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \text{cov}(\mathbf{x}_1, \mathbf{x}_N) \\ \text{cov}(\mathbf{x}_2, \mathbf{x}_1) & \text{cov}(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \text{cov}(\mathbf{x}_2, \mathbf{x}_N) \\ \cdots & \cdots & \cdots & \cdots \\ \text{cov}(\mathbf{x}_N, \mathbf{x}_1) & \text{cov}(\mathbf{x}_N, \mathbf{x}_2) & \cdots & \text{cov}(\mathbf{x}_N, \mathbf{x}_N) \end{pmatrix} \quad (4)$$

Sea $\mathbf{1}^T = (1, 1, \ldots, 1)$ un vector $M \times 1$ de unos. Mostrar que representan las siguientes expresiones:

a) $E1 = \mathbf{1}\mathbf{1}^T\text{X}$

Solución

Sabiendo que $\mathbf{1}$ es un vector $M \times 1$, $\mathbf{1}^T$ es un vector $1 \times M$ y que X es una matriz $M \times N$, podemos aplicar la propiedad asociativa para multiplicar $\mathbf{1}\mathbf{1}^T$, con lo cuál obtendríamos una matriz $M \times M$ de unos. Por tanto, al multiplicar ahora la matriz de unos por X, como éstas tienen dimensiones $M \times M$ y $M \times N$ respectivamente, obtenemos una matriz $M \times N$ en la que todos los elementos de una columna son la suma de los elementos de esa columna. Es decir, para la columna j-ésima de la matriz resultado (de forma que $j \in [1, 2, \ldots, N]$), el elemento i-ésimo de esa columna (de manera que $i \in [1, 2, \ldots, M]$), sería la suma de todos los elementos de la columna j-ésima de la matriz X original.

b) $E2 = (\mathrm{X} - \frac{1}{M}E1)^T(\mathrm{X} - \frac{1}{M}E1)$

Solución

Si comenzamos a operar dentro de los paréntesis, podemos ver que la primera operación que podemos realizar es el producto de $E1$ por un escalar.

Como sabemos de antes, $E1$ es una matriz en la que todos los elementos de una columna son la suma de todos los elementos de la columna correspondiente en X. Al realizar el producto por un escalar, en este caso $\frac{1}{M}$, realmente estamos calculando, para cada elemento de la matriz, la media, ya que como se ha dicho antes, cada elemento de una columna es el sumatorio de los M elementos de la misma columna de X. Con lo cuál, ahora cada elemento de una columna contendrá la media de la suma de los elementos de la misma columna de X. Llamémos a esta matriz $\overline{X}$.

La siguiente operación que podemos realizar, aun dentro de los paréntesis, es $\mathrm{X} - \overline{X}$ (lo que se corresponde con $\mathrm{X} - \frac{1}{M}E1$). Con esto, lo que obtenemos es la diferencia de cada elemento de X con respecto a la media, es decir su desviación con respecto a la media. Esto se puede ver de la siguiente forma:

$$\begin{aligned} X - \overline{X} &= \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ x_{M1} & x_{M2} & \cdots & x_{MN} \end{pmatrix} - \begin{pmatrix} \overline{x}_1 & \overline{x}_2 & \cdots & \overline{x}_N \\ \overline{x}_1 & \overline{x}_2 & \cdots & \overline{x}_N \\ \cdots & \cdots & \cdots & \cdots \\ \overline{x}_1 & \overline{x}_2 & \cdots & \overline{x}_N \end{pmatrix} \\ &= \begin{pmatrix} x_{11} - \overline{x}_1 & x_{12} - \overline{x}_2 & \cdots & x_{1N} - \overline{x}_N \\ x_{21} - \overline{x}_1 & x_{22} - \overline{x}_2 & \cdots & x_{2N} - \overline{x}_N \\ \cdots & \cdots & \cdots & \cdots \\ x_{M1} - \overline{x}_1 & x_{M2} - \overline{x}_2 & \cdots & x_{MN} - \overline{x}_N \end{pmatrix} \end{aligned} \tag{5}$$

Habiendo calculado esto, ahora solo nos queda calcular el producto. La traspuesta de la matriz que se ha obtenido anteriormente es la siguiente (llamemos

X_{dev} a esta matriz, para darle un nombre):

$$X_{dev}^T = \begin{pmatrix} x_{11} - \overline{x}_1 & x_{21} - \overline{x}_1 & \cdots & x_{M1} - \overline{x}_1 \\ x_{12} - \overline{x}_2 & x_{22} - \overline{x}_2 & \cdots & x_{M2} - \overline{x}_2 \\ \cdots & \cdots & \cdots & \cdots \\ x_{1N} - \overline{x}_N & x_{2N} - \overline{x}_N & \cdots & x_{MN} - \overline{x}_N \end{pmatrix} \tag{6}$$

Al realizar la multiplicación $(X_{dev}^T X_{dev})$, lo que obtenemos en realidad no es nada más ni nada menos que una expresión parecida a la covarianza que se puede ver en (4). Es decir, al multiplicar la fila i-ésima de $\mathrm{X}_{dev}^{\mathrm{T}}$ por la columna j-ésima X_{dev}, se tiene que para el elemento $E2_{ij}$ (el resultado de la multiplicación anterior):

$$E2_{ij} = \sum_{k=1}^{M} (x_{ik} - \overline{x}_i)(x_{kj} - \overline{x}_j) \tag{7}$$

para $i,\ j \in [1, 2, \ldots, N]$ (en $\mathrm{X}_{dev}^{\mathrm{T}}$ hay N filas porque es una matriz $N \times M$, y en X_{dev} hay N columnas porque es una matriz $M \times N$). Lo que pasa es que cada elemento $E2_{ij}$ es una sumatoria, pero en ningún momento ha sido dividida entre M para obtener la covarianza. Por eso, para poder expresar el resultado anterior en función de la covarianza, podemos suponer que cada $E2_{ij}$ se ha visto multiplicado por M, anulando por tanto el $\frac{1}{M}$ que se ve en (3). Así podemos expresar que para cada $E2_{ij}$:

$$E2_{ij} = M\mathrm{cov}(\mathbf{x}_i, \mathbf{x}_j) \tag{8}$$

Así que, sabiendo que cada E_{ij} se ve multiplicado por M, podemos sacar ese M fuera de la matriz y dejar el resultado como producto de una matriz por un escalar, siendo el resultado el siguiente:

$$E2 = M\mathrm{cov}(\mathrm{X}) \tag{9}$$

Ejercicio 6

Considerar la matriz **hat** definida en regresión, $\hat{\mathrm{H}} = \mathrm{X}(\mathrm{X}^{\mathrm{T}}\mathrm{X})^{-1}\mathrm{X}^{\mathrm{T}}$, donde X es la matriz de observaciones de dimensión $N \times (d+1)$, y $\mathrm{X}^{\mathrm{T}}\mathrm{X}$ es invertible. Justificar las respuestas.

a) ¿Que representa la matriz $\hat{\mathrm{H}}$ en un modelo de regresión?

Solución

La matriz $\hat{\mathrm{H}}$ es una matriz de proyección que transforma el vector de valores reales y al vector de valores predichos $\hat{\mathbf{y}}$. Dicho de otra forma, es una matriz que pondera cada uno de los valores reales que se tienen para ver cómo influyen en obtener un nuevo valor predicho.

b) Identifique la propiedad más relevante de dicha matriz en relación con regresión lineal.

Solución

La propiedad más importante de esta matriz es la idempotencia. Es decir, se da que $\hat{\mathrm{H}}^2 = \hat{\mathrm{H}}$. Esto se puede ver de la siguiente forma:

$$\begin{aligned} \hat{\mathrm{H}}^2 &= X(X^TX)^{-1}X^TX(X^TX)^{-1}X^T \\ &= X(X^TX)^{-1}(X^TX)(X^TX)^{-1}X^T \\ &= X(X^TX)^{-1}X^T = \hat{\mathrm{H}} \end{aligned} \tag{10}$$

Esta propiedad se puede ver de la siguiente forma. Si se intenta calcular $\hat{\mathbf{y}}$, es decir, las etiquetas predichas, a partir los y reales, es decir, las etiquetas reales, tenemos que:

$$\hat{\mathbf{y}} = \hat{\mathrm{H}}y \tag{11}$$

Al volver a intentar predecir $\hat{\mathbf{y}}$ con los valores que acabamos de predecir, obtenemos, aplicando la propiedad de la idempotencia:

$$\hat{\mathbf{y}} = \hat{\mathrm{H}}\hat{\mathbf{y}} = \hat{\mathrm{H}}\hat{\mathrm{H}}y = \hat{\mathrm{H}}^2y = \hat{\mathrm{H}}y \tag{12}$$

Con lo cuál, de aquí podemos deducir que si predecimos unos valores con los valores reales y luego intentamos volver a predecir con esos valores predichos, el resultado será exactamente el mismo, independientemente de la cantidad de datos que tengamos y del número de veces que hagamos esta predicción.

Ejercicio 7

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

Solución

Vamos a empezar diciendo que el vector $\mathbf{w}$ es normal al hiperplano que separa los puntos de una clase con los de otra.

La actualización de $\mathbf{w}$ depende del producto escalar $\mathbf{w^T x}$, que es el que nos da la el y predicho. Este producto se puede expresar de la siguiente forma:

$$\mathbf{w^T x} = |\mathbf{w}||\mathbf{x}| \cos(\alpha) \tag{13}$$

siendo α el ángulo entre los dos vectores. Este ángulo es importante, porque nos indica las direcciones en las que apuntan $\mathbf{w}$ y $\mathbf{x}$, además de cómo tendría que variar el vector $\mathbf{w}$ en caso de que no se clasificase bien un elemento. Vamos a proceder a ver dos ejemplos para entender mejor esto, explicando además la importancia de α en cada caso.

Por ejemplo, si se da que $\mathbf{w^T x} < 0$ y $\mathbf{x}(t) = 1$, nos encontramos ante un caso en el que no se ha clasificado correctamente el dato. De esto podemos deducir que en la expresión mostrada en (13) se da que $\alpha > 90^o$ (los módulos de los vectores son siempre positivos y lo único que puede ser negativo es el coseno, y éste es negativo cuando tiene valores entre 90^o y 270^o). Por tanto, con esto, podemos determinar que la dirección en la que apunta $\mathbf{w}$ es contraria a la que apunta $\mathbf{x}$, así que lo que hay que hacer es modificar la dirección de $\mathbf{w}$ sumándole el vector $\mathbf{x}$ con el objetivo de acercar $\mathbf{w}$ a $\mathbf{x}$. Con esto, lo que conseguimos es que α se haga menor a 90^o para así poder hacer que los dos vectores apunten en la misma dirección, y que por tanto, su producto escalar sea positivo.

La siguiente expresión muestra la actualización que se debería hacer en este caso, suponiendo que $y = 1$:

$$\mathbf{w}_{new} = \mathbf{w}_{old} + y\mathbf{x} = \mathbf{w}_{old} + \mathbf{x} \tag{14}$$

Miremos ahora el ejemplo contrario, en el que nos encontramos que $\mathbf{w^T x} > 0$

y $\mathbf{x}(t) = -1$. Como en el anterior caso, nos encontramos que no hemos predicho bien la etiqueta. En este caso, lo que sucede es que el producto escalar es positivo, y debería haber sido negativo. Por tanto, podemos determinar que $\alpha < 90^{\text{o}}$, así que nuestro objetivo es hacer que $\alpha > 90^{\text{o}}$ con el objetivo de que la siguiente vez el producto escalar sea negativo. Por tanto, lo que hay que hacer es alejar $\mathbf{w}$ de $\mathbf{x}$; dicho de otra forma, hacer que $\mathbf{w}$ y $\mathbf{x}$ apunten en direcciones contrarias. Esto se puede conseguir restándole $\mathbf{x}$ y $\mathbf{w}$, lo cuál se puede ver en la siguiente expresión, suponiendo que $y = -1$:

$$\mathbf{w}_{new} = \mathbf{w}_{old} + y\mathbf{x} = \mathbf{w}_{old} - \mathbf{x} \tag{15}$$

Ejercicio 8

Sea un problema probabilístico de clasificación binaria con etiquetas $\{0, 1\}$, es decir, $\mathrm{P}(\mathrm{Y} = 1) = h(\mathbf{x})$ y $\mathrm{P}(\mathrm{Y} = 0) = 1 - h(\mathbf{x})$, para una funión $h()$ dependiente de la muestra.

a) Considere una muestra i.i.d. de tamaño N $(\mathbf{x}_1, \ldots, \mathbf{x}_N)$. Mostrar que la función h que maximiza la verosimilitud de la muestra es la misma que minimiza:

$$E_{in}(\mathbf{w}) = \sum_{n=1}^{N} [\![y_n = 1]\!] \ln \frac{1}{h(\mathbf{x}_n)} + [\![y_n = 0]\!] \ln \frac{1}{1 - h(\mathbf{x}_n)}$$

donde $[\![\cdot]\!]$ vale 1 o 0 según sea verdad o falso respectivamente la expresión en su interior.

Solución

Vamos a partir de la expresión del *Maximum Likehood* que viene definida para dos elementos:

$$L(\mathbf{w}) = \prod_{n=1}^{N} \mathrm{P}(y_i \mid \mathbf{x}_n) = \prod_{i=1}^{N} h(\mathbf{x}_n)^{[\![y_n=1]\!]} (1 - h(\mathbf{x}_n))^{[\![y_n=0]\!]} \tag{16}$$

Para intentar llegar a una expresión parecida a la que se pide obtener (la cuál sigue el criterio *ERM*, *Empirical Risk Minimization*), podemos aplicar el logarit-

mo neperiano sobre la expresión del *ML*, aprovechando la propiedades de que el logaritmo de un producto es la suma de logaritmos y que $\ln(a^b) = b\ln(a)$:

$$\begin{aligned}
\ln\Big(L(\mathbf{w})\Big) &= \ln\left(\prod_{n=1}^{N} h(\mathbf{x}_n)^{[\![y_n=1]\!]}(1-h(\mathbf{x}_n))^{[\![y_n=0]\!]}\right) \\
&= \sum_{n=1}^{N} \ln\Big(h(\mathbf{x}_n)^{[\![y_n=1]\!]}(1-h(\mathbf{x}_n))^{[\![y_n=0]\!]}\Big) \\
&= \sum_{n=1}^{N} \ln(h(\mathbf{x}_n)^{[\![y_n=1]\!]}) + \ln\Big((1-h(\mathbf{x}_n))^{[\![y_n=0]\!]}\Big) \\
&= \sum_{n=1}^{N} \ln[\![y_n=1]\!](h(\mathbf{x}_n) + [\![y_n=0]\!]\ln\Big(1-h(\mathbf{x}_n)\Big)
\end{aligned} \tag{17}$$

La expresión que hemos obtenido de desarrollar es muy parecida a la que se había especificado en el enunciado, solo que en este caso $h()$ no está invertido. Intuitivamente, podemos afirmar que la expresión del *ERM* (la expresión del E_{in}) es la inversa de la del *ML*. Esto se puede ver de forma intuitiva de la siguiente forma: como en el *ML*, si se maximiza la expresión significa que $h()$ tiene que tener un valor muy grande para que al aplicarle el logaritmo después se siga manteniendo grande; y como $h()$ tiene un valor muy grande en esta expresión, si miramos lo que sucede en la expresión del *ERM* nos encontramos que en este caso, como $h()$ está invertido, el valor del logaritmo será muy pequeño. Con lo cuál, maximizar la expresión del *ML* es equivalente a minimizar la expresión del *ERM*.

b) Para el caso $h(x) = \sigma(\mathbf{w}^T\mathbf{x})$ mostrar que minimizar el error de la muestra en el apartado anterior es equivalente a minimizar el error muestral

$$E_{in}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} \ln\left(1 + e^{-y_n\mathbf{w}^T\mathbf{x}_n}\right)$$

Solución

Para facilitar el trabajo, tenemos que transformar las etiquetas a $\{-1, 1\}$, de forma que $\mathrm{P}(\mathrm{Y}=1) = h(\mathbf{x})$ y $\mathrm{P}(\mathrm{Y}=-1) = 1 - h(\mathbf{x})$. Entonces, partiendo de la

expresión de E_{in} proporcionada en el anterior apartado, tenemos que:

$$\begin{aligned} E_{in}(\mathbf{w}) &= \sum_{n=1}^{N} [\![y_n = 1]\!] \ln \frac{1}{h(\mathbf{x}_n)} + [\![y_n = 0]\!] \ln \frac{1}{1 - h(\mathbf{x}_n)} \\ &= \sum_{n=1}^{N} [\![y_n = 1]\!] \ln \frac{1}{h(\mathbf{x}_n)} + [\![y_n = -1]\!] \ln \frac{1}{1 - h(\mathbf{x}_n)} \end{aligned} \tag{18}$$

Ahora, sustituyendo $h(\mathbf{x}_n)$ por $\sigma(\mathbf{w}^T\mathbf{x})$, obtenemos que:

$$E_{in}(\mathbf{w}) = \sum_{n=1}^{N} [\![y_n = 1]\!] \ln \frac{1}{\sigma(\mathbf{w}^T\mathbf{x}_n)} + [\![y_n = -1]\!] \ln \frac{1}{1 - \sigma(\mathbf{w}^T\mathbf{x}_n)} \tag{19}$$

$$= \sum_{n=1}^{N} [\![y_n = 1]\!] \ln \frac{1}{\sigma(\mathbf{w}^T\mathbf{x}_n)} + [\![y_n = -1]\!] \ln \frac{1}{\sigma(-\mathbf{w}^T\mathbf{x}_n)} \tag{20}$$

$$= \sum_{n=1}^{N} \ln \frac{1}{\sigma(y_n\mathbf{w}^T\mathbf{x}_n)} \tag{21}$$

En (20) se ha aplicado una propiedad de la función sigmoide con tal de simplificar el denominador de uno de los sumandos, que es: $\sigma(-x) = 1 - \sigma(x)$. Y, en (21) se han juntado los dos sumandos en uno solo, ya que lo único diferente es el signo de la función sigmoide, la cuál depende de y_n, así que se han juntado añadiendo y_n a la expresión, que es lo que nos dará el signo.

Finalmente, ya solo nos queda sustituir la función sigmoide por su expresión original, es decir, $\sigma(y_n\mathbf{w}^T\mathbf{x}_n) = \frac{e^{y_n\mathbf{w}^T\mathbf{x}_n}}{1+e^{y_n\mathbf{w}^T\mathbf{x}_n}}$, lo cuál nos proporcionaría el siguiente

resultado:

$$\begin{aligned} E_{in}(\mathbf{w}) &= \sum_{n=1}^{N} \ln \frac{1}{\sigma(y_n \mathbf{w}^T \mathbf{x}_n)} \\ &= \sum_{n=1}^{N} \ln \frac{1}{\frac{e^{y_n \mathbf{w}^T \mathbf{x}_n}}{1+e^{y_n \mathbf{w}^T \mathbf{x}_n}}} \\ &= \sum_{n=1}^{N} \ln \left(\frac{1 + e^{y_n \mathbf{w}^T \mathbf{x}_n}}{e^{y_n \mathbf{w}^T \mathbf{x}_n}} \right) \\ &= \sum_{n=1}^{N} \ln \left(1 + \frac{1}{e^{y_n \mathbf{w}^T \mathbf{x}_n}} \right) \\ &= \sum_{n=1}^{N} \ln \left(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n} \right) \end{aligned} \tag{22}$$

Esta expresión obtenida es muy parecida a la que se ha presentado en el enunciado, con la única diferencia de que el sumatorio no está multiplicado por $\frac{1}{N}$. Este hecho, sin embargo, no influye mucho en la minimización del error muestral, ya que el objetivo es minimizar ese sumatorio, y mutliplicarlo o dividirlo por algo no va a cambiar el hecho de que se haya llegado al valor mínimo en ese sumatorio. Por tanto, al desarrollar la expresión del error en la muestra, hemos obtenido la expresión del error muestral, y por tanto, podemos afirmar que minimizar el primero será igual que minimizar el segundo.

Ejercicio 9

Derivar el error E_{in} para mostrar que en regresión logística se verifica:

$$\nabla E_{in}(\mathbf{w}) = -\frac{1}{N} \sum_{n=1}^{N} \frac{y_n \mathbf{x}_n}{1 + e^{y_n \mathbf{w}^T \mathbf{x}_n}} = \frac{1}{N} \sum_{n=1}^{N} -y_n \mathbf{x}_n \sigma(-y_n \mathbf{w}^T \mathbf{x}_n)$$

Argumentar sobre si un ejemplo mal clasificado contribuye al gradiente más que un ejemplo bien clasificado.

Solución

Partimos de que:

$$E_{in}(\mathbf{w}) = \frac{1}{N}\sum_{n=0}^{N} \ln\left(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}\right) \tag{23}$$

Vamos a calcular la derivada:

$$\nabla E_{in}(\mathbf{w}) = \frac{\partial}{\partial \mathbf{w}}\left(\frac{1}{N}\sum_{n=0}^{N} \ln\left(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}\right)\right) \tag{24}$$

$$= \frac{1}{N}\sum_{n=0}^{N} -y_n \mathbf{x}_n \frac{e^{-y_n \mathbf{w}^T \mathbf{x}_n}}{1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}} \tag{25}$$

$$= \frac{1}{N}\sum_{n=0}^{N} -y_n \mathbf{x}_n \sigma(-y_n \mathbf{w}^T \mathbf{x}_n) \tag{26}$$

Para explicar un poco el proceso, en (25) se ha aplicado la derivada del logaritmo neperiano $\ln(f(x))$, que es $\frac{1}{f(x)} f'(x)$. En (26) se ha aplicado la función sigmoide sobre $\frac{e^{-y_n \mathbf{w}^T \mathbf{x}_n}}{1+e^{-y_n \mathbf{w}^T \mathbf{x}_n}}$, sabiendo que $\sigma(x) = \frac{e^x}{1+e^x}$.

Una vez dicho esto, vamos a razonar sobre si un ejemplo mal clasificado contribuye más al gradiente que un ejemplo bien clasificado.

Si un ejemplo está bien clasificado, el signo dentro del sigmoide será negativo, con lo cuál el valor de la función sigmoide será pequeño, pudiendo ser incluso bastante próximo a 0. Este valor, aún multiplicado por lo que hay fuera de la función sigmoide, se seguirá manteniendo pequeño, y por tanto, su contribución al gradiente será muy pequeña o casi despreciable. En cambio, si un ejemplo no está bien clasificado, el signo dentro del sigmoide es positivo (se cancela el signo negativo con uno de los valores negativos, que pueden ser o bien el valor real o bien el valor predicho), y por tanto, el valor que devolverá el sigmoide será grande, incluso puede ser que sea próximo a 1. Este valor, multiplicado por lo que hay fuera de la función sigmoide se seguirá manteniendo grande, y por tanto, contriubuirá mucho al gradiente, mucho más de lo que lo haría un ejemplo bien clasificado.

Ejercicio 10

Definamos el error en un punto $(\mathbf{x}_n, y_n)$ por

$$\mathbf{e}_n(\mathbf{w}) = \max(0,\ -y_n \mathbf{w}^T \mathbf{x}_n)$$

Argumentar si con esta función de error el algoritmo PLA puede interpretarse como SGD sobre $\mathbf{e}_n$ con tasa de aprendizaje $\nu = 1$.

Solución

Vamos a partir de la expresión del algoritmo PLA para actualizar $\mathbf{w}$, la cual es la siguiente:

- $\mathbf{w}_{t+1} = \mathbf{w}_t$, si la predicción realizada es correcta. Es decir, los pesos no se ven modificados.
- $\mathbf{w}_{t+1} = \mathbf{w}_t + y\mathbf{x}$, si la predicción realizada es errónea, es decir, si el signo predicho no se corresponde con el signo real. El proceso se ha descrito en ejercicios anteriores. especificados anteriormente.

El objetivo es demostrar que el algoritmo PLA es equivalente al SGD con $\nu = 1$. Para eso, vamos a comenzar estudiando nuestra función de error, para ver como funciona. En la tabla que se muestar a continuación se puede ver qué valores tiene la función max en función de y_n y en función de $\mathbf{w}^T\mathbf{x}_n$:

y_n	$\mathbf{w}^T\mathbf{x}_n$	$\max(0,\ -y_n\mathbf{w}^T\mathbf{x}_n)$
1	1	$\max(0,-1) = 0$
-1	-1	$\max(0,-1) = 0$
-1	1	$\max(0,1) = 1$
1	-1	$\max(0,1) = 1$

Cuadro 1: Valor de la función max en función de las entradas.

De la tabla anterior podemos deducir que, en los casos en los que y_n coincide con $\mathbf{w}^T\mathbf{x}_n$ (cuando la etiqueta real se corresponde con la predicha), el máximo será 0, y en caso contrario será 1. Como nuestro objetivo es minimizar este error (según el criterio *ERM*), es decir, hacer que sea siempre o casi siempre 0 (depende de si los datos son linealmente separables), lo que tenemos que hacer es modificar los valores de $\mathbf{w}$ cada vez que obtengamos un valor superior a 0 con nuestra función de error, y en caso contrario, dejar los valores como están. Y de esto es lo que se encargará precisamente nuestro SGD. Pero antes de entrar a ver como sería, destripemos nuestra función de error para verla como una función por trozos:

$$\mathbf{e}_n(\mathbf{w}) = \begin{cases} 0 & -y_n\mathbf{w}^T\mathbf{x}_n \leq 0 \\ -y_n\mathbf{w}^T\mathbf{x}_n & -y_n\mathbf{w}^T\mathbf{x}_n > 0 \end{cases} \tag{27}$$

Ahora, veamos como sería la expresión del SGD con esta función:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \nu \nabla \mathbf{e}_n(\mathbf{w}) \tag{28}$$

Y por tanto, una vez visto esto, calculemos el gradiente de la función de error, para cada trozo de la función original:

$$\nabla \mathbf{e}_n(\mathbf{w}) = \begin{cases} 0 & -y_n \mathbf{w}^T \mathbf{x}_n \leq 0 \\ -y_n \mathbf{x}_n & -y_n \mathbf{w}^T \mathbf{x}_n > 0 \end{cases} \tag{29}$$

Una vez dicho todo esto, veamos como quedaría la expresión del SGD, sabiendo que $\nu = 1$:

- En el caso de que las etiquetas coincidan, tendremos que:

$$\mathbf{w}_{t+1} = \mathbf{w}_t \tag{30}$$

- En el caso de que las etiquetas no coincidan, tendremos que la expresión del SGD es:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + y\mathbf{x} \tag{31}$$

Estas expresiones obtenidas son, ni más ni menos, que las reglas que teníamos para actualizar el PLA. Con lo cuál, podemos ver que, con esta función de error, y con un $\nu = 1$, el SGD funciona exactamente igual que el algoritmo PLA. Es más, el algoritmo PLA es un caso particular del SGD, en el que $\nu = 1$ y la función de error es $\mathbf{e}_n(\mathbf{w}) = \max(0,\ -y_n \mathbf{w}^T \mathbf{x}_n)$.

Referencias

[1] Wikipedia. *Aprendizaje no supervisado*
https://en.wikipedia.org/wiki/Unsupervised_learning

[2] Wikipedia. *Aprendizaje por refuerzo*
https://en.wikipedia.org/wiki/Reinforcement_learning

[3] Wikipedia. *Proceso de decisión de Markov*
https://en.wikipedia.org/wiki/Markov_decision_process

[4] Descomposición en valores singulares, *Wikipedia*
https://en.wikipedia.org/wiki/Singular_value_decomposition

[5] Perceptron Learning Algorithm: A Graphical Explanation Of Why It Works. *Akshay Chandra Lagandula*. https://towardsdatascience.com/perceptron-learning-algorithm-d5db0deab975

[6] Neural Networks: A Systematic Introduction. *Raúl Rojas*. Capítulo 4, 85-57.
https://page.mi.fu-berlin.de/rojas/neural/neuron.pdf

[7] Wikipedia. *Matriz de proyección*
https://en.wikipedia.org/wiki/Projection_matrix

[8] Juan Vilar. *Teoría de Regresión Lineal*. Sección 8.3.
http://dm.udc.es/asignaturas/estadistica2/sec8_3.html