

Tema 13: Combinación de Clasificadores

Pedro Larrañaga, Iñaki Inza, Abdelmalik Moujahid

Intelligent Systems Group
Departamento de Ciencias de la Computación e Inteligencia Artificial
Universidad del País Vasco

Índice

- 1 **Introducción**
- 2 **Hibridaciones**
- 3 **Combinando el mismo clasificador base**
- 4 **Combinando distintos clasificadores base**
- 5 **Conclusiones**

Índice

- 1** **Introducción**
- 2 Hibridaciones
- 3 Combinando el mismo clasificador base
- 4 Combinando distintos clasificadores base
- 5 Conclusiones

Motivación

- *Non free-lunch theorem* (Wolpert y MacReady, 1996) en minería de datos
- **Combinación de opiniones** de expertos (modelos) antes de tomar una decisión final
- Cada paradigma asociado una **región de decisión** de un determinado tipo
- Al combinar paradigmas tratar de obtener la **región de decisión idonéa** para el problema

Motivación

- *Non free-lunch theorem* (Wolpert y MacReady, 1996) en minería de datos
- **Combinación de opiniones** de expertos (modelos) antes de tomar una decisión final
- Cada paradigma asociado una **región de decisión** de un determinado tipo
- Al combinar paradigmas tratar de obtener la **región de decisión idónea** para el problema

Motivación

- *Non free-lunch theorem* (Wolpert y MacReady, 1996) en minería de datos
- **Combinación de opiniones** de expertos (modelos) antes de tomar una decisión final
- Cada paradigma asociado una **región de decisión** de un determinado tipo
- Al combinar paradigmas tratar de obtener la **región de decisión idónea** para el problema

Motivación

- *Non free-lunch theorem* (Wolpert y MacReady, 1996) en minería de datos
- **Combinación de opiniones** de expertos (modelos) antes de tomar una decisión final
- Cada paradigma asociado una **región de decisión** de un determinado tipo
- Al combinar paradigmas tratar de obtener la **región de decisión idonéa** para el problema

Motivación

- *Non free-lunch theorem* (Wolpert y MacReady, 1996) en minería de datos
- **Combinación de opiniones** de expertos (modelos) antes de tomar una decisión final
- Cada paradigma asociado una **región de decisión** de un determinado tipo
- Al combinar paradigmas tratar de obtener la **región de decisión idonéa** para el problema

Tipos de combinaciones de clasificadores

- **Hibridaciones**: el clasificador se induce teniendo en cuenta dos o más paradigmas
- Combinación de decisiones obtenidas con el **mismo clasificador de base**
- Combinación de decisiones obtenidas con **distintos clasificadores de base**

Tipos de combinaciones de clasificadores

- **Hibridaciones**: el clasificador se induce teniendo en cuenta dos o más paradigmas
 - Combinación de decisiones obtenidas con el **mismo clasificador de base**
 - Combinación de decisiones obtenidas con **distintos clasificadores de base**

Tipos de combinaciones de clasificadores

- **Hibridaciones**: el clasificador se induce teniendo en cuenta dos o más paradigmas
- Combinación de decisiones obtenidas con el **mismo clasificador de base**
- Combinación de decisiones obtenidas con **distintos clasificadores de base**

Tipos de combinaciones de clasificadores

- **Hibridaciones**: el clasificador se induce teniendo en cuenta dos o más paradigmas
- Combinación de decisiones obtenidas con el **mismo clasificador de base**
- Combinación de decisiones obtenidas con **distintos clasificadores de base**

Índice

1 Introducción

2 Hibridaciones

3 Combinando el mismo clasificador base

4 Combinando distintos clasificadores base

5 Conclusiones

Hibridaciones

- **Lazy Bayesian Rules** (Zheng y Webb, 2000): para cada ejemplo a clasificar obtiene un naive Bayes con aquellos casos mas cercanos
- **Naive Bayes Tree** (Kohavi, 1996): induce árboles cuyas hojas son clasificadores naive Bayes, los cuales se usarán en los ejemplos que alcancen dichas hojas
- **Logistic Model Trees** (Landwerhr y col., 2003): árboles en cuyas hojas se lleva a cabo la clasificación, de aquellas ejemplos que los alcancen, por medio de una regresión logística

Hibridaciones

- **Lazy Bayesian Rules** (Zheng y Webb, 2000): para cada ejemplo a clasificar obtiene un naive Bayes con aquellos casos mas cercanos
- **Naive Bayes Tree** (Kohavi, 1996): induce árboles cuyas hojas son clasificadores naive Bayes, los cuales se usarán en los ejemplos que alcancen dichas hojas
- **Logistic Model Trees** (Landwerhr y col., 2003): árboles en cuyas hojas se lleva a cabo la clasificación, de aquellas ejemplos que los alcancen, por medio de una regresión logística

Hibridaciones

- **Lazy Bayesian Rules** (Zheng y Webb, 2000): para cada ejemplo a clasificar obtiene un naive Bayes con aquellos casos mas cercanos
- **Naive Bayes Tree** (Kohavi, 1996): induce árboles cuyas hojas son clasificadores naive Bayes, los cuales se usarán en los ejemplos que alcancen dichas hojas
- **Logistic Model Trees** (Landwerhr y col., 2003): árboles en cuyas hojas se lleva a cabo la clasificación, de aquellas ejemplos que los alcancen, por medio de una regresión logística

Hibridaciones

- **Lazy Bayesian Rules** (Zheng y Webb, 2000): para cada ejemplo a clasificar obtiene un naive Bayes con aquellos casos mas cercanos
- **Naive Bayes Tree** (Kohavi, 1996): induce árboles cuyas hojas son clasificadores naive Bayes, los cuales se usarán en los ejemplos que alcancen dichas hojas
- **Logistic Model Trees** (Landwerhr y col., 2003): árboles en cuyas hojas se lleva a cabo la clasificación, de aquellas ejemplos que los alcancen, por medio de una regresión logística

Índice

- 1 Introducción
- 2 Hibridaciones
- 3 Combinando el mismo clasificador base**
- 4 Combinando distintos clasificadores base
- 5 Conclusiones

Bagging (Breiman, 1994)

Ideas básicas

- Bagging (**Bootstrap AGG**regat**ING**)
- Se combinan clasificadores obtenidos de réplicas **bootstrap** del conjunto de entrenamiento
- Para cada ejemplo se selecciona la clase más frecuentemente predicha
- Bagging buenos resultados con clasificadores inestables (pequeña variación en los datos produce gran variación en la clasificación)
- Árboles (inestables); k -NN (estables)

Bagging (Breiman, 1994)

Ideas básicas

- Bagging (**Bootstrap AGGregatING**)
- Se combinan clasificadores obtenidos de réplicas **bootstrap** del conjunto de entrenamiento
- Para cada ejemplo se selecciona la **clase más frecuentemente predicha**
- Bagging buenos resultados con **clasificadores inestables** (pequeña variación en los datos produce gran variación en la clasificación)
- **Árboles (inestables); k -NN (estables)**

Bagging (Breiman, 1994)

Ideas básicas

- Bagging (**Bootstrap AGG**regat**ING**)
- Se combinan clasificadores obtenidos de réplicas **bootstrap** del conjunto de entrenamiento
- Para cada ejemplo se selecciona la **clase más frecuentemente predicha**
- Bagging buenos resultados con **clasificadores inestables** (pequeña variación en los datos produce gran variación en la clasificación)
- **Árboles (inestables); k-NN (estables)**

Bagging (Breiman, 1994)

Ideas básicas

- Bagging (**Bootstrap AGGregatING**)
- Se combinan clasificadores obtenidos de réplicas **bootstrap** del conjunto de entrenamiento
- Para cada ejemplo se selecciona la **clase más frecuentemente predicha**
- Bagging buenos resultados con **clasificadores inestables** (pequeña variación en los datos produce gran variación en la clasificación)
- **Árboles (inestables); k-NN (estables)**

Bagging (Breiman, 1994)

Ideas básicas

- Bagging (**Bootstrap AGGregatING**)
- Se combinan clasificadores obtenidos de réplicas **bootstrap** del conjunto de entrenamiento
- Para cada ejemplo se selecciona la **clase más frecuentemente predicha**
- Bagging buenos resultados con **clasificadores inestables** (pequeña variación en los datos produce gran variación en la clasificación)
- **Árboles (inestables); k -NN (estables)**

Bagging (Breiman, 1994)

Ideas básicas

- Bagging (**Bootstrap AGGregatING**)
- Se combinan clasificadores obtenidos de réplicas **bootstrap** del conjunto de entrenamiento
- Para cada ejemplo se selecciona la **clase más frecuentemente predicha**
- Bagging buenos resultados con **clasificadores inestables** (pequeña variación en los datos produce gran variación en la clasificación)
- Árboles (inestables); *k*-NN (estables)

Bagging (Breiman, 1994)

Ideas básicas

- Bagging (**Bootstrap AGGregatING**)
- Se combinan clasificadores obtenidos de réplicas **bootstrap** del conjunto de entrenamiento
- Para cada ejemplo se selecciona la **clase más frecuentemente predicha**
- Bagging buenos resultados con **clasificadores inestables** (pequeña variación en los datos produce gran variación en la clasificación)
- **Árboles (inestables); k -NN (estables)**

Bagging (Breiman, 1994)

Esquema del algoritmo

- **Generación del modelo**
 - Sea N el número de ejemplos en el conjunto de entrenamiento
 - Para cada una de las t iteraciones:
 - Muestrar N ejemplos con remplazamiento del conjunto de entrenamiento
 - Aplicar el conjunto de entrenamiento a la muestra
 - Almacenar el modelo resultante
- **Clasificación**
 - Para cada uno de los t modelos:
 - Predecir la clase del ejemplo usando el modelo
 - Devolver la clase que ha sido predicha mas veces

Bagging (Breiman, 1994)

Esquema del algoritmo

- Generación del modelo
 - Sea N el número de ejemplos en el conjunto de entrenamiento
 - Para cada una de las t iteraciones:
 - Muestrar N ejemplos con remplazamiento del conjunto de entrenamiento
 - Aplicar el conjunto de entrenamiento a la muestra
 - Almacenar el modelo resultante
- Clasificación
 - Para cada uno de los t modelos:
 - Predecir la clase del ejemplo usando el modelo
 - Devolver la clase que ha sido predicha mas veces

Bagging (Breiman, 1994)

Esquema del algoritmo

- **Generación del modelo**
 - Sea N el número de ejemplos en el conjunto de entrenamiento
 - Para cada una de las t iteraciones:
 - Muestrar N ejemplos con remplazamiento del conjunto de entrenamiento
 - Aplicar el conjunto de entrenamiento a la muestra
 - Almacenar el modelo resultante
- Clasificación
 - Para cada uno de los t modelos:
 - Predecir la clase del ejemplo usando el modelo
 - Devolver la clase que ha sido predicha mas veces

Bagging (Breiman, 1994)

Esquema del algoritmo

- **Generación del modelo**
 - Sea N el número de ejemplos en el conjunto de entrenamiento
 - Para cada una de las t iteraciones:
 - Muestrar N ejemplos con remplazamiento del conjunto de entrenamiento
 - Aplicar el conjunto de entrenamiento a la muestra
 - Almacenar el modelo resultante
- Clasificación
 - Para cada uno de los t modelos:
 - Predecir la clase del ejemplo usando el modelo
 - Devolver la clase que ha sido predicha mas veces

Bagging (Breiman, 1994)

Esquema del algoritmo

- **Generación del modelo**
 - Sea N el número de ejemplos en el conjunto de entrenamiento
 - Para cada una de las t iteraciones:
 - Muestrar N ejemplos con remplazamiento del conjunto de entrenamiento
 - Aplicar el conjunto de entrenamiento a la muestra
 - Almacenar el modelo resultante
- Clasificación
 - Para cada uno de los t modelos:
 - Predecir la clase del ejemplo usando el modelo
 - Devolver la clase que ha sido predicha mas veces

Bagging (Breiman, 1994)

Esquema del algoritmo

- **Generación del modelo**
 - Sea N el número de ejemplos en el conjunto de entrenamiento
 - Para cada una de las t iteraciones:
 - Muestrar N ejemplos con remplazamiento del conjunto de entrenamiento
 - Aplicar el conjunto de entrenamiento a la muestra
 - Almacenar el modelo resultante
- Clasificación
 - Para cada uno de los t modelos:
 - Predecir la clase del ejemplo usando el modelo
 - Devolver la clase que ha sido predicha mas veces

Bagging (Breiman, 1994)

Esquema del algoritmo

- **Generación del modelo**
 - Sea N el número de ejemplos en el conjunto de entrenamiento
 - Para cada una de las t iteraciones:
 - Muestrar N ejemplos con remplazamiento del conjunto de entrenamiento
 - Aplicar el conjunto de entrenamiento a la muestra
 - Almacenar el modelo resultante
- Clasificación
 - Para cada uno de los t modelos:
 - Predecir la clase del ejemplo usando el modelo
 - Devolver la clase que ha sido predicha mas veces

Bagging (Breiman, 1994)

Esquema del algoritmo

- **Generación del modelo**
 - Sea N el número de ejemplos en el conjunto de entrenamiento
 - Para cada una de las t iteraciones:
 - Muestrar N ejemplos con remplazamiento del conjunto de entrenamiento
 - Aplicar el conjunto de entrenamiento a la muestra
 - Almacenar el modelo resultante
- Clasificación
 - Para cada uno de los t modelos:
 - Predecir la clase del ejemplo usando el modelo
 - Devolver la clase que ha sido predicha mas veces

Bagging (Breiman, 1994)

Esquema del algoritmo

- **Generación del modelo**
 - Sea N el número de ejemplos en el conjunto de entrenamiento
 - Para cada una de las t iteraciones:
 - Muestrar N ejemplos con remplazamiento del conjunto de entrenamiento
 - Aplicar el conjunto de entrenamiento a la muestra
 - Almacenar el modelo resultante
- **Clasificación**
 - Para cada uno de los t modelos:
 - Predecir la clase del ejemplo usando el modelo
 - Devolver la clase que ha sido predicha mas veces

Bagging (Breiman, 1994)

Esquema del algoritmo

- **Generación del modelo**
 - Sea N el número de ejemplos en el conjunto de entrenamiento
 - Para cada una de las t iteraciones:
 - Muestrar N ejemplos con remplazamiento del conjunto de entrenamiento
 - Aplicar el conjunto de entrenamiento a la muestra
 - Almacenar el modelo resultante
- **Clasificación**
 - Para cada uno de los t modelos:
 - Predecir la clase del ejemplo usando el modelo
 - Devolver la clase que ha sido predicha mas veces

Bagging (Breiman, 1994)

Esquema del algoritmo

- **Generación del modelo**
 - Sea N el número de ejemplos en el conjunto de entrenamiento
 - Para cada una de las t iteraciones:
 - Muestrar N ejemplos con remplazamiento del conjunto de entrenamiento
 - Aplicar el conjunto de entrenamiento a la muestra
 - Almacenar el modelo resultante
- **Clasificación**
 - Para cada uno de los t modelos:
 - Predecir la clase del ejemplo usando el modelo
 - Devolver la clase que ha sido predicha mas veces

Bagging (Breiman, 1994)

Esquema del algoritmo

- **Generación del modelo**
 - Sea N el número de ejemplos en el conjunto de entrenamiento
 - Para cada una de las t iteraciones:
 - Muestrar N ejemplos con remplazamiento del conjunto de entrenamiento
 - Aplicar el conjunto de entrenamiento a la muestra
 - Almacenar el modelo resultante
- **Clasificación**
 - Para cada uno de los t modelos:
 - Predecir la clase del ejemplo usando el modelo
 - Devolver la clase que ha sido predicha mas veces

Bagging (Breiman, 1994)

Random forests (Breiman, 2001)

- Variante de bagging con **árboles de clasificación como clasificador de base**
- Clasificador consistente en una colección de L clasificadores con estructura de árbol, donde cada árbol se ha construido a partir de un **vector aleatorio**
- El vector aleatorio puede consistir en:
 - Aleatoriedad en los datos
 - Aleatoriedad en las variables

Bagging (Breiman, 1994)

Random forests (Breiman, 2001)

- Variante de bagging con **árboles de clasificación como clasificador de base**
- Clasificador consistente en una colección de L clasificadores con estructura de árbol, donde cada árbol se ha construido a partir de un **vector aleatorio**
- El vector aleatorio puede consistir en:
 - **Aleatoriedad en los datos**
 - **Aleatoriedad en las variables**

Bagging (Breiman, 1994)

Random forests (Breiman, 2001)

- Variante de bagging con **árboles de clasificación como clasificador de base**
- Clasificador consistente en una colección de L clasificadores con estructura de árbol, donde cada árbol se ha construido a partir de un **vector aleatorio**
- El vector aleatorio puede consistir en:
 - Aleatoriedad en los datos
 - Aleatoriedad en las variables

Bagging (Breiman, 1994)

Random forests (Breiman, 2001)

- Variante de bagging con **árboles de clasificación como clasificador de base**
- Clasificador consistente en una colección de L clasificadores con estructura de árbol, donde cada árbol se ha construido a partir de un **vector aleatorio**
- El vector aleatorio puede consistir en:
 - Aleatoriedad en los datos
 - Aleatoriedad en las variables

Bagging (Breiman, 1994)

Random forests (Breiman, 2001)

- Variante de bagging con **árboles de clasificación como clasificador de base**
- Clasificador consistente en una colección de L clasificadores con estructura de árbol, donde cada árbol se ha construido a partir de un **vector aleatorio**
- El vector aleatorio puede consistir en:
 - Aleatoriedad en los datos
 - Aleatoriedad en las variables

Bagging (Breiman, 1994)

Random forests (Breiman, 2001)

- Variante de bagging con **árboles de clasificación como clasificador de base**
- Clasificador consistente en una colección de L clasificadores con estructura de árbol, donde cada árbol se ha construido a partir de un **vector aleatorio**
- El vector aleatorio puede consistir en:
 - **Aleatoriedad en los datos**
 - Aleatoriedad en las variables

Bagging (Breiman, 1994)

Random forests (Breiman, 2001)

- Variante de bagging con **árboles de clasificación como clasificador de base**
- Clasificador consistente en una colección de L clasificadores con estructura de árbol, donde cada árbol se ha construido a partir de un **vector aleatorio**
- El vector aleatorio puede consistir en:
 - **Aleatoriedad en los datos**
 - **Aleatoriedad en las variables**

Boosting (Freund y Shapire, 1997)

Ideas básicas

- El conjunto de clasificador se desarrolla de manera **incremental**, añadiendo uno en cada iteración
- El clasificador en el paso t se entrena sobre un conjunto de entrenamiento **muestreado de manera selectiva**
- La distribución de muestreo comienza siendo **uniforme**
- Los objetos **mal clasificados** en la iteración anterior ven **aumentada su probabilidad de ser seleccionados**

Boosting (Freund y Shapire, 1997)

Ideas básicas

- El conjunto de clasificador se desarrolla de manera **incremental**, añadiendo uno en cada iteración
- El clasificador en el paso t se entrena sobre un conjunto de entrenamiento **muestreado de manera selectiva**
- La distribución de muestreo comienza siendo **uniforme**
- Los objetos **mal clasificados** en la iteración anterior ven **aumentada su probabilidad** de ser seleccionados

Boosting (Freund y Shapire, 1997)

Ideas básicas

- El conjunto de clasificador se desarrolla de manera **incremental**, añadiendo uno en cada iteración
- El clasificador en el paso t se entrena sobre un conjunto de entrenamiento **muestreado de manera selectiva**
- La distribución de muestreo comienza siendo **uniforme**
- Los objetos **mal clasificados** en la iteración anterior ven **aumentada su probabilidad** de ser seleccionados

Boosting (Freund y Shapire, 1997)

Ideas básicas

- El conjunto de clasificador se desarrolla de manera **incremental**, añadiendo uno en cada iteración
- El clasificador en el paso t se entrena sobre un conjunto de entrenamiento **muestreado de manera selectiva**
- La distribución de muestreo comienza siendo **uniforme**
- Los objetos **mal clasificados** en la iteración anterior ven **aumentada su probabilidad** de ser seleccionados

Boosting (Freund y Shapire, 1997)

Ideas básicas

- El conjunto de clasificador se desarrolla de manera **incremental**, añadiendo uno en cada iteración
- El clasificador en el paso t se entrena sobre un conjunto de entrenamiento **muestreado de manera selectiva**
- La distribución de muestreo comienza siendo **uniforme**
- Los objetos **mal clasificados** en la iteración anterior ven **aumentada su probabilidad** de ser seleccionados

Boosting (Freund y Shapire, 1997)

Ideas básicas

- El conjunto de clasificador se desarrolla de manera **incremental**, añadiendo uno en cada iteración
- El clasificador en el paso t se entrena sobre un conjunto de entrenamiento **muestreado de manera selectiva**
- La distribución de muestreo comienza siendo **uniforme**
- Los objetos **mal clasificados** en la iteración anterior ven **aumentada su probabilidad** de ser seleccionados

Boosting (Freund y Shapire, 1997)

AdaBoost

Fase de entrenamiento

- 1 Inicialización de parámetros: peso de cada uno de los N ejemplos: $w_j^1 = \frac{1}{N}$; número de iteraciones L ; conjunto de clasificadores $\mathcal{D} = \emptyset$
- 2 Para $k = 1 \dots L$
 - Extraer una muestra S_k del conjunto de datos usando la distribución w_j^k
 - Inducir el clasificador D_k usando S_k como conjunto de entrenamiento
 - Calcular el error en la iteración k : $\varepsilon_k = \sum_{j=1}^N w_j^k \delta_j^k$
 - If $\varepsilon_k = 0$ o $\varepsilon_k \geq 0,5$, ignorar D_k , reinicializar w_j^k a $\frac{1}{N}$ y continuar
 - Else, computar: $\beta_k = \frac{\varepsilon_k}{1-\varepsilon_k}$
 - Actualizar los pesos:

$$w_j^{k+1} = w_j^k \beta_k^{(1-\delta_j^k)} / \sum_{i=1}^N w_i^k \beta_k^{(1-\delta_i^k)}$$
- 3 Devolver \mathcal{D} y β_1, \dots, β_L

Boosting (Freund y Shapire, 1997)

AdaBoost

Fase de entrenamiento

- 1 Inicialización de parámetros: peso de cada uno de los N ejemplos: $w_j^1 = \frac{1}{N}$; número de iteraciones L ; conjunto de clasificadores $\mathcal{D} = \emptyset$
- 2 Para $k = 1 \dots L$
 - Extraer una muestra S_k del conjunto de datos usando la distribución w_j^k
 - Inducir el clasificador D_k usando S_k como conjunto de entrenamiento
 - Calcular el error en la iteración k : $\varepsilon_k = \sum_{j=1}^N w_j^k \delta_j^k$
 - If $\varepsilon_k = 0$ o $\varepsilon_k \geq 0,5$, ignorar D_k , reinicializar w_j^k a $\frac{1}{N}$ y continuar
 - Else, computar: $\beta_k = \frac{\varepsilon_k}{1-\varepsilon_k}$
 - Actualizar los pesos:

$$w_j^{k+1} = w_j^k \beta_k^{(1-\delta_j^k)} / \sum_{i=1}^N w_i^k \beta_k^{(1-\delta_i^k)}$$
- 3 Devolver \mathcal{D} y β_1, \dots, β_L

Boosting (Freund y Shapire, 1997)

AdaBoost

Fase de entrenamiento

- 1 Inicialización de parámetros: peso de cada uno de los N ejemplos: $w_j^1 = \frac{1}{N}$; número de iteraciones L ; conjunto de clasificadores $\mathcal{D} = \emptyset$
- 2 Para $k = 1 \dots L$
 - Extraer una muestra S_k del conjunto de datos usando la distribución w_j^k
 - Inducir el clasificador D_k usando S_k como conjunto de entrenamiento
 - Calcular el error en la iteración k : $\varepsilon_k = \sum_{j=1}^N w_j^k \delta_j^k$
 - If $\varepsilon_k = 0$ o $\varepsilon_k \geq 0,5$, ignorar D_k , reinicializar w_j^k a $\frac{1}{N}$ y continuar
 - Else, computar: $\beta_k = \frac{\varepsilon_k}{1-\varepsilon_k}$
 - Actualizar los pesos:

$$w_j^{k+1} = w_j^k \beta_k^{(1-\delta_j^k)} / \sum_{i=1}^N w_i^k \beta_k^{(1-\delta_i^k)}$$
- 3 Devolver \mathcal{D} y β_1, \dots, β_L

Boosting (Freund y Shapire, 1997)

AdaBoost

Fase de entrenamiento

- 1 Inicialización de parámetros: peso de cada uno de los N ejemplos: $w_j^1 = \frac{1}{N}$; número de iteraciones L ; conjunto de clasificadores $\mathcal{D} = \emptyset$
- 2 Para $k = 1 \dots L$
 - Extraer una muestra S_k del conjunto de datos usando la distribución w_j^k
 - Inducir el clasificador D_k usando S_k como conjunto de entrenamiento
 - Calcular el error en la iteración k : $\varepsilon_k = \sum_{j=1}^N w_j^k \delta_j^k$
 - If $\varepsilon_k = 0$ o $\varepsilon_k \geq 0,5$, ignorar D_k , reinicializar w_j^k a $\frac{1}{N}$ y continuar
 - Else, computar: $\beta_k = \frac{\varepsilon_k}{1-\varepsilon_k}$
 - Actualizar los pesos:

$$w_j^{k+1} = w_j^k \beta_k^{(1-\delta_j^k)} / \sum_{i=1}^N w_i^k \beta_k^{(1-\delta_i^k)}$$
- 3 Devolver \mathcal{D} y β_1, \dots, β_L

Boosting (Freund y Shapire, 1997)

AdaBoost

Fase de entrenamiento

- 1 Inicialización de parámetros: peso de cada uno de los N ejemplos: $w_j^1 = \frac{1}{N}$; número de iteraciones L ; conjunto de clasificadores $\mathcal{D} = \emptyset$
- 2 Para $k = 1 \dots L$
 - Extraer una muestra S_k del conjunto de datos usando la distribución w_j^k
 - Inducir el clasificador D_k usando S_k como conjunto de entrenamiento
 - Calcular el error en la iteración k : $\varepsilon_k = \sum_{j=1}^N w_j^k \delta_j^k$
 - If $\varepsilon_k = 0$ o $\varepsilon_k \geq 0,5$, ignorar D_k , reinicializar w_j^k a $\frac{1}{N}$ y continuar
 - Else, computar: $\beta_k = \frac{\varepsilon_k}{1-\varepsilon_k}$
 - Actualizar los pesos:

$$w_j^{k+1} = w_j^k \beta_k^{(1-\delta_j^k)} / \sum_{i=1}^N w_i^k \beta_k^{(1-\delta_i^k)}$$
- 3 Devolver \mathcal{D} y β_1, \dots, β_L

Boosting (Freund y Shapire, 1997)

AdaBoost

Fase de entrenamiento

- 1 Inicialización de parámetros: peso de cada uno de los N ejemplos: $w_j^1 = \frac{1}{N}$; número de iteraciones L ; conjunto de clasificadores $\mathcal{D} = \emptyset$
- 2 Para $k = 1 \dots L$
 - Extraer una muestra S_k del conjunto de datos usando la distribución w_j^k
 - Inducir el clasificador D_k usando S_k como conjunto de entrenamiento
 - Calcular el error en la iteración k : $\epsilon_k = \sum_{j=1}^N w_j^k \delta_j^k$
 - If $\epsilon_k = 0$ o $\epsilon_k \geq 0,5$, ignorar D_k , reinicializar w_j^k a $\frac{1}{N}$ y continuar
 - Else, computar: $\beta_k = \frac{\epsilon_k}{1-\epsilon_k}$
 - Actualizar los pesos:

$$w_j^{k+1} = w_j^k \beta_k^{(1-\delta_j^k)} / \sum_{i=1}^N w_i^k \beta_k^{(1-\delta_i^k)}$$
- 3 Devolver \mathcal{D} y β_1, \dots, β_L

Boosting (Freund y Shapire, 1997)

AdaBoost

Fase de entrenamiento

- 1 Inicialización de parámetros: peso de cada uno de los N ejemplos: $w_j^1 = \frac{1}{N}$; número de iteraciones L ; conjunto de clasificadores $\mathcal{D} = \emptyset$
- 2 Para $k = 1 \dots L$
 - Extraer una muestra S_k del conjunto de datos usando la distribución w_j^k
 - Inducir el clasificador D_k usando S_k como conjunto de entrenamiento
 - Calcular el error en la iteración k : $\varepsilon_k = \sum_{j=1}^N w_j^k \delta_j^k$
 - If $\varepsilon_k = 0$ o $\varepsilon_k \geq 0,5$, ignorar D_k , reinicializar w_j^k a $\frac{1}{N}$ y continuar
 - Else, computar: $\beta_k = \frac{\varepsilon_k}{1-\varepsilon_k}$
 - Actualizar los pesos:

$$w_j^{k+1} = w_j^k \beta_k^{(1-\delta_j^k)} / \sum_{i=1}^N w_i^k \beta_k^{(1-\delta_i^k)}$$
- 3 Devolver \mathcal{D} y β_1, \dots, β_L

Boosting (Freund y Shapire, 1997)

AdaBoost

Fase de entrenamiento

- 1 Inicialización de parámetros: peso de cada uno de los N ejemplos: $w_j^1 = \frac{1}{N}$; número de iteraciones L ; conjunto de clasificadores $\mathcal{D} = \emptyset$
- 2 Para $k = 1 \dots L$
 - Extraer una muestra S_k del conjunto de datos usando la distribución w_j^k
 - Inducir el clasificador D_k usando S_k como conjunto de entrenamiento
 - Calcular el error en la iteración k : $\varepsilon_k = \sum_{j=1}^N w_j^k \delta_j^k$
 - If $\varepsilon_k = 0$ o $\varepsilon_k \geq 0,5$, ignorar D_k , reinicializar w_j^k a $\frac{1}{N}$ y continuar
 - Else, computar: $\beta_k = \frac{\varepsilon_k}{1-\varepsilon_k}$
 - Actualizar los pesos:

$$w_j^{k+1} = w_j^k \beta_k^{(1-\delta_j^k)} / \sum_{i=1}^N w_i^k \beta_k^{(1-\delta_i^k)}$$
- 3 Devolver \mathcal{D} y β_1, \dots, β_L

Boosting (Freund y Shapire, 1997)

AdaBoost

Fase de entrenamiento

- 1 Inicialización de parámetros: peso de cada uno de los N ejemplos: $w_j^1 = \frac{1}{N}$; número de iteraciones L ; conjunto de clasificadores $\mathcal{D} = \emptyset$
- 2 Para $k = 1 \dots L$
 - Extraer una muestra S_k del conjunto de datos usando la distribución w_j^k
 - Inducir el clasificador D_k usando S_k como conjunto de entrenamiento
 - Calcular el error en la iteración k : $\varepsilon_k = \sum_{j=1}^N w_j^k \delta_j^k$
 - If $\varepsilon_k = 0$ o $\varepsilon_k \geq 0,5$, ignorar D_k , reinicializar w_j^k a $\frac{1}{N}$ y continuar
 - Else, computar: $\beta_k = \frac{\varepsilon_k}{1-\varepsilon_k}$
 - Actualizar los pesos:

$$w_j^{k+1} = w_j^k \beta_k^{(1-\delta_j^k)} / \sum_{i=1}^N w_i^k \beta_k^{(1-\delta_i^k)}$$
- 3 Devolver \mathcal{D} y β_1, \dots, β_L

Boosting (Freund y Shapire, 1997)

AdaBoost

Fase de entrenamiento

- 1 Inicialización de parámetros: peso de cada uno de los N ejemplos: $w_j^1 = \frac{1}{N}$; número de iteraciones L ; conjunto de clasificadores $\mathcal{D} = \emptyset$
- 2 Para $k = 1 \dots L$
 - Extraer una muestra S_k del conjunto de datos usando la distribución w_j^k
 - Inducir el clasificador D_k usando S_k como conjunto de entrenamiento
 - Calcular el error en la iteración k : $\varepsilon_k = \sum_{j=1}^N w_j^k \delta_j^k$
 - If $\varepsilon_k = 0$ o $\varepsilon_k \geq 0,5$, ignorar D_k , reinicializar w_j^k a $\frac{1}{N}$ y continuar
 - Else, computar: $\beta_k = \frac{\varepsilon_k}{1-\varepsilon_k}$
 - Actualizar los pesos:

$$w_j^{k+1} = w_j^k \beta_k^{(1-\delta_j^k)} / \sum_{i=1}^N w_i^k \beta_k^{(1-\delta_i^k)}$$
- 3 Devolver \mathcal{D} y β_1, \dots, β_L

Boosting (Freund y Shapire, 1997)

AdaBoost

Fase de entrenamiento

- 1 Inicialización de parámetros: peso de cada uno de los N ejemplos: $w_j^1 = \frac{1}{N}$; número de iteraciones L ; conjunto de clasificadores $\mathcal{D} = \emptyset$
- 2 Para $k = 1 \dots L$
 - Extraer una muestra S_k del conjunto de datos usando la distribución w_j^k
 - Inducir el clasificador D_k usando S_k como conjunto de entrenamiento
 - Calcular el error en la iteración k : $\varepsilon_k = \sum_{j=1}^N w_j^k \delta_j^k$
 - If $\varepsilon_k = 0$ o $\varepsilon_k \geq 0,5$, ignorar D_k , reinicializar w_j^k a $\frac{1}{N}$ y continuar
 - Else, computar: $\beta_k = \frac{\varepsilon_k}{1-\varepsilon_k}$
 - Actualizar los pesos:

$$w_j^{k+1} = w_j^k \beta_k^{(1-\delta_j^k)} / \sum_{i=1}^N w_i^k \beta_k^{(1-\delta_i^k)}$$

- 3 Devolver \mathcal{D} y β_1, \dots, β_L

Boosting (Freund y Shapire, 1997)

AdaBoost

Fase de entrenamiento

- 1 Inicialización de parámetros: peso de cada uno de los N ejemplos: $w_j^1 = \frac{1}{N}$; número de iteraciones L ; conjunto de clasificadores $\mathcal{D} = \emptyset$
- 2 Para $k = 1 \dots L$
 - Extraer una muestra S_k del conjunto de datos usando la distribución w_j^k
 - Inducir el clasificador D_k usando S_k como conjunto de entrenamiento
 - Calcular el error en la iteración k : $\varepsilon_k = \sum_{j=1}^N w_j^k \delta_j^k$
 - If $\varepsilon_k = 0$ o $\varepsilon_k \geq 0,5$, ignorar D_k , reinicializar w_j^k a $\frac{1}{N}$ y continuar
 - Else, computar: $\beta_k = \frac{\varepsilon_k}{1-\varepsilon_k}$
 - Actualizar los pesos:

$$w_j^{k+1} = w_j^k \beta_k^{(1-\delta_j^k)} / \sum_{i=1}^N w_i^k \beta_k^{(1-\delta_i^k)}$$
- 3 Devolver \mathcal{D} y β_1, \dots, β_L

Boosting (Freund y Shapire, 1997)

AdaBoost

Fase de clasificación

- Calcular el soporte que recibe cada clase c_r como

$$\sum_{D_k(\mathbf{x})=c_r} \ln\left(\frac{1}{\beta_k}\right)$$

- Clasificar el ejemplo \mathbf{x} como la clase que recibe el mayor soporte

Boosting (Freund y Shapire, 1997)

AdaBoost

Fase de clasificación

- Calcular el soporte que recibe cada clase c_r como

$$\sum_{D_k(\mathbf{x})=c_r} \ln\left(\frac{1}{\beta_k}\right)$$

- Clasificar el ejemplo \mathbf{x} como la clase que recibe el mayor soporte

Boosting (Freund y Shapire, 1997)

AdaBoost

Fase de clasificación

- Calcular el soporte que recibe cada clase c_r como

$$\sum_{D_k(\mathbf{x})=c_r} \ln\left(\frac{1}{\beta_k}\right)$$

- Clasificar el ejemplo \mathbf{x} como la clase que recibe el mayor soporte

Boosting (Freund y Shapire, 1997)

AdaBoost

Fase de clasificación

- Calcular el soporte que recibe cada clase c_r como
$$\sum_{D_k(\mathbf{x})=c_r} \ln\left(\frac{1}{\beta_k}\right)$$
- Clasificar el ejemplo \mathbf{x} como la clase que recibe el mayor soporte

Boosting (Freund y Shapire, 1997)

Boosting versus bagging

- Boosting combina múltiples modelos que se **complementan** unos a otros
- Al igual que bagging, usa **voto por la mayoría** para combinar las salidas de los modelos individuales
- Al igual que bagging, combina modelos del mismo **paradigma**
- En contraposición con bagging, donde cada modelo individual se induce de manera separada, en boosting cada **nuevo modelo** está influenciado por el comportamiento de los anteriores
- En boosting los nuevos modelos se convierten en expertos para **ejemplos mal clasificados** por los anteriores modelos
- En boosting la **contribución final de cada modelo** se basa en su **bondad**

Boosting (Freund y Shapire, 1997)

Boosting versus bagging

- Boosting combina múltiples modelos que se **complementan** unos a otros
- Al igual que bagging, usa **voto por la mayoría** para combinar las salidas de los modelos individuales
- Al igual que bagging, combina modelos del **mismo paradigma**
- En contraposición con bagging, donde cada modelo individual se induce de manera separada, en boosting cada **nuevo modelo está influenciado** por el comportamiento de los anteriores
- En boosting los nuevos modelos se convierten en expertos para **ejemplos mal clasificados** por los anteriores modelos
- En boosting la **contribución final de cada modelo** se basa en su bondad

Boosting (Freund y Shapire, 1997)

Boosting versus bagging

- Boosting combina múltiples modelos que se **complementan** unos a otros
- Al igual que bagging, usa **voto por la mayoría** para combinar las salidas de los modelos individuales
- Al igual que bagging, combina modelos del **mismo paradigma**
- En contraposición con bagging, donde cada modelo individual se induce de manera separada, en boosting cada **nuevo modelo está influenciado** por el comportamiento de los anteriores
- En boosting los nuevos modelos se convierten en expertos para **ejemplos mal clasificados** por los anteriores modelos
- En boosting la **contribución final de cada modelo** se basa en su bondad

Boosting (Freund y Shapire, 1997)

Boosting versus bagging

- Boosting combina múltiples modelos que se **complementan** unos a otros
- Al igual que bagging, usa **voto por la mayoría** para combinar las salidas de los modelos individuales
- Al igual que bagging, combina modelos del **mismo paradigma**
- En contraposición con bagging, donde cada modelo individual se induce de manera separada, en boosting cada **nuevo modelo está influenciado** por el comportamiento de los anteriores
- En boosting los nuevos modelos se convierten en expertos para **ejemplos mal clasificados** por los anteriores modelos
- En boosting la **contribución final de cada modelo** se basa en su bondad

Boosting (Freund y Shapire, 1997)

Boosting versus bagging

- Boosting combina múltiples modelos que se **complementan** unos a otros
- Al igual que bagging, usa **voto por la mayoría** para combinar las salidas de los modelos individuales
- Al igual que bagging, combina modelos del **mismo paradigma**
- En contraposición con bagging, donde cada modelo individual se induce de manera separada, en boosting cada **nuevo modelo está influenciado** por el comportamiento de los anteriores
- En boosting los nuevos modelos se convierten en expertos para **ejemplos mal clasificados** por los anteriores modelos
- En boosting la **contribución final de cada modelo** se basa en su bondad

Boosting (Freund y Shapire, 1997)

Boosting versus bagging

- Boosting combina múltiples modelos que se **complementan** unos a otros
- Al igual que bagging, usa **voto por la mayoría** para combinar las salidas de los modelos individuales
- Al igual que bagging, combina modelos del **mismo paradigma**
- En contraposición con bagging, donde cada modelo individual se induce de manera separada, en boosting cada **nuevo modelo está influenciado** por el comportamiento de los anteriores
- En boosting los nuevos modelos se convierten en expertos para **ejemplos mal clasificados** por los anteriores modelos
- En boosting la **contribución final de cada modelo** se basa en su bondad

Boosting (Freund y Shapire, 1997)

Boosting versus bagging

- Boosting combina múltiples modelos que se **complementan** unos a otros
- Al igual que bagging, usa **voto por la mayoría** para combinar las salidas de los modelos individuales
- Al igual que bagging, combina modelos del **mismo paradigma**
- En contraposición con bagging, donde cada modelo individual se induce de manera separada, en boosting cada **nuevo modelo está influenciado** por el comportamiento de los anteriores
- En boosting los nuevos modelos se convierten en expertos para **ejemplos mal clasificados** por los anteriores modelos
- En boosting la **contribución final de cada modelo** se basa en su bondad

Boosting (Freund y Shapire, 1997)

Boosting versus bagging

- Boosting combina múltiples modelos que se **complementan** unos a otros
- Al igual que bagging, usa **voto por la mayoría** para combinar las salidas de los modelos individuales
- Al igual que bagging, combina modelos del **mismo paradigma**
- En contraposición con bagging, donde cada modelo individual se induce de manera separada, en boosting cada **nuevo modelo está influenciado** por el comportamiento de los anteriores
- En boosting los nuevos modelos se convierten en expertos para **ejemplos mal clasificados** por los anteriores modelos
- En boosting la **contribución final de cada modelo** se basa en su bondad

Índice

- 1 Introducción
- 2 Hibridaciones
- 3 Combinando el mismo clasificador base
- 4 Combinando distintos clasificadores base**
- 5 Conclusiones

Métodos básicos

- Fusión de etiquetas
 - Plurality vote
 - Mayoría simple
 - Plurality vote con umbral
 - Voto por mayoría con peso
- Fusión de rankings
- Fusión de salidas continuas
- Stacked
- Cascada

Métodos básicos

- **Fusión de etiquetas**
 - Plurality vote
 - Mayoría simple
 - Plurality vote con umbral
 - Voto por mayoría con peso
- Fusión de rankings
- Fusión de salidas continuas
- Stacked
- Cascada

Métodos básicos

- **Fusión de etiquetas**
 - Plurality vote
 - Mayoría simple
 - Plurality vote con umbral
 - Voto por mayoría con peso
 - Fusión de rankings
 - Fusión de salidas continuas
 - Stacked
 - Cascada

Métodos básicos

- **Fusión de etiquetas**
 - Plurality vote
 - Mayoría simple
 - Plurality vote con umbral
 - Voto por mayoría con peso
- Fusión de rankings
- Fusión de salidas continuas
- Stacked
- Cascada

Métodos básicos

- **Fusión de etiquetas**
 - Plurality vote
 - Mayoría simple
 - Plurality vote con umbral
 - Voto por mayoría con peso
- Fusión de rankings
- Fusión de salidas continuas
- Stacked
- Cascada

Métodos básicos

- **Fusión de etiquetas**
 - Plurality vote
 - Mayoría simple
 - Plurality vote con umbral
 - Voto por mayoría con peso
- Fusión de rankings
- Fusión de salidas continuas
- Stacked
- Cascada

Métodos básicos

- **Fusión de etiquetas**
 - Plurality vote
 - Mayoría simple
 - Plurality vote con umbral
 - Voto por mayoría con peso
- **Fusión de rankings**
 - Fusión de salidas continuas
 - Stacked
 - Cascada

Métodos básicos

- **Fusión de etiquetas**
 - Plurality vote
 - Mayoría simple
 - Plurality vote con umbral
 - Voto por mayoría con peso
- **Fusión de rankings**
- **Fusión de salidas continuas**
- Stacked
- Cascada

Métodos básicos

- **Fusión de etiquetas**
 - Plurality vote
 - Mayoría simple
 - Plurality vote con umbral
 - Voto por mayoría con peso
- **Fusión de rankings**
- **Fusión de salidas continuas**
- **Stacked**
- Cascada

Métodos básicos

- **Fusión de etiquetas**
 - Plurality vote
 - Mayoría simple
 - Plurality vote con umbral
 - Voto por mayoría con peso
- **Fusión de rankings**
- **Fusión de salidas continuas**
- **Stacked**
- **Cascada**

Fusión de etiquetas

- L clasificadores: D_1, \dots, D_L
- Clase C con R posibles valores: c_1, \dots, c_R
- $d_{i,j}(\mathbf{x}) = \begin{cases} 1, & \text{si } D_i \text{ clasifica } \mathbf{x} \text{ en } c_j; \\ 0, & \text{en otro caso.} \end{cases}$
- $D_i(\mathbf{x}) = (d_{i,1}(\mathbf{x}), \dots, d_{i,R}(\mathbf{x}))$ con $j = 1, \dots, L$

Ejemplo: $R = 3; L = 5$

\mathbf{x}		0	1	0		1	0	0		1	0	0		0	1	0		0	1	0	
--------------	--	---	---	---	--	---	---	---	--	---	---	---	--	---	---	---	--	---	---	---	--

Fusión de etiquetas

- L clasificadores: D_1, \dots, D_L
- Clase C con R posibles valores: c_1, \dots, c_R
- $d_{i,j}(\mathbf{x}) = \begin{cases} 1, & \text{si } D_i \text{ clasifica } \mathbf{x} \text{ en } c_j; \\ 0, & \text{en otro caso.} \end{cases}$
- $D_i(\mathbf{x}) = (d_{i,1}(\mathbf{x}), \dots, d_{i,R}(\mathbf{x}))$ con $j = 1, \dots, L$

Ejemplo: $R = 3; L = 5$

\mathbf{x}	0	1	0	1	0	0	1	0	0	0	1	0	0	1	0
--------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Fusión de etiquetas

- L clasificadores: D_1, \dots, D_L
- Clase C con R posibles valores: c_1, \dots, c_R
- $d_{i,j}(\mathbf{x}) = \begin{cases} 1, & \text{si } D_i \text{ clasifica } \mathbf{x} \text{ en } c_j; \\ 0, & \text{en otro caso.} \end{cases}$
- $D_i(\mathbf{x}) = (d_{i,1}(\mathbf{x}), \dots, d_{i,R}(\mathbf{x}))$ con $j = 1, \dots, L$

Ejemplo: $R = 3; L = 5$

\mathbf{x}	0	1	0	1	0	0	1	0	0	0	1	0	0	1	0
--------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Fusión de etiquetas

- L clasificadores: D_1, \dots, D_L
- Clase C con R posibles valores: c_1, \dots, c_R
- $d_{i,j}(\mathbf{x}) = \begin{cases} 1, & \text{si } D_i \text{ clasifica } \mathbf{x} \text{ en } c_j; \\ 0, & \text{en otro caso.} \end{cases}$
- $D_i(\mathbf{x}) = (d_{i,1}(\mathbf{x}), \dots, d_{i,R}(\mathbf{x}))$ con $j = 1, \dots, L$

Ejemplo: $R = 3; L = 5$

\mathbf{x}	0	1	0	1	0	0	1	0	0	0	1	0	0	1	0
--------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Fusión de etiquetas

- L clasificadores: D_1, \dots, D_L
- Clase C con R posibles valores: c_1, \dots, c_R
- $d_{i,j}(\mathbf{x}) = \begin{cases} 1, & \text{si } D_i \text{ clasifica } \mathbf{x} \text{ en } c_j; \\ 0, & \text{en otro caso.} \end{cases}$
- $D_i(\mathbf{x}) = (d_{i,1}(\mathbf{x}), \dots, d_{i,R}(\mathbf{x}))$ con $j = 1, \dots, L$

Ejemplo: $R = 3; L = 5$

\mathbf{x}	0	1	0	1	0	0	1	0	0	0	1	0	0	1	0
--------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Fusión de etiquetas

- L clasificadores: D_1, \dots, D_L
- Clase C con R posibles valores: c_1, \dots, c_R
- $d_{i,j}(\mathbf{x}) = \begin{cases} 1, & \text{si } D_i \text{ clasifica } \mathbf{x} \text{ en } c_j; \\ 0, & \text{en otro caso.} \end{cases}$
- $D_i(\mathbf{x}) = (d_{i,1}(\mathbf{x}), \dots, d_{i,R}(\mathbf{x}))$ con $j = 1, \dots, L$

Ejemplo: $R = 3; L = 5$

$\mathbf{x} \mid 0 \ 1 \ 0 \mid 1 \ 0 \ 0 \mid 1 \ 0 \ 0 \mid 0 \ 1 \ 0 \mid 0 \ 1 \ 0 \mid$

Fusión de etiquetas

x	0	1	0	1	0	0	1	0	0	0	1	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

- **Plurality vote:** clase con mas votos

$$PV(D_1, \dots, D_L)(\mathbf{x}) = \arg_{j=1, \dots, R} \max \sum_{i=1}^L d_{i,j}(\mathbf{x})$$

$$\sum_{i=1}^L d_{i,1}(\mathbf{x}) = 2; \quad \sum_{i=1}^L d_{i,2}(\mathbf{x}) = 3; \quad \sum_{i=1}^L d_{i,3}(\mathbf{x}) = 0$$

- **Mayoría simple:** la clase debe de conseguir al menos la mitad de los votos mas uno
- **Plurality vote con umbral**

$$PV_U(D_1, \dots, D_L)(x) = \begin{cases} c_k, & \text{si } \sum_{i=1}^L d_{i,k} \geq \alpha L; \\ c_{R+1}, & \text{en otro caso.} \end{cases}$$

- **Mayoría simple:** $\alpha = \frac{1}{2} + \epsilon$
 - **Voto por unanimidad:** $\alpha = 1$
- **Voto por mayoría con peso**

$$VM_P(D_1, \dots, D_L; b_1, \dots, b_L)(\mathbf{x}) = \arg_{j=1, \dots, R} \max \sum_{i=1}^L b_i d_{i,j}(\mathbf{x})$$

con b_i proporcional a la bondad del i -ésimo clasificador: $b_i \propto p_i$

Fusión de etiquetas

\mathbf{x} | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

- **Plurality vote:** clase con mas votos

$$PV(D_1, \dots, D_L)(\mathbf{x}) = \arg_{j=1, \dots, R} \max \sum_{i=1}^L d_{i,j}(\mathbf{x})$$

$$\sum_{i=1}^L d_{i,1}(\mathbf{x}) = 2; \quad \sum_{i=1}^L d_{i,2}(\mathbf{x}) = 3; \quad \sum_{i=1}^L d_{i,3}(\mathbf{x}) = 0$$

- **Mayoría simple:** la clase debe de conseguir al menos la mitad de los votos mas uno
- **Plurality vote con umbral**

$$PV_U(D_1, \dots, D_L)(\mathbf{x}) = \begin{cases} c_k, & \text{si } \sum_{i=1}^L d_{i,k} \geq \alpha L; \\ c_{R+1}, & \text{en otro caso.} \end{cases}$$

- Mayoría simple: $\alpha = \frac{1}{2} + \epsilon$
 - Voto por unanimidad: $\alpha = 1$
- **Voto por mayoría con peso**

$$VM_P(D_1, \dots, D_L; b_1, \dots, b_L)(\mathbf{x}) = \arg_{j=1, \dots, R} \max \sum_{i=1}^L b_i d_{i,j}(\mathbf{x})$$

con b_i proporcional a la bondad del i -ésimo clasificador: $b_i \propto p_i$

Fusión de etiquetas

\mathbf{x} | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

- **Plurality vote**: clase con mas votos

$$PV(D_1, \dots, D_L)(\mathbf{x}) = \arg_{j=1, \dots, R} \max \sum_{i=1}^L d_{i,j}(\mathbf{x})$$

$$\sum_{i=1}^L d_{i,1}(\mathbf{x}) = 2; \quad \sum_{i=1}^L d_{i,2}(\mathbf{x}) = 3; \quad \sum_{i=1}^L d_{i,3}(\mathbf{x}) = 0$$

- **Mayoría simple**: la clase debe de conseguir al menos la mitad de los votos mas uno
- **Plurality vote con umbral**

$$PV_U(D_1, \dots, D_L)(\mathbf{x}) = \begin{cases} c_k, & \text{si } \sum_{i=1}^L d_{i,k} \geq \alpha L; \\ c_{R+1}, & \text{en otro caso.} \end{cases}$$

- Mayoría simple: $\alpha = \frac{1}{2} + \epsilon$
 - Voto por unanimidad: $\alpha = 1$
- Voto por mayoría con peso

$$VM_P(D_1, \dots, D_L; b_1, \dots, b_L)(\mathbf{x}) = \arg_{j=1, \dots, R} \max \sum_{i=1}^L b_i d_{i,j}(\mathbf{x})$$

con b_i proporcional a la bondad del i -ésimo clasificador: $b_i \propto p_i$

Fusión de etiquetas

\mathbf{x} | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

- **Plurality vote**: clase con mas votos

$$PV(D_1, \dots, D_L)(\mathbf{x}) = \arg_{j=1, \dots, R} \max \sum_{i=1}^L d_{i,j}(\mathbf{x})$$

$$\sum_{i=1}^L d_{i,1}(\mathbf{x}) = 2; \quad \sum_{i=1}^L d_{i,2}(\mathbf{x}) = 3; \quad \sum_{i=1}^L d_{i,3}(\mathbf{x}) = 0$$

- **Mayoría simple**: la clase debe de conseguir al menos la mitad de los votos mas uno
- **Plurality vote con umbral**

$$PV_U(D_1, \dots, D_L)(\mathbf{x}) = \begin{cases} c_k, & \text{si } \sum_{i=1}^L d_{i,k} \geq \alpha L; \\ c_{R+1}, & \text{en otro caso.} \end{cases}$$

- Mayoría simple: $\alpha = \frac{1}{2} + \epsilon$
- Voto por unanimidad: $\alpha = 1$
- **Voto por mayoría con peso**

$$VM_P(D_1, \dots, D_L; b_1, \dots, b_L)(\mathbf{x}) = \arg_{j=1, \dots, R} \max \sum_{i=1}^L b_i d_{i,j}(\mathbf{x})$$

con b_i proporcional a la bondad del i -ésimo clasificador: $b_i \propto p_i$

Fusión de etiquetas

\mathbf{x} | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

- **Plurality vote**: clase con mas votos

$$PV(D_1, \dots, D_L)(\mathbf{x}) = \arg_{j=1, \dots, R} \max \sum_{i=1}^L d_{i,j}(\mathbf{x})$$

$$\sum_{i=1}^L d_{i,1}(\mathbf{x}) = 2; \quad \sum_{i=1}^L d_{i,2}(\mathbf{x}) = 3; \quad \sum_{i=1}^L d_{i,3}(\mathbf{x}) = 0$$

- **Mayoría simple**: la clase debe de conseguir al menos la mitad de los votos mas uno
- **Plurality vote con umbral**

$$PV_U(D_1, \dots, D_L)(\mathbf{x}) = \begin{cases} c_k, & \text{si } \sum_{i=1}^L d_{i,k} \geq \alpha L; \\ c_{R+1}, & \text{en otro caso.} \end{cases}$$

- Mayoría simple: $\alpha = \frac{1}{2} + \epsilon$
 - Voto por unanimidad: $\alpha = 1$
- Voto por mayoría con peso

$$VM_P(D_1, \dots, D_L; b_1, \dots, b_L)(\mathbf{x}) = \arg_{j=1, \dots, R} \max \sum_{i=1}^L b_i d_{i,j}(\mathbf{x})$$

con b_i proporcional a la bondad del i -ésimo clasificador: $b_i \propto p_i$

Fusión de etiquetas

\mathbf{x} | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

- **Plurality vote**: clase con mas votos

$$PV(D_1, \dots, D_L)(\mathbf{x}) = \arg_{j=1, \dots, R} \max \sum_{i=1}^L d_{i,j}(\mathbf{x})$$

$$\sum_{i=1}^L d_{i,1}(\mathbf{x}) = 2; \quad \sum_{i=1}^L d_{i,2}(\mathbf{x}) = 3; \quad \sum_{i=1}^L d_{i,3}(\mathbf{x}) = 0$$

- **Mayoría simple**: la clase debe de conseguir al menos la mitad de los votos mas uno
- **Plurality vote con umbral**

$$PV_U(D_1, \dots, D_L)(\mathbf{x}) = \begin{cases} c_k, & \text{si } \sum_{i=1}^L d_{i,k} \geq \alpha L; \\ c_{R+1}, & \text{en otro caso.} \end{cases}$$

- Mayoría simple: $\alpha = \frac{1}{2} + \epsilon$
 - Voto por unanimidad: $\alpha = 1$
- **Voto por mayoría con peso**

$$VM_P(D_1, \dots, D_L; b_1, \dots, b_L)(\mathbf{x}) = \arg_{j=1, \dots, R} \max \sum_{i=1}^L b_i d_{i,j}(\mathbf{x})$$

con b_i proporcional a la bondad del i -ésimo clasificador: $b_i \propto p_i$

Fusión de rankings

1 2 3 4 | 2 1 4 3 | 2 4 1 3 | 1 3 2 4 | 1 4 2 3

- Para cada objeto se parte de L rankings (uno por clasificador)
 $D_i(\mathbf{x}) = \pi_i(1, \dots, R)$ con $i = 1, \dots, L$
- Se trata de encontrar el ranking consenso $\pi^*(1, \dots, R)$ de los L anteriores

Fusión de rankings

1 2 3 4 | 2 1 4 3 | 2 4 1 3 | 1 3 2 4 | 1 4 2 3

- Para cada objeto se parte de L rankings (uno por clasificador)

$$D_i(\mathbf{x}) = \pi_i(1, \dots, R) \text{ con } i = 1, \dots, L$$

- Se trata de encontrar el ranking consenso $\pi^*(1, \dots, R)$ de los L anteriores

Fusión de salidas continuas

0.1 0.5 0.4 | 0.0 0.0 1.0 | 0.4 0.3 0.4 | 0.2 0.7 0.1 | 0.1 0.8 0.1

- c_j , con **mayor soporte**: $\mu_j(\mathbf{x}) = F(d_{1,j}(\mathbf{x}), \dots, F(d_{1,j}(\mathbf{x})))$
- Ejemplos de función F :
 - **Media simple**: $\mu_j(\mathbf{x}) = \frac{1}{L} \sum_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = \frac{1}{5}(0,1 + \dots + 0,1); \dots; \mu_3(\mathbf{x}) = \frac{1}{5}(0,4 + \dots + 0,1)$
 - **Mínimo**: $\mu_j(\mathbf{x}) = \min_{i=1, \dots, L} \{d_{i,j}(\mathbf{x})\}$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0,8; \mu_3(\mathbf{x}) = 1$
 - **Máximo**
 - **Mediana**
 - **Trimed media (jurado de competición)**
 $\mu_1^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,1 + 0,2)$
 $\mu_2^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,3 + 0,5 + 0,7)$
 $\mu_3^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,4 + 0,4)$
 - **Producto**: $\mu_j(\mathbf{x}) = \prod_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0; \mu_3(\mathbf{x}) = 0,0032$
 - **Media generalizada**: $\mu_j^\alpha(\mathbf{x}) = \left(\frac{1}{L} \sum_{i=1}^L (d_{i,j}(\mathbf{x}))^\alpha\right)^{\frac{1}{\alpha}}$

Fusión de salidas continuas

0.1 0.5 0.4 | 0.0 0.0 1.0 | 0.4 0.3 0.4 | 0.2 0.7 0.1 | 0.1 0.8 0.1

- c_j , con **mayor soporte**: $\mu_j(\mathbf{x}) = F(d_{1,j}(\mathbf{x}), \dots, F(d_{1,j}(\mathbf{x})))$
- Ejemplos de función F :
 - **Media simple**: $\mu_j(\mathbf{x}) = \frac{1}{L} \sum_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = \frac{1}{5}(0,1 + \dots + 0,1); \dots; \mu_3(\mathbf{x}) = \frac{1}{5}(0,4 + \dots + 0,1)$
 - **Mínimo**: $\mu_j(\mathbf{x}) = \min_{i=1, \dots, L} \{d_{i,j}(\mathbf{x})\}$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0,8; \mu_3(\mathbf{x}) = 1$
 - **Máximo**
 - **Mediana**
 - **Trimed media** (jurado de competición)
 $\mu_1^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,1 + 0,2)$
 $\mu_2^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,3 + 0,5 + 0,7)$
 $\mu_3^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,4 + 0,4)$
 - **Producto**: $\mu_j(\mathbf{x}) = \prod_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0; \mu_3(\mathbf{x}) = 0,0032$
 - **Media generalizada**: $\mu_j^\alpha(\mathbf{x}) = (\frac{1}{L} \sum_{i=1}^L (d_{i,j}(\mathbf{x}))^\alpha)^\frac{1}{\alpha}$

Fusión de salidas continuas

0.1 0.5 0.4 | 0.0 0.0 1.0 | 0.4 0.3 0.4 | 0.2 0.7 0.1 | 0.1 0.8 0.1

- c_j , con **mayor soporte**: $\mu_j(\mathbf{x}) = F(d_{1,j}(\mathbf{x}), \dots, F(d_{1,j}(\mathbf{x})))$
- Ejemplos de función F :
 - **Media simple**: $\mu_j(\mathbf{x}) = \frac{1}{L} \sum_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = \frac{1}{5}(0,1 + \dots + 0,1); \dots; \mu_3(\mathbf{x}) = \frac{1}{5}(0,4 + \dots + 0,1)$
 - **Mínimo**: $\mu_j(\mathbf{x}) = \min_{i=1, \dots, L} \{d_{i,j}(\mathbf{x})\}$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0,8; \mu_3(\mathbf{x}) = 1$
 - **Máximo**
 - **Mediana**
 - **Trimed media** (jurado de competición)
 $\mu_1^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,1 + 0,2)$
 $\mu_2^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,3 + 0,5 + 0,7)$
 $\mu_3^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,4 + 0,4)$
 - **Producto**: $\mu_j(\mathbf{x}) = \prod_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0; \mu_3(\mathbf{x}) = 0,0032$
 - **Media generalizada**: $\mu_j^\alpha(\mathbf{x}) = \left(\frac{1}{L} \sum_{i=1}^L (d_{i,j}(\mathbf{x}))^\alpha\right)^{\frac{1}{\alpha}}$

Fusión de salidas continuas

0.1 0.5 0.4 | 0.0 0.0 1.0 | 0.4 0.3 0.4 | 0.2 0.7 0.1 | 0.1 0.8 0.1

- c_j , con **mayor soporte**: $\mu_j(\mathbf{x}) = F(d_{1,j}(\mathbf{x}), \dots, F(d_{1,j}(\mathbf{x})))$
- Ejemplos de función F :
 - **Media simple**: $\mu_j(\mathbf{x}) = \frac{1}{L} \sum_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = \frac{1}{5}(0,1 + \dots + 0,1); \dots; \mu_3(\mathbf{x}) = \frac{1}{5}(0,4 + \dots + 0,1)$
 - **Mínimo**: $\mu_j(\mathbf{x}) = \min_{i=1, \dots, L} \{d_{i,j}(\mathbf{x})\}$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0,8; \mu_3(\mathbf{x}) = 1$
 - **Máximo**
 - **Mediana**
 - **Trimed media** (jurado de competición)
 $\mu_1^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,1 + 0,2)$
 $\mu_2^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,3 + 0,5 + 0,7)$
 $\mu_3^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,4 + 0,4)$
 - **Producto**: $\mu_j(\mathbf{x}) = \prod_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0; \mu_3(\mathbf{x}) = 0,0032$
 - **Media generalizada**: $\mu_j^\alpha(\mathbf{x}) = (\frac{1}{L} \sum_{i=1}^L (d_{i,j}(\mathbf{x}))^\alpha)^\frac{1}{\alpha}$

Fusión de salidas continuas

0.1 0.5 0.4 | 0.0 0.0 1.0 | 0.4 0.3 0.4 | 0.2 0.7 0.1 | 0.1 0.8 0.1

- c_j , con **mayor soporte**: $\mu_j(\mathbf{x}) = F(d_{1,j}(\mathbf{x}), \dots, F(d_{1,j}(\mathbf{x})))$
- Ejemplos de función F :
 - **Media simple**: $\mu_j(\mathbf{x}) = \frac{1}{L} \sum_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = \frac{1}{5}(0,1 + \dots + 0,1); \dots; \mu_3(\mathbf{x}) = \frac{1}{5}(0,4 + \dots + 0,1)$
 - **Mínimo**: $\mu_j(\mathbf{x}) = \min_{i=1, \dots, L} \{d_{i,j}(\mathbf{x})\}$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0,8; \mu_3(\mathbf{x}) = 1$
 - **Máximo**
 - **Mediana**
 - **Trimed media** (jurado de competición)
 $\mu_1^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,1 + 0,2)$
 $\mu_2^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,3 + 0,5 + 0,7)$
 $\mu_3^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,4 + 0,4)$
 - **Producto**: $\mu_j(\mathbf{x}) = \prod_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0; \mu_3(\mathbf{x}) = 0,0032$
 - **Media generalizada**: $\mu_j^\alpha(\mathbf{x}) = (\frac{1}{L} \sum_{i=1}^L (d_{i,j}(\mathbf{x}))^\alpha)^\frac{1}{\alpha}$

Fusión de salidas continuas

0.1 0.5 0.4 | 0.0 0.0 1.0 | 0.4 0.3 0.4 | 0.2 0.7 0.1 | 0.1 0.8 0.1

- c_j , con **mayor soporte**: $\mu_j(\mathbf{x}) = F(d_{1,j}(\mathbf{x}), \dots, F(d_{1,j}(\mathbf{x})))$
- Ejemplos de función F :
 - **Media simple**: $\mu_j(\mathbf{x}) = \frac{1}{L} \sum_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = \frac{1}{5}(0,1 + \dots + 0,1); \dots ; \mu_3(\mathbf{x}) = \frac{1}{5}(0,4 + \dots + 0,1)$
 - **Mínimo**: $\mu_j(\mathbf{x}) = \min_{i=1, \dots, L} \{d_{i,j}(\mathbf{x})\}$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0,8 ; \mu_3(\mathbf{x}) = 1$
 - **Máximo**
 - **Mediana**
 - **Trimed media** (jurado de competición)
 $\mu_1^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,1 + 0,2)$
 $\mu_2^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,3 + 0,5 + 0,7)$
 $\mu_3^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,4 + 0,4)$
 - **Producto**: $\mu_j(\mathbf{x}) = \prod_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0 ; \mu_3(\mathbf{x}) = 0,0032$
 - **Media generalizada**: $\mu_j^\alpha(\mathbf{x}) = (\frac{1}{L} \sum_{i=1}^L (d_{i,j}(\mathbf{x}))^\alpha)^\frac{1}{\alpha}$

Fusión de salidas continuas

0.1 0.5 0.4 | 0.0 0.0 1.0 | 0.4 0.3 0.4 | 0.2 0.7 0.1 | 0.1 0.8 0.1

- c_j , con **mayor soporte**: $\mu_j(\mathbf{x}) = F(d_{1,j}(\mathbf{x}), \dots, F(d_{1,j}(\mathbf{x})))$
- Ejemplos de función F :
 - **Media simple**: $\mu_j(\mathbf{x}) = \frac{1}{L} \sum_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = \frac{1}{5}(0,1 + \dots + 0,1); \dots; \mu_3(\mathbf{x}) = \frac{1}{5}(0,4 + \dots + 0,1)$
 - **Mínimo**: $\mu_j(\mathbf{x}) = \min_{i=1, \dots, L} \{d_{i,j}(\mathbf{x})\}$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0,8; \mu_3(\mathbf{x}) = 1$
 - **Máximo**
 - **Mediana**
 - **Trimed media** (jurado de competición)
 $\mu_1^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,1 + 0,2)$
 $\mu_2^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,3 + 0,5 + 0,7)$
 $\mu_3^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,4 + 0,4)$
 - **Producto**: $\mu_j(\mathbf{x}) = \prod_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0; \mu_3(\mathbf{x}) = 0,0032$
 - **Media generalizada**: $\mu_j^\alpha(\mathbf{x}) = (\frac{1}{L} \sum_{i=1}^L (d_{i,j}(\mathbf{x}))^\alpha)^\frac{1}{\alpha}$

Fusión de salidas continuas

0.1 0.5 0.4 | 0.0 0.0 1.0 | 0.4 0.3 0.4 | 0.2 0.7 0.1 | 0.1 0.8 0.1

- c_j , con **mayor soporte**: $\mu_j(\mathbf{x}) = F(d_{1,j}(\mathbf{x}), \dots, F(d_{1,j}(\mathbf{x})))$
- Ejemplos de función F :
 - **Media simple**: $\mu_j(\mathbf{x}) = \frac{1}{L} \sum_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = \frac{1}{5}(0,1 + \dots + 0,1); \dots ; \mu_3(\mathbf{x}) = \frac{1}{5}(0,4 + \dots + 0,1)$
 - **Mínimo**: $\mu_j(\mathbf{x}) = \min_{i=1, \dots, L} \{d_{i,j}(\mathbf{x})\}$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0,8 ; \mu_3(\mathbf{x}) = 1$
 - **Máximo**
 - **Mediana**
 - **Trimed media** (jurado de competición)
 $\mu_1^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,1 + 0,2)$
 $\mu_2^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,3 + 0,5 + 0,7)$
 $\mu_3^{\text{Tri},0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,4 + 0,4)$
 - **Producto**: $\mu_j(\mathbf{x}) = \prod_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0 ; \mu_3(\mathbf{x}) = 0,0032$
 - **Media generalizada**: $\mu_j^\alpha(\mathbf{x}) = (\frac{1}{L} \sum_{i=1}^L (d_{i,j}(\mathbf{x}))^\alpha)^{\frac{1}{\alpha}}$

Fusión de salidas continuas

0.1 0.5 0.4 | 0.0 0.0 1.0 | 0.4 0.3 0.4 | 0.2 0.7 0.1 | 0.1 0.8 0.1

- c_j , con **mayor soporte**: $\mu_j(\mathbf{x}) = F(d_{1,j}(\mathbf{x}), \dots, F(d_{1,j}(\mathbf{x})))$
- Ejemplos de función F :
 - **Media simple**: $\mu_j(\mathbf{x}) = \frac{1}{L} \sum_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = \frac{1}{5}(0,1 + \dots + 0,1); \dots; \mu_3(\mathbf{x}) = \frac{1}{5}(0,4 + \dots + 0,1)$
 - **Mínimo**: $\mu_j(\mathbf{x}) = \min_{i=1, \dots, L} \{d_{i,j}(\mathbf{x})\}$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0,8; \mu_3(\mathbf{x}) = 1$
 - **Máximo**
 - **Mediana**
 - **Trimed media** (jurado de competición)
 $\mu_1^{Tri,0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,1 + 0,2)$
 $\mu_2^{Tri,0,20}(\mathbf{x}) = \frac{1}{3}(0,3 + 0,5 + 0,7)$
 $\mu_3^{Tri,0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,4 + 0,4)$
 - **Producto**: $\mu_j(\mathbf{x}) = \prod_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0; \mu_3(\mathbf{x}) = 0,0032$
 - **Media generalizada**: $\mu_j^\alpha(\mathbf{x}) = (\frac{1}{L} \sum_{i=1}^L (d_{i,j}(\mathbf{x}))^\alpha)^{\frac{1}{\alpha}}$

Fusión de salidas continuas

0.1 0.5 0.4 | 0.0 0.0 1.0 | 0.4 0.3 0.4 | 0.2 0.7 0.1 | 0.1 0.8 0.1

- c_j , con **mayor soporte**: $\mu_j(\mathbf{x}) = F(d_{1,j}(\mathbf{x}), \dots, F(d_{1,j}(\mathbf{x})))$
- Ejemplos de función F :
 - **Media simple**: $\mu_j(\mathbf{x}) = \frac{1}{L} \sum_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = \frac{1}{5}(0,1 + \dots + 0,1); \dots; \mu_3(\mathbf{x}) = \frac{1}{5}(0,4 + \dots + 0,1)$
 - **Mínimo**: $\mu_j(\mathbf{x}) = \min_{i=1, \dots, L} \{d_{i,j}(\mathbf{x})\}$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0,8; \mu_3(\mathbf{x}) = 1$
 - **Máximo**
 - **Mediana**
 - **Trimed media** (jurado de competición)
 $\mu_1^{Tri,0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,1 + 0,2)$
 $\mu_2^{Tri,0,20}(\mathbf{x}) = \frac{1}{3}(0,3 + 0,5 + 0,7)$
 $\mu_3^{Tri,0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,4 + 0,4)$
 - **Producto**: $\mu_j(\mathbf{x}) = \prod_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = 0; \mu_2(\mathbf{x}) = 0; \mu_3(\mathbf{x}) = 0,0032$
 - **Media generalizada**: $\mu_j^\alpha(\mathbf{x}) = (\frac{1}{L} \sum_{i=1}^L (d_{i,j}(\mathbf{x}))^\alpha)^\frac{1}{\alpha}$

Fusión de salidas continuas

0.1 0.5 0.4 | 0.0 0.0 1.0 | 0.4 0.3 0.4 | 0.2 0.7 0.1 | 0.1 0.8 0.1

- c_j , con **mayor soporte**: $\mu_j(\mathbf{x}) = F(d_{1,j}(\mathbf{x}), \dots, F(d_{1,j}(\mathbf{x})))$
- Ejemplos de función F :
 - **Media simple**: $\mu_j(\mathbf{x}) = \frac{1}{L} \sum_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = \frac{1}{5}(0,1 + \dots + 0,1)$; ... ; $\mu_3(\mathbf{x}) = \frac{1}{5}(0,4 + \dots + 0,1)$
 - **Mínimo**: $\mu_j(\mathbf{x}) = \min_{i=1, \dots, L} \{d_{i,j}(\mathbf{x})\}$
 $\mu_1(\mathbf{x}) = 0$; $\mu_2(\mathbf{x}) = 0,8$; $\mu_3(\mathbf{x}) = 1$
 - **Máximo**
 - **Mediana**
 - **Trimed media** (jurado de competición)
 $\mu_1^{Tri,0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,1 + 0,2)$
 $\mu_2^{Tri,0,20}(\mathbf{x}) = \frac{1}{3}(0,3 + 0,5 + 0,7)$
 $\mu_3^{Tri,0,20}(\mathbf{x}) = \frac{1}{3}(0,1 + 0,4 + 0,4)$
 - **Producto**: $\mu_j(\mathbf{x}) = \prod_{i=1}^L d_{i,j}(\mathbf{x})$
 $\mu_1(\mathbf{x}) = 0$; $\mu_2(\mathbf{x}) = 0$; $\mu_3(\mathbf{x}) = 0,0032$
 - **Media generalizada**: $\mu_j^\alpha(\mathbf{x}) = (\frac{1}{L} \sum_{i=1}^L (d_{i,j}(\mathbf{x}))^\alpha)^{\frac{1}{\alpha}}$

Stacked (Wolpert, 1992)

- Diversas capas (o niveles) de clasificadores
- Cada capa utiliza los resultados obtenidos por la capa anterior
- La última capa está formada por un único clasificador que es el que toma la decisión final

Stacked (Wolpert, 1992)

- Diversas capas (o niveles) de clasificadores
- Cada capa utiliza los resultados obtenidos por la capa anterior
- La última capa está formada por un único clasificador que es el que toma la decisión final

Stacked (Wolpert, 1992)

- Diversas capas (o niveles) de clasificadores
- Cada capa utiliza los resultados obtenidos por la capa anterior
- La última capa está formada por un único clasificador que es el que toma la decisión final

Stacked (Wolpert, 1992)

- Diversas capas (o niveles) de clasificadores
- Cada capa utiliza los resultados obtenidos por la capa anterior
- La última capa está formada por un único clasificador que es el que toma la decisión final

Combinación en cascada

- Secuencia de clasificadores ordenados $(D_1, D_2, \dots, D_j, D_{j+1}, \dots)$ en función ascendente del espacio, tiempo, coste de representación, complejidad,
- Se usa el clasificador D_j para el caso x si los clasificadores previos D_1, \dots, D_{j-1} no han proporcionado la suficiente confianza al clasificar x

Combinación en cascada

- Secuencia de clasificadores ordenados $(D_1, D_2, \dots, D_j, D_{j+1}, \dots)$ en función ascendente del espacio, tiempo, coste de representación, complejidad,
- Se usa el clasificador D_j para el caso x si los clasificadores previos D_1, \dots, D_{j-1} no han proporcionado la suficiente confianza al clasificar x

Combinación en cascada

- Secuencia de clasificadores ordenados $(D_1, D_2, \dots, D_j, D_{j+1}, \dots)$ en función ascendente del espacio, tiempo, coste de representación, complejidad,
- Se usa el clasificador D_j para el caso \mathbf{x} si los clasificadores previos D_1, \dots, D_{j-1} no han proporcionado la suficiente confianza al clasificar \mathbf{x}

Índice

- 1 Introducción
- 2 Hibridaciones
- 3 Combinando el mismo clasificador base
- 4 Combinando distintos clasificadores base
- 5 Conclusiones**

Conclusiones

- Combinando clasificadores tratar de encontrar superficies de decisión adecuadas al problema
 - *Híbridaciones*
 - Combinación de decisiones obtenidas con el mismo clasificador de base
 - Combinación de decisiones obtenidas con distintos clasificadores de base
- Campo activo de investigación empírica

Conclusiones

- Combinando clasificadores tratar de encontrar superficies de decisión adecuadas al problema
 - *Híbridaciones*
 - Combinación de decisiones obtenidas con el mismo clasificador de base
 - Combinación de decisiones obtenidas con distintos clasificadores de base
- Campo activo de investigación empírica

Conclusiones

- Combinando clasificadores tratar de encontrar superficies de decisión adecuadas al problema
 - *Híbridaciones*
 - Combinación de decisiones obtenidas con el mismo clasificador de base
 - Combinación de decisiones obtenidas con distintos clasificadores de base
 - Campo activo de investigación empírica

Conclusiones

- Combinando clasificadores tratar de encontrar superficies de decisión adecuadas al problema
 - *Híbridaciones*
 - Combinación de decisiones obtenidas con el **mismo clasificador de base**
 - Combinación de decisiones obtenidas con **distintos clasificadores de base**
- Campo activo de investigación empírica

Conclusiones

- Combinando clasificadores tratar de encontrar superficies de decisión adecuadas al problema
 - *Híbridaciones*
 - Combinación de decisiones obtenidas con el **mismo clasificador de base**
 - Combinación de decisiones obtenidas con **distintos clasificadores de base**
- Campo activo de investigación empírica

Conclusiones

- Combinando clasificadores tratar de encontrar superficies de decisión adecuadas al problema
 - *Híbridaciones*
 - Combinación de decisiones obtenidas con el **mismo clasificador de base**
 - Combinación de decisiones obtenidas con **distintos clasificadores de base**
- Campo activo de investigación empírica

Combinando clasificadores en Weka

Lazy Bayesian Rules	Lazy	LBR
Naive Bayes Tree	Trees	NBTree
Logistic Model Trees	Trees	LMT
Bagging	Meta	Bagging
Random Forest	Trees	RandomForest + RandomTree
Fusion	Meta	Vote
Stacked	Meta	Stacking + StackingC

Tema 13: Combinación de Clasificadores

Pedro Larrañaga, Iñaki Inza, Abdelmalik Moujahid

Intelligent Systems Group
Departamento de Ciencias de la Computación e Inteligencia Artificial
Universidad del País Vasco