P1.- En clase vimos, el caso de "múltiples clases ($K>2$)" puede ser enfrentado mediante una extensión del caso binario, con las técnicas

i) one-versus-rest

ii) one-versus-one

El problema con ambos, es que dejan regiones indefinidas en el espacio, pues solo consideran pares de clases que al ser agregados son incoherentes o podrían serlo.

Una alternativa más robusta, es construir un clasificador para K clases que contiene K funciones lineales de la forma

$$y_k(x) = a_k^T x + b_k \quad , \quad k=1, \dots, K.$$

Donde x es asignado a la clase C_k ssi $y_k(x) > y_j(x)$, $\forall j \neq k$, i.e.,

$$C(x) = \arg\max_{k} \; y_j(x)$$

Notemos que,

i) Las K componentes de la partición del espacio generado por este clasificador son convexas ya que es intersección finita de semiespacios

ii) Ningún punto queda sin clasificar. En efecto, x es asignado C_k ssi $y_k(x) > y_j(x)$, $\forall j \neq k$ con índices de $k=1, \dots, K$, es decir cada clase queda determinada por un máximo de un conjunto finito.

Supongamos que x es asignado a dos clases C_k y $C_{\tilde{k}}$, sin pérdida de generalidad, asumamos que $k > \tilde{k}$. Luego, como $x \in C_k$, $x \in C_{\tilde{k}}$ ssi $y_k(x) > y_j(x)$, $\forall j \neq k$ y $y_{\tilde{k}}(x) > y_j(x)$, $\forall j \neq \tilde{k}$, se sigue que $y_{\tilde{k}}(x) > y_k(x)$, lo cual no puede ocurrir.

iii) Si tenemos $K=2$, tenemos solo dos clases, es decir,

$$y_K(x) = a_K^T x + b_K \quad , K=1,2$$

donde x es asignado a C_1 ssi $y_1(x) > y_2(x)$, es decir

$$C(x) = \underset{\{1,2\}}{\arg\max}\ y_i(x)$$

y los puntos con más de una clase asignada, tienen medida cero.

P2: Convergencia del algoritmo del perceptrón.

Consideremos un conj. de datos $\{x_i\}_{i=1}^n \subset \mathbb{R}^m$ con etiquetas de clase $t_i \in \{-1,1\}$ $\forall i \in \{1,\dots,n\}$ separable. Pruebe que el algoritmo de entrenamiento del perceptrón converge a un hiperplano de separación en una cantidad finita de pasos.

Def: Diremos que dos conjuntos de puntos A, B en un espacio n-dimensional, son linealmente separable si existen $n+1$ números reales $w_1,\dots,w_n,K$ tal que, para todo punto $x \in A$ se cumple

$$\sum_{i=1}^{n} w_i x_i > K \quad , \forall x \in A$$

$$\sum_{i=1}^{n} w_i x_i < K \quad , \forall x \in B$$

donde x_i es la i-ésima componente de x.

P2: 2) Dado un θ_{old} el algoritmo identifica un punto z_i que esta mal clasificado y hace el update $\theta_{new} \leftarrow \theta_{old} + z_i t_i$

Pdq, $\|\theta_{new} - \theta_{sep}\|^2 \leq \|\theta_{old} - \theta_{sep}\|^2 - 1$ y por tanto el algoritmo converge.

Recordemos que el algoritmo de entrenamiento del perceptron es,

i) Recorrer el conjunto de puntos de entrenam. $\{x_i\}_{i=1}^{n}$

ii) Si x_i fue clasificado correctamente, se mantiene igual.

iii) Si x_i fue mal clasificado, el vector θ^k se actualiza, con $\eta = 1$,

$$\theta^{k+1} = \theta^k + \phi(x_i) t_i$$

$$\Rightarrow \|\theta^{k+1}\|^2 = \|\theta^k + \phi(x_i) t_i\|^2 = \|\theta^k\|^2 + \phi(x_i)^2 \|t_i\|^2 + 2\phi(x_i) t_i \cdot \theta^k$$

P3: Consideremos un modelo generativo de clasificación para K clases con priors $p(C_k) = \pi_k$ y probab. condicionales dado la clase $p(X|C_k)$. Supongamos que tenemos un conjunto de entrenamiento $\{(x^i, t^i)\}_{i=1}^N$, con $t^i = (t_k^i)_{k=1}^K$ codificando con un one hot encoding. todos los datos son independientes.

notemos que la función de verosimilitud para el modelo es

$$p(t^i, \mu_k, \Sigma) = \prod_{n=1}^{N} p(x_n, C_1)^{t_n} \cdot p(x_n, C_0)^{1-t_n}$$

Luego,

$$\log p(t, \mu_k, \Sigma) = \sum_n t_n \log (p_n, C_1) + \sum_n (1-t_n) \log (p_n, C_0)$$

Si $X|C_k \sim N(x, |\mu_k, \Sigma)$, entonces

$$\begin{aligned}\log p(t, \mu_k, \Sigma) = {} & \sum t_n \log \pi_k - \sum t_n \log (2\pi \sqrt{\Sigma}) \\ & - \frac{1}{2} \sum t_n (x_n - \mu_n)^T \Sigma^{-1} (x_n - \mu_1) \\ & + \sum_{n=1} (1-t_n) \log (1-\pi_k) \\ & - \sum_n (1-t_n) \log (2\pi |\Sigma|^{\frac{1}{2}}) - \frac{1}{2} \sum (1-t_n) \cdot (x_n - \mu)^T \Sigma^{-1} (x_n - \mu)\end{aligned}$$

Derivando con respecto a π_k

$$\frac{\partial \log p}{\partial \pi} = 0 \Rightarrow \frac{1}{\pi_k} \sum_n t_n = \frac{1}{1-\pi_k} \sum (1-t_n)$$

P3: