

CC5112 - Aprendizaje Computacional

Laboratorio 1: Perceptrón

Profesor: Pablo Barceló
Auxiliar: Mauricio Quezada

1 de septiembre de 2014

1 Repaso de *Perceptron Learning Algorithm*

Dado el dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$, donde $\mathbf{x}_i = (x_0, x_1, \dots, x_d)^T$, con $x_0 = 1$, $y_i = \pm 1$, para $i \in [1..N]$, y $d > 0$. El algoritmo es el siguiente:

1. Inicializar $\mathbf{w}(0) = (0, 0, \dots, 0)^T$
2. Iterar para $t = 0, 1, 2, \dots$. Actualizar $\mathbf{w}(t)$ según la siguiente regla:

$$\mathbf{w}(t+1) = \begin{cases} \mathbf{w}(t) + y(t)\mathbf{x}(t) & \text{si } y(t) \neq \text{sign}(\mathbf{w}^T(t)\mathbf{x}(t)) \\ \mathbf{w}(t) & \text{si no} \end{cases}$$

3. Si no hay ningún punto $(\mathbf{x}, y)$ tal que $y \neq \text{sign}(\mathbf{w}^T \mathbf{x})$, entonces terminar y retornar $\mathbf{w}$.

Si X es una matriz de $N \times (d+1)$ elementos, es decir, donde las filas contienen a cada uno de los puntos $\mathbf{x}_i$, entonces podemos computar las predicciones del perceptrón mediante:

$$h(\mathbf{w}) = \text{sign}(X\mathbf{w})$$

El error de la muestra $E_{\text{in}}(h)$ de una hipótesis h se define como la fracción de $\mathcal{D}$ donde la función objetivo f y la hipótesis h no concuerdan:

$$E_{\text{in}}(h) = \frac{1}{N} \sum_{i=1}^N \mathbb{I}[h(\mathbf{x}_i) \neq f(\mathbf{x}_i)]$$

Donde $\mathbb{I}[\text{sentencia}] = 1$ si la sentencia es verdadera, y 0 si no. Finalmente, el error fuera de la muestra $E_{\text{out}}(h)$ se define como:

$$E_{\text{out}}(h) = \mathbb{P}[h(\mathbf{x}) \neq f(\mathbf{x})]$$

2 Actividad

Trabaje con otro compañero o compañera. Entreguen al final de la clase en la sección Tareas de U-Cursos un documento PDF con los resultados, gráficos y conclusiones correspondientes.

1. Implemente PLA en Octave. Complete la siguiente plantilla (`pla.m`):

```
# X es una matrix de N x (d+1) siendo cada fila un punto x_i
# y es un vector de N x 1 con la clasificación de cada punto x_i
# retorna w, el vector de pesos y t, la cantidad de iteraciones realizadas
function [w, t] = pla(X, y)
    N = size(X, 1);      # filas de X
    d = size(X, 2) - 1;  # total dimensiones
    w = zeros(d + 1, 1); # w vector de pesos
    t = 0;                # iteraciones

    while 1
        t = t + 1;
        # COMPLETAR AQUI: actualizar w
    end
end

function [xt, yt] = misclassified(X, y, N, w)
    # COMPLETAR AQUI: escoger un punto (xt, yt) tal que yt != sign(w^T * xt)
end
```

2. Genere un dataset de tamaño $N = 20$ usando la función `gen_lindata` y luego grafique tanto los puntos como la recta correspondiente usando `plot_points` (ambos archivos en Material Docente), como sigue:

```
w0 = [-1; 1; 1]; # genere la recta que considere adecuada
[X, y] = gen_lindata(N=20, d=2, w=w0); # genera puntos linealmente separables usando w0
plot_points(X, w0, x_min = -1, x_max = 2) # grafica los puntos y la recta w0
print -dpng grafico.png # guarda el grafico en el directorio actual
```

3. Ejecute PLA sobre el dataset (X, y) obtenido en la parte anterior y grafique los puntos con la recta dada por el w resultante usando `plot_points`. Comente si la recta resultante es similar a la producida por w_0 .
4. Genere nuevos datasets con $N \in \{20, 100, 1000\}$ y $d = 2$ y repita el experimento de (3). Compare sus resultados en términos de tiempo de ejecución y la recta resultante. Para medir el tiempo de una función o script en Octave:

```
tic; funcion(); toc;
```

5. Genere un dataset de $N = 1000$ y $d = 10$ (usando un vector w_0 de $d + 1$ dimensiones tal que hayan $+1$ y -1 en la clasificación) y ejecute PLA. ¿Cuántas iteraciones hace el algoritmo antes de terminar?
6. Repita el experimento anterior, realizando 100 repeticiones. En esta ocasión, modifique la función `misclassified` de forma que a cada llamada escoja $\mathbf{x}(t)$ al azar y no de forma determinística. Genere un histograma con la cantidad de iteraciones que toma PLA antes de converger. Tome el tiempo del experimento. Use la función `hist(A)` para generar un histograma. Con `xlab`, `ylab` y `title` puede etiquetar el gráfico.
7. Resuma sus conclusiones con respecto a la cantidad de iteraciones y al tiempo de ejecución en función de N y d .