Combinación de clasificadores para bioinformática

Isis Bonet, Abdel Rodríguez, María M. García y Ricardo Grau

Centro de Estudios de Informática, Universidad Central "Marta Abreu" de Las Villas, Cuba

ibonetc@gmail.com

Resumen. Dentro de la bioinformática existen muchos problemas de clasificación, que resultan difícil de solucionar usando técnicas de inteligencia artificial por la diversidad de patrones de las bases de datos. En este trabajo se desarrolla un multiclasificador que combina clasificadores con el objetivo de mejorar los resultados de clasificación en bases de datos de bioinformática. Se basa en usar diferentes métodos de aprendizaje automatizado que funcionan como un método de agrupamiento para dividir la base a partir de los casos que son bien clasificados por cada método. El sistema aprende a decidir, mediante un metaclasificador, cuál o cuáles son los mejores clasificadores para un caso determinado. Se usaron once bases de datos internacionales para comparar el modelo propuesto con los multiclasificadores más conocidos en la literatura. Se usan pruebas estadísticas que demuestran que los resultados obtenidos por el nuevo multiclasificador son significativamente superiores a los obtenidos con otros modelos.

Palabras clave: clasificación, reconocimiento de patrones, aprendizaje, multiclasificador.

Combining Classifiers for Bioinformatics

Abstract. There are several classification problems in Bioinformatics which are difficult to solve using artificial intelligence techniques because of the diversity of patterns in datasets. In this paper, an ensemble of classifiers is developed to improve the accuracy of classification in bioinformatics datasets. This model is based on the use of different machine learning methods, and it forms clusters to divide the dataset taking into account the performance of the base methods. By means of a meta-classifier, the system learns to decide which classifiers are the best for a given case. In order to compare the new model with some well-known multi-classifiers, eleven international databases are used. It is demonstrated by statistical tests that results of our model are significantly better than those obtained with previous models.

Keywords. Model classification, pattern recognition, learning, multi-classifiers.

1 Introducción

La aparición de nuevas y eficientes técnicas experimentales en los últimos años, han revolucionado las ciencias de la vida. Con la secuenciación del ADN se ha producido un crecimiento exponencial de las descripciones lineales de proteínas y moléculas de ADN y ARN. Este incremento de la cantidad de datos biológicos ha generado dos problemas: el almacenamiento y manejo eficiente de la información y la extracción de información útil [14]. Investigar la esencia de la vida en estos grandes volúmenes de datos biológicos ha llevado a la necesaria unión de varias ciencias como la biología, la matemática y la ciencia de la computación; en un área de conocimientos reciente, pero con creciente auge: la bioinformática. Cada día se precisa más de herramientas computacionales para: clasificación de secuencias, detección de semejanzas, separación de las regiones codificantes de las no codificantes en secuencias de ADN, predicción de estructura molecular y la reconstrucción de la historia evolutiva. Esto, al igual que el descubrimiento de nuevos fármacos y terapias, es fundamental para el entendimiento de la vida y la evolución [2].

En cualquiera de los dominios de la bioinformática se puede encontrar problemas de predicción o clasificación. En este trabajo nos centramos en este tipo de problemas.

Aunque existen muchas técnicas para poder resolver problemas de clasificación, no existe todavía un clasificador por excelencia. Para un problema determinado es difícil seleccionar cual

será el clasificador que logre encontrar una mejor frontera de decisión para separar las clases. En la búsqueda de mejores métodos de clasificación aparece una tendencia a combinar varios clasificadores en el mismo problema. Esta tendencia está basada en la forma en que nos comportamos cuando nos enfrentamos a un problema de gran importancia en nuestra vida. Generalmente buscamos varias opiniones, luego pesamos estas opiniones y las combinamos de alguna manera para llegar a una conclusión final, que supuestamente debe ser más informada. Este proceso de consultar expertos es muy natural y precisamente constituye la base de estos algoritmos que son conocidos como sistemas de combinación de clasificadores o multiclasificadores: utilizar varios expertos (clasificadores) y combinar sus diferentes salidas [18] en aras de lograr un mejor rendimiento.

Los multiclasificadores pueden construirse de diversas formas. Existen varios algoritmos desarrollados, algunos para problemas generales y otros para problemas específicos. Dentro de los más populares podemos encontrar: *Bagging* [4], *Boosting* [9] y *Stacking* [22]. En esencia estos métodos tienen dos partes importantes: selección de los clasificadores de base y elección de la forma de combinar las salidas.

Una de las condiciones para obtener buenos clasificadores de base es lograr la diversidad de los mismos. La diversidad mide cuán correlacionados son los resultados de los diferentes clasificadores. Existen varias medidas que pueden ser usadas para definir cuál es la combinación de clasificadores más diversa [5, 13]. A pesar del gran número de medidas de diversidad que se han desarrollado, todavía hay divergencia de criterios sobre el concepto de diversidad y qué debe medir realmente.

Bagging y *Boosting* se centran en lograr la diversidad con la manipulación de la base, transformando el conjunto de entrenamiento. *Bagging* se basa en crear diferentes conjuntos de entrenamiento, extraídos del conjunto inicial de manera aleatoria y con reemplazo. *Boosting* es parecido a *Bagging*, solo que el reemplazo es realizado estratégicamente de forma que los casos mal clasificados tienen mayor probabilidad, que los bien clasificados, de pertenecer al conjunto de entrenamiento del siguiente clasificador del sistema. Ambos combinan las salidas usando la técnica de voto pesado. Estos multiclasificadores necesitan la selección de un modelo de clasificador inestable, o sea, un modelo que con pequeños cambios obtenga valores diferentes.

Stacking es un método diferente a los anteriores pues la diversidad la busca con el empleo de diferentes modelos de clasificación. Para combinar las salidas utiliza un metaclasificador que aprende la relación entre las salidas de los clasificadores de base y la clase original. Este metaclasificador tiene como base de entrenamiento un nuevo conjunto de instancias formadas a partir del conjunto de entrenamiento inicial para los clasificadores de base, donde por cada instancia del conjunto de entrenamiento se tiene ahora un vector de rasgos compuesto por las clases de salida de cada clasificador de base y como clase, la original de la instancia.

El modelo de expertos mixtos (EM) [12] es una técnica conceptualmente similar a *Stacking*, donde se tiene un conjunto de clasificadores de base de modelos diferentes; en un segundo nivel tiene un metaclasificador, cuya función es encontrar los pesos que asignará a cada clasificador para en un tercer nivel aplicar una técnica de combinación que usa estos pesos.

Actualmente continúan desarrollándose métodos que combinan clasificadores. Otros muchos autores han realizado propuestas de multiclasificadores [8, 15-17, 20]. En este trabajo nos concentramos en desarrollar un sistema que aprenda a decidir cuál o cuáles son los clasificadores idóneos para cada caso particular.

2 Materiales y métodos

Existen dos variantes para hacer un multiclasificador: basada en un único modelo para los clasificadores de base o con diferentes modelos para cada clasificador. Se ha seleccionado esta última variante para realizar un multiclasificador que se apoya en los conjuntos de instancias bien clasificadas. Se realiza una validación de este modelo comparando sus resultados con otros de los más populares para demostrar su efectividad.

2.1 Bases de casos

Para validar este modelo se escogieron once bases que representan problemas de Bioinformática, extraídas del repositorio UCI [1]. Las bases usadas son: *Audiology, Breast cancer, Diabetes, Heart-c, Heart-h, Heart-statlog, Horse-colic, Hypothyroid, Lung-cancer, Promoters* y *Yeast*.

- *Audiology* es una base donada por Bruce Porter en el año 1987. Contiene 226 casos definidos por 59 rasgos nominales y distribuidos en 24 clases. Esta base se compone de ejemplos de distintos padecimientos auditivos.
- *Breast cancer* recoge ejemplos de casos con cáncer de pecho. En esta base se describen las células estudiadas según 10 rasgos y se agrupan en 6 clases.
- *Diabetes*, por su parte, está descrita por ocho atributos continuos u ordinales. Recopila 768 casos con descripciones de posibles enfermos de diabetes mellitus.
- *Heart-c*, *Heart-h* y *Heart-statlog* son bases pequeñas descritas con pocos rasgos, numéricos en su mayoría. Los casos recogidos en éstas representan la presencia o no de enfermedades cardiovasculares.
- *Horse-colic* resulta una base muy interesante, recoge 368 casos descritos por 28 rasgos continuos, ordinales o nominales, distribuidos en dos clases.
- *Hypothyroid* está descrita por 29 rasgos continuos o dicotómicos, y contiene 3772 casos. En ella se describen pacientes de hipotiroidismo.
- *Lung-cancer* es una base que describe padecimientos de cáncer de pulmón. Ésta es una base pequeña pero está descrita por 56 rasgos nominales que definen 3 clases.
- *Promoters* describe la existencia o no de un promotor en ciertas posiciones de secuencias de ADN. Este es un problema de gran peso en el campo del análisis de secuencias dentro de la Bioinformática por las consecuencias que tiene aparejadas. La base se describe por 57 rasgos que representan una secuencia de aminoácidos, y recoge 106 ejemplos de este problema.
- *Yeast* es una base con información sobre la localización de proteínas dentro de secuencias de aminoácidos. Está descrita por 8 rasgos que definen las características de la secuencia en cuestión. Está compuesta por 1,484 ejemplos distribuidos en 10 clases.

2.2 Sistema multiclasificador mediante metaclasificador

Se propone un nuevo multiclasificador inspirado en los modelos diseñados hasta el momento. A continuación se realiza la definición de cada una de sus partes: la topología, los modelos o clasificadores de base y cómo combinar finalmente las salidas.

2.2.1 Topología del modelo

En una base de casos de bioinformática es típico que exista gran cantidad de casos parecidos con clases diferentes, por esto se hace difícil encontrar los patrones que caracterizan cada una de las clases del problema. Con un clasificador simple no es muy fácil obtener buenos resultados en una base con estas características, por esto surge la idea de usar un sistema que combine varios modelos de clasificadores. Lo ideal es seleccionar en cada caso el clasificador adecuado. De estos métodos se encuentran varios en la literatura e incluso hay autores que han hecho un buen agrupamiento y comparación de los mismo [7, 10, 18].

Dos modelos de clasificación distintos pueden aprender distintos conjuntos de patrones en la base, por esto la unión de varios modelos diferentes en este caso puede ser beneficiosa. Por supuesto que el éxito de este método estará determinado por cuáles son los modelos que se seleccionaron. Si están menos correlacionados los errores, los resultados potenciales serán mejores [6].

Después de la selección de los métodos base entonces el problema está en la combinación de los mismos. Si se tiene en cuenta que cada clasificador después de entrenado va a tener asociado un conjunto de casos que ha logrado aprender, a los cuales se les llama casos duros, entonces una idea para combinar los resultados podría ser aprender que patrones ha logrado aprender cada clasificador.

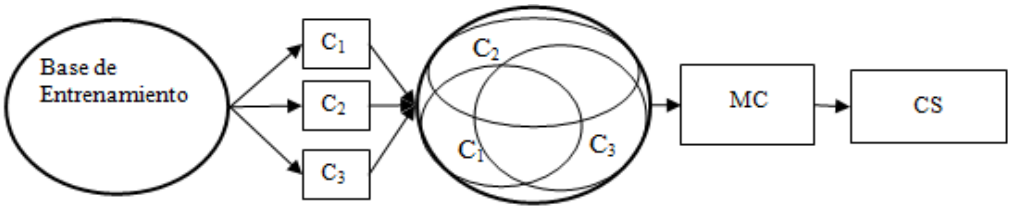

Fig. 1 Ejemplo de un modelo multiclasificador basado en conjuntos de instancias bien clasificadas con tres clasificadores de bases

A partir de aquí se podrían separar ahora los casos en grupos según el clasificador al que pertenecen. Luego si se supone que hay M clasificadores base entonces se tendrá M grupos.

La idea de este sistema se basa entonces en agregar un metaclasificador que aprenda a separar los casos según los M grupos. Pero en la mayoría de las bases al menos un 50% de la base son casos que son fáciles de aprender por cualquier clasificador y en este caso se tiene en la base este porciento de casos igual para los M grupos. En aras de reducir este número se crea un M+1 grupo que tendrá los ejemplos que son reconocidos por todos. Igual quedan casos que se pueden repetir en los diferentes pares de clasificadores, pero este número no debe ser tan mayoritario. En cualquier caso es importante no seleccionar un metaclasificador con clasificación dura, para dar margen a que decida cualquiera de los grupos que la contienen.

En la figura 1, se muestra un ejemplo de este modelo usando tres clasificadores de base.

A partir de una base de entrenamiento se entrenan los clasificadores C1, C2, y C3. Cada clasificador logrará un conjunto de instancias bien clasificadas, que para mejor comprensión se ha señalado de manera concentrada en esta figura.

Puede verse entonces cómo se solapan estos conjuntos, por lo que existirán casos que serán bien clasificados por los tres clasificadores, otros que sólo serán clasificados por dos de ellos, otros que serán bien clasificados por uno sólo de ellos y por último un conjunto que será mal clasificado por los tres clasificadores. Este último conjunto puede excluirse del modelo o utilizarse como un grupo aparte. Eliminar este grupo no es más que el margen de error lógico que tendrá el modelo.

Téngase en cuenta que cuando se entrena un clasificador simple, hay un conjunto de casos que son mal clasificados por el mismo, el porciento de mal clasificados. Cuando se intenta unir dos clasificadores, el conjunto de instancias que son mal clasificadas por los clasificadores, son parte del porciento de instancias mal clasificadas por el modelo. Sacarlas del sistema de aprendizaje del metaclasificador, facilita el aprendizaje del mismo.

El siguiente paso, después de entrenados los clasificadores de base, es seleccionar un modelo de clasificación para el metaclasificador (representado en la figura 1 por MC), que debe aprender a separar las instancias en estos grupos, o sea, será entrenado con una nueva base que se diseñará a partir de la original. A esta nueva base se adicionarán los casos que conforman cada uno de los grupos de casos duros asociados a los clasificadores de base. Cada caso será descrito por los mismos rasgos descriptivos que tenía en la base original, sólo el rasgo objetivo cambiará, ahora está asociado al conjunto de clasificadores, cada caso tendrá como clase el grupo al que pertenece.

Se formarán grupos de casos bien clasificados por cada uno de los clasificadores de base, a los que se les restará el conjunto de instancias que son bien clasificadas por todos, que se tendrá en otro conjunto (M+1). Si esta diferencia entre estos conjuntos produce conjuntos muy pequeños, se tendrán pocas instancias asociadas a los clasificadores que les suceda esto. Por ejemplo, cuando el conjunto de intersección de todos los grupos creados por los clasificadores de base es muy grande es típico que suceda esto y en este caso vamos a dejar para el metaclasificador una base de datos con pocas instancias para las

clases, dificultando el aprendizaje. Teniendo en cuenta esto, se estableció un umbral para decidir qué instancias, del conjunto de casos bien clasificados por todos los clasificadores, se pasarán al resto de los conjuntos. Para ello se seleccionan aquellas que son mejor clasificadas por cada uno, o sea, que la probabilidad con que el clasificador da la salida esté por encima del umbral establecido. Este umbral dependerá de la base que se trate, y puede ser un parámetro seleccionado por el usuario en el modelo. Un umbral igual a 0 significa que serán añadidas todas a cada conjunto, cada clasificador estará representado por el conjunto completo de las instancias que clasifica bien. Un umbral igual a 1 significa que se queda como la idea original, cada clasificador es representado por el conjunto dado por la diferencia entre el conjunto de los casos que él clasifica bien y el conjunto de aquellos que todos clasifican bien.

Formalmente el problema original a resolver tiene como función objetivo FO: $RD \rightarrow O$, donde RD es el conjunto de los rasgos del problema y O={1,2,…,N} es el conjunto de clases. El problema transformado para el metaclasificador quedaría con una función objetivo F: $RD \rightarrow O2$, donde $O2$ = {1,2,…,M+1} es el conjunto que etiqueta a los grupos formados por los clasificadores al que se une el grupo de las instancias bien clasificadas por todos.

Resumiendo, después de entrenados los clasificadores de base, se construye la nueva base de entrenamiento a partir de los casos duros obtenidos por cada clasificador entrenado.

Con esta nueva base se entrena entonces el metaclasificador. Este algoritmo puede definirse por los siguientes pasos, ya explicados anteriormente:

1. Entrenar los clasificadores de base
2. Formación de los conjuntos de casos bien clasificados
3. Detección del conjunto intersección
4. Formación de la nueva base de entrenamiento para el metaclasificador

– Para cada caso de la base original,
  – Para cada clasificador C_i,
    - Si el caso pertenece a la diferencia del conjunto de bien clasificados y el conjunto intersección, añadir a la base y asociar como clase *i*
    - Si el caso pertenece al conjunto intersección y la probabilidad está por encima del umbral establecido, añadir a la base y asociar como clase i
    - Si el caso pertenece al conjunto intersección y no fue añadido con ningún clasificador, añadir a la base y asociar como clase M.

5. Entrenar el metaclasificador con la nueva base creada.

Después de entrenado el metaclasificador sólo resta definir el módulo para calcular la salida general del sistema (CS). De esta manera queda definida la arquitectura de este multiclasificador.

Con esta nueva base se entrena el metaclasificador, el cual tendrá como salida un vector de probabilidades asociado ahora a los grupos que se crearon. El módulo de combinación de las salidas entonces se encarga pesar las salidas de los clasificadores con esta salida del metaclasificador.

2.2.2 Selección de los clasificadores de base

La selección de los modelos de clasificadores de base es un gran problema hoy en día. Algunos autores dicen que una forma de hacerlo es seleccionar aquellos, cuyos errores tengan muy baja correlación [6, 11]. Pero ésta es una labor difícil y no comprobada del todo. Muchos autores han propuesto varias formas de medir la diversidad de los clasificadores [3, 13, 21]. No obstante, el presente trabajo no pretende optimizar la búsqueda de los clasificadores, sólo se propone el uso de diferentes paradigmas de clasificación, en términos de distintas formas de buscar las regiones de decisión.

La selección de los clasificadores de base depende del problema. Lo ideal para una base determinada es seleccionar los clasificadores que puedan combinarse mejor, quizás con la primera medida que se menciona, aquellos clasificadores con errores menos correlacionados, o con cualquiera de las medidas propuestas por Kuncheva y Whitaker [13].

Una primera selección en el presente trabajo incluye un modelo de árbol de decisión, uno neuronal, un modelo probabilístico, uno perezoso

y un SVM (*Support Vector Machine*). Como ejemplo de árbol de decisión se selecciona el J48, MLP (*Multi-Layer Perceptron*) como modelo neuronal, y el *k*NN como modelo perezoso.

El último detalle a tener en cuenta es la función núcleo que se usará para el SVM, en este caso se usarán tres, uno polinomial de primer grado (lineal), otro polinomial también pero de segundo grado, y el último con función núcleo gaussiana de base radial. De todos estos modelos sólo se escogieron los que mejores resultados aportaron en estas bases respecto al porciento de clasificación correcta y el área bajo la curva ROC: AUC (*Area Under the Curve*).

Con el uso de técnicas estadísticas se seleccionó finalmente un J48, una red bayesiana y un SVM lineal. Como metaclasificador, por su potencia en problemas de clasificación, se seleccionó un MLP. Ésta es la misma topología de clasificadores que se empleó para *Stacking*. Para *Bagging* y *Boosting*, que usan un único modelo de clasificador de base, se hicieron pruebas usando estos mismos clasificadores como modelos de base, añadiendo un MLP, ya que es usado también en el modelo propuesto. En el caso de *Bagging* y *Boosting* se usaron 10 instancias del modelo de clasificador como base, ya que, con esta cantidad fue como se alcanzaron los mejores resultados.

Por las características de las investigaciones bioinformáticas, no se prioriza tampoco en este trabajo la minimización del costo computacional, o sea, se decide sacrificar tiempo de entrenamiento en aras de un mejor resultado en explotación. En definitiva, los análisis en muchas tareas de clasificación de secuencias, como la determinación de posible resistencia a un candidato antiviral, la localización de los genes, el pronóstico de cierta propiedad biológica de un compuesto identificado por una secuencia de datos; constituyen a veces sólo el preámbulo *in silico* de una investigación a posteriori *in vitro* infinitamente más costosa en tiempo y dinero.

2.2.3 Combinación de las salidas del multiclasificador

La combinación de las salidas es otro paso importante a la hora de construir un sistema multiclasificador. Puede dividirse en dos tipos: selección o fusión. La selección es la simple elección del "mejor" clasificador para una instancia determinada. La fusión, por otro lado, se basa en combinar, mediante alguna función, las salidas de los diferentes clasificadores. Esta segunda puede basarse en combinar las salidas de las clases ya etiquetadas o las salidas de valores continuos, correspondientes a la distribución de probabilidad dada por el clasificador.

En el método que se propone el metaclasificador tendrá como salida un vector de probabilidades en correspondencia con los grupos que se crearon. Para el módulo de combinación de salidas tienen en cuenta dos formas de combinarlas: una basada en selección y otra basada en fusión.

La combinación por fusión se basa en pesar las salidas de los clasificadores con la salida del metaclasificador como muestra la ecuación 1.

$$S = SC * SMC \tag{1}$$

El interlineado por arriba y por debajo de la fórmula o ecuación será de 6 pts.

Al igual que las tablas y figuras, las fórmulas o ecuaciones deberán ajustarse al ancho de la columna, si éstas sobrepasan la dimensión de la columna abarcarán el ancho de la página.

En todos los casos se utiliza el vector de probabilidades de salida de los clasificadores, de forma que en la ecuación 1 *SC* es la matriz que contiene las salidas de los clasificadores, o sea, *SC*(*i*,*j*) es la probabilidad de clasificar la clase *j* por el clasificador *i*. *SMC* es el vector columna *N*x1 de las probabilidades de salida del metaclasificador, y *S* es el vector columna *C*x1 de probabilidades resultantes del sistema.

La combinación por selección es más simple, consiste en seleccionar el clasificador de mayor probabilidad asignada atendiendo al vector de salida del metaclasificador. Éstas son las dos combinaciones de salidas que se proponen en este trabajo.

Esta propuesta de multiclasificador con varios modelos de clasificadores de base, basado en los casos duros y con funciones de combinación de salida, será denominada en lo sucesivo MEHI (*Muti-Expert by Hard Instances*).

2.2.4 Comparación con Stacking y el modelo de expertos mixtos

La variante de combinación de clasificadores que se propone está basada en dos multiclasificadores muy conocidos: *Stacking* y mezcla de expertos o modelo de expertos mixtos. Estos métodos se basan en diferentes modelos de clasificación y usan un metaclasificador diferente para combinar las salidas.

Stacking usa, como base de datos para entrenar al metaclasificador, la salida de los clasificadores de base, o sea, construye una nueva base de casos donde cada caso va a tener como rasgos la salida de los clasificadores y preserva la clase. Por otro lado, MEHI construye también una base de casos para entrenar el metaclasificador, solo que preserva los rasgos de los casos y lo que transforma es el rasgo objetivo, teniendo ahora tantas clases como clasificadores de base se tienen, más una clase que agrupa los casos bien clasificados por todos.

Mezcla de expertos usa un metaclasificador que se entrena con una base de casos parecida a la que se usa en MEHI, solo que va a tener todos los casos de la base original, con el clasificador que dio mayor probabilidad a la clase real del caso, aunque lo haya clasificado mal. MEHI elimina los casos ruidosos que no fueron aprendidos por ningún clasificador, ya que de ninguna forma van a ser bien clasificados y pueden perjudicar el entrenamiento. MEHI puede tener casos repetidos, aquellos que sean bien clasificados por varios clasificadores y no todos de ellos. En dependencia del umbral que se seleccione para decidir si un caso es bien clasificado por un clasificador base o no, este número de casos repetidos puede aumentar o bajar.

2.2.5 Evaluación de la clasificación

Las medidas más conocidas para evaluar la clasificación están basadas en la matriz de confusión que se obtiene cuando se prueba el clasificador en un conjunto de datos que no intervienen en el entrenamiento. Basados en estas medidas, se calcula el error, el porciento de clasificación correcta (conocido exactitud o como *accuracy*, en la literatura en inglés), la sensibilidad, la precisión y especificidad. Otra forma de evaluar el rendimiento de un clasificador es por el análisis de la llamada *Receiver Operator Curve* (ROC) [19]. También se usa el área bajo esta curva AUC, ya mencionada anteriormente, como un indicador de la calidad del clasificador.

La forma de dividir los datos en conjunto de entrenamiento y prueba es también muy importante. El método de validación cruzada con *k* subconjuntos (*k-fold cross-validation*) es uno de los más usados, este método se basa en dividir la base en *k* segmentos y realizar *k* procesos de entrenamientos y pruebas, de forma que el proceso *i* toma el segmento *i* para prueba y el resto para entrenamiento. Este método fue utilizado para realizar la comparación del multiclasificador propuesto con otros. Como medidas se analizaron tanto el porciento de clasificación correcta, como el AUC.

3 Resultados y discusión

En este trabajo se usaron varias topologías para escoger los clasificadores de bases y el metaclasificador. Finalmente se seleccionaron 3 clasificadores de base: J48, red bayesiana y SVM. Como metaclasificador se utilizó un MLP.

Se hizo un estudio del umbral para formar los grupos por clasificadores, en correspondencia con cada una de las bases. Para las diferentes bases utilizadas, los mejores resultados se alcanzaron con umbrales distintos como era de esperar, ya que las bases tienen características diversas. De manera general el valor del umbral osciló entre 0,6 y 0,9.

Para validar los resultados del modelo de combinación de clasificadores que se propone, se compara con B*agging*, *Boosting* y S*tacking*, por ser los multiclasificadores más comúnmente utilizados en la literatura. En los casos de *Bagging* y *Boosting* se probaron tres clasificadores bases: J48, SVM y MLP. En el caso de *Stacking* se utilizó la misma topología usada para el modelo aquí propuesto.

En la figura 2 se ilustran gráficamente los resultados de los diferentes multiclasificadores, haciendo una comparación de MEHI contra el resto de los métodos analizados (Ver Fig. 2).

En esa figura cada punto representa una de las 11 bases. Por ejemplo, en la primera gráfica

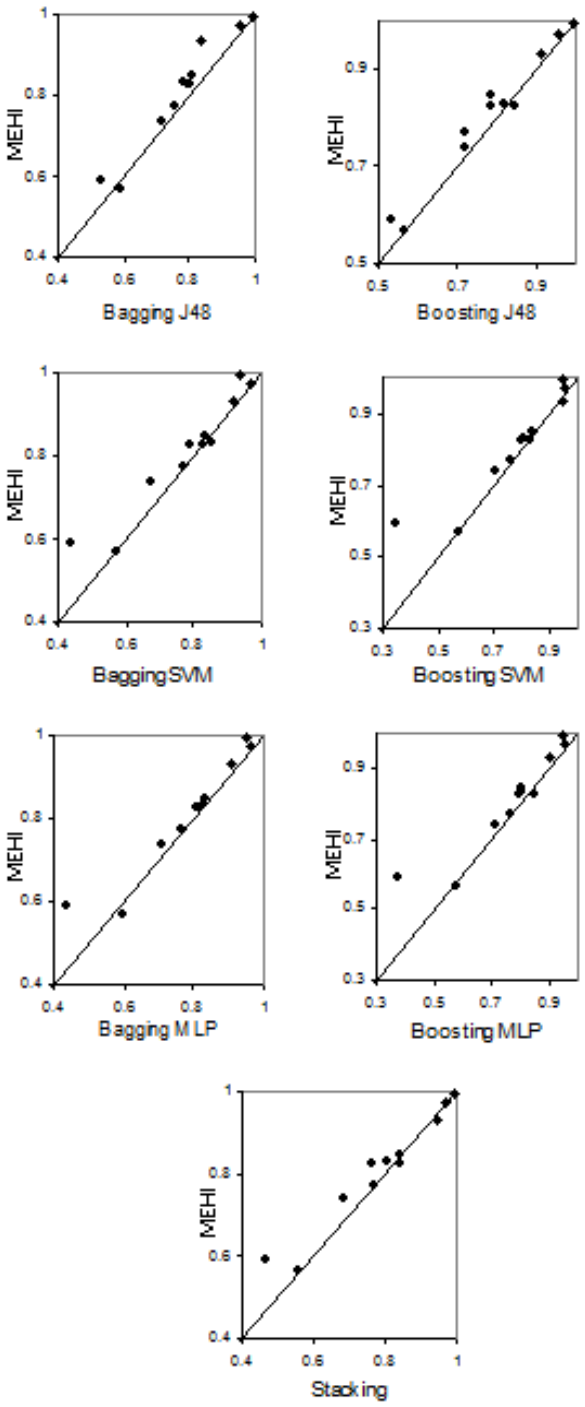

Fig. 2. Comparación de MEHI contra *Bagging, Boosting, Stacking*

se muestra el valor de exactitud de *Bagging con J48* por el eje de las "*x*," y el valor la exactitud de MEHI por las "*y*".

Un punto encima de la recta *y* =*x* significa que el resultado de la exactitud del MEHI es superior y un punto por debajo de la recta significa que el resultado de *Bagging* es superior.

Puede verse en esta comparación que en la mayoría de los casos (10 de 11 bases) los resultados del modelo propuesto son superiores o similares a los obtenidos por los demás multiclasificadores. Lo mismo se lustra en la comparación del clasificador propuesto respecto a los estándares mencionados.

Se puede apreciar por ejemplo, que el multiclasificador propuesto en este trabajo es superior en 10 de las 11 bases con respecto al *Bagging* usando J48 como clasificador de base, es superior en 9 bases contra *Bagging* usando SVM y superior en 10 contra el *Bagging* usando MLP. Así mismo en el análisis contra el *Boosting* es superior en 10 bases usando J48 como clasificador de base y superior en 9 bases usando SVM y MLP. En la comparación contra S*tacking* resultó mejor en 9 de las 11 bases. Además en la última columna se muestra en negritas que hay 5 bases para las cuales MEHI obtuvo mejor exactitud que todos los 7 métodos con los que se comparó.

Para hacer una comparación más particular con los diferentes multiclasificadores puede verse la tabla 1, donde se muestra la cantidad de veces (bases) que fue superior el método MEHI en una comparación, con cada uno de los métodos definidos anteriormente.

Los resultados de la exactitud lograda se muestran en la tabla 2. En esta tabla puede verse que, de manera general, en la mayoría de las bases de datos, MEHI supera a todos los demás multiclasificadores entrenados, o al menos obtiene resultados similares al que obtuvo mejores resultados.

Estos resultados mostrados fueron analizados usando la medida de exactitud; pero también se realizó un estudio similar para el indicador AUC donde los resultados fueron similares a los que se muestran para en la tabla 2 y la figura 2.

Tabla 1. Número de bases en las que MEHI supera a cada uno de los multiclasificados probados

Bagging-J48	Bagging-SVM	Bagging-MLP	Boosting-J48	Boosting-SVM	Boosting-MLP	Stacking	MEHI
10	9	10	10	9	9	9	9

Tabla 2. Porcentajes de buena clasificación obtenidos por cada uno de los métodos

Bases de Datos	Exactitud (Porciento de bien clasificados)							
	Bagging-J48	Bagging-SVM	Bagging-MLP	Boosting-J48	Boosting-SVM	Boosting-MLP	Stacking	MEHI
audiology	0,801	0,783	0,819	0,841	0,796	0,845	0,761	0,827
breast-cancer-w	0,956	0,969	0,966	0,954	0,956	0,959	0,969	**0,971**
diabetes	0,751	0,765	0,763	0,717	0,756	0,759	0,763	0,773
heart-c	0,776	0,848	0,825	0,815	0,800	0,799	0,802	0,832
heart-h	0,793	0,824	0,806	0,782	0,820	0,793	0,837	0,827
heart-statlog	0,804	0,830	0,830	0,785	0,830	0,796	0,841	**0,848**
horse-colic	0,710	0,673	0,707	0,717	0,697	0,710	0,680	**0,740**
hypothyroid	0,995	0,936	0,952	0,996	0,947	0,950	0,996	**0,997**
lung-cancer	0,531	0,438	0,438	0,531	0,344	0,375	0,469	**0,594**
promoters	0,840	0,915	0,906	0,915	0,943	0,906	0,943	0,934
yeast	0,590	0,571	0,596	0,563	0,571	0,577	0,559	0,570

Tabla 3. Resultados de la prueba de Wilcoxon para comparar Bagging, Boosting, Stacking y MEHI, tal como se obtiene en el paquete estadístico SPSS

	MEHI vs. Bagging_J48	MEHI vs. Bagging_SVM	MEHI vs. Bagging_MLP	MEHI vs. Boosting_J48	MEHI vs. Boosting_SVM	MEHI vs. Boosting_MLP	MEHI vs. Stacking
Z	-2,667	-2,401	-2,312	-2,667	-2,578	-2,490	-2,134
Sig. Exacta (bilateral)	0,005	0,014	0,019	0,005	0,007	0,010	0,032

Para probar hasta que punto estas ventajas de MEHI son estadísticamente significativas, se aplicó un Análisis de Friedman a los datos de la tabla 2 y se encontró que había diferencias entre las exactitudes de los 8 clasificadores con una alta significación (menor que 0.001). Luego se hicieron pruebas de rangos con signo de Wilcoxon comparando MEHI con cada uno de los

7 multiclasificadores y los resultados se muestran en la tabla 3.

Puede observarse que los resultados son significativos en todas las comparaciones. MEHI supera especialmente a Bagging_J48 a Boosting_J48 a Boosting_SVM con muy alta probabilidad (mayor del 99%). Como se ha demostrado que los multiclasificadores *Bagging*, *Boosting*, *Stacking* logran obtener resultados superiores a los clasificadores simples, es presumible que MEHI también.

5 Conclusiones

En las conclusiones se analizan, interpretan y califican los resultados, en especial con respecto al problema investigado. Cuando se justifique y sea apropiado, resaltar la importancia de los descubrimientos o innovaciones.

En este trabajo se diseña e implementa un modelo de combinación de clasificadores basado en el uso de varios que dividen la base de casos en grupos que luego son aprendidos por un metaclasificador.

Se compararon los resultados de MEHI con los multiclasificadores más reconocidos y usados en la literatura, resultando los de MEHI significativamente superiores para una muestra de 11 bases de datos internacionales de carácter biomédico o bioinformático.

Este es un modelo de multiclasificador de propósito general, que promete obtener en muchos casos los porcentajes de clasificación correcta y área bajo la curva ROC superiores o similares a los obtenidos con clasificadores simples.

Referencias

1. **Larrañaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J.A., Armañanzas, R., Santafé, G., Pérez, A., & Robles, V. (2006).** Machine learning in bioinformatics. *Briefings in Bioinformatics*, 7(1), 86–112.

2. **Baldi, P. & Soren, B. (2001).** *Bioinformatics: The Machine Learning Approach* (2nd ed.). Cambridge, Mass.: MIT Press.

3. **Polikar, R. (2006).** Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine,* 6(3), 21–45.

4. **Breiman, L. (1996).** Bagging predictors. *Machine Learning,* 24(2), 123–140.

5. **Freund, Y. & Schapire, R.E. (1996).** Experiments with a new boosting algorithm. T*hirteenth International Conference on Machine Learning (ICML'96)*, Bari, Italy, 148–156.

6. **Wolpert, D.H. (1992).** Stacked generalization. *Neural Networks*, 5(2), 241–259.

7. **Kuncheva, L.I. & Whitaker, C.J. (2003).** Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning,* 51(2), 181–207.

8. **Brown, G., Wyatt, J., Harris, R., & Yao, X. (2005).** Diversity creation methods: a survey and categorisation. *Information Fusion,* 6(1), 5–20.

9. **Jacobs, R.A., Jordan, M.I., Nowlan, S.J., & Hinton, G.E. (1991).** Adaptative mixtures of local experts. *Neural Computation,* 3(1), 79–87.

10. **Nanni, L. & Lumini, A. (2006).** FuzzyBagging: A novel ensemble of classifiers. *Pattern Recognition*, 39(3), 488–490.

11. **Nguyen, M.H., Abbass, H.A., & McKay, R.I. (2006).** A novel mixture of experts model based on cooperative coevolution. *Neurocomputing,* 70(1-3), 155–163.

12. **Saha, S., Murthy, C.A., & Pal, S.K. (2007).** Rough set based ensemble classifier for web page classification. *Fundamenta Informaticae,* 76(1-2), 171–187.

13. **Dimitrakakis, C. & Bengio, S. (2005).** Online adaptive policies for ensemble classifiers. *Neurocomputing,* 64, 211–221.

14. **Partalas, I., Tsoumakas, G., Katakis, I., & Vlahavas, I. (2006).** Ensemble pruning using reinforcement learning. *4th Helenic conference on Advances in Artificial Intelligence (SETN'06). Lecture Notes in Artificial Intelligence, 3955,* 301–310.

15. **Asuncion, A. & Newman, D.J. 2007.** *UCI Machine Learning Repository*. Retrieved from http://www.ics.uci.edu/$\sim$mlearn/MLRepository.html.

16. **Dietterich, T.G. (2000).** Ensemble methods in machine learning. *First International Workshop on Multiple Classifier Systems (MCS'00)*, Cagliari, Italy, 1–15.

17. **Ghosh, J. (2002).** Multiclassifier systems: Back to the future. *Multiple Classifier Systems: Third International Workshop (MCS 2002). Lecture Notes in Computer Science*, 2364, 1–15.

18. **Canuto, A.M.P., Abreu, M.C.C., Oliveira, L.M., Xavier Jr., J.C., & Santos, A.M. (2007).** Investigating the influence of the choice of the ensemble members in accuracy and diversity of selection-based and fusion-based methods for ensembles. *Pattern Recognition Letters,* 28(4), 472–486.
19. **Hansen, L.K. & Salamon, P. (1990).** Neural networks ensembles*. IEEE Transactions on Pattern Analisys and Machine Intelligence,* 12(10), 993–1001.
20. **Banfield, R.E., Hall, L.O., Bowyer, K.W., & Kegelmeyer, W.P. (2005).** Ensemble diversity measures and their application to thinning. *Information Fusion,* 6(1), 49–62.
21. **Tang, E.K., Suganthan, P.N., & Yao, X. (2006).** An analysis of diversity measures. *Machine Learning,* 65(1), 247-271.
22. **Provost, F.J. & Fawcett, T. (1997).** Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions. *Third International Conference on Knowledge Discovery and Data Mining (KDD-97),* Newport Beach, California, USA, 43–48.

Isis Bonet Cruz Licenciada en Ciencias de la Computación en la Universidad Central "Marta Abreu" de Las Villas (UCLV), Cuba (2001). Máster en Ciencia de la Computación, UCLV (2005). Doctora en Ciencias Técnicas, UCLV (2009). Experiencia profesional en Redes Neuronales, Clasificación, Bioinformática. Área de interés: Inteligencia Artificial y Bioinformática**.**

Abdel Rodríguez Abed Licenciado en Ciencias de la Computación, UCLV (2006). Máster en Ciencias de la Computación, UCLV (2007). Experiencia profesional en Sistemas multiagentes. Áreas de Interés: Inteligencia Artificial y Bioinformática.

María Matilde García Lorenzo Licenciada en Cibernética Matemática en la Universidad Central "Marta Abreu" de Las Villas, Cuba. Doctor en Ciencias Técnicas en la misma institución (1997). Experiencia profesional en Inteligencia Artificial, Redes Neuronales Artificiales, Reconocimiento de Patrones. Áreas de Interés: Inteligencia Artificial y Bioinformática.

Ricardo Grau Ábalo Licenciado en Ciencias Matemáticas en la Universidad Central "Marta Abreu" de Las Villas, Cuba, Doctor en Ciencias Matemáticas en la misma institución (1987). Experiencia profesional en Ecuaciones Diferenciales, Estadística y Computación aplicadas. Área de interés: Bioinformática.

Artículo recibido el 22/02/2011; aceptado el 19/10/2012.