

# Aprendizaje Automatico - Trabajo Practico 1

Gonzalo Castiglione - 49138

April 17, 2012

**Objetivo: Aprender a diseñar un sistema de aprendizaje y a aplicar los algoritmos para el aprendizaje de conceptos.**

## 1 Diseño de sistemas de aprendizaje

1. Tarea de aprendizaje: Aprender a jugar al 4 en línea
  - (a) Tarea: colocar 4 fichas en línea antes que el oponente.  
Medida de rendimiento: Cantidad de partidos ganados del total de partidos jugados.  
Experiencia de aprendizaje:
    - Cada vez que se logra bloquear una línea del oponente.
    - Cada vez que se logra alinear 4 fichas.
  - (b) Función Objetivo:
    - Si  $b$  es un estado final de *gana*  $\Rightarrow V(b) = 100$ ;
    - Si  $b$  es un estado final de *pierde*  $\Rightarrow V(b) = -100$ ;
    - Si  $b$  es un estado final de *empate*  $\Rightarrow V(b) = 5$ ;
    - Sino  $\Rightarrow V(b) = V(b')$  donde  $b'$  es el mejor estado del tablero que puede alcanzarse que esta a  $n$  movimientos de  $b$ .
2. Se podrian tomar tableros a partir de registros de jugadas realizadas entre personas.
3. Para probar la implementación del algoritmo se deberá seguir los siguientes pasos:
  - Descargar el archivo *tictactoe.jar*
  - Abrir una terminal y ejecutar: `java -jar tictactoe.jar [argumentos]`
  - Si no se envian argumentos al programa, comenzará una partida con un oponente sin entrenamiento (aprende en forma online, es decir, a medida que se juega contra el, mejor se vuelve)

- Si se manda la palabra *trained* seguida de un numero (ej: *trained 2000*), se creará un nuevo oponente, se lo hará jugar la cantidad de veces especificada contra otro oponente que coloque fichas al azar) y luego se habilitará la partida contra este nuevo jugador recién entrenado.
  - Ejemplo de invocación al programa: *java -jar tictactoe.jar trained 1500*
- (a) Inicialmente se implementó una función en la que se consideró los aspectos mas visibles a simple vista del juego:
- $x_1$ : Cantidad de fichas propias en el tablero.
  - $x_2$ : Cantidad de fichas del oponente en el tablero.
  - $x_3$ : Cantidad de fichas propias en línea.
  - $x_4$ : Cantidad de fichas del oponente en línea.
  - $x_5$ : Cantidad de líneas de longitud 2 de fichas del propias en las filas, columnas y diagonales.
  - $x_6$ : Cantidad de líneas de longitud 2 de fichas del oponente en las filas, columnas y diagonales.

Esta implementación funcionaba, pero no resultaba muy difícil ganarle al aprendiz (entrenado), por lo que se optó por una implementación mas "defensiva", en la que se da mas importancia a bloquear los 3 en línea del oponente a tratar de completar 3 fichas propias en línea. Proponiendose la siguiente implementación:

- $x_1$  = Cantidad de filas propias completadas.
- $x_2$  = Cantidad líneas *bloqueadas* del oponente.
- $x_3$  = 1 si se tiene el centro, 0 si no.

Esta última implementación dio resultados mas interesantes. Por ejemplo, si se presenta un tablero en el que el aprendiz puede bloquear o ganar, este termina optando por bloquear en vez de colocar la ficha ganadora. A simple vista parecería no ser una opción viable, pero tiene la ventaja que ya no resulta simple ganarle.

Para el entrenamiento del aprendiz, se implementó el algoritmo *LMS* dado en clase para que funcione en forma *online*, es decir, cada vez que se juega una partida, se asigna un puntaje a cada partido (a partir de la *función objetivo* mencionada) y utilizando la fórmula  $V_{train}(b) = \hat{V}(SUC(b))$  se crea el conjunto de entrenamiento y ajustan los  $w_i$  actuales.

- A continuación se presentan tablas de los resultados obtenidos a partir del *aprendiz vs jugador random* en modo de entrenamiento online:

Juego	Ganados	Empatados	Perdidos	Porcentaje de perdidos
1-2	1	1	0	0%
3-4	1	0	1	50%
4-6	1	1	0	0%
7-8	2	0	0	0%
9-10	1	0	1	50%
11-12	2	0	0	0%

Table 1: Total de resultados de las primeras partidas

Se presentan los resultados de a pares para simplicidad de la tabla ya que los  $w_i$  mas estables se alcanzan a partir de las 10 partidas aproximadamente. A partir de esto, la cantidad de partidos perdidos es aproximadamente del 7 al 15% del total de partidas jugadas.

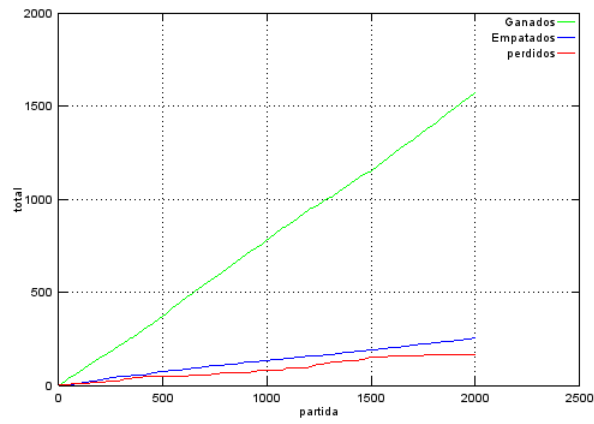


Table 2: Resultados Obtenidos en base a 2000 partidas

## 2 Aprendizaje de conceptos

1. Es 973 dado que por cada posible valor de cada campo de la hipótesis, se tiene el valor ? para representar que en ese lugar, cualquier valor es aceptado y el valor  $\emptyset$  para indicar que ningún valor es aceptado, por lo que el total de posibles instancias estaría dado por  $(x_1 + 2)(x_2 + 2) \dots (x_n + 2)$ . Sin embargo, sintácticamente resultan iguales todas aquellas instancias que contengan el valor  $\emptyset$ , por lo que se deben contar como si fuesen una sola, quedando así el total de instancias para *disfruta deporte* por:

$$(x_1 + 1)(x_2 + 1) \dots (x_n + 1) + 1 = 4 * 3 * 3 * 3 * 3 * 3 + 1 = 973$$

- (a) El nuevo conjunto de instancias X quedaría dado por:

- cielo  $\implies$  soleado, nublado, lluvioso
- aire  $\implies$  cálida, fría
- humedad  $\implies$  normal, alta
- viento  $\implies$  fuerte, débil
- agua  $\implies$  cálida, fría
- pronóstico  $\implies$  igual, cambiante
- corriente (del agua)  $\implies$  débil, moderada, fuerte

El nuevo tamaño de hipótesis *semanticamente* diferente es  $= (4 * 3 * 3 * 3 * 3 * 3) * 4 + 1$ . Es decir, casi se *quatriplicaría* el espacio de hipótesis.

- (b) Cada atributo  $A$  con  $k$  valores diferentes multiplica a la cantidad de hipótesis actuales en  $k + 1$  veces.

- $|V| = (|V_{ant}| - 1) * (k + 1) + 1$

2. Pasos:

$$S_0 = \{(\phi, \phi, \phi, \phi, \phi, \phi)\}$$

$$G_0 = \{(? , ? , ? , ? , ? , ?)\}$$

--><Soleada, Cálida, Alta, Fuerte, Fría, Cambiante> - Yes

$$S_1 = \{(Soleada, Cálida, Alta, Fuerte, Fría, Cambiante)\}$$

$$G_1 = \{(? , ? , ? , ? , ? , ?)\}$$

--><Nueblado, Fría, Alta, Fuerte, Calido, Cambiante> - No

$$S_2 = \{(Soleada, Cálida, Alta, Fuerte, Fría, Cambiante)\}$$

$$G_2 = \{(Soleada, ? , ? , ? , ? , ?), (? , Calida, ? , ? , ? , ?), (? , ? , ? , ? , Fría, ?)\}$$

--><Soleada, Cálida, Alta, Fuerte, Cálida, Igual> - Yes

$$S_3 = \{(Soleada, Cálida, Alta, Fuerte, ? , ?)\}$$

$$G_3 = \{(Soleada, ? , ? , ? , ? , ?), (? , Cálida, ? , ? , ? , ?)\}$$

--><Soleada, Cálida, Noarmal, Fuerte, Cálida, Igual> - Yes

$$S_4 = \{(Soleada, Cálida, ? , Fuerte, ? , ?)\}$$

$$G_4 = \{(Soleada, ? , ? , ? , ? , ?), <? , Cálida, ? , ? , ? , ?>\}$$

### 3. Pasos

$$S_0 = \{(\phi, \phi, \phi, \phi, \phi) \vee (\phi, \phi, \phi, \phi, \phi)\}$$

$$G_0 = \{(\text{?, ?, ?, ?, ?}) \vee (\text{?, ?, ?, ?, ?})\}$$

--><Soleado, Cálido, Normal, Fuerte, Cálido, Iguál> - Yes

$$S_1 = \{(\text{Soleado, Cálido, Normal, Fuerte, Cálido, Iguál}) \vee (\phi, \phi, \phi, \phi, \phi)\}$$

$$G_1 = \{(\text{?, ?, ?, ?, ?}) \vee (\text{?, ?, ?, ?, ?})\}$$

--><Soleado, Cálido, Alto, Fuerte, Cálido, Iguál> - Yes

$$S_2 = \{(\text{Soleado, Cálido, Normal, Fuerte, Cálido, Iguál}) \vee (\text{Soleado, Cálido, Alto, Fuerte, Cálido, Iguál}), (\text{Soleado, Cálido, ?, Fuerte, Cálido, Iguál}) \vee (\phi, \phi, \phi, \phi, \phi)\}$$

$$G_2 = \{(\text{?, ?, ?, ?, ?}) \vee (\text{?, ?, ?, ?, ?})\}$$

--><Rainy, Frío, Fuerte, Fuerte, Cálido, Change> - No

$$S_3 = \{(\text{Soleado, Cálido, Normal, Fuerte, Cálido, Iguál}) \vee (\text{Soleado, Cálido, Alto, Fuerte, Cálido, Iguál}), (\text{Soleado, Cálido, ?, Fuerte, Cálido, Iguál}) \vee (\phi, \phi, \phi, \phi, \phi)\}$$

$$G_3 = \{(\text{Soleado, ?, ?, ?, ?}) \vee (\text{?, Cálido, ?, ?, ?}), (\text{Soleado, ?, ?, ?, ?}) \vee (\text{?, ?, ?, Iguál}), (\text{?, Cálido, ?, ?, ?}) \vee (\text{?, ?, ?, ?, Iguál})\}$$

--><Soleado, Cálido, Fuerte, Fuerte, Fría, Cambiante> - Yes

$$S_4 = \{(\text{Soleado, Cálido, ?, Fuerte, Cálido, Iguál}) \vee (\text{Soleado, Calido, Alto, Fría, Cambiante})\}$$

$$G_4 = \{(\text{Soleado, ?, ?, ?, ?}) \vee (\text{?, Cálido, ?, ?, ?}), (\text{Soleado, ?, ?, ?, ?}) \vee (\text{?, ?, ?, Iguál}), (\text{?, Cálido, ?, ?, ?}) \vee (\text{?, ?, ?, ?, Iguál})\}$$

### 4. Opciones:

- Sexo: Mujer, Varon
- Color de cabello: morocho, castaño o rubio
- Altura: alto, mediano o bajo
- Nacionalidad: argentino, italiano, español, francés, alemán, inglés o japonés

#### (a) Pasos:

$$S_0 = \{<<0, 0, 0, 0> <0, 0, 0, 0>>\}$$

$$G_0 = \{<<?, ?, ?, ?> <?, ?, ?, ?>>\}$$

$$<<\text{mujer morocha baja argentina}> <\text{varón castaño alto argentino}>> = 1$$

$$S_1 = \{<<\text{mujer morocha baja argentina}> <\text{varón castaño alto argentino}>>\}$$

$$G_1 = \{<<?, ?, ?, ?> <?, ?, ?, ?>>\}$$

$$<<\text{mujer morocha baja argentina}> <\text{varón castaño bajo francés}>> = 1$$

$$S_2 = \{<<\text{mujer morocha baja argentina}> <\text{varón castaño ? ?}>>\}$$

$$G_2 = \{<<?, ?, ?, ?> <?, ?, ?, ?>>\}$$

$\langle\langle \text{mujer morocha baja inglesa} \rangle \langle \text{mujer castaña alta alemana} \rangle\rangle = 0$

$S_3 = \{ \langle\langle \text{mujer morocha baja ?} \rangle \langle \text{varon castaño ? ?} \rangle\rangle \}$

$G_3 = \{ \langle\langle ? ? ? ? \rangle \langle \text{varon ? ? ?} \rangle\rangle, \langle\langle ? ? ? \text{ argentina} \rangle \langle ? ? ? ? \rangle\rangle \}$

$\langle\langle \text{mujer castaña baja española} \rangle \langle \text{varón castaño alto español} \rangle\rangle = 1$

$S_4 = \{ \langle\langle \text{mujer ? baja ?} \rangle \langle \text{varon castaño ? ?} \rangle\rangle \}$

$G_4 = \{ \langle\langle ? ? ? ? \rangle \langle \text{varon ? ? ?} \rangle\rangle \}$

(b) Cada hipótesis consistente con el ejemplo dado, puede contener o bien un ? o el valor que tenga especificado por el ejemplo. Como se tienen 2 valores por atributo, daría un total de  $2^8 = 256$  hipótesis.

(c)

Sea  $h$  = conjunto mas general posible consistente con el ejemplo.

Para cada atributo restricción  $a_i$  en  $h$

Pregunto al entrenador si el elemento  $a_i >_g$  al atributo  $a_i$  de la hipótesis correcta. Si lo es, pruebo remover un atributo y repito hasta que deje de ser mas general

5. .

(a) El algoritmo esta implmentado en el ejecutable *finds.jar*. Para ejecutarlo simplemente ir a la línea de comandos y ejecutar *java -jar finds.jar*. Una vez realizado esto, se mostrara el conjunto que se usará para el entrenamiento y la hipótesis actual luego de aplicado cada caso del entrenamiento. Se puede observar que el conjunto final obtenido es el mismo que el visto en clase.

(b) (c)

i. Luego de corridas 20 simulaciones, se midió que en promedio se necesitan unos 27 ejemplos aletorios de aprendizaje para alcanzar la hipótesis objetivo (Soleado, Cálida, ?, ?, ?, ?).

En la gráfico a continuación se muestra la cantidad de ejemplos (aleatorios) que debieron presentarse al algoritmo *Find-S* para que alcance la función objetivo.

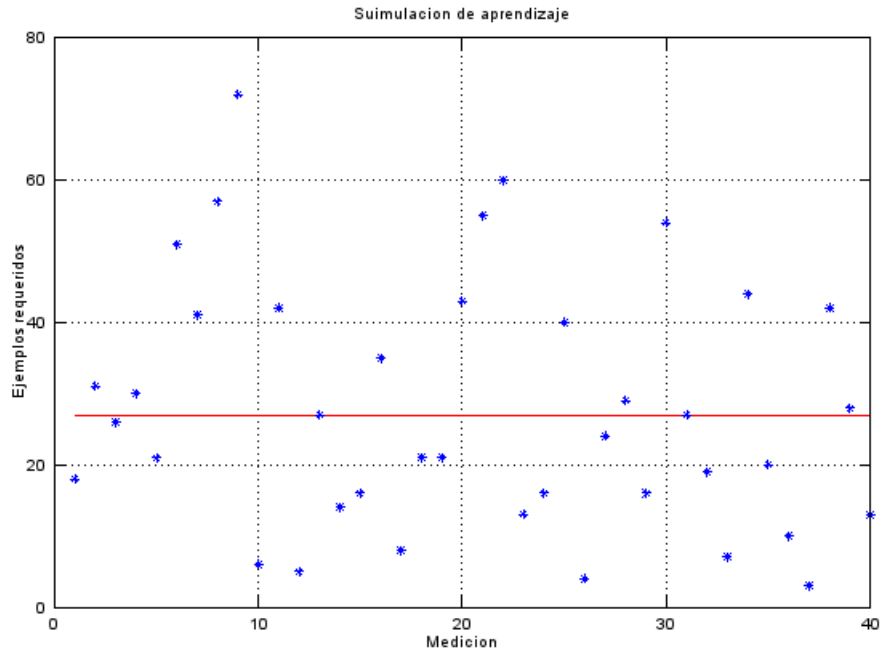


Figure 1: Cantidad media de ejemplos necesarios para alcanzar la hipótesis objetivo

- ii. A medida que se agregan mas "?", se tiene una hipótesis mas genérica, es decir, se pierde especificidad, por lo que encontrarla debería resultar mas sencillo. Si ponemos a modo de ejemplo, el caso mas extremo, que sería (?, ?, ?, ?, ?, ?), cualquier hipótesis que tomemos va a acercarnos un paso mas hacia ella, ya que sería siempre validada (por supuesto, siempre y cuando esta no halla sido presentada anteriormente).
- iii. A continuación se presenta un gráfico en donde se realizaron 100 simulaciones para tratar de alcanzar cada hipótesis. Se comenzó con un hipótesis inicial toda inicializada en "?", y para cada paso de la simulación, uno de estos simbolos era reemplazado por algún valor específico. Obteniednose así un gráfico de la cantidad promedio de ejemplos necesarios vs cantidad de valores "?".

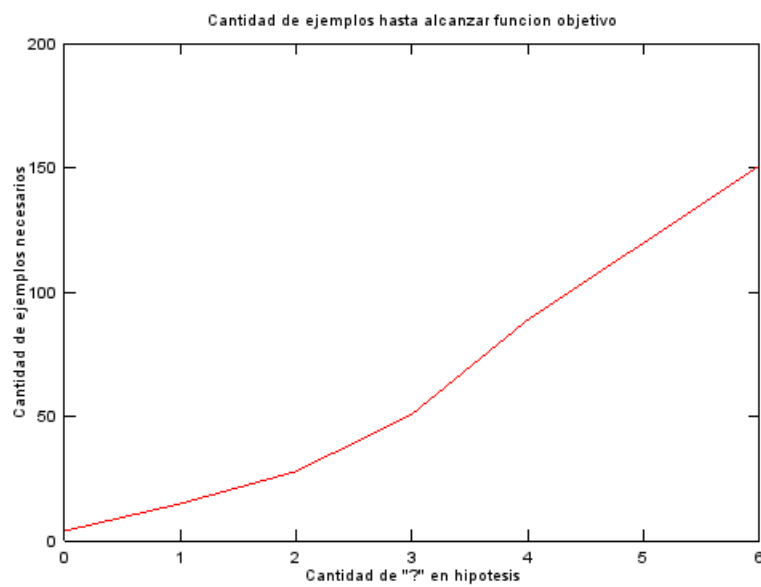


Figure 2: Resultados de la simulación para hipotesis variable