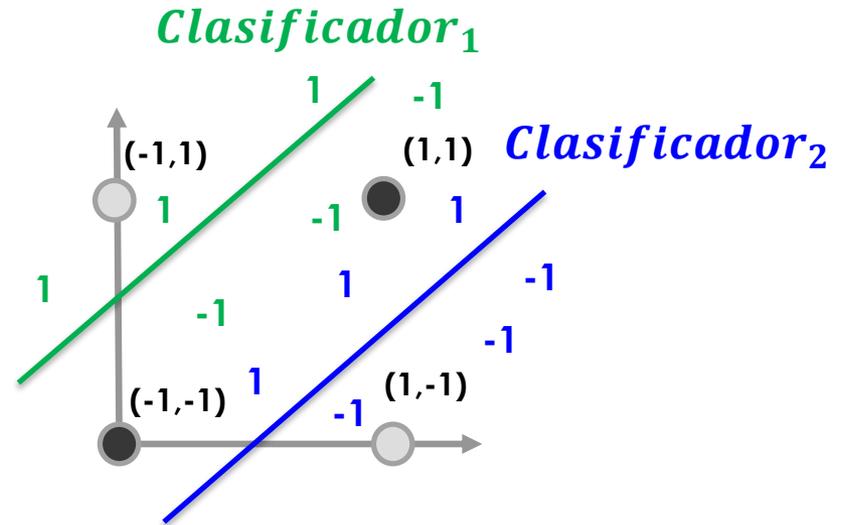
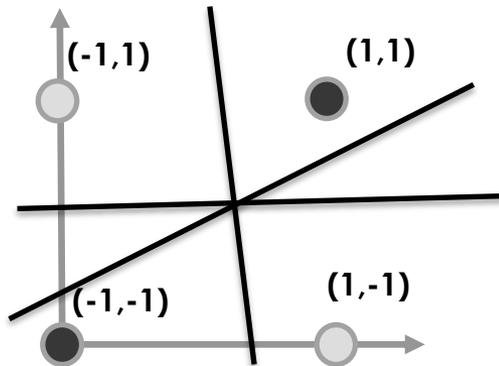
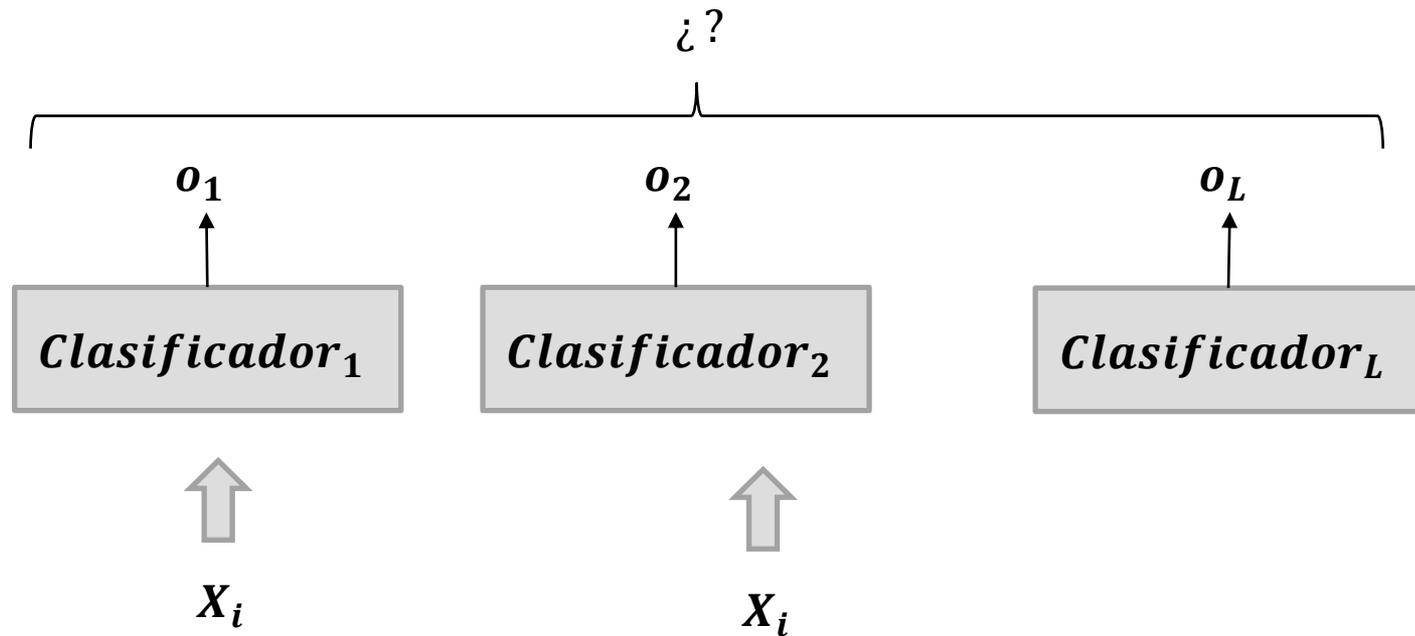


Combinación de Clasificadores

Motivación

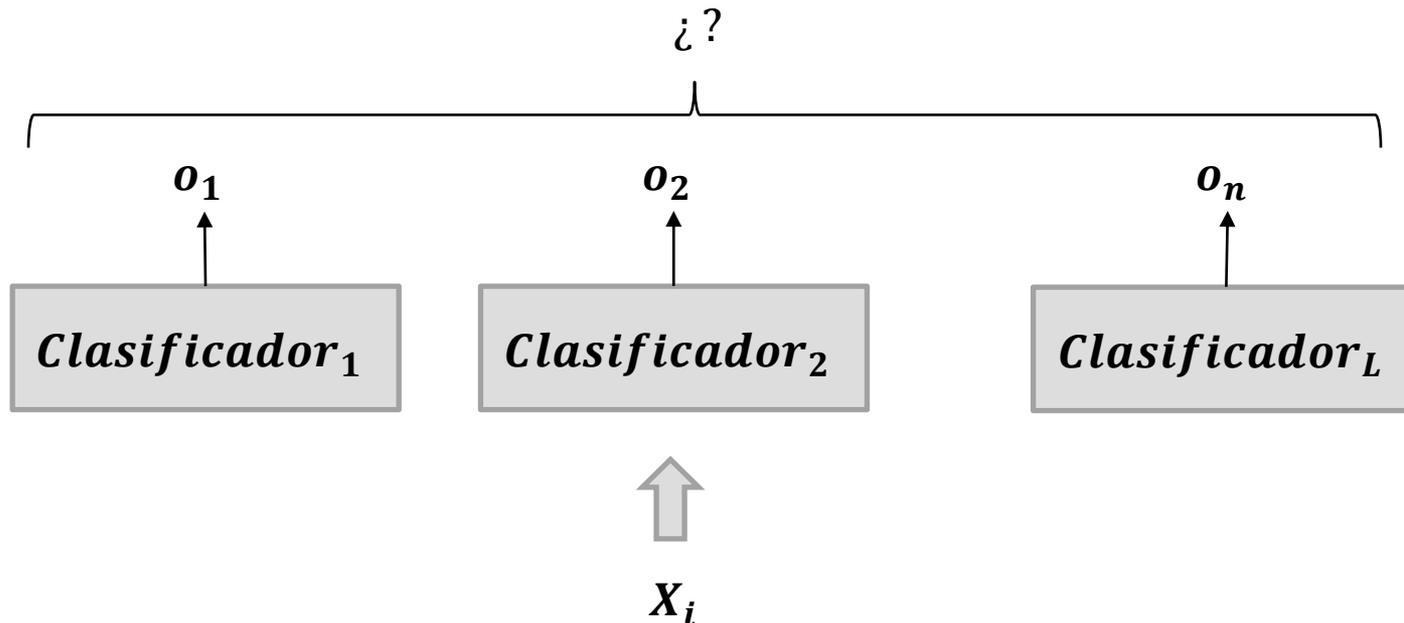


- si *Clasificador*₁ = 1 y *Clasificador*₂ = 1 entonces 1
 si *Clasificador*₁ = -1 y *Clasificador*₂ = 1 entonces -1
 si *Clasificador*₁ = -1 y *Clasificador*₂ = -1 entonces 1



¿Cuál es la salida combinada del sistema?

- Clasificación: clase mayoritaria entre $\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_L$
- Regresión: Promedio de $\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_L$



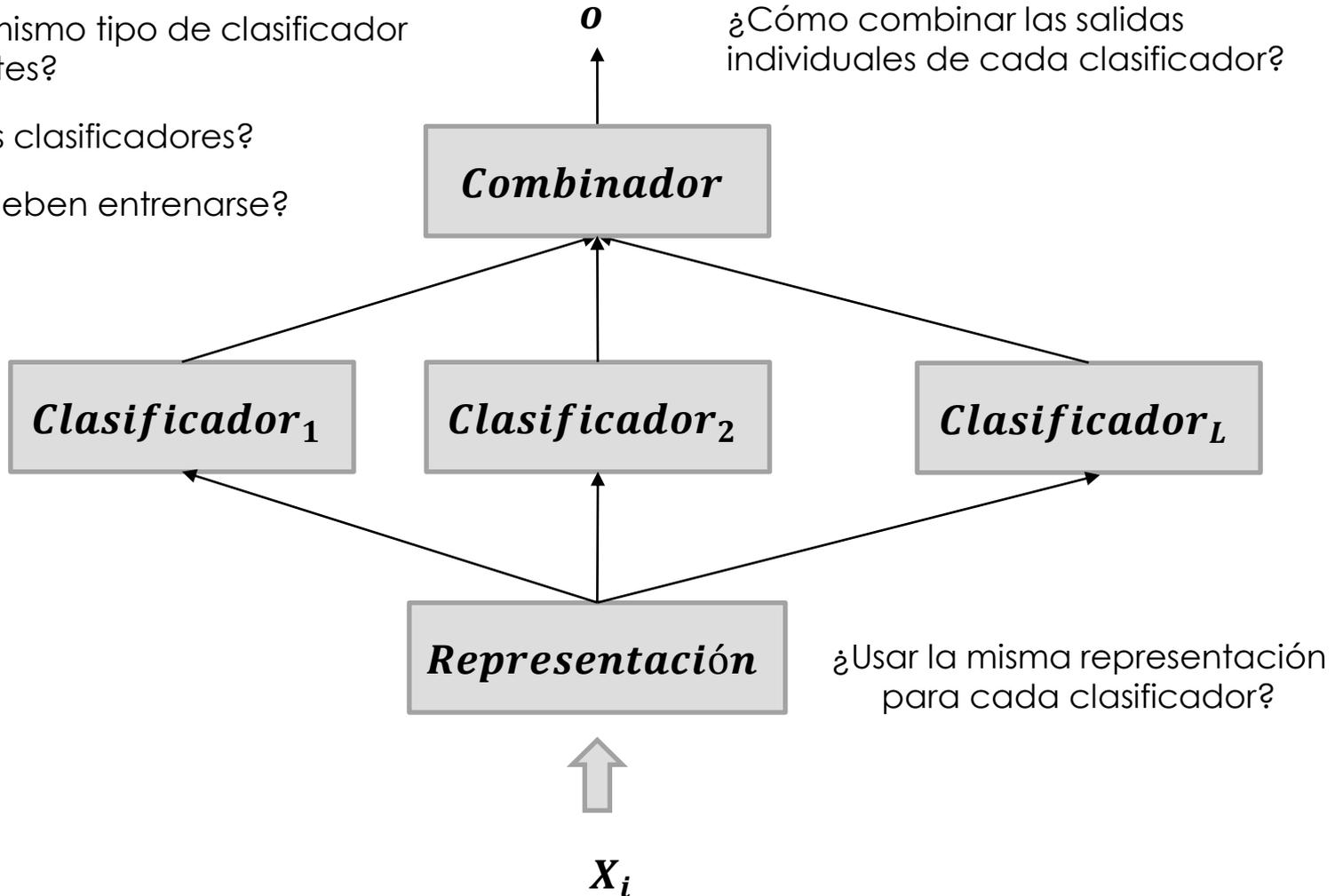
- Los datos para el entrenamiento pueden no proveer suficiente información para elegir un único mejor clasificador
- La combinación redundante y complementaria de clasificadores mejora la robustez, exactitud y generalidad de toda la clasificación
- Diferentes clasificadores utilizan distintas técnicas y métodos de representación de los datos, lo que permite obtener resultados de clasificación con diferentes patrones de generalización

¿Usar el mismo tipo de clasificador o diferentes?

¿Cuántos clasificadores?

¿Cómo deben entrenarse?

¿Cómo combinar las salidas individuales de cada clasificador?



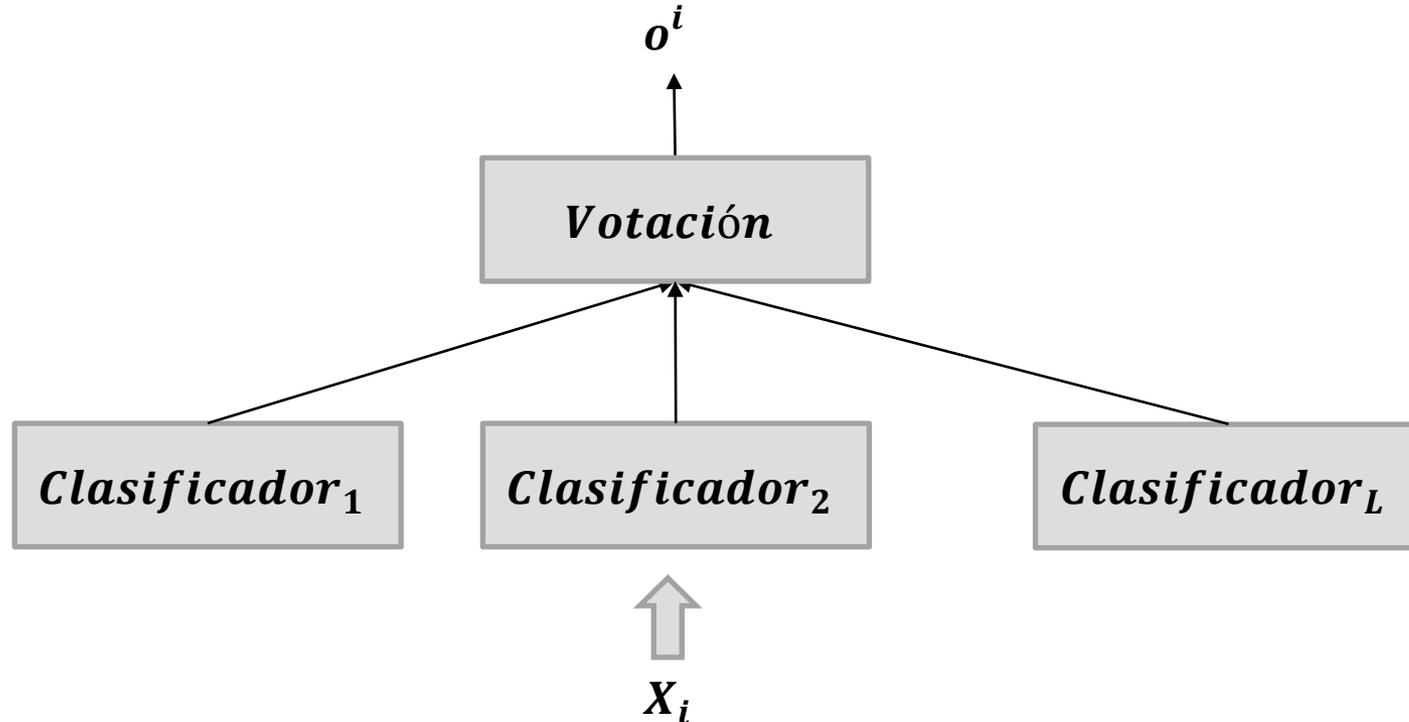
Votación

Bagging

Stacking

Boosting

Combinación de clasificadores por votación



- Voto por mayoría
- Voto por mayoría ponderado

Bagging (Bootstrap AGGREGATING)

- Se construyen nuevos conjuntos de entrenamiento replicando los datos de entrenamiento originales mediante muestreo con reemplazo (bootstrap).
- El conjunto de clasificadores se construye a partir de clasificadores entrenados con los diferentes conjuntos de entrenamiento.
- La salida combinada se obtiene mediante votación.
- Los clasificadores deben ser “inestables” es decir, pequeños cambios en los datos deben conducir a grandes cambios en la salida del clasificador. Por ejemplo, las Redes Neuronales y los Árboles de Decisión.
- Se busca que exista variedad entre un clasificador y otro, y no crear un conjunto de clasificadores casi idénticos.

L. Breiman. Bagging predictors. Technical Report 421, Department of Statistics, University of California, Berkeley, CA, 1994.

Bagging (Bootstrap AGGregating)

Entrenamiento

Dado el conjunto de entrenamiento $X = \{x_1, x_2, \dots, x_n\}$, la cantidad L de clasificadores y el tipo de clasificador base

- 1- Crear L conjuntos X_1, X_2, \dots, X_L de entrenamiento a partir de X
- 2- Entrenar los L clasificadores C_1, C_2, \dots, C_L , uno con cada X_i

Clasificación de nuevas instancias

- 1- Dada una instancia x_i , clasificarla con cada C_1, C_2, \dots, C_L
- 2- Obtener la salida combinada mediante una votación entre los clasificadores

Fin

AdaBoost (ADAPtative BOOsting)

- Se propone obtener un clasificador preciso a partir de la combinación de clasificadores moderadamente imprecisos llamados clasificadores débiles.
- Construye el conjunto de clasificadores de manera incremental, añadiendo un clasificador a la vez.
- El clasificador que se añade en la iteración k se entrena a partir de una muestra del conjunto de entrenamiento X , priorizando instancias que fueron mal clasificadas por los clasificadores ya integrados en el conjunto.
- Cada instancia x_1, x_2, \dots, x_n tiene asociado un peso $w_1^k, w_2^k, \dots, w_n^k \in [0,1]$ y $\sum_{i=1}^n w_i^k = 1$ que determina que tan importante es en el entrenamiento.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

AdaBoost (ADAPtative BOOsting)

Entrenamiento

Dado el conjunto de entrenamiento $X = \{x_1, x_2, \dots, x_n\}$, la cantidad L de clasificadores y el tipo de clasificador base

1- Fijar los pesos $w_1^1, w_2^1, \dots, w_n^1$, usualmente $w_i^1 = \frac{1}{n}$

2- Para cada $k = 1 \dots L$

2.1 Seleccionar una muestra X_k de X de acuerdo a la distribución que establecen los pesos, probabilidad de ser seleccionado directamente proporcional a w_i^k

2.2 Entrenar el clasificador C_k utilizando X_k

2.3 Calcular el error ponderado en el paso k como $\varepsilon_k = \sum_{i=1}^n w_i^k * l_k^i$
donde $l_k^i = \begin{cases} 1 & \text{si } C_k \text{ clasifica incorrectamente a } x_i \\ 0 & \text{eoc} \end{cases}$

2.4 **Si** $\varepsilon_k = 0$ reiniciar los pesos y continuar, **si no**:

2.4.1 **Si** $\varepsilon_k > 0.5$, ignorar C_k , reiniciar los pesos y continuar, **si no**, calcular $\beta_k = \frac{\varepsilon_k}{1-\varepsilon_k}$ donde $0 < \varepsilon_k < 0.5$

2.4.2 Actualizar los pesos de acuerdo a $w_j^{k+1} = \frac{w_j^k * \beta_k^{1-l_k^j}}{\sum_{i=1}^n w_i^k * \beta_k^{1-l_k^i}}$

3- Devolver C_1, C_2, \dots, C_L

Fin

AdaBoost (ADAPtative BOOsting)

Clasificación de nuevas instancias

1- Dada una instancia x_i , clasificarla con cada C_1, C_2, \dots, C_L

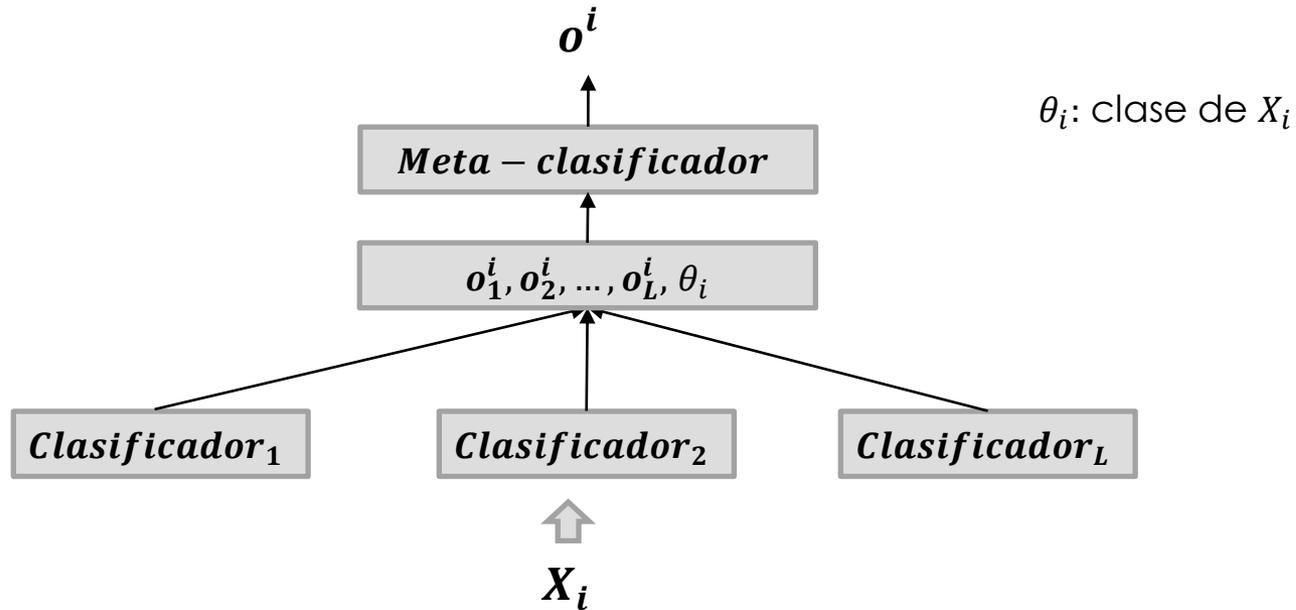
2- Calcular la confianza para la clase θ_t como $\mu_t(x_i) = \sum_{C_k(x_i)} \ln\left(\frac{1}{\beta_k}\right)$

3- Clasificar x_i de acuerdo a la clase con mayor confianza

Fin

- A diferencia de Bagging y Boosting, normalmente se combinan clasificadores de diferente tipo
- En general no hay consenso sobre la mejor forma de configurar el conjunto de clasificadores
- Se introduce el concepto de meta-clasificador, un algoritmo que toma como entrada las respuestas que obtienen cada algoritmo base. El meta-clasificador intenta aprender la mejor forma de combinar las salidas de los clasificadores bases
- En general, no se entrenan los clasificadores base utilizando todos los datos. Se reserva una parte para, una vez entrenados, evaluarlos y añadir esos resultados a los datos con los que se entrenará el metaclasificador

David H. Wolpert (1992). Stacked generalization. Neural Networks. 5:241-259.



David H. Wolpert (1992). Stacked generalization. *Neural Networks*. 5:241-259.

Entrenamiento

Dado el conjunto de entrenamiento $X = \{x_1, x_2, \dots, x_n\}$, la cantidad L de clasificadores y el tipo de clasificador base

- 1- Crear conjuntos X_{Tr} , X_{Te} a partir de X
- 2- Entrenar los L clasificadores C_1, C_2, \dots, C_L , con cada X_{Tr}
- 3- Evaluar cada instancia en X_{Te}
- 4- Construir X^M como la unión de los resultados de C_1, C_2, \dots, C_L en las instancias tanto en X_{Tr} como en X_{Te} . Notar que cada instancia en X^M tendrá la forma $x_i = \langle o_1^i, o_2^i, \dots, o_L^i, \theta_i \rangle$
- 5- Entrenar el meta-clasificador

Clasificación de nuevas instancias

- 1- Dada una instancia x_j , clasificarla con cada C_1, C_2, \dots, C_L y obtener $\langle o_1^j, o_2^j, \dots, o_L^j, ? \rangle$
- 2- Evaluar el meta-clasificador en $\langle o_1^j, o_2^j, \dots, o_L^j, ? \rangle$ y devolver su salida

Fin

Fin