

Agenda

Regresión

1 Introducción al problema de regresión

2 Regresión Lineal

3 Gradiente Descendente

4 Regresión Polinómica

Teoría de Regresión

Bases

Objetivo: Predecir el valor de Y a partir de $X \rightarrow \hat{Y} = \varphi(X)$

Función costo: Error cuadrático $\rightarrow \ell(x, y) = (y - \varphi(x))^2$

Riesgo Esperado: MSE $\rightarrow \mathbb{E}[\ell(X, Y)] = \mathbb{E}[(Y - \varphi(X))^2]$

Teoría de Regresión

Bases

Objetivo: Predecir el valor de Y a partir de $X \rightarrow \hat{Y} = \varphi(X)$

Función costo: Error cuadrático $\rightarrow \ell(x, y) = (y - \varphi(x))^2$

Riesgo Esperado: MSE $\rightarrow \mathbb{E}[\ell(X, Y)] = \mathbb{E}[(Y - \varphi(X))^2]$

Optimalidad

$$\mathbb{E}[(Y - \varphi(X))^2] \geq \mathbb{E}[\text{var}(Y|X)]$$

con igualdad si y solo si $\varphi(x) = \mathbb{E}[Y|X = x]$.

Regresor óptimo: $\varphi(x) = \mathbb{E}[Y|X = x]$

Error Bayesiano: $\mathbb{E}[\text{var}(Y|X)]$

Reconocimiento de patrones

Objetivo

Quiero buscar $\varphi(\cdot)$ que minimice $\mathbb{E}[\ell(X, Y)]$. Es decir aprender la “esperanza condicional”.

Reconocimiento de patrones

Objetivo

Quiero buscar $\varphi(\cdot)$ que minimice $\mathbb{E}[\ell(X, Y)]$. Es decir aprender la “esperanza condicional”.

Empirical Risk Minimization (ERM)

Propongo buscar $\varphi(\cdot)$ que minimice el riesgo empírico: $\frac{1}{n} \sum_{i=1}^n \ell(X_i, Y_i)$

Reconocimiento de patrones

Objetivo

Quiero buscar $\varphi(\cdot)$ que minimice $\mathbb{E}[\ell(X, Y)]$. Es decir aprender la “esperanza condicional”.

Empirical Risk Minimization (ERM)

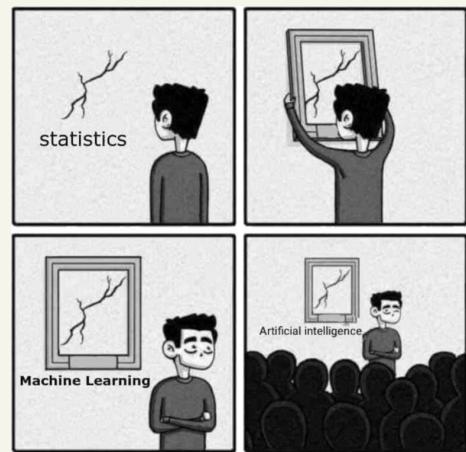
Propongo buscar $\varphi(\cdot)$ que minimice el riesgo empírico: $\frac{1}{n} \sum_{i=1}^n \ell(X_i, Y_i)$

Tradeoff: Sesgo/Varianza

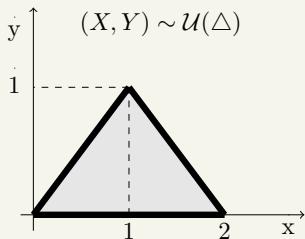
$$\mathbb{E}[\ell(X, Y)] = \underbrace{\frac{1}{n} \sum_{i=1}^n \ell(X_i, Y_i)}_{\text{Riesgo emp\'rico}} + \left(\underbrace{\mathbb{E}[\ell(X, Y)] - \frac{1}{n} \sum_{i=1}^n \ell(X_i, Y_i)}_{\text{Gap de generalizaci\'on}} \right)$$

Nota: El riesgo empírico se considera grande o pequeño comparándolo con el error bayesiano.

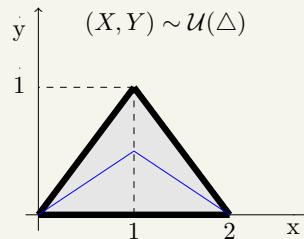
¿Qué es la Inteligencia Artificial?



Overfitting y Underfitting



Overfitting y Underfitting

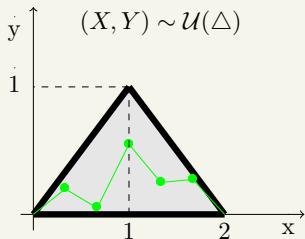


Solución Óptima

- El regresor elegido es efectivamente la esperanza condicional.
- El riesgo esperado alcanza el límite bayesiano

$$\mathbb{E} [\text{var}(Y|X)] = \frac{1}{24}$$

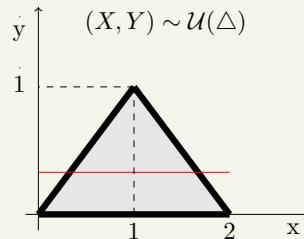
Overfitting y Underfitting



Problema de overfitting

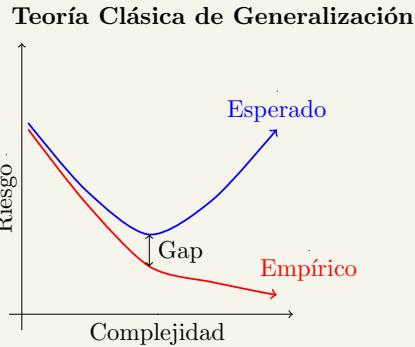
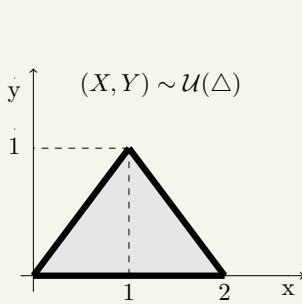
- Riesgo empírico muy bajo (puede ser menor incluso que el bayesiano)
- Se detecta por el alto gap de generalización.
- Exceso de complejidad en el modelado.
- Se dice que el algoritmo tiene un problema de varianza.

Overfitting y Underfitting



Problema de underfitting

- Suele tener bajo gap de generalización.
- Riesgo empírico muy superior al error bayesiano.
- Escasez de complejidad en el modelado.
- Se dice que el algoritmo tiene un problema de sesgo.



Regresión Lineal: $\hat{Y} = w^T \cdot X + b$

Idea

Me aseguro mantener acotado el problema de overfitting proponiendo una solución de extremadamente baja complejidad. Si se alcanza bajo error empírico, entonces tengo ciertas garantías de que el algoritmo alcanza un buen desempeño.

Regresión Lineal: $\hat{Y} = w^T \cdot X + b$

Idea

Me aseguro mantener acotado el problema de overfitting proponiendo una solución de extremadamente baja complejidad. Si se alcanza bajo error empírico, entonces tengo ciertas garantías de que el algoritmo alcanza un buen desempeño.

Empirical Risk Minimization

$$(w, b) \in \arg \min_{(w,b)} \sum_{i=1}^n (w^T \cdot X_i + b - Y_i)^2$$

Regresión Lineal: $\hat{Y} = w^T \cdot X + b$

Empirical Risk Minimization

$$(w, b) \in \arg \min_{\mathbf{w}} \|\mathbf{X} \cdot \mathbf{w} - \mathbf{y}\|^2,$$

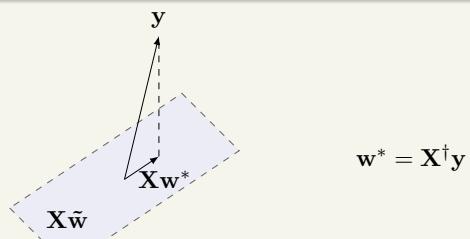
$$\mathbf{X} = \begin{pmatrix} 1 & X_1^T \\ 1 & X_2^T \\ \vdots & \vdots \\ 1 & X_n^T \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}, \quad \mathbf{w} = \begin{pmatrix} b \\ w \end{pmatrix}$$

Regresión Lineal: $\hat{Y} = w^T \cdot X + b$

Empirical Risk Minimization

$$(w, b) \in \arg \min_{\mathbf{w}} \|\mathbf{X} \cdot \mathbf{w} - \mathbf{y}\|^2,$$

$$\mathbf{X} = \begin{pmatrix} 1 & X_1^T \\ 1 & X_2^T \\ \vdots & \vdots \\ 1 & X_n^T \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}, \quad \mathbf{w} = \begin{pmatrix} b \\ w \end{pmatrix}$$



Regresión Lineal

Solución matricial óptima: Recta de Regresión

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

Regresión Lineal

Solución matricial óptima: Recta de Regresión

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

Derivadas Matriciales

$$\nabla(\mathbf{x}^T \mathbf{a}) = \nabla(\mathbf{a}^T \mathbf{x}) = \mathbf{a}$$

$$\nabla(\mathbf{x}^T \mathbf{Bx}) = (\mathbf{B} + \mathbf{B}^T) \mathbf{x}$$

$$\mathcal{H}(\mathbf{x}^T \mathbf{Bx}) = \mathbf{B} + \mathbf{B}^T$$

¿Cuál es el gradiente de $J(\mathbf{w}) = \frac{1}{n} \|\mathbf{X} \cdot \mathbf{w} - \mathbf{y}\|^2$?

Petersen and Pedersen - "Matrix Cookbook".

TPS-IIA

Matias Vera

Regresión

10 / 30

Petersen and Pedersen - "Matrix Cookbook".

TPS-IIA

Matias Vera

Regresión

10 / 30

Regresión Lineal

Solución matricial óptima: Recta de Regresión

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

Derivadas Matriciales

$$\nabla(\mathbf{x}^T \mathbf{a}) = \nabla(\mathbf{a}^T \mathbf{x}) = \mathbf{a}$$

$$\nabla(\mathbf{x}^T \mathbf{Bx}) = (\mathbf{B} + \mathbf{B}^T) \mathbf{x}$$

$$\mathcal{H}(\mathbf{x}^T \mathbf{Bx}) = \mathbf{B} + \mathbf{B}^T$$

¿Cuál es el gradiente de $J(\mathbf{w}) = \frac{1}{n} \|\mathbf{X} \cdot \mathbf{w} - \mathbf{y}\|^2$?

Optimización convexa

El problema de regresión lineal es un problema convexo.

Petersen and Pedersen - "Matrix Cookbook".

TPS-IIA

Matias Vera

Regresión

10 / 30

TPS-IIA

Matias Vera

Regresión

11 / 30

Outline

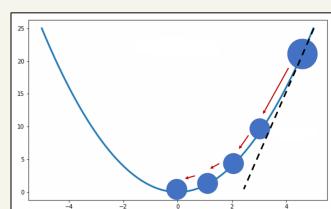
- ① Introducción al problema de regresión
- ② Regresión Lineal
- ③ Gradiente Descendente
- ④ Regresión Polinómica

Gradiente Descendente

Problema a resolver: $\min_{\theta \in \Theta} J(\theta)$.

Solución:

$$\theta_{t+1} = \theta_t - \alpha \nabla J(\theta_t)$$



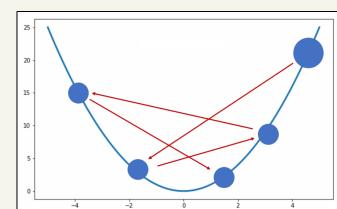
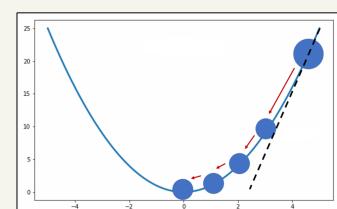
Gradiente Descendente

Problema a resolver: $\min_{\theta \in \Theta} J(\theta)$.

Solución:

$$\theta_{t+1} = \theta_t - \alpha \nabla J(\theta_t)$$

- Si α es chico la convergencia es lenta.
- Si α es grande puede no converger.



Cauchy 1847: "Méthode générale pour la résolution de systèmes d'équations simultanées".

TPS-IIA

Matias Vera

Regresión

12 / 30

Cauchy 1847: "Méthode générale pour la résolution de systèmes d'équations simultanées".

TPS-IIA

Matias Vera

Regresión

12 / 30

Convergencia y optimización para problemas convexos

Modelo convexo con derivadas segundas continuas

- $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \nabla J(\mathbf{w}_t)$.
- Existe un único \mathbf{w}^* tal que $\nabla J(\mathbf{w}^*) = 0$.
- $\mathcal{H}(\mathbf{w})$ es definido positivo para todo \mathbf{w} .

Convergencia y optimización para problemas convexos

Modelo convexo con derivadas segundas continuas

- $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \nabla J(\mathbf{w}_t)$.
- Existe un único \mathbf{w}^* tal que $\nabla J(\mathbf{w}^*) = 0$.
- $\mathcal{H}(\mathbf{w})$ es definido positivo para todo \mathbf{w} .

Teorema de Taylor

$$\nabla J(\mathbf{w}_t) = \nabla J(\mathbf{w}^*) + \mathcal{H}(\tilde{\mathbf{w}}) \cdot (\mathbf{w}_t - \mathbf{w}^*)$$

para algún $\tilde{\mathbf{w}}$ en el segmento que une \mathbf{w}_t y \mathbf{w}^* .

Convergencia y optimización para problemas convexos

Modelo convexo con derivadas segundas continuas

- $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \nabla J(\mathbf{w}_t)$.
- Existe un único \mathbf{w}^* tal que $\nabla J(\mathbf{w}^*) = 0$.
- $\mathcal{H}(\mathbf{w})$ es definido positivo para todo \mathbf{w} .

Teorema de Taylor

$$\nabla J(\mathbf{w}_t) = \nabla J(\mathbf{w}^*) + \mathcal{H}(\tilde{\mathbf{w}}) \cdot (\mathbf{w}_t - \mathbf{w}^*)$$

para algún $\tilde{\mathbf{w}}$ en el segmento que une \mathbf{w}_t y \mathbf{w}^* .

Diagonalización ortogonal

Toda matriz real, cuadrada y simétrica puede escribirse como $H = Q^T \Lambda Q$ con una matriz de autovalores Λ diagonal y una de vectores propios Q ortogonales $Q^T Q = Q Q^T = I$.

Convergencia y optimización para problemas convexos

$$\nabla J(\mathbf{w}_t) = \nabla J(\mathbf{w}^*) + \mathcal{H}(\tilde{\mathbf{w}}) \cdot (\mathbf{w}_t - \mathbf{w}^*) = Q^T \Lambda Q \cdot (\mathbf{w}_t - \mathbf{w}^*)$$

con Q y Λ las matrices correspondientes a la diagonalización de $\mathcal{H}(\tilde{\mathbf{w}})$.

Regresión lineal

Si $J(\mathbf{w}) = \frac{1}{n} \|\mathbf{X} \cdot \mathbf{w} - \mathbf{y}\|^2$, $\mathcal{H}(\tilde{\mathbf{w}}) = \frac{2}{n} \mathbf{X}^T \mathbf{X}$ no depende del valor de los parámetros (solo de los datos).

Convergencia y optimización para problemas convexos

$$\nabla J(\mathbf{w}_t) = \nabla J(\mathbf{w}^*) + \mathcal{H}(\tilde{\mathbf{w}}) \cdot (\mathbf{w}_t - \mathbf{w}^*) = Q^T \Lambda Q \cdot (\mathbf{w}_t - \mathbf{w}^*)$$

con Q y Λ las matrices correspondientes a la diagonalización de $\mathcal{H}(\tilde{\mathbf{w}})$.

Regresión lineal

Si $J(\mathbf{w}) = \frac{1}{n} \|\mathbf{X} \cdot \mathbf{w} - \mathbf{y}\|^2$, $\mathcal{H}(\tilde{\mathbf{w}}) = \frac{2}{n} \mathbf{X}^T \mathbf{X}$ no depende del valor de los parámetros (solo de los datos).

$$\begin{aligned}\mathbf{w}_{t+1} - \mathbf{w}^* &= \mathbf{w}_t - \mathbf{w}^* - \alpha \nabla J(\mathbf{w}_t) \\ &= (I - \alpha Q^T \Lambda Q) (\mathbf{w}_t - \mathbf{w}^*) \\ &= Q^T (I - \alpha \Lambda) Q (\mathbf{w}_t - \mathbf{w}^*)\end{aligned}$$

Convergencia y optimización para problemas convexos

$$\nabla J(\mathbf{w}_t) = \nabla J(\mathbf{w}^*) + \mathcal{H}(\tilde{\mathbf{w}}) \cdot (\mathbf{w}_t - \mathbf{w}^*) = Q^T \Lambda Q \cdot (\mathbf{w}_t - \mathbf{w}^*)$$

con Q y Λ las matrices correspondientes a la diagonalización de $\mathcal{H}(\tilde{\mathbf{w}})$.

Regresión lineal

Si $J(\mathbf{w}) = \frac{1}{n} \|\mathbf{X} \cdot \mathbf{w} - \mathbf{y}\|^2$, $\mathcal{H}(\tilde{\mathbf{w}}) = \frac{2}{n} \mathbf{X}^T \mathbf{X}$ no depende del valor de los parámetros (solo de los datos).

$$\begin{aligned} \mathbf{w}_{t+1} - \mathbf{w}^* &= \mathbf{w}_t - \mathbf{w}^* - \alpha \nabla J(\mathbf{w}_t) \\ &= (I - \alpha Q^T \Lambda Q) (\mathbf{w}_t - \mathbf{w}^*) \\ &= Q^T (I - \alpha \Lambda) Q (\mathbf{w}_t - \mathbf{w}^*) \end{aligned}$$

Defino $v_t = Q(\mathbf{w}_t - \mathbf{w}^*)$, luego:

$$v_{t+1} = (I - \alpha \Lambda) v_t,$$

$$\begin{aligned} \mathbf{w}_{t+1} - \mathbf{w}^* &= \mathbf{w}_t - \mathbf{w}^* - \alpha \nabla J(\mathbf{w}_t) \\ &= (I - \alpha Q^T \Lambda Q) (\mathbf{w}_t - \mathbf{w}^*) \\ &= Q^T (I - \alpha \Lambda) Q (\mathbf{w}_t - \mathbf{w}^*) \end{aligned}$$

Defino $v_t = Q(\mathbf{w}_t - \mathbf{w}^*)$, luego:

$$v_{t+1} = (I - \alpha \Lambda) v_t, \quad v_t = (I - \alpha \Lambda)^t v_0$$

Convergencia y optimización para problemas convexos

Condición y velocidad de convergencia

El GD convergerá si $|1 - \alpha \lambda_j| < 1$ para todo j y el learning rate óptimo estará asociado al criterio de peor caso:

$$\min_{\alpha} \max_j |1 - \alpha \lambda_j| \quad \text{s.t.} \quad |1 - \alpha \lambda_j| < 1 \quad \forall j$$

Convergencia y optimización para problemas convexos

Condición y velocidad de convergencia

El GD convergerá si $|1 - \alpha \lambda_j| < 1$ para todo j y el learning rate óptimo estará asociado al criterio de peor caso:

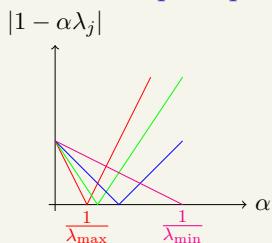
$$\min_{\alpha} \max_j |1 - \alpha \lambda_j| \quad \text{s.t.} \quad |1 - \alpha \lambda_j| < 1 \quad \forall j$$

Recordar que $\lambda_j > 0$ para todo j .

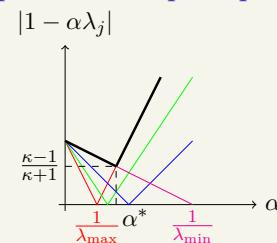
Condición de convergencia

$$|1 - \alpha \lambda_j| < 1 \text{ para todo } j \text{ equivale a pedir } \alpha < \frac{2}{\lambda_{\max}}.$$

Convergencia y optimización para problemas convexos



Convergencia y optimización para problemas convexos



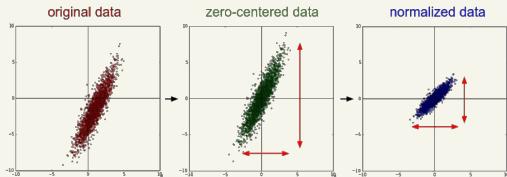
Velocidad de convergencia

El óptimo learning rate en este caso es $\alpha^* = \frac{2}{\lambda_{\max} + \lambda_{\min}}$ y su velocidad asociada $\left(\frac{\kappa-1}{\kappa+1}\right)^t$ depende del número de condición $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$.

Optimalidad

El óptimo no es el más grande convergente: $\alpha^* = \frac{2}{\lambda_{\max} + \lambda_{\min}} < \frac{2}{\lambda_{\max}}$

Normalización de la entrada



Normalizar *cada componente* de la entrada tiene sus beneficios:

$$(\mathbf{x})_k \leftarrow \frac{(\mathbf{x})_k - \mu_k}{\sigma_k}$$

donde las μ_k y σ_k son calculadas previo al entrenamiento como:

$$\mu_k = \frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} (\mathbf{x}_i)_k, \quad \sigma_k = \sqrt{\frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} [(\mathbf{x}_i)_k - \mu_k]^2}$$

¿Cuando y por que normalizar?

Normalizar si!

- Cuando quiero comparar magnitudes que por si solas no lo son.
- Cuando quiero corregir problemas de convergencia de los algoritmos.
- Cuando el algoritmo a utilizar, necesita la hipótesis de entradas normalizadas en su génesis.

Normalizar no!

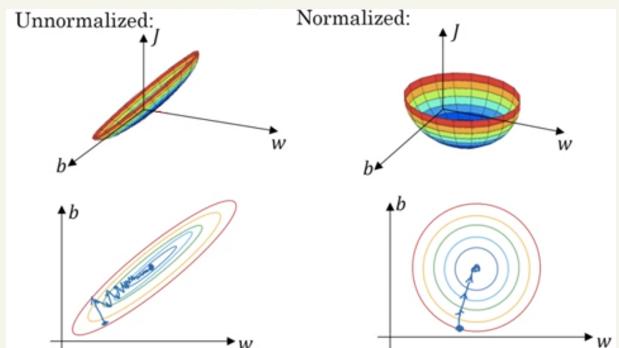
Normalizar por las dudas o por costumbre es una mala práctica.

¿Y si alguna varianza da cero?

Si algún $\sigma_k = 0$, significa que esa componente de la entrada es constante a lo largo de todo el conjunto de datos, y por lo tanto puede excluirse del análisis.

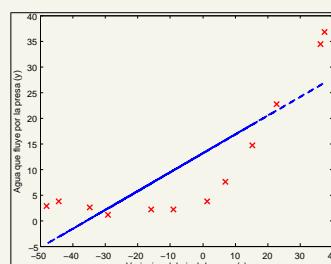
Normalización de la entrada

Normalizar me permite usar learning rates más grandes!



Outline

¿Y si la complejidad lineal no alcanza?



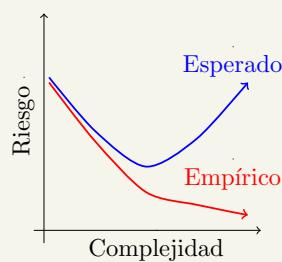
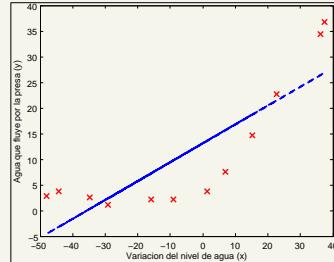
1 Introducción al problema de regresión

2 Regresión Lineal

3 Gradiiente Descendente

4 Regresión Polinómica

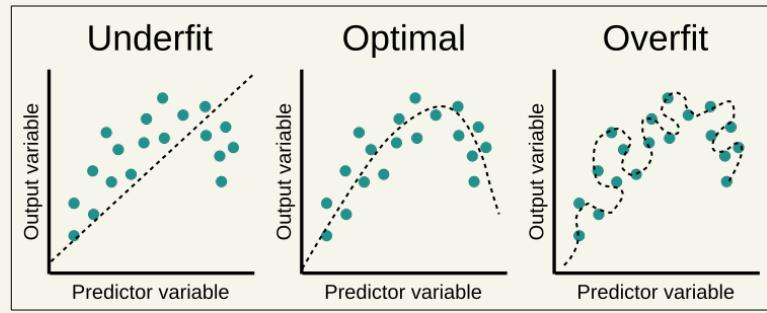
¿Y si la complejidad lineal no alcanza?



Regresión Polinómica

$$\mathbf{X} = \begin{pmatrix} 1 & X_{1,1} & X_{1,2} & X_{1,1}^2 & X_{1,2}^2 & X_{1,1}X_{1,2} \\ 1 & X_{2,1} & X_{2,2} & X_{2,1}^2 & X_{2,2}^2 & X_{2,1}X_{2,2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & X_{n,1} & X_{n,2} & X_{n,1}^2 & X_{n,2}^2 & X_{n,1}X_{n,2} \end{pmatrix}$$

Compromiso Sesgo/Varianza



Si no puedo confiar en los datos de entrenamiento ¿Como procedo?

Conjuntos de datos

- Conjunto de entrenamiento (train set):** Datos utilizados para minimizar el riesgo empírico. Sobre estos se produce el "aprendizaje". Las variables definidas a partir de este conjunto se llaman parámetros.
- Conjunto de validación (validation or development set):** Datos utilizados para comparar modelos. Las variables definidas a partir de este conjunto (o definidas previas al entrenamiento) se llaman hiperparámetros.
- Conjunto de testeo (test set):** Datos utilizados para evaluar la performance final del algoritmo. Su única función es presentar estimadores insesgados de las métricas de error y no es imprescindible.

Si la base de datos esta dividida, respetar la división!

Enfoque clásico: 60%/20%/20% - Típico para 1K, 10K muestras.

Big Data: Para 1M muestras, quizás alcanza con 98%/1%/1%.

Atacar el punto débil

¿Que conviene corregir? ¿Sesgo o varianza?

- Avoidable bias:** Error de train - Error bayesiano
- Generalization Gap:** Error de validación - Error de train

Atacar el punto débil

¿Que conviene corregir? ¿Sesgo o varianza?

- Avoidable bias:** Error de train - Error bayesiano
- Generalization Gap:** Error de validación - Error de train

Técnica Clásica de Regularización

Se agrega un término de penalización que perturba la optimización del riesgo empírico:

$$J(\theta) = \frac{1}{n} \sum_{i=1}^n L_i(\theta) + \lambda R(\theta)$$

Atacar el punto débil

¿Que conviene corregir? ¿Sesgo o varianza?

- Avoidable bias:** Error de train - Error bayesiano
- Generalization Gap:** Error de validación - Error de train

Técnica Clásica de Regularización

Se agrega un término de penalización que perturba la optimización del riesgo empírico:

$$J(\theta) = \frac{1}{n} \sum_{i=1}^n L_i(\theta) + \lambda R(\theta)$$

Motivación: Error de generalización

El regularizador trata ser representativo del error de generalización:

$$\mathbb{E}[L(\theta)] = \frac{1}{n} \sum_{i=1}^n L_i(\theta) + \left(\mathbb{E}[L(\theta)] - \frac{1}{n} \sum_{i=1}^n L_i(\theta) \right)$$

Regresión Lineal Regularizada

Weight decay or L2 regularization

$$J(\theta) = \frac{1}{n} \sum_{i=1}^n L_i(\theta) + \frac{\lambda}{n} \|\mathbf{w}\|^2 \rightarrow \frac{\partial \|\mathbf{w}\|^2}{\partial \mathbf{w}} = 2\mathbf{w}$$

Regresión Lineal Regularizada

Weight decay or L2 regularization

$$J(\theta) = \frac{1}{n} \sum_{i=1}^n L_i(\theta) + \frac{\lambda}{n} \|\mathbf{w}\|^2 \rightarrow \frac{\partial \|\mathbf{w}\|^2}{\partial \mathbf{w}} = 2\mathbf{w}$$

Interpretación 1: Apagar parámetros

$w_j \approx 0$ simplifica la complejidad del modelo.

Regresión Lineal Regularizada

Weight decay or L2 regularization

$$J(\theta) = \frac{1}{n} \sum_{i=1}^n L_i(\theta) + \frac{\lambda}{n} \|\mathbf{w}\|^2 \rightarrow \frac{\partial \|\mathbf{w}\|^2}{\partial \mathbf{w}} = 2\mathbf{w}$$

Interpretación 1: Apagar parámetros

$w_j \approx 0$ simplifica la complejidad del modelo.

Interpretación 2: Disminuir el máximo valor de la función costo

$$\mathbb{E}[L(\theta)] - \frac{1}{n} \sum_{i=1}^n L_i(\theta) \leq \max_{\phi \in \Theta} L(\phi)$$

Validación: ¿Como elijo el λ ?

Set de Validación

Si tengo una buena cantidad de datos de validación, elijo el λ con menor error de validación.

Validación: ¿Como elijo el λ ?

Set de Validación

Si tengo una buena cantidad de datos de validación, elijo el λ con menor error de validación.

Leave-one-out cross-validation (LOOCV)

Si tengo pocos datos no puedo tener un conjunto de datos de validación suficientemente rico. Entonces entreno con todas las muestras menos una y valido con la última. Luego repito esto con cada muestra y promedio.

Validación: ¿Como elijo el λ ?

Set de Validación

Si tengo una buena cantidad de datos de validación, elijo el λ con menor error de validación.

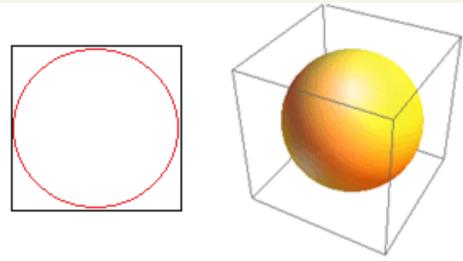
Leave-one-out cross-validation (LOOCV)

Si tengo pocos datos no puedo tener un conjunto de datos de validación suficientemente rico. Entonces entreno con todas las muestras menos una y valido con la última. Luego repito esto con cada muestra y promedio.

K-Fold

Separo en K subgrupos de $\frac{n}{K}$ muestras cada uno. Entreno con $K-1$ grupos y testeo con el último. Luego repito esto con cada grupo y promedio.

La maldición de la dimensionalidad



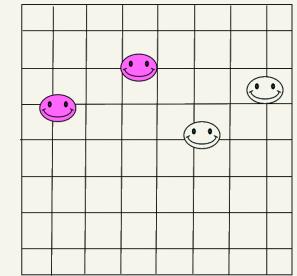
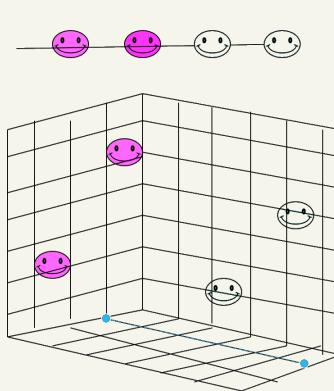
- 2d: $\frac{\pi r^2}{(2r)^2} \approx 78.5\%$
- 3d: $\frac{\frac{4}{3}\pi r^3}{(2r)^3} \approx 52.3\%$
- 10d: $\frac{r^{10}}{(2r)^{10}} \pi^5 \approx 0.25\%$

En grandes dimensiones:

- Los puntos están muy lejos.
- Las estructuras son muy sparce.
- La distancia euclídea no es buena métrica.
- La “necesidad” de muestras crece exponencialmente con la dimensión.

La maldición de la dimensionalidad

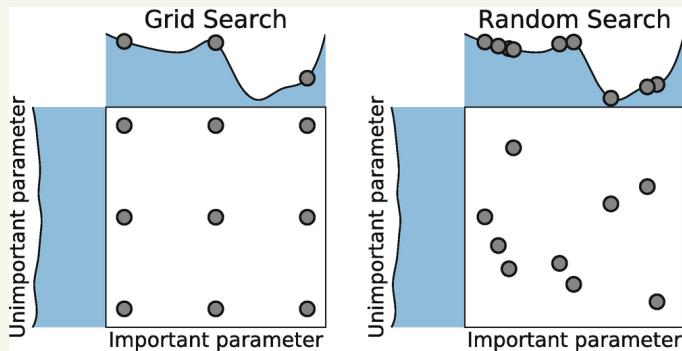
La maldición aplica a los hiperparámetros



La necesidad de pruebas crece exponencialmente con la cantidad de hiperparámetros!

Búsqueda aleatoria

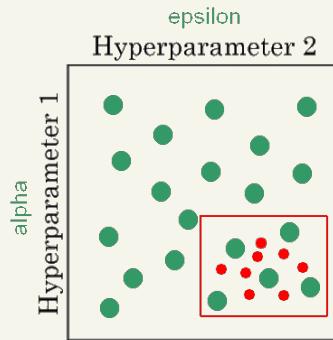
No todos los hiperparámetros son igual de importantes



Random search nos permite variar muchas veces todos los parámetros.

Búsqueda aleatoria

Hacerlo por etapas permite aprovechar más las simulaciones.



Simulo unos pocos puntos, veo donde está andando mejor y vuelvo a simular dentro de ese entorno.

Clasificación

Agenda

- ① Introducción al problema de clasificación
- ② Regresión Logística Binaria
- ③ Regresión Logística Categórica
- ④ Linear Discriminant Analysis
- ⑤ K-Vectinos más cercanos
- ⑥ Support Vector Machines
- ⑦ Árboles de decisión

Teoría de Clasificación

Bases

Objetivo: Clasificar Y (con $|\mathcal{Y}|$ finito) a partir del valor de X : $\hat{Y} = \varphi(X)$

Función costo: Hard $\rightarrow \ell(x, y) = \mathbb{1}\{y \neq \varphi(x)\}$

Riesgo Esperado: Probabilidad de error $\rightarrow \mathbb{P}(Y \neq \varphi(X))$

Teoría de Clasificación

Bases

Objetivo: Clasificar Y (con $|\mathcal{Y}|$ finito) a partir del valor de X : $\hat{Y} = \varphi(X)$

Función costo: Hard $\rightarrow \ell(x, y) = \mathbb{1}\{y \neq \varphi(x)\}$

Riesgo Esperado: Probabilidad de error $\rightarrow \mathbb{P}(Y \neq \varphi(X))$

Optimalidad

$$\mathbb{P}(Y \neq \varphi(X)) \geq 1 - \mathbb{E} \left[\max_y P_{Y|X}(y|X) \right]$$

con igualdad si y solo si $\varphi(x) = \arg \max_y P_{Y|X}(y|X)$.

Clasificador Bayesiano: $\varphi(x) = \arg \max_y P_{Y|X}(y|x)$

$$\text{Error Bayesiano: } 1 - \mathbb{E} \left[\max_y P_{Y|X}(y|X) \right]$$

Clasificadores extremos

Clasificador bayesiano

El mejor clasificador (en términos de la probabilidad de error) es:

$$\mathbb{P}(Y \neq \varphi(X)) \geq 1 - \mathbb{E} \left[\max_y P_{Y|X}(y|X) \right]$$

Clasificador al azar para k clases

Cualquier clasificador razonable debe ganarle a la decisión al azar:

$$\mathbb{P}(Y \neq \varphi(X)) \leq 1 - \frac{1}{k}$$

Clasificador dummy

Otro clasificador muy precario (pero mejor que el azaroso) es elegir siempre la clase más probable. La probabilidad de error del dummy es:

$$\mathbb{P}(Y \neq \varphi(X)) \leq 1 - \max_y P_Y(y)$$

Clasificador bayesiano

Interpretación Gráfica

$$1 - \mathbb{E} \left[\max_y P_{Y|X}(y|X) \right] = \sum_{y \in \mathcal{Y}} P_Y(y) \mathbb{P}(X \notin \mathcal{R}_y | Y = y)$$

donde \mathcal{R}_y es el conjunto de $x \in \mathcal{X}$ donde y es el máximo de $P_{Y|X=x}(y)$:

$$\mathcal{R}_y = \left\{ x \in \mathcal{X} : P_{Y|X=x}(y) = \max_{y' \in \mathcal{Y}} P_{Y|X=x}(y') \right\}$$

Para los $x \in \mathcal{X}$ donde haya dos o más máximos de $P_{Y|X=x}(y)$, se asigna dicho x a solo una de dichas \mathcal{R}_y elegida arbitrariamente.

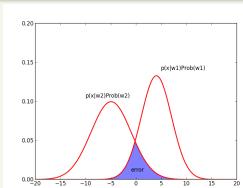
Clasificador bayesiano

Interpretación Gráfica

$$1 - \mathbb{E} \left[\max_y P_{Y|X}(y|X) \right] = \sum_{y \in \mathcal{Y}} P_Y(y) \mathbb{P}(X \notin \mathcal{R}_y | Y = y)$$

donde \mathcal{R}_y es el conjunto de $x \in \mathcal{X}$ donde y es el máximo de $P_{Y|X=x}(y)$:

$$\mathcal{R}_y = \left\{ x \in \mathcal{X} : P_{Y|X=x}(y) = \max_{y' \in \mathcal{Y}} P_{Y|X=x}(y') \right\}$$



Para los $x \in \mathcal{X}$ donde haya dos o más máximos de $P_{Y|X=x}(y)$, se asigna dicho x a solo una de dichas \mathcal{R}_y elegida arbitrariamente.

Clasificador bayesiano

Objetivo

Quiero buscar $\varphi(\cdot)$ que minimice $\mathbb{P}(Y \neq \varphi(X))$. Es decir aprender el “clasificador bayesiano”: $\varphi(x) = \arg \max_y P_{Y|X}(y|x)$.

Clasificador bayesiano

Objetivo

Quiero buscar $\varphi(\cdot)$ que minimice $\mathbb{P}(Y \neq \varphi(X))$. Es decir aprender el “clasificador bayesiano”: $\varphi(x) = \arg \max_y P_{Y|X}(y|x)$.

Problemas numéricos

La propuesta de buscar $\varphi(\cdot)$ que minimice el riesgo empírico: $\frac{1}{n} \sum_{i=1}^n \mathbb{1}\{Y_i \neq \varphi(X_i)\}$ suele tener problemas numéricos (no derivable).

Clasificador bayesiano

Objetivo

Quiero buscar $\varphi(\cdot)$ que minimice $\mathbb{P}(Y \neq \varphi(X))$. Es decir aprender el “clasificador bayesiano”: $\varphi(x) = \arg \max_y P_{Y|X}(y|x)$.

Problemas numéricos

La propuesta de buscar $\varphi(\cdot)$ que minimice el riesgo empírico: $\frac{1}{n} \sum_{i=1}^n \mathbb{1}\{Y_i \neq \varphi(X_i)\}$ suele tener problemas numéricos (no derivable).

Possible solución

El clasificador bayesiano se aprenderá en dos etapas:

- Aprender toda $P_{Y|X}(y|x)$.
- Quedarse con el máximo.

Elementos de Teoría de Información

- $H(X) = \mathbb{E}[-\log P_X(X)]$ Entropía
- $h(X) = \mathbb{E}[-\log p_X(X)]$ Entropía diferencial
- $H(Y|X) = \mathbb{E}[-\log P_{Y|X}(Y|X)]$ Entropía condicional
- $h(Y|X) = \mathbb{E}[-\log p_{Y|X}(Y|X)]$ Entropía diferencial condicional
- $KL(p_X \| q_X) = \mathbb{E}_{p_X} \left[\log \left(\frac{p_X(X)}{q_X(X)} \right) \right]$ Divergencia de Kullback Leibler
- $I(X; Y) = KL(p_{XY} \| p_X p_Y)$ Información Mutua

Elementos de Teoría de Información

- $H(X) = \mathbb{E}[-\log P_X(X)]$ Entropía
- $h(X) = \mathbb{E}[-\log p_X(X)]$ Entropía diferencial
- $H(Y|X) = \mathbb{E}[-\log P_{Y|X}(Y|X)]$ Entropía condicional
- $h(Y|X) = \mathbb{E}[-\log p_{Y|X}(Y|X)]$ Entropía diferencial condicional
- $KL(p_X \| q_X) = \mathbb{E}_{p_X} \left[\log \left(\frac{p_X(X)}{q_X(X)} \right) \right]$ Divergencia de Kullback Leibler
- $I(X; Y) = KL(p_{XY} \| p_X p_Y)$ Información Mutua

Teorema

$$KL(P \| Q) \geq 0$$

con igualdad si y solo si $P(y) = Q(y)$ para todo $y \in \mathcal{Y}$.
(Hint: $\log(x) \leq x - 1$).

Divergencia de Kullback Leibler

Propuesta inicial

Busco $\hat{P}(y|x)$ que minimice:

$$\underbrace{\mathbb{E} \left[KL \left(P_{Y|X}(\cdot|X) \| \hat{P}(\cdot|X) \right) \right]}_{\text{Kullback Leibler}} = \underbrace{\mathbb{E} \left[-\log \hat{P}(Y|X) \right]}_{\text{Cross-entropy}} - \underbrace{H(Y|X)}_{\text{Entropía condicional}}$$

Divergencia de Kullback Leibler

Propuesta inicial

Busco $\hat{P}(y|x)$ que minimice:

$$\underbrace{\mathbb{E} \left[KL \left(P_{Y|X}(\cdot|X) \| \hat{P}(\cdot|X) \right) \right]}_{\text{Kullback Leibler}} = \underbrace{\mathbb{E} \left[-\log \hat{P}(Y|X) \right]}_{\text{Cross-entropy}} - \underbrace{H(Y|X)}_{\text{Entropía condicional}}$$

Optimalidad para $\ell(x, y) = -\log \hat{P}(y|x)$

$$\mathbb{E} \left[-\log \hat{P}(Y|X) \right] \geq H(Y|X)$$

son igualdad si y solo si $\hat{P}(y|x) = P_{Y|X}(y|x)$ para todo (x, y) .

Divergencia de Kullback Leibler

Propuesta inicial

Busco $\hat{P}(y|x)$ que minimice:

$$\underbrace{\mathbb{E} \left[KL \left(P_{Y|X}(\cdot|X) \| \hat{P}(\cdot|X) \right) \right]}_{\text{Kullback Leibler}} = \underbrace{\mathbb{E} \left[-\log \hat{P}(Y|X) \right]}_{\text{Cross-entropy}} - \underbrace{H(Y|X)}_{\text{Entropía condicional}}$$

Optimalidad para $\ell(x, y) = -\log \hat{P}(y|x)$

$$\mathbb{E} \left[-\log \hat{P}(Y|X) \right] \geq H(Y|X)$$

son igualdad si y solo si $\hat{P}(y|x) = P_{Y|X}(y|x)$ para todo (x, y) .

Mismatch de métricas

El mínimo de la cross entropy no tiene por qué coincidir exactamente con el mínimo de la probabilidad de error. En general se mira la cross entropy para reducir el bias y la probabilidad de error para prevenir el overfitting.

TPS-IIA

Matías Vera

Clasificación

8 / 52

TPS-IIA

Matías Vera

Clasificación

9 / 52

Outline

1 Introducción al problema de clasificación

2 Regresión Logística Binaria

3 Regresión Logística Categórica

4 Linear Discriminant Analysis

5 K-Vecinos más cercanos

6 Support Vector Machines

7 Árboles de decisión

TPS-IIA

Matías Vera

Clasificación

10 / 52

TPS-IIA

Matías Vera

Clasificación

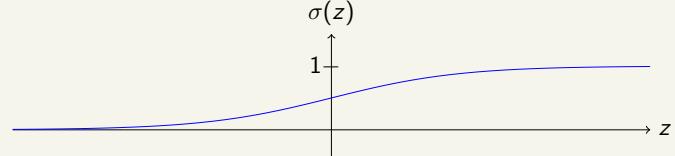
11 / 52

Regresión Logística Binaria

Función Sigmoide

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

- $\sigma(z)$ representa probabilidades.
- z recibe el nombre de *logit*.

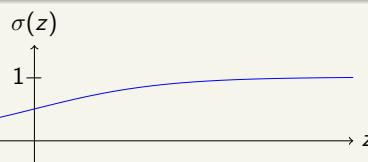


Regresión Logística Binaria

Función Sigmoide

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

- $\sigma(z)$ representa probabilidades.
- z recibe el nombre de *logit*.



Propuesta

$$\begin{aligned}\hat{P}(1|x) &= \sigma(w^T x + b) \\ \hat{P}(0|x) &= 1 - \sigma(w^T x + b)\end{aligned}$$

TPS-IIA

Matías Vera

Clasificación

11 / 52

Hard/Soft Decision

Decisión Suave

Sea $x \in \mathbb{R}^d$, llamamos predicción *soft* de un algoritmo a la predicción de las probabilidades estimadas $\hat{P}(\cdot|x)$. Esta estimación es un vector de probabilidades de todas las clases posibles (no negativas y suman 1). Su desempeño se suele medir con la entropía cruzada $\mathbb{E}[-\log \hat{P}(Y|X)]$.

Decisión Dura

Sea $x \in \mathbb{R}^d$, llamamos predicción *hard* de un algoritmo a la predicción final de la clase estimada $\varphi(x)$. Es decir, es una estimación del valor de Y . Generalmente se la suele definir a partir de la predicción *soft* como:

$$\varphi(x) = \arg \max_{y \in \mathcal{Y}} \hat{P}(y|x)$$

Se desempeño se suele medir con la probabilidad de acierto $\mathbb{P}(Y = \varphi(X))$.

TPS-IIA

Matías Vera

Clasificación

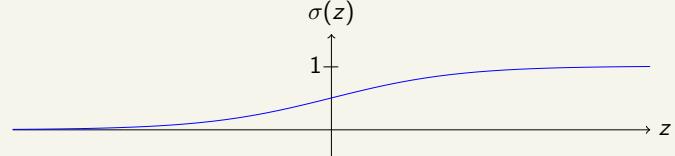
9 / 52

Regresión Logística Binaria

Función Sigmoide

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

- $\sigma(z)$ representa probabilidades.
- z recibe el nombre de *logit*.



Regresión Logística Binaria

Riesgo empírico

$$\begin{aligned}\frac{1}{n} \sum_{i=1}^n \ell(X_i, Y_i) &= \\ &- \frac{1}{n} \sum_{i=1}^n Y_i \log (\sigma(w^T X_i + b)) + (1 - Y_i) \log (1 - \sigma(w^T X_i + b))\end{aligned}$$

TPS-IIA

Matías Vera

Clasificación

12 / 52

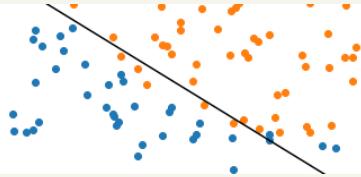
Regresión Logística Binaria

Riesgo empírico

$$\frac{1}{n} \sum_{i=1}^n \ell(X_i, Y_i) = -\frac{1}{n} \sum_{i=1}^n Y_i \log \left(\sigma(w^T X_i + b) \right) + (1 - Y_i) \log \left(1 - \sigma(w^T X_i + b) \right)$$

Elección del máximo

$$\hat{P}(1|x) \leq \hat{P}(0|x) \Leftrightarrow w^T x + b \leq 0$$



TPS-IIA

Matías Vera

Clasificación

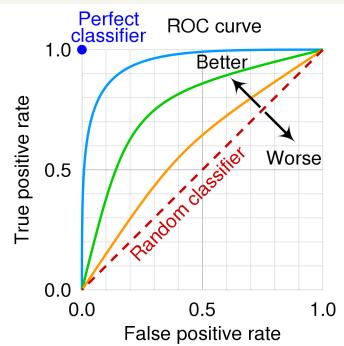
12 / 52

Curvas ROC

Agregar un umbral

Puedo darle más peso a una clase:

$$w^T x + b \leq t$$



$$TPR = \mathbb{P}(Y = \phi(X)|Y = 1)$$

$$FPR = \mathbb{P}(Y \neq \phi(X)|Y = 0)$$

Area Under the Curve (AUC)

El AUC es el área bajo la curva ROC.

Equal Error Rate (EER)

El EER es el error para el cuál los errores FPR = 1 - TPR.

TPS-IIA

Matías Vera

Clasificación

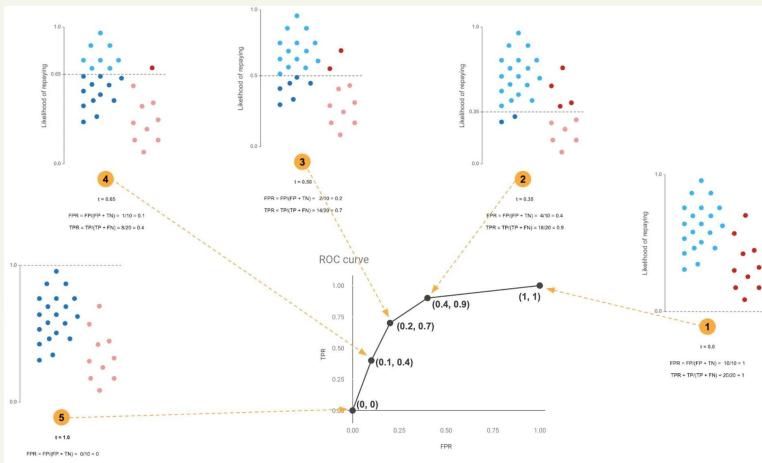
TPS-IIA

Matías Vera

Clasificación

13 / 52

Curvas ROC



TPS-IIA

Matías Vera

Clasificación

14 / 52

Métricas para clases desbalanceadas

Precision-Recall

Cuando el costo de las clases está desbalanceado se utilizan las métricas *Precision* y *Recall*.

- Precision = $\mathbb{P}(Y = \phi(X)|\phi(X) = 1)$. *Precision* se utiliza cuando los falsos positivos tiene consecuencias graves. Por ejemplo, diagnosticar erróneamente una enfermedad a una persona sana
- Recall (TPR) = $\mathbb{P}(Y = \phi(X)|Y = 1)$. *Recall* se utiliza cuando los falsos negativos tiene consecuencias graves. Por ejemplo, en la detección de fraudes, no detectar una transacción fraudulenta.

TPS-IIA

Matías Vera

Clasificación

15 / 52

Métricas para clases desbalanceadas

Precision-Recall

Cuando el costo de las clases está desbalanceado se utilizan las métricas *Precision* y *Recall*.

- Precision = $\mathbb{P}(Y = \phi(X)|\phi(X) = 1)$. *Precision* se utiliza cuando los falsos positivos tiene consecuencias graves. Por ejemplo, diagnosticar erróneamente una enfermedad a una persona sana
- Recall (TPR) = $\mathbb{P}(Y = \phi(X)|Y = 1)$. *Recall* se utiliza cuando los falsos negativos tiene consecuencias graves. Por ejemplo, en la detección de fraudes, no detectar una transacción fraudulenta.

F1-score

Cuando la proporción de las clases está desbalanceada se utiliza la métrica F1:

$$F_1 = 2 \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Outline

- 1 Introducción al problema de clasificación
- 2 Regresión Logística Binaria
- 3 Regresión Logística Categórica
- 4 Linear Discriminant Analysis
- 5 K-Vecinos más cercanos
- 6 Support Vector Machines
- 7 Árboles de decisión

TPS-IIA

Matías Vera

Clasificación

15 / 52

TPS-IIA

Matías Vera

Clasificación

16 / 52

Regresión Logística Categórica (k clases)

Regresión logística clásica

$$\hat{P}(y|x) = \begin{cases} \frac{e^{w_y^T x + b_y}}{1 + \sum_{j=1}^{k-1} e^{w_j^T x + b_j}} & y \in \{1, \dots, k-1\} \\ \frac{1}{1 + \sum_{j=1}^{k-1} e^{w_j^T x + b_j}} & y = k \end{cases}$$

Regresión Logística Categórica (k clases)

Regresión logística clásica

$$\hat{P}(y|x) = \begin{cases} \frac{e^{w_y^T x + b_y}}{1 + \sum_{j=1}^{k-1} e^{w_j^T x + b_j}} & y \in \{1, \dots, k-1\} \\ \frac{1}{1 + \sum_{j=1}^{k-1} e^{w_j^T x + b_j}} & y = k \end{cases}$$

Softmax

$$\hat{P}(y|x) = \frac{e^{w_y^T x + b_y}}{\sum_{j=1}^k e^{w_j^T x + b_j}}, \quad y \in \{1, \dots, k\}$$

TPS-IIA

Matias Vera

Clasificación

17 / 52

TPS-IIA

Matias Vera

Clasificación

17 / 52

Regresión Logística Categórica (k clases)

Regresión logística clásica

$$\hat{P}(y|x) = \begin{cases} \frac{e^{w_y^T x + b_y}}{1 + \sum_{j=1}^{k-1} e^{w_j^T x + b_j}} & y \in \{1, \dots, k-1\} \\ \frac{1}{1 + \sum_{j=1}^{k-1} e^{w_j^T x + b_j}} & y = k \end{cases}$$

Softmax

$$\hat{P}(y|x) = \frac{e^{w_y^T x + b_y}}{\sum_{j=1}^k e^{w_j^T x + b_j}}, \quad y \in \{1, \dots, k\}$$

Riesgo empírico

$$\frac{1}{n} \sum_{i=1}^n \ell(X_i, Y_i) = \frac{1}{n} \sum_{i=1}^n \left[\log \left(\sum_{j=1}^k e^{w_j^T X_i + b_j} \right) - (w_{Y_i}^T X_i + b_{Y_i}) \right]$$

TPS-IIA

Matias Vera

Clasificación

17 / 52

TPS-IIA

Matias Vera

Clasificación

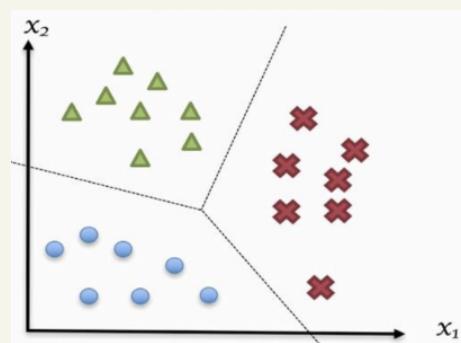
18 / 52

Regresión Softmax

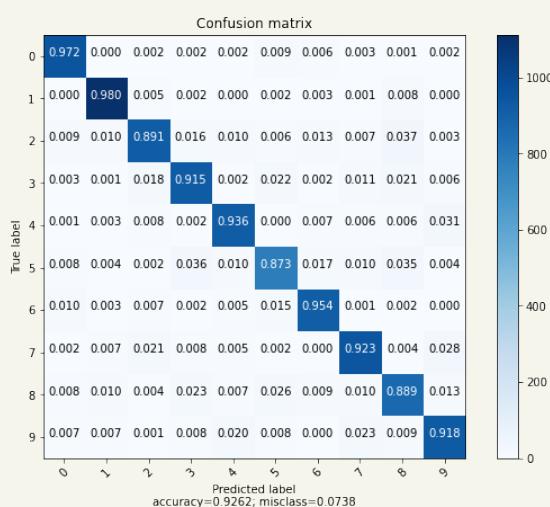
Elección del máximo

$$\arg \max_y \hat{P}(y|x) = \arg \max_y w_y^T x + b_y$$

Se separa con hiperplanos!



Confusion Matrix



Generalización del F1 score

		Predicted		
		Airplane	Boat	Car
Actual	Airplane	2	1	0
	Boat	0	1	0
		Airplane Boat Car		
		2	1	0

TPS-IIA

Matias Vera

Clasificación

19 / 52

TPS-IIA

Matias Vera

Clasificación

20 / 52

Generalización del F1 score

		Predicted		
		Airplane	Boat	Car
Actual	Airplane	2	1	0
	Boat	0	1	0
	Car	1	2	3

Label	True Positive (TP)	False Positive (FP)	False Negative (FN)	Precision	Recall	F1 Score
Airplane	2	1	1	0.67	0.67	$2 * (0.67 * 0.67) / (0.67 + 0.67) = \mathbf{0.67}$
Boat	1	3	0	0.25	1.00	$2 * (0.25 * 1.00) / (0.25 + 1.00) = \mathbf{0.40}$
Car	3	0	3	1.00	0.50	$2 * (1.00 * 0.50) / (1.00 + 0.50) = \mathbf{0.67}$

Generalización del F1 score

		Predicted		
		Airplane	Boat	Car
Actual	Airplane	2	1	0
	Boat	0	1	0
	Car	1	2	3

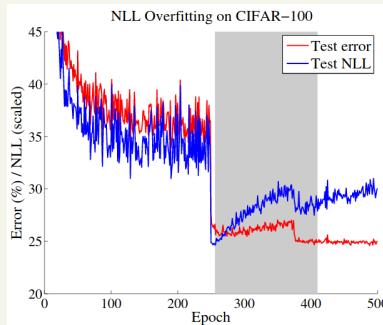
Label	True Positive (TP)	False Positive (FP)	False Negative (FN)	Precision	Recall	F1 Score
Airplane	2	1	1	0.67	0.67	$2 * (0.67 * 0.67) / (0.67 + 0.67) = \mathbf{0.67}$
Boat	1	3	0	0.25	1.00	$2 * (0.25 * 1.00) / (0.25 + 1.00) = \mathbf{0.40}$
Car	3	0	3	1.00	0.50	$2 * (1.00 * 0.50) / (1.00 + 0.50) = \mathbf{0.67}$

$$\text{Macro-}F_1 = \frac{0.67 + 0.40 + 0.67}{3} = 0.58$$

Calibración

Si valido hiperparámetros con respecto a la probabilidad de error, ¿la salida siguen siendo probabilidades?

La concentración de probabilidades natural del softmax es buena para acercarme al clasificador bayesiano, pero puede descalibrar la interpretación probabilística de \hat{P} .



Guo et al. 2017: "On Calibration of Modern Neural Networks".

no vamos a ver
Calibración calibracion en el curso
pero es datazo

Temperature Scaling

Se soluciona con la inclusión de un nuevo parámetro (o hiper) $T > 0$:

$$\hat{P}(y|x) = \frac{e^{z(x)_y/T}}{\sum e^{z(x)_i/T}}$$

esto es la
sumatoria,, es otra
forma de escribirlo
mas resumido. la T
es transpuesta

Es importante mirar la cross-entropy en la etapa de validación!

cuando hago la predicción
obtengo la hard pero si quiero
calibrar tengo que tener en
cuenta lo soft

Dataset	Model	Uncalibrated	Temp. Scaling
Birds	ResNet 50	0.9786	0.9902
Cifar	ResNet 30	0.5088	0.5111
CIFAR-10	ResNet 110	0.3285	0.2102
CIFAR-10	ResNet 110 (SD)	0.2959	0.1718
CIFAR-10	Wide ResNet 32	0.3293	0.2283
CIFAR-10	DenseNet 40	0.2228	0.1750
CIFAR-10	LeNet 5	0.4688	0.459
CIFAR-100	ResNet 110	1.4978	1.0442
CIFAR-100	ResNet 110 (SD)	1.1157	0.8613
CIFAR-100	Wide ResNet 32	1.3434	1.0565
CIFAR-100	DenseNet 40	1.0134	0.9026
CIFAR-100	LeNet 5	1.6639	1.6560
ImageNet	DenseNet 161	0.9338	0.8885
ImageNet	DenseNet 152	0.8961	0.8657
SVHN	ResNet 152 (SD)	0.0842	0.0821
20 News	DAN 3	0.7949	0.7387
Reuters	DAN 3	0.102	0.0994
SST Binary	TreeLSTM	0.3367	0.2739
SST Fine Grained	TreeLSTM	1.1475	1.1168

Guo et al. 2017: "On Calibration of Modern Neural Networks".

CLASE 4 DE ABRIL!!!

Outline

1 Introducción al problema de clasificación

2 Regresión Logística Binaria

3 Regresión Logística Categórica

4 Linear Discriminant Analysis

5 K-Vectinos más cercanos

6 Support Vector Machines

7 Árboles de decisión

Modelos Discriminativos y Generativos

Clasificación de Algoritmos

- **Modelos Discriminativos:** Modelan la dist. condicional $\hat{P}(y|x)$.
- **Modelos Generativos:** Modelan la dist. conjunta $\hat{P}(x,y)$.

Los modelos generativos permiten generar datos sintéticos!

Modelos Discriminativos y Generativos

Clasificación de Algoritmos

- Modelos Discriminativos:** Modelan la dist. condicional $\hat{P}(y|x)$.
- Modelos Generativos:** Modelan la dist. conjunta $\hat{P}(x,y)$.

Los modelos generativos permiten generar datos sintéticos!

Linear Discriminant Analysis (LDA)

$$Y \sim \text{Cat}(\{c_1, \dots, c_K\}), \quad X|Y = k \sim \mathcal{N}(\mu_k, \Sigma)$$



TPS-IIA

Matias Vera

Clasificación

24 / 52

TPS-IIA

Matias Vera

Clasificación

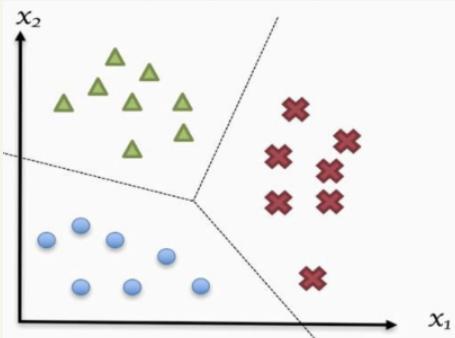
25 / 52

Regresión Softmax

Elección del máximo

$$\arg \max_y \hat{P}(y|x) = \arg \max_y w_y^T x + b_y$$

Se separa con hiperplanos!



TPS-IIA

Matias Vera

Clasificación

26 / 52

Estimación Insesgada de Parámetros

Estimadores

$$\mathcal{D}_k = \{x_i : 1 \leq i \leq n \wedge y_i = k\}$$

$$c_k = \frac{\#(\mathcal{D}_k)}{n}$$

$$\mu_k = \frac{1}{\#(\mathcal{D}_k)} \sum_{x \in \mathcal{D}_k} x$$

$$\Sigma_k = \frac{1}{\#(\mathcal{D}_k) - 1} \sum_{x \in \mathcal{D}_k} (x - \mu_k)(x - \mu_k)^T$$

$$\Sigma = \frac{1}{n - K} \sum_{k=1}^K (\#(\mathcal{D}_k) - 1) \Sigma_k$$

Quadratic Discriminant Analysis

Quadratic Discriminant Analysis (QDA)

$$Y \sim \text{Cat}(\{c_1, \dots, c_K\}), \quad X|Y = k \sim \mathcal{N}(\mu_k, \Sigma_k)$$

Quadratic Discriminant Analysis

Quadratic Discriminant Analysis (QDA)

$$Y \sim \text{Cat}(\{c_1, \dots, c_K\}), \quad X|Y = k \sim \mathcal{N}(\mu_k, \Sigma_k)$$

Expresiones Matemáticas

$$\hat{p}(x) = \sum_{k=1}^K c_k \frac{e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}}{(2\pi)^{d_x/2} |\Sigma_k|^{1/2}}$$

$$\hat{P}(y|x) = \frac{e^{-\frac{1}{2}(x-\mu_y)^T \Sigma_y^{-1}(x-\mu_y) + \log(c_y) - \frac{\log|\Sigma_y|}{2}}}{\sum_{k=1}^K e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k) + \log(c_k) - \frac{\log|\Sigma_k|}{2}}}$$

TPS-IIA

Matias Vera

Clasificación

28 / 52

TPS-IIA

Matias Vera

Clasificación

28 / 52

Quadratic Discriminant Analysis

Quadratic Discriminant Analysis (QDA)

$$Y \sim \text{Cat}(\{c_1, \dots, c_K\}), \quad X|Y = k \sim \mathcal{N}(\mu_k, \Sigma_k)$$

Expresiones Matemáticas

$$\hat{p}(x) = \sum_{k=1}^K c_k \frac{e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}}{(2\pi)^{d_x/2} |\Sigma_k|^{1/2}}$$

$$\hat{P}(y|x) = \frac{e^{-\frac{1}{2}(x-\mu_y)^T \Sigma_y^{-1}(x-\mu_y)} + \log(c_y) - \frac{\log|\Sigma_y|}{2}}{\sum_{k=1}^K e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)} + \log(c_k) - \frac{\log|\Sigma_k|}{2}}$$

Elección del máximo: NO ES LINEAL, ES CUADRÁTICO

$$\arg \max_y -\frac{1}{2}(x - \mu_y)^T \Sigma_y^{-1}(x - \mu_y) + \log(c_y) - \frac{\log|\Sigma_y|}{2}$$

Outline

- 1 Introducción al problema de clasificación
- 2 Regresión Logística Binaria
- 3 Regresión Logística Categórica
- 4 Linear Discriminant Analysis
- 5 K-Vectinos más cercanos
- 6 Support Vector Machines
- 7 Árboles de decisión

Modelos Paramétricos y No Paramétricos

Clasificación de Algoritmos

- **Modelos Paramétricos:** Asumen conocimiento parcial sobre la distribución, indexándola por parámetros.
- **Modelos No Paramétricos:** No se asume una estructura a priori para la distribución.

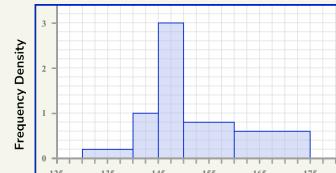
Modelos Paramétricos y No Paramétricos

Clasificación de Algoritmos

- **Modelos Paramétricos:** Asumen conocimiento parcial sobre la distribución, indexándola por parámetros.
- **Modelos No Paramétricos:** No se asume una estructura a priori para la distribución.

Histograma

El histograma asume una densidad constante por regiones. En cada región asigna $\hat{p}(x) = \frac{K}{n \cdot V}$ donde n es la cantidad de muestras totales, K la cantidad de muestras en dicha región y V el volumen de la región.



K-Vecinos más cercanos (KNN)

Adaptando el concepto a aprendizaje supervisado

Asumiendo que $\hat{P}(y) = \frac{N_y}{n}$ con N_y el número de muestras de la clase y , y que (en cada región) $\hat{p}(x|y) = \frac{K_y}{N_y \cdot V}$ con K_y la cantidad de muestras que caen en la región de la clase y , se obtiene:

$$\hat{P}(y|x) = \frac{\hat{p}(x|y)\hat{P}(y)}{\sum_{i=1}^K \hat{p}(x|i)\hat{P}(i)} = \frac{K_y}{K}$$

Es decir, la proporción de muestras de la clase y en la región.

K-Vecinos más cercanos (KNN)

Adaptando el concepto a aprendizaje supervisado

Asumiendo que $\hat{P}(y) = \frac{N_y}{n}$ con N_y el número de muestras de la clase y , y que (en cada región) $\hat{p}(x|y) = \frac{K_y}{N_y \cdot V}$ con K_y la cantidad de muestras que caen en la región de la clase y , se obtiene:

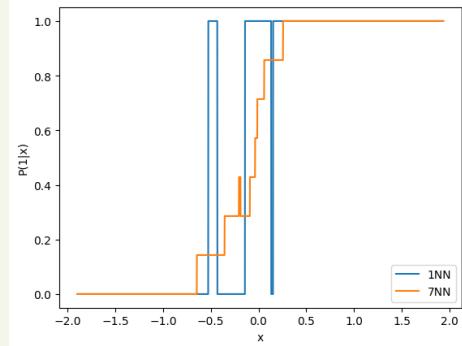
$$\hat{P}(y|x) = \frac{\hat{p}(x|y)\hat{P}(y)}{\sum_{i=1}^K \hat{p}(x|i)\hat{P}(i)} = \frac{K_y}{K}$$

Es decir, la proporción de muestras de la clase y en la región.

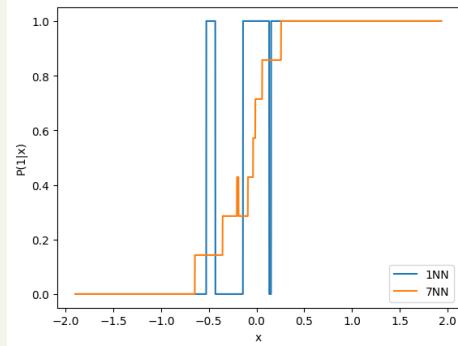
K-Vecinos más cercanos

KNN fija el valor de vecinos K y en base a esto define las regiones. Por ejemplo, la región utilizada para computar un *feature* x es la región centrada en x que posee K muestras (las K más cercanas a x).

K-Vecinos más cercanos (KNN)



K-Vecinos más cercanos (KNN)



Elección del máximo

Notar que para quedarse con el máximo de $\hat{P}(y|x)$ no hace falta computarla. Simplemente se clasifica según sus K vecinos más cercanos, por mayoría.

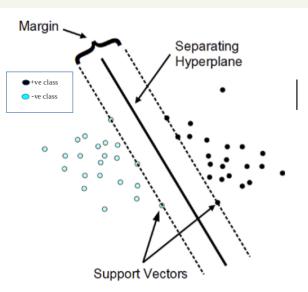
Outline

- 1 Introducción al problema de clasificación
- 2 Regresión Logística Binaria
- 3 Regresión Logística Categórica
- 4 Linear Discriminant Analysis
- 5 K-Vecinos más cercanos
- 6 Support Vector Machines
- 7 Árboles de decisión

Support Vector Machines

Clases linealmente separables

Sea la clasificación binaria $y \in \{-1, 1\}$ y $z(x) = w^T \cdot x + b = 0$ su frontera de decisión. Decimos que las clases son linealmente separables, si existen w y b tales que $y \cdot z(x) > 0$ para todo $(x, y) \in \mathcal{D}_n$ (set de entrenamiento). Llamamos $f_i(w, b) = y_i z(x_i) > 0$ con $1 \leq i \leq n$.



- w es ortogonal a la frontera y por lo tanto $w/(x - x_*)$ con x_* la proyección ortogonal de x sobre la frontera.

$$|w^T(x - x_*)| = \|w\| \|x - x_*\|$$

- Dado que x_* está sobre la frontera, $w^T(x - x_*) = z(x)$ y por lo tanto:

$$d(x_i) = \|x_i - x_*\| = \frac{|z(x_i)|}{\|w\|} = \frac{y_i \cdot z(x_i)}{\|w\|}$$

Support Vector Machines

Clases linealmente separables

Sea la clasificación binaria $y \in \{-1, 1\}$ y $z(x) = w^T \cdot x + b = 0$ su frontera de decisión. Decimos que las clases son linealmente separables, si existen w y b tales que $y \cdot z(x) > 0$ para todo $(x, y) \in \mathcal{D}_n$ (set de entrenamiento). Llamamos $f_i(w, b) = y_i z(x_i) > 0$ con $1 \leq i \leq n$.

Support Vector Machines

Margen

Se define el margen unilateral como criterio de peor caso:

$$m(w, b) = \min_{1 \leq i \leq n} \frac{y_i(w^T \cdot x_i + b)}{\|w\|} = \frac{1}{\|w\|} \min_{1 \leq i \leq n} f_i(w, b) = \frac{f_k(w, b)}{\|w\|}$$

con k un índice óptimo (función de w y b). Por lo tanto, el problema a resolver es maximizar el margen: $\max_{w, b} m(w, b)$ st. $f_i(w, b) > 0$ para $i = 1, \dots, n$.

Support Vector Machines

Margen

Se define el margen unilateral como criterio de peor caso:

$$m(w, b) = \min_{1 \leq i \leq n} \frac{y_i(w^T \cdot x_i + b)}{\|w\|} = \frac{1}{\|w\|} \min_{1 \leq i \leq n} f_i(w, b) = \frac{f_k(w, b)}{\|w\|}$$

con k un índice óptimo (función de w y b). Por lo tanto, el problema a resolver es maximizar el margen: $\max_{w, b} m(w, b)$ st. $f_i(w, b) > 0$ para $i = 1, \dots, n$.

Escala

Sea $\alpha > 0$, está claro la decisión $z(x) \geq 0$ no se ve afectada si reescalamos los parámetros $w \leftarrow \alpha w$ y $b \leftarrow \alpha b$. Esto mismo ocurre con el margen $m(\alpha w, \alpha b) = m(w, b)$. Con lo cuál no se pierde generalidad al asumir $f_k(w, b) = 1$. Luego $m(w, b) = \frac{1}{\|w\|}$ y $f_i(w, b) \geq 1$ para todo $1 \leq i \leq n$.

Las muestras en las que $f_i(w, b) = 1$ se denominan vectores soporte.

Support Vector Machines

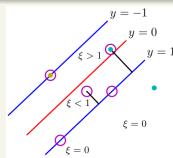
Problema de optimización primal

$$\min_{w, b} \frac{1}{2} \|w\|^2 \quad \text{s.t.} \quad y_i(w^T x_i + b) \geq 1 \quad (\forall 1 \leq i \leq n)$$

Relajando los márgenes

Mitigar problemas con outliers. Sea $C \geq 0$,

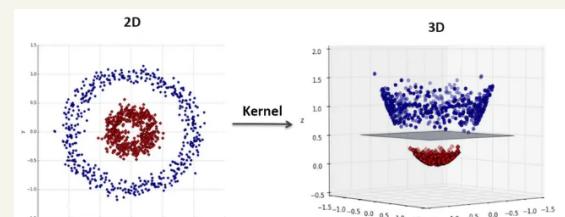
$$\min_{w, b, \xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \quad \text{s.t.} \quad \begin{cases} y_i(w^T x_i + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases} \quad (\forall 1 \leq i \leq n)$$



Support Vector Machines

Generalización a fronteras no lineales

Este método es adaptable a diferentes fronteras $z(x) = w^T \phi(x) + b$. Se puede demostrar, que el resultado final del entrenamiento depende de los predictores a través de $k(x_1, x_2) = \phi^T(x_1)\phi(x_2)$, función que recibe el nombre de kernel. Es por este motivo que se elige el kernel en lugar de $\phi(\cdot)$, siendo el más utilizado en SVM el denominado gaussiano o rbf: $k(x_1, x_2) = e^{-\gamma \|x_1 - x_2\|^2}$.



Support Vector Machines

Generalización a K-clases

- **one-vs-one**: Se toman todas las combinaciones de pares de clases (son $\frac{K(K-1)}{2}$) y se entrena clasificadores binarios. Se clasifica seleccionando a la clase con más votos.
- **one-vs-the-rest**: Se entrena K clasificadores binarios, donde cada uno toma una clase como positiva y el resto como negativa. Se clasifica según $\arg \max_k w_k^T \phi(x) + b_k$.

Generalización a K-clases

- **one-vs-one**: Se toman todas las combinaciones de pares de clases (son $\frac{K(K-1)}{2}$) y se entrena clasificadores binarios. Se clasifica seleccionando a la clase con más votos.
- **one-vs-the-rest**: Se entrena K clasificadores binarios, donde cada uno toma una clase como positiva y el resto como negativa. Se clasifica según $\arg \max_k w_k^T \phi(x) + b_k$.

Generalización a Regresión

$$\min_{w, b} \frac{1}{2} \|w\|^2 \quad \text{s.t.} \quad |w^T x_i + b - y_i| \leq \epsilon \quad (\forall 1 \leq i \leq n)$$

Optimización Convexa

Tomemos el problema básico de SVM. Sea

$$J_1 = \min_{w,b} \frac{1}{2} \|w\|^2 \quad \text{s.t.} \quad y_i(w^T \phi(x_i) + b) \geq 1 \quad (\forall 1 \leq i \leq n)$$

Dicho problema puede reescribirse usando multiplicadores de Lagrange α_i :

$$J_1 = \min_{w,b} \max_{\alpha_i \geq 0} \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i [y_i(w^T \phi(x_i) + b) - 1]$$

Vectores Soportes

Notar que, el multiplicador óptimo (la solución del problema) debe cumplir que $\alpha_i = 0$ para toda muestra que no sea vector soporte. En contraste, para los vectores soporte ocurre que $y_i(w^T \phi(x_i) + b) = 1$.

Llamamos problema dual al problema definido a partir de invertir el mínimo y el máximo:

$$J_2 = \max_{\alpha_i \geq 0} \min_{w,b} \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i [y_i(w^T \phi(x_i) + b) - 1]$$

Optimización Convexa

Teorema: Weak and Strong duality

Para cualquier problema de optimización $J_1 \geq J_2$. En el caso particular del problema de SVM, por ser convexo, se obtiene que $J_1 = J_2$.

Fijo los multiplicadores $\alpha_i \geq 0$, igualamos a cero la derivada respecto de los parámetros para buscar el mínimo:

- $w - \sum_{i=1}^n \alpha_i y_i \phi(x_i) = 0 \rightarrow w = \sum_{i=1}^n \alpha_i y_i \phi(x_i)$

- $-\sum_{i=1}^n \alpha_i y_i = 0 \rightarrow \sum_{i=1}^n \alpha_i y_i = 0$

La suma dentro de J_2 queda como:

$$\begin{aligned} \sum_{i=1}^n \alpha_i [y_i(w^T \phi(x_i) + b) - 1] &= \sum_{i=1}^n \alpha_i y_i w^T \phi(x_i) + \sum_{i=1}^n \alpha_i y_i b - \sum_{i=1}^n \alpha_i \\ &= \|w\|^2 + 0 - \sum_{i=1}^n \alpha_i \end{aligned}$$

Optimización Convexa

La función a optimizar se puede reescribir como

$$\frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i [y_i(w^T \phi(x_i) + b) - 1] = \left(\sum_{i=1}^n \alpha_i \right) - \frac{1}{2} \|w\|^2$$

La norma cuadrática puede vectorizarse como

$$\|w\|^2 = \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j) = \alpha^T Q \alpha$$

donde Q es una matriz de elementos $Q_{i,j} = y_i y_j \phi(x_i)^T \phi(x_j)$ (depende de los predictores a través del kernel). Entonces, el problema se reduce a:

Problema de optimización dual

$$\max_{\alpha} \alpha^T \mathbf{1} - \frac{1}{2} \alpha^T Q \alpha \quad \text{s.t.} \quad \alpha^T y = 0, \alpha_i \geq 0$$

Por cuestiones numéricas, suele traer complicaciones detectar vectores soportes como $\alpha_i > 0$. En la práctica suele compararse $\alpha_i > \epsilon$ con $\epsilon > 0$ un número pequeño.

Optimización Convexa

Bias

Sea \mathcal{S} el conjunto de índices de vectores soporte y N_S la cantidad de elementos de dicho conjunto; luego $y_i(w^T \phi(x_i) + b) = 1 \forall i \in \mathcal{S}$ y $\alpha_i = 0 \forall i \notin \mathcal{S}$. El bias solo depende de los predictores a través del kernel:

$$b = \frac{1}{N_S} \sum_{i \in \mathcal{S}} (y_i - w^T \phi(x_i)) = \frac{1}{N_S} \sum_{i \in \mathcal{S}} \left(y_i - \sum_{j \in \mathcal{S}} \alpha_j y_j \phi(x_j)^T \phi(x_i) \right)$$

Regla de decisión

Una vez entrenado, la regla de decisión para evaluar el signo de $z(x)$. Dicha decisión solo depende de los predictores a través del kernel:

$$z(x) = \sum_{j \in \mathcal{S}} \alpha_j y_j \phi(x_j)^T \phi(x) + b$$

Outline

1 Introducción al problema de clasificación

2 Regresión Logística Binaria

3 Regresión Logística Categórica

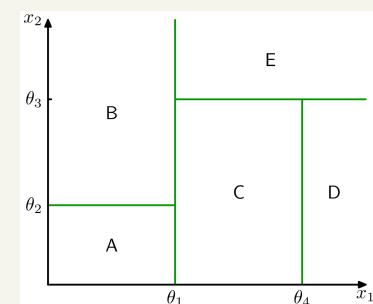
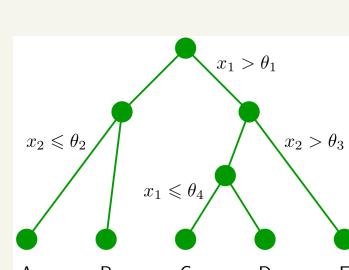
4 Linear Discriminant Analysis

5 K-Vecinos más cercanos

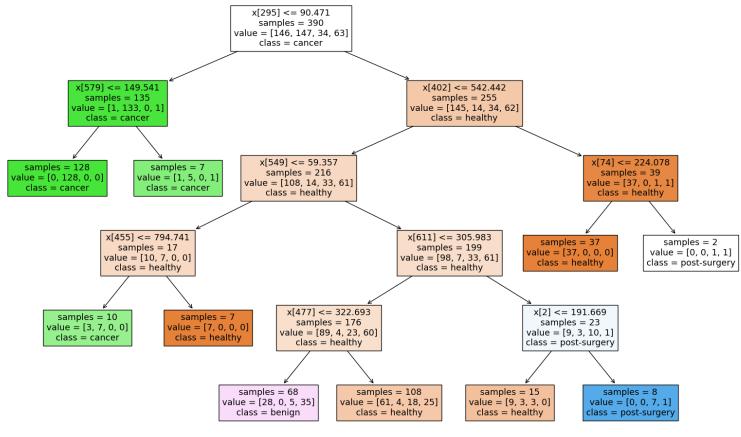
6 Support Vector Machines

7 Árboles de decisión

CART: Classification and Regression Trees



Árboles de decisión



TPS-IIA

Matías Vera

Clasificación

45 / 52

TPS-IIA

Matías Vera

Clasificación

46 / 52

Árboles de decisión

Modelado matemático por nodo

Llamamos:

- Q_m al conjunto de datos en el nodo m .
- $Q_m^L(j_m, t_m) = \{(x, y) \in Q_m : x_{j_m} \leq t_m\}$.
- $Q_m^R(j_m, t_m) = \{(x, y) \in Q_m : x_{j_m} > t_m\}$.
- $H(Q_m)$ a la función impureza del conjunto Q_m .
- $G_m(j_m, t_m) = \frac{|Q_m^L(j_m, t_m)|}{|Q_m|} H(Q_m^L(j_m, t_m)) + \frac{|Q_m^R(j_m, t_m)|}{|Q_m|} H(Q_m^R(j_m, t_m))$.
- Busco para cada nodo $(j_m^*, t_m^*) = \arg \min_{j_m, t_m} G_m(j_m, t_m)$

Funciones impurezas habituales

Sea $p_{m,k}$ la proporción de muestras de la clase k en el nodo m :

- Gini: $H(Q_m) = \sum_k p_{m,k} (1 - p_{m,k})$.
- Entropía: $H(Q_m) = \sum_k -p_{m,k} \log_2(p_{m,k})$.

TPS-IIA

Matías Vera

Clasificación

46 / 52

TPS-IIA

Matías Vera

Clasificación

47 / 52

Árboles de decisión

Condiciones de Parada

- Todas las observaciones tienen la misma etiqueta.
- Si la rama tiene menos de un número preestablecido de observaciones.
- Otras (ver documentación).

Variables Categóricas

Dada la característica binaria de los árboles, estas variables se codifican en estructuras binarias (ej. *one hot encoding*).

Árboles de decisión

Modelado matemático por nodo

Llamamos:

- Q_m al conjunto de datos en el nodo m .
- $Q_m^L(j_m, t_m) = \{(x, y) \in Q_m : x_{j_m} \leq t_m\}$.
- $Q_m^R(j_m, t_m) = \{(x, y) \in Q_m : x_{j_m} > t_m\}$.
- $H(Q_m)$ a la función impureza del conjunto Q_m .
- $G_m(j_m, t_m) = \frac{|Q_m^L(j_m, t_m)|}{|Q_m|} H(Q_m^L(j_m, t_m)) + \frac{|Q_m^R(j_m, t_m)|}{|Q_m|} H(Q_m^R(j_m, t_m))$.
- Busco para cada nodo $(j_m^*, t_m^*) = \arg \min_{j_m, t_m} G_m(j_m, t_m)$

Árboles de decisión

Condiciones de Parada

- Todas las observaciones tienen la misma etiqueta.
- Si la rama tiene menos de un número preestablecido de observaciones.
- Otras (ver documentación).

Árboles de decisión

Condiciones de Parada

- Todas las observaciones tienen la misma etiqueta.
- Si la rama tiene menos de un número preestablecido de observaciones.
- Otras (ver documentación).

Variables Categóricas

Dada la característica binaria de los árboles, estas variables se codifican en estructuras binarias (ej. *one hot encoding*).

Importancia de cada Feature

La *Gini Importance* se define como la disminución total de la impureza del nodo, ponderada por la probabilidad de llegar a ese nodo (normalizado).

TPS-IIA

Matías Vera

Clasificación

47 / 52

TPS-IIA

Matías Vera

Clasificación

47 / 52

Árboles de decisión

Problemas de regresión

Modelando la función regresión como constante por regiones, este método puede ser adaptado. Como función impureza suele usarse el error cuadrático medio:

$$H(Q_m) = \sum_{(x,y) \in Q_m} (y - \bar{y}_m)^2$$

donde \bar{y}_m es el promedio de las y en Q_m .

Árboles de decisión

Problemas de regresión

Modelando la función regresión como constante por regiones, este método puede ser adaptado. Como función impureza suele usarse el error cuadrático medio:

$$H(Q_m) = \sum_{(x,y) \in Q_m} (y - \bar{y}_m)^2$$

donde \bar{y}_m es el promedio de las y en Q_m .

Podado: Regularización

Sea T un árbol determinado (sin condiciones de parado fuertes), $L(T)$ su respectivo conjunto de hojas y α el parámetro de complejidad. Se denomina medida de costo-complejidad a

$$H_\alpha(T) = \sum_{m \in L(T)} \frac{|Q_m|}{n} \cdot H(Q_m) + \alpha \cdot |L(T)|$$

La poda se basa en quedarse con el subárbol de menor costo-complejidad.

TPS-IIA

Matias Vera

Clasificación

48 / 52

TPS-IIA

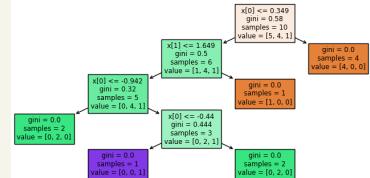
Matias Vera

Clasificación

48 / 52

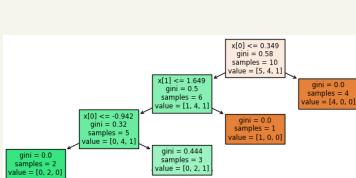
Poda

T_1



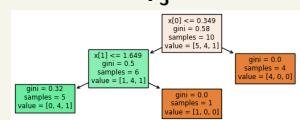
$$H_\alpha(T) = 0 + 5\alpha$$

T_2



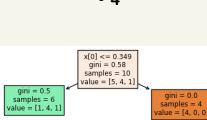
$$H_\alpha(T) = 0.13 + 4\alpha$$

T_3



$$H_\alpha(T) = 0.16 + 3\alpha$$

T_4



$$H_\alpha(T) = 0.3 + 2\alpha$$

T_5



$$H_\alpha(T) = 0.58 + \alpha$$

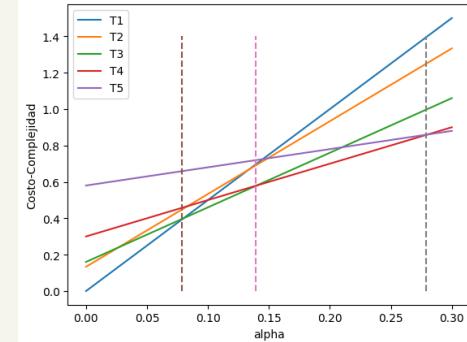
TPS-IIA

Matias Vera

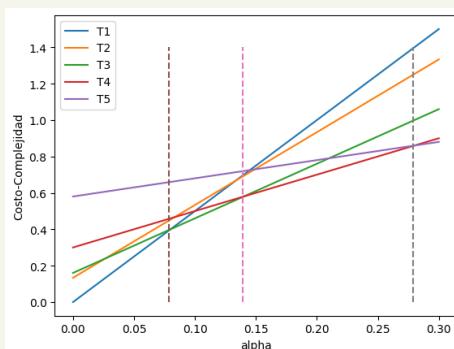
Clasificación

49 / 52

Poda



Poda



- La cantidad de candidatos a óptimos es menor a la cantidad de subárboles (el T_2 nunca es el de menor costo-complejidad).
- El subárbol se elige por validación (típicamente sobre el error de clasificación) comparando todos los casos posibles (en este caso 4 candidatos).

Bosques aleatorios

Bagging

El problema de los árboles de decisión es el *overfitting*. Las condiciones de stop y la poda ayudan a combatirlo, pero muchas veces no son suficiente. Es por eso que surge *Bagging*: Entrenar múltiples algoritmos y decidir por mayoría o promedio (en clasificación o regresión respectivamente). Un algoritmo de múltiples árboles se llama bosque.

¿Por que promediar?

- El promedio mantiene la esperanza y reduce la varianza en muestras i.i.d:

$$\mathbb{E} \left[\frac{1}{B} \sum_{b=1}^B Z_b \right] = \mu, \quad \text{var} \left(\frac{1}{B} \sum_{b=1}^B Z_b \right) = \frac{\sigma^2}{B}$$

- En clasificación, si se piensan etiquetas en codificación *one-hot*, promediar para luego elegir el máximo equivale a elegir la respuesta mayoritaria.

TPS-IIA

Matias Vera

Clasificación

50 / 52

TPS-IIA

Matias Vera

Clasificación

51 / 52

Bosques aleatorios

Se desea entrenar varios algoritmos (de manera que sean variados). Para asegurar ésto, se toman dos decisiones:

No usar todos los features

En lugar de usar todos los d_x features, para asegurar variedad en los árboles, para cada nodo se eligen al azar $\sqrt{d_x}$ features.

Bosques aleatorios

Se desea entrenar varios algoritmos (de manera que sean variados). Para asegurar ésto, se toman dos decisiones:

No usar todos los features

En lugar de usar todos los d_x features, para asegurar variedad en los árboles, para cada nodo se eligen al azar $\sqrt{d_x}$ features.

Bootstrap

Generar B conjuntos de datos diferentes del mismo tamaño que el dataset original n . Para esto, se utiliza una técnica llamada Bootstrap: Se eligen al azar n datos del conjunto *con reposición* y se arma cada conjunto Bootstrap, de manera que la probabilidad que un dato no esté en el conjunto es del $\approx 37\%$:

$$\left(1 - \frac{1}{n}\right)^n \rightarrow e^{-1}$$

Agenda

① Autoencoders

② Principal Components Analysis (PCA)

③ K-Means

④ Algoritmo EM

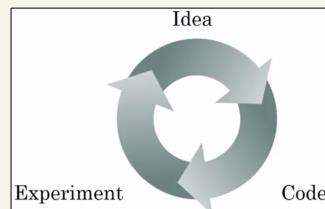
⑤ Factor Analysis

Aprendizaje no Supervisado

Taller de Procesamiento de Señales

Aprendizaje Estadístico

- No se conoce la verdadera estadística.
- Se aprende por medio de datos.
- El buen desempeño no debe limitarse a los datos conocidos.

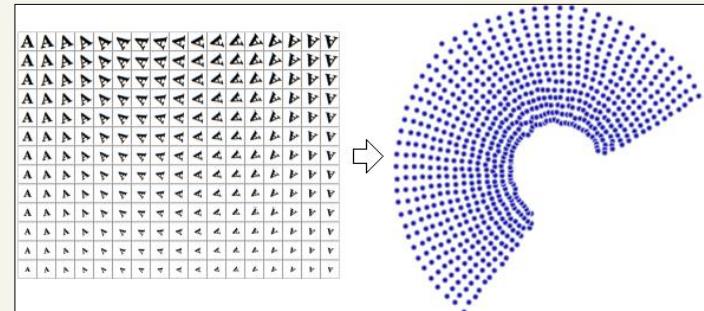


TIPOS DE APRENDIZAJES

- Aprendizaje supervisado: Cuento con pares de datos $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$.
- Aprendizaje no supervisado: Cuento solamente con datos $\{\mathbf{x}^{(i)}\}_{i=1}^n$.
- Aprendizaje semi-supervisado: Cuento con muchos datos no supervisados y unos pocos supervisados.

Manifold

¿Cuál es la dimensión efectiva de los datos?



Manifold

¿Cuál es la dimensión efectiva de los datos?

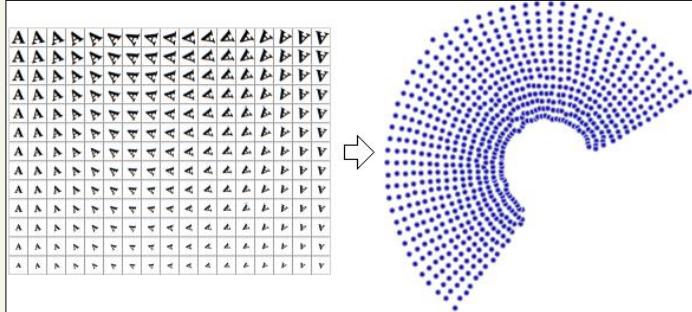
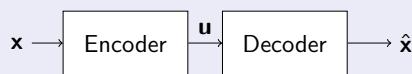


Diagrama en bloques de un Autoencoder



TPS

Matías Vera

No Supervisado

4 / 39

Mathematical Snippets - "An unexpected bijection between the real plane and the real line" <https://www.youtube.com/watch?v=XcMzsF4vDbo>

TPS

Matías Vera

No Supervisado

5 / 39

Manifold

¿Cuál es la dimensión efectiva de los datos?

Objetivo

Hay que entender que el objetivo no es simplemente reconstruir los datos. Sino que es reconstruir los datos a partir de una representación relevante para explicar algún fenómeno o resolver otra tarea. Si no se reconocen patrones en la naturaleza de los datos no hay aprendizaje.

Cuidado!

Existen transformaciones $\mathcal{T} : \mathbb{R}^{d_x} \rightarrow \mathbb{R}$ biyectivas (googlear por ejemplo Teorema de Cantor-Schröder-Bernstein). Pero las representaciones reducidas obtenidas de esta manera pueden no ser interesantes. Hay que tener en cuenta la precisión del computo y, sobre todo, la aplicación en la que se va a utilizar.

Mathematical Snippets - "An unexpected bijection between the real plane and the real line" <https://www.youtube.com/watch?v=XcMzsF4vDbo>

TPS

Matías Vera

No Supervisado

5 / 39

Manifold

Regularización de autoencoders

Bajo ECM para cualquier tipo de entrada



Bajo ECM para los sets de entrenamiento y testeо



Bajo ECM solamente en el set de entrenamiento

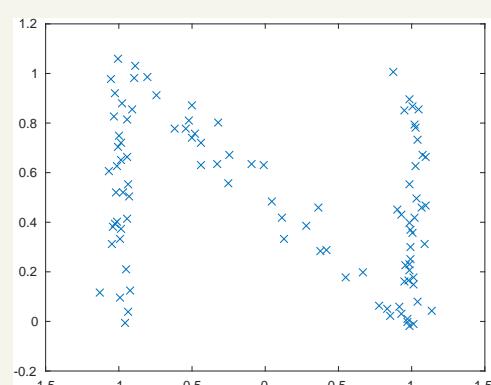


Objetivo

No quiero memorizar el conjunto de datos ni aprender una transformación biyectiva: Busco aprender el manifold. La regularización en un autoencoder busca balancear estos conceptos.

Manifold

Regularización de autoencoders



TPS

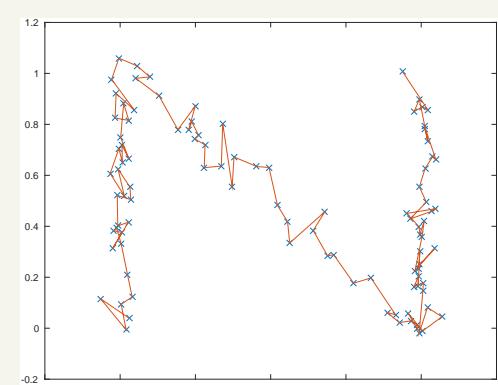
Matías Vera

No Supervisado

7 / 39

Manifold

Regularización de autoencoders



OVERFITTING

No hay aprendizaje, se están memorizando las muestras.

Necesito regularización

TPS

Matías Vera

No Supervisado

7 / 39

TPS

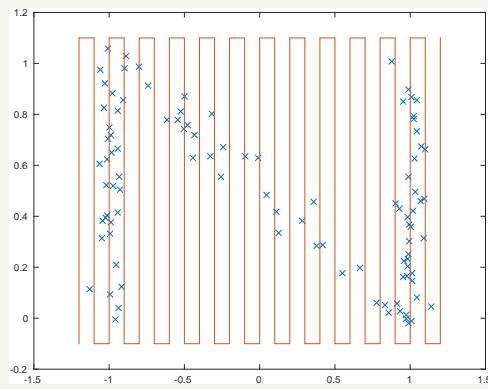
Matías Vera

No Supervisado

7 / 39

Manifold

Regularización de autoencoders



IDENTIDAD

Se está aprendiendo la función identidad y no la naturaleza de los datos.



Necesito regularización

TPS

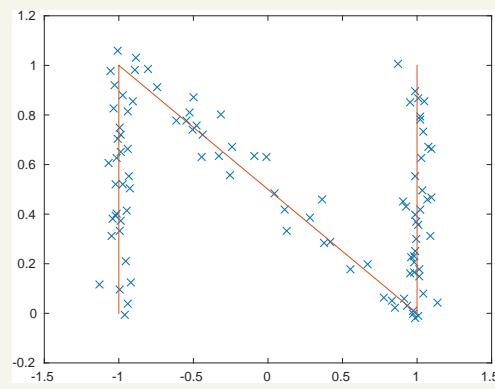
Matías Vera

No Supervisado

7 / 39

Manifold

Regularización de autoencoders



TPS

Matías Vera

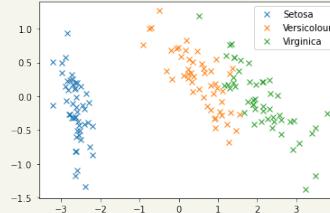
No Supervisado

7 / 39

Algunas Aplicaciones

- Para efectuar una inferencia más precisa
- Para pre-procesar los datos
- Para detectar anomalías

Inferencia



Visualizar en un gráfico 2d o 3d para explicar algunos fenómenos (iris dataset)

TPS

Matías Vera

No Supervisado

8 / 39

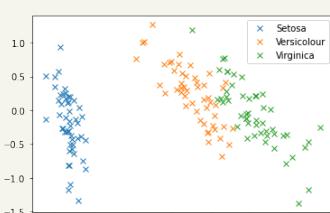
TPS

Matías Vera

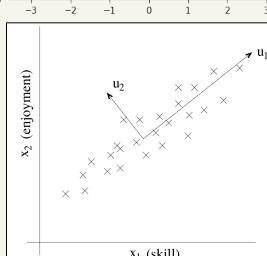
No Supervisado

9 / 39

Inferencia



Visualizar en un gráfico 2d o 3d para explicar algunos fenómenos (iris dataset)

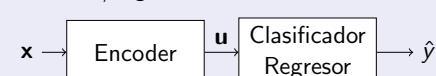


Generar alguna métrica que combine variables muy distintas entre si (radio-controlled helicopters)

Pre-processing

Preprocesing: Opción 1

Entrenar el autoencoder y luego usar las muestras en el espacio latente para entrenar el clasificador/regresor.



TPS

Matías Vera

No Supervisado

9 / 39

TPS

Matías Vera

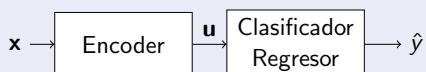
No Supervisado

10 / 39

Pre-processing

Preprocessing: Opción 1

Entrenar el autoencoder y luego usar las muestras en el espacio latente para entrenar el clasificador/regresor.



Preprocessing: Opción 2

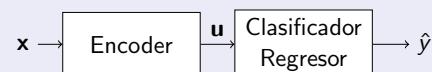
Entrenar el autoencoder y luego usar las reconstrucciones para entrenar el clasificador.



Pre-processing

Preprocesing: Opción 1

Entrenar el autoencoder y luego usar las muestras en el espacio latente para entrenar el clasificador/regresor.



Preprocesing: Opción 2

Entrenar el autoencoder y luego usar las reconstrucciones para entrenar el clasificador.



Semi-supervise learning

Puedo usar las muestras no supervisadas para entrenar el autoencoder y las supervisadas para el clasificador o el regresor final.

TPS

Matias Vera

No Supervisado

10 / 39

TPS

Matias Vera

No Supervisado

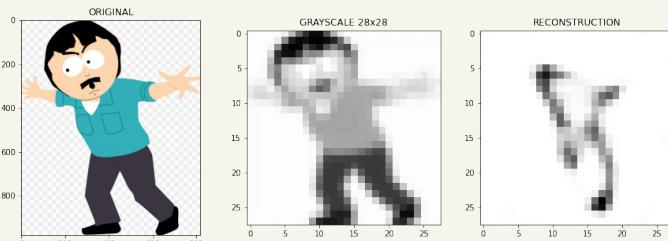
10 / 39

Detección de anomalías

Paradigma

Durante el entrenamiento un autoencoder aprende patrones en los datos para reconstruirlos con cierta facilidad. Entonces es de esperar que una muestra que no cumpla los patrones aprendidos sea más difícil de reconstruir.

EJEMPLO AUTOENCODER ENTRENADO CON MNIST:



TPS

Matias Vera

No Supervisado

11 / 39

TPS

Matias Vera

No Supervisado

12 / 39

¿Cuando usar un autoencoder?

Clasificación de las aplicaciones

Las aplicaciones de los autoencoders se dividen en dos grupos:

- Las que son relevantes por si mismas.
- Las que son un paso intermedio hacia una tarea de clasificación o regresión. ← **Siempre servirá?**

¿Cuando usar un autoencoder?

Clasificación de las aplicaciones

Las aplicaciones de los autoencoders se dividen en dos grupos:

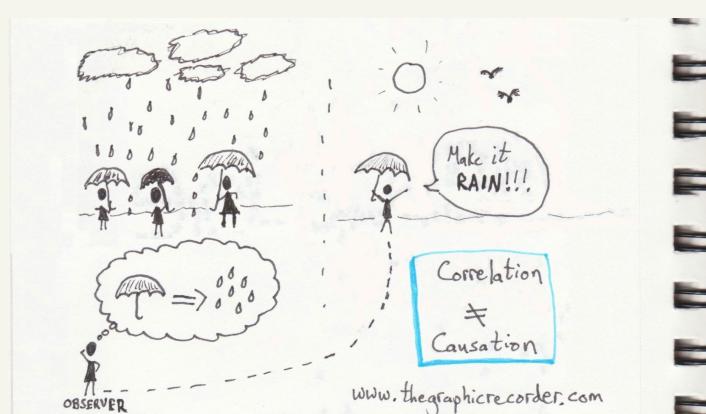
- Las que son relevantes por si mismas.
- Las que son un paso intermedio hacia una tarea de clasificación o regresión. ← **Siempre servirá?**

¿Que distribución aprende durante el entrenamiento?

Desde un punto de vista probabilístico, el entrenamiento de un algoritmo busca aprender la distribución estadística (total o parcial) de los datos:

- **Aprendizaje supervisado:** Para cada entrada x , se desea aprender parte de la información contenida en la distribución de una variable objetivo $Y|X = x$.
- **Aprendizaje no supervisado:** Toda la información aprendida estará contenida en distribución de los datos X .

Hablemos de causalidad



TPS

Matias Vera

No Supervisado

12 / 39

TPS

Matias Vera

No Supervisado

13 / 39

Causalidad: ¿Quién causa a quién?

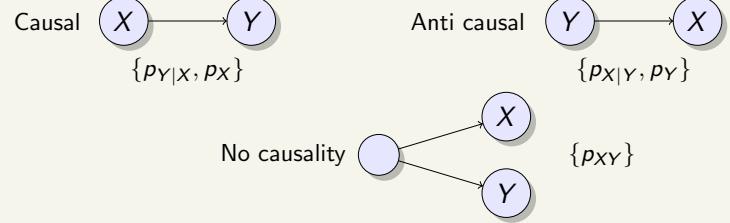
Independent Causal Mechanisms (ICM) Principle

The causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other. In the probabilistic case, this means that the conditional distribution of each variable given its causes (i.e., its mechanism) does not inform or influence the other mechanisms.

Causalidad: ¿Quién causa a quién?

Independent Causal Mechanisms (ICM) Principle

The causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other. In the probabilistic case, this means that the conditional distribution of each variable given its causes (i.e., its mechanism) does not inform or influence the other mechanisms.



TPS

Matías Vera

No Supervisado

14 / 39

TPS

Matías Vera

No Supervisado

14 / 39

Causalidad: ¿Quién causa a quién?

$$Y = g(X, U) \quad \text{con} \quad X \perp U \quad \text{o} \quad X = g(Y, U) \quad \text{con} \quad Y \perp U$$

Causalidad: ¿Quién causa a quién?

$$Y = g(X, U) \quad \text{con} \quad X \perp U \quad \text{o} \quad X = g(Y, U) \quad \text{con} \quad Y \perp U$$

La estadística no basta!

Para toda conjunta p_{XY} siempre existe $U \perp X$ y $g(\cdot, \cdot)$ tal que $Y = g(X, U)$

TPS

Matías Vera

No Supervisado

15 / 39

TPS

Matías Vera

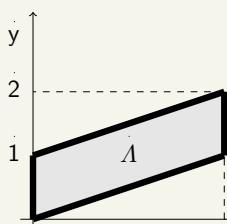
No Supervisado

15 / 39

Causalidad: ¿Quién causa a quién?

La estadística no basta!

Para toda conjunta p_{XY} siempre existe $U \perp X$ y $g(\cdot, \cdot)$ tal que $Y = g(X, U)$



$$Y|X = x \sim \mathcal{U}(x, x+1) \equiv x + \mathcal{U}(0, 1)$$

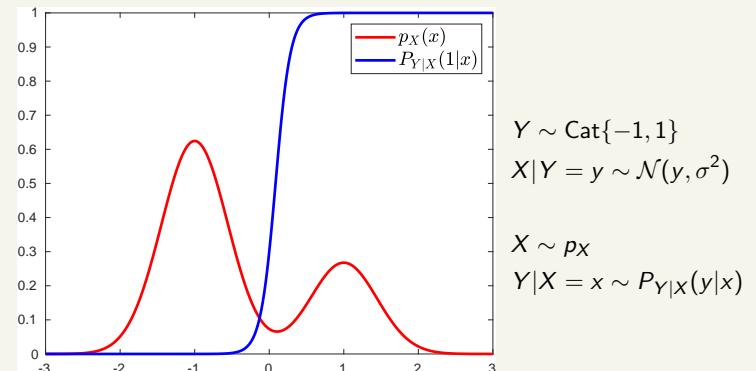
$$X|Y = y \sim \begin{cases} \mathcal{U}(0, y) & 0 < y < 1 \\ \mathcal{U}(y-1, 1) & 1 < y < 2 \end{cases}$$

$$(X, Y) \sim \mathcal{U}(A)$$

Causalidad: ¿Quién causa a quién?

La estadística no basta!

Para toda conjunta p_{XY} siempre existe $U \perp X$ y $g(\cdot, \cdot)$ tal que $Y = g(X, U)$



TPS

Matías Vera

No Supervisado

15 / 39

TPS

Matías Vera

No Supervisado

15 / 39

Causalidad: ¿Quién causa a quién?

La estadística no basta!

Para toda conjunta p_{XY} siempre existe $U \perp X$ y $g(\cdot, \cdot)$ tal que $Y = g(X, U)$

$$\begin{aligned} p_{XY}(x, y) &= e^{-x} \mathbb{1}_{\{0 < y < x\}} \\ &= \underbrace{xe^{-x} \mathbb{1}_{\{x > 0\}}}_{p_X(x)} \underbrace{\frac{1}{x} \mathbb{1}_{\{0 < y < x\}}}_{p_{Y|X}(y|x)} \\ &= \underbrace{e^{-(x-y)} \mathbb{1}_{\{x > y\}}}_{p_{X|Y}(x|y)} \underbrace{e^{-y} \mathbb{1}_{\{y > 0\}}}_{p_Y(y)} \end{aligned}$$

Causalidad: ¿Quién causa a quién?

La estadística no basta!

Para toda conjunta p_{XY} siempre existe $U \perp X$ y $g(\cdot, \cdot)$ tal que $Y = g(X, U)$

$$\begin{aligned} p_{XY}(x, y) &= e^{-x} \mathbb{1}_{\{0 < y < x\}} \\ &= \underbrace{xe^{-x} \mathbb{1}_{\{x > 0\}}}_{p_X(x)} \underbrace{\frac{1}{x} \mathbb{1}_{\{0 < y < x\}}}_{p_{Y|X}(y|x)} \\ &= \underbrace{e^{-(x-y)} \mathbb{1}_{\{x > y\}}}_{p_{X|Y}(x|y)} \underbrace{e^{-y} \mathbb{1}_{\{y > 0\}}}_{p_Y(y)} \end{aligned}$$

$$Y = X \cdot \mathcal{U}(0, 1), \quad X = Y + \mathcal{E}(1)$$

TPS

Matías Vera

No Supervisado

15 / 39

TPS

Matías Vera

No Supervisado

15 / 39

Causal and Anticausal Learning

Causal Learning

Desde esta perspectiva, en una configuración causal $X \rightarrow Y$ no debería ayudarnos conocer p_X a inferir $p_{Y|X}$.

Solución Óptima

Las decisiones óptimas $\hat{P}_\theta(y|x) = P_{Y|X}(y|x)$, $\varphi_\theta(x) = \mathbb{E}[Y|X=x]$ y $\phi_\theta(x) = \arg \max_y P_{Y|X}(y|x)$ no dependen de la marginal. Es decir, la solución es la misma por más que cambie la marginal p_X .

Causal and Anticausal Learning

Causal Learning

Desde esta perspectiva, en una configuración causal $X \rightarrow Y$ no debería ayudarnos conocer p_X a inferir $p_{Y|X}$.

Solución Óptima

Las decisiones óptimas $\hat{P}_\theta(y|x) = P_{Y|X}(y|x)$, $\varphi_\theta(x) = \mathbb{E}[Y|X=x]$ y $\phi_\theta(x) = \arg \max_y P_{Y|X}(y|x)$ no dependen de la marginal. Es decir, la solución es la misma por más que cambie la marginal p_X .

Igual un poquito ayuda

$$\begin{aligned} \arg \min_{\theta \in \Theta} \mathbb{E}[-\log \hat{P}_\theta(Y|X)] &= \arg \min_{\theta \in \Theta} \mathbb{E}_{p_X} [\text{KL}(P_{Y|X}(\cdot|X) \| \hat{P}_\theta(\cdot|X))] \\ \arg \min_{\theta \in \Theta} \mathbb{E}[(Y - \varphi_\theta(X))^2] &= \arg \min_{\theta \in \Theta} \mathbb{E}_{p_X} [(\varphi_\theta(X) - \mathbb{E}[Y|X])^2] \\ \arg \min_{\theta \in \Theta} \mathbb{P}(Y \neq \phi_\theta(X)) &= \arg \min_{\theta \in \Theta} \mathbb{E}_{p_X} \left[\max_y P_{Y|X}(y|X) - P_{Y|X}(\phi_\theta(X)|X) \right] \end{aligned}$$

TPS

Matías Vera

No Supervisado

16 / 39

TPS

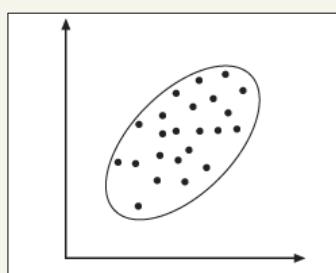
Matías Vera

No Supervisado

16 / 39

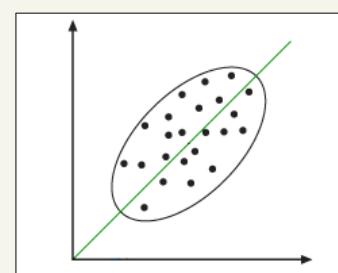
Principal Components Analysis

Reducción lineal



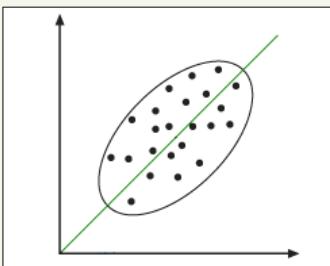
Principal Components Analysis

Reducción lineal



Principal Components Analysis

Reducción lineal



PASO 1: Normalizar

$$\tilde{x}_j^{(i)} = \frac{x_j^{(i)} - \mu_j}{\sigma_j}$$

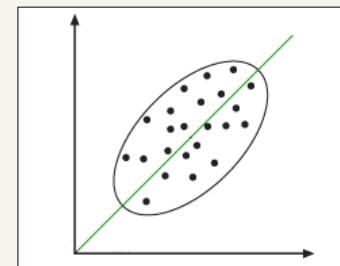
con

$$\mu_j = \frac{1}{n} \sum_{i=1}^n x_j^{(i)}$$

$$\sigma_j^2 = \frac{1}{n} \sum_{i=1}^n (x_j^{(i)} - \mu_j)^2$$

Principal Components Analysis

Reducción lineal



PASO 1: Normalizar

$$\tilde{x}_j^{(i)} = \frac{x_j^{(i)} - \mu_j}{\sigma_j}$$

con

$$\mu_j = \frac{1}{n} \sum_{i=1}^n x_j^{(i)}$$

$$\sigma_j^2 = \frac{1}{n} \sum_{i=1}^n (x_j^{(i)} - \mu_j)^2$$

PASO 2: Buscar el principal autovector \mathbf{v}_1

$$\min_{\substack{\mathbf{v}_1: \\ \|\mathbf{v}_1\|^2=1}} \sum_{i=1}^n \|\tilde{\mathbf{x}}^{(i)} - \alpha_i \mathbf{v}_1\|^2 \quad \text{con} \quad \langle \tilde{\mathbf{x}}^{(i)} - \alpha_i \mathbf{v}_1, \mathbf{v}_1 \rangle = 0$$

Lectura recomendada: Andrew Ng - "Lecture notes: Principal components analysis".

TPS

Matías Vera

No Supervisado

17 / 39

Lectura recomendada: Andrew Ng - "Lecture notes: Principal components analysis".

TPS

Matías Vera

No Supervisado

17 / 39

Principal Components Analysis

Algunas cuentas

Condición de ortogonalidad:

$$\langle \tilde{\mathbf{x}}^{(i)} - \alpha_i \mathbf{v}_1, \mathbf{v}_1 \rangle = 0 \rightarrow \langle \tilde{\mathbf{x}}^{(i)}, \mathbf{v}_1 \rangle = \alpha_i \|\mathbf{v}_1\|^2 = \alpha_i$$

Principal Components Analysis

Algunas cuentas

Condición de ortogonalidad:

$$\langle \tilde{\mathbf{x}}^{(i)} - \alpha_i \mathbf{v}_1, \mathbf{v}_1 \rangle = 0 \rightarrow \langle \tilde{\mathbf{x}}^{(i)}, \mathbf{v}_1 \rangle = \alpha_i \|\mathbf{v}_1\|^2 = \alpha_i$$

Optimización:

$$\min_{\substack{\mathbf{v}_1: \\ \|\mathbf{v}_1\|^2=1}} \sum_{i=1}^n \|\tilde{\mathbf{x}}^{(i)} - \alpha_i \mathbf{v}_1\|^2 = \min_{\substack{\mathbf{v}_1: \\ \|\mathbf{v}_1\|^2=1}} \sum_{i=1}^n \|\tilde{\mathbf{x}}^{(i)}\|^2 - \alpha_i^2$$

TPS

Matías Vera

No Supervisado

18 / 39

TPS

Matías Vera

No Supervisado

18 / 39

Principal Components Analysis

Algunas cuentas

Condición de ortogonalidad:

$$\langle \tilde{\mathbf{x}}^{(i)} - \alpha_i \mathbf{v}_1, \mathbf{v}_1 \rangle = 0 \rightarrow \langle \tilde{\mathbf{x}}^{(i)}, \mathbf{v}_1 \rangle = \alpha_i \|\mathbf{v}_1\|^2 = \alpha_i$$

Optimización:

$$\min_{\substack{\mathbf{v}_1: \\ \|\mathbf{v}_1\|=1}} \sum_{i=1}^n \|\tilde{\mathbf{x}}^{(i)} - \alpha_i \mathbf{v}_1\|^2 = \min_{\substack{\mathbf{v}_1: \\ \|\mathbf{v}_1\|=1}} \sum_{i=1}^n \|\tilde{\mathbf{x}}^{(i)}\|^2 - \alpha_i^2$$

$$\max_{\substack{\mathbf{v}_1: \\ \|\mathbf{v}_1\|=1}} \frac{1}{n} \sum_{i=1}^n \langle \tilde{\mathbf{x}}^{(i)}, \mathbf{v}_1 \rangle^2 = \max_{\substack{\mathbf{v}_1: \\ \|\mathbf{v}_1\|=1}} \mathbf{v}_1^T \underbrace{\left(\frac{1}{n} \sum_{i=1}^n \tilde{\mathbf{x}}^{(i)} (\tilde{\mathbf{x}}^{(i)})^T \right)}_{\Sigma} \mathbf{v}_1$$

Principal Components Analysis

Algunas cuentas

$$J(\mathbf{v}_1) = \mathbf{v}_1^T \Sigma \mathbf{v}_1 - \lambda (\mathbf{v}_1^T \mathbf{v}_1 - 1)$$

TPS

Matías Vera

No Supervisado

18 / 39

Lectura recomendada: Petersen and Pedersen - "Matrix Cookbook".

TPS

Matías Vera

No Supervisado

19 / 39

Principal Components Analysis

Algunas cuentas

$$J(\mathbf{v}_1) = \mathbf{v}_1^T \Sigma \mathbf{v}_1 - \lambda (\mathbf{v}_1^T \mathbf{v}_1 - 1)$$

$$\nabla J(\mathbf{v}_1) = 2\Sigma \mathbf{v}_1 - 2\lambda \mathbf{v}_1 = 0$$

$$\Sigma \mathbf{v}_1 = \lambda \mathbf{v}_1 \rightarrow \mathbf{v}_1 \text{ es AVE de } \Sigma \text{ y } \lambda \text{ es AVA}$$

Lectura recomendada: Petersen and Pedersen - "Matrix Cookbook".

TPS

Matías Vera

No Supervisado

19 / 39

Principal Components Analysis

Algunas cuentas

$$J(\mathbf{v}_1) = \mathbf{v}_1^T \Sigma \mathbf{v}_1 - \lambda (\mathbf{v}_1^T \mathbf{v}_1 - 1)$$

$$\nabla J(\mathbf{v}_1) = 2\Sigma \mathbf{v}_1 - 2\lambda \mathbf{v}_1 = 0$$

$$\Sigma \mathbf{v}_1 = \lambda \mathbf{v}_1 \rightarrow \mathbf{v}_1 \text{ es AVE de } \Sigma \text{ y } \lambda \text{ es AVA}$$

Lectura recomendada: Petersen and Pedersen - "Matrix Cookbook".

TPS

Matías Vera

No Supervisado

19 / 39

Principal Components Analysis

Algunas cuentas

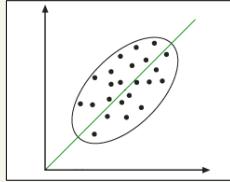
$$J(\mathbf{v}_1) = \mathbf{v}_1^T \Sigma \mathbf{v}_1 - \lambda (\mathbf{v}_1^T \mathbf{v}_1 - 1)$$

$$\nabla J(\mathbf{v}_1) = 2\Sigma \mathbf{v}_1 - 2\lambda \mathbf{v}_1 = 0$$

$$\Sigma \mathbf{v}_1 = \lambda \mathbf{v}_1 \rightarrow \mathbf{v}_1 \text{ es AVE de } \Sigma \text{ y } \lambda \text{ es AVA}$$

El problema de optimización pasa a ser de la forma

$$\max_{\substack{\mathbf{v}_1 \\ \|\mathbf{v}_1\|^2=1}} \mathbf{v}_1^T \Sigma \mathbf{v}_1 = \max_{\substack{\mathbf{v}_1 \\ \|\mathbf{v}_1\|^2=1}} \lambda(\mathbf{v}_1) \rightarrow \text{Máximo AVA}$$



Lectura recomendada: Petersen and Pedersen - "Matrix Cookbook".

TPS

Matías Vera

No Supervisado

19 / 39

Principal Components Analysis

Reducción y Reconstrucción

Componentes principales

Este procedimiento se puede repetir para encontrar el 2do, 3er, etc. componente principal. El resultado son el 2do, 3er, etc autovalor con su autovector como dirección.

Principal Components Analysis

Reducción y Reconstrucción

Componentes principales

Este procedimiento se puede repetir para encontrar el 2do, 3er, etc. componente principal. El resultado son el 2do, 3er, etc autovalor con su autovector como dirección.

Sobre los autovalores

El porcentaje de energía perdida puede medirse por la proporción de autovalores despreciados.

- \mathbf{V} : Matriz de autovectores más relevantes.
- \mathbf{x} : Variable de entrada a procesar (ya normalizada).
- \mathbf{u} : Variable latente.
- $\hat{\mathbf{x}}$: Reconstrucción

$$\mathbf{u} = \mathbf{V} \cdot \mathbf{x}, \quad \hat{\mathbf{x}} = \mathbf{V}^T \cdot \mathbf{u}$$

Outline

- ① Autoencoders
- ② Principal Components Analysis (PCA)
- ③ K-Means
- ④ Algoritmo EM
- ⑤ Factor Analysis

TPS

Matías Vera

No Supervisado

20 / 39

TPS

Matías Vera

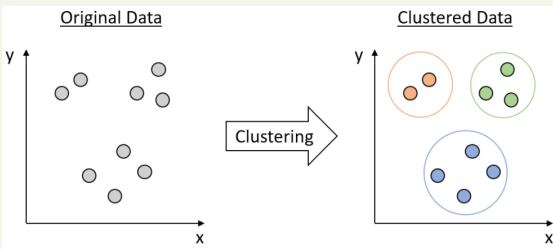
No Supervisado

21 / 39

Clustering

Clustering

Estos algoritmos son la versión no supervisada de la clasificación. Su objetivo es agrupar muestras de manera de tener un mayor entendimiento del *manifold*.



TPS

Matias Vera

No Supervisado

22 / 39

TPS

Matias Vera

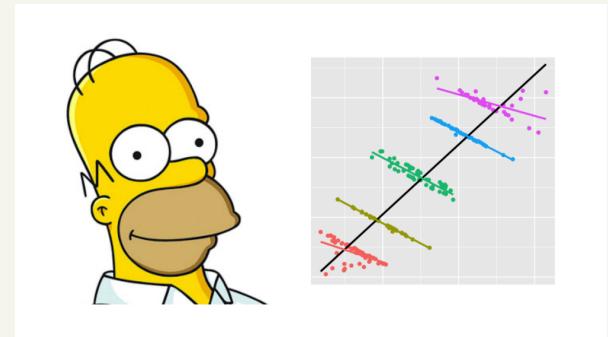
No Supervisado

23 / 39

Motivación: Paradoja de Simpson

Paradoja de Simpson

La paradoja de Simpson se da cuando dos (o más) variables tienen una correlación hacia un sentido pero al agrupar los datos se ve que, en cada cluster, la correlación posee en realidad el sentido opuesto.



Paradoja de Simpson: Covid-19 Case Fatality Rates (CFR)

Edad	0-9	10-19	20-29	30-39	40-49	50-59	60-69	70-79	≥ 80	Total
Italia	0%	0%	0%	0%	0.1%	0.2%	2.5%	6.4%	13.2%	4.4%
(0/43) (0/85) (0/296) (0/470) (1/891) (3/1453) (37/1471) (114/1785) (202/1532) (357/8026)										

Edad	0-9	10-19	20-29	30-39	40-49	50-59	60-69	70-79	≥ 80	Total
China	0%	0.2%	0.2%	0.2%	0.4%	1.3%	3.6%	8%	14.8%	2.3%
(0/0) (1/549) (7/3619) (18/7600) (38/8571) (130/10008) (309/8583) (312/3918) (208/1408) (1023/44672)										

Julius von Kugelgen, Luigi Greselle and Bernhard Scholkopf "Simpson's paradox in Covid-19 case fatality rates: A mediation analysis of age-related causal effects" IEEE Transactions on Artificial Intelligence 2021.

TPS

Matias Vera

No Supervisado

24 / 39

Algoritmo K-Means

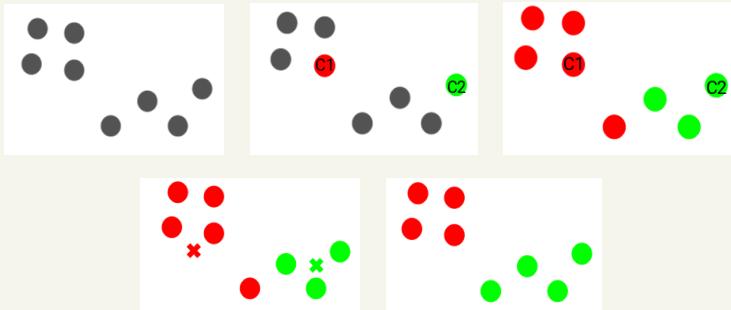
K-means

Algoritmo de clustering para agrupar los datos en K clusters (previamente definidos). Se basa en encontrar, de forma iterativa, los *centroídes* de cada clase y asignar cada muestra al centroide más cercano.

Algorithm 1 K-means

```
1: procedure KMEANS( $X, K$ )
   Input:  $X \in \mathbb{R}^{n \times d_x}$  matriz de datos y  $K$  número de clusters.
   Output:  $\mu \in \mathbb{R}^{K \times d_x}$  centroídes e  $y \in \{1, \dots, K\}^n$  etiquetas.
2:   Inicializar  $\mu$  con el valor de  $K$  columnas de  $X$  elegida al azar.
3:   repeat
4:      $y[i] = \arg \min_k \|X[i, :] - \mu[k, :]\|$  ▷ Con  $i = 1, \dots, n$ .
5:      $\mu[k, :] = \mathbb{E}[X[y == k, :]]$  ▷ Con  $k = 1, \dots, K$ 
6:   until convergencia
7:   Return:  $\mu$  e  $y$ 
8: end procedure
```

Algoritmo K-Means



TPS

Matias Vera

No Supervisado

26 / 39

Outline

- 1 Autoencoders
- 2 Principal Components Analysis (PCA)
- 3 K-Means
- 4 Algoritmo EM
- 5 Factor Analysis

27 / 39

Máxima Verosimilitud

Algoritmos de Máxima Verosimilitud

La minimización de la *cross-entropy* equivale a encontrar algoritmos de máxima verosimilitud. El problema es que estos son muchas veces analíticamente intratables y computacionalmente muy pesados de tratar (ej. mezcla de gaussianas).

$$\hat{\theta} = \arg \max_{\theta \in \Theta} \sum_{i=1}^n \log p(X_i | \theta)$$

Variables no observable

Sea Z una variable no observable del problema con densidad condicional $p(z|x, \theta)$, y sea \mathcal{P} la familia de todas las posibles densidades condicionales de $Z|X = x$. Luego, el estimador de MV puede reescribirse como:

$$\begin{aligned}\hat{\theta} &= \arg \max_{\theta \in \Theta} \max_{q \in \mathcal{P}} \sum_{i=1}^n [\log p(X_i | \theta) - \text{KL}(q(\cdot | X_i) \| p(\cdot | X_i, \theta))] \\ &= \arg \max_{\theta \in \Theta} \max_{q \in \mathcal{P}} \text{ELBO}(\theta, q)\end{aligned}$$

TPS

Matías Vera

No Supervisado

28 / 39

TPS

Matías Vera

No Supervisado

29 / 39

Algoritmo EM

Algoritmo Expectation - Maximization

El algoritmo EM consiste en inicializar en algún valor θ_0 e iterar entre:

- $q^{(t)} = \arg \max_{q \in \mathcal{P}} \text{ELBO}(\theta^{(t-1)}, q)$ (Expectation)
- $\theta^{(t)} = \arg \max_{\theta \in \Theta} \text{ELBO}(\theta, q^{(t)})$ (Maximization)

Expectación

El paso *Expectation* puede simplificarse a la relación $q^{(t)}(z|x) = p(z|x, \theta^{(t-1)})$. Es decir:

$$\theta^{(t)} = \arg \max_{\theta \in \Theta} \sum_{i=1}^n \left[\log p(X_i | \theta) - \text{KL} \left(p(\cdot | X_i, \theta^{(t-1)}) \| p(\cdot | X_i, \theta) \right) \right]$$

TPS

Matías Vera

No Supervisado

29 / 39

TPS

Matías Vera

No Supervisado

30 / 39

Algoritmo EM

Maximización

El paso *Maximization* puede simplificarse reescribiendo cada sumando como

$$\log p(x|\theta) - \text{KL}(q(\cdot|x) \| p(\cdot|x, \theta)) = H(q(\cdot|x)) + \mathbb{E}_q [\log p(x, Z|\theta) | X = x]$$

donde la entropía no depende de θ . Es decir,

$$\theta^{(t)} = \arg \max_{\theta \in \Theta} \sum_{i=1}^n \mathbb{E}_{q^{(t)}} [\log p(X_i, Z|\theta) | X_i]$$

Teorema: Monotonía

En el algoritmo EM ocurre que

$$\sum_{i=1}^n \log p(X_i | \theta^{(t)}) \geq \sum_{i=1}^n \log p(X_i | \theta^{(t-1)})$$

Hint: Expectación + KL ≥ 0 .

TPS

Matías Vera

No Supervisado

30 / 39

Algoritmo EM

Algoritmo Expectation - Maximization

El algoritmo EM consiste en inicializar en algún valor θ_0 e iterar entre:

- $q^{(t)} = \arg \max_{q \in \mathcal{P}} \text{ELBO}(\theta^{(t-1)}, q)$ (Expectation)
- $\theta^{(t)} = \arg \max_{\theta \in \Theta} \text{ELBO}(\theta, q^{(t)})$ (Maximization)

Algoritmo EM para mezcla de gaussianas

Definición del problema

Si $Z \sim \text{Cat}(\{c_1, \dots, c_K\})$ y $X|Z = k \sim \mathcal{N}(\mu_k, \Sigma_k)$, está claro que X es una mezcla de gaussianas. Sea $\theta = \{c_k, \mu_k, \Sigma_k\}_{k=1}^K$, se desea estimar estos parámetros (de forma no supervisada, es decir siendo Z no observable). El estimador de máxima verosimilitud es intratable y por eso recurrimos al algoritmo EM.

TPS

Matías Vera

No Supervisado

31 / 39

Algoritmo EM para mezcla de gaussianas

Definición del problema

Si $Z \sim \text{Cat}(\{c_1, \dots, c_K\})$ y $X|Z = k \sim \mathcal{N}(\mu_k, \Sigma_k)$, está claro que X es una mezcla de gaussianas. Sea $\theta = \{c_k, \mu_k, \Sigma_k\}_{k=1}^K$, se desea estimar estos parámetros (de forma no supervisada, es decir siendo Z no observable). El estimador de máxima verosimilitud es intratable y por eso recurrimos al algoritmo EM.

Expectación

El paso de expectación es simplemente elegir:

$$q(k|x) = p(k|x, \theta) = \frac{c_k \cdot |\Sigma_k|^{-1/2} \cdot e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}}{\sum_{m=1}^K c_m \cdot |\Sigma_m|^{-1/2} \cdot e^{-\frac{1}{2}(x-\mu_m)^T \Sigma_m^{-1}(x-\mu_m)}}$$

TPS

Matías Vera

No Supervisado

31 / 39

TPS

Matías Vera

No Supervisado

32 / 39

Algoritmo EM para mezcla de gaussianas

Derivada respecto a c_k

Igualamos a cero la derivada respecto a c_k y usamos que $\sum_{k=1}^K c_k = 1$.

$$\begin{aligned} \frac{\partial \mathcal{L}(\theta)}{\partial c_k} &= \left(\sum_{i=1}^n \frac{q(k|x_i)}{c_k} \right) - \lambda = 0 \\ \Rightarrow c_k &= \frac{1}{\lambda} \sum_{i=1}^n q(k|x_i) \\ \Rightarrow c_k &= \frac{1}{n} \sum_{i=1}^n q(k|x_i) \end{aligned}$$

TPS

Matías Vera

No Supervisado

33 / 39

TPS

Matías Vera

No Supervisado

34 / 39

Algoritmo EM para mezcla de gaussianas

Derivada respecto a μ_k

Igualamos a cero (vector) la derivada respecto a μ_k .

$$\begin{aligned} \frac{\partial \mathcal{L}(\theta)}{\partial \mu_k} &= \sum_{i=1}^n q(k|x_i) \Sigma_k^{-1}(x_i - \mu_k) = 0 \\ \Rightarrow \mu_k &= \frac{\sum_{i=1}^n q(k|x_i) \cdot x_i}{\sum_{i=1}^n q(k|x_i)} \end{aligned}$$

Derivada respecto a Σ_k

Igualamos a cero (matriz) la derivada respecto a Σ_k .

$$\begin{aligned} \frac{\partial \mathcal{L}(\theta)}{\partial \Sigma_k} &= \sum_{i=1}^n q(k|x_i) \left[-\frac{1}{2} \Sigma_k^{-1} + \frac{1}{2} \Sigma_k^{-1} (x_i - \mu_k)(x_i - \mu_k)^T \Sigma_k^{-1} \right] = 0 \\ \Rightarrow \Sigma_k &= \frac{\sum_{i=1}^n q(k|x_i) \cdot (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^n q(k|x_i)} \end{aligned}$$

Petersen and Pedersen - "Matrix Cookbook".

TPS

Matías Vera

No Supervisado

34 / 39

Algoritmo EM para mezcla de gaussianas

Maximización

Dado un q , se desea maximizar:

$$\max_{\theta} \sum_{i=1}^n \mathbb{E}_q [\log p(X_i, Z|\theta)|X_i] \quad \text{s.t.} \quad \sum_{k=1}^K c_k = 1$$

Es decir que, utilizando multiplicadores de Lagrange, la función a derivar e igualar a cero es:

$$\mathcal{L}(\theta) = \sum_{i=1}^n \sum_{k=1}^K q(k|x_i) \left[\log c_k - \frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k) \right] + \lambda \left(1 - \sum_{k=1}^K c_k \right)$$

Algoritmo EM para mezcla de gaussianas

Derivada respecto a μ_k

Igualamos a cero (vector) la derivada respecto a μ_k .

$$\begin{aligned} \frac{\partial \mathcal{L}(\theta)}{\partial \mu_k} &= \sum_{i=1}^n q(k|x_i) \Sigma_k^{-1} (x_i - \mu_k) = 0 \\ \Rightarrow \mu_k &= \frac{\sum_{i=1}^n q(k|x_i) \cdot x_i}{\sum_{i=1}^n q(k|x_i)} \end{aligned}$$

Petersen and Pedersen - "Matrix Cookbook".

TPS

Matías Vera

No Supervisado

34 / 39

Algoritmo EM para máximo a posteriori

Estimador puntual con enfoque Bayesiano

Si modelamos θ como variable aleatoria y suponemos alguna distribución *a priori* $\pi(\theta)$, definimos el estimador MAP como:

$$\begin{aligned} \hat{\theta} &= \arg \max_{\theta \in \Theta} \log p(\theta|X) \\ &= \arg \max_{\theta \in \Theta} \log \pi(\theta) + \sum_{i=1}^n \log p(X_i|\theta) \\ &= \arg \max_{\theta \in \Theta} \log \pi(\theta) + \max_{q \in \mathcal{P}} \text{ELBO}(\theta, q) \end{aligned}$$

M-step

La prior solo modifica la maximización:

$$\theta^{(t)} = \arg \max_{\theta \in \Theta} \log \pi(\theta) + \sum_{i=1}^n \mathbb{E}_{q^{(t)}} [\log p(X_i, Z|\theta)|X_i]$$

TPS

Matías Vera

No Supervisado

34 / 39

TPS

Matías Vera

No Supervisado

35 / 39

Outline

1 Autoencoders

2 Principal Components Analysis (PCA)

3 K-Means

4 Algoritmo EM

5 Factor Analysis

Aplicación de EM: Factor Analysis

Factor Analysis

Al igual que PCA, el algoritmo EM puede utilizarse para reducir la dimensión. El modelo consiste en suponer que los *features* se puede descomponer en factores: $X = \mu + W \cdot Z + \epsilon$ con $\mu \in \mathbb{R}^{d_x}$, $W \in \mathbb{R}^{d_x \times d_z}$, $Z \sim \mathcal{N}(0, I)$ (de dimensión d_z) y $\epsilon \sim \mathcal{N}(0, \Psi)$ (de dimensión d_x) con Ψ una matriz diagonal y con Z y ϵ independientes. En este caso, $\theta = \{\mu, W, \Psi\}$.

TPS

Matías Vera

No Supervisado

36 / 39

TPS

Matías Vera

No Supervisado

37 / 39

Aplicación de EM: Factor Analysis

Factor Analysis

Al igual que PCA, el algoritmo EM puede utilizarse para reducir la dimensión. El modelo consiste en suponer que los *features* se puede descomponer en factores: $X = \mu + W \cdot Z + \epsilon$ con $\mu \in \mathbb{R}^{d_x}$, $W \in \mathbb{R}^{d_x \times d_z}$, $Z \sim \mathcal{N}(0, I)$ (de dimensión d_z) y $\epsilon \sim \mathcal{N}(0, \Psi)$ (de dimensión d_x) con Ψ una matriz diagonal y con Z y ϵ independientes. En este caso, $\theta = \{\mu, W, \Psi\}$.

Expectación

Dado que la conjunta entre (X, Z) es una normal multivariada, la condicional de $Z|X = x$ también lo será. Es decir que $p(z|x, \theta)$ se caracterizará por una media (recta función de x) y una matriz de covarianza constante (no depende de x).

TPS

Matías Vera

No Supervisado

37 / 39

TPS

Matías Vera

No Supervisado

38 / 39

Aplicación: Factor Analysis

Maximización

Como la marginal $p(Z)$ no depende de θ , la maximización se reduce a:

$$\theta^{(t)} = \arg \max_{\theta \in \Theta} \sum_{i=1}^n \mathbb{E}_{q^{(t)}} [\log p(X_i|Z, \theta)|X_i]$$

Aplicación: Factor Analysis

Maximización

Como la marginal $p(Z)$ no depende de θ , la maximización se reduce a:

$$\theta^{(t)} = \arg \max_{\theta \in \Theta} \sum_{i=1}^n \mathbb{E}_{q^{(t)}} [\log p(X_i|Z, \theta)|X_i]$$

Encoder - Decoder

Una vez entrenado el algoritmo, se utiliza $\mathbb{E}[Z|X = x, \theta]$ como *encoder* y $\mathbb{E}[X|Z = z, \theta]$ como *decoder*.

Aplicación: Factor Analysis

Maximización

Como la marginal $p(Z)$ no depende de θ , la maximización se reduce a:

$$\theta^{(t)} = \arg \max_{\theta \in \Theta} \sum_{i=1}^n \mathbb{E}_{q^{(t)}} [\log p(X_i|Z, \theta)|X_i]$$

Encoder - Decoder

Una vez entrenado el algoritmo, se utiliza $\mathbb{E}[Z|X = x, \theta]$ como *encoder* y $\mathbb{E}[X|Z = z, \theta]$ como *decoder*.

Probabilistic PCA

Si en lugar de pedir que Ψ sea diagonal se pide que todas las varianzas coincidan $\Psi = \sigma^2 \mathbf{I}$, se obtiene un algoritmo conocido como Probabilistic PCA.

TPS

Matías Vera

No Supervisado

38 / 39

TPS

Matías Vera

No Supervisado

38 / 39

Esquema de despejes de ecuaciones

- ① Hallar la distribución condicional de $X|Z = z$.
- ② Hallar la distribución marginal de X .
- ③ Hallar la distribución conjunta de (X, Z) .
- ④ E-Step (los parámetros se asumen conocidos y se calcula la distribución): Hallar la distribución condicional de $Z|X = x$. Bautizar a los principales momentos: $m_i = \mathbf{E}[Z|X = x_i]$, $\Sigma_i = \text{var}(Z|X = x_i)$.
- ⑤ M-Step (la distribución anterior se asume conocida y se calculan los parámetros): Calcular la esperanza de $\log(p(X|Z, \theta))$, bajo $Z|X$.
- ⑥ Plantear la función a maximizar.
- ⑦ Para simplificar el análisis asumir que $\mu = \frac{1}{n} \sum_{i=1}^n x_i$ (no varía en las iteraciones). ¿Puede decir algo sobre $\sum_{i=1}^n m_i$?
- ⑧ Derivar respecto de W e igualar a cero. Despejar W .
- ⑨ Derivar respecto de Ψ e igualar a cero. Despejar Ψ .

Petersen and Pedersen - "Matrix Cookbook" (tenerlo a mano en todo el momento).

Andrew Ng - "Lecture notes: Factor Analysis" (chequear resultados).

Aplicaciones específicas

Taller de Procesamiento de Señales

Agenda

1 Lenguaje Natural

2 Sistemas de Recomendación

¿Cómo convertir un texto en un vector?

One-hot Encoding

Dado un vocabulario $V = \{\omega_1, \dots, \omega_{|V|}\}$, se puede convertir cada palabra en un vector *one-hot*.

$$\omega_i \xrightarrow{\text{One-Hot Encoding}} x_i = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{|V|}$$

i-ésima posición

¿Cómo convertir un texto en un vector?

One-hot Encoding

Dado un vocabulario $V = \{\omega_1, \dots, \omega_{|V|}\}$, se puede convertir cada palabra en un vector *one-hot*.

$$\omega_i \xrightarrow{\text{One-Hot Encoding}} x_i = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{|V|}$$

i-ésima posición

Procesamiento del Lenguaje Natural

Vectorizaciones Sofisticadas

En la práctica suelen utilizarse representaciones pre-entrenadas (ej. FastText).

Bolsa de palabras

Para vectorizar un documento $f(x_1, \dots, x_n)$, la manera más simple es *bolsa de palabras*: $f(x_1, \dots, x_n) = x_1 + \dots + x_n$.

Vectorizaciones Sofisticadas

En la práctica suelen utilizarse representaciones pre-entrenadas (ej. FastText).

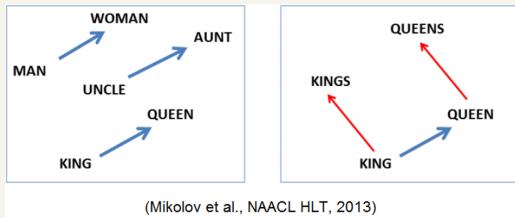
Normalizaciones de NLP

- Eliminar caracteres raros e inusuales
- Convertir todo a minúsculas
- Eliminar palabras no informativas (stop words)
- Descartar las palabras poco observadas
- Descartar las palabras más comunes
- Lemmatization (significado)
- Stemming (quedarse con la raíz)

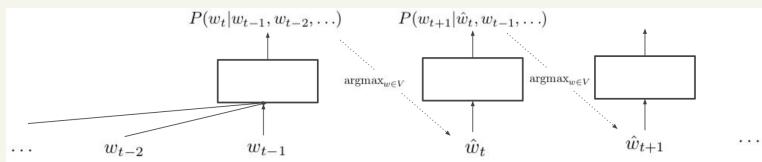
Transformación tf-idf

Medida numérica que expresa cuán relevante es una palabra para un documento dentro de un dataset. El tf-idf para un término t de un documento d perteneciente a una colección de n documentos es $\text{tf-idf}(t, d) = \text{tf}(t, d) \cdot \text{idf}(t)$. El primer factor $\text{tf}(t, d) = \frac{\#\{t \in d\}}{\#(d)}$ es la cantidad de veces que aparece el término t en el documento d dividido la cantidad de términos que aparecen en el documento d . El segundo factor $\text{idf}(t) = 1 - \log\left(\frac{\text{df}(t)}{n}\right)$, donde $\text{df}(t)$ es la cantidad de documentos que poseen el término t en su interior.

Word Vectors + PCA



Síntesis de texto



$$\text{vector(KINGS)} - \text{vector(KING)} + \text{vector(QUEEN)} = \text{vector(QUEENS)}$$

Outline

1 Lenguaje Natural

2 Sistemas de Recomendación

Sistemas de Recomendación



Problemáticas asociadas

- **Cámara de eco.** Los algoritmos de recomendación tienden a juntar a personas con ideología similar, creando un ciclo de realimentación donde todos escuchan lo que ya creen, no se expone a puntos de vista diferentes, fomenta la radicalización y el dogmatismo.
- **Filtro burbuja.** Los algoritmos filtran el contenido que no coincide con tus intereses o interacciones previas, creando una especie de burbuja en la que solo accedes a información que refuerza tus creencias.
- **Manipulaciones.** Muchos de estos algoritmos no publican su código, y por lo tanto no hay garantías que no se fomente algún tipo de contenido en particular.

Filtro Colaborativo

Aprender por Colaboración

	Item 1	Item 2	Item 3	Item 4	Item 5
Alice		👎		👍	
Bob	👍	👍	👎	?	👍
Charlie	👍	👎	?	?	👍

Bob ~ Charlie \Rightarrow ? = 👎

Filtro Colaborativo

Aprender por Colaboración

	Item 1	Item 2	Item 3	Item 4	Item 5
Alice		👎		👍	
Bob	👍	👍	👎	?	👍
Charlie	👍	👎	?	?	👍

Bob ~ Charlie \Rightarrow ? = 👎

Entrenamiento

$$\min_{x, \theta} \frac{1}{2} \sum_{(i,j): y_{i,j} > 0} (\theta_j^T \cdot x_i - y_{i,j})^2 + \frac{\lambda}{2} \left(\sum_{i=1}^{n_{\text{items}}} \|x_i\|^2 + \sum_{j=1}^{n_{\text{users}}} \|\theta_j\|^2 \right)$$

donde $y \in \mathbb{N}^{n_{\text{items}} \times n_{\text{users}}}$ contiene el dataset, $x \in \mathbb{R}^{n_{\text{items}} \times \nu}$ y $\theta \in \mathbb{R}^{n_{\text{users}} \times \nu}$ son los parámetros a entrenar; con ν la dimensión del espacio latente y $\lambda \geq 0$ un hiperparámetro de regularización.

TPS

Matías Vera

Aplicaciones

10 / 11

TPS

Matías Vera

Aplicaciones

10 / 11

Filtro Colaborativo

Inferencia (Rating)

$$\hat{y}_{i,j} = p(\theta_j^T \cdot x_i) + (1-p)\bar{y}_i$$

donde \bar{y}_i es la calificación promedio del item i -ésimo y $0 \leq p \leq 1$ es un hiperparámetro que indica cuanto peso le damos al aprendizaje y cuanto al valor medio.

Modelos Bayesianos

Taller de Procesamiento de Señales

TPS

Matías Vera

Aplicaciones

11 / 11

TPS

Matías Vera

Modelos Bayesianos

1 / 38

Agenda

1 Inferencia Bayesiana

2 Naive Bayes

3 Multinomial Naive Bayes

4 Variational Bayes

5 Técnicas de Muestreo

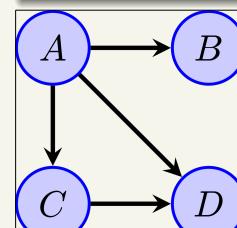
Redes Bayesianas

Modelos Gráficos

Modelos probabilísticos capaz de representarse con un grafo.

Red Bayesiana

Grafo acíclico dirigido que representa la relación de causalidad e independencia de sus variables. Dos variables aleatorias cualesquieras son condicionalmente independientes dados los valores de sus padres causales (y por lo tanto las raíces son independientes).



$$p(A, B, C, D) = p(A) \cdot p(B|A) \cdot p(C|A) \cdot p(D|A, C)$$

TPS

Matías Vera

Modelos Bayesianos

2 / 38

TPS

Matías Vera

Modelos Bayesianos

3 / 38

Inferencia Bayesiana

- Los parámetros θ deben ser considerados como realizaciones de una variable aleatoria T con una distribución a priori conocida $p(\theta)$.
- Las muestras son i.i.d. **cuando** se conoce el parámetro.
- La distribución a posteriori de los parámetros se calcula como:

$$p(\theta|\mathcal{D}_n) \propto p(\theta) \prod_{i=1}^n p(x_i|\theta)$$

con $\mathcal{D}_n = \{x_1, \dots, x_n\}$.

- Como estimador puntual suele elegirse el *maximo a posteriori* (Θ discreto) y el estimador bayesiano o *media a posteriori* (Θ continuo).
- No son necesarios los estimadores puntuales para predecir:

$$p(x_{\text{test}}|\mathcal{D}_n) = \int_{\Theta} p(x_{\text{test}}|\theta) p(\theta|\mathcal{D}_n) d\theta = \mathbb{E}[p(x_{\text{test}}|T)|\mathcal{D}_n]$$

Inferencia Bayesiana

Filosofía Bayesiana

La estadística bayesiana interpreta la probabilidad como una medida de credibilidad en un evento. Por eso se habla de que el enfoque Bayesiano busca verdades en contexto de incertidumbre.

No confundir Bayesiano con Relativista

¿Es posible entonces alcanzar verdades en las ciencias empíricas en las que es inevitable decir "no sé"? Sí. Podemos evitar mentir: maximizando incertidumbre (no afirmar más de lo que se sabe) dada la información disponible (sin ocultar lo que sí se sabe).

Inferencia Bayesiana

Filosofía Bayesiana

La estadística bayesiana interpreta la probabilidad como una medida de credibilidad en un evento. Por eso se habla de que el enfoque Bayesiano busca verdades en contexto de incertidumbre.

Inferencia Bayesiana

Filosofía Bayesiana

La estadística bayesiana interpreta la probabilidad como una medida de credibilidad en un evento. Por eso se habla de que el enfoque Bayesiano busca verdades en contexto de incertidumbre.

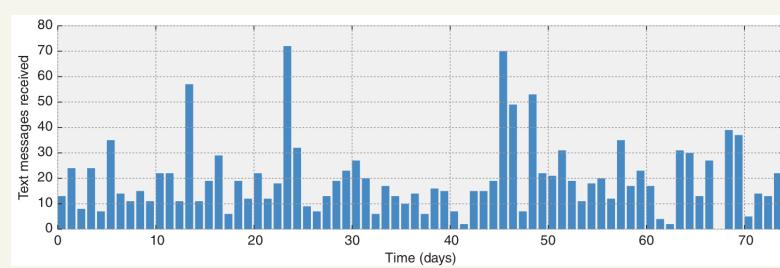
No confundir Bayesiano con Relativista

¿Es posible entonces alcanzar verdades en las ciencias empíricas en las que es inevitable decir "no sé"? Sí. Podemos evitar mentir: maximizando incertidumbre (no afirmar más de lo que se sabe) dada la información disponible (sin ocultar lo que sí se sabe).

¿Son prácticos los métodos Bayesianos?

Si, no solo por poder adaptarse a intentar resolver los mismos problemas que la estadística frecuentista (por ejemplo predicciones), sino que también pueden intentar resolver problemas donde la estadística clásica es insuficiente o iluminar el sistema subyacente con un modelado más flexible.

Ejemplo de modelado Bayesiano



Problema

Un usuario proporciona una serie de recuentos diarios de mensajes de whatsapp enviados. Tiene curiosidad por saber si los hábitos de envío de mensajes han cambiado con el tiempo. ¿Cómo puedes modelar esto?

Ejemplo de modelado Bayesiano

- La cantidad de mensajes en un día deberá ser modelada como una variable discreta cuyos átomos es \mathbb{N}_0 . Por ejemplo $X_i \sim \text{Poi}(\lambda_i)$.
- Si observamos los datos, parecería que el valor de λ_i aumenta en algún momento durante las observaciones. ¿Cómo podemos representar matemáticamente esta observación? Supongamos que algún día τ durante el período de observación, el parámetro λ_i se incrementa repentinamente. Entonces realmente tenemos dos tasas: una para el período anterior a τ y otra para el resto del período:

$$\lambda_i = \begin{cases} \beta_1 & i < \tau \\ \beta_2 & i \geq \tau \end{cases}$$

- Tanto β_1 como β_2 toman valores reales no negativos. Por ejemplo $\beta_1, \beta_2 \sim \mathcal{E}(\alpha)$. Nuestra estimación de α no influye demasiado en el modelo, por lo que tenemos cierta flexibilidad en nuestra elección. Para evitar ser demasiado obstinados con este parámetro se sugiere:

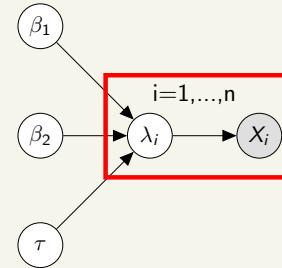
$$\alpha \approx \frac{1}{n} \sum_{i=1}^n X_i$$

Ejemplo de modelado Bayesiano

- ¿Qué pasa con τ ? Debido a la varianza de los datos, es difícil caracterizarlo en detalle. En cambio, podemos asignar la creencia menos informativa posibles $\tau \sim \mathcal{U}\{1 : T\}$.

Ejemplo de modelado Bayesiano

- ¿Qué pasa con τ ? Debido a la varianza de los datos, es difícil caracterizarlo en detalle. En cambio, podemos asignar la creencia menos informativa posibles $\tau \sim \mathcal{U}\{1 : T\}$.



TPS

Matías Vera

Modelos Bayesianos

8 / 38

TPS

Matías Vera

Modelos Bayesianos

8 / 38

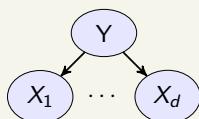
Naive Bayes

Naive Bayes

Estimar los parámetros (máxima verosimilitud o bayesiano) asumiendo que una relación de causalidad $Y \rightarrow X$ con las diferentes componentes $X_j|Y = k$ independientes.

Red Bayesiana

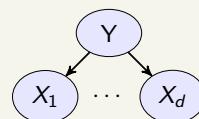
Cálculo



$$p(y|x) = \frac{p(y) \prod_{j=1}^d p(x_j|y)}{p(x)}$$

Red Bayesiana

Cálculo



$$p(y|x) = \frac{p(y) \prod_{j=1}^d p(x_j|y)}{p(x)}$$

Naive Bayes

Naive Bayes

Estimar los parámetros (máxima verosimilitud o bayesiano) asumiendo que una relación de causalidad $Y \rightarrow X$ con las diferentes componentes $X_j|Y = k$ independientes.

TPS

Matías Vera

Modelos Bayesianos

9 / 38

TPS

Matías Vera

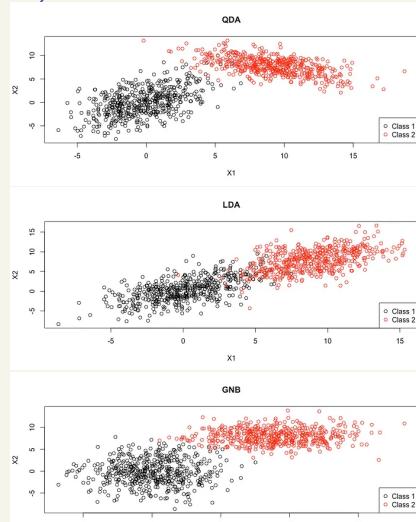
Modelos Bayesianos

9 / 38

Gaussian Naive Bayes (GNB)

Diferencias entre QDA, LDA y GNB

- QDA acepta como Σ_k cualquier conjunto de matrices definidas positivas.
- LDA acepta como Σ_k cualquier matriz pero todas iguales.
- GNB permite tener matrices Σ_k diferentes pero todas diagonales.



Gaussian Naive Bayes (GNB)

QDA

$$\Sigma_k = \frac{1}{|\mathcal{D}_k| - 1} \sum_{x \in \mathcal{D}_k} (x - \mu^{(k)}) (x - \mu^{(k)})^T$$

LDA

$$\Sigma = \frac{1}{n - K} \sum_{k=1}^K (|\mathcal{D}_k| - 1) \Sigma_k$$

GNB

$$\Sigma_k = \text{DIAG}(\sigma_1^{2(k)}, \dots, \sigma_d^{2(k)}), \quad \sigma_j^{2(k)} = \frac{1}{|\mathcal{D}_k| - 1} \sum_{x \in \mathcal{D}_k} (x_j - \mu_j^{(k)})^2$$

TPS

Matías Vera

Modelos Bayesianos

10 / 38

TPS

Matías Vera

Modelos Bayesianos

11 / 38

Outline

1 Inferencia Bayesiana

2 Naive Bayes

3 Multinomial Naive Bayes

4 Variational Bayes

5 Técnicas de Muestreo

Multinomial Naive Bayes

Multinomial Naive Bayes (MNB)

Sea $\pi = \{c_1, \dots, c_K\}$ y $\theta^{(k)} = \{\theta_1^{(k)}, \dots, \theta_V^{(k)}\}$, se modela como $Y \sim \text{Cat}(\{c_1, \dots, c_K\})$ y $X_j|Y=k \sim \text{Cat}(\{\theta_1^{(k)}, \dots, \theta_V^{(k)}\})$, utilizando estimadores puntuales.

TPS

Matías Vera

Modelos Bayesianos

12 / 38

TPS

Matías Vera

Modelos Bayesianos

13 / 38

Multinomial Naive Bayes

Multinomial Naive Bayes (MNB)

Sea $\pi = \{c_1, \dots, c_K\}$ y $\theta^{(k)} = \{\theta_1^{(k)}, \dots, \theta_V^{(k)}\}$, se modela como $Y \sim \text{Cat}(\{c_1, \dots, c_K\})$ y $X_j|Y=k \sim \text{Cat}(\{\theta_1^{(k)}, \dots, \theta_V^{(k)}\})$, utilizando estimadores puntuales.

Inferencia

$$p(y|\mathbf{x}) \propto c_y \prod_{j=1}^d \theta_{x_j}^{(y)} = c_y \prod_{m=1}^V (\theta_m^{(y)})^{N_m}$$

con N_m : Cantidad de predictores con valor m .

Multinomial Naive Bayes

Multinomial Naive Bayes (MNB)

Sea $\pi = \{c_1, \dots, c_K\}$ y $\theta^{(k)} = \{\theta_1^{(k)}, \dots, \theta_V^{(k)}\}$, se modela como $Y \sim \text{Cat}(\{c_1, \dots, c_K\})$ y $X_j|Y=k \sim \text{Cat}(\{\theta_1^{(k)}, \dots, \theta_V^{(k)}\})$, utilizando estimadores puntuales.

Inferencia

$$p(y|\mathbf{x}) \propto c_y \prod_{j=1}^d \theta_{x_j}^{(y)} = c_y \prod_{m=1}^V (\theta_m^{(y)})^{N_m}$$

con N_m : Cantidad de predictores con valor m .

Sobre las variables contadoras

Sea $\mathbf{N} = (N_1, \dots, N_V)$, es sencillo notar que $\sum_{m=1}^V N_m = d$ y $\mathbf{N}|Y=k \sim \mathcal{M}_n(d, [\theta_1^{(k)}, \dots, \theta_V^{(k)}])$.

TPS

Matías Vera

Modelos Bayesianos

13 / 38

TPS

Matías Vera

Modelos Bayesianos

13 / 38

Multinomial Naive Bayes

En inferencia se elige el máximo de una transformación afín

$$\arg \max_y p(y|\mathbf{x}) = \arg \max_y \log(c_y) + \sum_{m=1}^V N_m \log(\theta_m^{(y)})$$

Multinomial Naive Bayes

En inferencia se elige el máximo de una transformación afín

$$\arg \max_y p(y|\mathbf{x}) = \arg \max_y \log(c_y) + \sum_{m=1}^V N_m \log(\theta_m^{(y)})$$

Probabilidades de las Clases

Los parámetros c_1, \dots, c_K son estimados por máxima verosimilitud como:

$$\hat{c}_k = \frac{\#\{y_i = k\}}{n}$$

TPS

Matías Vera

Modelos Bayesianos

14 / 38

TPS

Matías Vera

Modelos Bayesianos

14 / 38

Multinomial Naive Bayes

En inferencia se elige el máximo de una transformación afín

$$\arg \max_y p(y|\mathbf{x}) = \arg \max_y \log(c_y) + \sum_{m=1}^V N_m \log(\theta_m^{(y)})$$

Probabilidades de las Clases

Los parámetros c_1, \dots, c_K son estimados por máxima verosimilitud como:

$$\hat{c}_k = \frac{\#\{y_i = k\}}{n}$$

Sobre el valor d

Cada muestra puede poseer un valor de d diferente. Eso es típico en texto, donde cada documento posee una cantidad diferente de palabras.

TPS

Matías Vera

Modelos Bayesianos

14 / 38

TPS

Matías Vera

Modelos Bayesianos

15 / 38

Multinomial Naive Bayes

Estimación de $\theta_m^{(k)}$

Se cuenta con datos $\{(\mathbf{N}_i, y_i)\}_{i=1}^n$. Sin embargo, para cada clase k se utilizarán solamente los datos con $\{y_i = k\}$ distribuidos como una multinomial de probabilidades $\theta_1^{(k)}, \dots, \theta_V^{(k)}$. A su vez, dado que las variables N_m cuentan ocurrencias, puedo compactar todas las muestras de entrenamiento de cada clase en una sola (suficiencia estadística).

$$\tilde{N}_m^{(k)} = \sum_{i=1}^n N_{i,m} \cdot \mathbb{1}\{y_i = k\}$$

Modelado: Estimador Bayesiano

Como modelado para el entrenamiento se supone $\mathbf{T} \sim \text{Dir}([\alpha_1, \dots, \alpha_V])$ y $(\tilde{N}_1^{(k)}, \dots, \tilde{N}_V^{(k)})|_{\mathbf{T}=\Theta} \sim \mathcal{M}_n(\tilde{d}^{(k)}, [\theta_1^{(k)}, \dots, \theta_V^{(k)}])$.

TPS

Matías Vera

Modelos Bayesianos

15 / 38

Multinomial Naive Bayes

Dirichlett Distribution

El vector aleatorio $(T_1, \dots, T_V) \sim \text{Dir}([\alpha_1, \dots, \alpha_V])$ puede ser pensado como una beta multivariada. Su densidad es de la forma

$$p(\theta_1, \dots, \theta_V) = \frac{\prod_{m=1}^V \Gamma(\alpha_m)}{\Gamma(\sum_{m=1}^V \alpha_m)} \left(\prod_{m=1}^V \theta_m^{\alpha_m-1} \right) \cdot \mathbb{1} \left\{ \sum_{m=1}^V \theta_m = 1, \theta_m \geq 0 \right\}$$

con sus marginales $T_m \sim \beta(\alpha_m, \sum_{\eta \neq m} \alpha_\eta)$.

Sobre la beta

Recordar que si $T \sim \beta(a, b)$, entonces $\mathbb{E}[T] = \frac{a}{a+b}$.

Multinomial Naive Bayes

Distribución a Posteriori

$$\begin{aligned} p(\theta_1^{(k)}, \dots, \theta_V^{(k)} | \tilde{N}_1^{(k)}, \dots, \tilde{N}_V^{(k)}) \\ \propto P(\tilde{N}_1^{(k)}, \dots, \tilde{N}_V^{(k)} | \theta_1^{(k)}, \dots, \theta_V^{(k)}) \cdot p(\theta_1^{(k)}, \dots, \theta_V^{(k)}) \\ \propto \left(\prod_{m=1}^V (\theta_m^{(k)})^{\tilde{N}_m^{(k)}} \right) \left(\prod_{m=1}^V (\theta_m^{(k)})^{\alpha_m-1} \cdot \mathbb{1}\{\theta_m^{(k)} \geq 0\} \right) \cdot \mathbb{1}\left\{ \sum_{m=1}^V \theta_m^{(k)} = 1 \right\} \end{aligned}$$

con lo cuál $\mathbf{T}|_{\tilde{N}_1^{(k)}, \dots, \tilde{N}_V^{(k)}} \sim \text{Dir}([\tilde{N}_1^{(k)} + \alpha_1, \dots, \tilde{N}_V^{(k)} + \alpha_V])$

TPS

Matías Vera

Modelos Bayesianos

17 / 38

Multinomial Naive Bayes

Distribución a Posteriori

$$\begin{aligned} p(\theta_1^{(k)}, \dots, \theta_V^{(k)} | \tilde{N}_1^{(k)}, \dots, \tilde{N}_V^{(k)}) \\ \propto P(\tilde{N}_1^{(k)}, \dots, \tilde{N}_V^{(k)} | \theta_1^{(k)}, \dots, \theta_V^{(k)}) \cdot p(\theta_1^{(k)}, \dots, \theta_V^{(k)}) \\ \propto \left(\prod_{m=1}^V (\theta_m^{(k)})^{\tilde{N}_m^{(k)}} \right) \left(\prod_{m=1}^V (\theta_m^{(k)})^{\alpha_m-1} \cdot \mathbb{1}\{\theta_m^{(k)} \geq 0\} \right) \cdot \mathbb{1}\left\{ \sum_{m=1}^V \theta_m^{(k)} = 1 \right\} \end{aligned}$$

con lo cuál $\mathbf{T}|_{\tilde{N}_1^{(k)}, \dots, \tilde{N}_V^{(k)}} \sim \text{Dir}([\tilde{N}_1^{(k)} + \alpha_1, \dots, \tilde{N}_V^{(k)} + \alpha_V])$

Estimador Bayesiano

$$\hat{\theta}_m^{(k)} = \mathbb{E}[T_m | \tilde{N}_1^{(k)}, \dots, \tilde{N}_V^{(k)}] = \frac{\tilde{N}_m^{(k)} + \alpha_m}{\sum_{\eta=1}^V \tilde{N}_\eta^{(k)} + \alpha_\eta}$$

TPS

Matías Vera

Modelos Bayesianos

17 / 38

TPS

Matías Vera

Modelos Bayesianos

17 / 38

Outline

1 Inferencia Bayesiana

2 Naive Bayes

3 Multinomial Naive Bayes

4 Variational Bayes

5 Técnicas de Muestreo

Variational Bayes

Variational Bayes

La idea es considerar los parámetros como parte del espacio latente. Sea \mathbf{Z} un vector no observable del problema, en general será prohibitivo calcular la distribución a posteriori $p(\mathbf{z}|\mathbf{x})$. Con lo cuál, uno approximará dicha distribución con la solución de

$$\begin{aligned}\arg \min_{q \in \mathcal{P}} \text{KL}(q(\cdot|\mathbf{x}) \| p(\cdot|\mathbf{x})) &= \arg \max_{q \in \mathcal{P}} H(q(\cdot|\mathbf{x})) + \mathbb{E}_q [\log p(\mathbf{x}, \mathbf{Z}) | \mathbf{X} = \mathbf{x}] \\ &= \arg \max_{q \in \mathcal{P}} \text{ELBO}(q)\end{aligned}$$

donde $q(\mathbf{z}|\mathbf{x})$ cumple ciertas restricciones \mathcal{P} .

TPS

Matías Vera

Modelos Bayesianos

18 / 38

TPS

Matías Vera

Modelos Bayesianos

19 / 38

Variational Bayes

Variational Bayes

La idea es considerar los parámetros como parte del espacio latente. Sea \mathbf{Z} un vector no observable del problema, en general será prohibitivo calcular la distribución a posteriori $p(\mathbf{z}|\mathbf{x})$. Con lo cuál, uno approximará dicha distribución con la solución de

$$\begin{aligned}\arg \min_{q \in \mathcal{P}} \text{KL}(q(\cdot|\mathbf{x}) \| p(\cdot|\mathbf{x})) &= \arg \max_{q \in \mathcal{P}} H(q(\cdot|\mathbf{x})) + \mathbb{E}_q [\log p(\mathbf{x}, \mathbf{Z}) | \mathbf{X} = \mathbf{x}] \\ &= \arg \max_{q \in \mathcal{P}} \text{ELBO}(q)\end{aligned}$$

donde $q(\mathbf{z}|\mathbf{x})$ cumple ciertas restricciones \mathcal{P} .

Mean field approximation

Aproximación que relaja el problema al suponer que q se puede factorizar como productos de densidades tratables. Por ejemplo, sea $\mathbf{z} = (\mathbf{u}, \phi)$ se relaja el problema suponiendo $q(\mathbf{z}|\mathbf{x}) = q_1(\mathbf{u}|\mathbf{x})q_2(\phi|\mathbf{x})$ para todo $q \in \mathcal{P}$.

TPS

Matías Vera

Modelos Bayesianos

19 / 38

TPS

Matías Vera

Modelos Bayesianos

20 / 38

Variational Bayes

Planteo del problema

Se buscan q_1 y q_2 que maximicen

$\text{ELBO}(q)$

$$\begin{aligned}&= H(q_1(\cdot|\mathbf{x})) + H(q_2(\cdot|\mathbf{x})) + \int_{\mathcal{U}} q_1(\mathbf{u}|\mathbf{x}) \left(\int_{\Phi} q_2(\phi|\mathbf{x}) \log p(\mathbf{x}, \mathbf{u}, \phi) d\phi \right) d\mathbf{u} \\ &= H(q_1(\cdot|\mathbf{x})) + H(q_2(\cdot|\mathbf{x})) + \int_{\Phi} q_2(\phi|\mathbf{x}) \left(\int_{\mathcal{U}} q_1(\mathbf{u}|\mathbf{x}) \log p(\mathbf{x}, \mathbf{u}, \phi) d\mathbf{u} \right) d\phi\end{aligned}$$

Variational Bayes

Planteo del problema

Se buscan q_1 y q_2 que maximicen

$\text{ELBO}(q)$

$$\begin{aligned}&= H(q_1(\cdot|\mathbf{x})) + H(q_2(\cdot|\mathbf{x})) + \int_{\mathcal{U}} q_1(\mathbf{u}|\mathbf{x}) \left(\int_{\Phi} q_2(\phi|\mathbf{x}) \log p(\mathbf{x}, \mathbf{u}, \phi) d\phi \right) d\mathbf{u} \\ &= H(q_1(\cdot|\mathbf{x})) + H(q_2(\cdot|\mathbf{x})) + \int_{\Phi} q_2(\phi|\mathbf{x}) \left(\int_{\mathcal{U}} q_1(\mathbf{u}|\mathbf{x}) \log p(\mathbf{x}, \mathbf{u}, \phi) d\mathbf{u} \right) d\phi\end{aligned}$$

Resolución Iterativa

Se simplificará el problema resolviéndolo de forma iterativa. Definimos

$$\begin{aligned}E_1(\mathbf{x}, \mathbf{u}) &= \int_{\Phi} q_2(\phi|\mathbf{x}) \log p(\mathbf{x}, \mathbf{u}, \phi) d\phi \equiv f(q_2) \\ E_2(\mathbf{x}, \phi) &= \int_{\mathcal{U}} q_1(\mathbf{u}|\mathbf{x}) \log p(\mathbf{x}, \mathbf{u}, \phi) d\mathbf{u} \equiv f(q_1)\end{aligned}$$

TPS

Matías Vera

Modelos Bayesianos

20 / 38

TPS

Matías Vera

Modelos Bayesianos

21 / 38

Variational Bayes

Problemas

$$q_1(\cdot|\mathbf{x}) = \arg \max_{q_1} \int_{\mathcal{U}} q_1(\mathbf{u}|\mathbf{x}) \log \frac{e^{E_1(\mathbf{x}, \mathbf{u})}}{q_1(\mathbf{u}|\mathbf{x})} d\mathbf{u}$$

$$q_2(\cdot|\mathbf{x}) = \arg \max_{q_2} \int_{\Phi} q_2(\phi|\mathbf{x}) \log \frac{e^{E_2(\mathbf{x}, \phi)}}{q_2(\phi|\mathbf{x})} d\phi$$

Variational Bayes

Problemas

$$q_1(\cdot|\mathbf{x}) = \arg \max_{q_1} \int_{\mathcal{U}} q_1(\mathbf{u}|\mathbf{x}) \log \frac{e^{E_1(\mathbf{x}, \mathbf{u})}}{q_1(\mathbf{u}|\mathbf{x})} d\mathbf{u}$$

$$q_2(\cdot|\mathbf{x}) = \arg \max_{q_2} \int_{\Phi} q_2(\phi|\mathbf{x}) \log \frac{e^{E_2(\mathbf{x}, \phi)}}{q_2(\phi|\mathbf{x})} d\phi$$

Soluciones

La solución consiste en iterar entre

$$q_1(\mathbf{u}|\mathbf{x}) \propto e^{E_1(\mathbf{x}, \mathbf{u})}, \quad q_2(\phi|\mathbf{x}) \propto e^{E_2(\mathbf{x}, \phi)}$$

Variational Bayes

Problemas

$$q_1(\cdot|\mathbf{x}) = \arg \max_{q_1} \int_{\mathcal{U}} q_1(\mathbf{u}|\mathbf{x}) \log \frac{e^{E_1(\mathbf{x}, \mathbf{u})}}{q_1(\mathbf{u}|\mathbf{x})} d\mathbf{u}$$

$$q_2(\cdot|\mathbf{x}) = \arg \max_{q_2} \int_{\Phi} q_2(\phi|\mathbf{x}) \log \frac{e^{E_2(\mathbf{x}, \phi)}}{q_2(\phi|\mathbf{x})} d\phi$$

Soluciones

La solución consiste en iterar entre

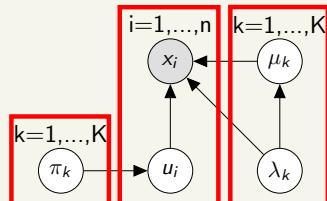
$$q_1(\mathbf{u}|\mathbf{x}) \propto e^{E_1(\mathbf{x}, \mathbf{u})}, \quad q_2(\phi|\mathbf{x}) \propto e^{E_2(\mathbf{x}, \phi)}$$

Sobre el ELBO

$$\text{ELBO}(q) = \log p(\mathbf{x}) - \text{KL}(q(\cdot|\mathbf{x})||p(\cdot|\mathbf{x})) \leq \log p(\mathbf{x})$$

$$\text{ELBO}(q) = \mathbb{E}_q [\log p(\mathbf{x}|\mathbf{z})|\mathbf{x}] - \text{KL}(q(\cdot|\mathbf{x})||p(\cdot)) \leq \mathbb{E}_q [\log p(\mathbf{x}|\mathbf{z})|\mathbf{x}]$$

Gaussian Variational Bayes



Modelo

$$p(\mathbf{x}, \mathbf{u}, \pi, \lambda, \mu) = p(\pi) \left(\prod_{k=1}^K p(\lambda_k) p(\mu_k | \lambda_k) \right) \left(\prod_{i=1}^n P(u_i | \pi) p(x_i | u_i, \mu, \lambda) \right)$$

con

$$\begin{aligned} \pi &\sim \text{Dir}(\alpha), & \lambda_k &\sim \Gamma(\nu, \beta), & \mu_k | \lambda_k &\sim \mathcal{N}(m, (\delta \lambda_k)^{-1}) \\ u | \pi &\sim \text{Cat}(\pi), & x | u, \mu, \lambda &\sim \mathcal{N}(\mu_u, \lambda_u^{-1}) \end{aligned}$$

Gaussian Variational Bayes

Mean field approximation

Se aproxima $q(\mathbf{u}, \pi, \lambda, \mu | \mathbf{x}) = Q_1(\mathbf{u}|\mathbf{x})q_2(\pi, \lambda, \mu | \mathbf{x})$ y luego

$$E_1(\mathbf{x}, \mathbf{u}) = \text{cte} + \sum_{i=1}^n \int q_2(\pi | \mathbf{x}) \log P(u_i | \pi) d\pi$$

$$+ \sum_{i=1}^n \int \int q_2(\mu, \lambda | \mathbf{x}) \log p(x_i | u_i, \mu, \lambda) d\mu d\lambda$$

$$\begin{aligned} E_2(\mathbf{x}, \pi, \lambda, \mu) &= \log p(\pi) + \sum_{k=1}^K \log p(\lambda_k) + \sum_{k=1}^K \log p(\mu_k | \lambda_k) \\ &+ \sum_{i=1}^n \sum_{k=1}^K Q_1(u_i = k | \mathbf{x}) \log P(u_i = k | \pi) \\ &+ \sum_{i=1}^n \sum_{k=1}^K Q_1(u_i = k | \mathbf{x}) \log p(x_i | u_i = k, \mu, \lambda) \end{aligned}$$

Gaussian Variational Bayes

Cálculo de $q_2(\pi, \lambda, \mu | \mathbf{x})$

Lo primero a notar es la factorización. Sea $\gamma_{i,k} = Q_1(u_i = k | \mathbf{x})$, luego

$$q_2(\pi, \lambda, \mu | \mathbf{x}) \propto p(\pi) \left(\prod_{k=1}^K p(\lambda_k) p(\mu_k | \lambda_k) \right) \prod_{k=1}^K e^{\sum_{i=1}^n \gamma_{i,k} [\log \pi_k + \log \mathcal{N}_{x_i}(\mu_k, \lambda_k^{-1})]}$$

Luego $q_2(\pi, \lambda, \mu | \mathbf{x}) = q_2(\pi | \mathbf{x}) \prod_{k=1}^K q_2(\mu_k, \lambda_k | \mathbf{x})$.

Gaussian Variational Bayes

Cálculo de $q_2(\pi, \lambda, \mu | \mathbf{x})$

Lo primero a notar es la factorización. Sea $\gamma_{i,k} = Q_1(u_i = k | \mathbf{x})$, luego

$$q_2(\pi, \lambda, \mu | \mathbf{x}) \propto p(\pi) \left(\prod_{k=1}^K p(\lambda_k) p(\mu_k | \lambda_k) \right) \prod_{k=1}^K e^{\sum_{i=1}^n \gamma_{i,k} [\log \pi_k + \log \mathcal{N}_{x_i}(\mu_k, \lambda_k^{-1})]}$$

Luego $q_2(\pi, \lambda, \mu | \mathbf{x}) = q_2(\pi | \mathbf{x}) \prod_{k=1}^K q_2(\mu_k, \lambda_k | \mathbf{x})$.

Cálculo de $q_2(\pi | \mathbf{x})$

$$q_2(\pi | \mathbf{x}) \propto \left(\prod_{k=1}^K \pi_k^{\alpha_k - 1} e^{\sum_{i=1}^n \gamma_{i,k} \log \pi_k} \right) \mathbb{1} \left\{ \sum_{k=1}^K \pi_k = 1, \pi_k \geq 0 \right\}$$

con lo cual $\pi | \mathbf{x} \sim \text{Dir}([\alpha_1 + \sum_{i=1}^n \gamma_{i,1}, \dots, \alpha_K + \sum_{i=1}^n \gamma_{i,K}])$

Gaussian Variational Bayes

Cálculo de $q_2(\mu_k, \lambda_k | \mathbf{x})$

$$q_2(\mu_k, \lambda_k | \mathbf{x}) \propto \underbrace{\lambda_k^{\nu-1} e^{-\beta \lambda_k} \mathbb{1}\{\lambda_k > 0\}}_{\propto p(\lambda_k)} \underbrace{\lambda_k^{1/2} e^{-\frac{\delta \lambda_k (\mu_k - m)^2}{2}}}_{\propto p(\mu_k | \lambda_k)} \lambda_k^{\frac{1}{2} \sum_{i=1}^n \gamma_{i,k}} e^{-\frac{\lambda_k \sum_{i=1}^n \gamma_{i,k} (x_i - \mu_k)^2}{2}}$$

con lo cual $\mu_k | \lambda_k, \mathbf{x} \sim \mathcal{N} \left(\frac{\delta m + \sum_{i=1}^n \gamma_{i,k} x_i}{\delta + \sum_{i=1}^n \gamma_{i,k}}, \frac{1}{\lambda_k (\delta + \sum_{i=1}^n \gamma_{i,k})} \right)$ y
 $\lambda_k | \mathbf{x} \sim \Gamma \left(\nu + \frac{1}{2} \sum_{i=1}^n \gamma_{i,k}, \beta + \frac{\delta m^2}{2} + \frac{1}{2} \sum_{i=1}^n \gamma_{i,k} x_i^2 - \frac{(\delta m + \sum_{i=1}^n \gamma_{i,k} x_i)^2}{2(\delta + \sum_{i=1}^n \gamma_{i,k})} \right)$

Gaussian Variational Bayes

Cálculo de $q_2(\mu_k, \lambda_k | \mathbf{x})$

$$q_2(\mu_k, \lambda_k | \mathbf{x}) \propto \underbrace{\lambda_k^{\nu-1} e^{-\beta \lambda_k} \mathbb{1}\{\lambda_k > 0\}}_{\propto p(\lambda_k)} \underbrace{\lambda_k^{1/2} e^{-\frac{\delta \lambda_k (\mu_k - m)^2}{2}}}_{\propto p(\mu_k | \lambda_k)} \lambda_k^{\frac{1}{2} \sum_{i=1}^n \gamma_{i,k}} e^{-\frac{\lambda_k \sum_{i=1}^n \gamma_{i,k} (x_i - \mu_k)^2}{2}}$$

con lo cual $\mu_k | \lambda_k, \mathbf{x} \sim \mathcal{N} \left(\frac{\delta m + \sum_{i=1}^n \gamma_{i,k} x_i}{\delta + \sum_{i=1}^n \gamma_{i,k}}, \frac{1}{\lambda_k (\delta + \sum_{i=1}^n \gamma_{i,k})} \right)$ y
 $\lambda_k | \mathbf{x} \sim \Gamma \left(\nu + \frac{1}{2} \sum_{i=1}^n \gamma_{i,k}, \beta + \frac{\delta m^2}{2} + \frac{1}{2} \sum_{i=1}^n \gamma_{i,k} x_i^2 - \frac{(\delta m + \sum_{i=1}^n \gamma_{i,k} x_i)^2}{2(\delta + \sum_{i=1}^n \gamma_{i,k})} \right)$

TPS

Matías Vera

Modelos Bayesianos

25 / 38

TPS

Matías Vera

Modelos Bayesianos

25 / 38

Gaussian Variational Bayes

Propiedad de las distribuciones Gamma y Dirichlett

Sean $\lambda \sim \Gamma(\nu, \beta)$ y $\pi \sim \text{Dir}(\alpha)$, se puede demostrar que

- $\mathbb{E}[\log \lambda] = \psi(\nu) - \log(\beta)$
- $\mathbb{E}[\log \pi_k] = \psi(\alpha_k) - \psi \left(\sum_{c=1}^K \alpha_c \right)$

donde $\psi(\cdot)$ es la función digamma $\psi(z) = \frac{\Gamma'(z)}{\Gamma(z)}$.

Gaussian Variational Bayes

Propiedad de las distribuciones Gamma y Dirichlett

Sean $\lambda \sim \Gamma(\nu, \beta)$ y $\pi \sim \text{Dir}(\alpha)$, se puede demostrar que

- $\mathbb{E}[\log \lambda] = \psi(\nu) - \log(\beta)$
- $\mathbb{E}[\log \pi_k] = \psi(\alpha_k) - \psi \left(\sum_{c=1}^K \alpha_c \right)$

donde $\psi(\cdot)$ es la función digamma $\psi(z) = \frac{\Gamma'(z)}{\Gamma(z)}$.

TPS

Matías Vera

Modelos Bayesianos

26 / 38

TPS

Matías Vera

Modelos Bayesianos

26 / 38

Gaussian Variational Bayes

Cálculo de $Q_1(u_i = k | \mathbf{x})$

Sean los parámetros de q_2 definidos como $\pi | \mathbf{x} \sim \text{Dir}(\alpha^*)$, $\mu_k | \lambda_k, \mathbf{x} \sim \mathcal{N}(m_k^*, (\delta_k^* \lambda_k)^{-1})$ y $\lambda_k | \mathbf{x} \sim \Gamma(\nu_k^*, \beta_k^*)$. Luego

$$Q_1(u_i = k | \mathbf{x}) \propto e^{\psi(\alpha_k^*) - \psi \left(\sum_{c=1}^K \alpha_c^* \right) + \frac{\psi(\nu_k^*) - \log(\beta_k^*)}{2} - \frac{1}{2} \mathbb{E}_{q_2} [\lambda_k (x_i - \mu_k)^2]}$$

con

$$\begin{aligned} \mathbb{E}_{q_2} [\lambda_k (x_i - \mu_k)^2] &= \mathbb{E}_{q_2} [\lambda_k \mathbb{E}_{q_2} [(x_i - \mu_k)^2 | \lambda_k]] \\ &= \mathbb{E}_{q_2} [\lambda_k (\mathbb{E}_{q_2} [(\mu_k - m_k^*)^2 | \lambda_k] + (m_k^* - x_i)^2)] \\ &= \frac{1}{\delta_k^*} + \frac{\nu_k^*}{\beta_k^*} (m_k^* - x_i)^2 \end{aligned}$$

Finalmente

$$Q_1(u_i = k | \mathbf{x}) \propto e^{\psi(\alpha_k^*) - \psi \left(\sum_{c=1}^K \alpha_c^* \right) + \frac{\psi(\nu_k^*) - \log(\beta_k^*)}{2} - \frac{1}{2\delta_k^*} - \frac{\nu_k^*}{2\beta_k^*} (m_k^* - x_i)^2}$$

Gaussian Variational Bayes

Solución: Inicializar $\gamma_{i,k}$ con EM e iterar entre

- Calcular $(\alpha_k^*, m_k^*, \delta_k^*, \nu_k^*, \beta_k^*)$ a partir de $\gamma_{i,k}$ como

$$\alpha_k^* = \alpha_k + \sum_{i=1}^n \gamma_{i,k}, \quad m_k^* = \frac{\delta m + \sum_{i=1}^n \gamma_{i,k} x_i}{\delta + \sum_{i=1}^n \gamma_{i,k}}$$

$$\delta_k^* = \delta + \sum_{i=1}^n \gamma_{i,k}, \quad \nu_k^* = \nu + \frac{1}{2} \sum_{i=1}^n \gamma_{i,k}$$

$$\beta_k^* = \beta + \frac{\delta m^2}{2} + \frac{1}{2} \sum_{i=1}^n \gamma_{i,k} x_i^2 - \frac{(\delta m + \sum_{i=1}^n \gamma_{i,k} x_i)^2}{2(\delta + \sum_{i=1}^n \gamma_{i,k})}$$

- Calcular $\gamma_{i,k} = \frac{\rho_{i,k}}{\sum_{c=1}^K \rho_{i,c}}$ a partir de $(\alpha^*, m_k^*, \delta_k^*, \nu_k^*, \beta_k^*)$ como

$$\rho_{i,k} = e^{\psi(\alpha_k^*) - \psi \left(\sum_{c=1}^K \alpha_c^* \right) + \frac{\psi(\nu_k^*) - \log(\beta_k^*)}{2} - \frac{1}{2\delta_k^*} - \frac{\nu_k^*}{2\beta_k^*} (m_k^* - x_i)^2}$$

TPS

Matías Vera

Modelos Bayesianos

27 / 38

TPS

Matías Vera

Modelos Bayesianos

28 / 38

Gaussian Variational Bayes

Predictiva: Es una mezcla!

$$\begin{aligned} p(x_{\text{test}} | \mathcal{D}_n) &= \mathbb{E}[p(x_{\text{test}} | \phi) | \mathcal{D}_n] \\ &= \sum_{k=1}^K \mathbb{E}[\pi_k | \mathcal{D}_n] \cdot \mathbb{E}\left[\sqrt{\frac{\lambda_k}{2\pi}} e^{-\frac{\lambda_k}{2}(x_{\text{test}} - \mu_k)^2} \middle| \mathcal{D}_n\right] \\ &= \sum_{k=1}^K \frac{\alpha_k^*}{\sum_{c=1}^K \alpha_c^*} \cdot \tilde{p}_k(x_{\text{test}} | \mathcal{D}_n) \end{aligned}$$

Gaussian Variational Bayes

Predictiva: Es una mezcla!

$$\begin{aligned} p(x_{\text{test}} | \mathcal{D}_n) &= \mathbb{E}[p(x_{\text{test}} | \phi) | \mathcal{D}_n] \\ &= \sum_{k=1}^K \mathbb{E}[\pi_k | \mathcal{D}_n] \cdot \mathbb{E}\left[\sqrt{\frac{\lambda_k}{2\pi}} e^{-\frac{\lambda_k}{2}(x_{\text{test}} - \mu_k)^2} \middle| \mathcal{D}_n\right] \\ &= \sum_{k=1}^K \frac{\alpha_k^*}{\sum_{c=1}^K \alpha_c^*} \cdot \tilde{p}_k(x_{\text{test}} | \mathcal{D}_n) \end{aligned}$$

Normal-Gamma Distribution

$$1 = \int_0^\infty \int_{-\infty}^\infty \frac{\beta^\nu}{\Gamma(\nu)} \lambda^{\nu-1} e^{-\beta\lambda} \sqrt{\frac{\delta\lambda}{2\pi}} e^{-\frac{\delta\lambda}{2}(\mu-m)^2} d\mu d\lambda$$

$$\int_0^\infty \int_{-\infty}^\infty \frac{1}{\sqrt{2\pi}} \lambda^{\nu-\frac{1}{2}} e^{-\frac{\delta\lambda}{2}(\mu-m)^2 - \lambda\beta} d\mu d\lambda = \frac{\Gamma(\nu)}{\beta^\nu \sqrt{\delta}}$$

Gaussian Variational Bayes

Calculo auxiliar de las densidades a mezclar

$$\begin{aligned} &\mathbb{E}\left[\sqrt{\frac{\lambda_k}{2\pi}} e^{-\frac{\lambda_k}{2}(x_{\text{test}} - \mu_k)^2} \middle| \mathcal{D}_n\right] \\ &= \int_0^\infty \int_{-\infty}^\infty \sqrt{\frac{\lambda}{2\pi}} e^{-\frac{\lambda}{2}(x_{\text{test}} - \mu)^2} \frac{\beta_k^{*\nu_k^*} \sqrt{\delta_k^*}}{\sqrt{2\pi} \Gamma(\nu_k^*)} \lambda^{\nu_k^* - \frac{1}{2}} e^{-\frac{\delta_k^*\lambda}{2}(\mu - m_k^*)^2 - \lambda\beta_k^*} d\mu d\lambda \\ &\propto \int_0^\infty \int_{-\infty}^\infty \sqrt{\frac{1}{2\pi}} \lambda^{\nu_k^*} e^{-\frac{\lambda}{2}(x_{\text{test}} - \mu)^2 - \frac{\delta_k^*\lambda}{2}(\mu - m_k^*)^2 - \lambda\beta_k^*} d\mu d\lambda \end{aligned}$$

Gaussian Variational Bayes

Calculo auxiliar de las densidades a mezclar

$$\begin{aligned} &\mathbb{E}\left[\sqrt{\frac{\lambda_k}{2\pi}} e^{-\frac{\lambda_k}{2}(x_{\text{test}} - \mu_k)^2} \middle| \mathcal{D}_n\right] \\ &= \int_0^\infty \int_{-\infty}^\infty \sqrt{\frac{\lambda}{2\pi}} e^{-\frac{\lambda}{2}(x_{\text{test}} - \mu)^2} \frac{\beta_k^{*\nu_k^*} \sqrt{\delta_k^*}}{\sqrt{2\pi} \Gamma(\nu_k^*)} \lambda^{\nu_k^* - \frac{1}{2}} e^{-\frac{\delta_k^*\lambda}{2}(\mu - m_k^*)^2 - \lambda\beta_k^*} d\mu d\lambda \\ &\propto \int_0^\infty \int_{-\infty}^\infty \sqrt{\frac{1}{2\pi}} \lambda^{\nu_k^*} e^{-\frac{\lambda}{2}(x_{\text{test}} - \mu)^2 - \frac{\delta_k^*\lambda}{2}(\mu - m_k^*)^2 - \lambda\beta_k^*} d\mu d\lambda \end{aligned}$$

Gaussian Variational Bayes

Calculo auxiliar de las densidades a mezclar

$$\begin{aligned} &\mathbb{E}\left[\sqrt{\frac{\lambda_k}{2\pi}} e^{-\frac{\lambda_k}{2}(x_{\text{test}} - \mu_k)^2} \middle| \mathcal{D}_n\right] \propto \left(\beta_k^* + \frac{\delta_k^*(x_{\text{test}} - m_k^*)^2}{2(\delta_k^* + 1)}\right)^{-(\nu_k^* + 1/2)} \\ &\propto \left(1 + \frac{\delta_k^* \nu_k^*}{(\delta_k^* + 1)\beta_k^*} \frac{(x_{\text{test}} - m_k^*)^2}{2\nu_k^*}\right)^{-\frac{2\nu_k^* + 1}{2}} \end{aligned}$$

Gaussian Variational Bayes

Calculo auxiliar de las densidades a mezclar

$$\begin{aligned} &\mathbb{E}\left[\sqrt{\frac{\lambda_k}{2\pi}} e^{-\frac{\lambda_k}{2}(x_{\text{test}} - \mu_k)^2} \middle| \mathcal{D}_n\right] \propto \left(\beta_k^* + \frac{\delta_k^*(x_{\text{test}} - m_k^*)^2}{2(\delta_k^* + 1)}\right)^{-(\nu_k^* + 1/2)} \\ &\propto \left(1 + \frac{\delta_k^* \nu_k^*}{(\delta_k^* + 1)\beta_k^*} \frac{(x_{\text{test}} - m_k^*)^2}{2\nu_k^*}\right)^{-\frac{2\nu_k^* + 1}{2}} \end{aligned}$$

Distribución t-Student Generalizada: $X \sim t(\mu, \Lambda, \nu)$ si

$$p(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})} \sqrt{\frac{\Lambda}{\pi\nu}} \left(1 + \Lambda \frac{(x - \mu)^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

Gaussian Variational Bayes

Calculo auxiliar de las densidades a mezclar

$$\begin{aligned} \mathbb{E}\left[\sqrt{\frac{\lambda_k}{2\pi}} e^{-\frac{\lambda_k}{2}(x_{\text{test}} - \mu_k)^2} \mid \mathcal{D}_n\right] &\propto \left(\beta_k^* + \frac{\delta_k^*(x_{\text{test}} - m_k^*)^2}{2(\delta_k^* + 1)}\right)^{-(\nu_k^* + 1/2)} \\ &\propto \left(1 + \frac{\delta_k^* \nu_k^*}{(\delta_k^* + 1)\beta_k^*} \frac{(x_{\text{test}} - m_k^*)^2}{2\nu_k^*}\right)^{-\frac{2\nu_k^* + 1}{2}} \end{aligned}$$

Distribución t-Student Generalizada: $X \sim t(\mu, \Lambda, \nu)$ si

$$p(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})} \sqrt{\frac{\Lambda}{\pi\nu}} \left(1 + \Lambda \frac{(x - \mu)^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

Predictiva

$$p(x_{\text{test}} | \mathcal{D}_n) = \sum_{k=1}^K \frac{\alpha_k^*}{\sum_{c=1}^K \alpha_c^*} \cdot t\left(m_k^*, \frac{\delta_k^* \nu_k^*}{(\delta_k^* + 1)\beta_k^*}, 2\nu_k^*\right)$$

TPS

Matías Vera

Modelos Bayesianos

31 / 38

TPS

Matías Vera

Modelos Bayesianos

32 / 38

Outline

1 Inferencia Bayesiana

2 Naive Bayes

3 Multinomial Naive Bayes

4 Variational Bayes

5 Técnicas de Muestreo

Monte-Carlo

Ley de los grandes números

Sean X_i variables aleatorias con esperanza finita $\mathbb{E}[X]$, entonces $\frac{1}{n} \sum_{i=1}^n X_i \xrightarrow{n \rightarrow \infty} \mathbb{E}[X]$ (w.p.1).

Monte-Carlo

Ley de los grandes números

Sean X_i variables aleatorias con esperanza finita $\mathbb{E}[X]$, entonces $\frac{1}{n} \sum_{i=1}^n X_i \xrightarrow{n \rightarrow \infty} \mathbb{E}[X]$ (w.p.1).

Método de Monte-Carlo

Sean X_i variables aleatorias i.i.d con pdf $p(x)$ o pmf $P(x)$, entonces

$$\int_{\mathbb{R}} g(x)p(x)dx \approx \frac{1}{n} \sum_{i=1}^n g(X_i), \quad \sum_{x \in \mathbb{A}} g(x)P(x) \approx \frac{1}{n} \sum_{i=1}^n g(X_i)$$

TPS

Matías Vera

Modelos Bayesianos

33 / 38

TPS

Matías Vera

Modelos Bayesianos

33 / 38

Monte-Carlo

Ley de los grandes números

Sean X_i variables aleatorias con esperanza finita $\mathbb{E}[X]$, entonces $\frac{1}{n} \sum_{i=1}^n X_i \xrightarrow{n \rightarrow \infty} \mathbb{E}[X]$ (w.p.1).

Técnicas de Muestreo

Ideas Principales

- En lugar de obtener la distribución a posteriori, vamos a generar muestras de esta distribución.
- Vamos aproximar la predictiva por Monte-Carlo.

$$p(x_{\text{test}} | \mathcal{D}_n) \approx \frac{1}{N_s} \sum_{i=1}^{N_s} p(x_{\text{test}} | T_i)$$

Método de Monte-Carlo

Sean X_i variables aleatorias i.i.d con pdf $p(x)$ o pmf $P(x)$, entonces

$$\int_{\mathbb{R}} g(x)p(x)dx \approx \frac{1}{n} \sum_{i=1}^n g(X_i), \quad \sum_{x \in \mathbb{A}} g(x)P(x) \approx \frac{1}{n} \sum_{i=1}^n g(X_i)$$

Algunas variantes interesantes

- $\int_a^b g(x)dx \approx \frac{b-a}{n} \sum_{i=1}^n g(X_i)$ con $X_i \sim U(a, b)$.
- $\int_a^b g(x)dx \approx \frac{1}{kn} \sum_{i=1}^n \mathbb{1}\{a < X_i < b\}$ con X_i de pdf $p(x) = k \cdot g(x)$.
- $\int_{\mathbb{R}} g(x)p(x)dx \approx \frac{1}{n} \sum_{i=1}^n \frac{p(X_i)}{q(X_i)} g(X_i)$ con X_i de pdf $q(x)$.

TPS

Matías Vera

Modelos Bayesianos

33 / 38

TPS

Matías Vera

Modelos Bayesianos

34 / 38

Técnicas de Muestreo

Ideas Principales

- En lugar de obtener la distribución a posteriori, vamos a generar muestras de esta distribución.
- Vamos aproximar la predictiva por Monte-Carlo.

$$p(x_{\text{test}} | \mathcal{D}_n) \approx \frac{1}{N_s} \sum_{i=1}^{N_s} p(x_{\text{test}} | T_i)$$

Muestreo de Gibbs

Supongamos que, debido a su complejidad, no podemos simular muestras de $p(x, y)$; pero que si es posible generar muestras de las condicionales $p(x|y)$ y $p(y|x)$. El muestreo de Gibbs consiste en, a partir de un x_0 , iterar entre:

$$y_k \sim p(y|x_k), \quad x_{k+1} \sim p(x|y_k)$$

Luego de suficientes simulaciones (resultado asintótico), el último par (x, y) estará distribuido (aproximadamente) por $p(x, y)$.

TPS

Matías Vera

Modelos Bayesianos

34 / 38

TPS

Matías Vera

Modelos Bayesianos

35 / 38

Técnicas de Muestreo

Markov Chain Monte-Carlo (MCMC)



Implementaciones más sofisticadas

- ① Comenzar en la posición actual.
- ② Proponer mudarse a una nueva posición cercana a la actual.
- ③ Aceptar/Rechazar la nueva posición basándose en el coherence de la posición con los datos y distribuciones anteriores.
- ④ ▶ Si acepta: Pasar a la nueva posición. Regresar al Paso 1.
▶ De lo contrario: no moverse de la posición actual. Regrese al Paso 1.
- ⑤ Después de una gran cantidad de iteraciones, se reportan todas las posiciones aceptadas.

TPS

Matías Vera

Modelos Bayesianos

35 / 38

TPS

Matías Vera

Modelos Bayesianos

36 / 38

PYMC

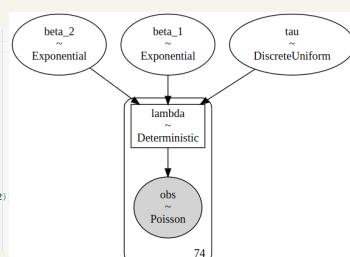
```

import pymc as pm
import numpy as np
import matplotlib.pyplot as plt

count_data = np.loadtxt("txtdata.csv")
n_count_data = len(count_data)

with pm.Model() as model:
    alpha = pm.Uniform('alpha', lower=0, upper=n_count_data)
    beta_1 = pm.Exponential('beta_1', alpha)
    beta_2 = pm.Exponential('beta_2', alpha)
    tau = pm.DiscreteUniform("tau", lower=0, upper=n_count_data - 1)
    idx = np.arange(n_count_data) # Index
    lambda_ = pm.Deterministic('lambda', pm.math.switch(tau > idx, beta_1, beta_2))
    observation = pm.Poisson("obs", lambda_, observed=count_data)
    trace = pm.sample(draws=1000, chains=2)

pm.model_to_graphviz(model)
  
```



PYMC

```

import pymc as pm
import numpy as np
import matplotlib.pyplot as plt

count_data = np.loadtxt("txtdata.csv")
n_count_data = len(count_data)

with pm.Model() as model:
    alpha = pm.Uniform('alpha', lower=0, upper=n_count_data)
    beta_1 = pm.Exponential('beta_1', alpha)
    beta_2 = pm.Exponential('beta_2', alpha)
    tau = pm.DiscreteUniform("tau", lower=0, upper=n_count_data - 1)
    idx = np.arange(n_count_data) # Index
    lambda_ = pm.Deterministic('lambda', pm.math.switch(tau > idx, beta_1, beta_2))
    observation = pm.Poisson("obs", lambda_, observed=count_data)
    trace = pm.sample(draws=1000, chains=2)

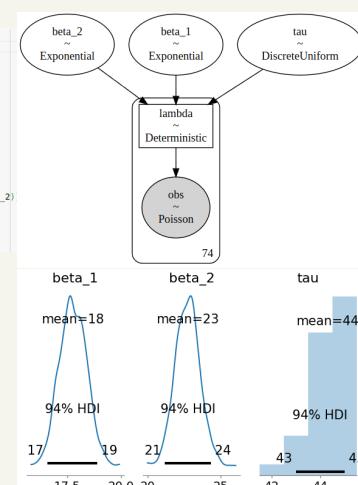
pm.model_to_graphviz(model)
  
```

```

beta_1_samples = trace.posterior['beta_1']
beta_2_samples = trace.posterior['beta_2']
tau_samples = trace.posterior['tau']
lambda_samples = trace.posterior['lambda']

with model:
    posterior_pred = pm.sample_posterior_predictive(trace)
    pred_samples = posterior_pred.posterior_predictive['obs']

pm.plot_posterior(trace.posterior[['beta_1','beta_2','tau']], figsize=(7,4))
  
```



TPS

Matías Vera

Modelos Bayesianos

37 / 38

TPS

Matías Vera

Modelos Bayesianos

37 / 38

Anexo: Distribuciones

Normal(μ, σ^2)

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Gamma(ν, β)

$$p(x) = \frac{\beta^\nu}{\Gamma(\nu)} x^{\nu-1} e^{-\beta x} \mathbb{1}\{x > 0\}$$

T-Student generalizada(μ, Λ, ν)

$$p(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})} \sqrt{\frac{\Lambda}{\pi\nu}} \left(1 + \Lambda \frac{(x-\mu)^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

Dirichlett($\alpha_1, \dots, \alpha_K$)

$$p(x_1, \dots, x_K) = \frac{\prod_{k=1}^K \Gamma(\alpha_k)}{\Gamma\left(\sum_{k=1}^K \alpha_k\right)} \left(\prod_{k=1}^K x_k^{\alpha_k-1}\right) \cdot \mathbb{1}\left\{\sum_{k=1}^K x_k = 1, x_k \geq 0\right\}$$