RNPS No. 2142
ISSN 2072-6287
Versión Digital

SERIE AZUL

CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

REPORTE TÉCNICO
Reconocimiento de Patrones

Combinación de clasificadores supervisados: estado del arte

Miguel A. Duval-Poo, Sandro Vega-Pons
y José Ruiz-Shulcloper

RT_048 **abril 2012**

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

RNPS No. 2142
ISSN 2072-6287
Versión Digital

SERIE AZUL

CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

REPORTE TÉCNICO
Reconocimiento de Patrones

Combinación de clasificadores supervisados: estado del arte

Miguel A. Duval-Poo, Sandro Vega-Pons y
Jose Ruiz-Shulcloper

RT_048 **abril 2012**

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

Tabla de contenido

Combinación de clasificadores supervisados: estado del arte

Miguel A. Duval-Poo[1], Sandro Vega-Pons[2] y José Ruiz-Shulcloper[1]

[1] Dpto. Reconocimiento de Patrones, Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV), La Habana, Cuba
{mduval, jshulcloper}@cenatav.co.cu
[2] Dpto. Minería de Datos, Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV), La Habana, Cuba
svega@cenatav.co.cu

RT_048, Serie Azul, CENATAV
Aceptado: 16 de febrero de 2012

Resumen. Numerosas son las áreas del reconocimiento de patrones en las que son necesario el empleo de clasificadores confiables y muy precisos. Tradicionalmente estos problemas han sido resueltos con el empleo de un solo clasificador, sin embargo, en la actualidad existe un auge en la conformación de sistemas compuestos por varios clasificadores con el objetivo de mejorar los resultados de clasificación. Por tanto, resulta necesario el desarrollo de un trabajo donde se realice un estudio de los diversos métodos existentes en esta área y que además sirva de punto de partida para especialistas e investigadores que deseen adentrarse en el tema. A pesar de existir varios trabajos e incluso libros que abordan la combinación de clasificadores supervisados con estos mismos objetivos, el constante desarrollo de nuevos métodos y sistemas de combinación junto con la creciente bibliografía en este campo, hacen necesario realizar estados del arte periódicos con el objetivo de mantener actualizada la comunidad científica. En este trabajo se presenta un estado del arte actualizado de la combinación de clasificadores supervisados donde se realiza un análisis crítico a los diversos métodos de combinación existentes. También son analizados métodos para la selección y generación de ensamblados de clasificadores junto con el concepto de diversidad entre clasificadores.

Palabras clave: sistemas multiclasificadores, combinación de clasificadores, selección de clasificadores, generación de ensamblados, diversidad.

Abstract. There are several areas in pattern recognition where are necessary to use reliable and very accurate classifiers. Traditionally these problems have been solved with the use of a single classifier; however, there is currently a boom in the creation of systems composed of several classifiers in order to improve the classification results. Therefore is necessary to develop a paper where a study of the various methods in this area is performed and also serve as a starting point for experts and researchers who wish to venture into the topic. Although several papers and even books exists that address the combination of supervised classifiers for the same purposes, the continuing development of new methods and systems combined with the growing literature in this field, periodic state of the art are required in order to keep updated the scientific community. This paper presents an updated state of the art for combining supervised classifiers, where a critical analysis is performed for the various existing combination methods. We also discussed methods for the selection and ensemble generation of classifiers along with the concept of diversity between classifiers.

Keywords: multiple classifiers systems, classifier combination, classifier selection, ensemble generation, diversity.

1 Introducción

El objetivo fundamental de la clasificación es la asignación de etiquetas a objetos los cuales están descritos a través de un conjunto de valores denominados atributos o rasgos. La clasificación puede ser: supervisada, parcialmente supervisada o no supervisada. En la no supervisada, el problema consiste en agrupar una colección de objetos en grupos donde los elementos de un mismo grupo sean similares y los elementos entre diferentes grupos no son tan parecidos de acuerdo a algún criterio. En la parcialmente supervisada el objetivo central es clasificar nuevos objetos en el que no se tienen muestras de todas las clases y en las que incluso pudieran existir clases que se desconocen. Por otro lado, en la supervisada, cada objeto de una muestra del conjunto de datos viene asociado con la etiqueta de la clase a la cual pertenece. El objetivo radica en descubrir las relaciones existentes entre los rasgos y las clases. La relación o relaciones encontradas son representadas a través de un clasificador. Donde lo usual es que estos clasificadores sean usados para predecir la etiqueta de la clase a la cual pertenece un nuevo objeto.

El proceso de clasificación supervisada se puede dividir en tres fases fundamentales: el entrenamiento o aprendizaje, la evaluación o validación y la clasificación. Un conjunto de entrenamiento se define como un grupo de objetos representados a través de un conjunto de rasgos donde uno de los rasgos es denominado atributo objetivo o etiqueta, la cual representa la clase a la que pertenece cada objeto. El proceso de entrenamiento es donde un clasificador debe aprender cómo clasificar los objetos generalizando a partir de los datos de entrenamiento a las situaciones no vistas. En el proceso de evaluación es donde usando un conjunto de validación comprueba la efectividad del clasificador recién entrenado, y así se garantiza que el clasificador no se haya sobre entrenado o simplemente no se haya entrenado correctamente. Finalmente, en el proceso de clasificación una vez entrenado y validado el clasificador, este está listo para predecir la etiqueta de cualquier nuevo objeto que se le presente. La salida de un clasificador supervisado puede ser: la etiqueta de la clase del nuevo objeto clasificado, un conjunto de etiquetas ordenadas por la probabilidad de ser la etiqueta correcta, así como un vector numérico donde cada valor representa el valor de pertenencia otorgado por el clasificador a cada clase. En la actualidad existen diversos clasificadores supervisados descritos en la literatura. Entre los más usados se encuentran el Vecino más Cercano, las Redes Neuronales, el clasificador Bayesiano, los Árboles de Decisión, la Máquina de Soporte de Vectores, entre muchísimos otros, cada uno con sus fortalezas y debilidades. La efectividad de un clasificador supervisado depende en gran medida de las características de los datos que deben clasificarse, no existe un clasificador único que sea el mejor en todos los problemas dados.

Sin embargo, se ha llegado al punto donde la precisión o eficacia que ofrece un solo clasificador no siempre satisface los requerimientos para un determinado problema. Por tal razón, se han utilizado en conjunto múltiples clasificadores para tratar de alcanzar resultados superiores a los de un clasificador individual.

En el campo del reconocimiento de patrones se investigan en la actualidad los sistemas multiclasificadores con el fin de desarrollar sistemas de clasificación altamente precisos.Estos sistemas, también conocido como combinación de clasificadores, ensamblado de clasificadores, fusión de clasificadores, comité de expertos, etc. han tomado un gran auge debido a la necesidad de desarrollar clasificadores altamente precisos y confiables para muchas aplicaciones prácticas y principalmente debido al éxito que han tenido en las mismas. Los sistemas multiclasificadores han sido empleados para mejorar los resultados de clasificación en numerosos campos como la biometría [1, 2], las finanzas [3], medicina [4], quimio-informática [5], manufacturas [6], geografía [7], seguridad de la información [8], recuperación de información [9], recuperación de imágenes [10] y muchos otros más.

La idea de combinar clasificadores parte del simple hecho de que si tenemos un conjunto de clasificadores donde sus respuestas poseen un cierto grado de independencia entre sí, estas respuestas pueden complementarse unas a otras. Es decir, una muestra mal clasificada por un clasificador puede ser correctamente clasificada por otro clasificador y viceversa. Por tal razón resulta sensato combinar

los resultados de esos clasificadores de alguna manera con el objetivo de incrementar la precisión y confiabilidad con relación a un clasificador por separado.

Debido a su gran utilidad en diversas aplicaciones prácticas de la actualidad en el campo del reconocimiento de patrones, es nuestro objetivo realizar un estudio profundo y un análisis crítico de los diversos sistemas multiclasificadores existentes en el estado del arte.

El presente trabajo está dividido en ocho secciones. En la segunda se realiza una breve introducción a la clasificación supervisada. La tercera nos adentra en el área de los sistemas multiclasificadores comenzando con la definición formal de los mismos, seguido de una taxonomía de estos y de las distintas arquitecturas empleadas. En la Sección 4, son expuestos una serie de métodos de combinación de resultados de clasificadores. Conformada por tres subsecciones, en cada subsección se abordan los métodos de combinación de resultados concernientes a cada tipo de salida de los clasificadores supervisados: etiqueta de clase, ranking de etiquetas y valores de soporte de cada clase. En la quinta sección van a ser abordados los métodos que en vez de realizar la combinación de resultados con todos los clasificadores disponibles, la realizan con un subgrupo de estos, seleccionados en dependencia de ciertas propiedades deseadas. Estos métodos son conocidos como métodos de selección de clasificadores. La sexta sección aborda los métodos que generan el ensamblado de clasificadores. Es decir, fijada una manera de realizar la combinación de resultados, estos métodos construyen un grupo de clasificadores lo más diferentes entre sí en cuanto a la toma de decisión. A continuación, en la Sección 7 se abordan la diversidad entre clasificadores, así como las medidas de diversidad más populares. Finalmente en la Sección 7, se exponen las conclusiones del trabajo.

2 Clasificación supervisada

Sea $\Omega = \{\omega_1, \dots, \omega_c\}$ un conjunto de etiquetas que representan las c clases que conforman un problema de aprendizaje supervisado y $\mathbf{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_N\}$, $\mathbf{z}_j \in \mathbb{F}^n$ el conjunto de datos de entrenamiento donde cada $\mathbf{z}_j$ es una tupla de un cierto espacio n-dimensional. La etiqueta de clase de $\mathbf{z}_j$ es denotada por $l(\mathbf{z}_j) \in \Omega$. Un clasificador es una función $D: \mathbb{F}^n \to \mathcal{W}$ la cual a cada objeto $\mathbf{x} \in \mathbb{F}^n$ le asigna un cierto valor en el espacio de resultados $\mathcal{W}$. De acuerdo al tipo de resultado, los clasificadores pueden categorizarse en tres tipos [11]:

- **Tipo 1** (*nivel abstracto*). Para este tipo de salida, un clasificador D retorna la etiqueta de clase $s \in \Omega$ asignada a un objeto $\mathbf{x}$. En este formato de salida no existe ninguna información acerca de la certeza de la etiqueta predicha, así como ninguna etiqueta alternativa es sugerida. Formalmente un clasificador con este tipo de salida quedaría definido como una función $D^a: \mathbb{F}^n \to \Omega$.
- **Tipo 2** (*nivel de rango*). Este tipo de clasificador tiene como salida un subconjunto de Ω, donde las etiquetas están ordenadas de acuerdo a su posibilidad de ser la etiqueta de la clase correcta. Este tipo de salida es especialmente útil en problemas que poseen un elevado número de clases. Un clasificador con este formato de salida se define como una función $D^r: \mathbb{F}^n \to 2^\Omega$, donde 2^Ω es el conjunto potencia de Ω.
- **Tipo 3** (*nivel de medida*). Para este formato de salida un clasificador D retorna un vector c-dimesional $[d_1, \dots, d_c]^T$, donde el valor d_j representa el soporte a la hipótesis de que un objeto $\mathbf{x}$ pertenezca a la clase ω_j. Este es el formato de salida que más información brinda al clasificar un objeto. Un clasificador con tal salida quedaría definido por una función $D^m: \mathbb{F}^n \to \mathbb{R}^c$. Donde $\mathbb{R}$ es el conjunto de los números reales.

El tipo de espacio de representación de los objetos va a estar determinado por el tipo de rasgo que se emplee en la descripción de los objetos, el conjunto de sus valores, los criterios de comparación de dichos valores y propiedades de los mismos en ese espacio. En muchos casos, todos estos rasgos toman

valores en el conjunto de los números reales, quedando la representación de cada objeto, como un vector de $\mathbb{R}^n$, lo que puede permitir la utilización de una gran variedad de herramientas matemáticas definidas sobre los espacios vectoriales. En otros casos, todas las variables son nominales, pudiendo tomar valores en un conjunto de posibles valores. Sin embargo, en ocasiones existen problemas a resolver cuyas modelaciones obliga a usar variables numéricas mezcladas con variables no numéricas (nominales, ordinales), e incluso se hace necesario el empleo de un símbolo especial para denotar la ausencia de información. En este caso la representación de los objetos es solamente una tupla de tamaño n donde ninguna estructura algebraica, topológica o lógica es asumida sobre los datos; es decir, sobre el espacio de representación de los objetos solamente se asume que es un producto cartesiano de dimensión n donde es conocido el dominio de definición de cada rasgo.

La efectividad de un clasificador con respecto a un conjunto de datos es usualmente comprobada usando un conjunto de medias de evaluación (también denominadas funcionales de calidad). Dado un conjunto de validación el cual contenga N muestras. Denotemos por TP (True Positives) al número de muestras positivas correctamente clasificadas y por FP (False Positives) al número de muestras negativas clasificadas como positivas. Sea además denotado TN (True Negative) y FN (False Negative) al conjunto de muestras negativas correctamente e incorrectamente clasificadas respectivamente. A continuación veremos una tabla con algunas de las medidas de calidad mayormente utilizadas.

Tabla 1. Medidas de evaluación de clasificadores.

Nombre de la Medida	*Definición*	*Descripción*
Precisión	$\frac{TP+TN}{N}$	Mide el porciento de muestras correctamente clasificadas.
Especificidad	$\frac{TN}{TN+FP}$	Mide la proporción de muestras negativas correctamente clasificadas.
Sensibilidad (recobrado)	$\frac{TP}{TP+FN}$	Mide la proporción de muestras positivas correctamente clasificadas.
Valor de predicción positiva	$\frac{TP}{TP+FP}$	Da una medida del porciento de los verdaderos positivos del total de muestras clasificadas como positivas. Mide la probabilidad de que una muestra clasificada como positiva sea verdaderamente positiva.
Valor de predicción negativa	$\frac{TN}{FN+TN}$	Da una medida del porciento de los verdaderos negativos del total de muestras clasificadas como negativas. Mide la probabilidad de que una muestra clasificada como negativa sea verdaderamente negativa.

Otro aspecto importante a la hora de realizar el entrenamiento o la evaluación de un o un grupo de clasificadores es disponer de la mayor cantidad de objetos posibles. Sin embargo, utilizar todos los objetos de un conjunto de datos para realizar el entrenamiento y usar los mismos para evaluarlo, puede sobre entrenar el clasificador de forma tal que aprenda perfectamente de esos datos y falle con datos no vistos. De ahí la importancia de tener un conjunto de datos separado para realizar la evaluación. Dado que usualmente se dispone de un solo conjunto de datos **Z**, existen varias alternativas para hacer un mejor uso del conjunto. A continuación mencionaremos algunas:

- *Hold-out*. Particiona el conjunto de datos a la mitad, o en alguna proporción, utilizando una parte para entrenar y la otra para evaluar. También resulta válido intercambiar las particiones y promediar el resultado de la evaluación.

- *Data Shuffle*. Parecido al anterior, se realizan K cortes aleatorios del conjunto para su entrenamiento y evaluación. Finalmente las K evaluaciones son promediadas.
- *Cross-validation*. El conjunto de datos **Z** es aleatoriamente dividido en K subconjuntos de tamaño N/K. Un subconjunto es empleado para evaluar un clasificador D, el cual fue entrenado con los $K-1$ restantes. Este procedimiento es repetido K veces tomando un subconjunto diferente en cada ocasión para evaluar. Finalmente los K valores resultantes de cada evaluación son promediados y devueltos como resultado final.

3 Sistemas multiclasificadores

Sea $\mathcal{D} = \{D_1, \ldots, D_L\}$ un conjunto (ensamblado, comité) de L clasificadores, agrupados con el objetivo de mejorar el resultado individual de cada clasificador D_i. Este conjunto, también conocido como sistema multiclasificador, puede ser categorizado de acuerdo a diferentes criterios. Teniendo en cuenta la manera en que se construyen los ensamblados, Kuncheva [12] propuso los cuatro enfoques que se muestran a continuación (ver figura 1).

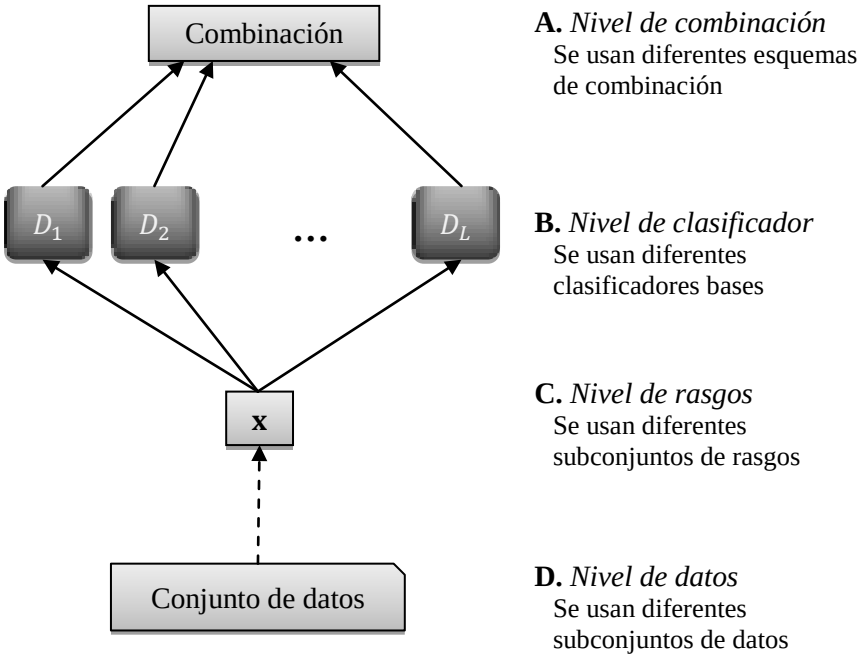


Fig. 1. Enfoques para la construcción de ensamblados de clasificadores[1].

- *Nivel de combinación.* Dado un conjunto de L clasificadores ya entrenados, el problema consiste en escoger y entrenar, en caso de ser necesario, un esquema de combinación para tomar una decisión final sobre las L respuestas. Dentro de los posibles esquemas, los más usados son el voto mayoritario, combinación Bayesiana, integrales difusas, etc.
- *Nivel de clasificador.* En este nivel el problema consiste en seleccionar los L clasificadores bases que van a conformar el ensamblado. Cualquier método de clasificación supervisada puede ser empleado. Los ensamblados conformados pueden ser homogéneos o heterogéneos. Los homogéneos utilizan un mismo modelo de clasificación lo que con

[1] Figura tomada del libro "*Combining Pattern Classifiers: Methods and Algorithms*" [12]

diferentes estructuras, parámetros de inicialización, etc. Mientras que los heterogéneos están conformados por diferentes modelos de clasificación.

- *Nivel de rasgo.* Dado un problema de clasificación conformado por r rasgos, los clasificadores que van a integrar el ensamblado son entrenados usando un subconjunto diferente de rasgos para cada clasificador. Ejemplo de esta manera de construir un ensamblado es el método del sub-espacio aleatorio. Este enfoque resulta útil cuando el número de rasgos es alto y cuando los rasgos provienen de distintas fuentes.
- *Nivel de datos.* En este enfoque los *L* clasificadores que van a conformar el ensamblado son entrenados usando diferentes subconjuntos del conjunto original de datos. Los subconjuntos pueden ser formados lo mismo particionando el conjunto original, que tomando muestras con reemplazo, entre otras maneras. Los dos métodos más conocidos en este enfoque son el Bagging y el Boosting.

Sin embargo, a través de estos enfoques no se puede representar todas las maneras de conformar un sistema multiclasificador, como es el caso de un sistema basado en selección de clasificadores. Por otra parte, existen otros criterios para agrupar los sistemas multiclasificadores, por ejemplo, de acuerdo a si hacen fusión o selección de clasificadores, o si generan o no el ensamblado de clasificadores [12].

En este trabajo se propone la siguiente caracterización de los sistemas multiclasificadores, donde se unifica en una sola taxonomía los criterios anteriores:

- *Métodos basados en generación de ensamblados.* Estos sistemas fijan un esquema de combinación como por ejemplo el voto mayoritario y se encargan de generar los clasificadores que van a conformar el ensamblado. La tendencia general consiste en conformar clasificadores independientes en cuanto a sus respuestas. Ejemplos de estos métodos son el Bagging, Boosting, Bosques Aleatorios, etc. Es equivalente con los enfoques B, C y D propuestos por Kuncheva. Estos métodos van a ser abordados en la Sección 5.
- *Métodos basados en selección de clasificadores.* Sea un conjunto de *L* clasificadores ya entrenados. Estos métodos tratan de seleccionar cuál de los *L* clasificadores es el más apropiado para asignar una clase a un objeto. Estos métodos se pueden ver en detalle en la Sección 4.
- *Métodos basados en combinación de clasificadores.* Dado un conjunto de *L* clasificadores, estos métodos combinan o fusionan los *L* resultados de sus miembros para retornar una respuesta final. Es equivalente al nivel A de los enfoques propuestos por Kuncheva. La Sección 3 va a abordar estos métodos.
- *Métodos híbridos.* Aquí se agrupan los métodos que combinan varias o todas las estrategias descritas anteriormente.

La figura 2 muestra la taxonomía usada.

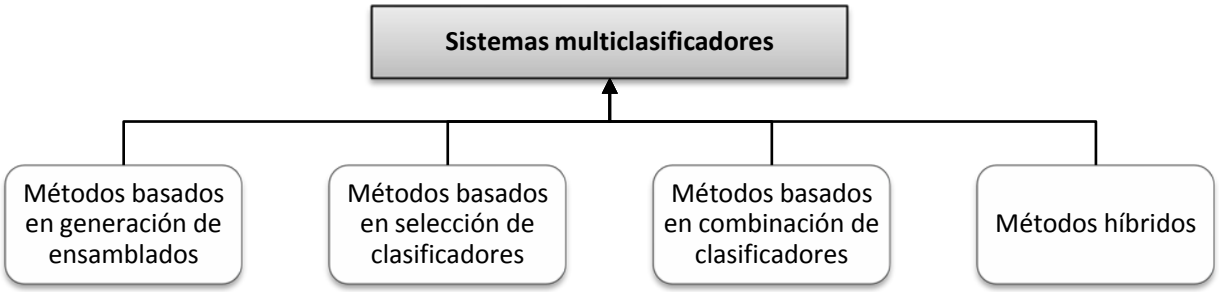


Fig. 2. Taxonomía de los sistemas multiclasificadores.

Otra manera de agrupar los sistemas multiclasificadores es en *entrenables* y no *entrenables* [12]. Los métodos entrenables son aquellos que requieren de un entrenamiento posterior una vez que se hayan entrenados individualmente los clasificadores que conforman el ensamblado. Esto se requiere usualmente para ajustar parámetros, estimar valores, etc. Estos tienen como ventaja que permiten ajustarse a un determinado problema en particular que se quiere resolver y así obtener mejores resultados en el mismo. Su desventaja radica en que en ocasiones se puede producir sobre entrenamiento y por tanto una pobre generalización del problema. Los no entrenables son por el contrario aquellos que combinan los clasificadores tal y como vienen, sin necesidad de estimar valores ni ajustar parámetros mediante un conjunto de entrenamiento. Son más generales y pueden emplearse en cualquier problema que se quiera resolver.

Otro aspecto importante en los sistemas multiclasificadores es la arquitectura o topología de los mismos. La arquitectura no es más que la forma en que se desea integrar a un conjunto de L clasificadores para garantizar una toma de decisión. Lu [13] las categorizó en tres grupos: en cascada (vertical), paralela (horizontal) e híbrida (jerárquica).

La arquitectura en cascada consiste de una secuencia de clasificadores donde el resultado de un clasificador influye en la entrada del próximo clasificador de la secuencia. Este esquema queda representado en la figura 3. Existen dos enfoques para esta arquitectura: la re-evaluación y la reducción del conjunto de clases. En la re-evaluación, la idea consiste en validar el resultado de un clasificador que otorgue un bajo valor de soporte a su decisión usando el próximo clasificador de la secuencia. Así sucesivamente hasta que uno de los clasificadores devuelva una decisión con un alto grado de soporte. La idea en la reducción de clases es que el número de posibles clases de un objeto de entrada, se reduzca a medida que sea analizado por los distintos clasificadores de la secuencia. Finalmente el último clasificador de la secuencia es el que va devolver la etiqueta de la clase a la cual pertenece el objeto.

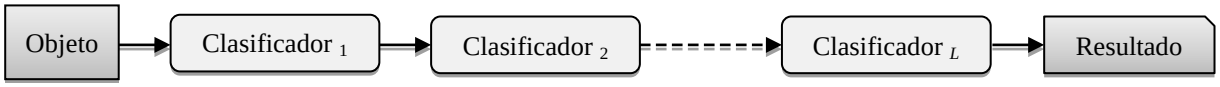


Fig. 3. Arquitectura en cascada (vertical).

En el caso de la arquitectura paralela, los resultados de un conjunto de *L* clasificadores son obtenidos de forma independiente. Estos resultados son luego fusionados usando un esquema de combinación el cual busca un consenso para llegar a una única decisión final. La figura 4 muestra gráficamente la arquitectura paralela. Una gran ventaja de esta arquitectura es puede ser fácilmente paralelizada.

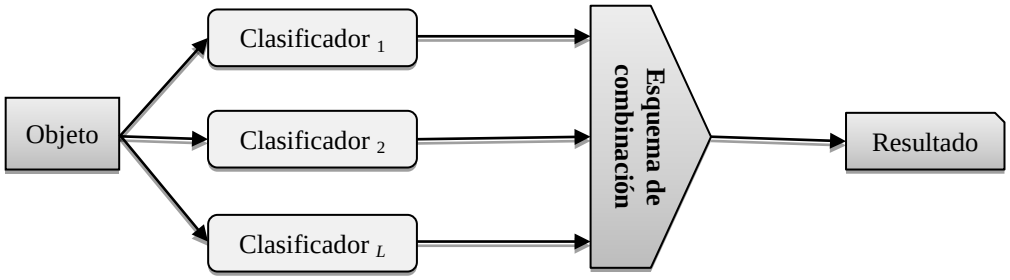


Fig. 4. Arquitectura paralela (horizontal).

Finalmente, en la arquitectura híbrida se combinan las dos arquitecturas anteriores: la secuencial y la paralela. Al unir estas dos arquitecturas en una, se puede obtiener mejor provecho de cada uno de los clasificadores utilizados. La arquitectura híbrida se muestra en la figura 5.

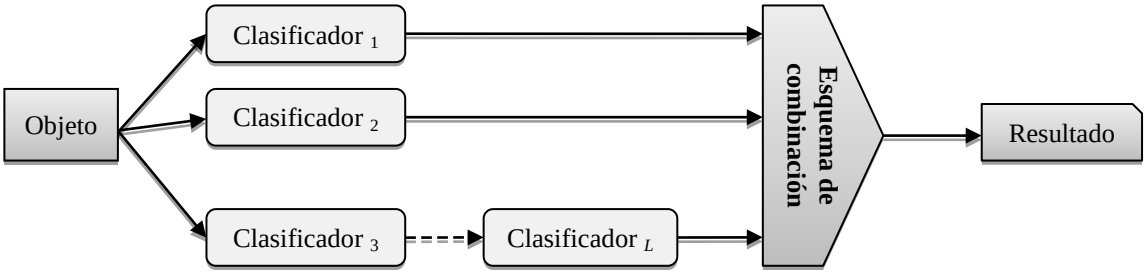


Fig. 5. Arquitectura híbrida (jerárquica).

4 Métodos basados en combinación de clasificadores

Los métodos de combinación de clasificadores son aquellos donde dado un conjunto de clasificadores ya entrenados, combinan los resultados de los distintos miembros del ensamblado para supuestamente retornar un resultado más preciso que el de los clasificadores bases.

Estos métodos se dividen principalmente en tres categorías, en dependencia del tipo de salida de los clasificadores: los que combinan etiquetas de clases, los que combinan un ranking de etiquetas y finalmente los que combinan los grados de pertenencia a una clase. La siguiente figura muestra la taxonomía propuesta para los métodos de combinación de clasificadores basada en las realizadas por [12, 14].

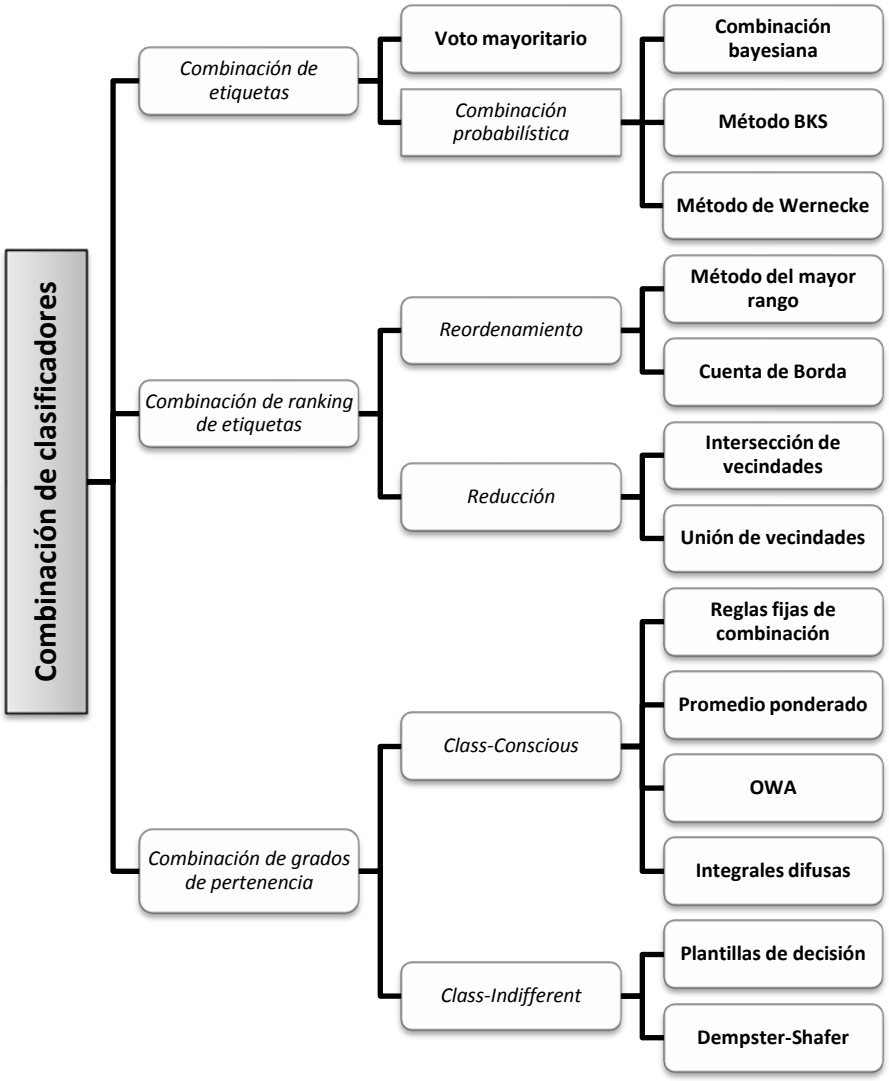


Fig. 6. Taxonomía de los métodos de combinación de clasificadores.

4.1 Combinación de etiquetas

En los métodos que combinación etiquetas cada clasificador D_i retorna una etiqueta de clase $s_i \in \Omega, i = 1, \dots, L$, para cada nuevo objeto $\mathbf{x} \in \mathbb{F}^n$ a ser clasificado. Finalmente, usando el vector $s = [s_1, \dots, s_L]^T \in \Omega^L$ el método es el encargado de tomar una decisión final.

4.1.1 Voto mayoritario

El voto mayoritario es la manera más intuitiva y simple de realizar la combinación de las etiquetas resultantes de un grupo de clasificadores.

Sea $[d_{i,1}, \dots, d_{i,c}]^T \in \{1, 0\}^c, i = 1, \dots, L$ un vector c-dimensional binario, resultado de un clasificador, donde $d_{i,j} = 1$ si el clasificador D_i etiqueta $\mathbf{x}$ en la clase ω_j y 0 en otro caso. La combinación de votos para cada clase ω_j es calculada

$$y_j = \sum_{i=1}^{L} d_{i,j} \,. \tag{4.1}$$

Para tomar la decisión final, existen tres patrones clásicos de consenso: unanimidad, simple mayoría y pluralidad. Además, incorporemos la clase ω_{c+1} al conjunto de clases ω, para clasificar los objetos en los cuales no sea posible determinar una etiqueta de clase con suficiente confiabilidad o se produjera un empate. El consenso final quedaría de la siguiente manera:

Unanimidad
$$\begin{cases} \omega_j, & \text{si } y_j = L \\ \omega_{c+1}, & \text{en otro caso} \end{cases}. \tag{4.2}$$

Simple mayoría
$$\begin{cases} \omega_j, & \text{si } y_j \geq \left\lfloor \frac{L}{2} \right\rfloor + 1 \\ \omega_{c+1}, & \text{en otro caso} \end{cases}. \tag{4.3}$$

Pluralidad
$$\begin{cases} \omega_j, & j = \arg\max_l y_l \\ \omega_{c+1}, & \text{en caso de empate} \end{cases}. \tag{4.4}$$

La pluralidad es el patrón de consenso que más suele emplearse en el voto mayoritario. Veamos a continuación algunas características del voto mayoritario.

Supongamos que L es un número impar de clasificadores con resultados independientes, los cuales poseen una probabilidad p de clasificar correctamente cualquier $\mathbf{x} \in \mathbb{F}^n$. De acuerdo con la ecuación (4.4) el voto mayoritario va a devolver una etiqueta de clase correcta si al menos $\lfloor L/2 \rfloor + 1$ clasificadores devuelven una respuesta correcta[2]. Entonces la precisión del ensamblado puede ser calculada mediante:

$$P_{vm} = \sum_{m=\lfloor L/2 \rfloor + 1}^{L} \binom{L}{m} p^m (1-p)^{L-m} \,. \tag{4.5}$$

También se cumple que si $p > 0.5$ entones P_{vm} es monótona creciente y $\lim_{L\to\infty} P_{vm} \to 1$, si $p < 0.5$ entonces P_{vm} es monótona decreciente y $\lim_{L\to\infty} P_{vm} \to 0$, y si $p = 0.5$ entonces $P_{vm} = 0.5$ para cualquier L. El planteamiento anterior también se le conoce como el Teorema del Jurado Condorcet [15]. En ese mismo artículo Shapley y Grofman validan su resultado para distintos valores de p, siempre y cuando los valores p distribuyen normalmente (o cualquier otra distribución que sea simétrica con respecto a la media). Lam y Suen [16] en su estudio analizan el caso de un número par L de clasificadores y del efecto de la precisión del agrupamiento al añadírsele o al eliminársele clasificadores.

Como puede notarse, todas las características anteriores están basadas en la suposición de que los clasificadores que conforman el agrupamiento sean independientes. ¿Y qué sucedería si trabajáramos con clasificadores dependientes?

Kuncheva [17] define los límites del voto mayoritario a través de los patrones de *éxito* y de *fracaso*. Donde el primero es la distribución más favorable de votos correctos, en la que se alcanza el mayor valor de P_{vm}, mejorando la precisión individual de los clasificadores mientras que la segunda es cuando el mayor número de votos correctos no son tenidos en cuenta en la decisión final provocando el menor valor de P_{vm}, empeorando la precisión de los clasificadores que conforman el agrupamiento.

[2] Nótese que la afirmación es suficiente y necesaria para dos clases, no así para $c > 2$ donde es solo suficiente.

Recientemente Brown y Kuncheva [18] descompusieron el error de clasificación del voto mayoritario en tres términos, la precisión individual del clasificador y dos medidas de diversidad, denominada una "buena" y la otra "mala" pues el aumento de sus valores provoca en la primera un decremento y en la segunda un aumento del error de clasificación.

Una manera de lidiar con clasificadores que no asumen independencia entre sus resultados, o que posean desigual precisión, es dándole mayor poder a los clasificadores más capaces a través de pesos.

Sea w_i es el peso asignado al clasificador D_i, donde $0 \leq w_i \leq 1$ y $\sum_{i=1}^{L} w_i = 1$ es recomendable pero no obligatorio, la ecuación (3.1) puede ser reescrita como:

$$y_j = \sum_{i=1}^{L} w_i d_{i,j} \, . \tag{4.6}$$

Esta técnica se le conoce en la literatura como Voto Mayoritario Ponderado. Kuncheva demuestra en [12] que combinando L clasificadores independientes con precisiones $p_1, \ldots, p_L$, la precisión del agrupamiento se maximiza asignando pesos proporcionales a

$$w_i \propto \log \frac{p_i}{1 - p_i} \, . \tag{4.7}$$

Estos pesos pueden ser obtenidos estimando la eficacia de los clasificadores usando un conjunto de entrenamiento o también mediante el uso de algoritmos genéticos [19].

El voto mayoritario es sin duda el esquema de combinación más empleado debido principalmente a su simpleza y fácil implementación. No utiliza memoria adicional a excepción cuando se utilizan pesos, siendo igual despreciable. El uso de clasificadores independientes como precondición para asegurar mejorar la precisión del resultado, más que una limitante se puede ver como una manera de construir los ensamblados de clasificadores. Diversos métodos de generación de ensamblados independientes como el Bagging y el Boosting han logrado muy buenos resultados usando el voto mayoritario como esquema de combinación.

4.1.2 Combinación probabilística

Otra de las maneras de tratar el problema de la fusión de etiquetas duras es a través del enfoque probabilístico. Definamos a la clase retornada por el clasificador D_i como una variable aleatoria discreta $s_i \in \{\omega_1, \ldots, \omega_c\}$. El grado de soporte que le da el ensamblado a cada clase se puede ver como $P(\omega_i | s_1, \ldots, s_L)$ siendo la clase con mayor valor de soporte el consenso final del agrupamiento.

$$\max_i \; P(\omega_i | s_1, \ldots, s_L) \, . \tag{4.8}$$

Combinación bayesiana

Este esquema de combinación asume que los clasificadores del ensamblado son mutuamente independientes en cuanto a sus salidas. Utilizando el teorema de Bayes

$$P(\omega_i | s_1, \ldots, s_L) = \frac{P(\omega_i) P(s_1, \ldots, s_L | \omega_i)}{P(s_1, \ldots, s_L)} \, . \tag{4.9}$$

y debido a la independencia condicional

$$P(s_1, \dots, s_L|\omega_k) = \prod_{i=1}^{L} P(s_i|\omega_k)\,. \tag{4.10}$$

sustituyendo (3.10) en (3.9), el grado de soporte para la clase ω_k puede ser calculado

$$\mu_k(\mathbf{x}) \propto P(\omega_k) \prod_{i=1}^{L} P(s_i|\omega_k)\,. \tag{4.11}$$

dado que el denominador $P(s_1, \dots, s_L)$ puede ser ignorado debido a que no depende de ω_k.

En la práctica, $P(\omega_k)$ y $P(s_i|\omega_k)$ pueden ser estimadas mediante un conjunto de datos $\mathbf{Z}$ de cardinalidad N, donde $P(\omega_k)$ es aproximada por N_k/N siendo N_k el número de elementos de $\mathbf{Z}$ que pertenecen a la clase ω_k y $P(s_i|\omega_k)$ por $cm^i_{k,s_i}/N_k$ donde cm^i_{k,s_i} representa el número de elementos de $\mathbf{Z}$ cuyas clases verdaderas son ω_k y fueron etiquetados en la clase ω_s por el clasificador D_i.

El principal inconveniente de esta combinación es que asume independencia en los resultados de los clasificadores como precondición, sin embargo ha resultado ser precisa y eficiente en varios estudios experimentales. Incluso ha sido robusta ante clasificadores que no asumen independencia entre sus resultados [12].

Espacio de comportamiento-conocimiento

También conocido en inglés como Behavior-Knowledge Space (BKS) [20]. Este método al contrario de la combinación bayesiana, no asume independencia en los resultados de los clasificadores del ensamblado. La idea de este método es la de estimar la probabilidad a posteriori de que una clase sea la repuesta correcta al calcular la frecuencia de cada clase para cada combinación de las decisiones de los clasificadores, usando un conjunto de entrenamiento.

Formalmente el BKS es un espacio L dimensional donde cada dimensión representa la decisión de un clasificador. Este espacio es representado a través de una tabla con c^L columnas (una para cada posible combinación de respuestas) y c filas, es decir c^{L+1} celdas. Cada celda va estar denotada por $bks(j_1, \dots, j_L)(i)$ y va a representar la cantidad del objetos del conjunto de entrenamiento cuya verdadera etiqueta de clase es ω_i y los clasificadores del ensamblado retornaron las etiquetas $\omega_{j_1}, \dots, \omega_{j_L}$ como respuesta.

Sea $\omega_{j_1}, \dots, \omega_{j_L}$ el conjunto de etiquetas devueltas por los L clasificadores respectivamente para un determinado objeto $\mathbf{x}$, este método retorna la siguiente respuesta

$$\begin{cases} \omega_{R(j_1,\dots,j_L)}\,, & \sum_{i=1}^{C} bks(j_1, \dots, j_L)(i) > 0 \ \wedge\ \mu_{R(j_1,\dots,j_L)}(\mathbf{x}) \geq \alpha \\ \omega_{c+1}, & \text{en otro caso} \end{cases}, \tag{4.12}$$

donde

$$\mu_i(\mathbf{x}) = \frac{bks(j_1,\dots,j_L)(i)}{\sum_{i=1}^{C} bks(j_1,\dots,j_L)(i)}\,, \tag{4.13}$$

es el grado de soporte de que la clase ω_i sea la clase correcta para el conjunto de respuestas $\omega_{j_1}, \dots, \omega_{j_L}$ dadas por los L clasificadores. $R(j_1, \dots, j_L) = \arg\max_i\,(bks(j_1, \dots, j_L)(i))$, es la etiqueta de la clase representativa (clase más probable) para el conjunto de respuestas $\omega_{j_1}, \dots, \omega_{j_L}$. Y α, $0 \leq \alpha \leq 1$ es un umbral de rechazo. En caso que el grado de soporte sea menor que el umbral α, la clase ω_{c+1} se usa para representar la imposibilidad del método de tomar una decisión.

Se puede dar el caso que existan celdas en la tabla BKS con valor cero. Esto quiere decir que durante la creación de la tabla esta combinación de salidas no fue vista. En ese caso al objeto a clasificar se le asigna la etiqueta ω_{c+1}.

Este método de combinación sufre dos grandes desventajas [21]. La primera es el hecho de que necesita de muy grandes conjuntos de entrenamiento para representar todas las posibles combinaciones de salidas de los clasificadores y a su vez lograr que no queden celdas vacías en la tabla BKS. La segunda es el alto error producto a que una clase representativa posee muy baja probabilidad, quedando el resultado ambiguo. Otra desventaja desde el punto de vista computacional es la memoria que requiere C^{L+1}, siendo imposible resolver problemas de gran número de clases con muchos clasificadores.

En esencia este método trata de recrear todas las posibles combinaciones que pudieran darse a la hora de clasificar un objeto por un grupo de clasificadores. La respuesta final está basada en el comportamiento observado mediante un conjunto de entrenamiento. Esto hace que el método sea extremadamente dependiente al conjunto de datos empleado. Un conjunto de datos el cual no generalice lo suficiente el espacio del problema que se quiere resolver puede provocar un sobre entrenamiento.

Extensiones de este método han sido realizadas para la fusión de clasificadores que devuelven ranking de etiquetas [22] y resultados continuos (grados de pertenencia a cada clase) [23]. El primero, según sus autores, ofrece un mejor control del rechazo sin tener que usar nuevos clasificadores para las celdas vacías. Mientras que el segundo presenta como ventaja que no realiza ninguna asunción del ensamblado de clasificadores. Al trabajar con los grados de pertenencia a cada clase, este método posee más información para la toma de la decisión final.

Método de Wernecke

Propuesto por Wernecke [24], este método busca reducir el sobre entrenamiento a veces producido por el BKS. Inicialmente se construye una tabla al igual que se hace en el BKS, pero con la diferencia que al construirla se tiene en cuenta un 95 por ciento del intervalo de confianza en cada celda de la tabla. De ocurrir un solapamiento de los intervalos de confianza en las celdas de las clases para una configuración de salida de los clasificadores, la clase representativa de dicha configuración no se considera lo suficientemente dominante como para ser retornada. En ese caso el clasificador menos erróneo de los L que conforman el ensamblado, es decir el de menor probabilidad $P[error \wedge D_i(\mathbf{x}) = s_i]$ es el encargado de clasificar el objeto **x**.

Sea $k_1, \dots, k_c$ el número de elementos del conjunto de entrenamiento pertenecientes a las clases $\omega_1, \dots, \omega_c$ respectivamente donde la respuesta de los clasificadores fue $\mathbf{s} = (s_{j_1}, \dots, s_{j_L})$, $s_i \in \Omega$, es decir $k_i = bks(j_1, \dots, j_L)(i)$. Este método asume a los k_i como una variable aleatoria que sigue una distribución normal. Por tanto los intervalos de confianza pueden ser calculados usando la aproximación normal de la distribución binomial. Para esto, la siguiente *regla del pulgar*[3] debe cumplirse

$$k = \sum_{i=1}^{c} k_i \geq 30 \ , \qquad k_j \geq 5 \qquad k - k_j \geq 5 \tag{4.14}$$

calculándose el intervalo de confianza de la siguiente manera:

[3] Principio o criterio de amplia aplicación que no necesariamente es estrictamente preciso ni fiable en cada situación. Establece una especie de formula u observación generalmente aceptada como conocimiento práctico basado en la experiencia.

$$\mathrm{IC}(\omega_j, 95) = \left[k_j - 1.96\sqrt{\frac{k_j(k-k_j)}{k}} + \frac{1}{2}, k_j + 1.96\sqrt{\frac{k_j(k-k_j)}{k}} + \frac{1}{2} \right]. \quad (4.15)$$

Este método con la utilización de intervalos de confianza evita en muchas ocasiones la toma de decisiones ambiguas tal y como sucedía en el BKS. Sin embargo al igual que el BKS tiene la desventaja que necesita de grandes conjuntos de datos para construir la tabla y calcular los intervalos de confianza. Con pequeños conjuntos de datos existe la posibilidad de quedar celdas vacías e incluso con un pequeño valor tal que no se pueda cumplir la regla (4.14). Además esto provoca que en muchas ocasiones los intervalos de confianza queden muy abiertos, provocando que casi siempre exista solapamiento. La memoria también es una limitante ya que al igual que el BKS se necesita de una tabla con c^{L+1} celdas.

4.2 Combinación de ranking de etiquetas

Existe un conjunto de clasificadores que en vez de solo devolver la etiqueta de la clase que considere correcta, devuelven una secuencia ordenada de etiquetas de clase candidatas. En la secuencia devuelta, la etiqueta de la primera posición representa la clase que el clasificador considere con mayor posibilidad de ser la correcta mientras que la ultima la de menor posibilidad.

A la hora de combinar clasificadores con este tipo de salida, Ho [14] propone dos enfoques para abordar la problemática. El primero, se basa en reducir al mínimo el número de etiquetas que conforman el ranking devuelto por los clasificadores tratando de asegurar que la clase correcta esté incluida en el ranking reducido. Para este enfoque propone dos métodos: la intersección de vecindades y la unión de vecindades [14]. El segundo enfoque se basa en el reordenamiento de los rankings de tal manera que la etiqueta verdadera esté lo más cercana posible de las primeras posiciones. A continuación veremos dos métodos de combinación de ranking.

4.2.1 Método del mayor rango

Sean $\mathrm{r}_{\mathrm{i},1}, \ldots, \mathrm{r}_{\mathrm{i},\mathrm{c}}$ los valores de los rangos asignados por un clasificador D_i a la clase ω_j. Donde el menor valor va a representar el mayor rango. A cada clase se le asigna el valor

$$y_j = \min_i r_{i,j}\,, \quad (4.16)$$

es decir, el mínimo de los rangos asignados por los L clasificadores que conforman el ensamblado. El nuevo ranking que va a devolver el método no es más que el conjunto de clases ordenadas de acuerdo con su valor y_j. Pueden existir empates los cuales son resueltos de manera arbitraria.

Este método es particularmente útil en problemas que involucren grandes números de clases y un conjunto reducido de clasificadores. Sin embargo, su principal desventaja radica en que el nuevo ranking devuelto puede poseer muchos empates. Por esa razón este método es más apropiado cuando el número de clasificadores es pequeño con relación al número de clases [25]. Además, consideramos que este método es demasiado optimista en cuanto las respuesta de los clasificadores, sin tener en cuenta el consenso entre los clasificadores. Supongamos un conjunto de clasificadores donde la mayoría otorgue un bajo rango para una determinada entrada **x**; mientras exista al menos un clasificador que otorgue un rango cercano al tope para esa misma entrada, el método se va a dejar influenciar más por este clasificador individual sin tener en cuenta el consenso existente entre la mayoría de los clasificadores.

4.2.2 Método de la cuenta de Borda

La combinación de decisiones en los sistemas multiclasificadores puede ser vista como un problema de votación dentro de la teoría de decisión en grupos. Por tanto, la combinación puede ser realizada

mediante una función de consenso de grupo. Una de las funciones de consenso de grupo más usadas es la cuenta de Borda, la cual es un sistema de votación desarrollado por Jean-Charles de Borda en 1770.

Sea $B_i(\omega_j)$ el número de clases en Ω que poseen un rango por debajo a la clase ω_j en el ranking realizado por el clasificador D_i, para todo $\omega_j \in \Omega$. La cuenta de Borda [14] para una clase ω_j quedaría definida como

$$B(\omega_j) = \sum_{i=1}^{L} B_i(\omega_j) \ . \tag{4.17}$$

El ranking combinado va a ser el conjunto de clases ordenadas de forma descendiente por su valor de cuenta de Borda.

El valor de la cuenta de Borda mide la fuerza en el consenso de los clasificadores, de que un objeto pertenece a una determinada clase. De forma intuitiva, si una clase ω_j obtiene un rango cerca al tope por varios clasificadores, entonces su valor de cuenta de Borda va a ser elevado, por consiguiente va a tener una buena ubicación en el ranking combinado que va a ser devuelto por el método.

Este método es muy simple de implementar y no requiere un conocimiento previo del comportamiento de los clasificadores. Es decir, todos los clasificadores son tratados de manera igual, lo cual no es siempre deseable, en especial cuando se tiene conocimiento de que ciertos clasificadores se desempeñan mejor que otros en algunos casos.

Con vistas a resolver la limitante anterior, se suele utilizar pesos asociados a cada uno de los clasificadores. De esta manera la ecuación (4.17) puede ser reescrita como

$$WB(\omega_j) = \sum_{i=1}^{L} w_i B_i(\omega_j) \, , \tag{4.18}$$

donde los pesos w_i pueden representar simplemente la eficacia de cada uno de los clasificadores medida usando un conjunto de validación.

Este método resuelve una de las limitantes que presentaba el anterior, ya que la ubicación final de una clase en el ranking final obedece más al consenso de los clasificadores y no es influenciada por un clasificador en particular. Sin embargo, la presencia de empates entre distintas clases en una misma posición del ranking final sigue siendo una posibilidad a resolver de forma arbitraria, sin que el método presente una solución efectiva a esta problemática.

Para concluir, se puede decir que los métodos de combinación de ranking de etiquetas son quizás los menos empleados en la combinación de clasificadores supervisados debido, primeramente, a que pocos clasificadores están diseñados para producir valores de ranking de salida. Segundo, porque a pesar de ser posible convertir aquellos que devuelven el grado de pertenencia a una clase en un ranking de clases, esto desaprovecharía información que sería empleada por otros métodos diseñados para combinar salidas continuas. A continuación abordaremos dichos métodos.

4.3 Combinación de grados de pertenencia a una clase

Existen diversos clasificadores que devuelven como resultados vectores numéricos c-dimensionales, donde cada valor corresponde a una clase del problema. Estos valores pueden ser interpretados de varias maneras, pero las dos más comunes son como el grado de confianza que otorga el clasificador a que esa sea la clase correcta, y la otra, como un estimado de la probabilidad posterior de la clase.

Sea $\mathbf{x} \in \mathbb{F}^n$ un objeto representado por una tupla de un cierto espacio n-dimensional $\mathbb{F}^n$ y $\Omega = \{\omega_1, \dots, \omega_c\}$ el conjunto de etiquetas de clases. Cada clasificador D_i que pertenece al ensamblado $\{D_1, \dots, D_L\}$ devuelve c grados de confianza, es decir $D_i: \mathbb{F}^n \to \mathbb{R}^c$, donde $\mathbb{R}$ es el conjunto de los números reales. Sin embargo, lo más común es que los valores de los grados de confianza estén normalizados en el intervalo $[0, 1]$. Además denotemos por $d_{i,j}(\mathbf{x})$ el soporte del clasificador D_i a la

hipótesis de que **x** pertenezca a la clase ω_j, donde mientras más grande el soporte, más probable de que ω_j sea la etiqueta de **x**. Los resultados de los L clasificadores que conforman el ensamblado pueden ser organizados mediante una matriz denominada perfil de decisión, conocida en inglés como decision profile ($DP(\mathbf{x})$).

Los esquemas que combinen semejantes clasificadores, pueden ser divididos en dos grupos. El primero, conocido en inglés como Class-Conscious, busca estimar el grado de soporte en conjunto para cada clase ω_j partiendo de los grados de soporte individuales dados por el ensamblado de clasificadores, los cuales están almacenados en la columna j-ésima de la $DP(\mathbf{x})$. A diferencia de este, el segundo trata los valores $d_{i,j}(\mathbf{x})$ como rasgos de un nuevo espacio de rasgos denominado *espacio de rasgos intermedio*. La decisión final es realizada por otro clasificador que toma como entrada el espacio de rasgos intermedio y devuelve una etiqueta de clase. Este grupo de métodos son conocidos en inglés como Class-Indifferent. A continuación abordaremos ambos enfoques.

4.3.1 Esquemas de combinación "Class-Conscious"

Como habíamos visto anteriormente, estos esquemas buscan estimar el grado de soporte $\mu_j(\mathbf{x})$ para cada clase ω_j. El vector de entrada **x** recibirá la etiqueta de la clase con mayor valor de $\mu_j(\mathbf{x})$. Otro aspecto importante es la transformación de las salidas de los clasificadores, debido a que no todos los clasificadores devuelven sus resultados en la misma escala además de estos tener diferentes significados. Estas salidas deben ser por lo menos normalizadas a una misma escala, preferiblemente en el intervalo $[0, 1]$. Aunque es más aconsejable que cada valor represente la probabilidad posterior de que un nuevo objeto **x** pertenezca a su respectiva clase. Kuncheva [12] define varias maneras de lograr dicho propósito. A continuación serán abordados cuatro métodos que siguen este esquema de combinación.

Reglas fijas de combinación

De esta manera es como son conocidas un conjunto de funciones de combinación, las cuales no necesitan de parámetros ajustables. Por lo que no necesitan de un entrenamiento posterior, una vez que los clasificadores que la conforman hayan sido entrenados. El grado de soporte para la clase ω_j quedaría como

$$\mu_j(\mathbf{x}) = \mathcal{F}\left[d_{1,j}(\mathbf{x}), \dots, d_{L,j}(\mathbf{x})\right], \tag{4.19}$$

donde $\mathcal{F}$ es una función de combinación (máximo, mínimo, producto, promedio, etc.).

Las funciones de combinación máximo y mínimo definen los niveles de optimismo extremos a la hora de tomar una decisión final. El mínimo tomaría la decisión más pesimista pues devolvería el menor soporte dado por los clasificadores mientras que el máximo sería el más optimista al devolver el mayor soporte otorgado por uno de los clasificadores. El promedio y el producto son las dos funciones más usadas y estudiadas, sin embargo no existe una regla clara de cuál es la mejor para un determinado problema [12]. La selección de una determinada función de combinación está muy asociada al problema que se quiera resolver.

En general, las reglas fijas de combinación son uno de los esquemas Class-Conscious mayormente usado debido principalmente a su simpleza y fácil implementación.

Promedio ponderado

También conocido como combinación ponderada, los soportes otorgados a una clase por los L clasificadores son promediados usando pesos. El soporte para una clase ω_j quedaría:

$$\mu_j(\mathbf{x}) = \sum_{i=1}^{L} w_i d_{i,j}(\mathbf{x}). \tag{4.20}$$

Existen dos variantes las cuales discrepan en la cantidad de pesos utilizados. En la primera L pesos son utilizados, uno para cada clasificador. Mientras que en la segunda son empleados $c \times L$ pesos, cada uno asignado a una clase por cada clasificador. Usualmente los pesos son condicionados a ser no negativos, incluso a que $\sum_{i=1}^{L} w_i = 1$. Sin embargo se ha comprobado que la primera restricción ni mejora ni empeora el resultado de la combinación.

La regresión lineal es el procedimiento más común para estimar los pesos, siendo el Mínimo Error Cuadrático (MEC) el criterio tradicional para la regresión [26]. El MEC conduce a una mejora en la clasificación de una manera indirecta a través de la aproximación de las probabilidades a posteriori. Resulta más natural minimizar el error de clasificación aunque su inconveniente radica en que no existe una solución analítica fácil para la estimación de los pesos. Una posibilidad sería tratar la combinación de los resultados de los clasificadores como un problema de reconocimiento de patrones y aplicar Análisis Lineal Discriminante (LDA) [27]. Ueda [27] propone utilizar un método de descenso probabilístico para derivar los pesos. Otros autores han utilizado Algoritmos Genéticos para hallar los pesos [19, 28].

Promedio ponderado ordenado

También conocido en la literatura en inglés como Ordered Weighted Averaging (OWA) [29]. Este método utiliza *L* coeficientes, uno para cada clasificador que conforma el ensamblado de clasificadores. Vale precisar que los coeficientes no están asociados a cada clasificador sino a la posición de sus salidas ordenadas.

Para calcular el grado de soporte que le es otorgado a cada clase, primeramente los valores de soporte de cada clasificador asociado a dicha clase son ordenados en forma descendente. Acto seguido la suma ponderada se calcula usando los coeficientes asociados a la posición en la ordenación.

Formalmente, sea $\mathbf{b} = [b_1, \dots, b_L]$ un vector con coeficientes tal que $\sum_{k=1}^{L} b_k = 1$. El soporte para la clase ω_j es estimado como el producto escalar de $\mathbf{b}$ con el vector $\left[d_{i_1,j}(\mathbf{x}), \dots, d_{i_L,j}(\mathbf{x})\right]^T$ donde $i_1, \dots, i_L$ es la permutación de los índices $1, \dots, L$ tal que $d_{i_1,j}(\mathbf{x}) \geq d_{i_2,j}(\mathbf{x}) \geq \cdots \geq d_{i_L,j}(\mathbf{x})$. Por lo tanto el grado de soporte para la clase ω_j quedaría como

$$\mu_j(\mathbf{x}) = \frac{1}{L} \sum_{k=1}^{L} b_k d_{i_k,j}(\mathbf{x}) \,. \tag{4.21}$$

Este método impide acreditar a un clasificador en particular, con la más alta capacidad en todo el espacio, como sería el caso si se asignaran pesos fijos a los clasificadores. Por ejemplo, si el mejor clasificador ha recibido el crédito debido a un sobre entrenamiento de los datos, producto de eso, podemos enfrentarnos a una pobre generalización. Por lo que este método parece ser más robusto que el promedio ponderado, donde los coeficientes son calculados sobre la base de los resultados del clasificador [30]. El vector de coeficientes **b** lo mismo puede ser escogido por antelación que estimado de tal manera que minimice el error de clasificación usando un conjunto de entrenamiento.

Integrales difusas

La idea detrás de la combinación mediante integrales difusas [31, 32] es la de medir la "fortaleza", no solo para cada clasificador, sino para todos los subconjuntos de clasificadores. Cada subconjunto de clasificadores posee una medida de aptitud la cual describe qué tan bueno es ese subconjunto para una determinada entrada **x**. El soporte $\mu_j(\mathbf{x})$ otorgado a la clase ω_j por el ensamblado de clasificadores se

obtiene de los valores de soporte $d_{i,j}(\mathbf{x}), i = 1, \ldots, L$ teniendo en cuenta la aptitud de los grupos de diversos subgrupos de clasificadores.

Sea H un conjunto difuso sobre un ensamblado de clasificadores $\mathcal{D}$ el cual expresa el soporte para una clase ω_j. Definamos como medida difusa a la función $g: 2^{\mathcal{D}} \rightarrow [0,1]$ tal que $g(\emptyset) = 0$, $g(\mathcal{D}) = 1$ y para cualquier A y B subconjunto de $\mathcal{D}$, $A \subset B \Rightarrow g(A) \leq g(B)$. La medida difusa g es empleada para tener en cuenta la importancia de cualquier subconjunto de clasificadores con respecto a una clase ω_j. Dos tipos básicos de integrales difusas han sido propuestas: la Sugeno y la Choquet. La integral difusa de Sugeno con respecto a una medida difusa g, es calculada de la siguiente manera:

$$\mathcal{A}_g^{FI} = \max_{\alpha} \left\{ \min(\alpha, g(H_\alpha)) \right\}, \tag{4.22}$$

donde H_α es un α-corte de H, $H_\alpha = \{x | d_{i,j}(\mathbf{x}) \geq \alpha\}$.

Una medida difusa g es denominada λ-difusa si para cualquier A y B subconjunto de $\mathcal{D}$, tal que $A \cap B \neq \emptyset$,

$$g(A \cup B) = g(A) + g(B) + \lambda g(A) g(B), \qquad \lambda \in (-1, \infty). \tag{4.23}$$

La medida λ-difusa es estimada usando un conjunto de L valores g^i, denominados densidades difusas. Estas densidades difusas representan la importancia individual de un clasificador D_i. El valor de λ es obtenido como la única raíz real mayor que -1 del polinomio

$$\lambda + 1 = \prod_{i=1}^{L} (1 + \lambda g^i), \qquad \lambda \neq 0. \tag{4.24}$$

La siguiente tabla muestra el pseudo-código para combinar clasificadores mediante integrales difusas.

Tabla 2. Combinación de clasificadores mediante integrales difusas.

Entrada:
$D_1, \ldots, D_L$: Conjunto de clasificadores ya entrenados.
$g^1, \ldots, g^L$: Densidades difusas.
x: Objeto a clasificar.
Salida:
ω: Etiqueta asignada al objeto **x.**

1. **Para cada** clase ω_j, $j = 1, \ldots, c$
   a. Obtener los grados de pertenencia de la clase ω_j para cada clasificador del ensamblado.
      $$d_{i,j}(\mathbf{x}) = D_i(\mathbf{x}), \qquad i = i, \ldots, L$$
   b. Ordenar los grados de pertenencia $d_{i,j}(\mathbf{x})$ de mayor a menor, obteniendo un nuevo vector $[d_{i_1,j}(\mathbf{x}), d_{i_2,j}(\mathbf{x}), \ldots, d_{i_L,j}(\mathbf{x})]^T$.
   c. Reordenar las densidades difusas en correspondencia con la ordenación de los grados de pertenencia, obteniendo $g^{i_1}, \ldots, g^{i_L}$.
   d. $g(1) = g^{i_1}$
   e. **Para** $t = 2$ **hasta** L calcular recursivamente
      $$g(t) = g^{i_t} + g(t-1) + \lambda g^{i_t} g(t-1).$$
   f. Calcular el valor de soporte para la clase ω_j

$$\mu_j(\mathbf{x}) = \max_{1 \le t \le L} \left\{ \min\{ d_{i_t,j}(\mathbf{x}), g(t) \} \right\}.$$

2. **Retornar** la etiqueta de la clase con mayor valor de soporte.

El valor de soporte $\mu_j(\mathbf{x})$ para una clase ω_j, puede ser visto como el entendimiento entre la competencia (representada por la medida difusa g) y la evidencia (representada por los grados de pertenencia de una clase otorgado por los L clasificadores). Nótese que el vector de medida difuso puede ser diferente para cada clase y es específico al objeto **x** en cuestión.

El otro tipo de integral difusa utilizada, la Choquet, utiliza la misma medida λ-difusa. Su única diferencia es la manera en estimar el valor de soporte para una clase ω_j, el cual sería

$$\mu_j(\mathbf{x}) = d_{i_1,k}(\mathbf{x}) + \sum_{k=2}^{L} \left[d_{i_{k-1},j}(\mathbf{x}) - d_{i_k,j}(\mathbf{x}) \right] g(k-1) \,. \tag{4.25}$$

La idea de combinación mediante integrales difusas fue usada por Bulacio [33] en una estrategia de selección-combinación. En un primer proceso, un grupo conformado por clasificadores que sean cooperativos son seleccionados del total, para posteriormente ser combinados usando integrales difusas de Sugeno.

4.3.2 Esquemas de combinación "Class-Indifferent"

Los esquemas en este grupo obtienen el grado de soporte $\mu_j(\mathbf{x})$ para una determinada clase ω_j, usando los $L \times c$ grados de soporte otorgados por los clasificadores y almacenados en la $DP(\mathbf{x})$. Cada vector en el espacio de rasgos intermedio es una versión expandida del $DP(\mathbf{x})$ obtenida al concatenar las L filas del $DP(\mathbf{x})$. En este enfoque, cualquier clasificador puede utilizarse para tomar la decisión final, desde una simple regresión lineal [27] hasta incluso redes neuronales [34]. A continuación veremos dos métodos que utilizan este esquema de combinación.

Plantillas de decisión

La idea de este esquema de combinación, propuesto por Kuncheva [35], es la de recordar el perfil de decisión $DP(\mathbf{x})$ más típico para cada clase ω_j denominado *Plantilla de Decisión* DT_j. A la hora de clasificar el objeto de entrada **x**, el esquema le asignará la etiqueta de la clase cuya plantilla sea más similar al perfil de decisión $DP(\mathbf{x})$ de **x** usando una medida de similaridad S. Las tablas 3 y 4 mostradas a continuación contienen el pseudo-código del método para su mejor entendimiento.

Tabla 3. Plantillas de decisión (fase de entrenamiento).

Entrada:

$D_1, \ldots, D_L$: Conjunto de clasificadores ya entrenados.

Z: Conjunto de datos etiquetados.

Salida:

$DT_1, \ldots, DT_c$: Conjunto de plantillas de decisión.

1. **Para cada** clase ω_j, $j = 1, \ldots, c$
   a. Calcular la plantilla de decisión para la clase ω_j

$$DT_j = \frac{1}{N_j} \sum_{\substack{\mathbf{z}_j \in \omega_j \\ \mathbf{z}_j \in \mathbf{Z}}} DP(\mathbf{z}_j) \,,$$

donde N_j es el número de elementos de $\mathbf{Z}$ que pertenecen a la clase ω_j.

2. **Retornar** $DT_1, \ldots, DT_c$.

Tabla 4. Plantillas de decisión (fase de clasificación).

Entrada:

$\mathbf{x}$: Objeto a clasificar.

$D_1, \ldots, D_L$: Conjunto de clasificadores ya entrenados.

$DT_1, \ldots, DT_c$: Conjunto de plantillas de decisión.

Salida:

ω: Etiqueta asignada al objeto $\mathbf{x}$.

1. Construir la $DP(\mathbf{x})$.
2. **Para cada** clase ω_j, $j = 1, \ldots, c$
   a. Calcular la similaridad entre $DP(\mathbf{x})$ y DT_j
      $\mu_j(\mathbf{x}) = S\left(DP(\mathbf{x}), DT_j\right)$
3. **Retornar** la etiqueta de la clase con mayor valor de similaridad.

Como medida de similaridad, la más comúnmente empleada es la *Distancia Euclidiana Cuadrada*. Donde el grado de soporte otorgado por el método quedaría como

$$\mu_j(\mathbf{x}) = 1 - \frac{1}{L \times c} \sum_{i=1}^{L} \sum_{k=1}^{c} \left[DT_j(i,k) - d_{i,k}(\mathbf{x}) \right]^2 . \quad (4.26)$$

Si $DP(\mathbf{x})$ y DT_j son vistos como vectores en el espacio de rasgos intermedio $c \times L$ dimensional, el grado de soporte vendría dado por el valor negativo de la distancia euclidiana cuadrada entre los dos vectores. Otras medidas de distancia también pueden ser usadas como es el caso de la Minkowski, Mahalanobis, entre otras.

Otra medida de similaridad fue presentada en [36], para este esquema. Esta proviene de la teoría de conjuntos difusos. El grado de soporte para una clase vendría dado por

$$\mu_j(\mathbf{x}) = 1 - \frac{1}{L \times c} \sum_{i=1}^{L} \sum_{k=1}^{c} \max\left\{ \min\left\{ DT_j(i,k), 1 - d_{i,k}(\mathbf{x}) \right\}, \min\left\{ 1 - DT_j(i,k), d_{i,k}(\mathbf{x}) \right\} \right\} . \quad (4.27)$$

Las plantillas de decisión son un esquema de combinación dentro del grupo class-indifferent debido a que tratan las salidas de los clasificadores como un conjunto de rasgos fuera de contexto. Todos los esquemas que pertenecen al grupo class-conscious son idempotentes por diseño, es decir, que si el conjunto de clasificadores consiste de L copias de un clasificador D, la decisión del esquema de combinación no va a ser diferente a la decisión de D. Sin embargo, la respuesta dada por el esquema basado en plantillas de decisión no tiene por qué ser idéntica a la de D, puede ser mejor o incluso peor [12].

Combinación Dempster-Shafer

La teoría de evidencia Dempster-Shafer (D-S) [37], es una poderosa herramienta para representar conocimiento incierto. Su enunciado estuvo motivado por las dificultades encontradas en la Teoría de la

Probabilidad para representar la ignorancia y manejar la necesidad de que las creencias asignadas a un evento y su negación sumen uno. Esta intenta sacar beneficio de la utilización de conjuntos de hipótesis en lugar de las hipótesis por separado y procura facilitar la reasignación de probabilidad de creencia en las hipótesis cuando cambian las evidencias.

En esta teoría, el dominio de un problema es representado por un conjunto finito Θ de hipótesis mutuamente excluyentes denominado *marco de discernimiento*. En la teoría convencional de probabilidad, a todos los elementos de Θ le son asignados probabilidades y cuando el grado de soporte de un evento es conocido, el resto del soporte es automáticamente asignado a la negación del evento. Por el contrario, en la teoría D-S, se realizan asignaciones masivas a los eventos tal como son, y otorgar soporte a un evento no necesariamente implica que los soportes restantes son otorgados a su negación. Formalmente, una *asignación de probabilidad básica* (BPA, por sus siglas en inglés) es una función $m: 2^{\Theta} \rightarrow [0, 1]$ que cumpla

$$m(\emptyset) = 0 \ \wedge \ \sum_{A \in 2^{\Theta}} m(A) = 1 \, . \tag{4.28}$$

Esta función asigna a cada elemento de 2^{Θ} un valor indicativo de la creencia que, dada una evidencia, se deposita en él. Un conjunto $A \in 2^{\Theta}$ con $m(A) > 0$ es denominado elemento focal de m. Y m es denominado vacío si $m(\Theta) = 1$ y $m(A) = 0$ para todo $A \neq \emptyset$.

La función de creencia es una función la cual asigna un valor en el intervalo [0,1] a cualquier subconjunto no vacío A de Θ. Esta se define como

$$Bel_m(A) = \sum_{B \subseteq A} m(B) \, . \tag{4.29}$$

La diferencia entre $m(A)$ y $Bel_m(A)$ es que mientras $m(A)$ es la creencia dada al subconjunto A excluyendo cualquiera de sus subconjuntos propios, $Bel_m(A)$ es el grado de creencia en A así como en todos sus subconjuntos.

Una de las operaciones más útiles que desempeña un papel central en la manipulación de las funciones de creencia es la *regla de combinación de Dempser*. Consideremos dos muestras de evidencia del mismo marco de discernimiento Θ representadas por dos BPA m_1 y m_2. La regla de combinación Dempster es entonces usada para generar una nueva BPA denotada por $m_1 \oplus m_2$ (también se le conoce como suma ortogonal entre m_1 y m_2) la cual se define como

$$m_1 \oplus m_2(\emptyset) = 0, \qquad m_1 \oplus m_2(A) = \frac{1}{1-k} \sum_{B \cap C = A} m_1(B) m_2(C) \ , \tag{4.30}$$

donde $k = \sum_{B \cap C = \emptyset} m_1(B) m_2(C)$. Nótese que la combinación mediante suma ortogonal es aplicable a aquellas BPA que cumplan la condición $k < 1$.

La teoría de evidencia Dempster-Shafer posee la habilidad de representar incertidumbre y falta de conocimiento. Esto es bastante importante en problemas de combinación de clasificadores debido a que existe un cierto nivel de incertidumbre asociado al desempeño de cada uno de los clasificadores que se deseen combinar.

Diversos han sido los métodos de combinación de clasificadores los cuales utilizan la teoría D-S. Uno de los primeros métodos fue desarrollado por Mandler y Shurmann [38]. Ellos propusieron un método que transforma las medidas de distancia de los diferentes clasificadores en evidencia. Esto es logrado calculando primeramente la distancia entre los conjuntos de entrenamiento y un número de puntos de referencia con el objetivo de estimar la distribución estadística de las distancias entre e intra-clases. Para ambas, la función de probabilidad a posteriori es estimada indicando el grado con el cual un patrón de entrada pertenece a cierto punto de referencia. Por eso, para cada etiqueta de clase, las

probabilidades condicionales de las clases son combinadas dentro de valores de evidencia entre 0 y 1, lo cual es considerado como la BPA de la clase. Finalmente, la regla de combinación de Dempster es usada para combinar las BPA de los diferentes clasificadores para devolver el resultado final. Según Rogova [39] este método trae muchas interrogantes acerca de la elección de los vectores de referencia y las medidas de distancia. Más aun, las aproximaciones asociadas con la estimación de los parámetros de los modelos estadísticos para las distancias entre e intra clases, pueden llevar a medidas de evidencia inexactas.

Rogova [39] propuso un método el cual utiliza varias medidas de proximidad entre un vector de referencia y el vector de salida del clasificador. La medida de proximidad que otorgue la mayor precisión de clasificación es la trasformada en evidencia. Finalmente, la regla de combinación de Dempster es usada para combinar las evidencias de todos los clasificadores para obtener un grado de soporte para cada clase. La siguiente tabla muestra el pseudo-código del método donde DT_j^i denota la i-ésima fila de la plantilla de decisión DT_j. $D_i(\mathbf{x})$ representa los valores de soporte $D_i(\mathbf{x}) = [d_{i,1}(\mathbf{x}), \dots, d_{i,c}(\mathbf{x})]^T$ otorgados por el clasificador D_i al objeto $\mathbf{x}$. Además, siendo K una constante de normalización.

Tabla 5. Combinación Dempster-Shafer (fase de clasificación).

Entrada:

$\mathbf{x}$: Objeto a clasificar.

$D_1, \dots, D_L$: Conjunto de clasificadores ya entrenados.

$DT_1, \dots, DT_c$: Conjunto de plantillas de decisión.

Salida:

ω: Etiqueta asignada al objeto $\mathbf{x}$.

1. Construir la $DP(\mathbf{x})$.
2. **Para cada** clase ω_j, $j = 1, \dots, c$ y **para cada** clasificador $i = 1, \dots, L$
   Calcular los siguientes grados de creencia
   $$b_j(D_i(\mathbf{x})) = \frac{\Phi_{j,i}(\mathbf{x}) \prod_{k \neq j}(1 - \Phi_{k,i}(\mathbf{x}))}{1 - \Phi_{j,i}(\mathbf{x})\left[1 - \prod_{k \neq j}(1 - \Phi_{k,i}(\mathbf{x}))\right]},$$
   donde
   $$\Phi_{j,i}(\mathbf{x}) = \frac{\left(1 + \left\| DT_j^i - D_i(\mathbf{x}) \right\|\right)^{-1}}{\sum_{k=1}^{c}\left(1 + \left\| DT_k^i - D_i(\mathbf{x}) \right\|\right)^{-1}}.$$
   Es la proximidad Φ entre DT_j^i y la salida del clasificador $D_i(\mathbf{x})$ para una entrada $\mathbf{x}$.
3. **Para cada** clase ω_j, $j = 1, \dots, c$
   Calcular el grado de soporte para la clase ω_j
   $$\mu_j(\mathbf{x}) = K \prod_{i=1}^{L} b_j(D_i(\mathbf{x})).$$
4. **Retornar** la etiqueta de la clase con mayor grado de soporte.

Sin embargo, Ani [40] destaca que el principal inconveniente de este método radica en la manera en que el vector de referencia es calculado, donde la media del vector de salida no tiene por qué ser la mejor opción. Partiendo de que los métodos existentes no estiman con precisión la evidencia de los clasificadores, Ani propone un nuevo método [40] el cual ajusta la evidencia de los distintos clasificadores minimizando el error cuadrático medio del conjunto de entrenamiento. Según su autor,

este método brinda buenos resultados en términos de desempeño general y en tasa de reducción de error siendo su principal inconveniente su alto costo computacional.

Recientemente Quost et al. [41] señalan que la regla de combinación de Dempster asume que los clasificadores deben ser independientes. Para solucionar tal limitante, los autores investigan el uso de otros operadores para combinar clasificadores no independientes, en especial reglas basadas en t-norma.

Como se pudo observar esta combinación depende de gran medida de cómo es calculada la evidencia de los clasificadores.

5 Métodos basados en selección de clasificadores

A diferencia de la fusión de clasificadores, la selección de clasificadores asume que cada clasificador posee una determinada habilidad en ciertas regiones locales del espacio de rasgos, por lo que estos esquemas tratan de estimar cual clasificador es el más apropiado para clasificar una determinada muestra.

Sea $\mathcal{D} = \{D_1, \dots, D_L\}$ el conjunto de clasificadores ya entrenados que conforman el esquema. Definamos a $\mathbb{F}^n$ como un espacio de los rasgos n-dimensional, el cual va a ser dividido en K regiones de selección $R_1, \dots, R_K$, también denominadas regiones de aptitud, con $K > 1$. Definamos además, una medida de disimilitud $d: \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{R}$, donde para dos objetos que pertenezcan a $\mathbb{F}^n$ el valor de la medida de disimilitud va a ser alto si los objetos no son parecidos y bajo en el caso contrario.

Las principales tareas que debe resolver la selección de clasificadores están asociadas a su fase de entrenamiento, donde deben ser encontradas las regiones de aptitud, estimadas las aptitudes de los clasificadores para cada región y elegir el modo de selección a emplear.

Los métodos de selección de clasificadores pueden ser divididos en dos grupos principales: Los que estiman dinámicamente la región de aptitud de forma local y los que pre-estiman las regiones de aptitud. Los métodos que conforman el primer grupo no hacen más que estimar la precisión local de los clasificadores que conforman el esquema usando un conjunto de datos, durante la fase de clasificación. El clasificador con mayor precisión es el seleccionado para etiquetar el objeto de entrada. Mientras que el segundo grupo estima de antemano el clasificador o clasificadores asociados a determinada región en la fase de entrenamiento. La figura 7 muestra una taxonomía de los métodos de selección de clasificadores [12].

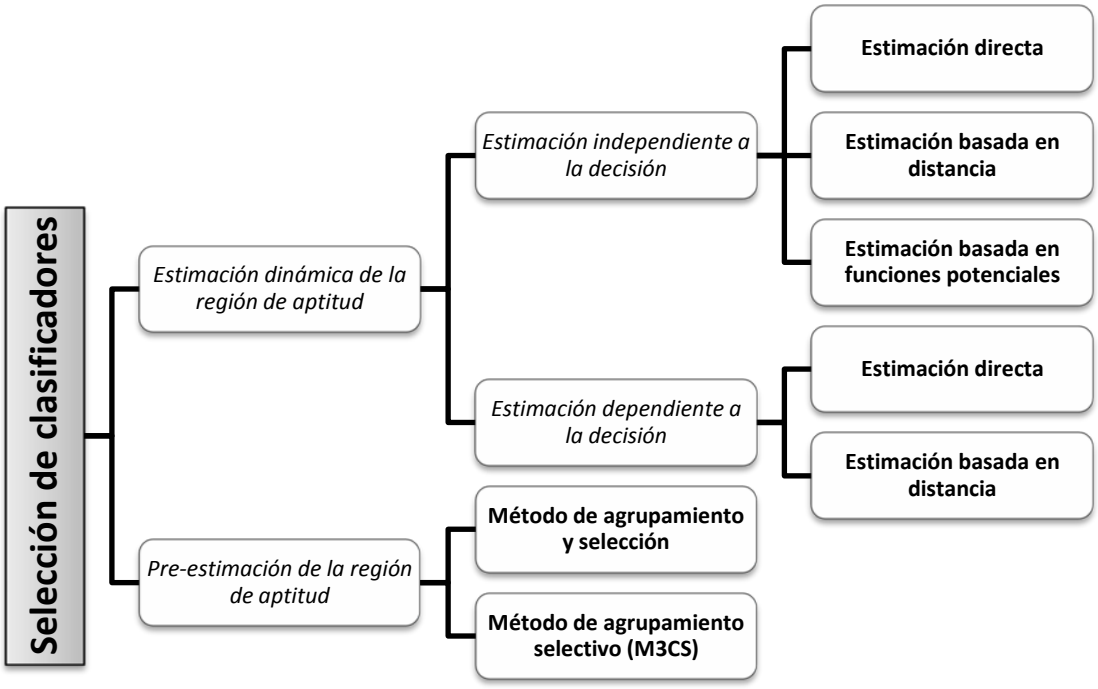


Fig. 7. Taxonomía de los métodos de selección de clasificadores.

5.1 Estimación dinámica de la región de aptitud local

Como vimos anteriormente, en esta estrategia el clasificador escogido para etiquetar el objeto de entrada **x** es seleccionado de los clasificadores del esquema con mayor precisión local. Estos estimadores de la región de aptitud pueden ser clasificados en dos grupos. El primero, denominado independientes a la decisión y el segundo dependientes a la decisión. El primero también conocido como enfoque *a priori*, estima la aptitud del clasificador basándose solo en la posición de **x** en el espacio de rasgos previo a determinar cuál etiqueta el clasificador le sugiere a **x**. Al contrario del segundo, donde la clase sugerida por el clasificador es tenida en cuenta. Este grupo también se le puede encontrar en la literatura como enfoque *a posteriori*.

5.1.1 Estimación independiente a la decisión

Estimación directa
Una de las formas más intuitivas de estimar la aptitud de los clasificadores es dado un conjunto de entrenamiento, buscar los K vecinos más cercanos a x respecto una medida de disimilitud d y calcular cuán preciso es cada clasificador en esos K objetos. El clasificador más preciso, será el encargado de etiquetar a x [42]. El valor del parámetro K es seleccionado por el usuario, o estimado antes de la fase operacional. Esta estimación también se le conoce como Precisión Local en Conjunto.

Estimación basada en disimilitud
Esta forma de estimación utiliza clasificadores que devuelven grados de soporte a cada clase, aprovechando esta información. La aptitud de un clasificador D_i es estimada como el promedio ponderado de los grados de soporte otorgados por el clasificador a las clases correctas en los K vecinos más cercanos de x respecto una medida de disimilitud d. Denotemos por $P_i(l(z_j)| z_j)$ como el grado de

soporte otorgado por el clasificador D_i a la clase correcta de $\mathbf{z}_j$ y a $N_\mathbf{x}$ como el conjunto de K vecinos más cercanos de **x**. La aptitud para el clasificador D_i para **x** quedaría estimada como

$$C(D_i, \mathbf{x}) = \frac{\sum_{\mathbf{z}_j \in N_\mathbf{x}} P_i(l(\mathbf{z}_j)|\, \mathbf{z}_j)(1/d(\mathbf{x}, \mathbf{z}_j))}{\sum_{\mathbf{z}_j \in N_\mathbf{x}} (1/d(\mathbf{x}, \mathbf{z}_j))}\ , \tag{5.1}$$

donde $d(\mathbf{x}, \mathbf{z}_j)$ es una medida de disimilitud entre **x** y el vecino cercano $\mathbf{z}_j \in N_\mathbf{x}$.

Estimación basada en funciones potenciales

Este enfoque también tiene en cuenta la disimilitud para estimar la aptitud de los clasificadores. Propuesto por Rastrigin y Erenstain [43], ellos consideran el espacio de rasgos como un campo y asumen que cada punto del conjunto de datos contribuye al "potencial" en **x**. El potencial de un clasificador D_i en **x** sería equivalente a su aptitud. La contribución individual de $\mathbf{z}_j \in \mathbf{Z}$ a $C(D_i, \mathbf{x})$ sería

$$\phi(\mathbf{x}, \mathbf{z}_j) = \frac{g_{ij}}{1 + \alpha_{ij}(d(\mathbf{x}, \mathbf{z}_j))^2}\ , \tag{5.2}$$

donde

$$g_{ij} = \begin{cases} 1, & si\ D_i \text{ reconoce correctamente } \mathbf{z}_j \in N_\mathbf{x} \\ -1, & en\ otro\ caso \end{cases}\ , \tag{5.3}$$

y α_{ij} es un parámetro que pondera la contribución de $\mathbf{z}_j$ y $d(\mathbf{x}, \mathbf{z}_j)$ es una medida de disimilitud entre **x** y $\mathbf{z}_j$. En el caso más trivial, α_{ij} pudiera ser una constante α fija para todo $i = 1, \ldots, L$ y $j = 1, \ldots, N$. Finalmente la aptitud del clasificador D_i quedaría como la sumatoria de las contribuciones individuales de los objetos que conforman el conjunto de datos, es decir

$$C(D_i, \mathbf{x}) = \sum_{\mathbf{z}_j \in \mathbf{Z}} \phi(\mathbf{x}, \mathbf{z}_j)\ . \tag{5.4}$$

Todavía no queda claro cuándo las dos versiones basadas en disimilitud, son mejores que la estimación directa, aunque es válido señalar que en las versiones basadas en disimilitud es mucho menos probable que ocurran empates, contrario a la estimación directa [12].

5.1.2 Estimación dependiente a la decisión

Estimación directa

Sea $s_i \in \Omega$ la etiqueta de clase asignada a **x** por el clasificador D_i y denotemos por $N_\mathbf{x}^{(s_i)}$al conjunto de K vecinos más cercanos a **x** de **Z** respecto una medida de disimilitud d, los cuales hayan sido etiquetados como s_i por el clasificador D_i. La aptitud del clasificador D_i para un determinado **x** es calculada como la proporción de elementos de $N_\mathbf{x}^{(s_i)}$ cuya verdadera clase es s_i [42]. Esta estimación también recibe el nombre de *Precisión Local de la Clase*. Según Woods [42] esta estimación directa es superior a la otra directa, independiente a la decisión.

Estimación basada en disimilitud

Denotemos por $P_i(s_i|\mathbf{z}_j)$ el estimado de la probabilidad (grado de soporte) otorgado por el clasificador D_i de que la clase verdadera de $\mathbf{z}_j$ sea s_i. La aptitud de D_i puede ser estimada por los $P_i(s_i|\mathbf{z}_j)$ promediados a través de la vecindad de puntos de **x** cuya clase verdadera es s_i. Usando la disimilitud hacia **x** como pesos, la aptitud del clasificador sobre **x** quedaría como

$$C(D_i, \mathbf{x}) = \frac{\sum_{\mathbf{z}_j} P_i(s_i|\mathbf{z}_j)(1/d(\mathbf{x}, \mathbf{z}_j))}{\sum_{\mathbf{z}_j}(1/d(\mathbf{x}, \mathbf{z}_j))}\ , \tag{5.5}$$

donde la sumatoria es realizada en $\mathbf{z}_j \in \mathbf{Z}$ tal que $l(\mathbf{z}_j) = s_i$ y $d(\mathbf{x}, \mathbf{z}_j)$ es una medida de disimilitud entre $\mathbf{x}$ y $\mathbf{z}_j$.

5.2 Pre-estimación de la región de aptitud

Uno de los principales inconvenientes que presenta la estimación dinámica de la región de aptitud es que es computacionalmente costosa en su fase de clasificación. En la mayoría de sus variantes es necesario encontrar los K vecinos más cercanos, y en el caso particular de la estimación dependiente a la decisión, son necesarios $K \times L$ vecinos. Con el objetivo de reducir la complejidad computacional en la fase de clasificación, la aptitud puede ser pre-estimada a través de regiones de aptitud en la fase de entrenamiento. Donde uno o un grupo de clasificadores son los responsables de clasificar los objetos pertenecientes a una o varias regiones. Dado un nuevo objeto a clasificar $\mathbf{x}$ se le determina la región de aptitud a la cual pertenece y finalmente se clasifica usando el clasificador o los clasificadores asignados a dicha región de aptitud. El problema a resolver de estos métodos entonces sería la identificación de las regiones de aptitud y sus clasificadores correspondientes.

Sea K el número de regiones de aptitud, el número de clasificadores L no tiene por qué ser igual al número de regiones. A la hora de decidir cuáles de los L clasificadores quedaran asociados a cada región $R_j, j = 1, \dots, K$, puede darse el caso de clasificadores que no sean asignados a ninguna región, incluso el clasificador con mayor precisión sobre todo el espacio de rasgos puede no ser tenido en cuenta en ninguna región. Por otro lado, pueden existir clasificadores asignados a más de una región de aptitud.

Para determinar las regiones de aptitud, el espacio de rasgos puede ser dividido en regiones regulares. Esta idea presenta como inconveniente que algunas regiones contengan pocos objetos del conjunto de datos provocando esto estimaciones poco confiables. Una manera de asegurarse que cada región contenga suficientes objetos, es mediante algoritmos de agrupamiento, donde cada agrupamiento quedaría asociado a una región de aptitud. A continuación expondremos algunos métodos que siguen esta idea.

5.2.1 Agrupamiento y selección

Este método fue presentado por Kuncheva [44] y su idea principal se basa en agrupar un conjunto de datos de entrenamiento para la selección de las regiones de aptitud. Finalmente el clasificador con mayor precisión en cada región es el seleccionado para clasificar los nuevos objetos correspondientes a dicha región. A continuación el método es mejor explicado a través de su pseudo-código.

Tabla 6. Método de agrupamiento y selección (fase de entrenamiento).

Entrada:
$D_1, \dots, D_L$: Conjunto de clasificadores ya entrenados.
$\mathbf{Z}$: Conjunto de datos etiquetados.
K: Número de regiones deseadas.
Salida:
$\mathbf{v}_1, \dots, \mathbf{v}_K$: Conjunto de centroides asociados a cada grupo.
$D_{i(1)}, \dots, D_{i(K)}$: Conjunto de clasificadores asociado a cada Región.

1. Sin tener en cuenta las etiquetas, agrupar $\mathbf{Z}$ en K grupos $C_1, \dots, C_K$.

2. Encontrar los centroides de cada grupo $\mathbf{v}_1, \ldots, \mathbf{v}_K$ como la media aritmética de los puntos que conforman cada grupo respectivamente.
3. **Para cada** grupo C_j (el cual define la región R_j)
   a. Calcular la precisión de clasificación de $D_1, \ldots, D_L$.
   b. Escoger el clasificador más competente sobre R_j y denotarlo como $D_{i(j)}$.
4. **Retornar** $\mathbf{v}_1, \ldots, \mathbf{v}_K$ y $D_{i(1)}, \ldots, D_{i(K)}$.

Tabla 7. Método de Agrupamiento y Selección (Fase de Clasificación)

Entrada:
x: Objeto a clasificar.
$\mathbf{v}_1, \ldots, \mathbf{v}_K$: Conjunto de centroides asociados a cada grupo.
$D_{i(1)}, \ldots, D_{i(K)}$: Conjunto de clasificadores asociado a cada Región.
Salida:
ω: Etiqueta asignada al objeto **x.**

1. Encontrar de entre $\mathbf{v}_1, \ldots, \mathbf{v}_K$ centroide $\mathbf{v}_j$ más cercano a **x.**
2. **Retornar** la etiqueta asignada al objeto **x** por el clasificador $D_{i(j)}$ correspondiente a la región R_j.

En [44], el método de agrupamiento empleado es el K-means. Además según su autora, este método por diseño garantiza al menos la misma precisión que el mejor clasificador que conforma la combinación.

El principal aporte que realiza este algoritmo es que al no dividir el espacio en regiones regulares, sino definir las regiones como los grupos formados en un conjunto de entrenamiento, permite que las regiones queden definidas por objetos que posean características afines. Esto permite que una vez identificado el clasificador que mejor clasifica a los objetos en una determinada región, cualquier nuevo objeto que se quiera clasificar que pertenezca a esa región, es más probable que el clasificador también lo clasifique bien dado que ese mismo clasificador mantiene una buena efectividad en objetos similares a él. Además, el agrupamiento garantiza que no existan regiones con pocos elementos lo cual dificultaría la apropiada elección del clasificador correspondiente a la región. Sin embargo hay que tener mucho cuidado con el conjunto de entrenamiento empleado, ya que un conjunto de datos que no generalice bien el problema puede conducir a la conformación de falsas regiones y por tanto afectar la selección del clasificador que vaya a clasificar un nuevo objeto.

5.2.2 Agrupamiento selectivo

Propuesto por Lui y Yuan en [45], este método realiza un proceso de agrupamiento más selectivo. En vez de realizar un único agrupamiento para todo el espacio de rasgos como el método anterior, ellos realizan un agrupamiento por cada clasificador. También denominado como M3CS, este método en su fase de entrenamiento descompone el espacio de rasgos en varias regiones agrupando por separado los conjuntos formados por los objetos clasificados correcta e incorrectamente por cada clasificador y se estima la precisión de cada clasificador para cada región. En la fase de clasificación, por cada clasificador, el objeto de entrada se asocia a la región a la cual pertenece y almacena la aptitud de dicho clasificador para esa región. Finalmente el clasificador con mayor aptitud es el seleccionado para etiquetar el objeto de entrada. El pseudo-código mostrado a continuación, expone mejor este método.

Tabla 8. Método de agrupamiento selectivo (fase de entrenamiento).

Entrada:

$\mathcal{D} = \{D_1, \dots, D_L\}$: Conjunto de clasificadores ya entrenados.

$\mathbf{Z}$: Conjunto de datos etiquetados.

α, β: Parámetros pre estimados.

Salida:

$m^1_{1,\dots,c+\bar{N}_1}, \dots, m^L_{1,\dots,c+\bar{N}_L}$: Conjunto de los centroides del agrupamiento para cada clasificador.

$a^1_{1,\dots,c+\bar{N}_1}, \dots, a^L_{1,\dots,c+\bar{N}_L}$: Conjunto con los valores de aptitud de cada clasificador para cada uno de los grupos de su agrupamiento.

1. **Para cada** clasificador $D_k \in \mathcal{D}$
   a. Clasificar los elementos de $\mathbf{Z}$ usando D_k .$\mathbf{Z}$ quedaría particionado en dos conjuntos disjuntos $\mathbf{Z}_c^k$ y $\mathbf{Z}_f^k$ donde el primero contiene los elementos correctamente clasificados por D_k y el segundo los incorrectos.
   b. Agrupar $\mathbf{Z}_f^k$.
      i. **Para cada** $\mathbf{z}_j \in \mathbf{Z}_f^k$
         1. Hallar la disimilitud $d(\mathbf{z}_j)$ entre $\mathbf{z}_j$ y su vecino más cercano en $\mathbf{Z}_f^k$.
      ii. Calcular la disimilitud promedio al vecino más cercano:

$$d_{avg} = \frac{1}{|\mathbf{Z}_f^k|} \sum_{j=1}^{|\mathbf{Z}_f^k|} d(\mathbf{z}_j)$$

      iii. Considerar los elementos de $\mathbf{Z}_f^k$ como vértices de un grafo y calcular la matriz de adyacencia A de acuerdo con la disimilitud entre dos elementos:

$$A(i,j) = \begin{cases} 1, & \text{si } d(\mathbf{z}_i, \mathbf{z}_j) \le a \cdot d_{avg} \\ 0, & \text{en otro caso} \end{cases}$$

      Donde $\mathbf{z}_i, \mathbf{z}_j \in \mathbf{Z}_f^k, 1 \le i, j \le |\mathbf{Z}_f^k|$ y α es un parámetro pre estimado.
      iv. Encontrar las componentes conexas del grafo formado. Estas serán denotadas como $C_r, r = 1, \dots, N_k$ donde N_k va a ser el número de componentes conexas. Cada componente conexa v ser vista como un grupo.
      v. Eliminar los grupos que contengan muy pocos elementos:

$$C = \left\{ C_r \,\middle|\, |C_r| > \frac{|\mathbf{Z}_f^k|}{\beta \cdot N_k} \right\}$$

      β es un parámetro pre estimado y denotaremos a $\overline{N_k}$ como el número de grupos restante.
      vi. Calcular el vector medio $m^k_{f,r}$ (centroide) para cada uno de los grupos restantes.
   c. Considerar cada clase como un grupo en el conjunto $\mathbf{Z}_c^k$ y calcular su vector medio (centroide):

$$m^k_{c,s} = \frac{1}{|\mathbf{Z}_c^k|} \sum_j \{\mathbf{z}_j \in \mathbf{Z}_c^k, D_k(\mathbf{z}_j) = \omega_s \wedge l(\mathbf{z}_j) = \omega_s\}$$

   d. Asignar cada uno de los elementos de $\mathbf{Z}$ en uno de los $c + \overline{N_k}$ grupos de acuerdo con la disimilitud entre el elemento y los

centroides, conformando un nuevo agrupamiento.

e. Estimar la aptitud a_i^k del clasificador D_k en cada grupo.

2. **Retornar** $m^1_{1,\dots,c+\bar{N}_1}, \dots, m^L_{1,\dots,c+\bar{N}_L}$ y $a^1_{1,\dots,c+\bar{N}_1}, \dots, a^L_{1,\dots,c+\bar{N}_L}$.

Tabla 9. Método de agrupamiento selectivo (fase de clasificación).

Entrada:

x: Objeto a clasificar.

$m^1_{1,\dots,c+\bar{N}_1}, \dots, m^L_{1,\dots,c+\bar{N}_L}$: Conjunto de los centroides del agrupamiento para cada clasificador.

$a^1_{1,\dots,c+\bar{N}_1}, \dots, a^L_{1,\dots,c+\bar{N}_L}$: Conjunto con los valores de aptitud de cada clasificador para cada uno de los grupos de su agrupamiento.

$\mathcal{D} = \{D_1, \dots, D_L\}$: Conjunto de clasificadores ya entrenados.

Salida:

ω: Etiqueta asignada al objeto **x.**

1. **Para cada** clasificador $D_k \in \mathcal{D}$
   a. Asignar **x** a uno de los $c + \bar{N}_k$ grupos que conforman el agrupamiento asociado a D_k cuya disimilitud al centroide sea menor.
   b. Recuperar el valor de aptitud $a_{\mathbf{x}}^k$ del clasificador D_k en la región en la cual pertenece **x.**
2. **Retornar** la etiqueta devuelta por el clasificador D_k. Donde D_k es el clasificador seleccionado que posea mayor aptitud:

$$k = \arg\max_{i=1,\dots,L} a_{\mathbf{x}}^k$$

Este método se diferencia con el anterior en la manera en que son estimadas las regiones de aptitud. Los autores reportaron resultados superiores al método propuesto por Kuncheva. Además destacan que los parámetros, en especial α, afectan el resultado, por lo que recomiendan prestar atención a la hora de determinar dichos parámetros una vez que se decida poner en práctica el método propuesto.

6 Métodos basados en generación de ensamblados

El rendimiento de un sistema multiclasificador no solo depende del esquema de combinación empleado, sino también depende de la independencia (diversidad) existente entre los clasificadores que conforman el ensamblado. Comparado el rendimiento del mejor clasificador del ensamblado con la combinación de un grupo de clasificadores independientes, se logra un mayor rendimiento, en especial en el voto mayoritario. En la práctica es muy difícil lograr un grupo de clasificadores totalmente independientes, aunque existen diversas maneras de lograr la mayor diversidad posible mediante técnicas como el entrenamiento con diferentes conjuntos de datos o rasgos, variando la estructura del clasificador o incluso los parámetros de entrenamiento, etc.

En la literatura existen diversas maneras de incorporar diversidad en un ensamblado de clasificadores. Sharkey [46] fue unos de los primeros en sugerir una taxonomía para los métodos de incorporar diversidad, en el caso especial, de un ensamblado de redes neuronales. Sharkey propone el logro de diversidad variando cuatro aspectos fundamentalmente: los pesos iniciales, el conjunto de entrenamiento usado, la arquitectura de la red neuronal y el algoritmo de entrenamiento empleado.

Posteriormente Brown [47] propone una taxonomía diferente. El señala que para que un ensamblado de clasificadores sea diverso, los clasificadores que lo integren deben estar situados en diferentes puntos en un espacio de hipótesis. Basado en esto, categoriza los métodos de generación de ensamblados en los siguientes grupos: en los que varían el punto inicial dentro del espacio de hipótesis, los que varían el conjunto de hipótesis accesibles por los miembros del ensamblado y los que varían la manera en que cada miembro recorre el espacio de hipótesis.

Más reciente Rocatch [48] propone una nueva taxonomía conformada por las siguientes categorías donde estas no son mutualmente exclusivas:

Manipulación del aprendizaje: En esta categoría son agrupados los métodos que ganan diversidad manipulando el aprendizaje de los clasificadores. De manera más específica, cada miembro del ensamblado es entrenado con un método de aprendizaje manipulado de manera diferente. Esta categoría está dividida en 3 subcategorías:

- Manipulación de los parámetros de aprendizaje
- Variación del punto inicial dentro del espacio de hipótesis
- Variación en la manera en que cada miembro recorre el espacio de hipótesis

Manipulando las muestras de entrenamiento: Esta categoría agrupa a los métodos que varían el conjunto de entrenamiento con el cual el clasificador individual es entrenado. Es decir, que cada miembro del ensamblado es entrenado con un conjunto de entrenamiento distinto (usualmente una variación o subconjunto de conjunto original). Aquí los métodos se dividen en dos subgrupos:

- Los que realizan re-muestreo
- Los que crean el conjunto

Manipulando las etiquetas de clase: Aquí son agrupados los métodos que manipulan las etiquetas de clase del conjunto de entrenamiento. Es decir, en vez de entrenar un solo clasificador complicado, varios clasificadores con diferentes representaciones (usualmente más simples) de las etiquetas de clase son entrenados.

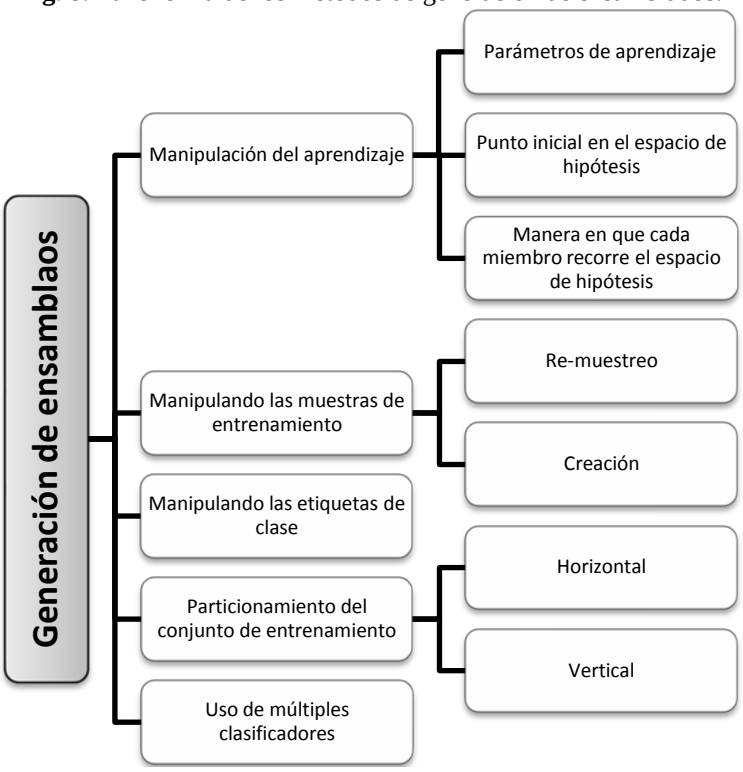
Particionamiento del conjunto de entrenamiento: como su nombre lo indica, los métodos agrupados en esta categoría particionan el conjunto de entrenamiento para que cada miembro del ensamblado sea entrenado en una partición diferente. Existen dos maneras de particionar el conjunto de entrenamiento:

- Horizontal: Aquí el conjunto original es particionado en diferentes conjuntos los cuales tienen el mismo número de rasgos que el original, conteniendo cada uno un subconjunto de instancias del conjunto original.
- Vertical: En esta forma de partición, el conjunto original es particionado en varios conjuntos los cuales poseen el mismo número de instancias que le original, conteniendo cada uno un subconjunto de rasgos del conjunto original.

Uso de múltiples clasificadores: en esta categoría, la diversidad es obtenida conformado un ensamblado el cual contenga distintos modelos de aprendizaje.

La figura 8 muestra gráficamente la taxonomía descrita y a continuación abordaremos algunos métodos de generación de ensamblados.

Fig. 8. Taxonomía de los métodos de generación de ensamblados.



6.1 Bagging

El término de *Bagging* fue introducido por Breiman [49] como acrónimo de *b*ootstrap *agg*regat*ing*. La idea de este método es bastante simple: el ensamblado es conformado por clasificadores entrenados con una muestra de instancias tomadas con reemplazos del conjunto de entrenamiento. El resultado de los clasificadores es finalmente combinado usando voto mayoritario.

La diversidad necesaria para hacer que el ensamblado funcione se logra usando diferentes conjuntos de entrenamiento para cada clasificador. Idealmente, los conjuntos de entrenamiento deberían ser generados de manera aleatoria a partir de la distribución del problema. Sin embargo en la práctica, usualmente solo se dispone de un conjunto de entrenamiento, por lo que el proceso de generación de L conjuntos de entrenamientos aleatorios debe ser imitado. Para eso, se realiza un muestreo con remplazo (bootstrap [50]) al conjunto de entrenamiento con el objetivo de crear un nuevo conjunto de entrenamiento del mismo tamaño que el conjunto original. Nótese que algunas de las instancias del conjunto de entrenamiento inicial pueden aparecer en más de una ocasión en los nuevos conjuntos generados, e incluso no pertenecer a ninguno.

Para hacer uso de las variaciones de los nuevos conjuntos de entrenamiento generados, los clasificadores empleados deben ser inestables, es decir que pequeños cambios en el conjunto de entrenamiento provoquen grandes variaciones en la salida del clasificador. De lo contrario, el ensamblado resultante sería una colección de clasificadores casi idénticos, lo cual no llevaría a grandes mejoras del rendimiento con respecto al mejor clasificador individual. Desde el punto de vista estadístico, un grupo de clasificadores inestables poseen una elevada varianza del error de predicción y la combinación de varios clasificadores generados de la manera descrita pueden reducir dicha varianza.

Las tablas 10 y 11 muestran el pseudo-código del método expuesto.

Tabla 10. Método de Bagging (fase de entrenamiento).

Entrada:
Z: Conjunto de datos etiquetados.
L: Número de clasificadores a entrenar
Salida:
$\mathcal{D} = \{D_1, \dots, D_L\}$: Conjunto de clasificadores ya entrenados.

1. Inicializar los parámetros
   a. $\mathcal{D} = \emptyset$
2. **Para** $k = 1, \dots, L$
   a. Tomar una muestra con remplazo (bootstrap) S_k de **Z**.
   b. Entrenar el clasificador D_k usando el conjunto S_k.
   c. $\mathcal{D} = \mathcal{D} \cup D_k$.
3. **Retornar** $\mathcal{D}$

Tabla 11. Método de Bagging (fase de clasificación).

Entrada:
$\mathcal{D} = \{D_1, \dots, D_L\}$: Conjunto de clasificadores ya entrenados.
x: Objeto a clasificar.
Salida:
ω: Etiqueta asignada al objeto **x.**

1. Clasificar **x** con $D_1, \dots, D_L$.
2. **Retornar** la etiqueta de la clase con mayor número de votos obtenidos.

Teóricamente el funcionamiento de este método está respaldado en la hipótesis de que un conjunto de clasificadores independientes combinados mediante el voto mayoritario mejora el rendimiento comparado con el mejor clasificador del ensamblado. Sin embargo, a pesar de que los clasificadores generados no son completamente independientes desde el punto de vista estadístico, la práctica ha demostrado que en la mayoría de los casos el método de bagging logra una mejor precisión que el mejor clasificador de ensamblado. Otra ventaja que presenta este método es que es paralelizable en sus dos etapas (clasificación y entrenamiento).

6.2 Método del sub-espacio aleatorio

El método del sub-espacio aleatorio genera múltiples clasificadores en paralelo al realizar un muestreo de rasgos en vez de los objetos de entrenamiento. Propuesto por Ho [51], en este método cada clasificador individual es entrenado con un subconjunto de rasgos seleccionados de manera aleatoria usando todos los objetos del conjunto de entrenamiento.

Sea $\mathbf{z}_j = (z_{j1}, \dots, z_{jr}),\ \mathbf{z}_j \in \mathbf{Z}$ un vector r-dimesional de entrenamiento, donde $\mathbf{Z}$ es un conjunto de entrenamiento conformado por r rasgos y n muestras. En el método del sub-espacio aleatorio, $s < r$ rasgos son seleccionados de manera aleatoria del espacio de rasgos r-dimesional para conformar un nuevo conjunto de entrenamiento $\tilde{\mathbf{Z}}$ s-dimesional de tamaño n. El nuevo conjunto $\tilde{\mathbf{Z}}$ está compuesto por los vectores del conjunto original $\mathbf{Z}$ representados por los k rasgos seleccionados anteriormente.

Posteriormente el clasificador es entrenado usando el nuevo conjunto de datos $\tilde{\mathbf{Z}}$, repitiendo el procedimiento para los L clasificadores que van a conformar el ensamblado. Finalmente la decisión final es tomada usando voto mayoritario con las L respuestas. A continuación, la tabla 12 muestra el pseudo-código de la fase de entrenamiento del método. La fase de clasificación es equivalente a la descrita en la tabla 11 para el método de Bagging.

Tabla 12. Método del sub-espacio aleatorio (fase de entrenamiento).

Entrada:
- $\mathbf{Z}$: Conjunto de datos etiquetados.
- L: Número de clasificadores a entrenar

Salida:
- $\mathcal{D} = \{D_1, \ldots, D_L\}$: Conjunto de clasificadores ya entrenados.

1. **Para** $k = 1, \ldots, L$
   a. Conformar un nuevo conjunto de entrenamiento $\tilde{\mathbf{Z}}_k$ seleccionado aleatoriamente un sub-espacio s-dimesional a partir de $\mathbf{Z}$.
   b. Entrenar el clasificador D_k usando el conjunto $\tilde{\mathbf{Z}}_k$.
   c. $\mathcal{D} = \mathcal{D} \cup D_k$.

2. **Retornar** $\mathcal{D}$

A pesar que este método fue inicialmente concebido para su uso con árboles de decisión como clasificadores base, es aplicable con otros clasificadores. Cuando el número de objetos de entrenamiento es relativamente pequeño en comparación con la dimensionalidad de los datos, construyendo los clasificadores usando este método es posible resolver el problema del pequeño tamaño de la muestra. Otras ventajas que posee este método es el hecho que al trabajar con un espacio de rasgos de menor dimensionalidad, el entrenamiento de los clasificadores es realizado de forma más rápida. También el uso de este método es ventajoso cuando se tiene un conjunto de entrenamiento con un número grande de rasgos y no tanto cuando el número de rasgos es reducido. Además resulta provechoso el uso del método del sub-espacio aleatorio cuando existe cierta redundancia en el conjunto de entrenamiento, en especial en los rasgos.

6.3 Bosques aleatorios

El concepto de Bosques Aleatorios fue introducido por Breiman en [52] para definir una variante del Bagging la cual utiliza árboles de decisión como clasificadores base. Para que un ensamblado de árboles de decisión cumpla con el concepto de Bosques Aleatorios definido por Breiman, el ensamblado debe ser formado con árboles construidos mediante vectores aleatorios independientes e idénticamente distribuidos.

Esto se logra de la siguiente manera. Cada árbol es construido usando una muestra bootstrap diferente, la cual contenga instancias aleatorias con reemplazo del conjunto de datos de entrenamiento original. Además, se realiza una selección aleatoria de rasgos. En cada nodo de un árbol de decisión, s rasgos son seleccionados aleatoriamente de los r existentes, seleccionando el mejor de los s como corte del nodo. Los árboles son construidos alcanzado la mayor altura posible sin realizar podas. Una nueva instancia es clasificada con la clase que mayor cantidad de votos haya obtenido de los L árboles que conformen el ensamblado. El pseudo-código de la fase de entrenamiento puede verse en la tabla 13, mientras que el de la fase de clasificación es equivalente con el de la tabla 11.

Tabla 13. Bosques aleatorios (Fase de entrenamiento).

Entrada:
Z: Conjunto de datos etiquetados.
L: Número de clasificadores a entrenar
Salida:
$\mathcal{D} = \{D_1, \dots, D_L\}$: Conjunto de clasificadores ya entrenados.

1. **Para** $k = 1, \dots, L$
   a. Formar un nuevo conjunto $\tilde{\mathbf{Z}}_k$ tomando n muestras aleatorias con reemplazo (bootstrap) de las n que conforman **Z.**
   b. Construir el árbol de decisión D_k usando el nuevo conjunto $\tilde{\mathbf{Z}}_k$ hasta mu máxima extensión, sin realizar podas.
      i. Seleccionar un número $s \ll r$.
      ii. Para cada nodo del árbol, seleccionar r rasgos aleatorios usando el mejor de los s como corte del árbol.
   c. $\mathcal{D} = \mathcal{D} \cup D_k$.

2. **Retornar** $\mathcal{D}$

El uso de la aleatoriedad convierte a los bosques aleatorios en un clasificador muy preciso en distintos dominios. En [52] Breiman muestra que el índice de error de los bosques aleatorios depende de dos factores fundamentalmente. El primero, la correlación entre dos árboles del ensamblado. El aumento de dicha correlación propicia el aumento del índice de error. Y el segundo, la efectividad de cada árbol individual. Un árbol con bajo índice de error es un clasificador efectivo. Aumentando la efectividad de los árboles del ensamblado, disminuye el índice de error del bosque aleatorio.

Además, mientras el Bagging aumenta la estabilidad del árbol de decisión original, la selección aleatoria de rasgos incrementa la robustez sobre la presencia de rasgos redundantes, haciéndolo viable en conjuntos de entrenamiento con grandes cantidades de rasgos. Sin embargo, según Deng [53]cuando son usados rasgos nominales, los bosques aleatorios tienden a favorecer aquellos rasgos cuya cardinalidad del conjunto de valores admisibles sea mayor, provocando que este método no sea tan viable para manejar rasgos nominales.

A pesar que el método está concebido para árboles de decisión, Prinzie [54, 55] se basa en la hipótesis de que la incorporación de aleatoriedad mejora la generación del ensamblado. Prinzie utiliza la idea de los bosques aleatorios de tomar muestras bootstrap junto con la selección aleatoria de rasgos en otros clasificadores como el clasificador bayesiano y multinomial logit. Los nuevos métodos propuestos, Random Naive Bayes y Random Multinomial Logit, mejoran los resultados de clasificación con relación a sus respectivos clasificadores individuales.

6.4 Boosting

Se define como Boosting [56] al problema general de crear una regla de predicción muy precisa al combinar un conjunto de clasificadores moderadamente imprecisos (clasificadores débiles). La idea general del boosting es de generar un ensamblado $\mathcal{D}$ de clasificadores de manera incremental, añadiendo un clasificador a la vez. El conjunto de entrenamiento usado para cada miembro del ensamblado es escogido basado en el desempeño de los clasificadores de las iteraciones anteriores. Los objetos del conjunto de entrenamiento que fueron incorrectamente clasificados en las iteraciones anteriores, son escogidos con mayor frecuencia que los objetos correctamente clasificados. Por lo que el Boosting trata

de generar nuevos clasificadores que clasifiquen mejor los objetos en los que los clasificadores bases más se equivoquen.

El algoritmo de Boosting más popular y empleado es el AdaBoost [56] y su nombre proviene ADAptive BOOSTing. Existen dos maneras de implementar el AdaBoost: con re-ponderamiento y con re-muestreo. La idea principal de este algoritmo es la de asignar pesos a cada objeto del conjunto de entrenamiento. La implementación que vamos a exponer más adelante es la basada en re-muestreo. En la implementación por re-ponderamiento se asume que los clasificadores bases pueden directamente usar las probabilidades en **Z** como pesos. La variante por re-muestreo asume inicialmente que todos los pesos tienen el mismo valor, sin embargo, en el transcurso de las iteraciones el valor de los pesos de los objetos mal clasificados es aumentado, mientras que el de los correctamente clasificados disminuye. Esto provoca como consecuencia, que los clasificadores débiles de las iteraciones anteriores estén enfocados en aprender de los objetos difíciles de clasificar, produciendo una serie de clasificadores que se complementan unos a otros. El algoritmo original de AdaBoost fue diseñado para la clasificación de problemas binarios (dos clases). Sin embargo, existen extensiones del mismo para problemas multi-clases, siendo los algoritmos más conocidos el AdaBoost.M1 y el AdaBoost.M2. Las tablas 14 y 15 muestran el pseudo-código del Adaboost.M1 tanto para sus fases de entrenamiento como de clasificación.

Tabla 14. AdaBoost.M1 (fase de entrenamiento).

Entrada:

Z: Conjunto de datos etiquetados.

L: Número de clasificadores a entrenar

Salida:

$\mathcal{D} = \{D_1, \dots, D_L\}$: Conjunto de clasificadores ya entrenados.

$\beta_1, \dots, \beta_L$

1. Inicializar parámetros
   a. $\mathcal{D} = \emptyset$
   b. $\mathbf{w}^1 = [w_1, \dots, w_N], w_j^1 \in [0,1], \sum_{j=1}^{N} w_j^1 = 1$ (usualmente $w_j^1 = 1/N$)

2. **Para** $k = 1, \dots, L$
   a. Tomar una muestra S_k de **Z** usando distribución $\mathbf{w}^k$.
   b. Entrenar el clasificador D_k usando S_k.
   c. Calcular el error ponderado del ensamblado en el paso k:

$$\epsilon_k = \sum_{j=1}^{N} w_j^k l_k^j$$

   (Donde $l_k^j = 1$ si D_k mal clasifica $\mathbf{z}_j$ y $l_k^j = 0$ en otro caso.)
   d. **Si** $\epsilon_k = 0$ o $\epsilon_k \geq 0.5$, ignorar D_k, reinicializar los pesos w_j^k en 1/N y continuar con el próximo clasificador.
   e. **Sino** calcular

$$\beta_k = \frac{\epsilon_k}{1 - \epsilon_k}, \epsilon_k \in (0, 0.5)$$

   f. Actualizar los pesos individuales

$$w_j^{k+1} = \frac{w_j^k \beta_k^{(1-l_k^j)}}{\sum_{i=1}^{N} w_i^k \beta_k^{(1-l_k^i)}}, j = 1, \dots, N$$

   g. $\mathcal{D} = \mathcal{D} \cup D_k$.

3. **Retornar** $\mathcal{D}$ y $\beta_1, \dots, \beta_L$.

Tabla 15. AdaBoost.M1 (Fase de clasificación).

Entrada:

$\mathcal{D} = \{D_1, \dots, D_L\}$: Conjunto de clasificadores ya entrenados.

x: Objeto a clasificar.

$\beta_1, \dots, \beta_L$

Salida:

ω: Etiqueta asignada al objeto **x.**

1. Calcular el grado de soporte para cada clase $\omega_t, t = 1, \dots, c$

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

2. **Retornar** la etiqueta de la clase con mayor valor de soporte.

Una de las explicaciones del éxito del AdaBoost, según Kuncheva [12], proviene de la propiedad del algoritmo de llevar el error de entrenamiento del ensamblado rápidamente a cero, prácticamente en las primeras iteraciones. Según Rokach [48], mejora el rendimiento de un ensamblado por dos razones fundamentalmente. Primeramente porque genera un clasificador final cuyo error sobre el conjunto de entrenamiento es menor al combinar muchas hipótesis cuyos errores pueden ser grandes. Y segundo, porque produce un clasificador combinado cuya varianza es significativamente inferior que la producida por los clasificadores débiles. Sin embargo, el Bosting conduce en ocasiones al deterioro del rendimiento del ensamblado. Según Quinlan [57], la principal razón por la cual el Boosting en ocasiones falla es el sobre ajuste, dado que un gran número de iteraciones puede generar un ensamblado muy complejo, siendo este mucho menos preciso que un clasificador individual.

6.4.1 Variantes del Boosting

arc-x4

Breiman propuso un algoritmo de Boosting el cual denominó arc-x4 [58]. Su diferencia con el AdaBoost es primeramente, que el peso del objeto $\mathbf{z}_j$ en la iteración k se calcula como la proporción de veces en la cual los $k-1$ clasificadores anteriores hayan mal clasificado $\mathbf{z}_j$. Y segundo, la decisión final se toma usando pluralidad a diferencia del voto mayoritario ponderado que utiliza el AdaBoost.

AveBoost

Este método propuesto por Orza [59] representa la distribución sobre los objetos de entrenamiento como un vector, y lo construye de tal manera que sea ortogonal con respecto al vector de objetos mal clasificados por el clasificador base de la iteración anterior. La idea es hacer que los errores del próximo clasificador base no estén correlacionados con aquellos del clasificador de la iteración anterior. Los resultados experimentales muestran una significativa mejora sobre al AdaBoost.

MultiBoost

MultiBoost es una extensión del AdaBoost propuesta por Webb [60], la cual no es más que una combinación del AdaBoost con el método de Wagging. Este método es capaz de combinar el alto sesgo y la reducción de varianza del AdaBoosting junto a la superior reducción de varianza del Wagging. Usando C4.5 como clasificador base, este método ha probado generar ensamblados con menor error que los generados por AdaBoost y Wagging por separado.

Real AdaBoost
Real AdaBoost es una versión generalizada del AdaBoost propuesta por Friedman [61]. Esta generalización combina el estimado de la probabilidad de clase de los clasificadores mediante el ajuste de un modelo de regresión logístico aditivo, de forma iterativa. Esta variante reduce el costo computacional, además de propiciar un mejor desempeño, especialmente usando árboles de decisión. Además, puede proveer descripciones interpretables de las reglas de decisión agregadas.

Gradient Boosting
Desarrollado por Friedman [62], este método genera el ensamblado ajustando secuencialmente los parámetros de los clasificadores bases a un pseudo-residual actual mediante mínimos cuadrados en cada iteración. El pseudo-residual no es más que el gradiente de la función de pérdida que es minimizada con respecto a los valores del modelo en cada objeto de entrenamiento evaluado en la iteración actual. Para mejorar la precisión, aumentar la robustez y reducir el costo computacional, en cada iteración una sub-muestra del conjunto de entrenamiento es seleccionada (sin reemplazo) y usada para ajustar el clasificador base.

AdaBoostKL y AdaBoostNorm2
AdaBoost en muy pocas ocasiones sufre de sobreajuste, sin embargo en conjunto de datos con alta presencia de ruido, el sobreajuste suele ocurrir con mayor frecuencia. Con vista a esto, Sun [63] propone una estrategia la cual penaliza el sesgo de la distribución de los datos en el proceso de aprendizaje para prevenir que varios ejemplos difíciles arruinen los límites de decisión. El autor utiliza dos funciones de penalidad basadas en la divergencia de Kullback-Leibler y la norma l2, originado dos nuevos métodos: el AdaBoostKL y el AdaBoostNorm2. Estas dos variaciones alcanzan mejores resultados en conjuntos ruidosos que el AdaBoost.

SABoost
Propuesto por Tsao y Chang [64], donde sus autores se refieren al boosting como un procedimiento de aproximación estocástica. Basados en este punto de vista desarrollaron este método el cual es similar al AdaBoost con la salvedad en la manera en los que los pesos de los miembros son calculados.

P-AdaBoost
Este algoritmo es una versión distribuida del AdaBoost. Propuesto por Merler [65], este método en vez de actualizar los pesos asociados a una instancia de manera secuencial, trabaja en dos fases. En su primera fase el algoritmo AdaBoost es corrido en su forma tradicional de manera secuencial, por un número limitado de iteraciones. En su segunda fase, los clasificadores son entrenados en paralelo usando los pesos calculados en la primera fase. Este método produce aproximaciones a los modelos generados por el AdaBoost convencional, los cuales pueden ser fácil y eficientemente distribuidos sobre una red o nodos de cómputo.

Quadratic Boosting
Phama y Smeuldersb [66] presentaron una estrategia para mejorar el AdaBoosting mediante una combinación cuadrática de los clasificadores bases, realizando una optimización iterativa indirecta. La idea consiste en construir un clasificador intermedio, el cual opere sobre la combinación de los términos lineales y cuadráticos. Primeramente, un clasificador es entrenado con un conjunto de datos donde las etiquetas de clase fueron aleatorizadas. Posteriormente el clasificador de entrada es llamado repetidamente con una actualización sistemática de las etiquetas de clase del conjunto de entrenamiento en cada iteración.

Local Boosting
Este método fue propuesto como una nueva variante del AbaBoost basada en re-muestreo, desarrollada por Zhang y Zhang [67]. Un error local es calculado para cada instancia de entrenamiento, el cual es usado para actualizar la probabilidad de que esa instancia es seleccionada para el conjunto de

entrenamiento de la siguiente iteración. La clasificación de una nueva instancia está basada en su similaridad con cada instancia de entrenamiento.

7 Diversidad en los ensamblados de clasificadores

Resulta intuitivo pensar que el resultado de combinar un grupo de clasificadores idénticos no va a ser mejor que el resultado de uno solo de sus miembros. Al contrario, resultaría más conveniente si combináramos un grupo de clasificadores diferentes entre sí, dado que al menos uno de ellos debe dar la respuesta correcta cuando el resto falle. Dicha diferencia, conocida principalmente como *diversidad*, también se le conoce como ortogonalidad, independencia, dependencia negativa o complementariedad. A pesar de que no existe una definición formal de lo que es intuitivamente percibido como diversidad, no al menos en el vocabulario de la Ciencia de la Computación, es ampliamente aceptado por la comunidad científica el hecho de que la existencia de diversidad en un grupo de clasificadores base es una condición necesaria para la mejora del desempeño de un ensamblado de clasificadores [68]. De acuerdo con Ho [69], un ensamblado de clasificadores diversificados conduce a errores no correlacionados, que a su vez mejoran la precisión de clasificación.

Comprender y cuantificar la diversidad que existente en un ensamblado de clasificadores es un aspecto importante en la combinación de clasificadores. En la literatura existen diferentes medidas usadas para tal propósito, cuyo objetivo es cuantificar la dependencia existente entre clasificadores. Estas medidas para cuantificar la diversidad, pueden ser categorizadas en dos tipos [70], medidas de pareja (pairwise) y medidas de grupo (non pairwise). La figura 9 muestra la taxonomía descrita. A continuación veremos algunas de las medidas más usadas.

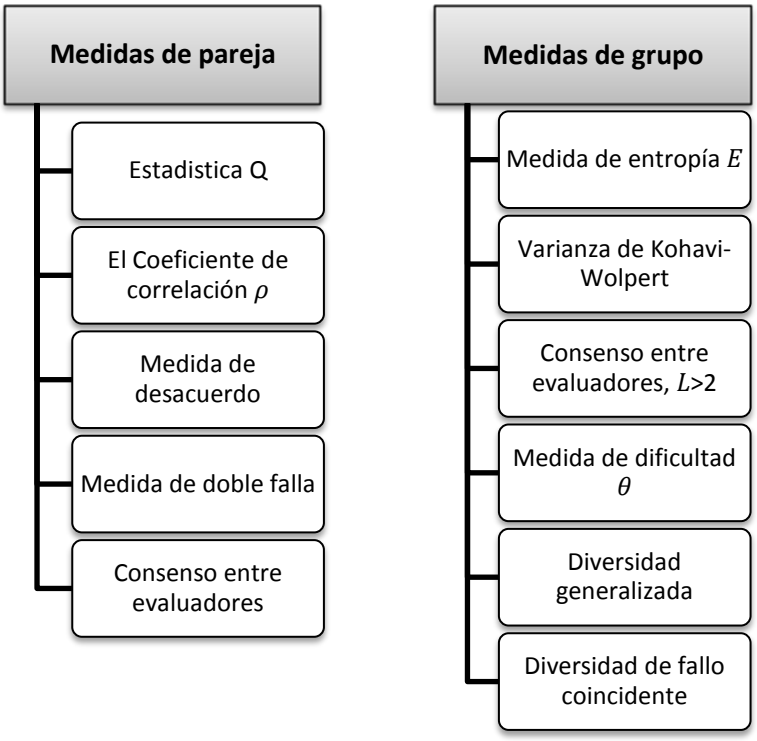


Fig. 9. Taxonomía de las medidas utilizadas para cuantificar la diversidad en un ensamblado de clasificadores.

7.1 Medidas de diversidad de pareja

Una medida de diversidad de pareja es usada para cuantificar la diversidad existente entre un par de clasificadores. Para medir la diversidad de un ensamblado de L clasificadores suele promediarse los valores de las $L(L-1)/2$ parejas de clasificadores. Algunas de estas medidas son derivadas de la rama de la estadística.

7.1.1 La estadística Q

Sea $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ un conjunto de datos etiquetados. Representamos la salida del clasificador D_i como un vector binario N-dimensional $\mathbf{y}_i = [y_{1,i}, \ldots, y_{N,i}]^T$, tal que $y_{j,i} = 1$ si D_i reconoce correctamente $\mathbf{z}_j$, y 0 en el caso contrario para todo $i = 1, \ldots, L$.

Entonces, la estadística Q definida por Yule [71] para dos clasificadores D_i y D_k quedaría como

$$Q_{i,k} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}}\ , \tag{7.1}$$

donde N^{ab} es el número de elementos $\mathbf{z}_j$ de $\mathbf{Z}$ los cuales $y_{j,i} = a$ y $y_{j,k} = b$.

Para un par de clasificadores estadísticamente independientes, su valor de $Q_{i,k}$ va a ser 0. En general, el valor que Q va a oscilar entre -1 y 1. Aquellos clasificadores que tienden a reconocer los mismos objetos correctamente tendrán un valor positivo de Q, y aquellos que comentan errores en diferentes objetos poseerán un valor negativo de Q.

7.1.2 El coeficiente de correlación $\boldsymbol{\rho}$

La correlación entre dos resultados de clasificadores binarios $\mathbf{y}_i$ y $\mathbf{y}_k$ quedaría como

$$\rho_{i,k} = \frac{N^{11}N^{00} - N^{01}N^{10}}{\sqrt{(N^{11}+N^{10})(N^{01}+N^{00})(N^{11}+N^{01})(N^{10}+N^{00})}} \ . \tag{7.2}$$

El coeficiente de correlación también puede ser calculado para pares de clasificadores que devuelven grados de pertenencia a cada clase. Para cada par de clasificadores van a existir c coeficientes de correlación, uno por cada clase. La medida final sería el promedio de los valores asociados a cada clase [12]. Para cualquier par de clasificadores, sus valores de Q y ρ tendrán el mismo signo y puede probarse que $|\rho| \leq |Q|$ [70].

7.1.3 Medida de desacuerdo

La medida de desacuerdo es probablemente la medida de diversidad más intuitiva para un par de clasificadores. Introducida por Skalak [72], esta medida no es más que la probabilidad de que dos clasificadores discrepen en sus respuestas. Para clasificadores binarios, su valor quedaría dado por

$$D_{i,k} = \frac{N^{01} + N^{10}}{N^{11} + N^{01} + N^{10} + N^{00}} . \tag{7.3}$$

7.1.4 Medida de doble falla

Esta medida no es más que la proporción de elementos mal clasificados por ambos clasificadores. Introducida por Giacinto y Roli [73], su valor queda definido para un par de clasificadores binarios como

$$DF_{i,k} = \frac{N^{00}}{N^{11} + N^{01} + N^{10} + N^{00}} . \tag{7.4}$$

Ruta y Gabrys [74] definen a esta medida como una medida no-simétrica. Esto quiere decir que si intercambiamos los unos con los ceros en los resultados de los clasificadores, el valor de la medida no va a ser el mismo. Esta medida está basada en el concepto de que es más importante tener en cuenta los errores simultáneos que cuando ambos clasificadores sean correctos.

7.1.5 Consenso entre evaluadores

Desarrollada en la estadística como una medida de confiabilidad entre evaluadores. Conocida en inglés como Interrater Agreement, esta medida puede ser usada cuando diferentes evaluadores (clasificadores en este caso) evalúan sujetos ($\mathbf{z}_j$ en nuestro caso) para medir el nivel de consenso [75].

Para c clases, se conforma una matriz M de tamaño $c \times c$ donde la entrada $m_{k,s}$ es la proporción del conjunto de datos, en la cual el clasificador D_i etiqueta como ω_k y el clasificador D_j etiqueta como ω_s. El consenso entre D_i y D_j viene dado por

$$\mathrm{k_{i,j}} = \frac{\sum_k \mathrm{m_{k,s}} - \mathrm{ABC}}{1 - \mathrm{ABC}} \ , \tag{7.5}$$

donde $\sum_k m_{k,s}$ es el consenso observado entre clasificadores y ABC es el Agreement By Chance, que viene dado por

$$ABC = \sum_k \left(\sum_s m_{k,s} \right) \left(\sum_s m_{s,k} \right) . \tag{7.6}$$

Valores bajos de k representan mayor desacuerdo, por ende mayor diversidad. Para clasificadores binarios la medida quedaría como

$$k_{i,j} = \frac{2(N^{11}N^{00} - N^{01}N^{10})}{(N^{11} + N^{10})(N^{01} + N^{00}) + (N^{11} + N^{01})(N^{10} + N^{00})}. \tag{7.7}$$

Hasta ahora todas las medidas vistas cuantifican la diversidad entre un par de clasificadores. Si se deseara estimar la diversidad de un ensamblado $\mathcal{D}$ de L clasificadores usando cualquier medida de pareja $M_{i,k}$, la medida de diversidad sobre el ensamblado quedaría como

$$Div(\mathcal{D}) = \frac{2}{L(L-1)} \sum_{i=1}^{L-1} \sum_{k=i+1}^{L} M_{i,k} \ . \tag{7.8}$$

A continuación veremos un conjunto de medidas las cuales calculan la diversidad del ensamblado usando todos sus miembros en conjunto.

7.2 Medidas de diversidad de grupo

Las medidas de diversidad catalogadas como medias de grupo, tienen en consideración todos los clasificadores en conjunto para cuantificar el valor de diversidad del ensamblado. A continuación presentaremos algunas de las más conocidas

7.2.1 Medida de entropía $\boldsymbol{E}$

La mayor diversidad que intuitivamente puede ser lograda para un $\mathbf{z}_j \in \mathbf{Z}$ en particular, es cuando $\lfloor L/2 \rfloor$ de los votos en $\mathbf{y}_j$ poseen el mismo valor (0 o 1) y el restante $L - \lfloor L/2 \rfloor$ poseen el valor alterno. Si todos los votos tuviesen el mismo valor, no habría descuerdo por lo que los clasificadores no serían diversos.

Una posible medida de diversidad basada en este concepto fue introducida por Cunningham y Carney [76], la cual quedaría como

$$E = \frac{1}{N} \frac{2}{L-1} \sum_{j=1}^{N} \min \left\{ \left(\sum_{i=1}^{L} y_{j,i} \right), \left(L - \sum_{i=1}^{L} y_{j,i} \right) \right\}. \tag{7.9}$$

El valor de E varía entre 0 y 1, donde 0 representa no diferencia y 1 la mayor diversidad posible.

7.2.2 La varianza de Kohavi-Wolpert

La varianza de Kohavi-Wolpert fue inicialmente propuesta por Kohavi y Wolpert [77]. Esta medida es originada de la descomposición de la varianza del sesgo del error de un clasificador. La expresión original de la variabilidad de una etiqueta de clase predicha y para una muestra $\mathbf{x}$ es

$$varianza_x = \frac{1}{2} \left(1 - \sum_{i=1}^{c} P(y = \omega_i | \mathbf{x})^2 \right), \tag{7.10}$$

donde c es el número de clases. Kuncheva y Whitaker presentaron en [70] una modificación para medir la diversidad de un ensamblado compuesto por clasificadores binarios, quedando la medida de diversidad como

$$KW = \frac{1}{NL^2}\sum_{j=1}^{N} Y(\mathbf{z}_j)(L - Y(\mathbf{z}_j)), \tag{7.11}$$

donde $Y(\mathbf{z}_j)$ es el número de clasificadores que reconocieron correctamente $\mathbf{z}_j$, es decir $\sum_{i=1}^{L} y_{i,j}$.

Vale señalar que esta medida difiere de la medida de desacuerdo promediada D_{avg} en un coeficiente

$$KW = \frac{L-1}{2L} D_{avg}\ , \tag{7.12}$$

la demostración de equivalencia puede ser encontrada en [70].

7.2.3 Medida de consenso entre evaluadores, para $\boldsymbol{L > 2}$

Denotemos con $\overline{p}$ la precisión de clasificación individual promediada, esta sería

$$\overline{p} = \frac{1}{NL}\sum_{j=1}^{N}\sum_{i=1}^{L} y_{j,i}\ . \tag{7.13}$$

La medida de consenso entre evaluadores [75] quedaría

$$k = 1 - \frac{\frac{1}{L}\sum_{j=1}^{N} Y(\mathbf{z}_j)(L - Y(\mathbf{z}_j))}{N(L-1)\overline{p}\,(1-\overline{p}\,)}. \tag{7.14}$$

Esta medida queda relacionada con la varianza de Kohavi-Wolpert (KW) y con la medida de desacuerdo promediada ($\mathrm{D_{avg}}$) de la siguiente manera

$$k = 1 - \frac{L}{(L-1)\overline{p}\,(1-\overline{p}\,)} KW = 1 - \frac{1}{2\overline{p}(1-\overline{p})} D_{avg}\,. \tag{7.15}$$

Es también valido señalar que la medida de consenso entre evaluadores para $\mathrm{L} > 2$ (7.14) no es obtenida promediando el consenso entre evaluadores para parejas (7.7).

7.2.4 Medida de dificultad $\boldsymbol{\theta}$

La idea para esta medida viene del estudio realizado por Hansen y Salamon [78]. Ellos definen una variable aleatoria discreta $V, V_i = (L - l_i)/L$ y representa la proporción de clasificadores en $\mathcal{D}$ que clasifican correctamente una muestra $\mathbf{x}$ extraída aleatoriamente del conjunto de datos. Para estimar la función de probabilidad de X, los L clasificadores en $\mathcal{D}$ necesitan ser corridos en el conjunto de datos $\mathbf{Z}$.

Basado en esto, ellos definen la *dificultad* θ como

$$\theta = Var(V)\,. \tag{7.16}$$

Para conveniencia, θ suele ser escalada linealmente en el intervalo [0,1] tomando como $\mathrm{p}(1-\mathrm{p})$ como el mayor valor posible, donde p es la precisión individual de cada clasificador. La diversidad del

ensamblado aumenta con el decremento del valor de la medida de dificultad. Idealmente $\theta = 0$, pero este es un escenario poco realista.

La intuición de esta medida puede ser explicada de la siguiente manera: Un ensamblado de clasificadores diverso tiene un valor pequeño de medida de dificultad, dado que cada muestra de entrenamiento puede al menos ser clasificada correctamente por una proporción de todos los clasificadores base, lo cual es más probable con una baja varianza de V.

7.2.5 Diversidad generalizada

Esta medida fue propuesta por Partridge y Krzanowski [79]. Sea Y una variable aleatoria que representa la proporción de clasificadores que clasificaron incorrectamente una muestra $\mathbf{x} \in \mathbb{R}^n$ extraída aleatoriamente del conjunto de datos. Denotemos por p_i la probabilidad de que $Y = i/L$ y $p(i)$ la probabilidad de que i clasificadores extraídos de manera aleatoria fallen en clasificar correctamente un objeto **x** extraído aleatoriamente. Supongamos que dos clasificadores son tomados de forma aleatoria del ensamblado $\mathcal{D}$, Partridge y Krzanowski exponen en su trabajo que la máxima diversidad es lograda cuando el uno de los dos clasificadores se equivoca en clasificar un objeto y el otro lo clasifica correctamente. En este caso la probabilidad de equivocarse los dos clasificadores es $p(2) = 0$. Por otra parte argumentan que la mínima diversidad se lograría cuando el fallo de un clasificador es siempre acompañado con el fallo del otro, entonces la probabilidad de que los dos clasificadores fallen es la misma que la probabilidad de que un clasificador escogido de forma aleatoria falle. Usando

$$p(1) = \sum_{i=i}^{L} \frac{i}{L} p_i \ ; \ p(2) = \sum_{i=i}^{L} \frac{i}{L} \frac{(i-1)}{(L-1)} p_i \ , \tag{7.17}$$

la medida de diversidad generalizada (GD) quedaría definida como

$$GD = 1 - \frac{p(2)}{p(1)} \ . \tag{7.18}$$

El valor de GD varía entre 0 y 1, siendo 0 la menor diversidad cuando $p(2) = p(1)$ y 1 la mayor diversidad cuando $p(2) = 0$.

7.2.6 Diversidad de fallo coincidente

La medida de diversidad de fallo coincidente, también conocida en inglés como Coincident Failure Diversity, es una modificación de la medida de diversidad generalizada también propuesta por Partridge y Krzanowski [79].

$$CFD = \begin{cases} 0, & p_0 = 1 \\ \dfrac{1}{1-p_0} \displaystyle\sum_{i=i}^{L} \frac{L-i}{L-1} p_i \ , & p_0 < 1 \end{cases} . \tag{7.19}$$

Esta medida está diseñada tal que tenga un valor mínimo 0 cuando todos los clasificadores siempre clasifiquen correctamente o cuando todos los clasificadores lo mismo clasifiquen correcta o incorrectamente al mismo tiempo. Su máximo valor 1 es alcanzado cuando todos los errores de clasificación son únicos, es decir cuando al menos un clasificador va a clasificar incorrectamente cualquier objeto aleatorio.

8 Conclusiones

En este trabajo se presenta una primera aproximación al estado del arte de la combinación de clasificadores supervisados donde se realiza un análisis crítico de los principales esquemas de combinación existentes. Su objetivo fundamental es proveerle a la comunidad científica una actualización de los diversos métodos y sistemas desarrollados en los últimos años, además de servir como un punto de partida a aquellos investigadores y especialistas que deseen adentrarse en la temática.

El tema de la combinación de clasificadores supervisados, es un área con gran auge en la actualidad debido al alto impacto que ha tenido en numerosas aplicaciones y en diversos campos dentro del reconocimiento de patrones. Diferentes herramientas han sido desarrolladas para la combinación de clasificadores supervisados. Sin embargo, en este trabajo no fue posible abarcarlas todas y de las que sí fueron vistas, no todas poseen la misma profundidad. Además existen elementos que han sido referenciados por no considerárseles esenciales para la comprensión del trabajo.

Los sistemas multiclasificadores fueron agrupados en cuatro grupos: los basados en combinación de clasificadores, basados en selección de clasificadores, basados en generación de ensamblados de clasificadores y los sistemas híbridos.

En cuanto a los métodos basados en combinación de clasificadores, estos fueron divididos en dependencia del tipo de salida de los clasificadores que combinan. Se pudo observar que los métodos más usados que combinan etiquetas de clase, como el voto mayoritario y la combinación Bayesiana, necesitan como precondición independencia entre los resultados de los clasificadores para alcanzar teóricamente una mejora en la precisión. Sin embargo, aquellos métodos que no asumen independencia entre los resultados, como BKS y Wernecke, necesitan de grandes conjuntos de entrenamiento para modelar el problema que se quiere resolver. Además estos métodos no son computacionalmente viables para problemas que utilicen muchas clases con muchos clasificadores, debido a la gran cantidad de memoria que requieren. Otra observación realizada fue que el enfoque que combina clasificadores que devuelven ranking de etiquetas es al que menor énfasis se le ha prestado. Esto se debe al hecho que son muy pocos los clasificadores que devuelven ranking de etiquetas, donde la gran mayoría devuelven la etiqueta de la clase correcta o los grados de pertenencia a cada clase. Sin embargo, existen clasificadores importantes, como pueden ser clasificadores biométricos, que solo devuelven un ranking de etiquetas. Esto hace que sea también de interés combinar esos resultados, para alcanzar una mejor precisión.

En cuanto a los métodos basados en selección de clasificadores fueron abordados dos grupos, los que realizan una estimación dinámica de la región de aptitud local y los que pre-estiman las regiones de aptitud. El principal inconveniente en los del primer grupo es que son computacionalmente costosos en su fase de clasificación. En el caso de los del segundo grupo, estos reducen el costo computacional en la fase de clasificación al pre-estimar las regiones de aptitud en su fase de entrenamiento. Sin embargo, los resultados van a depender de la manera en que son halladas las regiones de aptitud junto con la correcta estimación del clasificador más competente para cada región.

Con respecto a los métodos basados en generación de ensamblados de clasificadores, uno de sus principales aportes radica en construir un conjunto de clasificadores, independientes en cuanto a sus resultados, desde la misma fase de entrenamiento de los clasificadores. De esta manera, sacan provecho de esquemas de combinación, como el voto mayoritario, que garantizan teóricamente una mejora de la precisión al combinar clasificadores independientes.

Otra manera en que pueden ser divididos los sistemas multiclasificadores es en entrenables y no entrenables. Los no entrenables combinan a los clasificadores tal y como vienen y no necesitan de ningún paso extra. Son más generales y pueden emplearse en cualquier problema que se quiera resolver. En el caso de los entrenables, estos necesitan de un paso extra el cual sirve lo mismo para construir el propio esquema de combinación que para ajustar ciertos parámetros. Los entrenables tienen como ventaja que permiten ajustarse a un determinado problema en particular que se quiere resolver y así obtener mejores resultados en el mismo. La desventaja radica en que en ocasiones se puede producir

sobre entrenamiento y por tanto una pobre generalización del problema. En esencia, los entrenables dependen en gran medida de la correcta selección del conjunto de entrenamiento a emplear.

Una observación realizada en general fue el hecho de que la gran mayoría de las herramientas dentro de los sistemas multiclasificadores presuponen que los datos son numéricos, es decir los objetos son representados a través de vectores en $\mathbb{R}^n$. Muchas de estas herramientas pueden ser generalizadas a otros espacios fácilmente, sin embargo otras requieren de un estudio un poco más profundo, en especial cuando se trabaja con datos mezclados e incompletos.

Como se pudo ver, a pesar de lo mucho que se ha avanzado en esta área, todavía existe mucho por hacer. Por lo tanto, es nuestro objetivo proponernos abordar algunas de las observaciones planteadas anteriormente, lo mismo desarrollando nuevas herramientas que mejorando las existentes.

Referencias bibliográficas

1. Jain, A.K. and A. Ross, *Multibiometric systems.* Commun. ACM, 2004. **47**(1): p. 34-40.
2. Ross, A. and A. Jain, *Information fusion in biometrics.* Pattern Recognition Letters, 2003. **24**(13): p. 2115-2125.
3. Leigh, W., R. Purvis, and J.M. Ragusa, *Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in romantic decision support.* Decision Support Systems, 2002. **32**(4): p. 361-377.
4. Mangiameli, P., D. West, and R. Rampal, *Model selection for medical diagnosis decision support systems.* Decision Support Systems, 2004. **36**(3): p. 247-259.
5. Merkwirth, C., et al., *Ensemble Methods for Classification in Cheminformatics.* Journal of Chemical Information and Computer Sciences, 2004. **44**(6): p. 1971-1978.
6. Rokach, L., *Mining manufacturing data using genetic algorithm-based feature set decomposition.* Int. J. Intell. Syst. Technol. Appl., 2008. **4**(1/2): p. 57-78.
7. Bruzzone, L., R. Cossu, and G. Vernazza, *Detection of land-cover transitions by combining multidate classifiers.* Pattern Recognition Letters, 2004. **25**(13): p. 1491-1500.
8. Menahem, E., et al., *Improving malware detection by applying multi-inducer ensemble.* Computational Statistics & Data Analysis, 2009. **53**(4): p. 1483-1494.
9. Rokach, L., R. Romano, and O. Maimon, *Negation recognition in medical narrative reports.* Information Retrieval, 2008. **11**(6): p. 499-538.
10. Tao, D., et al., *Asymmetric Bagging and Random Subspace for Support Vector Machines-Based Relevance Feedback in Image Retrieval.* IEEE Trans. Pattern Anal. Mach. Intell., 2006. **28**(7): p. 1088-1099.
11. Xu, L., A. Krzyzak, and C. Suen, *Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition.* IEEE Transactions on Systems, Man and Cybernetics, 1992. **22**(3): p. 418-435.
12. Kuncheva, L.I., *Combining Pattern Classifiers: Methods and Algorithms.*2004: Wiley Interscience.
13. Lu, Y., *Knowledge integration in a multiple classifier system*. Applied Intelligence, 1996. **6**(2): p. 75-86.
14. Ho, T.H., J.J. Hull, and S.N. Srihari. *Decision Combination in Multiple Classifier System*. in *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1994.
15. Shapley, L. and B. Grofman, *Optimizing group judgmental accuracy in the presence of interdependencies.* Public Choice, 1984. **43**(3): p. 329-343.
16. Lam, L. and C.Y. Suen, *Application of majority voting to pattern recognition: An analysis of its behavior and performance.* IEEE Transactions on Systems, Man, and Cybernetics, 1997. **27**(5): p. 553–568.
17. Kuncheva, L.I., et al., *Limits on the majority vote accuracy in classifier fusion.* Pattern Analysis and Applications, 2003. **6**(1): p. 22-31.
18. Brown, G. and L. Kuncheva, *"Good" and "Bad" Diversity in Majority Vote Ensembles*, in *Multiple Classifier Systems*, N. El Gayar, J. Kittler, and F. Roli, Editors. 2010, Springer Berlin / Heidelberg. p. 124-133.
19. Lam, L. and C.Y. Suen, *Optimal combinations of pattern classifiers.* Pattern Recognition Letters, 1995. **16**(9): p. 945-954.
20. Huang, Y.S. and C.Y. Suen, *A method of combining multiple experts for the recognition of unconstrained handwritten numerals.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995. **17**(1): p. 90 - 94

21. Raudys, Š. and F. Roli, *The Behavior Knowledge Space Fusion Method: Analysis of Generalization Error and Strategies for Performance Improvement*, in *Multiple Classifier Systems*, T. Windeatt and F. Roli, Editors. 2003, Springer Berlin / Heidelberg. p. 160-160.
22. Cecotti, H. and A. Belaïd, *Hierarchical Behavior Knowledge Space*, in *Multiple Classifier Systems*, M. Haindl, J. Kittler, and F. Roli, Editors. 2007, Springer Berlin / Heidelberg. p. 421-430.
23. Souvannavong, F. and B. Huet. *Continuous Behaviour Knowledge Space For Semantic Indexing of Video Content*. in *2006 9th International Conference on Information Fusion*. 2006. IEEE.
24. Wernecke, K.D., *A coupling procedure for the discrimination of mixed data.* Biometrics, 1992. **48**(2): p. 497-506.
25. Ho, T.K., J.J. Hull, and S.N. Srihari. *On Multiple Classifier Systems for Pattern Recognition*. in *IEEE Trans. Pattern Anal. Machine Intell*. 1992.
26. Hashem, S., *Optimal Linear Combinations of Neural Networks.* Neural Networks, 1997. **10**(4): p. 599-614.
27. Ueda, N., *Optimal Linear Combination of Neural Networks for Improving Classification Performance.* IEEE Trans. Pattern Anal. Mach. Intell., 2000. **22**(2): p. 207-215.
28. Cho, S.-B., *Pattern recognition with neural networks combined by genetic algorithm.* Fuzzy Sets and Systems, 1999. **103**(2): p. 339-347.
29. Kuncheva, L.I., *An application of OWA operators to the aggregation of multiple classification decisions*, in *The ordered weighted averaging operators*1997, Kluwer Academic Publishers. p. 330-343.
30. Kuncheva, L.I., *Combining classifiers: Soft computing solutions*, in *Pattern Recognition: From Classical to Modern Approaches*, S.K. Pal and A. Pal, Editors. 2001, World Scientific. p. 427-452.
31. Cho, S.-B. and J.H. Kim, *Combining multiple neural networks by fuzzy integral for robust classification.* IEEE Transactions on Systems, Man and Cybernetics, 1995. **25**(2): p. 380 - 384
32. Cho, S.-B. and J.H. Kim, *Multiple network fusion using fuzzy logic.* IEEE Transactions on Neural Networks, 1995. **6**(2): p. 497 - 501
33. Bulacio, P., et al., *A selection approach for scalable fuzzy integral combination.* Information Fusion, 2010. **11**(2): p. 208-213.
34. Huang, Y.S. and C.Y. Suen, *A method of combining multiple classifiers-a neural network approach*, in *12th International Conference on Pattern Recognition*1994: Jerusalem , Israel p. 473–475.
35. Kuncheva, L., *Decision templates for multiple classifier fusion: an experimental comparison.* Pattern Recognition, 2001. **34**(2): p. 299-314.
36. Kuncheva, L.I., *Using measures of similarity and inclusion for multiple classifier fusion by decision templates.* Fuzzy Sets and Systems, 2001. **122**(3): p. 401-407.
37. Shafer, G., *A mathematical theory of evidence*1976: Princeton University Press.
38. Mandler, E. and J. Schurmann, *Combining the classification results of independent classifiers based on the dempster–shafer theory of evidence.* Pattern recognition and artificial intelligence, 1988: p. 381–393.
39. Rogova, G., *Combining the results of several neural network classifiers.* Neural Networks, 1994. **7**(5): p. 777-781.
40. Ani, A. and M. Deriche, *A New Technique for Combining Multiple Classifiers using The Dempster-Shafer Theory of Evidence*. Journal of Artificial Intelligence Research, 2002. **17**(1): p. 333-361.
41. Quost, B., M.-H. Masson, and T. Denoeux, *Classifier fusion in the Dempster-Shafer framework using optimized t-norm based combination rules.* International Journal of Approximate Reasoning, 2011. **52**(3): p. 353-374.
42. Woods, K., K. Bowyer, and P. Kegelmeyer, *Combination of Multiple Classifiers Using Local Accuracy Estimates*, in *Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*1996, IEEE Computer Society. p. 391.
43. Rastrigin, L.A. and R.H. Erenstein, *Method of Collective Recognition.*1981, Moscow: Energoizdat.
44. Kuncheva, L.I. *Clustering-and-selection model for classifier combination*. in *Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on*. 2000.
45. Liu, R. and B. Yuan, *Multiple classifiers combination by clustering and selection.* Information Fusion, 2001. **2**(3): p. 163-168.
46. Sharkey, A.J.C., *Combining Artificial Neural Nets : Ensemble and Modular Multi-Net Systems (Perspectives in Neural Computing)*1999: Springer Verlag.
47. Brown, G., et al., *Diversity creation methods: a survey and categorisation.* Information Fusion, 2005. **6**(1): p. 5-20.
48. Rokach, L., *Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography.* Computational Statistics & Data Analysis, 2009. **53**(12): p. 4046-4072.

49. Breiman, L., *Bagging predictors.* Machine Learning, 1996. **24**(2): p. 123-140.
50. Efron, B. and R.J. Tibshirani, *An Introduction to the Bootstrap* 1994: Chapman and Hall/CRC.
51. Ho, T., *The random subspace method for constructing decision forests.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998. **20**(8): p. 832-844.
52. Breiman, L., *Random Forests.* Machine Learning, 2001. **45**(1): p. 5-32.
53. Deng, H., G. Runger, and E. Tuv. *Bias of Importance Measures for Multi-valued Attributes and Solutions*. in *21st International Conference on Articial Neural Networks*. 2011.
54. Prinzie, A. and D.V.d. Poel, *Random Forests for multiclass classification: Random MultiNomial Logit.* Expert Syst. Appl., 2008. **34**(3): p. 1721-1732.
55. Prinzie, A. and D. Van den Poel, *Random Multiclass Classification: Generalizing Random Forests to Random MNL and Random NB*, in *Database and Expert Systems Applications*, R. Wagner, N. Revell, and G. Pernul, Editors. 2007, Springer Berlin / Heidelberg. p. 349-358.
56. Freund, Y. and R. Schapire, *A desicion-theoretic generalization of on-line learning and an application to boosting*, in *Computational Learning Theory*, P. Vitányi, Editor 1995, Springer Berlin / Heidelberg. p. 23-37.
57. Quinlan, J.R. *Bagging, Boosting, and C4.5*. in *In Proceedings of the Thirteenth National Conference on Artificial Intelligence*. 1996.
58. Breiman, L., *Arcing Classifiers.* The Annals of Statistics, 1998. **26**(3): p. 801-824
59. Oza, N., *Boosting with Averaged Weight Vectors*, in *Multiple Classifier Systems*, T. Windeatt and F. Roli, Editors. 2003, Springer Berlin / Heidelberg. p. 159-159.
60. Webb, G.I., *MultiBoosting: A Technique for Combining Boosting and Wagging.* Machine Learning, 2000. **40**(2): p. 159-196.
61. Friedman, J., T. Hastie, and R. Tibshirani, *Additive logistic regression : A statistical view of boosting.* The Annals of Statistics, 2000. **28**(2).
62. Friedman, J.H., *Stochastic gradient boosting.* Computational Statistics & Data Analysis, 2002. **38**(4): p. 367-378.
63. Sun, Y., S. Todorovic, and J. Li, *Reducing the overfitting of adaboost by controlling its data distribution skewness.* International Journal of Pattern Recognition and Artificial Intelligence, 2006. **20**(7): p. 1093-1116.
64. Tsao, C.A. and Y.-c.I. Chang, *A stochastic approximation view of boosting.* Computational Statistics & Data Analysis, 2007. **52**(1): p. 325-334.
65. Merler, S., B. Caprile, and C. Furlanello, *Parallelizing AdaBoost by weights dynamics.* Computational Statistics & Data Analysis, 2007. **51**(5): p. 2487-2498.
66. Pham, T.V. and A.W.M. Smeulders, *Quadratic boosting.* Pattern Recognition, 2008. **41**(1): p. 331-341.
67. Zhang, C.-X. and J.-S. Zhang, *A local boosting algorithm for solving classification problems.* Computational Statistics & Data Analysis, 2008. **52**(4): p. 1928-1941.
68. Windeatt, T., *Diversity measures for multiple classifier system analysis and design.* Information Fusion, 2005. **6**(1): p. 21-36.
69. Ho, T.K., *Data Complexity Analysis for Classifier Combination*, in *Multiple Classifier Systems*, J. Kittler and F. Roli, Editors. 2001, Springer Berlin / Heidelberg. p. 53-67.
70. Kuncheva, L.I. and C.J. Whitaker, *Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy.* Machine Learning, 2003. **51**(2): p. 181-207.
71. Yule, G.U., *On the Association of Attributes in Statistics: With Illustrations from the Material of the Childhood Society, &c.* Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 1900. **194**(252-261): p. 257-319.
72. Skalak, D.B., *The Sources of Increased Accuracy for Two Proposed Boosting Algorithms*, 1996.
73. Giacinto, G. and F. Roli, *Design of effective neural network ensembles for image classification purposes.* Image and Vision Computing, 2001. **19**(9-10): p. 699-707.
74. Ruta, D. and B. Gabrys, *Analysis of the Correlation Between Majority Voting Error and the Diversity Measures in Multiple Classifier Systems*, in *Soft Computing and Intelligent Systems for Industry: Proceedings and Scientific Program : Fourth International ICSC Symposium 2001*2001, ICSC-NAISO Academic Press: Paisley, Scotland. p. 50.
75. Fleiss, J.L., *Statistical Methods for Rates and Proportions.*1981: John Wiley & Sons.
76. Cunningham, P. and J. Carney, *Diversity versus Quality in Classification Ensembles Based on Feature Selection*, in *Machine Learning: ECML 2000*, R. López de Mántaras and E. Plaza, Editors. 2000, Springer Berlin / Heidelberg. p. 109-116.
77. Kohavi, R. and D.H. Wolpert. *Bias Plus Variance Decomposition for Zero-One Loss Functions*. in *Machine Learning: Proceedings of the Thirteenth International Conference*. 1996.

78. Hansen, L.K. and P. Salamon, *Neural Network Ensembles.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990. **12**: p. 993-1001.
79. Partridge, D. and W. Krzanowski, *Software diversity: practical statistics for its measurement and exploitation.* Information and Software Technology, 1997. **39**(10): p. 707-717.

RT_048, abril 2012
Aprobado por el Consejo Científico CENATAV

Editor: Lic. Lucía González Bayona
Diseño de Portada: Di. Alejandro Pérez Abraham
RNPS No. 2142
ISSN 2072-6287
Indicaciones para los Autores:
Seguir la plantilla que aparece en www.cenatav.co.cu
C E N A T A V
7ma. No. 21812 e/218 y 222, Rpto. Siboney, Playa;
La Habana. Cuba. C.P. 12200
Impreso en Cuba