



Aprendizaje Automático

TRABAJO PRÁCTICO 1

Diseño de Sistemas de Aprendizaje
Aprendizaje de Conceptos

Alumna:

Fontanella De Santis, Teresa Natalia

Legajo 52.455

Responsable Académica:

Dra. VILLAR, Ana Julia

ÍNDICE

	Página
OBJETIVO	3
I. Diseño de Sistemas de Aprendizaje	3
II. Aprendizaje de Conceptos	5
III. Recursos Consultados	11
APÉNDICE	12
Código de las Funciones de los Ejercicios I.3. y II.5.	

OBJETIVO

El presente Trabajo Práctico consiste en aprender a diseñar un sistema de aprendizaje, y a aplicar algoritmos para el aprendizaje de conceptos.

I. Diseño de Sistemas de Aprendizaje

1. Dar un ejemplo de una tarea de aprendizaje.

(a) Describir la tarea, la medida de rendimiento y la experiencia de aprendizaje.

(b) Proponer una función objetivo que deba ser aprendida y su representación.

Tarea (T): Filtrar correo spam.

Medida de rendimiento o performance (P): porcentaje de correos clasificados correctamente.

Experiencia de aprendizaje (E): conjunto de correos con su categoría correspondiente (si son o no spam).

La función objetivo a aprender sería reconocer correo no deseado con una probabilidad dada. Se representaría del siguiente modo:

$$\hat{V}: T \rightarrow \{0,1\} \quad f(\text{texto}) = \begin{cases} 1 & \text{si archivo es spam } (p_n \geq p_0) \\ 0 & \text{sino} \end{cases}$$

donde T es el conjunto de los correos posibles a analizar, $\text{texto} = w_1 w_2 \dots w_n \in T$ (cada w_i es una palabra del diccionario), p_0 es el porcentaje máximo permitido de palabras no deseadas, y p_n se calcula de la siguiente manera:

$$x_i = \begin{cases} 0 + x_{i-1} & \text{si } w_i \text{ es una palabra no deseada} \\ 1 + x_{i-1} & \text{si } w_i \text{ es una palabra deseada} \\ 0 & \text{si } i = 0 \end{cases}$$

$p_n = \frac{x_n}{n}$, siendo n la cantidad total de palabras del texto

2. Considerar estrategias alternativas para el generador de experimentos visto en clase para el juego de damas.

En particular, considerar estrategias en las cuales el generador de experimentos sugiere nuevas posiciones del tablero:

- *generando al azar posiciones del tablero válidas;*

- *generando una posición escogiendo un estado del tablero de un juego previo, y luego aplicando un movimiento que no haya sido ejecutado;*
- *una nueva propuesta de estrategia.*

Una estrategia alternativa para el generador de experimentos, podría ser elegir al azar un estado del tablero que represente cierto momento de la partida (apertura, desarrollo, cierre, etcétera), y generar una posición moviendo la pieza con la que se puede comer más fichas del adversario.

3. Implementar un algoritmo similar al del juego de damas pero para el juego del tatetí.

El tatetí es un juego entre dos jugadores: O y X, que marcan los espacios de un tablero de 3×3 alternadamente. Un jugador gana si consigue tener una línea de tres de sus símbolos: la línea puede ser horizontal, vertical o diagonal.

Por ejemplo, una partida ganada por el primer jugador, X:



Una partida que termina en empate:



(a) *Representar la función aprendida como una combinación lineal de características del tablero.*

(b) *Para entrenar al programa, jugar repetidamente contra una segunda copia del programa, que use una función de evaluación fija.*

Hacer un gráfico del porcentaje de juegos ganados por su sistema, versus el número de juegos de entrenamiento jugados.

La función aprendida puede representarse como sigue a continuación:

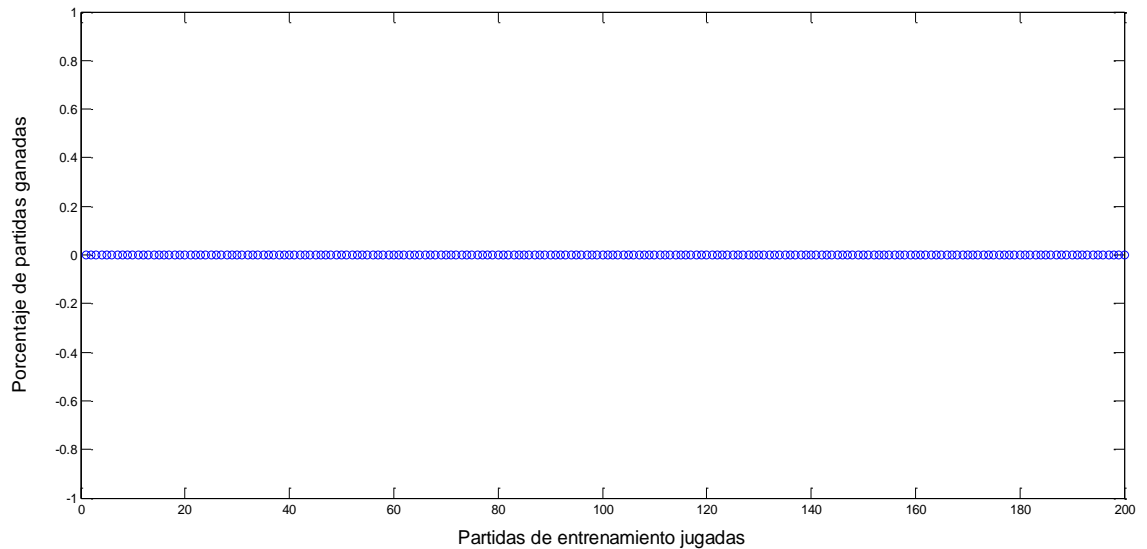
$$\hat{V}(b) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4$$

x_1 : cantidad de casilleros ocupados por el jugador

x_2 : cantidad de casilleros ocupados por el contrincante

x_3 : número de secuencias de dos casilleros propios seguidos de uno libre en el tablero

x_4 : número de secuencias de dos casilleros del oponente seguidos de uno libre en el tablero



II. Aprendizaje de Conceptos

1. *Explicar por qué el tamaño del espacio de hipótesis en la tarea de aprendizaje disfruta-deporte es 973.*

- (a) *¿Cómo aumentaría el número de instancias y de hipótesis posibles si se agrega el atributo corriente (del agua) el cual puede tomar los valores débil, moderada o fuerte?*
- (b) *En general, ¿cómo crece el número de posibles instancias e hipótesis si se agrega un nuevo atributo A que puede tomar k valores posibles?*

El tamaño del espacio de hipótesis en la tarea *disfruta-deporte* es 973, porque

$$973 = 4 \times 3 \times 3 \times 3 \times 3 \times 3 + 1$$

donde 1 es la hipótesis más específica (con todos \emptyset), y los otros valores son las posibilidades de cada atributo incrementadas en 1 (por el símbolo ?).

Si se agrega el atributo *corriente del agua*, el número de instancias y de hipótesis posibles sería

$$4 \times 3 \times 3 \times 3 \times 3 \times 3 \times 4 + 1 = 3889$$

En el caso más general, llamando x_{new} y x_{old} a la cantidad de posibles instancias con o sin el atributo A respectivamente, puede decirse que: $x_{new} = x_{old} \times k$

Y llamando h_{new} y h_{old} a la cantidad de posibles hipótesis con o sin el atributo A respectivamente, se obtiene que: $h_{new} = h_{old} \times (k + 1) - k$

2. Dar la secuencia de conjunto frontera S y G obtenida mediante el algoritmo de ELIMINACIÓN-DE-CANDIDATOS, si la secuencia de ejemplos de entrenamiento se da en orden inverso (4-3-2-1) a los de la siguiente tabla:

Ejemplo	Cielo	Temperatura del aire	Humedad	Viento	Temperatura del agua	Pronóstico del tiempo	Disfruta deporte
1	Soleado	Cálida	Normal	Fuerte	Cálida	Igual	Si
2	Soleado	Cálida	Alta	Fuerte	Cálida	Igual	Si
3	Nublado	Fría	Alta	Fuerte	Cálida	Cambiante	No
4	Soleado	Cálida	Normal	Fuerte	Fría	Cambiante	Si

Primero inicializo S y G

$$S = \{ \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle \} \text{ y } G = \{ \langle ?, ?, ?, ?, ?, ? \rangle \}$$

Utilizo la muestra 4

$$S = \{ \langle soleado, cálida, alta, fuerte, fría, cambiante \rangle \} \text{ y } G = \{ \langle ?, ?, ?, ?, ?, ? \rangle \}$$

Entreno con la muestra 3

$$S = \{ \langle soleado, cálida, alta, fuerte, fría, cambiante \rangle \} \text{ y } G = \{ \langle soleado/lluvioso, ?, ?, ?, ?, ? \rangle, \langle ?, cálida, ?, ?, ?, ? \rangle, \langle ?, ?, normal, ?, ?, ? \rangle, \langle ?, ?, ?, débil, ?, ? \rangle, \langle ?, ?, ?, ?, fría, ? \rangle, \langle ?, ?, ?, ?, igual \rangle \}$$

Utilizo la muestra 2

$$S = \{ \langle soleado, cálida, ?, fuerte, ?, ? \rangle \} \text{ y } G = \{ \langle soleado/lluvioso, ?, ?, ?, ?, ? \rangle, \langle ?, cálida, ?, ?, ?, ? \rangle \}$$

Utilizo la muestra 1

$$S = \{ \langle \text{soleado/lluvioso}, \text{cálida}, ?, \text{fuerte}, ?, ? \rangle \} \text{ y } G = \{ \langle \text{soleado/lluvioso}, ?, ?, ?, ?, ? \rangle, \langle ?, \text{cálida}, ?, ?, ?, ? \rangle \}$$

3. Considerar la tarea de aprendizaje disfruta-deporte y el espacio de hipótesis H , descrito como una conjunción de restricciones sobre los atributos Cielo, Aire, Humedad, Viento, Agua y Pronóstico.

Definamos un nuevo espacio de hipótesis H' que contenga todas las disyunciones de pares de las hipótesis en H . Por ejemplo, una hipótesis en H' es

$$\langle ?, \text{cálida}, \text{alta}, ?, ?, ? \rangle \vee \langle \text{soleado}, ?, \text{alta}, ?, ?, \text{igual} \rangle$$

Aplicar el algoritmo de ELIMINACIÓN-DE-CANDIDATOS para el espacio de hipótesis H' , para la secuencia de entrenamiento de la tabla del ejercicio 2.

Para aplicar el algoritmo de ELIMINACIÓN-DE-CANDIDATOS con los 4 ejemplos de entrenamiento, primero inicializo S y G :

$$S = \{ \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle \} \text{ y } G = \{ \langle ?, ?, ?, ?, ?, ? \rangle \}$$

Utilizo la muestra 4:

$$S = \{ \langle \text{soleado}, \text{cálida}, \text{normal}, \text{fuerte}, \text{fría}, \text{cambiante} \rangle \} \text{ y } G = \{ \langle ?, ?, ?, ?, ?, ? \rangle \}$$

Utilizo la muestra 3:

$$\begin{aligned} S &= \{ \langle \text{soleado}, \text{cálida}, \text{normal}, \text{fuerte}, \text{fría}, \text{cambiante} \rangle \} \text{ y } \\ G &= \{ \langle \text{soleado/lluvioso}, ?, ?, ?, ?, ? \rangle \vee \langle ?, \text{cálida}, ?, ?, ?, ? \rangle, \langle \text{soleado/lluvioso}, ?, ?, ?, ?, ? \rangle \vee \\ &\quad \vee \langle ?, ?, \text{normal}, ?, ?, ? \rangle, \langle \text{soleado/lluvioso}, ?, ?, ?, ?, ? \rangle \vee \\ &\quad \langle ?, ?, ?, \text{débil}, ?, ? \rangle, \langle \text{soleado/lluvioso}, ?, ?, ?, ?, ? \rangle \vee \langle ?, ?, ?, ?, \text{fría}, ? \rangle, \\ &\quad \langle \text{soleado/lluvioso}, ?, ?, ?, ?, ? \rangle \vee \langle ?, ?, ?, ?, \text{igual} \rangle, \langle ?, \text{cálida}, ?, ?, ?, ? \rangle \\ &\quad \vee \langle ?, ?, \text{normal}, ?, ?, ? \rangle, \\ &\quad \langle ?, \text{cálida}, ?, ?, ?, ? \rangle \vee \langle ?, ?, ?, \text{débil}, ?, ? \rangle, \langle ?, \text{cálida}, ?, ?, ?, ? \rangle \vee \\ &\quad \langle ?, ?, ?, ?, \text{fría}, ? \rangle, \langle ?, \text{cálida}, ?, ?, ?, ? \rangle \vee \langle ?, ?, ?, ?, \text{igual} \rangle, \\ &\quad \langle ?, ?, \text{normal}, ?, ?, ? \rangle \vee \langle ?, ?, ?, \text{débil}, ?, ? \rangle, \langle ?, ?, \text{normal}, ?, ?, ? \rangle \vee \\ &\quad \langle ?, ?, ?, ?, \text{fría}, ? \rangle, \langle ?, ?, \text{normal}, ?, ?, ? \rangle \vee \langle ?, ?, ?, ?, \text{igual} \rangle, \\ &\quad \langle ?, ?, ?, \text{débil}, ?, ? \rangle \vee \langle ?, ?, ?, ?, \text{fría}, ? \rangle, \langle ?, ?, ?, \text{débil}, ?, ? \rangle \vee \\ &\quad \langle ?, ?, ?, ?, \text{igual} \rangle, \langle ?, ?, ?, ?, \text{fría}, ? \rangle \vee \langle ?, ?, ?, ?, \text{igual} \rangle \} \end{aligned}$$

Utilizo la muestra 2:

$$S = \{ \langle \text{soleado, cálida, normal, fuerte, fría, cambiante} \rangle \vee \\ \langle \text{soleado, cálida, alta, fuerte, cálida, igual} \rangle \} y$$

$$G = \{ \langle \text{soleado/lluvioso, ?, ?, ?, ?} \rangle \vee \langle \text{?, cálida, ?, ?, ?, ?} \rangle, \langle \text{soleado/lluvioso, ?, ?, ?, ?} \rangle \\ \vee \langle \text{?, ?, normal, ?, ?, ?} \rangle, \langle \text{soleado/lluvioso, ?, ?, ?, ?} \rangle \vee \\ \langle \text{?, ?, ?, débil, ?, ?} \rangle, \langle \text{soleado/lluvioso, ?, ?, ?, ?} \rangle \vee \langle \text{?, ?, ?, ?, fría, ?} \rangle, \\ \langle \text{soleado/lluvioso, ?, ?, ?, ?} \rangle \vee \langle \text{?, ?, ?, ?, igual} \rangle, \langle \text{?, cálida, ?, ?, ?, ?} \rangle \\ \vee \langle \text{?, ?, normal, ?, ?, ?} \rangle, \\ \langle \text{?, cálida, ?, ?, ?, ?} \rangle \vee \langle \text{?, ?, ?, débil, ?, ?} \rangle, \langle \text{?, cálida, ?, ?, ?, ?} \rangle \vee \\ \langle \text{?, ?, ?, ?, fría, ?} \rangle, \langle \text{?, cálida, ?, ?, ?, ?} \rangle \vee \langle \text{?, ?, ?, ?, ?, igual} \rangle, \\ \langle \text{?, ?, normal, ?, ?, ?} \rangle \vee \langle \text{?, ?, ?, ?, ?, igual} \rangle, \langle \text{?, ?, ?, débil, ?, ?} \rangle \vee \\ \langle \text{?, ?, ?, ?, ?, igual} \rangle, \langle \text{?, ?, ?, ?, fría, ?} \rangle \vee \langle \text{?, ?, ?, ?, ?, igual} \rangle \}$$

Utilizo la muestra 1:

$$S = \{ \langle \text{soleado, cálida, normal, fuerte, fría, cambiante} \rangle \vee \\ \langle \text{soleado, cálida, ?, fuerte, cálida, igual} \rangle \} y$$

$$G = \{ \langle \text{soleado/lluvioso, ?, ?, ?, ?} \rangle \vee \langle \text{?, cálida, ?, ?, ?, ?} \rangle, \langle \text{soleado/lluvioso, ?, ?, ?, ?} \rangle \\ \vee \langle \text{?, ?, normal, ?, ?, ?} \rangle, \langle \text{soleado/lluvioso, ?, ?, ?, ?} \rangle \vee \\ \langle \text{?, ?, ?, débil, ?, ?} \rangle, \langle \text{soleado/lluvioso, ?, ?, ?, ?} \rangle \vee \langle \text{?, ?, ?, ?, fría, ?} \rangle, \\ \langle \text{soleado/lluvioso, ?, ?, ?, ?} \rangle \vee \langle \text{?, ?, ?, ?, ?, igual} \rangle, \langle \text{?, cálida, ?, ?, ?, ?} \rangle \\ \vee \langle \text{?, ?, normal, ?, ?, ?} \rangle, \\ \langle \text{?, cálida, ?, ?, ?, ?} \rangle \vee \langle \text{?, ?, ?, débil, ?, ?} \rangle, \langle \text{?, cálida, ?, ?, ?, ?} \rangle \vee \\ \langle \text{?, ?, ?, ?, fría, ?} \rangle, \langle \text{?, cálida, ?, ?, ?, ?} \rangle \vee \langle \text{?, ?, ?, ?, ?, igual} \rangle, \\ \langle \text{?, ?, normal, ?, ?, ?} \rangle \vee \langle \text{?, ?, ?, ?, ?, igual} \rangle, \langle \text{?, ?, ?, débil, ?, ?} \rangle \vee \\ \langle \text{?, ?, ?, ?, ?, igual} \rangle, \langle \text{?, ?, ?, ?, fría, ?} \rangle \vee \langle \text{?, ?, ?, ?, ?, igual} \rangle \}$$

4. Considerar la siguiente secuencia de ejemplos de entrenamiento positivos y negativos, que describen el concepto “pares de personas que viven en la misma casa”.

Cada ejemplo de entrenamiento se describe como un par ordenado de personas, en el cual cada persona es descripta por su sexo, color de cabello (morocha, castaño o rubio), altura (alto, mediano o bajo) y nacionalidad (argentino, italiano, español, francés, alemán, inglés o japonés)
+ <<mujer morocha baja argentina>> <varón castaño alto argentino>>

+ <<mujer morocha baja argentina> <varón castaño bajo francés>>

- <<mujer morocha baja inglesa> <mujer castaña alta alemana>>

+ <<mujer castaña baja española> <varón castaño alto español>>

Considerar un espacio de hipótesis basado en estas instancias, en el cual cada hipótesis se represente con una 4-upla, y donde las restricciones de cada atributo pueden ser un valor específico, "?" o " ϕ ". Por ejemplo, la hipótesis

<<mujer ? ? japonesa> <varón ? alto ?>>

representa a todos los pares de personas en los que la primera persona es una mujer japonesa y la segunda un varón alto.

(a) Aplicar el algoritmo de ELIMINACIÓN-DE-CANDIDATOS para los ejemplos de entrenamiento previos.

(b) ¿Cuántas hipótesis distintas del espacio hipótesis dado son consistentes con el siguiente ejemplo de entrenamiento positivo?

+ <<mujer rubia alta inglesa> <varón morocho bajo italiano>>

(c) Supongamos que el aprendiz ha encontrado sólo un ejemplo positivo en la parte (b), y que se le permite consultar al entrenador, después de haber generado cualquier instancia, para que la clasifique.

i. Dar una secuencia específica de consultas, que asegure al aprendiz que convergerá a una única hipótesis correcta.

ii. Dar la secuencia más corta de consultas que puede encontrar. ¿Cómo se relaciona el largo de esta secuencia con la respuesta a la pregunta (b)?

Para poder aplicar el algoritmo de ELIMINACIÓN-DE-CANDIDATOS con los primeros 4 ejemplos de entrenamiento, primero inicializo S y G :

$$S = \{ \langle \phi, \phi, \phi, \phi, \phi, \phi, \phi, \phi \rangle \} \text{ y } G = \{ \langle ?, ?, ?, ?, ?, ?, ?, ? \rangle \}$$

Utilizo la muestra 1:

$$S = \{ \langle \text{mujer, morocha, baja, argentina, varón, castaño, alto, argentino} \rangle \} \text{ y } G = \{ \langle ?, ?, ?, ?, ?, ?, ?, ? \rangle \}$$

Entreno con la muestra 2:

$$S = \{ \langle \text{mujer, morocha, baja, argentina, varón, castaño, ?, ?} \rangle \} \text{ y } G = \{ \langle ?, ?, ?, ?, ?, ?, ?, ? \rangle \}$$

Utilizo la muestra 3:

$$S = \{ \langle \text{mujer, morocha, baja, argentina, varón, castaño, ?} \rangle \} \text{ y } G = \{$$

$$\langle \text{varón, ?, ?, ?, ?, ?, ?} \rangle, \langle \text{?, rubia, ?, ?, ?, ?, ?} \rangle, \langle \text{?, castaña, ?, ?, ?, ?, ?} \rangle,$$

$$\langle \text{?, ?, alta, ?, ?, ?, ?} \rangle, \langle \text{?, ?, mediana, ?, ?, ?, ?} \rangle,$$

$$\langle \text{?, ?, ?, japonesa, ?, ?, ?, ?} \rangle, \langle \text{?, ?, ?, argentina, ?, ?, ?, ?} \rangle,$$

$$\langle \text{?, ?, ?, italiana, ?, ?, ?, ?} \rangle, \langle \text{?, ?, ?, alemana, ?, ?, ?, ?} \rangle,$$

$$\langle \text{?, ?, ?, francesa, ?, ?, ?, ?} \rangle, \langle \text{?, ?, ?, española, ?, ?, ?, ?} \rangle,$$

$$\langle \text{?, ?, ?, ?, varón, ?, ?, ?} \rangle, \langle \text{?, ?, ?, ?, morocho, ?, ?} \rangle, \langle \text{?, ?, ?, ?, rubio, ?, ?} \rangle,$$

$$\langle \text{?, ?, ?, ?, ?, bajo, ?} \rangle, \langle \text{?, ?, ?, ?, ?, mediano, ?} \rangle,$$

$$\langle \text{?, ?, ?, ?, ?, japonés} \rangle, \langle \text{?, ?, ?, ?, ?, argentino} \rangle,$$

$$\langle \text{?, ?, ?, ?, ?, italiano} \rangle, \langle \text{?, ?, ?, ?, ?, español} \rangle,$$

$$\langle \text{?, ?, ?, ?, ?, francés} \rangle, \langle \text{?, ?, ?, ?, ?, inglés} \rangle \}$$

Utilizo la muestra 4:

$$S = \{ \langle \text{mujer, ?, baja, ?, varón, castaño, ?} \rangle \} \text{ y }$$

$$G = \{ \langle \text{?, castaña, ?, ?, ?, ?} \rangle, \langle \text{?, ?, ?, española, ?, ?, ?, ?} \rangle, \langle \text{?, ?, ?, ?, varón, ?, ?, ?} \rangle,$$

$$\langle \text{?, ?, ?, ?, ?, español} \rangle \}$$

Las hipótesis que pertenecen al espacio de versión son:

$\{ \langle \text{mujer, ?, baja, ?, varón, castaño, ?} \rangle, \langle \text{mujer, ?, baja, ?, varón, ?, ?, ?} \rangle, \langle$
 $\text{mujer, ?, ?, ?, varón, castaño, ?} \rangle, \langle \text{mujer, ?, baja, ?, ?, castaño, ?} \rangle, \langle$
 $\text{?, ?, baja, ?, varón, castaño, ?} \rangle, \langle \text{mujer, ?, baja, ?, ?, ?, ?} \rangle, \langle$
 $\text{mujer, ?, ?, ?, varón, ?, ?, ?} \rangle, \langle \text{mujer, ?, ?, ?, ?, castaño, ?} \rangle,$
 $\langle \text{?, ?, baja, ?, varón, ?, ?, ?} \rangle,$
 $\langle \text{?, ?, baja, ?, ?, castaño, ?} \rangle, \langle \text{?, ?, ?, ?, varón, castaño, ?} \rangle, \langle$
 $\text{?, castaña, ?, ?, ?, ?} \rangle, \langle \text{?, ?, ?, española, ?, ?, ?, ?} \rangle, \langle \text{?, ?, ?, ?, varón, ?, ?, ?} \rangle, \langle$
 $\text{?, ?, ?, ?, ?, español} \rangle \}$. Es por esto que, la cantidad de hipótesis que son consistentes
 con la muestra $\langle \langle \text{mujer rubia alta inglesa} \rangle \langle \text{varón morocho bajo italiano} \rangle \rangle$ es 2.

Una secuencia de consultas posible, para que haya una única hipótesis correcta para el punto b) sería: + $\langle \langle \text{mujer morocha alta alemana} \rangle \langle \text{varón rubio bajo francés} \rangle \rangle$. Ésta es una de las secuencias más cortas de consultas que se puede encontrar, y es la mitad de las hipótesis consistentes con la muestra del inciso b). Generalmente, una estrategia óptima de consulta es generar instancias que satisfagan la mitad de las hipótesis del espacio-de-versión correcto.

5. Implementar el algoritmo ENCONTRAR-S.

- (a) Verificar qué produce la traza vista en clase para el ejemplo disfruta-deporte.*
- (b) Usar el programa para estudiar el número de ejemplos de entrenamiento al azar requeridos para aprender exactamente el concepto objetivo, y luego clasificarlos de acuerdo al siguiente concepto objetivo <soleado, cálida,?,?,?,?>.*
- (c) Aplicar su programa de entrenamiento sobre ejemplos generados al azar, y medir el número de ejemplos necesarios hasta que la hipótesis del programa sea idéntica a la del concepto objetivo.*
 - i. ¿Podría predecir el número promedio de ejemplos necesarios? Realizar el experimento 20 veces y reportar el número medio de experimentos necesarios.*
 - ii. ¿Cómo espera que varíe este número con respecto al número de “?” en el concepto objetivo?*
 - iii. ¿Cómo variaría este número en función del número de atributos usados para describir las instancias y las hipótesis?*

Haciendo cálculos teóricos, el número promedio de ejemplos necesarios podría ser

$$2 \times 2 \times 2 \times 2 = 16$$

Efectuando el experimento varias veces (ejecutando la función ejercicio5), se tiene que el rango de cantidad mínima promedio de muestras necesarias se encuentra entre 11 y 15.

Este número debería aumentar a medida que la función objetivo es más general (es decir, hay más “?”), y disminuir cuando ésta sea más específica. Asimismo, debería ser proporcional (o aproximado) a 2^x , siendo x el grado de generalización de la función objetivo (o la cantidad de atributos con “?”).

III. Recursos Consultados

- *Material aportado por la Cátedra.*
- *Consultas a la Cátedra.*

APÉNDICE

Código de las Funciones de los Ejercicios I.3. y II.5.

Ejercicio 5 de la Parte II.

```
function ejercicio5()
    varQty=6;
    %ejercicio 5.a
    trainingSet=[1,1,1,1,1,1,1 ; 1,1,2,1,1,1,1 ; 2,2,2,1,1,2, -1 ; 1,1,2,1,2,2,1];
    resultado=DISFRUTADEPORTE(trainingSet);
    disp(resultado);

    %ejercicio 5.b
    description=[3,2,2,2,2,2];
    funcionObjetivo=[1,1,0,0,0,0];
    trainingQtyAverage=0;
    for j=1:20
        trainingQty=4;
        rightnessPercentage=0;
        while(rightnessPercentage==0 && trainingQty<prod(description))
            trainingQty=trainingQty+1;
            trainingSet=zeros(trainingQty,varQty+1);
            trainingSetChosen=zeros(1,trainingQty);
            % se crea conjunto de entrenamiento aleatorio
            for i=1:trainingQty
                auxi=[randi(3,1,1) randi(2,1,5)];
                trainKey=generateKey(auxi);
                while(sum(trainingSetChosen==trainKey)~=0)
                    auxi=[randi(3,1,1) randi(2,1,5)];
                    trainKey=generateKey(auxi);
                end
                if(auxi(1)==funcionObjetivo(1) && auxi(2)==funcionObjetivo(2))
                    result=1;
                else
                    result=(-1);
                end
                trainingSet(i,:)=[auxi result];
            end
        end
    end
```

```

        trainingSetChosen(i)=trainKey;
    end
    %      disp('trainingSet');
    %      disp(trainingSet);
    resultado=ENCONTRARS(description,trainingSet);
    %      disp(resultado);
    equals=true;
    for k=1:varQty
        if(funcionObjetivo(1,k)~=resultado(1,k))
            equals=false;
        end
    end
    if(equals==true)
        rightnessPercentage=rightnessPercentage+1;
    end
end
if(rightnessPercentage~=0)
    disp('Cantidad minima de muestras necesarias: ');
    disp(trainingQty);
    trainingQtyAverage=trainingQtyAverage+trainingQty;
end
end
trainingQtyAverage=trainingQtyAverage/20;
disp('Promedio de cantidad de muestras necesarias: ');
disp(trainingQtyAverage);
end

```

%genera una clave única para cada muestra de entrenamiento (se usa para
 %evitar que haya inconsistencias)

```

function number=generateKey(trainExample)
    primes=[2,3,5,7,11,13];
    primesQty=6;
    number=0;
    for i=1:primesQty
        number=number+(primes(i))*(trainExample(i));
    end
end
end

```

%Utiliza la función ENCONTRARS para resolver el problema DISFRUTADEPORTE, y
 %mostrar la hipótesis resultado de manera legible

```
function hipotesisSolution= DISFRUTADEPORTE(trainingSet)
    description=[3,2,2,2,2,2];
    nullVal=-1;
    generalVal=0;
    hipotesis=ENCONTRARS(description,trainingSet);
    textCell=cell(3,6);
    %descripcion de cielo
    textCell(1,1)=java.lang.String('soleado');
    textCell(2,1)=java.lang.String('nublado');
    textCell(3,1)=java.lang.String('lluvioso');
    %descripcion del aire
    textCell(1,2)=java.lang.String('cálida');
    textCell(2,2)=java.lang.String('fría');
    %descripcion de humedad
    textCell(1,3)=java.lang.String('normal');
    textCell(2,3)=java.lang.String('alta');
    %descripcion de viento
    textCell(1,4)=java.lang.String('fuerte');
    textCell(2,4)=java.lang.String('débil');
    %descripcion de agua
    textCell(1,5)=java.lang.String('cálida');
    textCell(2,5)=java.lang.String('fría');
    %descripcion de pronostico
    textCell(1,6)=java.lang.String('igual');
    textCell(2,6)=java.lang.String('cambiante');
    [r, variablesQty]=size(hipotesis);
    hipotesisSolution=cell(1,variablesQty);
    for i=1: variablesQty
        if(hipotesis(i)==nullVal)
            hipotesisSolution(1,i)=java.lang.String('0');
        elseif (hipotesis(i)==generalVal)
            hipotesisSolution(i)=java.lang.String('?');
        else
            hipotesisSolution(1,i)=textCell(hipotesis(i),i);
        end
    end
end
```

end

```
%description es un array en donde para cada elemento se especifica la
%cantidad de posibles valores que pueden tener (puede ir de 1 hasta
%infinito positivo)
%trainingSet es el conjunto de entrenamiento a utilizar
%Retorna un array con los valores de cada atributo, -1 si es nulo y 0 si es "?"
function hipotesis = ENCONTRARS( description,trainingSet )
    [rows, varQty]=size(description);
    [trainQty, result]=size(trainingSet);
    nullVal=-1;
    generalVal=0;
    hipotesis=repmat(nullVal,1,varQty);
    for i=1:trainQty
        if(trainingSet(i,result)>0)
            for j=1:varQty
                if(hipotesis(j)~=generalVal && trainingSet(i,j)~=hipotesis(j))
                    if(hipotesis(j)== nullVal)
                        hipotesis(j)=trainingSet(i,j);
                    else
                        hipotesis(j)=generalVal;
                    end
                end
            end
        end
    end
end
end
end
```

Ejercicio 3 de la Parte I.

%Sistema de aprendizaje. Ejecutar esta función para el ejercicio 3.

```
function learningSystem()
    maxIterations=200;
    weights=rand(1,5);
    player=1; %player es el aprendiz
    results=zeros(1,maxIterations);
    for i=1:maxIterations
```

```

board=experimentGenerator(weights);
[trainingBoards, result]=systemPerformance(board,weights);
[boardsQty, boardsCol]=size(trainingBoards);
trainingMatrix=[]; %almacena estados de un juego con su respectivo puntaje
for j=1:boardsQty
    trainingMatrix=trainingPair(trainingMatrix,trainingBoards{j,boardsCol},playe
    r,weights);
end
weights=LMSAlgoritm(trainingMatrix,weights);
if(i==1)
    results(i)=result;
else
    results(i)=(results(i-1)*(i-1)+result)/i;
end
end
plot([1:maxIterations],results,'o');
end

function board=experimentGenerator(weights)
    board= zeros(3,3);
end

function [trainingBoards, result]=systemPerformance(board,weights)
    trainingBoards={};
    i=0;
    result=0;

    while(i<9 && result==0)
        if( mod(i,2)==0)
            player=1;
        else
            player= -1;
        end
        [row,col]=smartPosition(board,player,weights);
        board=moveTicTacToe(board,row,col,player);
        i=i+1;
        disp(i);
        showBoard(board);
    end
end

```



```

    [trainRows, trainCols]=size(trainingBoards);
    trainingBoards(trainRows+1,1)={board};
    result=hasWinnerTicTacToe(board);
end
if(result==1)
    disp('X es el ganador');
end
if(result==(-1))
    disp('O es el ganador');
end
if(result==0)
    disp('Empate');
end
end
end

function [row, col]= smartPosition(board,player,weights)
[rows, cols]=size(board);
if(player==-1)
    [row, col]=pcMove(board,player);
    return;
else
    [row, col]=robotMove(board,player,weights);
end
end

function showBoard(board)
    for i=1:3
        disp(board(i,:));
    end
end

%asigna a cada muestra un score
% si gana es +100
% si pierde es -100
% si empata es 0
% sino, se da lo del sucesor y todo eso
function trainingMatrix=trainingPair(trainingMatrix, board,player,weights)
stats=calculateStats(board,player);

```

```
[rowsTraining, colsTraining]=size(trainingMatrix);
enemy=player*(-1);
score=0;
if(sum(sum(board==player))>0)
    score=100;
elseif(sum(sum(board==enemy))>0)
    score=(-100);
elseif(stats(1,1)+stats(1,2)==9)
    score=0;
else
    [rowEnemy colEnemy]=pcMove(board,enemy);
    boardAuxi=moveTicTacToe(board,rowEnemy,colEnemy,enemy);
    stats=calculateStats(boardAuxi,player);
    score=0;
    if (sum(sum(board==player))>0)
        score=100;
    elseif(sum(sum(board==enemy))>0)

        score=(-100);
    elseif(stats(1,1)+stats(1,2)==9)
        score=0;
    else
        score=weights(1,1)+weights(1,2:end)*(stats');
    end
end
trainingMatrix(rowsTraining+1,:)=[stats score];
end

function weights=LMSAlgoritm(trainingMatrix,weights)
    n=0.1; %factor de aprendizaje
    [trainingQty, colTrain]=size(trainingMatrix);
    for i=1:trainingQty
        vn=weights(1,1)+weights(1,2:end)*(trainingMatrix(i,1:4)');
        weights(1,:)=weights(1,:)+n*(trainingMatrix(i,end)-vn)*[1
            trainingMatrix(i,1:4)];
    end
end
```

```
function [row, col]=robotMove(board,player,weights)
[rows, cols]=size(board);
row= -1;
col= -1;
maxScore= -1;
for i=1:rows
    for j=1:cols
        if(board(i,j)==0)
            board=moveTicTacToe(board,i,j,player);
            actualScore=(weights(2:5)')*calculateStats(board,player);
            board(i,j)=0; %deshago el movimiento hecho
            if(row== -1 && col== -1)
                row=i;
                col=j;
                maxScore=actualScore;
            else
                if(maxScore<actualScore)
                    row=i;
                    col=j;
                    maxScore=actualScore;
                end
            end
        end
    end
end
end
end

end
```

%funcion de evaluacion fija (contrincante del robot que esta aprendiendo)

```
function [row, col]=pcMove(board,player)
[rows, cols]=size(board);
row= -1;
col= -1;
maxScore= -1;
for i=1:rows
    for j=1:cols
        if(board(i,j)==0)
            actualScore=fixedEvaluation(board,i,j,player);
```

```

        if(row== -1 && col== -1)
            row=i;
            col=j;
            maxScore=actualScore;
        else
            if(maxScore<actualScore)
                row=i;
                col=j;
                maxScore=actualScore;
            end
        end
    end
end
end
end

function board=moveTicTacToe(board,row,col,player)
    board(row,col)=player;
end

function answer=hasWinnerTicTacToe(board )
    [rows, cols]=size(board);
    for i=1:rows
        auxi=sum(board(i,:));
        if(auxi==3)
            answer=1;
            return;
        end
        if(auxi==(-3))
            answer=(-1);
            return;
        end
    end

    for i=1:cols
        auxi=sum(board(:,i));
        if(auxi==3)

```

```
        answer=1;
        return;
    end
    if(auxi==(-3))
        answer=(-1);
        return;
    end
end
if(board(1,1)==board(2,2) && board(2,2)==board(3,3) && board(3,3)~=0)
    answer=board(1,1);
    return;
end
if(board(1,3)==board(2,2) && board(2,2)==board(3,1)&& board(3,1)~=0)
    answer=board(1,3);
    return;
end
answer=0;
end
```