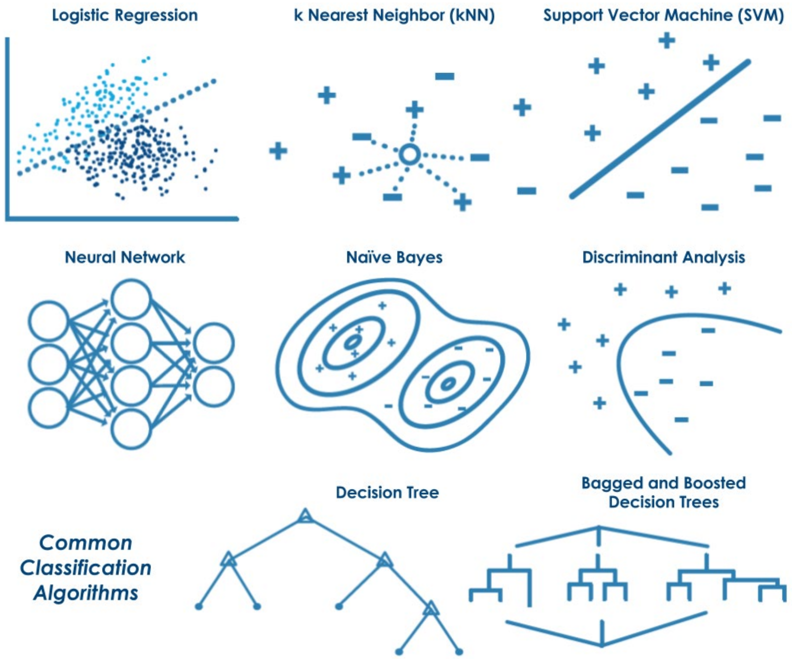


Clasificación II
Modelos lineales de clasificación

Jonnatan Arias Garcia
jonnatan.arias@utp.edu.co
jariasg@uniquindio.edu.co

David Cardenas peña - dcardenasp@utp.edu.co
Hernán Felipe Garcia - hernanf.garcia@udea.edu.co

Contenido

Modelos discriminativos probabilísticos

- ❑Logístico
- ❑Optimizadores
  - ❑Gradiente descendiente
  - ❑Mínimos reponderados

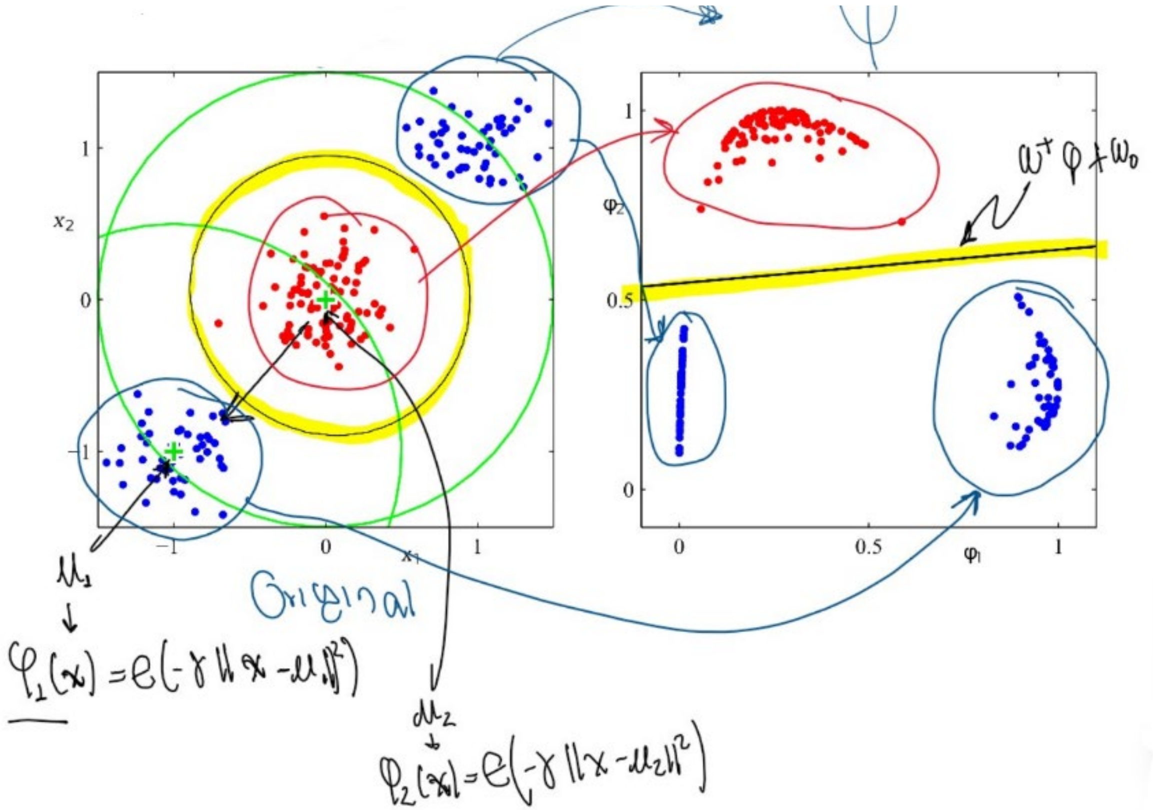
Modelos discriminativos probabilísticos

Modelos discriminativos

- Consiste en ligar modelos o distribuciones de cada clase por separado, se enfocan principalmente en la mejor manera de diferenciar entre clases.
- Proporcionan una estimación de la probabilidad de pertenencia
- Dan información de la incertidumbre debido a la probabilidad de pertenencia

Introducción

- Se modela directamente la función de probabilidad a posteriori $p(\mathcal{C}_k|\mathbf{x})$.
- En general, se necesita determinar un número menor de parámetros.
- Se pueden introducir funciones base $\phi(\mathbf{x})$. En el espacio transformado la separación podría ser lineal.



Regresión Logística (i)

- En forma general

$$p(\mathcal{C}_1|\phi) = y(\phi) = \sigma(\mathbf{w}^\top \phi),$$

donde $\sigma(\cdot)$ es la función logística sigmoidal.

- En estadística este modelo se conoce como *regresión logística*.

- Sea un conjunto de datos $\{\phi_n, t_n\}_{n=1}^N$, con $\phi_n = \phi(\mathbf{x}_n)$ y $t_n \in \{0, 1\}$.

$$t_n = \begin{cases} 1 & \boldsymbol{x}_n \in \mathcal{C}_1 \\ 0 & \boldsymbol{x}_n \in \mathcal{C}_2 \end{cases}$$

$$\boldsymbol{\varphi}_n = \boldsymbol{\varphi}(\boldsymbol{x}_n)$$

- La función de verosimilitud se define como

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n}\{1 - y_n\}^{1-t_n},$$

donde $\mathbf{t} = [t_1 \cdots t_N]^\top$, y $y_n = p(\mathcal{C}_1|\phi_n)$.

Regresión Logística (i. A demostración)

- Usando la variable booleana t_n como interruptor, la verosimilitud se puede escribir de forma resumida:

$$p(C_1|\varphi_n) + p(C_2|\varphi_n) = 1$$

$$p(C_2|\varphi_n) = 1 - p(C_1|\varphi_n)$$

$$p(\boldsymbol{t}|\boldsymbol{w}) = \prod_{n=1}^{N} p(C_1|\boldsymbol{\varphi}_n)^{t_n} p(C_2|\boldsymbol{\varphi}_n)^{1-t_n}$$

$$p(\boldsymbol{t}|\boldsymbol{w}) = \prod_{n=1}^{N} p(C_1|\boldsymbol{\varphi}_n)^{t_n} [1 - p(C_1|\boldsymbol{\varphi}_n)]^{1-t_n}$$

$$p(\boldsymbol{t}|\boldsymbol{w}) = \prod_{n=1}^{N} {y_n}^{t_n} (1 - y_n)^{1-t_n}$$

Regresión Logística (i. B análisis)

- Se define una función de costo tomando el logaritmo negativo de la función de verosimilitud:

costo $$E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = -\sum_{n=1}^{N} t_n \ln y_n + (1 - t_n)\ln(1 - y_n)$$

$\min_{w}$ costo

$$t_{nk} = \begin{cases} 1 & : x_n \in G_k \\ 0 & : x_n \notin G_k \end{cases}$$

Regresión Logística (ii)

- Se define una función de error tomando el logaritmo negativo de la función de verosimilitud

$$E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = -\sum_{n=1}^{N} t_n \ln y_n + (1 - t_n)\ln(1 - y_n).$$

donde $\sigma(\cdot)$ es la función logística sigmoidal.

- Se tiene en cuenta la relación $\frac{d\sigma}{da} = \sigma(1-\sigma)$.

- El gradiente de la función de error con respecto a **w** sigue la forma

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N}(y_n - t_n)\phi_n = \mathbf{\Phi}^\top(\mathbf{y} - \mathbf{t}),$$

donde $\mathbf{y} = [y_1 \cdots y_N]^\top$.

Optimización por Gradiente Descendiente (i)

- La minimización de la función de costo se realiza mediante gradiente descendiente, así que:

$$\nabla E(\boldsymbol{w}) = \sum_{n=1}^{N} \frac{de_n(y_n)}{dy_n} \frac{dy_n(\boldsymbol{w})}{d\boldsymbol{w}}$$

$$\nabla E(\boldsymbol{w}) = \sum_{n=1}^{N} \frac{de_n(y_n)}{dy_n} \frac{dy_n(a_n)}{da_n} \frac{da_n}{d\boldsymbol{w}}$$

donde:

$$e_n = -t_n \ln y_n - (1 - t_n) \ln(1 - y_n)$$

$$y_n = \sigma(a_n)$$

$$a_n = \boldsymbol{w}^\top \boldsymbol{\varphi}_n$$

Optimización por Gradiente Descendiente (i. Dem. II

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} \frac{de_n(y_n)}{dy_n}\frac{dy_n(a_n)}{da_n}\frac{da_n}{d\mathbf{w}}$$

$$\frac{dy_n(a_n)}{da_n} = \sigma(a)\left[1-\sigma(a)\right]$$

$$\frac{da_n}{d\mathbf{w}} = \varphi$$

$$\frac{de_n(y_n)}{dy_n} = -\frac{t_n}{y_n} + \frac{(1-t_n)}{1-y_n}$$

$$\sigma'(a) = \frac{d}{da}\left\{\frac{1}{1+e(-a)}\right\} = -(1+e(-a))^{-2}\cdot e(-a)(-1)$$

$$= \frac{e(-a)}{(1+e(-a))^2} = \underbrace{\frac{1}{1+e(-a)}}_{\sigma(a)}\cdot\frac{e(-a)}{1+e(-a)}$$

$$= \sigma(a)\,\frac{e(-a)+1-1}{1+e(-a)} = \sigma(a)\left[\frac{1+e(-a)}{1+e(-a)} \xrightarrow{} 1 - \underbrace{\frac{1}{1+e(-a)}}_{\sigma(a)}\right]$$

$$\nabla_w E = \sum_n\left(-\frac{t_n}{y_n} + \frac{(1-t_n)}{1-y_n}\right)\cancel{\sigma(a)}^{\,y_n}\left[1-\cancel{\sigma(a_n)}^{\,y_n}\right]\varphi_n$$

$$= \sum_n\left[-\frac{\cancel{y_n}\,t_n}{\cancel{y_n}} + \frac{y_n(1-t_n)}{1-y_n}\right]\left[1-y_n\right]\varphi_n$$

$$= \sum_n\left[-(1-y_n)t_n + \frac{y_n(1-t_n)\cancel{(1-y_n)}}{\cancel{1-y_n}}\right]\varphi_n$$

$$= \sum_n\left(-t_n + \cancel{y_n t_n} + y_n - \cancel{y_n t_n}\right)\varphi_n$$

$$\nabla_w E = \sum_n \underbrace{(y_n - t_n)}_{\text{“}e_n\text{”}}\varphi_n \rightarrow \sum_n\left(\sigma(w^T\varphi_n) - t_n\right)\varphi_n$$

0.5

Optimización por Gradiente Descendiente (ii)

- El algoritmo de gradiente descendiente queda:

1. Inicializar el modelo con $\boldsymbol{w}_0$ y fijar una tasa de aprendizaje μ.

2. Realizar las predicciones con los parámetros actuales $\boldsymbol{y} = \sigma(\boldsymbol{\Phi}\boldsymbol{w}_k)$

3. Calcular la función de costo total para los parámetros actuales $L(\boldsymbol{w}_k) = -\boldsymbol{t}^\top \ln \boldsymbol{y} - (1 - \boldsymbol{t})^\top \ln(1 - \boldsymbol{y})$.

4. Actualizar los parámetros para la siguiente iteración $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \mu\nabla E(\boldsymbol{w}_k)$

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \mu\boldsymbol{\Phi}^\top(\boldsymbol{t} - \boldsymbol{y})$$

5. Hasta la convergencia volver al paso 2.

Mínimos cuadrados reponderados iterativos (i)

- El gradiente de la función de costo con respecto a $\boldsymbol{w}$ sigue la forma

$$\nabla E(\boldsymbol{w}) = \sum_{n=1}^{N} (y_n - t_n)\boldsymbol{\varphi}_n = \boldsymbol{\Phi}^\top(\boldsymbol{y} - \boldsymbol{t})$$

donde $\boldsymbol{y} \in [0,1]^N$

- La función de costo puede minimizarse también usando el algoritmo de *Newton-Raphson* bajo la regla

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \boldsymbol{H}^{-1}\nabla E(\boldsymbol{w})$$

donde $\boldsymbol{H}$ es la matriz de segundas derivadas (Hessiana), $\boldsymbol{H} = \nabla\nabla E(\boldsymbol{w})$:

$$\boldsymbol{H} = \nabla\nabla E(\boldsymbol{w}) = \frac{d}{d\boldsymbol{w}}\{\boldsymbol{\Phi}^\top(\boldsymbol{y}(\boldsymbol{w}) - \boldsymbol{t})\}$$

$$\boldsymbol{H} = \nabla\nabla E(\boldsymbol{w}) = \sum_{n=1}^{N} y_n(\boldsymbol{w})(1 - y_n(\boldsymbol{w}))\boldsymbol{\varphi}_n\boldsymbol{\varphi}_n^\top$$

$$\boldsymbol{H} = \nabla\nabla E(\boldsymbol{w}) = \boldsymbol{\Phi}^\top\mathbf{R}(\boldsymbol{w})\boldsymbol{\Phi}$$

siendo $\boldsymbol{R} \in [0,1]^{N\times N}$ una matriz diagonal con elementos $R_{nn} = y_n(1 - y_n)$

Mínimos cuadrados reponderados iterativos (ii)

- Note que la solución para $\boldsymbol{w}$ debe encontrarse de forma iterativa, debido a que los elementos de $\boldsymbol{R}$ dependen de $\boldsymbol{w}$.

- La solución para $\boldsymbol{w}$ se puede escribir como

$$\boldsymbol{w}_{k+1} = (\boldsymbol{\Phi}^\top \mathbf{R}_k \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top \mathbf{R}_k \boldsymbol{z}_k$$

donde $\boldsymbol{z}_k = \boldsymbol{\Phi} \boldsymbol{w}_k - \boldsymbol{R}_k (\boldsymbol{y}_k - \boldsymbol{t})$

- Teniendo en cuenta que:

  - La solución anterior es parecida a la solución de mínimos cuadrados para el problema de regresion lineal.

  - Las ecuaciones normales se deben aplicar iterativamente.

Este algoritmo se conoce como *mínimos cuadrados reponderados iterativos* (IRLS - Iterative Reweighted Least Squares).