

# Problemas no balanceados y aprendizaje sensitivo a costos

Fernando Lozano

Universidad de los Andes

21 de noviembre de 2022



# Desbalance y costos de errores no simétricos

# Desbalance y costos de errores no simétricos

- Algoritmos estudiados asumen  $\sim$ igual proporción de datos en cada clase: **datos balanceados**

# Desbalance y costos de errores no simétricos

- Algoritmos estudiados asumen  $\sim$ igual proporción de datos en cada clase: **datos balanceados**
- Medida de error de hipótesis  $h$ :

$$e(h) = \mathbf{P}_{\mathcal{D}} [[h(x) \neq y]]$$

# Desbalance y costos de errores no simétricos

- Algoritmos estudiados asumen  $\sim$ igual proporción de datos en cada clase: **datos balanceados**
- Medida de error de hipótesis  $h$ :

$$e(h) = \mathbf{P}_{\mathcal{D}} [[h(x) \neq y]]$$

- Algoritmos minimizan función de error en los datos de manera que  $e(h) \ll$ .

# Desbalance y costos de errores no simétricos

- Algoritmos estudiados asumen  $\sim$ igual proporción de datos en cada clase: **datos balanceados**
- Medida de error de hipótesis  $h$ :

$$e(h) = \mathbf{P}_{\mathcal{D}} [[h(x) \neq y]]$$

- Algoritmos minimizan función de error en los datos de manera que  $e(h) \ll$ .
- Sin embargo:

# Desbalance y costos de errores no simétricos

- Algoritmos estudiados asumen  $\sim$ igual proporción de datos en cada clase: **datos balanceados**
- Medida de error de hipótesis  $h$ :

$$e(h) = \mathbf{P}_{\mathcal{D}} [[h(x) \neq y]]$$

- Algoritmos minimizan función de error en los datos de manera que  $e(h) \ll$ .
- Sin embargo:
  - ❶ En muchas aplicaciones los datos no están **balanceados**.

# Desbalance y costos de errores no simétricos

- Algoritmos estudiados asumen  $\sim$ igual proporción de datos en cada clase: **datos balanceados**
- Medida de error de hipótesis  $h$ :

$$e(h) = \mathbf{P}_{\mathcal{D}} [[h(x) \neq y]]$$

- Algoritmos minimizan función de error en los datos de manera que  $e(h) \ll$ .
- Sin embargo:
  - ❶ En muchas aplicaciones los datos no están **balanceados**.
  - ❷ El **costo** de cometer un error de clasificación en un dato **depende** de la clase a la cual éste pertenece.



# Desbalance y costos de errores no simétricos

- Algoritmos estudiados asumen  $\sim$ igual proporción de datos en cada clase: **datos balanceados**
- Medida de error de hipótesis  $h$ :

$$e(h) = \mathbf{P}_{\mathcal{D}} [[h(x) \neq y]]$$

- Algoritmos minimizan función de error en los datos de manera que  $e(h) \ll$ .
- Sin embargo:
  - ❶ En muchas aplicaciones los datos no están **balanceados**.
  - ❷ El **costo** de cometer un error de clasificación en un dato **depende** de la clase a la cual éste pertenece.
- En estos casos  $e(h)$  **no** es una buena medida de la calidad de  $h$ .

# Desbalance y costos de errores no simétricos

- Algoritmos estudiados asumen  $\sim$ igual proporción de datos en cada clase: **datos balanceados**
- Medida de error de hipótesis  $h$ :

$$e(h) = \mathbf{P}_{\mathcal{D}} [\llbracket h(x) \neq y \rrbracket]$$

- Algoritmos minimizan función de error en los datos de manera que  $e(h) \ll$ .
- Sin embargo:
  - ❶ En muchas aplicaciones los datos no están **balanceados**.
  - ❷ El **costo** de cometer un error de clasificación en un dato **depende** de la clase a la cual éste pertenece.
- En estos casos  $e(h)$  **no** es una buena medida de la calidad de  $h$ .
- Entrenamiento debe tener en cuenta costos/datos no balanceados.

# Evaluación del clasificador

		Etiqueta	
		0	1
Predicción	0	$TN$	$FN$
	1	$FP$	$TP$

# Evaluación del clasificador

		Etiqueta	
		0	1
Predicción	0	$TN$	$FN$
	1	$FP$	$TP$

$$T = TN + FN + FP + TP$$

# Evaluación del clasificador

		Etiqueta	
		0	1
Predicción	0	$TN$	$FN$
	1	$FP$	$TP$

$$T = TN + FN + FP + TP$$

- Error de clasificación:  $\frac{FP+FN}{T}$

# Evaluación del clasificador

		Etiqueta	
		0	1
Predicción	0	$TN$	$FN$
	1	$FP$	$TP$

$$T = TN + FN + FP + TP$$

- Error de clasificación:  $\frac{FP+FN}{T}$ , Accuracy:  $\frac{TP+TN}{T}$

# Evaluación del clasificador

		Etiqueta	
		0	1
Predicción	0	$TN$	$FN$
	1	$FP$	$TP$

$$T = TN + FN + FP + TP$$

- Error de clasificación:  $\frac{FP+FN}{T}$ , Accuracy:  $\frac{TP+TN}{T}$
- Sensitividad (recall):  $S = \frac{TP}{TP+FN}$

# Evaluación del clasificador

		Etiqueta	
		0	1
Predicción	0	$TN$	$FN$
	1	$FP$	$TP$

$$T = TN + FN + FP + TP$$

- Error de clasificación:  $\frac{FP+FN}{T}$ , Accuracy:  $\frac{TP+TN}{T}$
- Sensitividad (recall):  $S = \frac{TP}{TP+FN}$
- Especificidad:  $E = \frac{TN}{FP+TN}$



# Evaluación del clasificador

		Etiqueta	
		0	1
Predicción	0	$TN$	$FN$
	1	$FP$	$TP$

$$T = TN + FN + FP + TP$$

- Error de clasificación:  $\frac{FP+FN}{T}$ , Accuracy:  $\frac{TP+TN}{T}$
- Sensitividad (recall):  $S = \frac{TP}{TP+FN}$
- Especificidad:  $E = \frac{TN}{FP+TN}$
- Precisión:  $P = \frac{TP}{TP+FP}$

# Evaluación del clasificador

		Etiqueta	
		0	1
Predicción	0	$TN$	$FN$
	1	$FP$	$TP$

$$T = TN + FN + FP + TP$$

- Error de clasificación:  $\frac{FP+FN}{T}$ , Accuracy:  $\frac{TP+TN}{T}$
- Sensitividad (recall):  $S = \frac{TP}{TP+FN}$
- Especificidad:  $E = \frac{TN}{FP+TN}$
- Precisión:  $P = \frac{TP}{TP+FP}$
- F-score:  $F = 2\frac{S \times P}{S+P}$

# Area bajo la curva ROC (AUC)

- Clasificador retorna valor real (probabilidad)  $\longrightarrow$  umbral.

# Area bajo la curva ROC (AUC)

- Clasificador retorna valor real (probabilidad)  $\longrightarrow$  umbral.
- Calidad del clasificador al variar el umbral.

# Area bajo la curva ROC (AUC)

- Clasificador retorna valor real (probabilidad)  $\rightarrow$  umbral.
- Calidad del clasificador al variar el umbral.
- Escogencia de umbral óptimo de acuerdo al problema.

# Area bajo la curva ROC (AUC)

- Clasificador retorna valor real (probabilidad)  $\rightarrow$  umbral.
- Calidad del clasificador al variar el umbral.
- Escogencia de umbral óptimo de acuerdo al problema.
- Visualización

# Entrenamiento

# Entrenamiento

- Algoritmo de entrenamiento debe tener en cuenta desbalance y/o costos asimétricos.



# Entrenamiento

- Algoritmo de entrenamiento debe tener en cuenta desbalance y/o costos asimétricos.
- Sea  $C(i, j)$  es el costo incurrido al predecir etiqueta  $i$ , cuando la etiqueta es  $j$ .

# Entrenamiento

- Algoritmo de entrenamiento debe tener en cuenta desbalance y/o costos asimétricos.
- Sea  $C(i, j)$  es el costo incurrido al predecir etiqueta  $i$ , cuando la etiqueta es  $j$ .
- Meta: Minimizar **costo esperado** de equivocarse:

$$i^* = \arg \min_i R(i|x) = \sum_j P(j|x) C(i, j)$$

## Matriz de Costos (caso binario)

		Etiqueta	
		0	1
Predicción	0	$c_{00}$	$c_{01}$
	1	$c_{10}$	$c_{11}$

## Matriz de Costos (caso binario)

		Etiqueta	
		0	1
Predicción	0	$c_{00}$	$c_{01}$
	1	$c_{10}$	$c_{11}$

- Costo de falsos positivos.

# Matriz de Costos (caso binario)

		Etiqueta	
		0	1
Predicción	0	$c_{00}$	$c_{01}$
	1	$c_{10}$	$c_{11}$

- Costo de falsos positivos.
- Costo de falsos negativos.

## Matriz de Costos (caso binario)

		Etiqueta	
		0	1
Predicción	0	$c_{00}$	$c_{01}$
	1	$c_{10}$	$c_{11}$

- Costo de falsos positivos.
- Costo de falsos negativos.
- Debemos tener  $c_{10} > c_{00}$  y  $c_{01} > c_{11}$

## Matriz de Costos (caso binario)

		Etiqueta	
		0	1
Predicción	0	$c_{00}$	$c_{01}$
	1	$c_{10}$	$c_{11}$

- Costo de falsos positivos.
- Costo de falsos negativos.
- Debemos tener  $c_{10} > c_{00}$  y  $c_{01} > c_{11}$
- No debe haber flia dominante.

- Decisión óptima es predecir clase 1 si:

$$P(j = 0|x)c_{10} + P(j = 1|x)c_{11} \leq P(j = 0|x)c_{00} + P(j = 1|x)c_{01}$$



- Decisión óptima es predecir clase 1 si:

$$P(j = 0|x)c_{10} + P(j = 1|x)c_{11} \leq P(j = 0|x)c_{00} + P(j = 1|x)c_{01}$$

- Si  $p = P(j = 1|x)$ ,

$$(1 - p)c_{10} + pc_{11} \leq (1 - p)c_{00} + pc_{01}$$

- Decisión óptima es predecir clase 1 si:

$$P(j = 0|x)c_{10} + P(j = 1|x)c_{11} \leq P(j = 0|x)c_{00} + P(j = 1|x)c_{01}$$

- Si  $p = P(j = 1|x)$ ,

$$(1 - p)c_{10} + pc_{11} \leq (1 - p)c_{00} + pc_{01}$$

- Umbral óptimo: predecir clase 1 iff  $p \geq p^*$ :

$$p^* = \frac{c_{10} - c_{00}}{c_{10} - c_{00} + c_{01} - c_{11}}$$

- Note que para el caso balanceado (sin costos)  $p^{\star} = \frac{1}{2}$ .

- Note que para el caso balanceado (sin costos)  $p^* = \frac{1}{2}$ .
- Cómo hacer que un algoritmo estándar clasifique de acuerdo a un  $p^*$  dado?

- Note que para el caso balanceado (sin costos)  $p^* = \frac{1}{2}$ .
- Cómo hacer que un algoritmo estándar clasifique de acuerdo a un  $p^*$  dado?

## Teorema

*Para hacer un umbral objetivo  $p^*$  corresponder a un umbral dado  $p_0$ , el número de ejemplos negativos en el conjunto de entrenamiento debe ser multiplicado por:*

$$\frac{p^*}{1 - p^*} \frac{1 - p_0}{p_0}$$

- Note que para el caso balanceado (sin costos)  $p^* = \frac{1}{2}$ .
- Cómo hacer que un algoritmo estándar clasifique de acuerdo a un  $p^*$  dado?

## Teorema

*Para hacer un umbral objetivo  $p^*$  corresponder a un umbral dado  $p_0$ , el número de ejemplos negativos en el conjunto de entrenamiento debe ser multiplicado por:*

$$\frac{p^*}{1 - p^*} \frac{1 - p_0}{p_0}$$

- Por ejemplo, si  $p_0 = 0,5$  y  $c_{00} = c_{11} = 0$ , el número de ejemplos negativos debe ser multiplicado por  $p^*/(1 - p^*) = c_{10}/c_{01}$

# Técnicas de remuestreo

- Sobremuestreo de clase minoritaria.

# Técnicas de remuestreo

- Sobremuestreo de clase minoritaria.
  - ▶ Muestrear con reemplazo (bootstrap).



# Técnicas de remuestreo

- Sobremuestreo de clase minoritaria.
  - ▶ Muestrear con reemplazo (bootstrap).
  - ▶ Añadir datos sintéticos.

# Técnicas de remuestreo

- Sobremuestreo de clase minoritaria.
  - ▶ Muestrear con reemplazo (bootstrap).
  - ▶ Añadir datos sintéticos.
- Submuestreo de clase mayoritaria.

# Técnicas de remuestreo

- Sobremuestreo de clase minoritaria.
  - ▶ Muestrear con reemplazo (bootstrap).
  - ▶ Añadir datos sintéticos.
- Submuestreo de clase mayoritaria.
  - ▶ Muestreo sin reemplazo

# Técnicas de remuestreo

- Sobremuestreo de clase minoritaria.
  - ▶ Muestrear con reemplazo (bootstrap).
  - ▶ Añadir datos sintéticos.
- Submuestreo de clase mayoritaria.
  - ▶ Muestreo sin reemplazo
  - ▶ Eliminación selectiva (outliers, etc).

# Técnicas de remuestreo

- Sobremuestreo de clase minoritaria.
  - ▶ Muestrear con reemplazo (bootstrap).
  - ▶ Añadir datos sintéticos.
- Submuestreo de clase mayoritaria.
  - ▶ Muestreo sin reemplazo
  - ▶ Eliminación selectiva (outliers, etc).
- SMOTE: Synthetic Minority Over-sampling TEchnique

# Técnicas de remuestreo

- Sobremuestreo de clase minoritaria.
  - ▶ Muestrear con reemplazo (bootstrap).
  - ▶ Añadir datos sintéticos.
- Submuestreo de clase mayoritaria.
  - ▶ Muestreo sin reemplazo
  - ▶ Eliminación selectiva (outliers, etc).
- SMOTE: Synthetic Minority Over-sampling TEchnique



- Solución óptima:

$$i^* = \arg \min_i R(i|x) = \sum_j P(j|x) C(i, j)$$



- Solución óptima:

$$i^* = \arg \min_i R(i|x) = \sum_j P(j|x) C(i, j)$$

- Partición del espacio  $\mathcal{X}$  en  $k$  regiones.

- Solución óptima:

$$i^* = \arg \min_i R(i|x) = \sum_j P(j|x)C(i, j)$$

- Partición del espacio  $\mathcal{X}$  en  $k$  regiones.
- Si los costos cambian, partición cambia **aún** si probabilidades de clase no cambian.

- Solución óptima:

$$i^* = \arg \min_i R(i|x) = \sum_j P(j|x) C(i, j)$$

- Partición del espacio  $\mathcal{X}$  en  $k$  regiones.
- Si los costos cambian, partición cambia **aún** si probabilidades de clase no cambian.
- Las predicciones óptimas en el *conjunto de entrenamiento* no se conocen.

- Solución óptima:

$$i^* = \arg \min_i R(i|x) = \sum_j P(j|x) C(i, j)$$

- Partición del espacio  $\mathcal{X}$  en  $k$  regiones.
- Si los costos cambian, partición cambia **aún** si probabilidades de clase no cambian.
- Las predicciones óptimas en el *conjunto de entrenamiento* no se conocen.
- MetaCost **estima** etiquetas correctas usando **Bagging**, y entrena clasificador con esas etiquetas.

---

**Algorithm 1** Bagging

---

**for**  $t = 1$  to  $T$  **do**

    Obtenga  $\mathcal{S}_t$  de  $\mathcal{S}$  muestreando con reemplazo.

$h_t \leftarrow A(\mathcal{S}_t)$

**end for**

Retorne  $f(x) = \text{votacion } \{h_t(x)\}$

---

---

**Algorithm 2** MetaCost

---

**for**  $t = 1$  to  $T$  **do**

Obtenga  $\mathcal{S}_t$  de  $\mathcal{S}$  muestreando  $m \leq n$  datos con reemplazo

$h_t \leftarrow A(\mathcal{S}_t)$

**end for**

**for** cada  $x \in \mathcal{S}$  **do**

**for** cada clase  $j$  **do**

$$\check{P}(j|x) = \frac{1}{T} \sum_{t=1}^T \mathbb{I}[h_t(x) = j]$$

**end for**

$\check{y} = \arg \min_i \sum_j \check{P}(j|x) C(i, j)$

**end for**

Retorne  $h_M(x) = A(\check{\mathcal{S}})$

---

# Variantes

# Variantes

- Reducir tamaño de muestra bootstrap ( $m \ll n$ ).



# Variantes

- Reducir tamaño de muestra bootstrap ( $m \ll n$ ).
- Si clasificador base retorna probabilidades, usarlas en estimativo  $\check{P}(j|x)$

# Variantes

- Reducir tamaño de muestra bootstrap ( $m \ll n$ ).
- Si clasificador base retorna probabilidades, usarlas en estimativo  $\check{P}(j|x)$
- Incluir en estimativo de  $\check{P}(j|x)$  sólo  $h_t$  entrenado en los que  $x \notin \mathcal{S}_t$

# Costing (Zadrozny, Langford and Abe, 2003)

# Costing (Zadrozny, Langford and Abe, 2003)

- $(x, y, c) \sim \mathcal{D}$  con  $x \in \mathcal{X}$ ,  $y \in \{-1, 1\}$  y  $c \in \mathbb{R}^+$ .

# Costing (Zadrozny, Langford and Abe, 2003)

- $(x, y, c) \sim \mathcal{D}$  con  $x \in \mathcal{X}$ ,  $y \in \{-1, 1\}$  y  $c \in \mathbb{R}^+$ .
- Datos de entrenamiento  $\mathcal{S} = \{(x_i, y_i, c_i)\}$ .

# Costing (Zadrozny, Langford and Abe, 2003)

- $(x, y, c) \sim \mathcal{D}$  con  $x \in \mathcal{X}$ ,  $y \in \{-1, 1\}$  y  $c \in \mathbb{R}^+$ .
- Datos de entrenamiento  $\mathcal{S} = \{(x_i, y_i, c_i)\}$ .
- Meta: obtener hipótesis  $h : \mathcal{X} \longrightarrow \{-1, 1\}$  que minimice

$$\mathbb{E}_{x,y,c \sim \mathcal{D}} [c \mathbb{I}[h(x) \neq y]]$$

# Costing (Zadrozny, Langford and Abe, 2003)

- $(x, y, c) \sim \mathcal{D}$  con  $x \in \mathcal{X}$ ,  $y \in \{-1, 1\}$  y  $c \in \mathbb{R}^+$ .
- Datos de entrenamiento  $\mathcal{S} = \{(x_i, y_i, c_i)\}$ .
- Meta: obtener hipótesis  $h : \mathcal{X} \longrightarrow \{-1, 1\}$  que minimice

$$\mathbb{E}_{x,y,c \sim \mathcal{D}} [c \mathbb{I}[h(x) \neq y]]$$

- Más general que métodos que usan matriz de costos.

# Teorema de traducción

## Teorema

*Para toda distribución  $\mathcal{D}$  existe una constante  $N = \mathbb{E}_{x,y,c \sim \mathcal{D}} [c]$  tal que para todo clasificador  $h$ :*

$$\mathbb{E}_{x,y,c \sim \hat{\mathcal{D}}} [\llbracket h(x) \neq y \rrbracket] = \frac{1}{N} \mathbb{E}_{x,y,c \sim \mathcal{D}} [c \llbracket h(x) \neq y \rrbracket]$$

*donde  $\hat{\mathcal{D}}(x, y, c) = \frac{c}{N} \mathcal{D}(x, y, c)$*



# Teorema de traducción

## Teorema

Para toda distribución  $\mathcal{D}$  existe una constante  $N = \mathbb{E}_{x,y,c \sim \mathcal{D}} [c]$  tal que para todo clasificador  $h$ :

$$\mathbb{E}_{x,y,c \sim \hat{\mathcal{D}}} [\llbracket h(x) \neq y \rrbracket] = \frac{1}{N} \mathbb{E}_{x,y,c \sim \mathcal{D}} [c \llbracket h(x) \neq y \rrbracket]$$

donde  $\hat{\mathcal{D}}(x, y, c) = \frac{c}{N} \mathcal{D}(x, y, c)$

## Demostración.

# Teorema de traducción

## Teorema

Para toda distribución  $\mathcal{D}$  existe una constante  $N = \mathbb{E}_{x,y,c \sim \mathcal{D}} [c]$  tal que para todo clasificador  $h$ :

$$\mathbb{E}_{x,y,c \sim \hat{\mathcal{D}}} [\llbracket h(x) \neq y \rrbracket] = \frac{1}{N} \mathbb{E}_{x,y,c \sim \mathcal{D}} [c \llbracket h(x) \neq y \rrbracket]$$

donde  $\hat{\mathcal{D}}(x, y, c) = \frac{c}{N} \mathcal{D}(x, y, c)$

## Demostración.

$$\mathbb{E}_{x,y,c \sim \mathcal{D}} [c \llbracket h(x) \neq y \rrbracket] = \int c \llbracket h(x) \neq y \rrbracket d\mathcal{D}$$

# Teorema de traducción

## Teorema

Para toda distribución  $\mathcal{D}$  existe una constante  $N = \mathbb{E}_{x,y,c \sim \mathcal{D}} [c]$  tal que para todo clasificador  $h$ :

$$\mathbb{E}_{x,y,c \sim \hat{\mathcal{D}}} [\llbracket h(x) \neq y \rrbracket] = \frac{1}{N} \mathbb{E}_{x,y,c \sim \mathcal{D}} [c \llbracket h(x) \neq y \rrbracket]$$

donde  $\hat{\mathcal{D}}(x, y, c) = \frac{c}{N} \mathcal{D}(x, y, c)$

## Demostración.

$$\begin{aligned} \mathbb{E}_{x,y,c \sim \mathcal{D}} [c \llbracket h(x) \neq y \rrbracket] &= \int c \llbracket h(x) \neq y \rrbracket d\mathcal{D} \\ &= N \int \llbracket h(x) \neq y \rrbracket d\hat{\mathcal{D}} \end{aligned}$$

# Teorema de traducción

## Teorema

Para toda distribución  $\mathcal{D}$  existe una constante  $N = \mathbb{E}_{x,y,c \sim \mathcal{D}} [c]$  tal que para todo clasificador  $h$ :

$$\mathbb{E}_{x,y,c \sim \hat{\mathcal{D}}} [\llbracket h(x) \neq y \rrbracket] = \frac{1}{N} \mathbb{E}_{x,y,c \sim \mathcal{D}} [c \llbracket h(x) \neq y \rrbracket]$$

donde  $\hat{\mathcal{D}}(x, y, c) = \frac{c}{N} \mathcal{D}(x, y, c)$

## Demostración.

$$\begin{aligned} \mathbb{E}_{x,y,c \sim \mathcal{D}} [c \llbracket h(x) \neq y \rrbracket] &= \int c \llbracket h(x) \neq y \rrbracket d\mathcal{D} \\ &= N \int \llbracket h(x) \neq y \rrbracket d\hat{\mathcal{D}} \\ &= N \mathbb{E}_{x,y,c \sim \hat{\mathcal{D}}} [\llbracket h(x) \neq y \rrbracket] \end{aligned}$$

- El clasificador  $h$  que minimiza el costo esperado con respecto a  $\mathcal{D}$  minimiza el error esperado con respecto a  $\hat{\mathcal{D}}$ .

- El clasificador  $h$  que minimiza el costo esperado con respecto a  $\mathcal{D}$  minimiza el error esperado con respecto a  $\hat{\mathcal{D}}$ .
- Para obtener clasificador que minimiza el costo esperado:

- El clasificador  $h$  que minimiza el costo esperado con respecto a  $\mathcal{D}$  minimiza el error esperado con respecto a  $\hat{\mathcal{D}}$ .
- Para obtener clasificador que minimiza el costo esperado:
  - ① Escalar distribución de manera que  $(x, y) \sim \hat{\mathcal{D}}$ ,

- El clasificador  $h$  que minimiza el **costo esperado con respecto a  $\mathcal{D}$**  minimiza el **error esperado con respecto a  $\hat{\mathcal{D}}$** .
- Para obtener clasificador que minimiza el costo esperado:
  - 1 Escalar distribución de manera que  $(x, y) \sim \hat{\mathcal{D}}$ ,
  - 2 Entrenar clasificador con datos modificados.



- El clasificador  $h$  que minimiza el **costo esperado con respecto a  $\mathcal{D}$**  minimiza el **error esperado con respecto a  $\hat{\mathcal{D}}$** .
- Para obtener clasificador que minimiza el costo esperado:
  - 1 Escalar distribución de manera que  $(x, y) \sim \hat{\mathcal{D}}$ ,
  - 2 Entrenar clasificador con datos modificados.
  - ★ Caja transparente.

- El clasificador  $h$  que minimiza el **costo esperado con respecto a  $\mathcal{D}$**  minimiza el **error esperado con respecto a  $\hat{\mathcal{D}}$** .
- Para obtener clasificador que minimiza el costo esperado:
  - 1 Escalar distribución de manera que  $(x, y) \sim \hat{\mathcal{D}}$ ,
  - 2 Entrenar clasificador con datos modificados.
    - ★ Caja transparente.
    - ★ Caja negra.

# Caja transparente

- Conocimiento del algoritmo de clasificación específico.

# Caja transparente

- Conocimiento del algoritmo de clasificación específico.
- Modificar algoritmo de clasificación de manera que tenga en cuenta los pesos  $c/N$

# Caja transparente

- Conocimiento del algoritmo de clasificación específico.
- Modificar algoritmo de clasificación de manera que tenga en cuenta los pesos  $c/N$
- Adaboost: Pesos iniciales  $D_i = \frac{c_i}{N}$

# Caja transparente

- Conocimiento del algoritmo de clasificación específico.
- Modificar algoritmo de clasificación de manera que tenga en cuenta los pesos  $c/N$
- Adaboost: Pesos iniciales  $D_i = \frac{c_i}{N}$
- SVMs:

# Caja transparente

- Conocimiento del algoritmo de clasificación específico.
- Modificar algoritmo de clasificación de manera que tenga en cuenta los pesos  $c/N$
- Adaboost: Pesos iniciales  $D_i = \frac{c_i}{N}$
- SVMs:

$$\text{mín} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n c_i \zeta_i$$

$$\text{sujeto a} \quad y_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 + \zeta_i \geq 0 \quad i = 1, \dots, n$$
$$\zeta_i \geq 0$$

# Caja Negra



# Caja Negra

- No se tiene acceso al funcionamiento del algoritmo de clasificación.

# Caja Negra

- No se tiene acceso al funcionamiento del algoritmo de clasificación.
- **Filtrar** distribución de manera que  $(x, y) \sim \hat{\mathcal{D}}$

# Caja Negra

- No se tiene acceso al funcionamiento del algoritmo de clasificación.
- **Filtrar** distribución de manera que  $(x, y) \sim \hat{\mathcal{D}}$ 
  - ▶ Muestreo sin reemplazo:

# Caja Negra

- No se tiene acceso al funcionamiento del algoritmo de clasificación.
- **Filtrar** distribución de manera que  $(x, y) \sim \hat{\mathcal{D}}$ 
  - ▶ Muestreo sin reemplazo:
    - ★ Conjunto de entrenamiento  $\ll$ .

# Caja Negra

- No se tiene acceso al funcionamiento del algoritmo de clasificación.
- **Filtrar** distribución de manera que  $(x, y) \sim \hat{\mathcal{D}}$ 
  - ▶ Muestreo sin reemplazo:
    - ★ Conjunto de entrenamiento  $\ll$ .
    - ★ Datos resultantes **No** satisfacen  $(x, y) \sim \hat{\mathcal{D}}$ .

# Caja Negra

- No se tiene acceso al funcionamiento del algoritmo de clasificación.
- **Filtrar** distribución de manera que  $(x, y) \sim \hat{\mathcal{D}}$ 
  - ▶ Muestreo sin reemplazo:
    - ★ Conjunto de entrenamiento  $\ll$ .
    - ★ Datos resultantes **No** satisfacen  $(x, y) \sim \hat{\mathcal{D}}$ .
  - ▶ Muestreo con reemplazo:

# Caja Negra

- No se tiene acceso al funcionamiento del algoritmo de clasificación.
- **Filtrar** distribución de manera que  $(x, y) \sim \hat{\mathcal{D}}$ 
  - ▶ Muestreo sin reemplazo:
    - ★ Conjunto de entrenamiento  $\ll$ .
    - ★ Datos resultantes **No** satisfacen  $(x, y) \sim \hat{\mathcal{D}}$ .
  - ▶ Muestreo con reemplazo:
    - ★ Datos resultantes satisfacen  $(x, y) \sim \hat{\mathcal{D}}$ .

# Caja Negra

- No se tiene acceso al funcionamiento del algoritmo de clasificación.
- **Filtrar** distribución de manera que  $(x, y) \sim \hat{\mathcal{D}}$ 
  - ▶ Muestreo sin reemplazo:
    - ★ Conjunto de entrenamiento  $\ll$ .
    - ★ Datos resultantes **No** satisfacen  $(x, y) \sim \hat{\mathcal{D}}$ .
  - ▶ Muestreo con reemplazo:
    - ★ Datos resultantes satisfacen  $(x, y) \sim \hat{\mathcal{D}}$ .
    - ★ **No** son independientes.



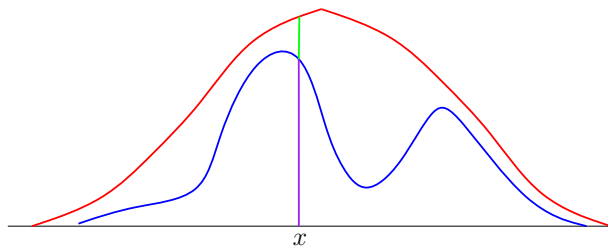
# Caja Negra

- No se tiene acceso al funcionamiento del algoritmo de clasificación.
- **Filtrar** distribución de manera que  $(x, y) \sim \hat{\mathcal{D}}$ 
  - ▶ Muestreo sin reemplazo:
    - ★ Conjunto de entrenamiento  $\ll$ .
    - ★ Datos resultantes **No** satisfacen  $(x, y) \sim \hat{\mathcal{D}}$ .
  - ▶ Muestreo con reemplazo:
    - ★ Datos resultantes satisfacen  $(x, y) \sim \hat{\mathcal{D}}$ .
    - ★ **No** son independientes.
    - ★ Overfitting.

# Caja Negra

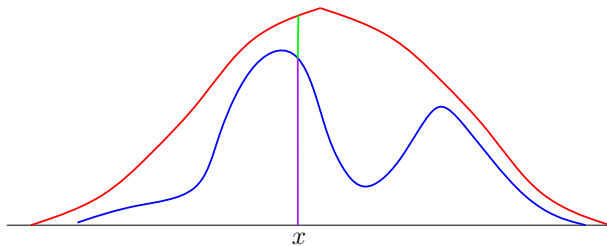
- No se tiene acceso al funcionamiento del algoritmo de clasificación.
- **Filtrar** distribución de manera que  $(x, y) \sim \hat{\mathcal{D}}$ 
  - ▶ Muestreo sin reemplazo:
    - ★ Conjunto de entrenamiento  $\ll$ .
    - ★ Datos resultantes **No** satisfacen  $(x, y) \sim \hat{\mathcal{D}}$ .
  - ▶ Muestreo con reemplazo:
    - ★ Datos resultantes satisfacen  $(x, y) \sim \hat{\mathcal{D}}$ .
    - ★ **No** son independientes.
    - ★ Overfitting.
  - ▶ **Muestreo con rechazo.**

# Muestreo con rechazo



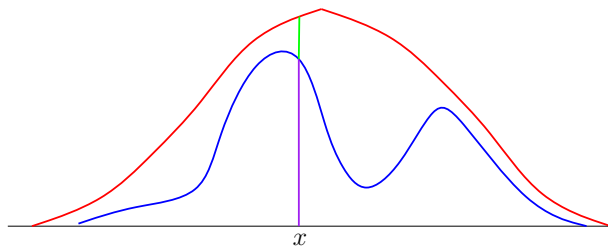
- Distribución difícil de muestrear, pero  $\alpha f$  evaluable .

# Muestreo con rechazo



- Distribución difícil de muestrear, pero  $\alpha f$  evaluable .
- Muestrear de acuerdo a distribución envolvente  $g$ .

# Muestreo con rechazo



- Distribución difícil de muestrear, pero  $\alpha f$  evaluable .
- Muestrear de acuerdo a distribución envolvente  $g$ .
- Aceptar con probabilidad  $\propto f(x)/g(x)$

# Muestreo con rechazo proporcional al costo

- En este caso:

# Muestreo con rechazo proporcional al costo

- En este caso:

① Datos originales  $\mathcal{S} = \{(x_i, y_i)\} \sim \mathcal{D}$

# Muestreo con rechazo proporcional al costo

- En este caso:
  - 1 Datos originales  $\mathcal{S} = \{(x_i, y_i)\} \sim \mathcal{D}$
  - 2 Seleccionar  $(x_i, y_i)$  aleatoriamente.



# Muestreo con rechazo proporcional al costo

- En este caso:
  - 1 Datos originales  $\mathcal{S} = \{(x_i, y_i)\} \sim \mathcal{D}$
  - 2 Seleccionar  $(x_i, y_i)$  aleatoriamente.
  - 3 Aceptar con probabilidad  $c/Z$

# Muestreo con rechazo proporcional al costo

- En este caso:
  - 1 Datos originales  $\mathcal{S} = \{(x_i, y_i)\} \sim \mathcal{D}$
  - 2 Seleccionar  $(x_i, y_i)$  aleatoriamente.
  - 3 Aceptar con probabilidad  $c/Z$
- $Z$  es una cota superior e los costos.

---

**Algorithm 3** Costing

---

**for**  $t = 1$  to  $T$  **do**

    Obtenga  $\mathcal{S}_t$  de  $\mathcal{S}$  muestreando con rechazo y probabilidad de aceptar  $c/Z$ .

$h_t \leftarrow A(\mathcal{S}_t)$

**end for**

Retorne  $f(x) = \text{votacion } \{h_t(x)\}$

---

# Propiedades

# Propiedades

- $\mathbb{E}_{x,y,c \sim \hat{\mathcal{D}}} [\llbracket h_t(x) \neq y \rrbracket] \leq \epsilon \Rightarrow \mathbb{E}_{x,y,c \sim \mathcal{D}} [c \llbracket h_t(x) \neq y \rrbracket] \leq N\epsilon$

# Propiedades

- $\mathbb{E}_{x,y,c \sim \hat{\mathcal{D}}} [\llbracket h_t(x) \neq y \rrbracket] \leq \epsilon \Rightarrow \mathbb{E}_{x,y,c \sim \mathcal{D}} [c \llbracket h_t(x) \neq y \rrbracket] \leq N\epsilon$
- Complejidad de muestra de aprendizaje con datos muestreados con rechazo es menor que complejidad de muestra sin costos (modelo PAC).

# Propiedades

- $\mathbb{E}_{x,y,c \sim \hat{\mathcal{D}}} [\llbracket h_t(x) \neq y \rrbracket] \leq \epsilon \Rightarrow \mathbb{E}_{x,y,c \sim \mathcal{D}} [c \llbracket h_t(x) \neq y \rrbracket] \leq N\epsilon$
- Complejidad de muestra de aprendizaje con datos muestreados con rechazo es menor que complejidad de muestra sin costos (modelo PAC).
- Promediar sobre múltiples modelos mejora generalización.

# Propiedades

- $\mathbb{E}_{x,y,c \sim \hat{\mathcal{D}}} [\llbracket h_t(x) \neq y \rrbracket] \leq \epsilon \Rightarrow \mathbb{E}_{x,y,c \sim \mathcal{D}} [c \llbracket h_t(x) \neq y \rrbracket] \leq N\epsilon$
- Complejidad de muestra de aprendizaje con datos muestreados con rechazo es menor que complejidad de muestra sin costos (modelo PAC).
- Promediar sobre múltiples modelos mejora generalización.
- Tiempo de corrida de cada entrenamiento es pequeño porque típicamente  $|\mathcal{S}_t| < |\mathcal{S}|$