



Métodos de Consenso,
potenciación y algo más

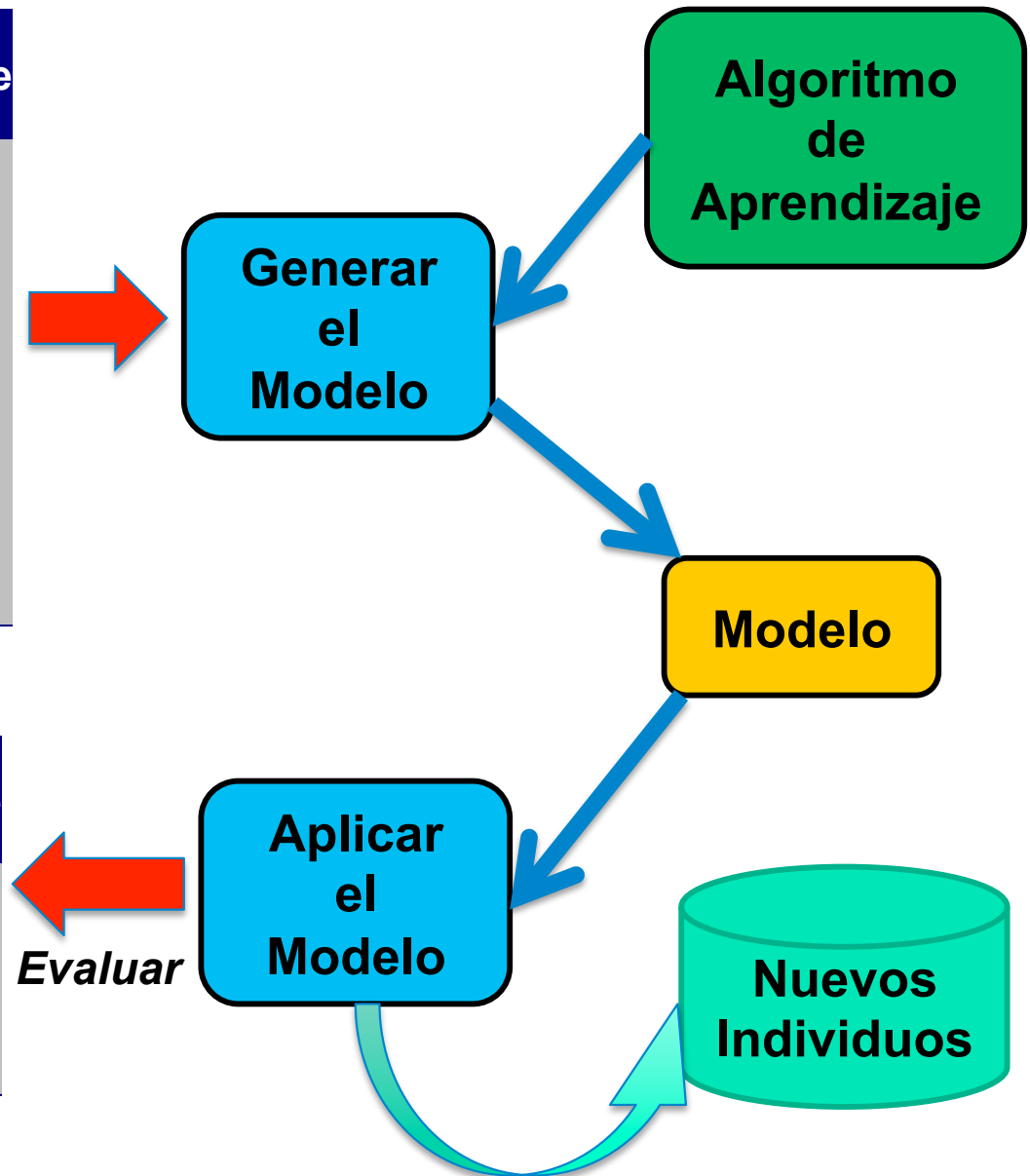
Modelo general de los métodos de Clasificación

/d	Reembolso	Estado Civil	Ingresos Anuales	Fraude
1	Sí	Soltero	125K	No
2	No	Casado	100K	No
3	No	Soltero	70K	No
4	Sí	Casado	120K	No
5	No	Divorciado	95K	Sí
6	No	Casado	60K	No

Tabla de Aprendizaje

Id	Reembolso	Estado Civil	Ingresos Anuales	Fraude
7	No	Soltero	80K	No
8	Si	Casado	100K	No
9	No	Soltero	70K	No

Tabla de Testing



Definición de Clasificación

- Dada una base de datos $D = \{t_1, t_2, \dots, t_n\}$ de tuplas o registros (individuos) y un conjunto de clases $C = \{C_1, C_2, \dots, C_m\}$, el **problema de la clasificación** es encontrar una función $f: D \rightarrow C$ tal que cada t_i es asignada una clase C_j .
- $f: D \rightarrow C$ podría ser una Red Neuronal, un Árbol de Decisión, un modelo basado en Análisis Discriminante, o una Red Bayesiana.



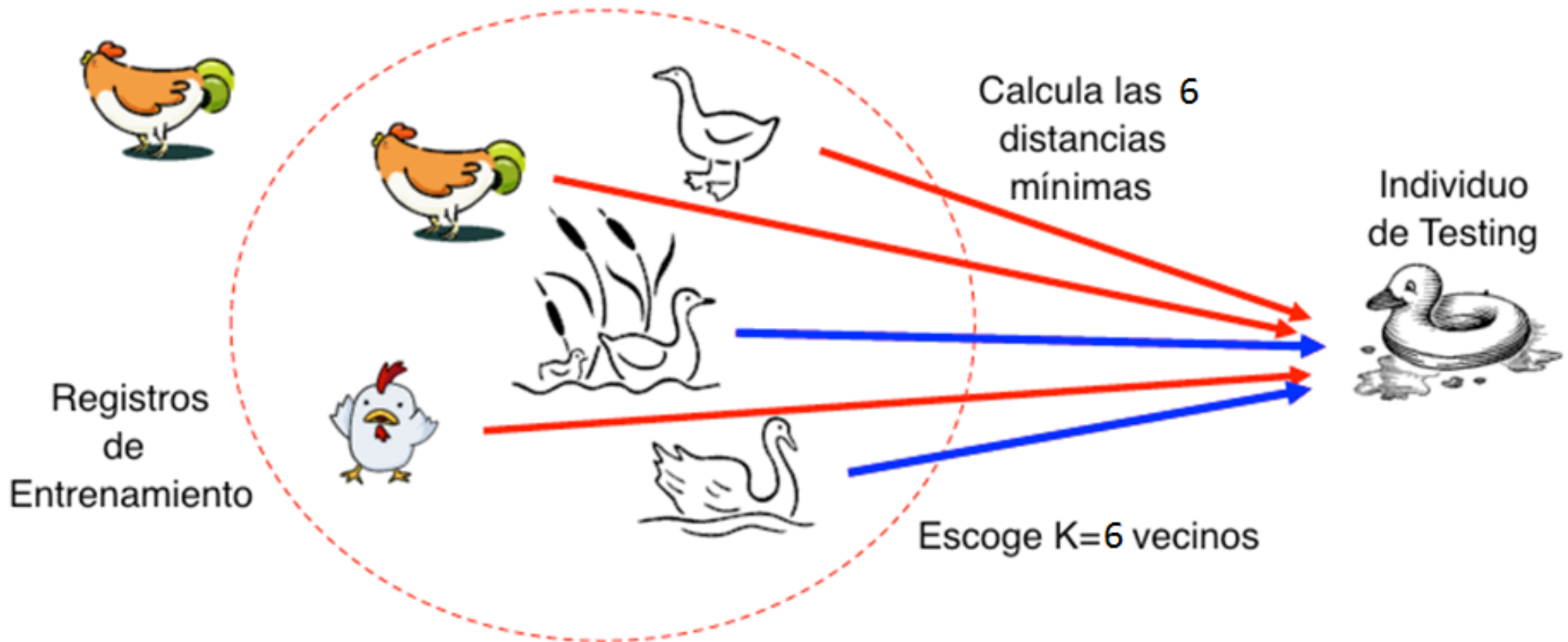
Aprendizaje Supervisado

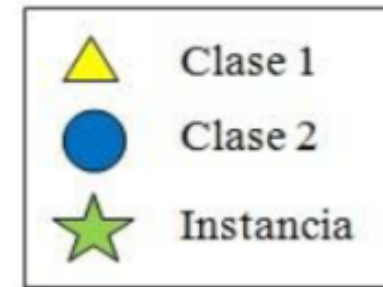
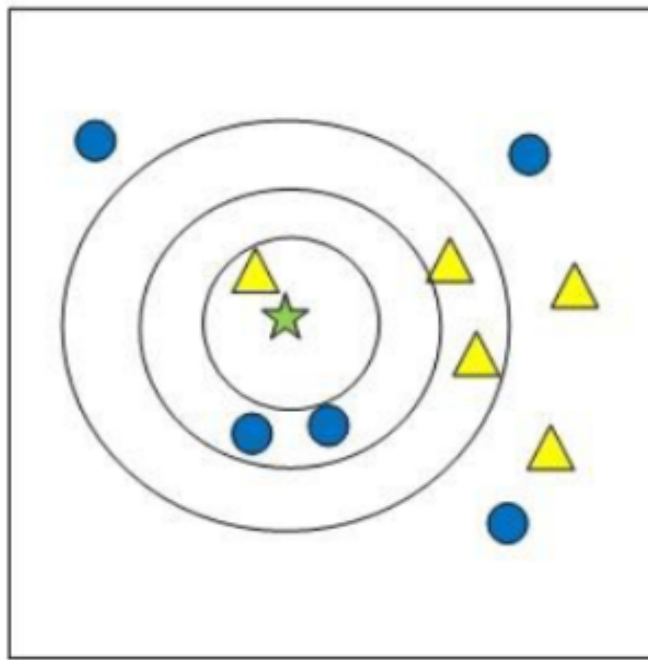
K - Vecinos más cercanos

Knn-Method



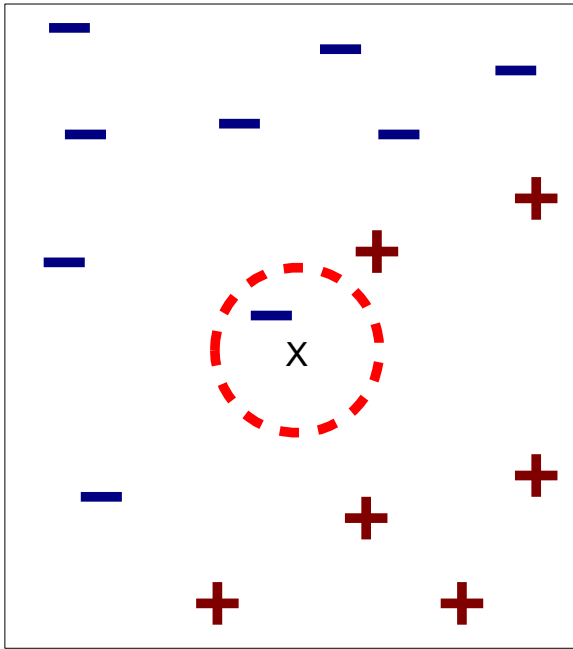
Como de los $K=6$ “individuos” de entrenamiento 4 son patos entonces el “individuo” de testing se clasifica como pato. Criterio “**Majority Vote**”



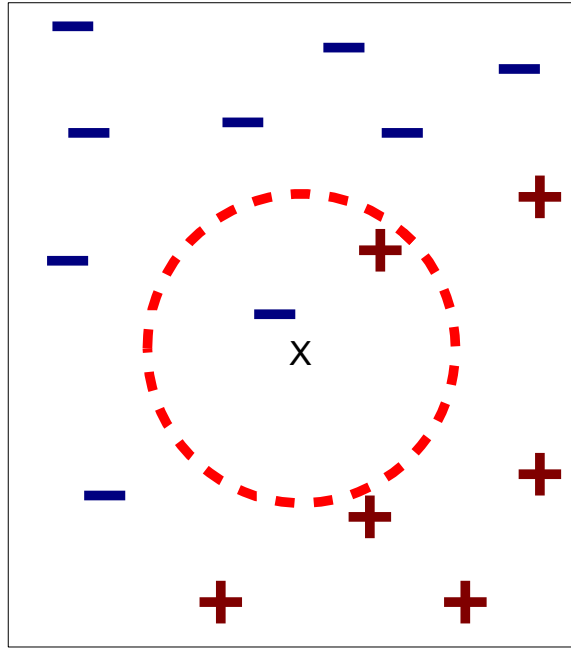


Para $K=1$ (círculo más pequeño), la clase de la nueva instancia sería la Clase 1, ya que es la clase de su vecino más cercano, mientras que para $K=3$ la clase de la nueva instancia sería la Clase 2 pues habrían dos vecinos de la Clase 2 y solo 1 de la Clase 1

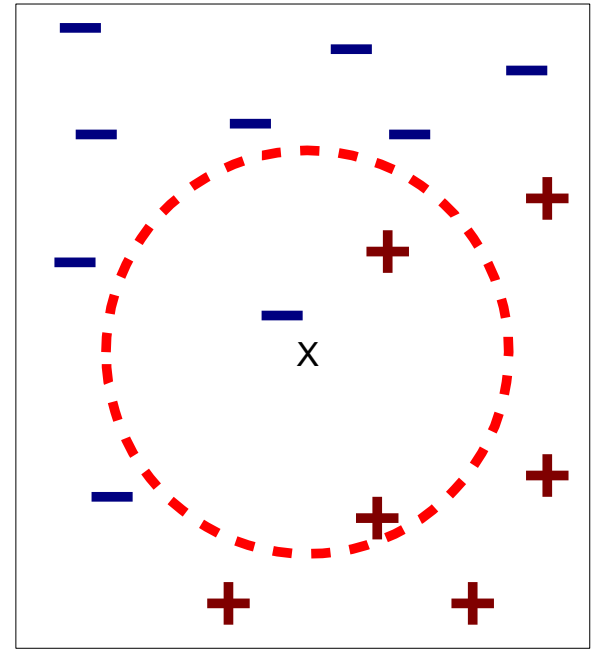
¿Cómo escoger K?



(a) 1-nearest neighbor



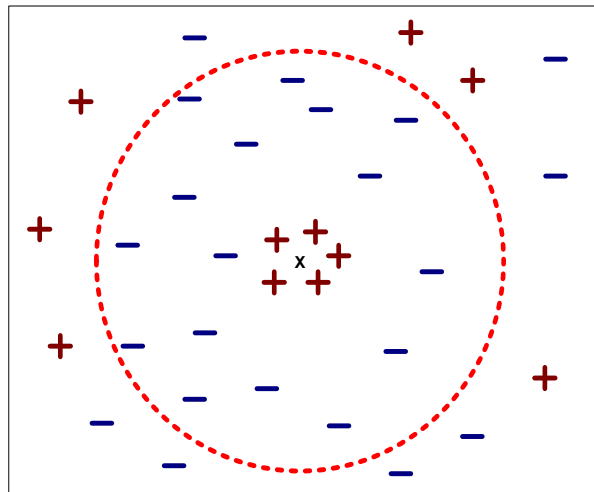
(b) 2-nearest neighbor



(c) 3-nearest neighbor

¿Cómo escoger K?

- Escogiendo el valor de K:
 - Si K es muy pequeño el modelo será muy sensitivo a puntos que son atípicos o que son ruido (datos corruptos)
 - Si K es muy grande, el modelo tiende a asignar siempre a la clase más grande.



K - Vecinos más cercano en R

Package 'kkn'

October 30, 2012

Title Weighted k-Nearest Neighbors

Version 1.2-1

Date 2012-30-10

Author Klaus Schliep & Klaus Hechenbichler

Description Weighted k-Nearest Neighbors Classification, Regression and Clustering

Maintainer Klaus Schliep <klaus.schliep@gmail.com>

Depends R (>= 2.10), igraph (>= 0.6)

Imports Matrix, stats

License GPL (>= 2)

Repository CRAN

Date/Publication 2012-10-30 12:23:58

A 3D pie chart is positioned on the left side of the slide. It features two slices: a larger, light gray slice on the left and a smaller, light orange slice on the right. The orange slice is slightly exploded from the rest of the chart. The text is centered over the orange slice.

Aprendizaje Supervisado

Método de Bayes

Teorema de Naïve Bayes

Teorema de Bayes

Sea A_1, A_2, \dots, A_k una colección de k eventos mutuamente excluyentes y exhaustivos con $P(A_i) > 0$ para $i = 1, \dots, k$. Entonces para cualquier otro evento B para el cual $P(B) > 0$,

$$P(A_j | B) = \frac{P(A_j \cap B)}{P(B)} = \frac{P(B | A_j)P(A_j)}{\sum_{i=1}^k P(B | A_i) \cdot P(A_i)} \quad j = 1, \dots, k \quad (2.6)$$

Ejemplo: Créditos en un Banco

OLDEMARRR.DMEx...ditoViviendaPeq							
	Id	MontoCredito	IngresoNeto	CoeficienteCre...	MontoCuota	GradoAcademico	BuenPagador
▶	1	2	4	3	1	4	1
	2	2	3	2	1	4	1
	3	4	1	1	4	2	2
	4	1	4	3	1	4	1
	5	3	3	1	3	2	2
	6	3	4	3	1	4	1
	7	4	2	1	3	2	2
	8	4	1	3	3	2	2
	9	3	4	3	1	3	1
	10	1	3	2	2	4	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Dada esta de Aprendizaje predecir para los siguientes individuos si van a ser buenos o malos pagadores.

Ejemplo: Créditos en un Banco

Nuevos Individuos

OLDEMARRR.DMEx ...editoViviendaNI Object Explorer Details							
	Id	MontoCredito	IngresoNeto	CoficienteCre...	MontoCuota	GradoAcademico	BuenPagador
	100	2	4	2	2	3	?
▶	101	1	4	3	2	4	?
	102	3	2	3	4	2	?
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- Se tiene una nueva fila de la base de datos $\mathbf{t} = (100, 2, 4, 2, 2, 3, ?)$.
- El problema es: a partir de la tabla de aprendizaje y usando Clasificación Bayesiana predecir si el individuo #100 corresponde a un buen pagador o a un mal pagador.
- Lo que se hace en estos caso es calcular $P(\text{Bueno}/t)$ y $P(\text{Malo}/t)$ para determinar cuál es mayor, donde por bueno se entiende que la variable BuenPagador=1 y por malo que BuenPagador=2.
- Hay que hacer los mismo con el #101 y el #102 **TAREA**.

Ejemplo de Clasificación Bayesiana

$$P(\text{Bueno}|t) = \frac{P(t|\text{Bueno}) \cdot P(\text{Bueno})}{P(t|\text{Bueno}) \cdot P(\text{Bueno}) + P(t|\text{Malo}) \cdot P(\text{Malo})}$$

$$P(\text{Bueno}) = \frac{6}{10} \text{ y } P(\text{Malo}) = \frac{4}{10}$$

Como $t = (100, 2, 4, 2, 2, 3, ?)$, este es un evento que corresponde realmente a 5 eventos independientes, ser MontoCredito=2, IngresoNeto=4, CoeficienteCreditoAvaluo=2, MontoCuota=2 y GradoAcademico=3. Así:

$$\begin{aligned} P(t|\text{Bueno}) &= P((\text{MontoCredito} = 2)|\text{Bueno}) \cdot P((\text{IngresoNeto} = 4)|\text{Bueno}) \\ &\cdot P((\text{CoeficienteCreditoAvaluo} = 2)|\text{Bueno}) \cdot P((\text{MontoCuota} = 2)|\text{Bueno}) \\ &\cdot P((\text{GradoAcademico} = 3)|\text{Bueno}) = \frac{2}{6} \cdot \frac{4}{6} \cdot \frac{2}{6} \cdot \frac{1}{6} \cdot \frac{1}{6} = \frac{16}{7776} = 0,002. \end{aligned}$$

Ejemplo de Clasificación Bayesiana

$$P(t|Malo)$$

$$\begin{aligned} &= P((MontoCredito = 2)|Malo) \cdot P((IngresoNeto = 4)|Malo) \\ &\cdot P((CoeficienteCreditoAvaluo = 2)|Malo) \cdot P((MontoCuota = 2)|Malo) \\ &\cdot P((GradoAcademico = 3)|Malo) = \frac{0}{4} \cdot \frac{0}{4} \cdot \frac{0}{4} \cdot \frac{0}{4} \cdot \frac{0}{4} = 0. \end{aligned}$$

$$P(Bueno|t) = \frac{P(t|Bueno) \cdot P(Bueno)}{P(t|Bueno) \cdot P(Bueno) + P(t|Malo) \cdot P(Malo)} = \frac{\frac{16}{7776} \cdot \frac{6}{10}}{\frac{16}{7776} \cdot \frac{6}{10} + 0 \cdot \frac{4}{10}} = 1$$

Por lo que $P(Malo|t) = 0$. Pero lo vamos a verificar:

$$P(Malo|t) = \frac{P(t|Malo) \cdot P(Malo)}{P(t|Bueno) \cdot P(Bueno) + P(t|Malo) \cdot P(Malo)} = \frac{0 \cdot \frac{4}{10}}{\frac{16}{7776} \cdot \frac{6}{10} + 0 \cdot \frac{4}{10}} = 0.$$

Por lo que claramente el individuo #100 tiene una probabilidad máxima de ser un buen pagador.

Naïve Bayes en R

Paquete 'e1071'

Instalación y carga

```
install.packages('e1071', dependencies = TRUE)
```

```
library(class)
```

```
library(e1071)
```




Aprendizaje Supervisado

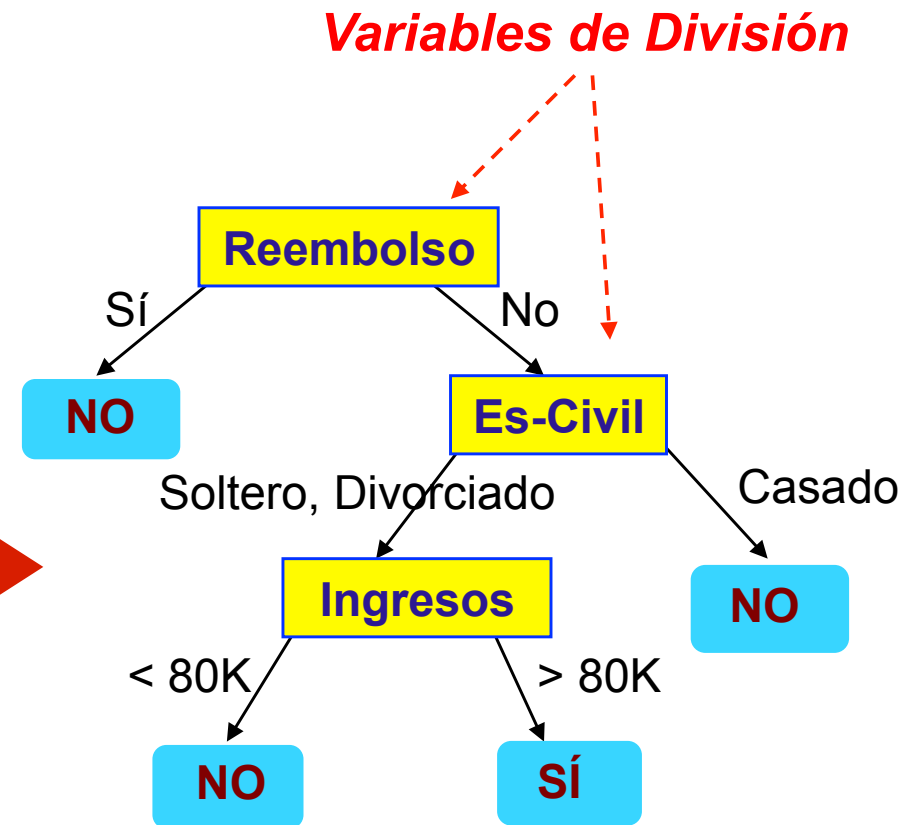
Árboles de Decisión

Un ejemplo de un árbol de decisión

categórica categórica continua clase

Id	Reembolso	Estado Civil	Ingresos Anuales	Fraude
1	Sí	Soltero	125K	No
2	No	Casado	100K	No
3	No	Soltero	70K	No
4	Sí	Casado	120K	No
5	No	Divorciado	95K	Sí
6	No	Casado	60K	No
7	Sí	Divorciado	220K	No
8	No	Soltero	85K	Sí
9	No	Casado	75K	No
10	No	Soltero	90K	Sí

Tabla de Aprendizaje



Modelo: Árbol de Decisión



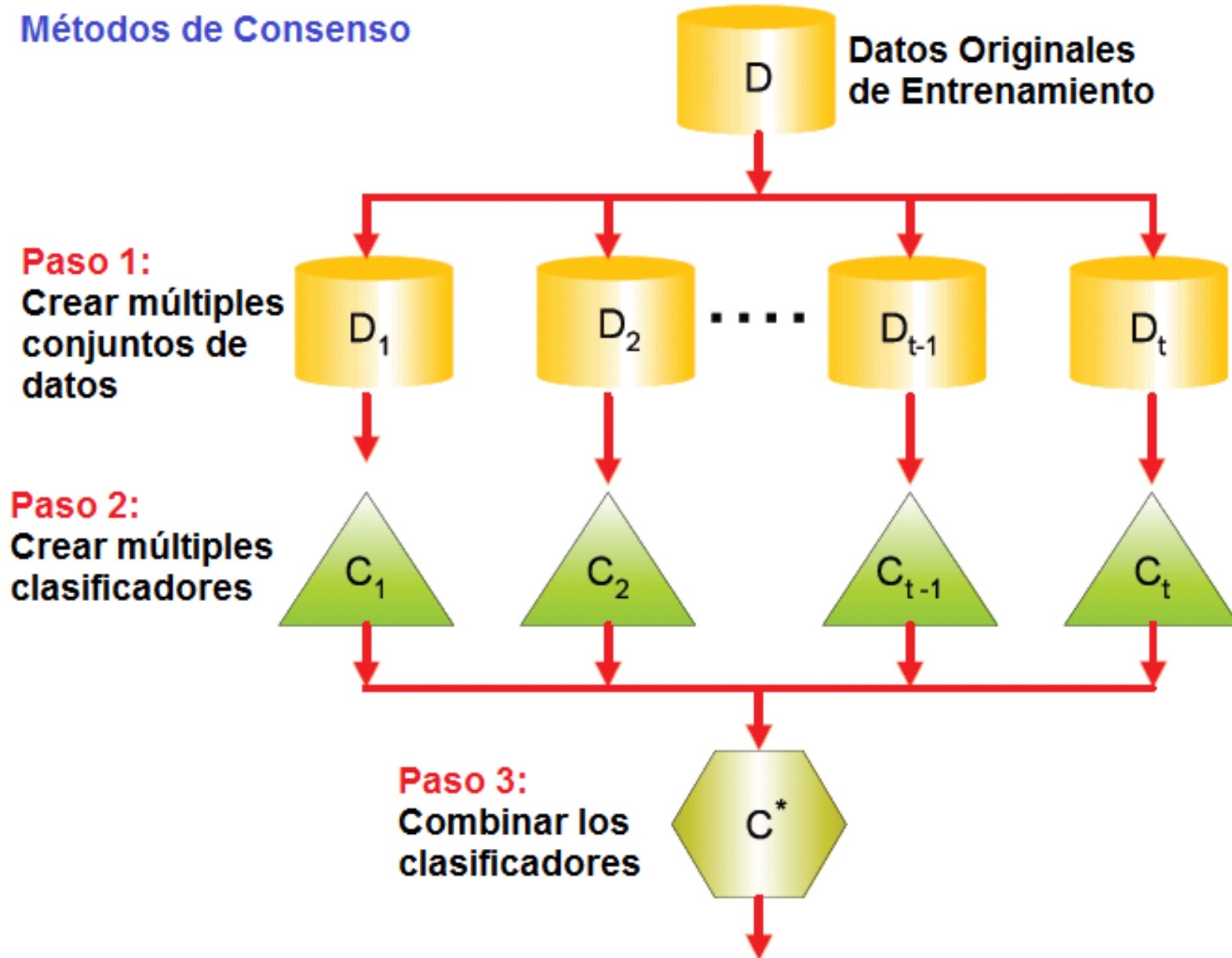
Aprendizaje Supervisado

Métodos de Consenso (Bagging)

Métodos de Consenso (Bagging)

- La idea es tomar m muestras aleatorias con reemplazo (Boostraps) de los datos originales y luego aplicar en cada una de ellas un método predictivo para luego con algún criterio establecer un consenso de todos los resultados
- El consenso podría ser un promedio, un promedio ponderado basado en cuál método obtuvo los mejores resultados
- El consenso más usado es el que obtenga la “***mayor cantidad de votos***”

Métodos de Consenso



Métodos de Consenso (Bagging)

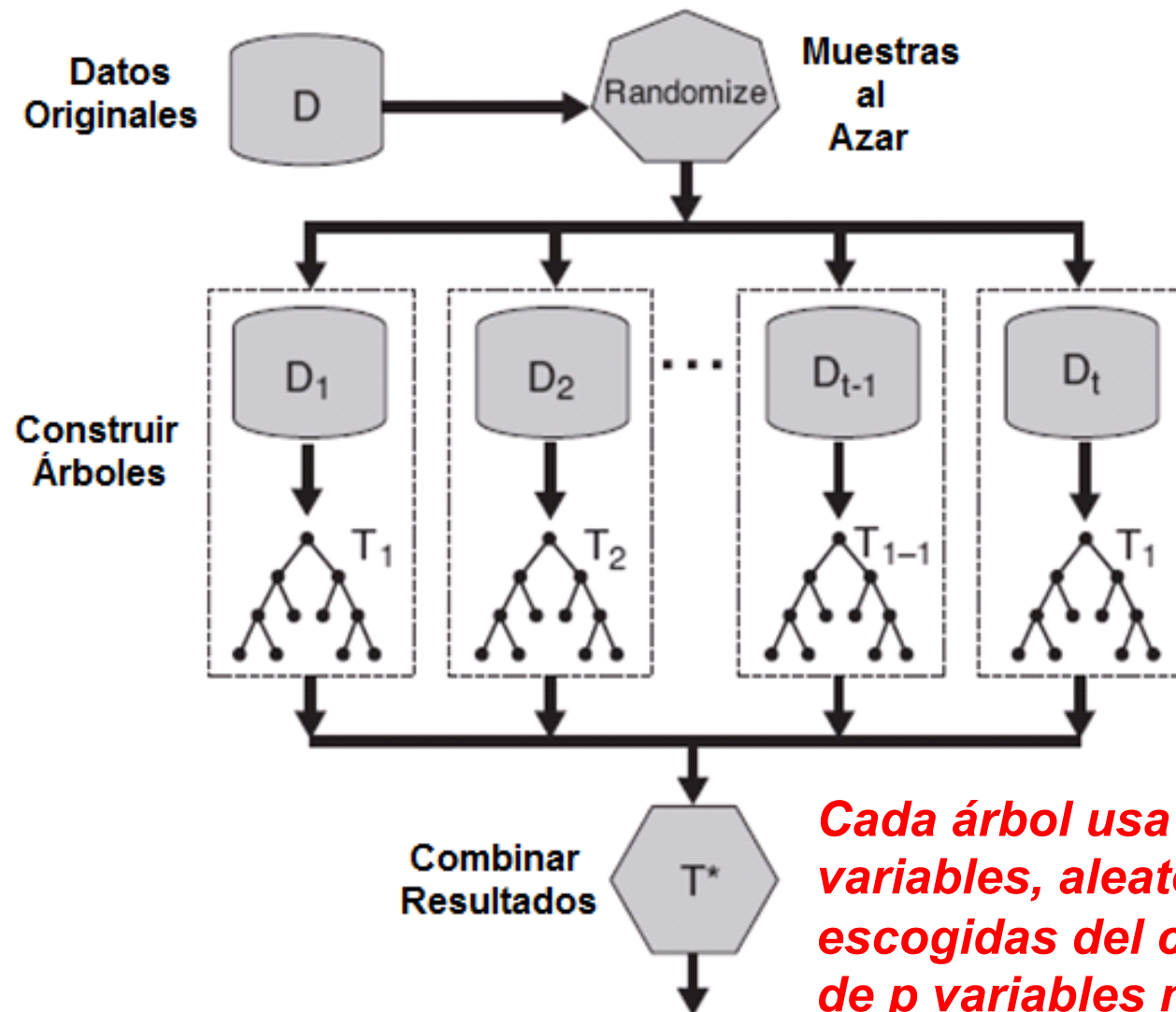
- Cada muestra de arranque (bootstrap) tiene el mismo tamaño que los datos originales. Debido a que el muestreo se realiza con reemplazo, algunos individuos pueden aparecer varias veces en el mismo conjunto de entrenamiento, mientras que otros podrían no parecer nunca.
- **Teorema:** En promedio, una muestra de arranque D_i (bootstrap) contiene aproximadamente 63% de los datos de entrenamiento originales.
- **Prueba:** Cada individuo tiene una probabilidad de $1 - (1 - 1/n)^n$ de ser seleccionados en cada D_i . Si n es suficientemente grande, esta probabilidad converge a $1 - \frac{1}{e} = 0.632$.

Bosques Aleatorios (Random Forest)

- El caso en el que todos los clasificadores del Método de Consenso son Árboles dicho método se denomina Bosques Aleatorios (Random Forest)



Bosques Aleatorios (Random Forest)



Cada árbol usa m diferentes variables, aleatoriamente escogidas del conjunto de p variables $m < p$ ($m = m_{try}$ en R)



Aprendizaje Supervisado

Métodos Potenciación

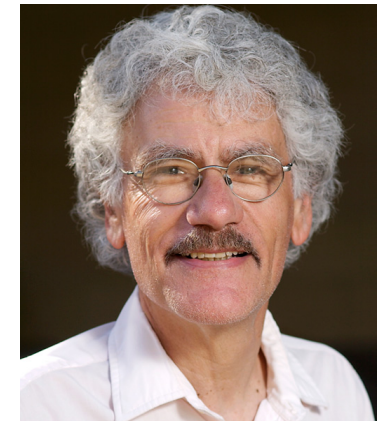
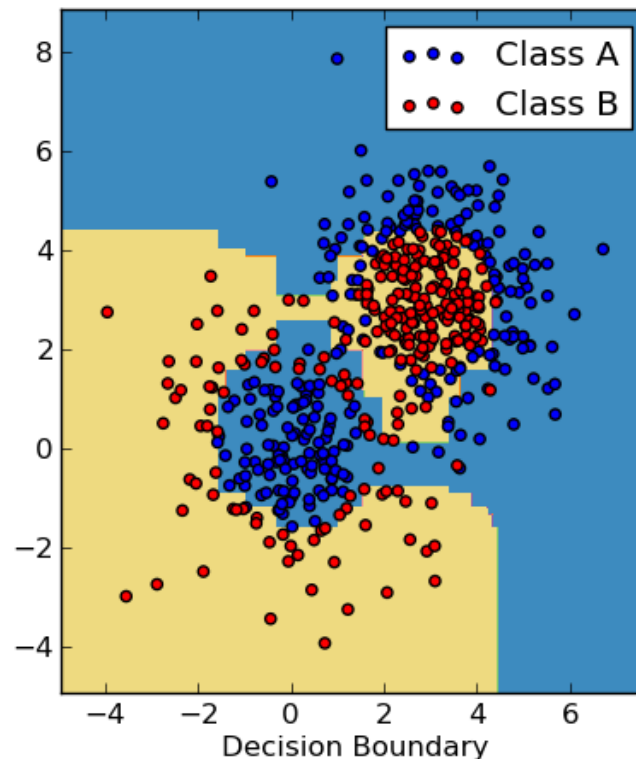
Métodos de Potenciación

"Best off-the-shelf classifier in the world"

[Breiman, NIPS Workshop, 1996]



Breiman

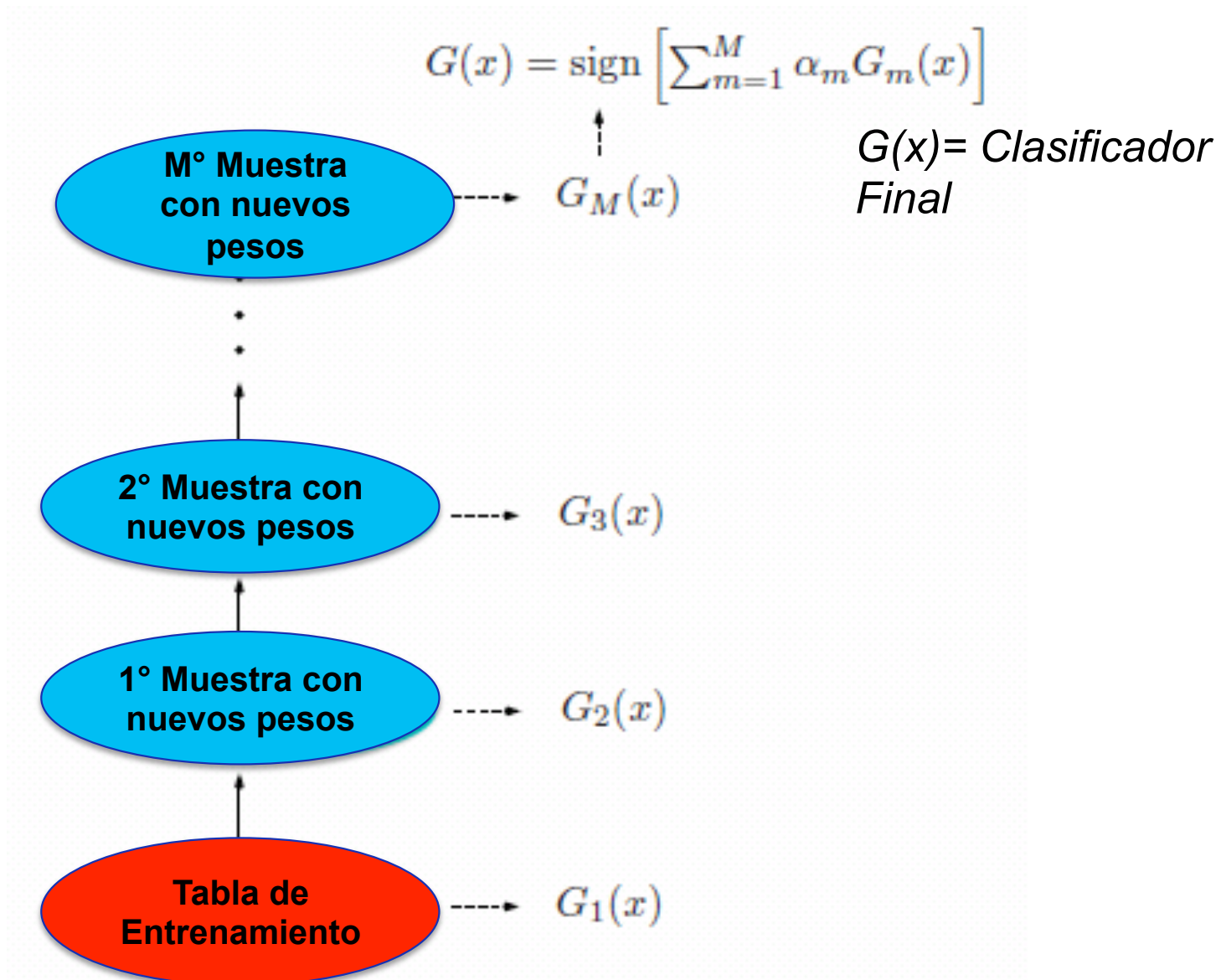


Friedman

Métodos de Potenciación

- La idea es tomar una muestra aleatoria de los datos originales y aplicar sobre esta un método clasificador luego *aumentar el peso (potenciar)* a los individuos mal clasificados para que en la siguiente aplicación del método clasificador se enfoque más en estos individuos mal clasificados, mejorando su clasificación, y así sucesivamente ...
- **Observación:** Solo funciona para problemas de clasificación **binarios** (de 2 clases).

Métodos de Potenciación



Métodos de Potenciación

- ✓ Al inicio el algoritmo de Potenciación utiliza los mismos pesos ($1/n$) en todos los individuos para determinar la distribución de muestreo de la muestra de entrenamiento, es decir, todos los individuos tienen la misma probabilidad de ser elegido.
- ✓ A continuación, un clasificador se aplica en la muestra de entrenamiento y se utiliza para clasificar todos los datos originales.
- ✓ Luego los pesos de los individuos de entrenamiento se actualizan para “impulsarlos” en la siguiente ejecución del clasificador. Así los individuos que se clasificaron incorrectamente tendrán un incremento en sus pesos, mientras que aquellos que se clasificaron correctamente tendrán una disminución en su peso.
- ✓ Esto obliga, en iteraciones posteriores, al clasificador a concentrarse en los individuos que son más difíciles de clasificar.

Métodos de Potenciación

Algoritmo: AdaBoost.M1

1. Initialize the observation weights $w_i = 1/N$, $i = 1, 2, \dots, N$.
2. For $m = 1$ to M :
 - (a) Fit a classifier $G_m(x)$ to the training data using weights w_i .
 - (b) Compute
$$\text{err}_m = \frac{\sum_{i=1}^N w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^N w_i}.$$
 - (c) Compute $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$.
 - (d) Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))]$, $i = 1, 2, \dots, N$.
3. Output $G(x) = \text{sign} \left[\sum_{m=1}^M \alpha_m G_m(x) \right]$.

Ejemplo: Algoritmo:AdaBoost.M1

Boosting Round 1:

x	0.1	0.4	0.5	0.6	0.6	0.7	0.7	0.7	0.8	1
y	1	-1	-1	-1	-1	-1	-1	-1	1	1

Boosting Round 2:

x	0.1	0.1	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3
y	1	1	1	1	1	1	1	1	1	1

Boosting Round 3:

x	0.2	0.2	0.4	0.4	0.4	0.4	0.5	0.6	0.6	0.7
y	1	1	-1	-1	-1	-1	-1	-1	-1	-1

(a) Training records chosen during boosting

Round	x=0.1	x=0.2	x=0.3	x=0.4	x=0.5	x=0.6	x=0.7	x=0.8	x=0.9	x=1.0
1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2	0.311	0.311	0.311	0.01	0.01	0.01	0.01	0.01	0.01	0.01
3	0.029	0.029	0.029	0.228	0.228	0.228	0.228	0.009	0.009	0.009

(b) Weights of training records