Tópicos de investigación CM072

César Lara Avila

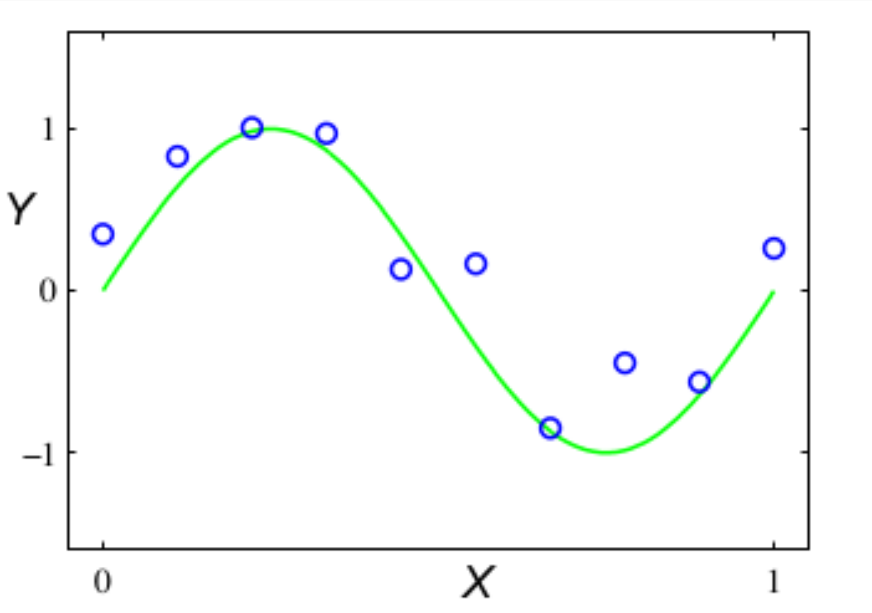
27 de septiembre de 2017

https://github.com/C-Lara

2. Teoria del aprendizaje

Ejemplo de regresión :

Conjunto de datos: 10 puntos (X, Y) generados de una función seno con ruido.

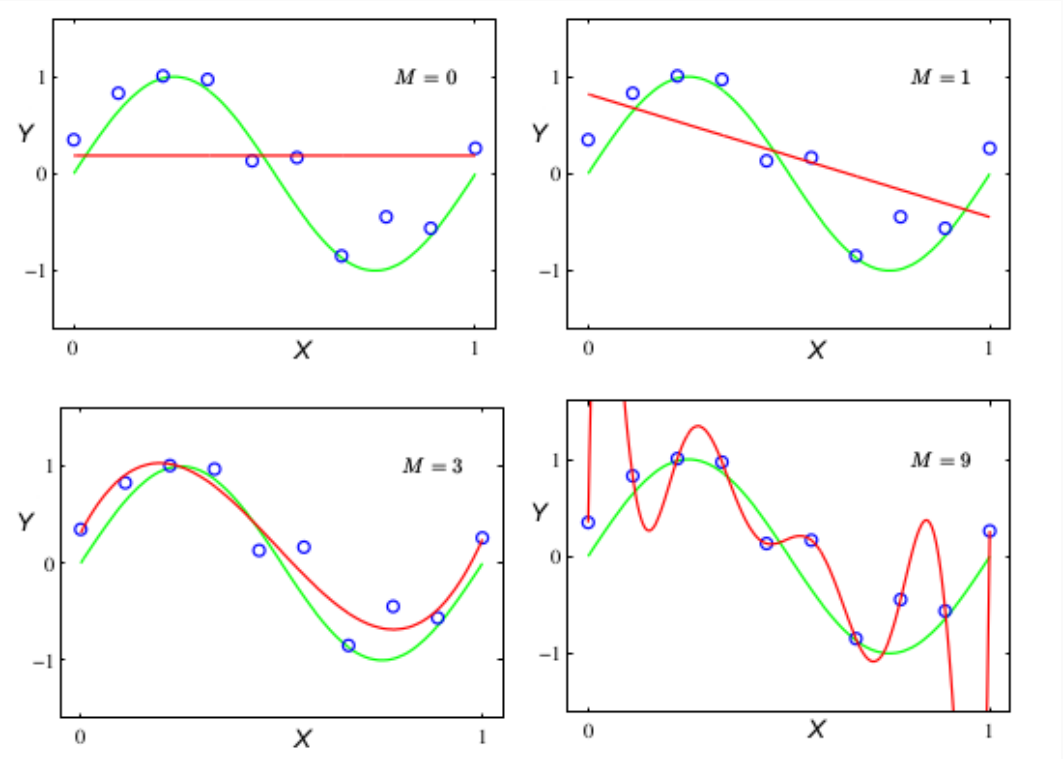


- **Regresión**
  - $f : X \to Y$
  - $X = \mathbb{R}$
  - $Y = \mathbb{R}$

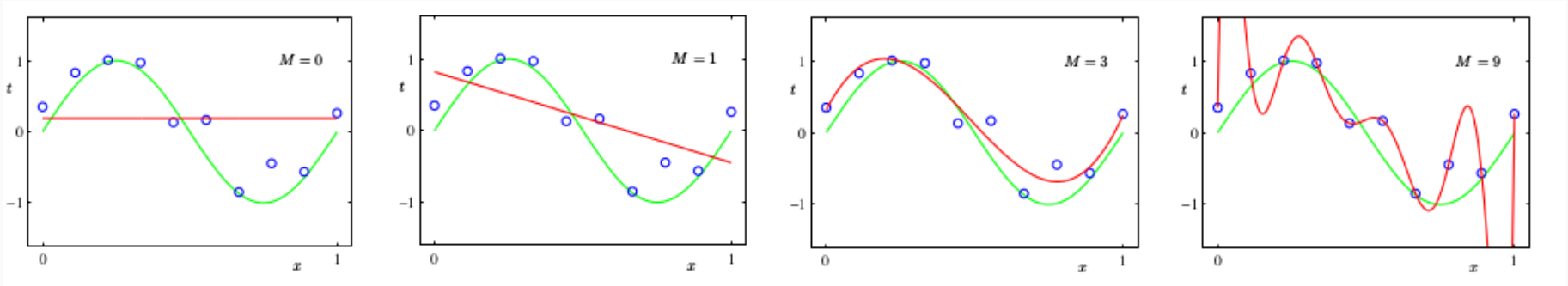
Polinomios de grado M

¿Que tal si dejamos que f sea un polinomio de grado M?

- ¿ Cuál es el **mejor**?



Espacio de hipótesis: Polinomios de grado N



Medimos el error usando una función de pérdida $L(y, \hat{y})$.

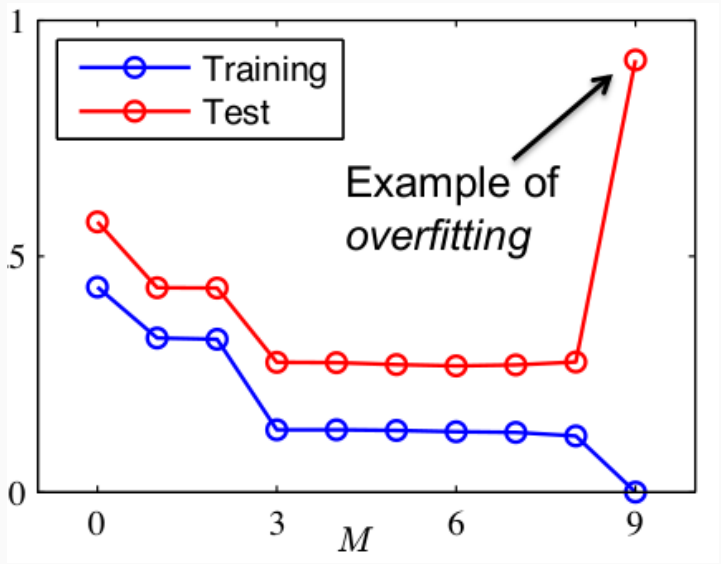
Para la regresión, una opción común es la pérdida al cuadrado:

$$L(y_i, f(x_i)) = (y_i - f(x_i))^2$$

La pérdida empírica de la función f aplicada a los datos de entrenamiento es entonces:

$$\frac{1}{N}\sum_{i=1}^{N} L(x_i, f(x_i)) = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))^2$$

Curva de aprendizaje



Medida de la complejidad del modelo

Clasificación binaria

- **Entradas:** Email,
- **Salida:** Spam/Ham
- **Preparación:**
  - Obtenemos una gran colección de correos electrónicos de ejemplo, cada uno etiquetado como `spam` o ham.
  - Nota: alguien tiene que manejar las etiquetas de todos estos datos!.
  - Quieres aprender a predecir etiquetas de nuevos, futuros correos electrónicos.
- **Características:** Los atributos utilizados para tomar la decisión de `ham/spam`.
  - Palabras: GRATIS!
  - Patrones de texto: $*dd* , CAPS
  - Sin texto: SenderInContacts

Estimado Estudiante.

Debo solicitar su confianza en esta transacción, esto es en virtud de su naturaleza como totalmente confidencial y secreto. ... ✗

PARA SER REMOVIDO DE CORREOS FUTUROS, PONER **REMOVER** EN EL ASUNTO MENSAJE.
99 MILLONES DE DIRECCIONES DE CORREO ELECTRONICO POR SOLO 59 SOLES. ✗

Eliminé mi archivo configuración `.gitconfig` Linux. Tuve que utilizar `git reset`, con un poco de suerte completaré las tareas y asignaciones bajo cambio de fechas. ✓

El algoritmo del perceptron

- 1957: El algoritmo perceptron fue inventado por Frank Rosenblatt.
- Construido sobre el trabajo de Hebbs (1949); también desarrollado por Widrow-Hoff (1960).
- El algoritmo perceptrón es un algoritmo de descenso por gradiente y puede necesitar que se le presente más de una vez un determinado patrón del conjunto de entrenamiento.
- El **teorema de convergencia del perceptrón** demuestra que si las muestras son linealmente separables, el algoritmo converge a una solución adecuada.

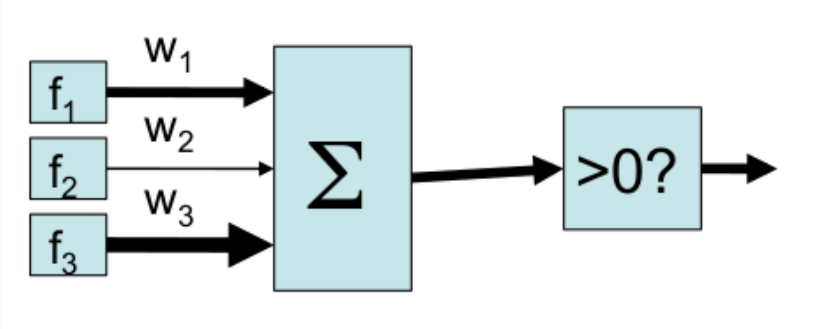
Clasificadores lineales

- Las entradas son **valores de características**.
- Cada característica tiene un **peso**.
- La suma es la **activación**.

$$\text{activacion}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$

- Si la activación es:

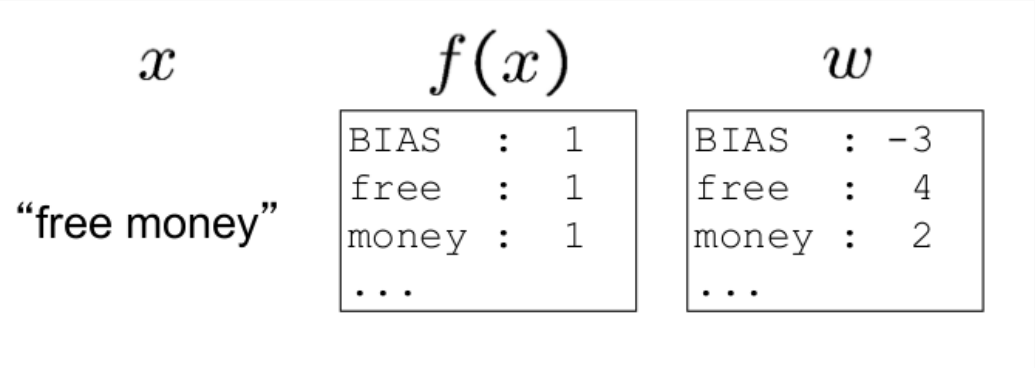
- Positiva, salida de la clase 1
- Negativa, salida de la clase 2



Ejemplo: Spam

Imagina 3 características (el spam es una clase positiva):

1. `free` (número de ocurrencias de `free`).
2. `money` (ocurrencias de `money`)
3. BIAS (interceptar, siempre tiene el valor 1)



$$\sum_i w_i \cdot f_i(x) = (1)(-3) + (1)(4) + (1)(2) + \cdots = 3$$

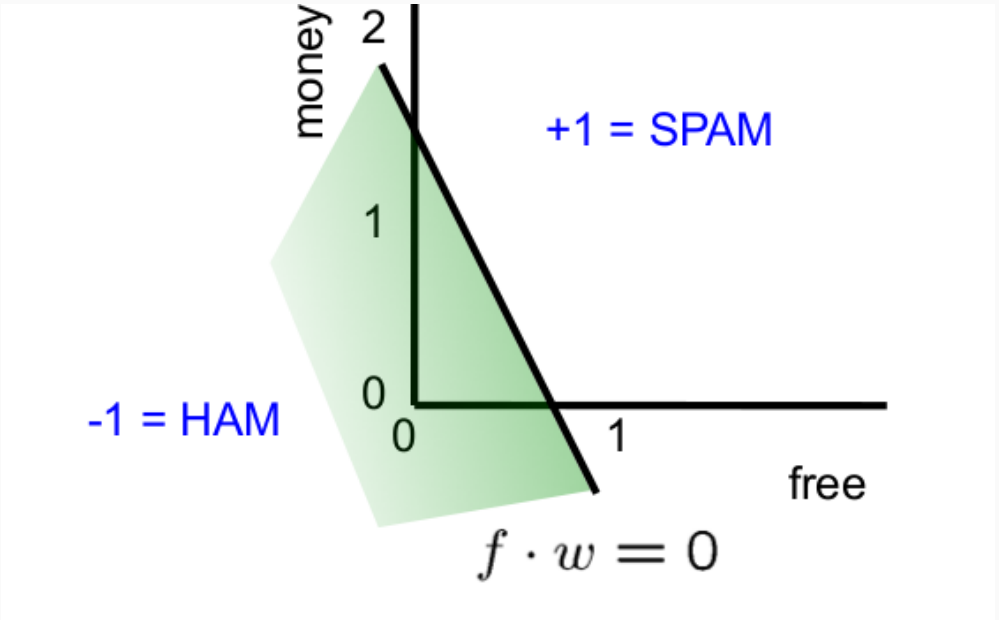
$\mathbf{w}.\mathbf{f}(\mathbf{x}) > \mathbf{0} \rightarrow$ SPAM .

Regla de decisión binaria

- En el espacio de los vectores de características
  - Ejemplos son los puntos
  - Cualquier vector de peso es un hiperplano
  - Un lado corresponde a $Y = +1$
  - Otro corresponde a $Y = -1$

w

```
BIAS  : -3
free  :  4
money :  2
...
```



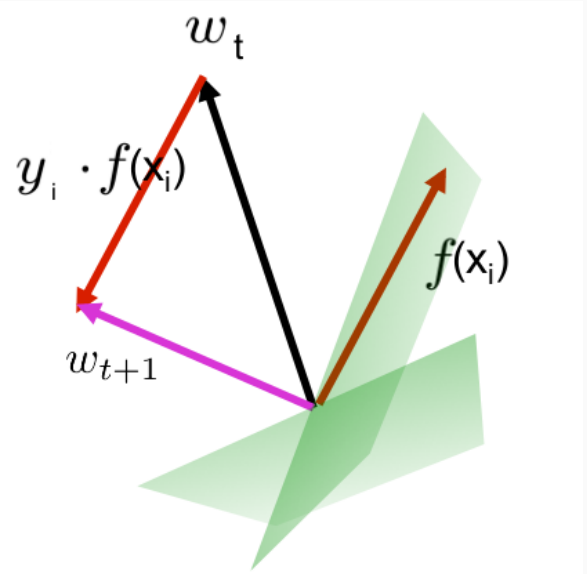
El algoritmo del perceptron

- Empezamos con un vector peso $\mathbf{0}$.
- Para cada instancia de entrenamiento (x_i, y_i):
  - Clasificamos con los pesos actuales

$$y = \begin{cases} +1 \text{ si } w \cdot f(x_i) \geq 0 \\ -1 \text{ si } w \cdot f(x_i) < 0 \end{cases}$$

  - Si es correcto (es decir, $y = y_i$), no hay cambio!.
  - Si es incorrecto: actualizamos

$$w = w + y_i f(x_i)$$



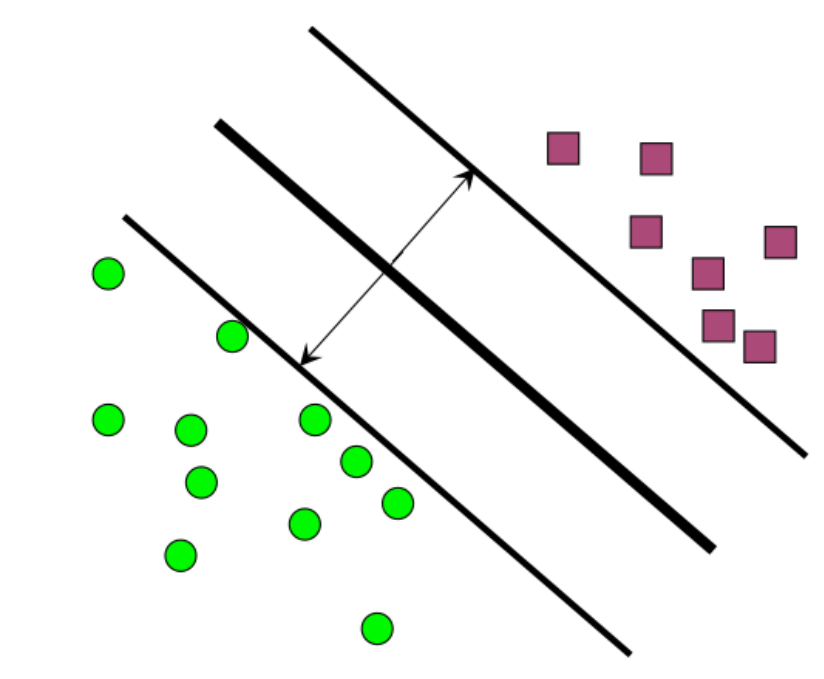
Preguntas acerca de un algoritmo de aprendizaje

- ¿Cuál es el tiempo de ejecución del algoritmo perceptron?.
- Si existe un vector de peso con pequeño error de entrenamiento, puede el perceptrón encontrar el error?.
- ¿Qué tan bien el clasificador resultante generaliza los datos no vistos?.

Separabilidad lineal

$\exists \mathbf{w}$ tal que $\forall t \qquad y_t(\mathbf{w} \cdot \mathbf{x}_t) \geq \gamma > 0$

donde γ es llamado el margen funcional con respecto al conjunto de entrenamiento.



Equivalentemente, para $y_t = +1$,

$$w \cdot x_t \geq \gamma$$

y para $y_t = -1$,

$$w \cdot x_t \leq \gamma$$

Error para el perceptrón

- Supongamos que el conjunto de datos D es linealmente separable con un margen geométrico γ,

$$\exists w^* \text{tal que} \quad \|w^*\| = 1 \quad \text{y} \quad \forall t \; y_t(w^* \cdot x_t) \geq \gamma$$

- Asumimos que $\|x_t\| \leq R, \; \forall t$

Teorema

El número máximo de errores cometidos por el algoritmo perceptrón está limitado por R^2/γ^2.

Problemas con el algoritmo perceptron

- Si los datos no son linealmente separables, no hay garantías de convergencia o precisión de entrenamiento.
- Incluso si los datos de entrenamiento son linealmente separables, perceptrón puede ser sobrecargado
- El perceptrón promedio es una modificación algorítmica que ayuda con ambos problemas
  - Promedio de los vectores de peso a través de todas las iteraciones.