UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD #6

DESARROLLO DE MODELOS DE CLASIFICACIÓN DE ACTIVIDAD BIOLÓGICA EMPLEANDO MÁQUINAS DE SOPORTE VECTORIAL

TESIS EN OPCIÓN AL GRADO DE MÁSTER EN BIOINFORMÁTICA

AUTOR:

ING. YAIKIEL HERNÁNDEZ DÍAZ.

TUTOR:

DR. RAMÓN CARRASCO VELAR.

CIUDAD DE LA HABANA, CUBA

FEBRERO, 2010

TESIS EN OPCIÓN AL GRADO DE MÁSTER EN BIOINFORMÁTICA

Desarrollo de modelos de clasificación de actividad biológica empleando Máquinas de Soporte Vectorial.

Autor: Ing. Yaikiel Hernández Díaz

Universidad de las Ciencias Informáticas

Km 2 ½ Carretera a San Antonio de los Baños, Torrens, Boyeros, Ciudad de la Habana

yhernandezd@uci.cu

Tutor: Dr. Ramón Carrasco Velar

Universidad de las Ciencias Informáticas

Km 2 ½ Carretera a San Antonio de los Baños, Torrens, Boyeros, Ciudad de la Habana

rcarrasco@uci.cu

Agradecimientos

Les agradezco infinitamente a mis padres y familia en general por el apoyo brindado para el desarrollo de este trabajo y mantenerme motivado para continuar con mi superación personal.

A mi tutor Ramón Carrasco Velar, por todo su tiempo, paciencia y ayuda brindada. Te estaré eternamente agradecido.

A la gente del 54102 por su preocupación constante y ser quienes son.

Dedicatoria

En la vida es importante mantener los sueños y hacerte cada día mejor persona, tanto profesional como sentimentalmente. Pero nada se hace posible sin el amor de todas aquellas personas que hacen que tu mundo sea cada día diferente y especial. Quiero dedicar esta tesis a quienes son y serán siempre parte importante de mi corazón:

A Dios por demostrarme que existe y que más haya de toda ciencia hay que tener fe.

A mi alma gemela de cada día, quien después de nueve meses y 26 años su amor se incrementa y me demuestra que se es madre para toda la vida porque siendo tan pequeña es tan grande a la vez, mi mami Titica.

A quien me enseñó a ser un hombre de bien, alguien tan especial que aunque no se lo demuestre tanto lo quiero infinitamente y tengo el honor de tenerlo como padre, a mi papá Nelson.

A quienes tienen un corazón inmenso, y son grandes de sentimiento por ser solamente quien son, mis yeyos de toda la vida, porque entre guitarras y amor se merecen lo mejor del mundo, Pino y Moisés.

A mi hermanita Lili, por ser tan especial en las buenas o malas, aunque aún siendo grande das perretas por todo, se que siempre estarás.

Por último y no menos importante, a la persona que será el centro de una nueva familia, quien hace hasta lo imposible por ser siempre el centro de mi vida y mi mundo para lograrlo finalmente, quien sabe el significado de la palabra linda, mi esposa Martha.

A todos gracias por existir y que Dios los bendiga.

Resumen

En este trabajo se presentan modelos de clasificación de antibióticos de la familia de las cefalosporinas y de inhibidores del Factor Esteroidogénico-1 utilizando Máquinas de Soporte Vectorial. Dada la gran cantidad de información estructural fue necesario realizar la reducción de la cantidad de variables, por lo que se propone un procedimiento general para la selección de variables a partir de varios métodos de reducción de variables basados en técnicas de inteligencia artificial que han sido implementadas y evaluadas. Se aborda el problema de la identificación y reducción de un conjunto representativo de atributos para así contribuir al mejoramiento de los modelos de clasificación. Se implementaron procedimientos de búsqueda por algoritmos genéticos, enfriamiento simulado, búsqueda secuencial y una hibridación entre este último y algoritmos genéticos; con el fin de alcanzar mayor robustez y eficiencia. Se implementaron además, varias medidas de asociación entre subconjuntos variables, a partir de conceptos de la estadística clásica o tomadas de la Teoría de la Información de Shannon. En todos los casos analizados se reduce el espacio de variables en más del 65%. Todos estos procedimientos de búsqueda presentan una complejidad temporal de orden polinomial; lo cual demuestra la viabilidad práctica en cuanto a costo y recursos computacionales necesarios para cada procedimiento implementado. Los modelos desarrollados presentaron una correcta clasificación de entre 83 y 100 % y precisión entre 0.88 y 1 para cefalosporinas y los inhibidores del factor Esteroidogénico-1 respectivamente.

Palabras Claves: *reducción, clasificación, subconjuntos, atributos.*

Abstract

In this investigation, classification models of the family of antibiotics cephalosporins and inhibitor of steroidogenic factor-1 using support vector machines. Given the large amount of structural information was needed to reduce the number of variables. We propose a general procedure for selecting variables based on several variables reduction methods based on artificial intelligence techniques that have been implemented and evaluated. It addresses the problem of identification and reduction of a representative set of attributes to assist in the improvement of classification models. Search procedures were implemented by genetic algorithms, simulated cooling, sequential search and a hybrid between this and genetic algorithms, in order to achieve greater robustness and efficiency. It also implemented several measures of association between variable subsets, based on concepts borrowed from classical statistical theory of Shannon Information. In all cases analyzed reduces the space of variables in more than 65%. All of these search procedures present a polynomial time complexity of order, which demonstrates the practical feasibility and cost in computational resources required for each procedure implemented. The developed models showed accuracy of between 83 and 100% and accuracy between 0.88 and 1 for cephalosporins and inhibitors of steroidogenic factor-1 respectively.

Keywords: *reduction, classification, sub-attributes.*

Tabla de Contenido

Tabla de Contenido

Introducción

Los avances en el sector de la biorgánica, junto con los extraordinarios progresos de la fisiología, la bioquímica, la medicina y las técnicas de computación han promovido una revolución en el ámbito del diseño y producción de fármacos. Entre las muchas funciones de la farmacología la más importante es la creación de medicamentos de alta calidad para la preservación de la salud de los seres humanos; de ahí que los medicamentos son la base para casi cualquier programa de salud pública intencionado a reducir la morbilidad o mortalidad.

La predicción de la actividad biológica de compuestos químicos es hoy día un objetivo principal dentro de la Industria Médico Farmacéutica Mundial. El alto costo del proceso de investigación - desarrollo de nuevos fármacos ha obligado, a este sector económico, a adoptar la estrategia del uso de técnicas de la computación y la informática para acelerar el proceso y disminuir los costos. En los últimos años, la industria farmacéutica ha reorientado sus investigaciones y prestado más atención a aquellos métodos que permitan una selección racional o el diseño de nuevos compuestos con propiedades deseadas (1).

En la literatura se han reportado varios enfoques para el diseño molecular asistido por computadora, muchos de los cuales están basados en la correlación entre la estructura química y diferentes propiedades de las moléculas. La efectividad de estos métodos depende en gran medida de la forma de describir la estructura química, así como de la técnica de procesamiento de los datos.

La búsqueda racional de entidades biológicamente activas candidatas a medicamentos sin el empleo de las técnicas de computación es un proceso complejo que suele llevar alrededor de una década de investigación y desarrollo con un altísimo gasto en recursos; tras lo cual tan solo un 5 ó 10% de las moléculas que llegan a fase de ensayos clínicos terminan siendo comercializadas (2). Por lo anteriormente expuesto, el presente trabajo se encuentra

enmarcado dentro del proyecto CITMA 01700060 Plataforma Inteligente para la Predicción de Actividad Biológica de Compuestos Orgánicos (alasGRATO) que se desarrolla en la Facultad 6 de la Universidad de las Ciencias Informáticas.

Actualmente la plataforma alasGRATO cuenta con una base de datos de regular tamaño formada por moléculas y un gran número de descriptores asociados. Estos serán utilizados por los métodos de inteligencia artificial implementados en la plataforma para la predicción de actividad biológica asociando esta a la estructura química. Una de estas técnicas son las Máquinas de Soporte Vectorial (MSV), técnica potente para el desarrollo de modelos de clasificación y regresión que ha constituido un aporte relativamente reciente para el establecimiento de modelos de relación estructura actividad (SAR) y de relación estructura actividad cuantitativa (QSAR). Esta técnica es particularmente eficiente cuando se aplica a muestras con un gran número de datos. En la presente investigación se deberá realizar la predicción partiendo de la utilización de una cantidad elevada de descriptores topológicos, topográficos e híbridos; aunque solo algunos de ellos aportan información realmente útil para el establecimiento de los modelos. La generalidad de esos descriptores parte de formulismos que se basan en la matriz de conectividad de los vértices o aristas del grafo químico por lo que se encuentra elevada redundancia en la información que ellos contienen. Otro problema es el elevado consumo de los recursos de cómputo cuando se necesita procesar una cifra tan elevada de datos. Por lo tanto, para el desarrollo de los modelos que se desean, se hace necesario reducir los descriptores, con el fin de eliminar gran parte de la redundancia de información en la base de datos y para mejorar la calidad de los modelos.

Por lo tanto, para el desarrollo de modelos SAR por MSV en las muestras de estudio se necesita reducir previamente el tamaño de dichas muestras, de forma que se obtengan modelos de calidad computacionalmente tratables. Son diferentes los procedimientos que se emplean en la actualidad para la reducción de la dimensión en una muestra dada. Entre los más modernos se destacan las técnicas de inteligencia artificial, que se emplean solas o combinadas con técnicas clásicas de la estadística avanzada. Mediante las cuales se

determina la presencia de variables irrelevantes o redundantes y se seleccionan las más representativas del fenómeno en estudio. Por lo que se plantea como **Problema Científico:** ¿Cómo obtener modelos de relación cualitativa entre la estructura química y la actividad biológica a partir de una elevada cifra de datos estructurales empleando Máquinas de Soporte Vectorial?

Este problema científico tiene como **Objeto de Estudio** la reducción del número de variables independientes previo el empleo de clasificadores y como **Campo de Acción** la aplicación de las Máquinas de Soporte Vectorial para el desarrollo de modelos de clasificación en muestras reducidas a partir de gran número de datos estructurales.

Para dar solución al problema planteado se traza como **Objetivo General:**

Desarrollar modelos de clasificación de compuestos con actividad biológica a partir de gran número de datos estructurales utilizando Máquinas de Soporte Vectorial.

Para dar cumplimiento al objetivo general se definieron como **Objetivos Específicos:**

- ✓ Identificar algoritmos de búsqueda aplicables a la reducción de variables independientes.
- ✓ Identificar las medidas de evaluación apropiadas para garantizar la calidad de los modelos predictivos.
- ✓ Disponer de un procedimiento para la reducción del número de variables independientes.
- ✓ Disponer de modelos de clasificación de compuestos orgánicos basados en MSV.

Para dar cumplimiento a los objetivos específicos se definieron las siguientes tareas:

- ✓ Revisión del estado del arte acerca de algoritmos de búsqueda y criterios de evaluación reportados en la literatura.
- ✓ Aplicación de técnicas estadísticas para la reducción inicial del espacio de búsqueda.
- ✓ Implementación de los algoritmos de búsqueda y medidas de evaluación seleccionados para dar respuesta a la reducción de variables.

- ✓ Aplicación de los diferentes algoritmos de búsqueda y medidas de evaluación implementados a una muestra de cefalosporinas.
- ✓ Aplicación de los diferentes algoritmos de búsqueda y medidas de evaluación implementados a una muestra de inhibidores del Factor Esteroidogénico-1 (SF-1).
- ✓ Análisis cluster a una muestra de Inhibidores del Factor Esteroidogénico-1 (SF-1).
- ✓ Desarrollar modelos de clasificación de cefalosporinas empleando MSV
- ✓ Desarrollar modelos de clasificación de inhibidores del Factor Esteroidogénico-1 (SF-1) empleando MSV.

Como aporte científico se espera:

Un procedimiento para la selección de variables para los ensayos existentes en la base de datos de la plataforma alasGrato y nuevos modelos de clasificación de cefalosporinas e inhibidores del Factor Esteroidogénico-1 empleando las MSV como clasificador.

Como aporte práctico se espera:

Se esperan tres aportes de naturaleza práctica en este trabajo: uno de ellos es que se dispondrá de algoritmos para la reducción de variables como servicio en la Plataforma alasGRATO; otro es que se podrá contar con modelos de clasificación que enriquezcan la base de conocimientos del proyecto y por último se tendrá una técnica de inteligencia artificial como las MSV para la clasificación.

El presente trabajo cuenta con tres capítulos:

Capítulo # 1: Revisión Bibliográfica

En este capítulo se hace una introducción al problema de selección de variables exponiéndose los conceptos fundamentales. Se plantean las técnicas y métodos que se usan a nivel mundial, las medidas de evaluación utilizadas por estos procedimientos según la bibliografía consultada; presentando además algunos sistemas automatizados que se utilizan en estos métodos y finalmente se muestran las características fundamentales del clasificador Máquinas de Soporte Vectorial.

Capítulo # 2: Métodos y programas

Se presentan los algoritmos de búsqueda de la Inteligencia Artificial para la resolución de problemas de reducción de variables, así como el seudocódigo de cada uno de ellos. Se explican las características favorables que presentan los mismos; así como las distintas herramientas utilizadas para la implementación y las vinculadas a la investigación. Se presentan las características de las muestras analizadas en esta investigación.

Capítulo # 3: Resultados y Discusión

En este capítulo se realiza una descripción de los algoritmos implementados y el funcionamiento general del sistema de selección de variables y su acoplamiento con la librería *libsvm* de las Máquinas de Soporte Vectorial incorporada al Weka. Se realizan y discuten los experimentos realizados con una muestra de antibióticos y una antitumoral para obtener los modelos de clasificación empleando Máquinas de Soporte Vectorial.

CAPÍTULO 1

Revisión Bibliográfica

Teniendo en cuenta que el primer paso para el establecimiento de modelos SAR con MSV a partir de un gran número de datos estructurales es la necesidad de reducir el tamaño de la muestra; fue preciso incursionar primeramente en las posibilidades de realizar ese trabajo evaluando críticamente las diferentes técnicas empleadas en la actualidad para cumplir ese objetivo. Por lo tanto, gran parte de la revisión bibliográfica se dedicará a la actualización del conocimiento en estos procederes que permitirán con su aplicación, el desarrollo de modelos SAR eficientes. Se hace una reseña de la selección de variables comenzando por los conceptos básicos, la complejidad, clasificación de los métodos de selección; así como los métodos de búsqueda, software existentes a nivel mundial; realizando un análisis crítico de cada uno de ellos y se culmina con un bosquejo de algunos aspectos teóricos de las Máquinas de Soporte Vectorial empleadas como clasificador en esta tesis.

1.1 Introducción a la Selección de Variables

La selección de variables consiste en encontrar un subconjunto de variables que sean relevantes para una aplicación y lograr el máximo rendimiento con el mínimo esfuerzo, con las que se pueda llevar a cabo la tarea de clasificar de forma óptima. Reducir la dimensión presenta diversas ventajas como la reducción del coste en la adquisición de datos, mejora en la comprensión del modelo final obtenido, incremento de la eficiencia del clasificador y mejora en la eficacia del clasificador. En resumen sería: (3)

Menos datos → los algoritmos pueden aprender más rápidamente

Mayor exactitud → el clasificador generaliza mejor

Resultados más simples → más fácil de entender

El problema de selección de variables para un modelo es de tal relevancia que ha propiciado la creación de una impresionante variedad de métodos. Miller recopila y trata excelentemente los principales métodos previos a ese año. Aún con moderadas cantidades de variables, el análisis del espacio de todas las posibles combinaciones demanda demasiado tiempo de procesamiento. En este sentido, muchas estrategias se basan en la reducción de dicho espacio como los métodos de Efromyson y Furnival y Wilson. Posteriormente, sin dejar de prestar atención a la reducción del espacio de modelos, se comienzan a profundizar y mejorar

en los criterios estadísticos de selección. Los más familiares quizás son los propuestos por Mallows, Akaike y Schwarz, los cuales se basan en la familia de criterios de la suma de cuadrados penalizada. Más recientemente se han propuesto nuevos ajustes basados en esta familia de criterios, entre los cuales figura el *Risk Inflation Criterion* propuesto por Foster y George.

La búsqueda de un subconjunto de variables es un problema de tipo No Polinomial completo (NP-completo) (1), los que universalmente no tienen solución práctica, el uso de meta-heurísticas permite obtener soluciones razonablemente buenas sin explorar todo el espacio de soluciones.

Si se tratara de seleccionar un subconjunto de '*m'* características de entre un conjunto original de '*n'* características candidatas, bajo algún criterio de desempeño, se encontraría un total de:

$$\binom{n}{m} = \frac{n!}{m!(n-m)!} \quad \text{subconjuntos}$$

El número de posibilidades crece exponencialmente, haciendo impráctica la búsqueda exhaustiva, aún para valores moderados de '*n'* (**Tabla 1)**. Si se evalúa todo el espacio de posibles combinaciones, el costo computacional es muy alto. Ejemplo de esto sería.

Si '*n*' es la cantidad de características identificadas y '*m*' es la cantidad de características deseadas, el número total de posibles subconjuntos a evaluar es:

$$S = \sum_m C(n,m) = \sum_m \frac{n!}{m!(n-m)!} \qquad Si\ n = m \Rightarrow S = 2^n$$

Tabla 1: Total de subconjuntos generados para algunos valores de 'n'.

N	2^n
10	***1 024***
20	***1 048 576***
30	***1 073 741 824***
40	***1 099 511 627 776***
406	**No pudo ser calculado**

Este problema es conocido como Maldición de la Dimensión debido a la gran cantidad de datos que generalmente se manejan, y en la mayoría de los casos se está en presencia de problemas de tipo No Polinomial Completo (NP-completo). (3)

La plataforma específicamente presenta un problema de este tipo, pues cuenta con una base de datos que contiene alrededor de 406 descriptores moleculares, los cuales se utilizan para la predicción de actividad biológica en los clasificadores implementados en la plataforma. Y para lograr una buena predicción es necesario seleccionar un subconjunto de estos descriptores que aporten buena información.

La inteligencia artificial brinda varios artificios para la resolución de muchos problemas de manera eficiente, problemas que por lo general son complejos o muy complejos, lo que imposibilita en gran medida hallarles una solución exacta. Por lo cual este nuevo campo de la ciencia de la computación tiene muchas aplicaciones no solo para la resolución de los problemas matemáticos, estadísticos, naturales, sociales; sino también para la mejora de los mismos hasta que esta se encuentre cerca de lo que se conoce como solución óptima o mejor solución posible.

El hecho de encontrar un método eficaz para la selección de variables es una tarea ardua y difícil, debido a la poca información que se puede hallar al respecto.

Antes de enfrentar a un problema de selección de características o de reducción de espacio muestral es necesario conocer ciertos elementos básicos del entorno del problema para una

mejor comprensión del mismo. Entre ellos están el Dominio, Universo de Discurso, Atributo y Espacio Muestral.

1.2 Conceptos Básicos

Dominio

Un dominio es un conjunto de valores del mismo tipo. Existen distintas clasificaciones de los dominios pero para los propósitos de esta investigación se distinguen dos tipos: continuo (conjunto infinito de valores reales) y nominal (conjunto finito de valores discretos) que se representa *Dom ().*

Universo de Discurso

Se denomina universo de discurso al entorno donde se define un determinado problema y viene representado como el producto cartesiano de un conjunto finito de dominios.

Atributo

Un atributo, o también denominado característica o variable, es la descripción de alguna medida existente en el universo de discurso que toma valores en un determinado dominio.

El atributo i-ésimo se representa X_i, su valor x_i y su dominio como $Dom(x_i)$, que según la clasificación descrita previamente puede ser discreto o continuo. Si es continuo existe un rango $[a;b] \in R$ de valores posibles, y si es discreto existe un conjunto finito de valores posibles. Se denomina vector atributos $x = x_1;....;\ x_n$ al conjunto de valores correspondiente a cada uno de los atributos, y X al espacio formado por el conjunto de los atributos, $X = Dom(x_1)x....xDom(x_n)$ siendo '*n*' el total de atributos.

Espacio Muestral

Se denomina resultado básico o elemental, comportamiento individual o punto muestral a cada uno de los posibles resultados de un experimento aleatorio. Los resultados básicos elementales serán definidos de forma que no puedan ocurrir dos simultáneamente pero si uno necesariamente.

Se denomina conjunto universal, espacio muestral o espacio de comportamiento al conjunto de todos los resultados elementales del experimento aleatorio. Pueden ser de varios tipos, como por ejemplo el Espacio Muestral Discreto y el Espacio Muestral Continuo (4).

En esta investigación se define un espacio muestral discreto constituido por el conjunto de valores que toman los descriptores moleculares, variables (características) predictivas de la actividad biológica de un experimento dado.

1.3 Métodos de selección de Variables o Características

Desde su origen, la Inteligencia Artificial (IA) se encuentra enmarcada en la resolución de problemas para los que no existe método analítico alguno que permita obtener, con seguridad y en un tiempo conveniente, el óptimo teórico. Este es por ejemplo, el caso de los problemas combinatorios en que el sentido común da por imposible la enumeración. Es más que normal que el tamaño y la naturaleza de ciertos problemas combinatorios prohíban abordarlos por la vía del sentido común. Dicha investigación distingue particularmente los problemas NP-completos, para los cuales no existe un algoritmo que en tiempo exponencial sea capaz de encontrar la solución (5). La investigación de operaciones ha establecido por tales razones, métodos denominados heurísticos o meta-heurísticos, incapaces de proporcionar el óptimo formal, pero susceptibles de llegar a soluciones buenas, tanto más fiables en cuanto que permiten determinar al mismo tiempo una cota (superior o inferior) del óptimo teórico con el que se comparan.

En un sistema de aprendizaje, pueden utilizarse diversas herramientas en combinación con el propio algoritmo de aprendizaje, como por ejemplo: la discretización o continuación de características, la sustitución de valores nulos; en esta investigación, se trabajará con modelos basados en aprendizaje supervisado. El objetivo de la selección de características (variables) es reducir la dimensión de los datos. Esto se consigue eligiendo características que sean útiles para resolver el problema de aprendizaje y descartando las demás. Aunque teóricamente si la distribución estadística completa se conociese, usar más características solo podría mejorar los resultados, en los escenarios prácticos de aprendizaje puede ser mejor

usar un conjunto reducido de características (6). Para los modelos basados en aprendizaje supervisado existen diferentes métodos para la selección de variables.

1.4 Clasificaciones de los métodos de selección de características

La metodología de filtro (filter): es probablemente la primera y más conocida. En ella, se aplica primero el algoritmo de selección de características y posteriormente, el de aprendizaje empleando solo las características seleccionadas.

La única información que intercambian es el conjunto de características, lo que aporta la principal ventaja de este tipo de métodos: que son independientes del algoritmo de aprendizaje. Por ello, pueden ser utilizados con cualquier algoritmo de aprendizaje, independientemente de su eficiencia u otras propiedades, que sí afectan a otros modos de aplicación.

En la estrategia envolvente (wrapper): el método de selección de características usa el algoritmo de aprendizaje para evaluar la calidad de los conjuntos de características, utilizando alguna medida de la calidad de las soluciones que obtiene este con cada uno de los conjuntos de características candidatos. En este proceso, hay un flujo de información en ambos sentidos. En uno, el método de selección de características indica un conjunto de características a usar y en el otro sentido, se devuelve una evaluación de lo útiles que son esas características. Esto se repite hasta que finalmente se selecciona un conjunto definitivo.

La principal ventaja de esta estrategia es que la selección de características tiene una evaluación de las características en el entorno real en que serán aplicadas y, por tanto, tiene en cuenta las posibles particularidades del algoritmo de aprendizaje que se va a usar. Sin embargo, se genera una relación de dependencia entre ambos algoritmos que impone ciertos requisitos sobre el algoritmo de aprendizaje, ya que este debería ser capaz de trabajar con los conjuntos de características que determine probar el método envolvente.

Inmersa (embebed): en algunos algoritmos de aprendizaje, la selección de características está incluida en él mismo como una parte no separable. En este caso, la ventaja es que la selección de características esté diseñada de forma específica para ese aprendizaje, con lo que se espera que su rendimiento sea mejor.

Hibrida (Hybrid): usan una combinación de los dos criterios de evaluación en diferentes etapas del proceso de búsqueda.

Independientemente del método de selección de variables que se emplee, se hace evidente la descomposición modular de los mismos.

1.5 Descomposición Modular de los métodos de selección

Dicha descomposición modular para el proceso de selección de características está basada en las cuatro funcionalidades identificadas por Langley (7). Las funciones son similares a las propuestas por Dash y Liu (8).

La descomposición modular se muestra en la **Figura 1**. En ella, agrupamos las funciones en dos bloques principales:

- Evaluación de las características
- Método de búsqueda en el espacio de conjuntos de características

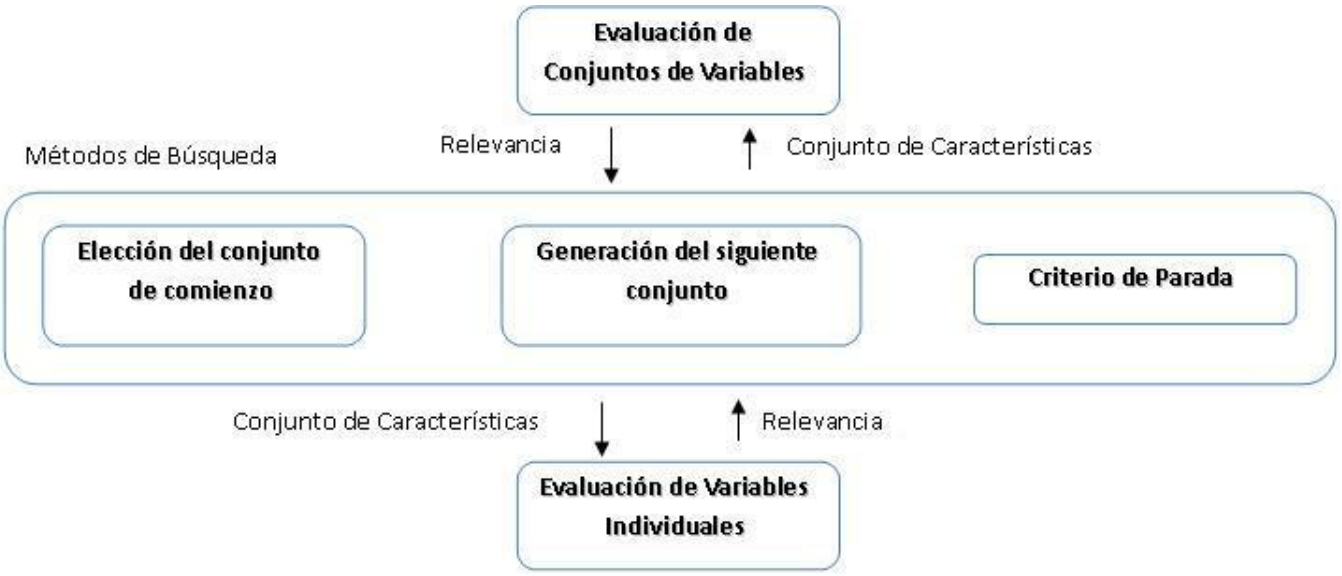

Figura 1: Descomposición Modular de los Métodos de Selección

El método de búsqueda gobierna el flujo de control del algoritmo, mientras que las funciones de evaluación de características son herramientas usadas por este. Dentro del método de búsqueda se consideraron tres partes importantes:

- La elección del punto de comienzo de la búsqueda, que puede ser inmediata o elegirse mediante un proceso más elaborado.
- La estrategia de orientación de la búsqueda, o lo que es lo mismo, elección del siguiente conjunto a evaluar.
- El criterio de parada, que puede depender de diversos factores como: el número de evaluaciones, tiempo de ejecución, o el alcance de alguna condición sobre los resultados.

En la evaluación de las características es importante diferenciar dos tipos de medidas. Por un lado las que evalúan conjuntos de características y por el otro las que evalúan características individualmente.

La visión modular del proceso de selección de características presentada aporta diversas ventajas. En primer lugar permite alcanzar una mejor comprensión de los métodos de selección de características al permitir comprender su estructura interna de forma más ordenada. Usando este modelo, también es posible crear una gran variedad de algoritmos de selección de características al combinar diferentes funciones de evaluación y mecanismos de búsqueda.

1.6 Métodos de Búsqueda

La estrategia de búsqueda representa las combinaciones de subconjuntos de características que serán evaluados hasta encontrar la solución y puede ser de tres tipos:

- Completa, donde se cubren todas las combinaciones posibles de selección.
- Heurística, al reducir el número de combinaciones a evaluar basándose en la información disponible, aunque sea mínima.
- No determinista, con un tipo de búsqueda donde no se puede esperar la misma solución en cada ejecución. Se pretende con ello no perderse en mínimos locales y encontrar algunas interdependencias entre características que la búsqueda heurística es incapaz de capturar.

La dirección de búsqueda hace referencia al modo en el cual se va creando el conjunto de características seleccionadas. Se puede llevar acabo de tres formas:

- Búsqueda secuencial hacia adelante, donde se comienza con un conjunto vacío de características al que se le van añadiendo secuencialmente nuevas; una a una, procedentes del conjunto inicial hasta que se alcanza una condición de parada.
- Búsqueda secuencial hacia atrás, en la que se parte de un conjunto con todas las características del que se va eliminando secuencialmente una a una hasta que se satisface una condición de parada.

- Búsqueda aleatoria, esquema de búsqueda que produce conjuntos de características siguiendo un patrón aleatorio. De esta forma se evita la posibilidad de acabar en un óptimo local como le puede suceder a los dos esquemas previos. (9)

A continuación se clasificarán los diferentes algoritmos de selección de características según los componentes anteriores:

Métodos de completitud: en este grupo se encuentran aquellas técnicas que emplean búsqueda completa, cubriendo totalmente el espacio de búsqueda. Dentro de este conjunto de mecanismos de selección se encuentran los siguientes:

- Focus
- Best First
- Beam Search
- Branch & Bound
- A*
- IDA*
- SMA*

De todos, uno se destaca en especial: A*

A*

Este algoritmo, a pesar de haber sido creado alrededor de los años 60, sigue en la actualidad siendo uno de los más utilizados. Desafortunadamente, es ineficiente en cuanto al uso de memoria durante el proceso de búsqueda. Por ello, en las décadas de los 80 y 90, aparecieron algoritmos basados en el propio A*, pero que limitaban el uso de memoria. Dos de los algoritmos más representativos de esta última tendencia son el IDA* (*Iterative-Deepening A*)* y el SMA* (*Simplified Memory-bounded A*).*

IDA*

Consiste en un algoritmo de profundización iterativa en el que se hace uso de la información heurística de que se dispone sobre el problema, para decidir qué nodo expandir a continuación y hasta dónde llegar en cada una de las iteraciones del proceso. En este algoritmo, como en cualquier algoritmo de profundización iterativa, cada iteración es una búsqueda "primero en profundidad". En este caso la profundización se basa en la información heurística y terminará no a una determinada profundidad, sino cuando se llegue a un nodo cuyo coste de la función heurística de evaluación f = g + h sea mayor que el actual límite de coste de f. Este método tiene las mismas ventajas y desventajas que A*, excepto en lo referente al coste espacial. En este aspecto IDA* presenta notables ventajas ya que únicamente necesita un espacio proporcional a la longitud de la ruta más larga que se explore. Esta limitación en el uso de la memoria resulta beneficiosa pero también tiene sus desventajas, ya que al convertir la búsqueda de la solución en un proceso iterativo se expandirá varias veces los mismos nodos. Esto es algo a tener en cuenta, ya que dependiendo de las características de los problemas a resolver se obtendrán mejores o peores prestaciones.

SMA*

Hace un uso más inteligente del espacio de almacenamiento y tiene la ventaja de usar toda la memoria que se disponga, de forma óptima. En este caso se puede considerar que el coste espacial es constante, aunque para poder obtener la solución óptima la ruta entre el nodo inicial y el final deberá caber en la memoria disponible. El coste temporal de este algoritmo está muy relacionado con el tamaño de la memoria: si en ella cabe todo el árbol de búsqueda, el algoritmo expandirá exactamente los mismos nodos que A*; si no es así, las prestaciones se reducen. El uso de más memoria permite mejorar la eficiencia de la búsqueda. En cambio, la implementación de éste resulta muy compleja.

Ramificación y Acotamiento (*Branch & Bound*)

Dentro de búsqueda completa se encuentran además los algoritmos de Ramificación y Acotamiento (*Branch & Bound*), propuesto por Narendra y Fukunaga. Este método garantiza la selección de un conjunto óptimo de características que cumplen la condición de monotonicidad. La base del algoritmo es que la función criterio sea monotónica e imponer una cota inferior al valor máximo (óptimo) de esta función.

La principal desventaja de estos algoritmos de búsqueda en profundidad es que pueden ser incapaces de encontrar la solución óptima si la región de factibilidad es inconexa. Además que realiza una exploración exhaustiva de la región factible, la que crece en función de 2^{n-m}, donde n es la dimensión del espacio original de entradas y m es la dimensión del espacio de entradas seleccionadas.

Métodos Heurísticos: son técnicas caracterizadas por sacrificar la promesa del subconjunto solución óptimo por obtener una solución rápida. Para ello emplean el conocimiento disponible para dirigir la búsqueda. A continuación se presentan algunos de estos métodos:

- Wrap1
- SetCover
- SOAP
- Algoritmos de búsqueda no informada
- La búsqueda tabú

Dentro de los que se puede resaltar:

Algoritmos de búsqueda no informada

Estos algoritmos no tienen en cuenta el coste de la solución durante la búsqueda. Su funcionamiento es sistemático, siguen un orden de visitas de nodos fijos, establecido por la estructura del espacio de búsqueda. Los principales ejemplos de estos algoritmos son el de anchura prioritaria, el de profundidad prioritaria y el de profundidad iterativa.

Entre los distintos métodos y técnicas heurísticas meta-heurísticas de resolución de problemas combinatorios surge, en un intento de dotar de "inteligencia" a los algoritmos de búsqueda local, el algoritmo de búsqueda tabú («tabu search») (10).

La búsqueda tabú

A diferencia de otros algoritmos basados en técnicas aleatorias de búsqueda de soluciones cercanas, este algoritmo se caracteriza porque utiliza una estrategia basada en el uso de estructuras de memoria para escapar de los óptimos locales; en los que se puede caer al "moverse" de una solución a otra por el espacio de soluciones. Al igual que en la búsqueda local, la búsqueda tabú selecciona de modo agresivo el mejor de los movimientos posibles en cada paso. Al contrario que sucede en la búsqueda local, se permiten movimientos a soluciones del entorno aunque se produzca un empeoramiento de la función objetivo; de manera que sea posible escapar de los óptimos locales y continuar estratégicamente la búsqueda de mejores soluciones.

Una ventaja importante que presentan las heurísticas y meta-heurísticas frente a las técnicas que buscan soluciones exactas, es que por lo general permiten una mayor flexibilidad para el manejo de las características del problema. No suele ser complejo utilizar algoritmos heurísticos y meta-heurísticas que en lugar de funciones lineales utilicen no linealidades. Habitualmente las heurísticas proponen un conjunto de soluciones, ampliando de esta forma las posibilidades de elección del decisor, especialmente cuando existen factores no cuantificables que no han podido ser reflejados en el modelo pero deben ser tenidos en cuenta. Se trata de un enfoque diferente al utilizado por los sistemas expertos en el campo de la inteligencia artificial. Su aplicación a los problemas de secuenciación de todo tipo es una finalidad típica y clásica; prácticamente todos ellos están basados en intentar resolver, de la mejor forma posible, problemas típicos de organización de la producción. Así, los problemas típicos de secuenciación de trabajos en máquinas, de equilibrado de líneas de montaje, de asignación de rutas, de planificación de la producción han sido, son y serán el banco de

pruebas de las más modernas técnicas de búsqueda de soluciones a problemas en los que de entrada se declina la posibilidad de encontrar la solución óptima.

Métodos Estocásticos: este tipo de técnicas permiten la búsqueda del subconjunto de características óptimas mediante la generación aleatoria de subconjuntos. A continuación se relacionan algunos de estos métodos:

- Algoritmos Genéticos
- Enfriamiento Simulado
- Las Vegas Filter
- Las Vegas Wrapper

Algoritmos genéticos («Genetic Algorithms»)

Estos algoritmos fueron introducidos por Holland (11) para imitar algunos de los mecanismos que se observan en la evolución de las especies. Los mecanismos no son conocidos en profundidad pero sí algunas de sus características: la evolución ocurre en los cromosomas, un ser vivo da vida a otro mediante la decodificación de los cromosomas de sus progenitores, el cruce de los mismos, y la codificación de los nuevos cromosomas formando los descendientes; donde las mejores características de los progenitores se trasladan a los descendientes, mejorando progresivamente las generaciones.

Basándose en estas características, Holland creó un algoritmo que genera nuevas soluciones a partir de la unión de soluciones progenitoras utilizando operadores similares a los de la reproducción, sin necesidad de conocer el tipo de problema a resolver.

Algoritmos de recocido simulado: («Simulated Annealing»)

Estos fueron introducidos por Cerny (12) y Kirkpatrick (13) para la optimización de problemas combinatorios con mínimos locales. Utilizan técnicas de optimización no determinista: no buscan la mejor solución en el entorno de la solución actual sino que generan aleatoriamente una solución cercana y la aceptan como la mejor si tiene menor coste; o en caso contrario con

una cierta probabilidad p. Esta probabilidad de aceptación irá disminuyendo con el número de iteraciones y está relacionada con el empeoramiento del coste.

Estos algoritmos derivan de la analogía termodinámica con el proceso metalúrgico del recocido: cuando se enfría un metal fundido lo suficientemente despacio, tiende a solidificar en una estructura de mínima energía (equilibrio térmico); a medida que disminuye la temperatura, las moléculas tienen menos probabilidad de moverse de su nivel energético; la probabilidad de movimiento se ajusta a la función de Boltzmann.

Las Vegas Filter (14) (LVF)

Es un algoritmo de selección de características probabilístico pensado para medidas de evaluación de tipo filtro monótonas. Explora aleatoriamente conjuntos con igual o menor número de características que el mejor que ha encontrado hasta el momento, y finaliza después de un número de pasos especificados en un parámetro. Si bien este algoritmo se emplea para problemas NP-completo que serían intratables con métodos determinísticos, existe el riesgo de no encontrar solución debido a que se hacen elecciones de rutas aleatorias que pueden no llevar a ningún sitio. Además la selección de un umbral pequeño implicará la selección de un número grande de variables, aumentando así el número de iteraciones y disminuyendo la variabilidad del conjunto haciendo lento el tiempo de cómputo.

Las Vegas Wrapper (LVW) (15)

Es un algoritmo similar al anterior pero pensado para aplicar la estrategia envolvente. También es apropiado para medidas de tipo filtro no monótonas. En este algoritmo, los conjuntos evaluados aleatoriamente pueden tener más o menos características que el mejor encontrado hasta el momento.

Métodos Ponderando Características: este tipo de técnicas se distinguen por no llevar a cabo ningún tipo de selección de forma explícita. En lugar de eso asocian a cada característica un valor de ponderación con el cual podrán modificar su participación en el posterior proceso de aprendizaje automático. De entre estos métodos se puede destacar:

- Relief

Es un algoritmo inspirado en el aprendizaje basado en casos que intenta obtener los atributos estadísticamente más relevantes. El algoritmo se basa en asignar un peso a cada atributo y seleccionar los atributos cuyo peso supera un umbral prefijado. Para este propósito, dada una instancia, Relief busca en el conjunto de datos vecinos, el más cercano de la misma clase y el de clase distinta. El peso asociado a un atributo se modifica a partir de la distancia euclidiana entre el valor del atributo de la instancia y el valor del mismo atributo de los vecinos encontrados. Las instancias se escogen aleatoriamente del conjunto de datos, un número determinado de veces. Cuando el número de instancias de la base de datos es pequeño, se realiza el proceso para cada una de las instancias, no existiendo aleatoriedad, por lo que en algunas bibliografías se incluye en el grupo de búsqueda secuencial. Relief favorece a los atributos correlacionados sobre los relevantes, por lo que no garantiza un conjunto óptimo. (3)

Métodos Híbridos: con la hibridación de técnicas se pretende explotar las ventajas de unos métodos, eliminando sus inconvenientes. De entre estas técnicas se puede mencionar:

- Quick Branch & Bound

Aproximación Incremental: estas técnicas se basan en la idea de llevar a cabo la selección del subconjunto de características sin utilizar el conjunto completo de instancias que se dispone. Se pretende con ello hacer frente al problema que aparece en los algoritmos de selección de características cuando se enfrentan a conjuntos de datos de elevado tamaño. Como técnica representativa es válido citar:

- Las Vegas Incremental

Búsqueda secuencial: estas técnicas toman decisiones sin replantearse las anteriores y se caracterizan por su simplicidad y eficiencia, por lo que habitualmente se emplean para implementar heurísticas. En el caso de los procedimientos de búsqueda, siguen un camino sin volver nunca hacia atrás. También se les denomina búsqueda secuencial porque siguen una secuencia de pasos sin vuelta atrás. Siendo estos los algoritmos de búsquedas más simples debido a que su exploración en complejidad es polinomial; por lo que en espacios de búsqueda con gran cantidad de variables independientes tienden a convergen a máximos locales, lo que hace que no lleguen a explorar todo el espacio muestral.

Existen varios algoritmos secuenciales entre los que se encuentran el SFS (búsqueda secuencial hacia delante), que parte del conjunto vacío de características y va añadiendo la variable que en mayor medida mejora la selección. Por otro lado, está el que hace la búsqueda en sentido inverso SBS (búsqueda secuencial hacia atrás), partiendo del conjunto de todas las características y eliminando las de menor significación. Pueden además encontrarse algoritmos que combinan ambas estrategias, conocidos como bidireccionales para tratar de generar estados que son imposibles de hallar mediante los algoritmos anteriores.

Estos algoritmos de búsqueda arrojan resultados significativos cuando la cantidad de variables independientes existentes en la muestra no alcanzan una gran dimensión, proporcionando de esta manera valores próximos al máximo global.

1.7 Medidas de Evaluación

En todo proceso de selección de características, se hace necesario valorar la utilidad de estas en la resolución del problema de aprendizaje a abordar. En la mayor parte de los métodos, puede hacerse la valoración de forma independiente al proceso de búsqueda.

Hay dos tipos de aproximaciones principales al problema de la valoración de relevancia de las características. La primera es valorar cada una de las características de forma independiente. La ventaja de este tipo de medidas es que suelen ser medidas bastante simples y como solo se pueden hacer tantas valoraciones, como características tenga el conjunto de datos; apoyarse en estas medidas es muy rápido. Como contrapartida, el inconveniente principal es que estas medidas no aportan ninguna información sobre la posible relación existente entre todo el subconjunto. La segunda aproximación a la valoración de características es trabajar con conjuntos de las mismas. En este caso, al valorar conjuntos completos, se puede obtener información sobre las posibles interrelaciones entre características y de esta forma averiguar, por ejemplo, si la inclusión de una característica aporta algo o es completamente redundante a

las de un conjunto dado. En cambio, el inconveniente es que valorar todos los posibles subconjuntos de características normalmente no es factible.

1.7.1 Medidas de evaluación sobre variables individuales

Entre las muchas medidas de evaluación de características individuales se describen las más comúnmente usadas.

Ganancia de información (información mutua)

De la teoría de la información, se usa la cantidad de información que aporta una característica sobre la clase a predecir, para valorar la relevancia de dicha característica. Quinlan utilizaba la información mutua para elegir las características que dividirán nodos en generación de árboles. (16)

Gain Ratio

La medida anterior de ganancia de información favorece a las características con muchos valores. Puede ocurrir que esta sobre-estimación no sea un comportamiento deseable y para evitarlo se puede usar como medida el ratio entre la ganancia de información y la entropía de la característica.

Índice de Gini

El índice de Gini (Gini índex) toma su nombre del estadístico italiano Conrado Gini. Conocido como una medida de la desigualdad usada en economía (17), introducido para la generación de árboles de clasificación y regresión por Breiman (18). Interpretando como probabilidad que dos ejemplos elegidos aleatoriamente tengan una clase diferente.

ReliefF

Originalmente, Kira y Rendell (19) propusieron Relief como un método de selección de características. Estaba basado en una valoración novedosa de las características y en la elección de aquellas que obtuviesen una valoración mayor que un umbral dado. La valoración original de relevancia de las características solo estaba definida para problemas lógicos (cuyo resultado solo puede ser verdadero o falso) pero posteriormente, Kononenko desarrolló extensiones (20) que pueden trabajar con problemas de clasificación y tolerar valores nulos.

1.7.2 Medidas de evaluación sobre conjuntos variables

Las medidas sobre conjuntos de características son funciones que; dado un conjunto de datos de entrenamiento $T \in R$, donde se denomina R a todos los posibles conjuntos de entrenamiento y un subconjunto de características $(S \subset P(F))$, devuelven una valoración de la relevancia de esas características. El resultado será un número real, normalmente dentro de un intervalo, como $[0,1]$ ó $[-1,1]$, pero también puede ser un resultado discreto, por ejemplo en $\{0,1\}$, representando un valor booleano que indique si el conjunto es aceptable o no como resultado de la selección.

Hay una gran variedad de medidas de relevancia para conjuntos de características. Algunas trabajan tanto con características discretas como continuas, otras solo aceptan uno de los dos tipos. Los algoritmos que las calculan pueden ser exactos o aproximados, determinísticos o no. Las medidas cumplen o no diversas propiedades como la monotonía o la invariabilidad a transformaciones lineales.

Las categorías identificadas son: medidas de distancia, medidas de información, medidas de dependencia, medidas de consistencia y las basadas en el porcentaje de acierto del algoritmo de aprendizaje. Se describe a continuación las medidas más importantes de cada una de las categorías.

1.7.2.1 Medidas basadas en consistencia

Para poder predecir correctamente la clase asociada a las instancias de un conjunto de datos, es necesario que este sea consistente. Un conjunto de datos se considera consistente siempre que en él no haya ningún par de instancias que perteneciendo a clases distintas, tengan los mismos valores en todas sus características. Si en un conjunto de datos se eliminan algunas características, dejando solo las seleccionadas, habrá menos valores que diferencien las instancias y por tanto, podrán aparecer más casos de inconsistencia. La idea que persiguen las medidas basadas en consistencia es valorar el nivel de consistencia del conjunto de datos que tiene únicamente las características seleccionadas.

Como siempre que se aumenta el número de características, aumenta el número de hipótesis consistentes que se pueden definir, el requisito de presentar consistencia suele acompañarse con el de tener un número reducido de características. En cualquier caso, la búsqueda de un conjunto de características pequeño es un objetivo común de todos los métodos de selección de características. Así esta estrategia no es una particularidad exclusiva de los métodos basados en medidas de consistencia, sino que también es aplicada en algoritmos de búsqueda que se usan con otros tipos de medidas.

Hay otros métodos basados en consistencia que aunque no definen medidas específicas, puede ser interesante tener en cuenta. Schlimmer (21) describe un algoritmo para deducir determinaciones lógicas usando el menor número posible de características, es un algoritmo de selección de características inmerso. También existen otras medidas como las de consistencia básica (22), la consistencia de Liu (23) y la consistencia de la teoría de Rougth Sets (24).

1.7.2.2 Medidas basadas en la Teoría de la Información

Las medidas de este apartado se basan en la teoría de la información de Shannon (25). Midiendo la información que aportan las características sobre la clase se puede saber cuáles son más informativas siendo estas, desde el punto de vista de la teoría de la información, las

más apropiadas para la clasificación. Muchos algoritmos de aprendizaje se basan en principios de la teoría de la información, lo que además de indicar que el uso de estas medidas es prometedor, lleva a pensar que habrá una sinergia positiva entre los métodos de selección de características que usen estas medidas y los algoritmos de aprendizaje basados en teoría de información. Algunas de estas medidas como la Incertidumbre Simétrica y la Información Mutua se describen a continuación.

Información Mutua

La teoría de la información establece una forma básica de medir la información que aporta el conocimiento de los valores que toman una o más variables sobre otra. Sea C la variable aleatoria que define la clase de un problema de clasificación. La entropía de C viene dada por:

$$H(C) = -\sum_{c \in C} p(c) \log p(c) \qquad \text{(I)}$$

El objetivo del algoritmo de aprendizaje es reducir la incertidumbre sobre el valor de la clase. Para ello, el conjunto de características seleccionadas S aporta la cantidad de información dada por:

$$I(C,S) = H(C) - H(C \mid S) \qquad \text{(II)}$$

Lo ideal sería encontrar el menor conjunto de características que determine completamente C, esto es $I(C,S) = H(C)$, pero no siempre es posible. Esta medida cumple la propiedad de la monotonía. Al añadir una característica más, esta siempre aportará algo de información o en el peor de los casos, nada, pero nunca reducirá la información que aportan las características ya seleccionadas.

Los algoritmos de búsqueda que usen esta medida deberán tener en cuenta la propiedad de monotonía, pues en un conjunto de datos no completamente determinado se tenderá a seleccionar todas las características. Así se debe buscar un conjunto, que aporte mucha información, pero con un número reducido de características.

Incertidumbre Simétrica

La medida de información mutua es simétrica, de (II) se tiene:

$$I(C,S) = H(C) - H(C \mid S) = H(S) - H(S \mid C) = I(S,C) \qquad \text{(III)}$$

Por otra parte, la medida de información mutua tiende a dar mayor valor a las características con más valores. Si se desea que todas las características sean valoradas equitativamente, se puede usar:

$$U(C,S) = \frac{H(C) - H(C \mid S)}{H(S)} \qquad \text{(IV)}$$

Sin embargo, con esta definición, la medida dejaría de ser simétrica. Por esta razón, se define la medida de incertidumbre simétrica (24) como:

$$SU(C,S) = 2 \cdot \frac{H(C) - H(C \mid S)}{H(S) + H(C)} \qquad \text{(V)}$$

Intuitivamente, esta medida puede interpretarse como la razón (ratio) entre la cantidad de información que aportan las características seleccionadas y la cantidad de información total que contienen y por tanto la información que podrá aportar. Al ser un ratio, cuyo denominador

puede crecer más rápido que el numerador al incluir características, esta medida no cumple la propiedad de monotonía.

MDL (Longitud Mínima de Descripción)

Las medidas con la propiedad de monotonía tienen el inconveniente de no indicar directamente que características son completamente irrelevantes, pues al añadir una característica más, esta incrementa la medida aunque sea muy ligeramente. La medida MDL (27) pretende resolver este problema aplicando el criterio MDLC (longitud mínima de la descripción de un modelo). Este criterio sostiene que entre varios modelos de ajuste se debe elegir el que tenga una descripción mas corta.

Se puede ver como el principio de la navaja de Ockham aplicado a la teoría de la información. De esta forma, se tiene en cuenta que cuantas más características hay, más complejo es el modelo; permitiéndose discernir cuando ya no merece la pena usar más características.

1.7.2.3 Medidas Basadas en la Distancia

Estas medidas valoran las distancias que hay entre las distribuciones de probabilidad de cada clase. La idea detrás de estas medidas es que aquellas características que hagan mayor la distancia entre las distribuciones las separaran mejor y por tanto deben permitir hacer mejor la clasificación. (28)

1.7.2.4 Medidas Basadas en la Dependencia

CFS (Correlación Basada en Selección de Características)

CFS es una heurística para evaluar subconjuntos de atributos pero que al mismo tiempo, tiene en cuenta el valor predictivo de cada elemento del subconjunto sobre la clase y la intercorrelación entre estos. Así la hipótesis que plantea esta heurística es la siguiente:

Buenos subconjuntos de atributos, poseen una alta relación con el atributo clase y una baja relación entre estos.

Esta medida de evaluación se formaliza en la siguiente ecuación:

$$CFS = \frac{k \cdot \bar{r}_{ic}}{\sqrt{k + k(k-1) \cdot \bar{r}_{ij}}} \qquad \text{(VI)}$$

Donde:

- ✓ $\bar{r}_{ic}$ representa la correlación promedio de los elementos del subconjunto con respecto a la variable de salida (correlación intra-clase)
- ✓ $\bar{r}_{ij}$ representa la correlación promedio de los elementos del subconjunto (correlación inter-clase)
- ✓ k representa la cardinalidad del subconjunto

Esta heurística es aplicable tanto a modelos lineales como no lineales, donde para el primer caso su valor depende del cálculo de los coeficientes de Pearson **"r"**. Esta medida permite eliminar elementos redundantes y poco significativos. Es inversamente proporcional a la cardinalidad del subconjunto, si aumenta el número de elementos entonces disminuye el valor (mérito) de CFS.

Según la descomposición modular del procedimiento de selección de variables o características, todo subconjunto seleccionado por el método de búsqueda se le emplea una medida de evaluación en correspondencia con la naturaleza de los datos y las propiedades de los algoritmos; arrojando como resultado final las variables seleccionadas. Según las propiedades tanto de las medidas como de los algoritmos, se les hace corresponder a estos últimos una o varias medidas de evaluación. Tal es el caso de los algoritmos: SFS, SBF con las medidas basadas en distancia, LVF, LVI con las medidas basadas en consistencia entre otros; siendo Algoritmos Genéticos y Enfriamiento Simulados los únicos que pueden emplear

todas las medidas. Para determinar la calidad del modelo generado se hace necesario el uso de los clasificadores.

1.8 Software vinculados a la Selección de Variables a nivel mundial

RapidMiner (anteriormente Yale)

Es un software de código abierto para el análisis inteligente de datos, descubrimiento de conocimientos, minería de datos, aprendizaje automático, visualización; con numerosas características y funciones para la selección de variables. Constituye además un entorno de aprendizaje automático y de extracción de datos para todo tipo de experimentos. Permite que los experimentos sean realizados con un gran número de variables arbitrarias, las cuales se escriben en archivos XML que son fácilmente creados con la interfaz gráfica de RapidMiner. Ofrece más de 400 operadores para los principales procedimientos de aprendizaje de máquinas, incluidos los de entrada, salida, pre procesamiento de datos y visualización de los mismos.

Está escrito en el lenguaje de programación Java y por tanto pueden trabajar en todos los Sistemas Operativos populares. También integra todos los sistemas de aprendizaje y de atributo de los evaluadores Weka. Cuenta con una licencia GNU GPL, Propietaria y Comercial.

Keel

Es un software para evaluar la evolución de los algoritmos de minería de datos y problemas de regresión, entre ellos: clasificación, agrupamiento, patrón de la minería. Contiene una gran colección de algoritmos clásicos de extracción de conocimientos, técnicas de pre procesamiento (selección de instancias, selección de características, discretización, métodos de imputación de valores), Inteligencia Computacional de aprendizaje basado en algoritmos; incluido el estado evolutivo de algoritmos de aprendizaje basados en diferentes enfoques (Pittsburgh, Michigan y IRL) y modelos híbridos como sistemas difusos genéticos, redes neuronales evolutivas. Permite realizar un análisis completo de cualquier modelo de

aprendizaje en comparación con los existentes, incluido un módulo de prueba estadística para la comparación entre ellos. El uso más común de esta herramienta para un investigador será la ejecución automatizada de los experimentos y el análisis estadístico de sus resultados. Esta herramienta no está diseñada para ofrecer un tiempo real del progreso de los algoritmos. Trabaja muy bien en ambiente distribuido de sistemas. Fue diseñado con doble objetivo: la investigación y la educación. Cuenta con licencia comercial, lo que lo convierte en Software propietario.

Weka

Es un paquete de software de Java para la extracción de conocimientos desde bases de datos; incluye además una recopilación de algoritmos de aprendizaje automático para tareas de minería de datos. Este software ha sido desarrollado en la universidad de Waikato (Nueva Zelanda) bajo la licencia GPL, lo que significa que este programa es de libre distribución y difusión; lo cual ha impulsado que sea una de las suites más utilizadas en el área en los últimos años. Es de gran utilidad al ser utilizado mediante las interfaces que ofrece o para embeberlo dentro de cualquier aplicación.

Además Weka contiene las herramientas necesarias para realizar transformaciones sobre los datos, tareas de clasificación, regresión, clustering, asociación y visualización. Está diseñado como una herramienta orientada a la extensibilidad por lo que añadir nuevas funcionalidades es una tarea sencilla.

Sin embargo tiene un gran defecto y es la escasa documentación orientada al usuario que tiene junto a una usabilidad bastante pobre, lo que la hace una herramienta difícil de comprender y manejar sin información adicional. Además como Weka está programado en Java, es independiente de la arquitectura, ya que funciona en cualquier plataforma sobre la que haya una máquina virtual de Java disponible.

Una de las propiedades más interesantes de este software, es su facilidad para añadir extensiones y modificar sus métodos.

1.9 Clasificación Supervisada

A la hora de resolver los problemas de clasificación supervisada, los campos de la estadística y el aprendizaje automático han desarrollado diferentes técnicas: Análisis Discriminante, Redes Neuronales, Clasificadores K-NN, Sistemas Clasificadores, Árboles de Clasificación, Regresión Logística, inducción de reglas y Máquinas de Soporte Vectorial entre otras.

1.9.1 Clasificador Máquinas de Soporte Vectorial

El algoritmo Vector de Soporte (VS) es una generalización no-lineal del algoritmo Semblanza Generalizada, desarrollado en la Rusia en los años sesenta. Está firmemente enlazado a la Teoría del Aprendizaje Estadístico, la cual se desarrolló a finales de las últimas tres décadas por Vapnik y Chervonenkis. Por otra parte, esta Teoría de Aprendizaje Estadístico caracteriza propiedades de aprendizaje, habilitándole el poder de generalización de datos desconocidos. El desarrollo de los VS trae consigo el surgimiento de las Máquinas de Soporte Vectorial (MSV). Estos sistemas de aprendizaje que usan un espacio de hipótesis de funciones lineales en un espacio de rasgos de mayor dimensión, entrenadas por un algoritmo proveniente de la teoría de optimización. Esta técnica de inteligencia artificial se enmarca dentro de las Redes Neuronales de aprendizaje supervisado, siendo un clasificador eficiente en modelos donde la dependencia existente entre los datos es desconocida y le permita la generalización de los mismos.

Las MSV pertenecen a la familia de clasificadores lineales puesto que inducen separadores lineales o hiperplanos en espacios de características de muy alta dimensión introducidos por funciones kernels con un sesgo inductivo muy particular, la maximización del margen (29). Sin embargo, la formulación matemática de las Máquinas de Soporte Vectorial varía dependiendo de la naturaleza de los datos; es decir, existe una formulación para los casos lineales y, por otro lado, una formulación para casos no lineales. Es importante tener claro que, de manera general para clasificación, las Máquinas de Vectores Soporte buscan encontrar un hiperplano óptimo que separe las clases.

1.9.1.1 Tipos de Máquinas de Soporte Vectorial

Para construir un hiperplano óptimo, las MSV emplean un algoritmo de entrenamiento iterativo else emplea para minimizar una función error. Acorde a la forma de la función error los prototipos de MSV pueden ser clasificados en cuatro grupos diferentes.

Clasificación MSV Tipo 1 (también conocido como clasificación C-SVC)

Clasificación MSV Tipo 2 (también conocido como clasificación nu-SVC)

Regresión MSV Tipo 1 (también conocido como regresión épsilon-SVR)

Regresión MSV Tipo 2 (también conocido como regresión nu-SVR)

1.9.2 Evaluación del rendimiento de una clasificador

Existen diferentes parámetros que evalúan la eficiencia del clasificador validando así la calidad del modelo. Las medidas más conocidas para evaluar la clasificación están basadas en la matriz de confusión que se obtiene al aplicar el clasificador en el conjunto de datos del entrenamiento **(Tabla2)**.

Tabla 2: Matriz de Confusión

	Positivos	Negativos
Positivos	Verdaderos Positivos(VP)	Falsos Positivos(FP)
Negativos	Falsos Negativos(FN)	Verdaderos Negativos(VN)

Se consideran verdaderos positivos aquellos compuestos cuyos valores de actividad son positivos bien clasificados mientras que los verdaderos negativos son aquellos compuestos cuyos valores negativos de actividad son igualmente bien clasificados. Mediante un análisis contrario, se definen los falsos positivos y los falsos negativos. Basados en esta matriz, se calculan la exactitud y la precisión mediante las ecuaciones 1 y 2. (30)

Ecuación 1: $Exactitud = (VP + VN)/(VP + FP + VN + FN)$

Ecuación 2: $\Pr ecisión = (VP + VN)/(VP + FP + VN + FN)$

Evaluar el comportamiento de los algoritmos de aprendizaje es un aspecto fundamental del aprendizaje automático, no solo es importante para comparar algoritmos entre sí, sino que en muchos casos forma parte del propio algoritmo de aprendizaje. La forma más habitual de medir la eficiencia de un clasificador es la precisión predictiva.

La precisión es una buena estimación de cómo se va a comportar el modelo para datos desconocidos similares a los de prueba. Sin embargo, si se calcula la precisión sobre el propio conjunto de datos utilizado para generar el modelo, se obtiene con frecuencia estimaciones muy optimistas por utilizar los mismos ejemplos en la inducción del algoritmo y en su comprobación. Por los que se hace necesario dividir la muestra en datos de entrenamiento y prueba.

Dentro de las técnicas de evaluación, validación cruzada, se emplea para evitar la ocultación de parte de las muestras al algoritmo y la consiguiente perdida de información. Con esta técnica el conjunto de datos se divide en *k* particiones mutuamente exclusivas, conteniendo todo aproximadamente el mismo número de ejemplos, indicándose en muchos casos como validación cruzada con *k* particiones. En cada evaluación, se deja uno de los subconjuntos para la prueba, y se entrena el sistema con los *k*-1 restantes.

Otra forma de evaluar el rendimiento de un clasificador es por las curvas ROC. En esta curva se representa el valor de la razón de VP contra la razón de FP, mediante la variación del umbral de decisión. Se denomina umbral de decisión a aquel que decide si una instancia x, a partir del vector de salida del clasificador, pertenece o no a cada una de las clases. Esta última y la precisión serán los criterios empleados para evaluar el clasificador.

Conclusiones:

La selección de variables es un problema altamente combinatorio para seleccionar el número de subconjuntos a evaluar. Dentro de la bibliografía consultada se identificaron cuatro clasificaciones de métodos de selección:

- Metodología de Filtro
- Estrategia Envolvente
- Inversa
- Híbrida

A partir de su potencial en la descomposición modular de los algoritmos y mantener los métodos de búsqueda y evaluación independientes el uno del otro y su eficiencia, se empleará la Metodología de Filtro. Los métodos de búsqueda para explorar el espacio muestral se agrupan según sus características en:

- Búsqueda Secuencial
- Búsqueda Completa
- Búsqueda Aleatoria

A partir de la complejidad de los algoritmos que la componen y su capacidad de llegar al óptimo global, la búsqueda seleccionada es la aleatoria, específicamente los Algoritmos Genéticos y Enfriamiento Simulado. Aunque para las muestras donde la cantidad de variables independientes no sea elevada, los algoritmos de búsqueda secuencial desempeñan un papel importante en la obtención del óptimo global. Las medidas de evaluación seleccionadas son: Consistencia, Correlación basada en Selección de Características (CFS). Se emplean de las Máquinas de Soporte Vectorial, las de Tipo 1 y 2 de clasificación para la crear los modelos de clasificación.

CAPÍTULO 2

Métodos y Programas

Teniendo como precedente las características de esta investigación; en la sección de métodos se describirán las características de los algoritmos a emplear para explorar el espacio de búsqueda, así como las características de las muestras. En la sección de programas se identificarán las herramientas informáticas y el lenguaje de programación a utilizar.

Métodos

2.1 Los Algoritmos Genéticos

Los algoritmos genéticos son procesos de búsqueda basados en los principios de la selección y la evolución natural. Las posibles soluciones a un problema son codificadas en forma de cadenas binarias, y la búsqueda se inicia con una población de posibles soluciones generadas aleatoriamente. (31)

Los algoritmos genéticos son algoritmos matemáticos altamente paralelos que transforman un conjunto de objetos matemáticos individuales con respecto al tiempo. Estos usan operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto y tras haberse presentado de forma natural una serie de operaciones genéticas de entre las que destaca la recombinación sexual. Cada uno de estos objetos matemáticos suele ser una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo de las cadenas de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud.

2.1.1 Pasos para construir un Algoritmo Genético. Propuesta de seudocódigo

- ✓ Diseñar una representación
- ✓ Decidir cómo inicializar una población
- ✓ Diseñar una forma de evaluar un individuo
- ✓ Diseñar un operador de mutación adecuado
- ✓ Diseñar un operador de cruce adecuado
- ✓ Decidir cómo seleccionar los individuos para ser padres
- ✓ Decidir cómo reemplazar a los individuos

- ✓ Decidir la condición de parada

Definiéndose como parámetros fundamentales a introducir: número de población y generaciones, probabilidad de mutación y cruce.

2.1.2 Algunos esquemas de selección

Se debe garantizar que los mejores individuos tengan una mayor posibilidad de ser padres (reproducirse) frente a los individuos menos buenos. Se debe ser cuidadoso para dar una oportunidad de reproducirse a estos últimos. Estos pueden incluir material genético útil en el proceso de reproducción. Esta idea define la presión selectiva que determina en qué grado la reproducción está dirigida por los mejores individuos. Existen varios esquemas de selección dentro de los más empleados se encuentran:

Selección por Torneo (TS): escoge al individuo de mejor aptitud de entre *N* individuos seleccionados aleatoriamente $(N = 2,3,\ldots)$.

Orden Lineal (LR): la población se ordena en función de su aptitud y se asocia una probabilidad de selección a cada individuo que depende de su orden.

Selección Aleatoria (RS): un padre lo escoge aleatoriamente, para el otro selecciona *N* padres y escoge el más lejano al primer $(N = 3,5,\ldots)$. Está orientado a generar diversidad.

Selección por Ruleta: se asigna una probabilidad de selección proporcional al valor de aptitud del cromosoma.

Siendo este último el esquema de selección empleado, en la **Ilustración 1** se muestra el seudocódigo del mismo**.**

```
function GeneticSearch(eval)

  t := 0;
  Inicializar P(t) ;
  Evaluar P(t) ;
  Escalar P(t) ;
  Obtener_Mejor_Individuo P(t) ;
  Para t := 1 hasta cantidad Generaciones(max) hacer :
      Seleccionar P(t) desde P(t − 1) ;
      Cruzar P(t) ;
      Mutar P(t) ;
      Evaluar P(t) ;
      Escalar P(t) ;
      converge := Obtener_Mejor_Individuo P(t) ;
      Estadisticas P(t) ;
Si (i = max) or (converge = true) entonces
        break;
      fin Si
  fin Para
  atributos := Listar(Mejor Individuo);
  return atributos;
fin funcion

Donde, P(t) es la población en la iteración t.
```

Ilustración 1

2.2 Algoritmo de Enfriamiento Simulado

El enfriamiento simulado (*Simulated Annealing* (SA)) (32) (12) es una meta-heurística para problemas de optimización global que se basa en conceptos de la mecánica estadística y es una generalización del Método de Monte Carlo. Fue propuesto por primera vez por Metrópolis (33) y usado en optimización combinatoria por Kirkpatrick (34). Este método heurístico se basa en los conceptos descritos originalmente por el proceso físico sufrido por un sólido al ser sometido a un baño térmico.

Se sabe en ingeniería, que una manera de encontrar los estados de energía de sistemas complejos, tales como sólidos, consiste en utilizar la técnica de enfriamiento, en la que el sistema se calienta primero a una temperatura en la que sus granos deformados recristalizan para producir nuevos granos; luego se enfría suavemente y de esta manera, cada vez que se

baja la temperatura, las partículas se reacomodan en estados de más baja energía; hasta que se obtiene un sólido con sus partículas acomodadas conforme a una estructura de cristal (estado fundamental). En la fase de enfriamiento, para cada valor de la temperatura, debe permitirse que el sistema alcance su equilibrio térmico (35).

De forma análoga, en el algoritmo de enfriamiento simulado los estados del sistema corresponden a las soluciones del problema, la energía de los estados a los criterios de evaluación de la calidad de la solución (generalmente se utiliza la función objetivo), el estado fundamental a la solución óptima del problema, los estados meta estables a los óptimos locales, y la temperatura a una variable de control. *"El éxito del Enfriamiento Simulado se basa en la escogencia de una buena temperatura inicial y una adecuada velocidad de enfriamiento."* (36)

"La característica principal de este algoritmo es que al buscar una nueva solución S_{n+1} dada una solución S_n, acepta en ocasiones una de inferior calidad a la de Sn por medio de una función probabilística la cual depende del parámetro variable de temperatura y de la calidad ofrecida por las dos soluciones S_n y S_{n+1}. Mientras más bajo sea el parámetro de temperatura, menor será la probabilidad de aceptar una solución peor, y viceversa." (36)

Enfriamiento Simulado es una poderosa herramienta de búsqueda estocástica que se ha hecho muy popular dado el amplio espectro de problemas que puede resolver. En particular en el área de la optimización combinatoria y la selección de variables o de características. En la **Ilustración 2** se muestra la estructura del mismo.

```
function SimmulatedAnnealing (T0, Tf, k, nVecinos)
      T = T0
      Sactual = Genera solución aleatoria;
      Mientras T >= Tf hacer:
          Para i en nVecinos (T) hacer:
              Scandidata = Genera un vecino (Sactual)
              λ = coste (Scandidata) − coste (Sactual)
              Si U (0, 1) < e−λ/T or λ < 0 entonces:
                Sactual = Scandidata;
              fin Si
          fin Para
          Estadisticas();
          T = k*(T);
       fin Mientras
      atributos := Listar(SActual);
      return atributos;
fin function

Donde:
Sactual : solución actual
Scandidata: solución candidata
T0 es la temperatura inicial
Tf: la temperatura final
k: es el coeficiente de enfriamiento elegido
nVecinos(T ): el número de vecinos generados en cada ciclo según T
U(0,1) :es un generador de números aleatorios uniformemente distribuidos
```

Ilustración 2

2.3 Características de las Muestras

2.3.1 Cefalosporinas

Las cefalosporinas son compuestos antibacteriales pertenecientes a la familia de los β-lactámicos. Todas las cefalosporinas se derivan de la cefalosporina C, un antibiótico natural producido por la cepa de *Cephalosporium acremonium* aislado por primera vez en 1945. Las cefalosporinas se parecen a las penicilinas[i] en que tienen un anillo β-lactámico, pero en lugar del anillo de 5 miembros de tiazolidina, presentan un anillo de dihidrotiazina como se muestra en la **Figura 2**. .

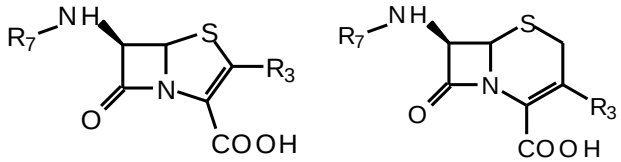

Figura 2: Esquema general de la estructura de a) penicilinas y, b) cefalosporinas

Las cefalosporinas poseen un amplio rango de actividad antibacterial, una excelente tolerancia en niños y casi ninguna toxicidad asociada a las dosis. Estos antibióticos pueden emplearse con seguridad en niños de todas las edades con fallos renales o hepáticos.

La muestra empleada consta de 104 compuestos pertenecientes a cuatro generaciones distintas de cefalosporinas, y sus estructuras y valores de actividad se muestran en la tabla 1 de los anexos.

2.3.2 Inhibidores del Factor Esteroidogénico1

Contraensayo para inhibidores del Factor 1 del receptor nuclear esteroidogénico (SF-1). Un ensayo de dosis-respuesta basado en células para la inhibición del receptor huérfano *A* relacionado a RAR (RORA).

Tipo de ensayo: confirmatorio. Se observa relación concentración-respuesta.

219 activos

138 inactivos

El receptor nuclear SF-1 (factor esteroidogenico-1) pertenece a la clase de receptores nucleares huérfanos que han sido poco investigados al nivel farmacológico (celular) que se reporta.

El SF-1 se expresa en las glándulas adrenal, pituitaria, testículos, y ovarios y regula la producción de la hormona esteroidea a diferentes niveles, incluyendo la expresión directa de la enzima P-450 principal involucrada en la síntesis de la hormona esteroidea.

Para explorar el potencial del SF-1 como diana para nuevos fármacos, se identificaron pequeñas moléculas como ligando. Con este fin, se realizó el tamizaje de la biblioteca MLSCN mediante un ensayo basado en células desarrollado por Orphagen Pharmaceuticals (San Diego, CA).

Los ligandos para el SF-1 pueden tener aplicaiones clínicas como moduladores de síntesis esteroidea adrenal. Se predice, en particular, que el diseño apropiado de antagonistas del SF-1 tienen utilidad terapéutica en el tratamiento del cáncer de próstata metastásico a través de la supresión, tanto de la síntesis de testosterona gonadal como de andrógeno adrenal. Otro beneficio potencial de este esfuerzo puede ser la identificación de ligandos del SF-1 que pudieran convertirse en una nueva clase de pequeñas moléculas reguladoras del metabolismo energético y la obesidad.

Programas

2.4 Herramientas para el desarrollo del sistema

2.4.1 Plataforma de Desarrollo y Lenguaje de Programación

En cuanto a plataforma de desarrollo se optó por jdk version 1.5.0_10, mientras que como lenguaje de programación se decidió utilizar Java. Este es un lenguaje de programación orientado a objetos, desarrollado por Sun Mycrosystems a principios de los años 90. El lenguaje en sí mismo toma mucha de la sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Entre sus características fundamentales se encuentran:

- ✓Usa la metodología de la programación orientada a objetos
- ✓Permite la ejecución de un mismo programa en múltiples sistemas operativos
- ✓Incluye por defecto soporte para trabajo en red
- ✓Está diseñado para ejecutar código en sistemas remotos de forma segura
- ✓Es fácil de usar y toma lo mejor de otros lenguajes orientados a objetos, como C++
- ✓Es independiente de la plataforma en la que se ejecuta, esto significa que programas

escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware, tal como reza el axioma de Java, ”write once, run everywhere”.(“escrito una vez, corre donde sea”).

2.4.2 Entorno de Desarrollo

Se utilizó Eclipse como entorno de desarrollo en su versión 3.4, plataforma de software de código abierto y extensible. Ha sido usada para desarrollar entornos integrados de desarrollo, como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se embarca como parte de Eclipse y que son usados también para desarrollar este entorno. Sin embargo, para otros tipos de aplicaciones cliente, además de ser un entorno de desarrollo integrado, ofrece el control del editor de códigos, del compilador y del depurador desde una única interfaz de usuario. Además Eclipse es una plataforma universal para integrar herramientas de desarrollo basada en plug-ins.

2.4.3 Herramientas acopladas a la investigación

Weka

Se decidió utilizar como herramienta para apoyar en la investigación a Weka, que es un software que posee una colección extensa de algoritmos de máquinas de conocimiento, conteniendo las herramientas necesarias para la realización de minería de datos, transformaciones necesarias en los datos, tareas de clasificación, regresión, clustering, asociación y visualización.

Librería de las Máquinas de Soporte Vectorial (LibSVM):

Software integrado para la clasificación, regresión, estimación de la distribución de los datos y soporta la clasificación multiclase empleando las MSV. Dentro de sus prestaciones se encuentran:

- Diferentes formularios de MSV
- Validación para la selección de los modelos

- Estimaciones Probabilísticas

En primer lugar, el entrenamiento de datos es separado en varios segmentos. Secuencialmente un segmento está considerado como el conjunto de validación y el resto son para el entrenamiento.

Los tipos de MSV son:

- C-SVC
- nu-SVC
- one-class SVM
- épsilon-SVR
- nu-SVRT

Los tipos de Kernels son:

- Lineal: K(u, v) = u' * v
- Polinomial: K(u, v) = (gamma * u' * v + coef0)^degree
- Función de Base Radial (RBF): K(u, v) = exp(-gamma * |u – v|^2)
- Sigmoidal: K(u, v) = tanh (gamma * u * v + coef0)

Esta librería brinda la posibilidad de integrarse al software Weka, permitiendo una mejor interpretación de los resultados y usabililidad.

CAPÍTULO 3

Resultados y Discusión

3.1 Algoritmos Implementados

3.1.1 Algoritmo Genético

Para la implementación de este algoritmo se llega a un conjunto de aspectos fundamentales a tener en cuenta que se relacionan a continuación:

- ✓ Diseñar una representación eficiente para optimizar el procedimiento de búsqueda, o sea codificar las posibles soluciones del problema en forma de cadenas binarias; donde cada una de estas (individuos) constituyen un subconjunto de variables candidato a ser solución. Así, dado un conjunto potencia *U*, tal que $U = \{A, B, C, D, E, F, G\}$ con cardinalidad 7, un subconjunto $U' = \{B, D, G\} \in U$; este se representa por la siguiente cadena binaria $L = [0101001]$ donde el valor '1' representa la presencia de variables, y el valor '0', representa la ausencia de variables en el subconjunto.

- ✓ Decidir cómo inicializar una población para garantizar una mayor diversidad de soluciones y el procedimiento no converja prematuramente. En este sentido, se tienen dos variantes de solución, una es inicializar aleatoriamente toda la población, y la otra es sembrar un individuo prefijado en la población inicial, para acelerar el proceso evolutivo.

- ✓ Diseñar la forma de evaluar un individuo. Se trata de atribuir numéricamente las aptitudes de los cromosomas en una población, asignarle el valor numérico del resultado de la evaluación de la función objetivo. Existen sin embargo, dos problemas importantes asociados a este método, como son la **competición próxima** (individuos cuya aptitud relativa son próximas numéricamente) y el efecto de **súper individuos** (individuos con evaluación muy superior a la media, capaces de dominar el proceso de selección, haciendo que el AG converja prematuramente hacia un óptimo local). Para resolver estos problemas, se desarrollaron métodos de transformación numérica de evaluación de aptitud de cada subconjunto generado, como son las técnicas de escalado lineal y potencial. Estas garantizan, que un

elemento típico de la población contribuya en promedio con un descendiente de la próxima generación, proporcionando a su vez una mayor variedad de esquemas.

- ✓ Diseñar operadores genéticos que proporcionen un mecanismo estructurado de intercambio de información útil (bloques constructivos para el cruzamiento) entre individuos y a su vez explorar nuevas zonas del espacio de búsqueda y permita escapar de máximos locales.

- ✓ Decidir cómo seleccionar los individuos (soluciones) para la próxima generación de modo que contribuya geométricamente a la presencia de esquemas aventajados y reducir la presencia de los retrasados. Dada la variedad existente de esquemas de selección, se desarrolla un mecanismo de selección elitista combinado con selección por ruleta, donde las dos mejores soluciones de la población actual son insertadas en la siguiente generación para mejorar el proceso de convergencia del algoritmo, y los restantes individuos son seleccionados probabilísticamente en proporción a la aptitud que estos posean.

- ✓ Decidir la condición de parada que puede ser por un número fijado de generaciones o cuando el algoritmo converge a una misma solución, lo cual se explica cuando toda la población posea una misma solución.

A partir de los aspectos anteriores la implementación del algoritmo se muestra a continuación:

```
public int[] search(ASEval ASEval, Instances data) throws Exception {

    m_best = null;
    int[] attributes;
    if (!(ASEval instanceof SubsetEval)) {
        throw new Exception(ASEval.getClass().getName() + " no es "
                + "un evaluador de subconjuntos!");
    }


    if (data.classIndex() == -1 || !this.m_hasClass || this.m_numAttribs == 0)
    {

        throw new Exception("Debe inicilizar el metodo de busqueda");
    }

    SubsetEval ASEvaluator = (SubsetEval)ASEval ;

    m_random = new Random(m_seed);
    m_population = new GABitSet[m_popSize];

    initPopulation();
    evaluatePopulation(ASEvaluator);
    populationStatistics();
    scalePopulation(m_scaled);
    checkBest();

    boolean converged;
    for (int i = 1; i <= m_maxGenerations; i++) {
        generation();
        evaluatePopulation(ASEvaluator);

        populationStatistics();
        scalePopulation(m_scaled);
        converged = checkBest();
        {
            attributes = attributeList(m_best.getChromosome());
            m_statistic.statisticalSet(attributes, m_best.m_objective);
        }

        if ((i == m_maxGenerations) || (converged == true)) {
            if (converged == true) {
                break;
            }
        }
    }

    attributes = attributeList(m_best.getChromosome());
    m_statistic.removeInHash(attributes);
    return attributes;
}
```

3.1.2 Enfriamiento Simulado

Este algoritmo es una metaheurística para problemas de optimización global que se basa en conceptos de la mecánica estadística. La característica principal de este algoritmo es que al buscar una nueva solución Sn+1 dada una solución Sn, acepta en ocasiones una de inferior aptitud a la de Sn por medio de una función probabilística. Para el desarrollo de este algoritmo de búsqueda se tienen en cuenta los siguientes aspectos:

- ✓ Diseñar una representación simple y eficiente de la solución a través de una cadena binaria, similar al procedimiento de búsqueda genética. De esta forma se discretiza el espacio de búsqueda y contribuye a explorarlo eficientemente.

- ✓ Determinar los parámetros que influyen el proceso de enfriamiento, estos son: la Temperatura Final (*Tf*), la Temperatura Inicial (*Ti*), el número de iteraciones por temperatura*(N*), el tipo de coeficiente de enfriamiento; estos constituyen un factor primordial para el éxito del algoritmo. Estos parámetros pueden ser cambiados para un fin específico, aunque poseen valores predeterminados en el procedimiento de búsqueda.

- ✓ Se determina la primera solución aleatoriamente, o sea se genera una cadena binaria que representa el subconjunto de variables solución. En estos casos la inicialización de la primera solución no constituye un gran peso en la solución para valores altos de temperatura, puesto que es muy probable que esa solución sea rechazada por una de menor calidad.

- ✓ Decidir cuántas iteraciones hacer por cada temperatura, iterar muchas veces para temperaturas tempranas podría significar un alto coste; ya que el procedimiento de búsqueda es inestable y no existe un patrón de estados candidatos a ser buenas soluciones.

- ✓ Generar una solución vecina de la solución anterior, de modo que se explore en gran

medida el espacio de búsqueda. Se calcula la diferencia de ajuste entre ambas $Dif = (F_k - F_{k+1})$, donde F_k es la solución anterior y F_{k+1} es la actual; de tal forma que siempre se acepta la nueva solución si se cumple $(Dif < 0)$. Así si la nueva solución generada tiene menor ajuste $(Dif > 0)$ la probabilidad de aceptar F_{k+1} como solución para una temperatura (T), esta dada por $N(0,1) < e^{(-Dif/T)}$ donde $N(0,1)$ es un número aleatorio con distribución uniforme entre cero y uno.

- ✓ Decidir el valor del coeficiente de enfriamiento de temperatura, quién influye en la calidad del procedimiento y en la rapidez de ejecución del mismo. (37)

A partir de los aspectos anteriores la implementación del algoritmo se muestra a continuación:

```
public int[] search(ASEval ASEval, Instances data) throws Exception {

    int attributes[];
    boolean limit;
    if (m_numAttribs > m_iterates)
        m_iterates = m_numAttribs;

    SubsetEval ASEvaluator = (SubsetEval) ASEval;

    if (data.classIndex() == -1 || !this.m_hasClass || this.m_numAttribs == 0){

        throw new Exception("Debe inicilizar el metodo de busqueda");
    }

    double new_merit, energy = 0, prob = 0.0, prob_normal = 0.0;
    BitSet new_sol;

    while (m_To >= m_Tf) {
        limit = false;
        for (int i = 0; i <m_iterates ; i++) {
            new_sol = generateSubset();
            new_merit = ASEvaluator.evaluateSubset(new_sol);
            energy = m_fitness - new_merit;

            if (energy < 0) {
                m_best = new_sol;
                m_fitness = new_merit;
                limit = true;
            } else {
                if (Utils.eq(m_fitness, new_merit)) {

                    int count_old = countFeatures(m_best);
                    int count_new = countFeatures(new_sol);
                    if (count_old > count_new) {
                        m_best = new_sol;
                        m_fitness = new_merit;
                        limit = true;
                    }
                } else {

                    double den = Math.sqrt(3 * Math.PI);
                    double exp = -0.5 * Math.pow(m_gauss.nextDouble(), 2);
                    double num = Math.exp(exp);
                    prob_normal = (double) num / den;
                    prob = Math.exp(-1 * energy / m_To);

                    if (prob_normal < prob) {
                        m_best = new_sol;
                        m_fitness = new_merit;
                    }
                }
            }

            if(m_statistic.count != 0)
            {
                attributes = attributeList(m_best);
                m_statistic.statisticalSet(attributes, m_fitness);
            }

            if (limit) {

                if(m_statistic.count==0 && m_To*m_To*m_To*m_To < m_Tf)
                {
                    attributes = attributeList(m_best);
                    m_statistic.statisticalSet(attributes, m_fitness);
                }
            }
        }
        m_To = m_alpha * m_To;
    }
    attributes = attributeList(m_best);
    m_statistic.removeInHash(attributes);
    return attributes;
}
```

3.1.3 Hibridación entre Algoritmo Genético y algoritmo de Búsqueda Secuencial

Los algoritmos genéticos son por construcción métodos de búsqueda ciega, el proceso de optimizar es una caja negra que asigna a cada individuo una aptitud. Esta opacidad en la medida que proporciona un algoritmo de propósito general y permite realizar la búsqueda con información mínima, tienen la contrapartida de que son intrínsecamente débiles. Como la debilidad es intrínseca, cualquier intento de mejora cualitativa implica incorporarle al algoritmo un mecanismo de explotación de la solución, después de explorar el espacio de búsqueda.

La idea general de esta técnica de hibridación consiste en utilizar el algoritmo genético para realizar la búsqueda global y encargar la búsqueda local greedy (secuencial) para explotar la solución. Para esto fue necesario llevar a cabo la hibridación de forma modular, incorporando el procedimiento de búsqueda secuencial como un operador más del algoritmo genético.

El procedimiento de búsqueda local, toma como punto de partida las soluciones brindadas por el algoritmo genético en cada generación después de aplicarles los operadores probabilísticos; así el método de búsqueda secuencial, explota los estados vecinos que generan estas soluciones globales considerando solo aquellas que sean mejores.

A partir de el estudio bibliográfico en profundidad (**Anexo 1**) realizado, en el mismo no se brinda un resultado concluyente, en la bibliografía no se muestran resultados de su aplicación, solamente la mención de su empleo y las características del mismo. A continuación se muestra la implementación del mismo.

```
public int[] search(ASEval ASEval, Instances data) throws Exception {

    m_best = null;
    int[] attributes;
    m_random = new Random(m_seed);
    m_population = new GABitSet[m_popSize];
    SubsetEval ASEvaluator = (SubsetEval) ASEval;

    if (data.classIndex() == -1 || !this.m_hasClass || this.m_numAttribs == 0){
        throw new Exception("Debe inicilizar el metodo de busqueda antes de efectuarlo.");
    }



    initPopulation();
    evaluatePopulation(ASEvaluator);

    populationStatistics();
    scalePopulation(m_scaled);
    checkBest();

    boolean converged;
    for (int i = 1; i <= m_maxGenerations; i++) {
        generation();
        localSearch(ASEvaluator, data);
        evaluatePopulation(ASEvaluator);
        populationStatistics();
        scalePopulation(m_scaled);

        converged = checkBest();


        {
            attributes = attributeList(m_best.getChromosome());
            m_statistic.statisticalSet(attributes, m_best.getObjective());
        }

        if ((i == m_maxGenerations) || (converged == true)) {
            if (converged == true) {
                break;
            }
        }
    }


    attributes = attributeList(m_best.getChromosome());
    m_statistic.removeInHash(attributes);
    return attributes;
}
```

3.2 Funcionamiento General del Servicio de Selección de variables

A todos los algoritmos de búsqueda planteados anteriormente se le incorporó un mecanismo de almacenamiento de las mejores soluciones durante su ejecución. O sea, se implementó un proceso de almacenamiento de aquellas soluciones cuya aptitud fuera superior a la aptitud promedio del conjunto de soluciones almacenadas. Para lograr mayor eficiencia en consultas de selección e inserción al conjunto de soluciones, estas se almacenan en Tablas Hash. Este mecanismo de estadística simple, permite obtener un subconjunto de soluciones (ordenadas por aptitud) finales al problema, permitiéndole al usuario escoger cualquiera de estas.

Los métodos de evaluación antes referidos necesitan una medida o criterio de evaluación por lo que fueron implementadas las siguientes medidas de evaluación:

- ✓ Para las variables individuales se implementaron las de CHI2, Correlación de Pearson, Incertidumbre Simétrica y la de Gain Ratio.
- ✓ Para subconjuntos fueron implementadas las de Correlación de subconjuntos y la de consistencia.

A continuación se muestra un diagrama de flujo (**Figura 2**) donde se explica de modo general la funcionalidad de reducción de espacio muestral y la de ordenamiento de las características independientes de acuerdo con la relevancia que estas presentan con respecto a la clase o característica dependiente (actividad biológica) empleando la Metodología de Filtro.

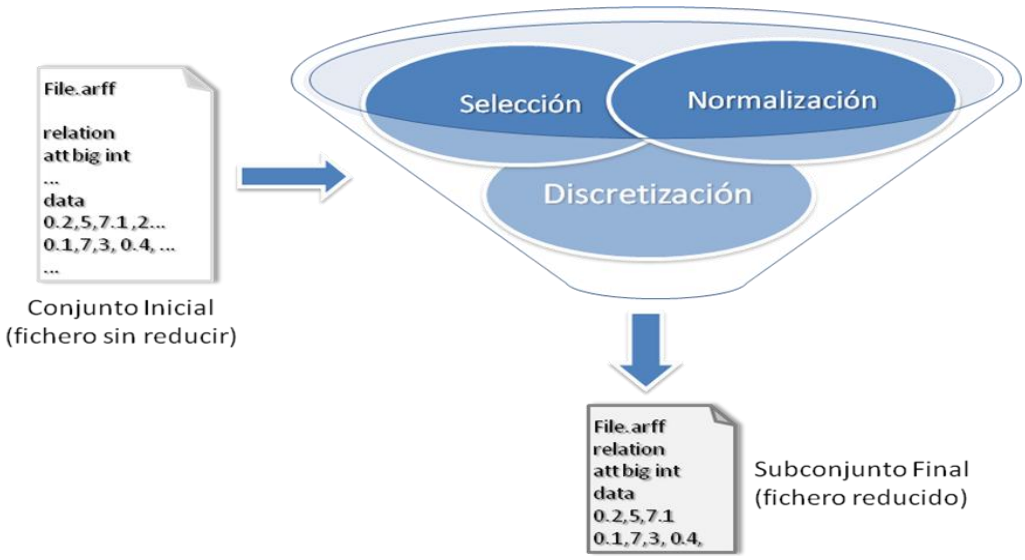

Figura 3: Diagrama general del flujo de eventos

De manera general este diagrama muestra cómo son cargados al sistema los ficheros de tipo *.arff* y por los procesos que pasan para ser reducidos u ordenados en dependencia de la orden que el administrador le pase al sistema. Al final del proceso se entrega un fichero en el que se encuentran los datos reducidos, este fichero es de tipo *.arff*.

De manera más específica el siguiente diagrama de flujo (**Figura 3)** muestra el proceso de la reducción de las características para tener una visión ampliada del proceso de reducción y lograr un mejor entendimiento de la solución del problema.

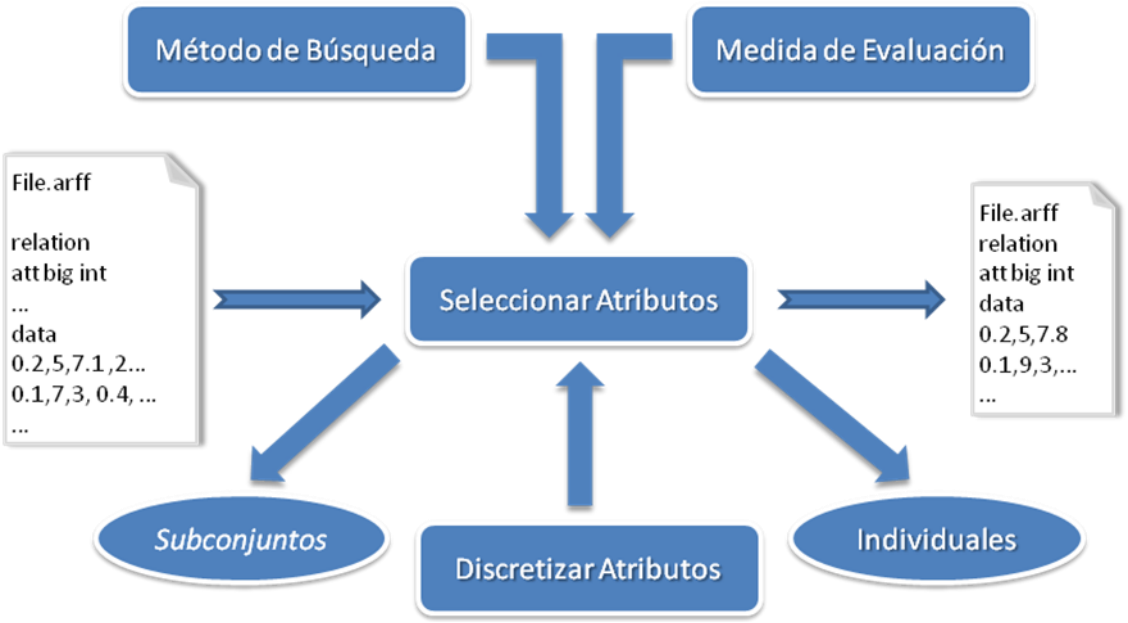

Figura 4: Diagrama de flujo para el proceso de selección de atributos o reducción de espacio muestral

En el diagrama anterior se describe cómo el fichero de entrada de tipo *.arff* es incorporado desde una base de datos al sistema, con el objetivo de realizarle una reducción de sus atributos. El administrador selecciona el método de búsqueda y la medida de evaluación, en caso de que este fichero presente datos con valores continuos y la variable dependiente o clase posea valores discretos; entonces se procede a la discretización de los datos. Este proceso de discretización no es más que convertir los valores continuos a valores de tipo discreto; buscando así uniformidad entre los datos de las variables independientes y la variable dependiente (clase).

Las medidas de evaluación son utilizadas para la selección de características de tipo individual o de subconjuntos, de acuerdo a la que se haya seleccionado se realiza la reducción correspondiente.

Al terminarse la selección los nuevos datos son guardados en un fichero nuevo de tipo *.arff* y este es guardado en una base de datos.

Implícito dentro del proceso de selección se encuentra el proceso de ordenamiento de los atributos, el cual se explica de la siguiente manera y basándose en el diagrama anterior:

- ✓ El administrador selecciona el método de búsqueda y la medida de evaluación; en caso de que este fichero presente datos con valores continuos y la variable dependiente o clase posea valores discretos, entonces se procede a la discretización de los datos.
- ✓ Al finalizar este proceso de ordenamiento en vez de crearse un fichero *.arff* se crea un fichero *.txt* con los datos ordenados de acuerdo con la relevancia que estos presentan con la variable dependiente o clase. Este proceso es muy importante pues con sus resultados se pueden realizar estudios estadísticos por parte de los especialistas en la parte de predicción y clasificación para de alguna manera tener una visión de la relación que tienen los atributos o variables independientes con respecto a la variable dependiente o clase.

3.3 Análisis de los resultados

3.3.1 Análisis de la muestra de Cefalosporinas

Para comprobar la eficiencia y rapidez de los métodos implementados se tomaron muestras de datos reales de una familia de cefalosporinas (38). A continuación se muestran las características de la familia:

Cantidad inicial de variables: 180

Instancias: 104

Esta muestra mantiene dentro de sus características principales que todos sus compuestos son reportados como activos en los ensayos realizados, por lo que este estudio va encaminado a determinar, cuáles son las características estructurales distintivas dentro de los activos. Para emplear un criterio de clasificación sobre esta muestra se tomó la variable dependiente perteneciente a la actividad biológica y se consideraron como activos aquellos compuestos cuyo valor fuese mayor o igual que el promedio e inactivo en caso contrario.

Para la clasificación de las muestras se emplean las máquinas de soporte vectorial C-SVC y nu-SVC, pertenecientes ambas a la librería libSVM en su versión 2.8. Dicha librería posee varias funciones Kernels que le permiten redimensionar los valores de entrada:

- De base radial (RBF)
- Polinomial
- Lineal
- Sinusoidal

De estas funciones y partiendo de las características fundamentales de la muestra se emplea la RBF debido a que la misma posee más de 50 espacios de nueva dimensión lo que le permite encontrar los mejores valores de clasificación, además que soporta la no linealidad entre los datos.

La MSV es una técnica de aprendizaje supervisado en la que los parámetros C (Costo), nu (Valor empleado por nu-SVC) y alfa (Valor empleado por la función kernel) son fundamentales para garantizar el entrenamiento de la misma, evitando así la memorización o sobre entrenamiento de las máquinas. El método para determinar las muestras de entrenamiento y prueba que se empleó es la validación cruzada (*cross validation*). La cantidad de subconjuntos destinados para la misma, según la cantidad de instancias presentadas por la muestra, alcanzaron valores entre 2 y 100, mientras que nu y gamma tomaron valores entre 0.01 y 0.99(**Anexo 2**), C tomó valores entre 1 y 100 (**Anexo 3**). Los resultados del mejor modelo se muestran en la **Tabla 3** con los valores fijados.

Tabla 3: Resultados de la Clasificación para costo 100,nu 0.9 y ganma 0.5

Valor de Validación	MSV	% clasificación
2	C-SVC	67
	nu-SVC	66
5	C-SVC	66
	nu-SVC	66
10	C-SVC	71
	nu-SVC	71

20	C-SVC	71
	nu-SVC	68
100	C-SVC	71
	nu-SVC	68

Los mejores resultados se alcanzaron con la validación cruzada en 10, los resultados se muestran en la **Tabla 4,** encontrándose además los valores de precisión y Área ROC.

Tabla 4: Calidad de la Clasificación para la muestra completa

MSV	Precisión	Área ROC	% Clasificación
nu-SVC	0.712	0.712	71.15
C-SVC	0.712	0.712	71.15

Los resultados obtenidos de la clasificación para la muestra completa permiten fijar las condiciones para los experimentos con las muestras reducidas en busca de los mejores modelos de clasificación. Dentro de las ventajas de la reducción de variables se encuentra la mejora de la eficiencia del clasificador, y para ello se emplean los algoritmos propuestos en el capítulo anterior (AG, ES, AH).

Para evaluar la calidad de las variables seleccionadas se emplearon las medidas de evaluación basadas en Consistencia y CFS para evaluar subconjuntos de atributos. Cada algoritmo posee parámetros que son influyentes dentro de los resultados del mismo, en el caso de Algoritmo Genético (AG) y Algoritmo Híbrido (AH) sus resultados dependen de la Probabilidad de Cruzamiento (Pc), la cual permite el cruzamiento de dos individuos para lograr uno con mejores potencialidades que los dos anteriores, por lo que sus valores deben permanecer por encima de 0.50. Para estudiar el comportamiento de los mismos con respecto a la generación de los modelos se utilizaron los valores de 0.6, 0.7, 0.8 y 0.9. Otro de los parámetros que influyen en estos algoritmos es la Probabilidad de Mutación (Pm), la cual permite la creación de nuevos individuos a partir de las características de los anteriores. No obstante, la Pm puede dar lugar a combinaciones de variables que generen malas soluciones por correlaciones casuales. Estas van a provocar que los algoritmos pierdan el sentido de la búsqueda al brindar respuestas ajenas a la fenomenología estudiada. En la bibliografía

consultada (39) han propuesto minimizar el valor de Pm por debajo de 0.1. En la presente tesis se definió el valor de Pm como 0.01.

El sistema propuesto permite la generación de varias combinaciones de solución posibles. En este trabajo se fijó como máximo diez de las mejores soluciones que se obtienen. A partir de estas condiciones se realizó la selección de variables especificando el método de búsqueda, la Pc, la medida de evaluación a valor de Pm constante. Para cada caso se obtuvieron los valores correspondientes de cantidad final de variables y porciento de reducción. Esos resultados se muestran en la tabla 3.

Tabla 3: Resultados de la selección de variables

Método de Búsqueda	Medida de Evaluación	Cantidad de Variables	% Reducción	Pc
AG	CFS con PL	54	70	0.6
		46	75	0.7
		9	95	0.8
		15	91	0.9
AG	Consistencia	57	68	0.6
		71	60	0.7
		50	72	0.8
		74	58	0.9
AH	CFS con PL	13	92	0.6
		12	91	0.7
		9	95	0.8
		13	92	0.9
AH	Consistencia	9	95	0.6
		10	94	0.7
		10	94	0.8
		10	94	0.9

Dada la complejidad polinomial para explorar todo el espacio de búsqueda que tienen estos algoritmos, el tiempo de respuesta varía en dependencia del método empleado. En la tabla 4 se presentan los tiempos de respuesta resultantes al aplicar AG y AH.

Tabla 4: Tiempo de respuesta de AG y AH

Algoritmo	Medida	Tiempo(minutos)
AG	Consistencia	0.23
AG	CFS	0.58
AH	Consistencia	0.73
AH	CFS	0.59

Para validar la calidad de la selección se le aplicó el clasificador Máquinas de Soporte Vectorial con los parámetros definidos en la **Tabla 5**. Los resultados para el Kernel y los dos tipos de máquinas de soporte vectorial se muestran a continuación, donde la identificación del modelo viene dada por: método de búsqueda, más la medida de evaluación y la probabilidad.

Tabla 5: Clasificación de la selección por AG

No. Modelo	Modelo	No. Variables	MSV	Precisión	Área ROC	% Clasificación
1	AG-CFS-0.6	54	C-SVC	0.98	0.98	98
			nu-SVC	0.934	0.933	93
2	AG-CFS-0.7	46	C-SVC	0.915	0.92	91
			nu-SVC	0.898	0.894	89
3	AG-CFS-0.8	9	C-SVC	0.78	0.779	78
			nu-SVC	0.838	0.837	84
4	AG-CFS-0.9	15	C-SVC	0.929	0.923	92
			nu-SVC	0.885	0.884	89
5	AG-Consistencia-0.6	54	C-SVC	0.952	0.952	95
			nu-SVC	0.904	0.904	90
6	AG- Consistencia-0.7	46	C-SVC	0.942	0.942	94
			nu-SVC	0.914	0.913	91
7	AG- Consistencia-0.8	9	C-SVC	0.894	0.885	89
			nu-SVC	0.825	0.837	84
8	AG- Consistencia-0.9	15	C-SVC	0.924	0.923	92
			nu-SVC	0.875	0.875	88

Los resultados de la clasificación se comportan en este algoritmo entre un 78 y 98% de clasificación correcta, de la misma manera que la precisión y el área debajo de la curva se mantienen entre rangos de valores que permiten validar la eficiencia del clasificador empleado para cada uno de los modelos. De los dos tipos de máquinas de soporte vectorial empleadas, los mejores resultados se obtienen con C-SVC. Según los valores, el mejor modelo es el 1, al alcanzar un 98 % de buena clasificación; sin embargo, este modelo cuenta con 54 variables; mientras que el modelo 4, con solo 15 variables, alcanza un 92 % para ambas medidas de evaluación y, teniendo como precedente el principio de parsimonia, este es el mejor de los

modelos creados por los Algoritmos Genéticos demostrándose que mientas mayor es la Pc y menor la Pm se tienen mejores resultados. Para este algoritmo, el promedio de reducción de variables por ambas técnicas fue de 31 variables. Basado en los mismos criterios, el AH se comporta de la siguiente manera.

Tabla 6: Clasificación de la selección por AH

No. Modelo	Modelo	No. Variables	MSV	Precisión	Área ROC	% Clasificación
1	AH-CFS-0.6	13	C-SVC	0.706	0.702	70
			nu-SVC	0.799	0.798	80
2	AH-CFS-0.7	12	C-SVC	0.799	0.798	70
			nu-SVC	0.799	0.798	80
3	AH-CFS-0.8	9	C-SVC	0.714	0.712	71
			nu-SVC	0.799	0.798	80
4	AH-CFS-0.9	13	C-SVC	0.799	0.798	70
			nu-SVC	0.799	0.798	80
5	AH-Consistencia-0.6	9	C-SVC	0.695	0.692	70
			nu-SVC	0.827	0.827	83
6	AH- Consistencia-0.7	10	C-SVC	0.695	0.692	70
			nu-SVC	0.808	0.808	81
7	AH- Consistencia-0.8	10	C-SVC	0.695	0.692	70
			nu-SVC	0.808	0.808	81
8	AH- Consistencia-0.9	10	C-SVC	0.695	0.692	70
			nu-SVC	0.808	0.808	81

Los valores de % de clasificación se encuentran entre un 70 y 83 %, siendo la mejor máquina de soporte vectorial nu-SVC. El modelo 5 con 9 variables es el mejor de los creados con un 83 % de clasificación y 0.827 de precisión y Área respectivamente.

Resultados de Enfriamiento Simulado

El método de búsqueda ES, es una técnica para explorar todo el espacio de búsqueda basada en una probabilidad, donde las dos condiciones fundamentales para la realización de una buena exploración son: a.- definir una temperatura inicial alta para garantizar que se cubra

todo el espacio de búsqueda, y b.- mantener una temperatura final baja. Dentro de la probabilidad de moverse o no hacia una mejor o peor solución juega un papel fundamental el valor de alfa, quien determina la velocidad de enfriamiento. Se reporta que los valores más acertados deben ser superiores a 0.5, que corresponden a velocidades de enfriamiento lentas lo que permite explorar todo el espacio de búsqueda (37) (40). Los empleados en este trabajo son 0.7, 0.8 y 0.9. Los resultados de la reducción de variables se presentan en la tabla 7.

Tabla 7: Resultados de la selección de variables utilizando enfriamiento simulado.

Métodos Búsqueda	Medida de Evaluación	Cantidad de Variables	% Reducción	alfa
ES	CFS	20	88.8	0.7
		9	95	0.8
		1	99.4	0.9
ES	Consistencia	69	61.6	0.7
		74	58.8	0.8
		64	64.4	0.9

Teniendo en cuenta la complejidad polinomial del algoritmo ES, los tiempos de cómputo promedio para explorar toda la muestra resultaron ser de 0,68 y 0,28 segundos para Consistencia y CFS respectivamente, como medidas de evaluación, lo cual contrasta con los resultados para AG, que resultan ser mayores (**Tabla 4**).

A los conjuntos de descriptores seleccionados se les aplicó el clasificador con los mismos parámetros, y se obtuvieron resultados significativos. En la **Tabla 9** se describen los resultados teniendo en el nombre del modelo el algoritmo seguido de la medida de evaluación y la variación de alfa.

Tabla 8: Clasificación de la selección de variables con ES

No. Modelo	Modelo	No. Variables	MSV	Precisión	Área ROC	% Clasificación
1	ES-CFS-0.7	20	C-SVC	1	1	100
			nu-SVC	0.991	0.99	99.03
2	ES-CFS-0.8	9	C-SVC	0.933	0.933	93.26
			nu-SVC	0.904	0.904	90.38

3	ES-CFS-0.9	1	C-SVC	0.751	0.75	75
			nu-SVC	0.751	0.75	75
4	ES-Consistencia-0.7	69	C-SVC	1	1	100
			nu-SVC	1	1	100
5	ES- Consistencia-0.8	74	C-SVC	1	1	100
			nu-SVC	1	1	100
6	ES- Consistencia-0.9	64	C-SVC	1	1	100
			nu-SVC	1	1	100

Los resultados de la clasificación se comportan con ES entre un 75 y 100% de clasificación correcta para ambas MSV, de la misma manera que la precisión y el área debajo de la curva se mantienen entre rangos de valores que permiten validar la eficiencia del clasificador empleado para cada uno de los modelos.

Aunque los modelos 4, 5 y 6 poseen 100% de clasificación en ambas máquinas de soporte vectorial la cantidad de variables que poseen los hace imprácticos; mientras que los modelos 1 y 2 poseen 20 y 9 variables respectivamente y tienen 100 y 93 % de clasificación con precisión y Área en 1 y 0.99 respectivamente. El promedio de reducción de variables es de 16.

Los modelos de clasificación basados en MSV para esta muestra con los parámetros anteriormente identificados poseen entre 9 y 20 variables y los porcientos de clasificación entre 83 y 100 % respectivamente.

3.3.2 Análisis de la muestra de Inhibidores del Factor Esteroidogénico1

Las Máquinas de Soporte Vectorial es un clasificador, donde una propiedad influyente en sus resultados es el tamaño de la muestra. Para ello se realizarán las pruebas con la muestra 599 con las siguientes características:

Cantidad inicial de variables (Descriptores): 291

Instancias: 315

Para la clasificación de la misma, se mantienen los valores los parámetros Costo, nu y gamma en 100,0.9 y 0.5 respectivamente para el entrenamiento de las MSV siendo los mejores modelos los obtenidos con la validación cruzada en 100 (**Anexo 4),** los que se muestran de manera ampliada en la **Tabla 10.**

Tabla 10: Clasificación total de la muestra para validación cruzada 100

MSV	Precisión	Área ROC	% Clasificación
nu-SVC	0.537	0.516	56
C-SVC	0.559	0.548	57

Los valores de la clasificación son de 56 y 57 % para nu-SVC y C-SVC respectivamente, en los modelos los valores de precisión y Área muestran la existencia de un incremento en la razón de falsos positivos, por lo que existe una mala clasificación de los modelos con respecto a la actividad biológica. Manteniendo los mismos parámetros, métodos de búsqueda y medidas de evaluación se realiza la reducción de las variables para eliminar aquellas variables que no brinden información significativa, en la **Tabla 11** se muestra la cantidad de variables y el porciento en el que se reduce la misma.

Tabla 9: Resultados de la selección de variables con AG y AH

Método de Búsqueda	Medida de Evaluación	Cantidad de Variables	% Reducción	Pc
AG	CFS con PL	32	90	0.6
		55	80	0.7
		14	95	0.8
		27	91	0.9
AG	Consistencia	52	82	0.6
		62	78	0.7
		65	77	0.8
		28	90	0.9
AH	CFS con PL	2	99	0.6
		2	99	0.7
		2	99	0.8
		2	99	0.9

AH	Consistencia	1	99	0.6
		1	99	0.7
		1	99	0.8
		1	99	0.9

El tiempo empleado para reducir esta muestra exiló entre 0.14 y 0.61 segundos (**Tabla 12**).

Tabla 10: Tiempo de respuesta del AG y AH

Algoritmo	Medida	Tiempo(minutos)
AG	Consistencia	0.27
AG	CFS	0.17
AH	Consistencia	0.61
AH	CFS	0.14

Comparando con la muestra anterior, aunque existe mayor cantidad de variables, el tiempo de cómputo para algunos métodos es menor; lo que evidencia la eficiencia de los mismos. Si bien el tiempo de ejecución mejora considerablemente, los resultados de los modelos de clasificación no muestran resultados alentadores (**Tabla 13)**

Tabla 11: Clasificación de la selección de variables por AG

No. Modelo	Modelo	No. Variables	MSV	Precisión	Área ROC	% Clasificación
1	AG-CFS-0.6	32	C-SVC	0.682	0.537	60
			nu-SVC	0.623	0.613	63
2	AG-CFS-0.7	55	C-SVC	0.68	0.553	62
			nu-SVC	0.63	0.608	64
3	AG-CFS-0.8	14	C-SVC	0.682	0.537	61
			nu-SVC	0.637	0.613	65
4	AG-CFS-0.9	27	C-SVC	0.66	0.534	60
			nu-SVC	0.622	0.614	61
5	AG-Consistencia-0.6	52	C-SVC	0.682	0.643	67
			nu-SVC	0.657	0.638	66
6	AG- Consistencia-0.7	62	C-SVC	0.69	0.578	64
			nu-SVC	0.605	0.593	61
7	AG- Consistencia-0.8	65	C-SVC	0.662	0.604	65
			nu-SVC	0.658	0.648	66
8	AG- Consistencia-0.9	28	C-SVC	0.678	0.64	67
			nu-SVC	0.616	0.597	62

La MSV nu-SVC mantiene los mejores resultados, los mismos se encuentran entre 62 y 67 % respectivamente. Mientras que la C-SVC se mantiene entre un 60 y 64%, aunque los valores de precisión y el Área ROC muestran que no existe un balance entre las razones de verdaderos positivos y falsos positivos al existir un incremento de los falsos positivos. El modelo 3 con 14 variables y 65 % de clasificación resulta el de mejores resultados; siendo además el de menor cantidad de variables. El AH por su parte mantiene un comportamiento estable en la cantidad de descriptores seleccionados; pues para todas las combinaciones de medidas y parámetros en CFS seleccionó solo 2 descriptores y en Consistencia uno con los valores de precisión y área que se muestran en la **Tabla 14**.

Tabla 12: Clasificación de la selección de variables por AH

No.	Modelo	Variables	MSV	Precisión	Área ROC	% Clasificación
1	AH-CFS-0.6-0.9	2	C-SVC	0.671	0.568	63
			nu-SVC	0.394	0.437	38
2	AH-Consistencia-0.6-0.9	1	C-SVC	0.674	0.55	61
			nu-SVC	0.444	0.459	41

Para ambos modelos nu-SVC obtiene resultados bajos, no siendo de la misma manera para C-SVC. Los resultados mostrados por el algoritmos en cuanto a la cantidad de descriptores seleccionados y para evitar que el mismo se estanque en valores que no fueran significativos para la muestra en general, se eliminaron dichas variables; obteniendo como respuesta la misma cantidad de seleccionados. Los valores de precisión y Área muestran un comportamiento similar a los modelos analizados anteriormente para C-SVC, mientras que para nu-SVC los valores del Área ROC son mayores que la precisión lo que muestra para estos modelos una disminución de los falsos positivos. Al emplear como algoritmo de búsqueda ES el comportamiento de la muestra es el siguiente, que se observa en la **Tabla 15**.

Tabla 13: Clasificación de la selección de variables por ES

Métodos Búsqueda	Medida de Evaluación	Cantidad de Variables	% Reducción	alfa
ES	CFS con PL	10	96	0.7
		4	98	0.8
		9	95	0.9
ES	Consistencia	5	98	0.7
		2	99	0.8
		5	98	0.9

Con este algoritmo se logra reducir mejor la cantidad de variables para las dos medidas de evaluación, los tiempos promedios se muestran en la **Tabla 16**.

Tabla 14: Tiempo de respuesta del algoritmo ES

Algoritmo	Medida	Tiempo(minutos)
ES	Consistencia	0.52
ES	CFS	0.45

En la tabla 17 se muestran los modelos de clasificación de la muestra reducida empleando ES como método de búsqueda para las distintas variaciones de alfa y medidas de evaluación.

Tabla 15: Clasificación de la selección de variables por ES

No. Modelo	Modelo	No. Var.	MSV	Precisión	Área ROC	% Clasificación
1	ES-CFS-0.7	10	C-SVC	0.709	0.645	68
			nu-SVC	0.663	0.615	64
2	ES-CFS-0.8	4	C-SVC	0.656	0.571	63
			nu-SVC	0.572	0.572	60
3	ES-CFS-0.9	9	C-SVC	0.655	0.596	64
			nu-SVC	0.572	0.572	60
4	ES-Consistencia-0.7	5	C-SVC	0.674	0.55	61
			nu-SVC	0.543	0.529	56
5	ES- Consistencia-0.8	2	C-SVC	0.648	0.535	60
			nu-SVC	0.599	0.554	60
6	ES- Consistencia-0.9	5	C-SVC	0.657	0.552	62
			nu-SVC	0.555	0.53	55

Una vez concluidas las pruebas realizadas y descritas en tablas anteriores, los mejores modelos se obtienen con C-SVC. Los porcientos de buena clasificación se mantiene entre un 60 y 68 %, siendo los modelos 1 y 2 los de mejores resultados, los cuales son superiores a los de modelos obtenidos con la muestra inicia; aunque mantiene el mismo comportamiento de los valores de precisión y Área comentados anteriormente.

3.3.3 Clusterización de la Muestra de Inhibidores del Factor Esteroidogénico1

Se identificaron de manera espontánea cuatro grupos de compuestos pertenecientes a grupos estructuralmente diferentes por lo que la actividad biológica no debe, necesariamente, responder a iguales aspectos estructurales de las moléculas. En la **Tabla 18** se muestran los

resultados de la clusterización empleando los algoritmos Simple Kmeans y el Método de Ward.

Tabla 16: Compuestos por cluster

Clúster	Cantidad de Instancias
0	57
1	92
2	92
3	96

Para la cantidad de instancias se empleó validación cruzada 10, kernel RBF, gamma 0.9, nu 0.5 siendo estos los parámetros fijados para la obtención de los modelos de clasificación. Los modelos para cada clúster se muestran en la siguiente tabla:

Tabla 17: Clasificación de los clústeres

Clúster	MSV	Precisión	Área ROC	% Clasificación
0	C-SVC	0.811	0.605	74
	nu-SVC	0.811	0.605	74
1	C-SVC	0.523	0.508	56
	nu-SVC	0.319	0.5	57
2	C-SVC	0.319	0.5	57
	nu-SVC	0.319	0.5	57
3	C-SVC	0.316	0.5	56
	nu-SVC	0.316	0.5	56

Los mejores modelos de clasificación se obtienen en el clúster 0 por ambas MSV, de la misma manera que los valores de precisión y área no se encuentran en total correspondencia. Para realizar la selección de variables se escogieron el de mejor y peores resultados. El análisis con los mismos se muestran en el siguiente epígrafe.

3.3.4 Selección y clasificación para el Clúster 0

Resultados del Enfriamiento Simulado

En las tablas 20, 21 y 22 se muestran los resultados de selección de variables, tiempo de cómputo y clasificación para este clúster. Los tiempos de respuesta del algoritmo en explorar

todo el espacio de búsqueda no superan los 0.50 minutos, los modelos de clasificación poseen entre un 90 y 100 % de clasificación para ambas MSV respectivamente. Siendo los, modelos 1,2 y 3 los que muestran menor cantidad de variables y mejores resultados de clasificación en comparación con los demás obtenidos (**Tabla 22**) aunque en los tres últimos se logra alcanzar el 100 % de clasificación. Para estos modelos se logra un total balance entre precisión, Área y el porciento de buena clasificación. El promedio general de variables es de 20.

Tabla 18: Resultados de la selección de variables con ES

Métodos Búsqueda	Medida de Evaluación	Cantidad de Variables	% Reducción	Alfa
ES	CFS con PL	7	97	0.7
		8	97	0.8
		14	95	0.9
ES	Consistencia	26	91	0.7
		46	84	0.8
		22	92	0.9

Tabla 19: Tiempo de respuesta

Algoritmo	Medida	Tiempo(minutos)
ES	Consistencia	0.14
ES	CFS	0.17

Tabla 20: Clasificación de la selección de variables por ES

No. Modelo	Modelo	No. Variables	MSV	Precisión	Área ROC	% Clasificación
1	ES-CFS-0.7	7	C-SVC	0.967	0.947	96
			nu-SVC	0.898	0.885	90
2	ES-CFS-0.8	8	C-SVC	0.983	0.974	98
			nu-SVC	0.93	0.908	93
3	ES-CFS-0.9	14	C-SVC	1	1	100
			nu-SVC	0.922	0.868	91
4	ES-Consistencia-0.7	26	C-SVC	1	1	100
			nu-SVC	0.937	0.895	92
5	ES- Consistencia-0.8	46	C-SVC	1	1	100
			nu-SVC	0.951	0.921	95
6	ES- Consistencia-0.9	22	C-SVC	1	1	100
			nu-SVC	0.896	0.816	88

Resultados de los Algoritmos Genéticos y Algoritmo Híbrido

Las tablas 23 y 24 muestran los modelos de clasificación obtenidos una vez seleccionada las variables con AG y AH respectivamente, los resultados se encuentran entre un 94 y 100 % de buena clasificación para AG; aunque la cantidad de variables que poseen los modelos para ambas medidas de evaluación los hacen imprácticos, mientras que los modelos donde se empleó para la selección de variables el AH cuenta 2 descriptores para todas las posibles combinaciones de reducción empleadas, siendo estos los mismos seleccionados por este algoritmo en la muestra inicial, aunque los modelos mejoran la clasificación a 79%. Los valores de Área y precisión para AG logran un balance entre la razón de verdaderos y falsos positivos, no comportándose de la misma manera para el AH.

Tabla 21: Clasificación de la selección de variables por AG

No. Modelo	Modelo	No. Variables	MSV	Precisión	Área ROC	% Clasificación
1	AG-CFS-0.6	67	C-SVC nu-SVC	1 0.967	1 0.947	100 96
2	AG-CFS-0.7	46	C-SVC nu-SVC	1 0.965	1 0.961	100 96
3	AG-CFS-0.8	36	C-SVC nu-SVC	1 0.951	1 0.921	100 94
4	AG-CFS-0.9	37	C-SVC nu-SVC	1 0.983	1 0.974	100 98
5	AG-Consistencia-0.6	153	C-SVC nu-SVC	1 1	1 1	100 100
6	AG-Consistencia-0.7	115	C-SVC nu-SVC	1 0.983	1 0.974	100 98
7	AG-Consistencia-0.8	103	C-SVC nu-SVC	1 0.983	1 0.974	100 98
8	AG-Consistencia-0.9	63	C-SVC nu-SVC	1 0.967	1 0.947	100 96

Tabla 22: Clasificación de la selección de variables por AH

No. Modelo	Modelo	No. Variables	MSV	Precisión	Área ROC	% Clasificación
1	AH-CFS-0.6-0.9	2	C-SVC	0.787	0.724	79
			nu-SVC	0.787	0.724	79
2	AH-Consistencia-0.6-0.9	2	C-SVC	0.787	0.724	79
			nu-SVC	0.787	0.724	79

3.3.5 Selección y clasificación para el Cluster3.

Resultados de los Algoritmos Genéticos y Algoritmo Híbrido

En la Tabla 25 se muestran los modelos de clasificación obtenidos al reducir las variables por AG, la máquina C-SVC se encuentra entre un 68 y 100% al mismo tiempo que nu-SVC se comporta entre 65 y 99%. Los modelos 1 y 4 con 43 y 46 variables respectivamente tiene los mejores resultados, aunque, con la cantidad de instancias que cuenta inicialmente el clúster, la cantidad de variables no cumple con las proporciones aceptadas, no siendo así con el modelo 3 aunque los resultados de la clasificación se encuentran por debajo del 70%.

Tabla 23: Clasificación de la selección de variables por AG

No. Modelo	Modelo	No. Variables	MSV	Precisión	Área ROC	% Clasificación
1	AG-CFS-0.6	43	C-SVC	0.99	0.988	99
			nu-SVC	0.905	0.869	88
2	AG-CFS-0.7	101	C-SVC	1	1	100
			nu-SVC	0.952	0.94	95
3	AG-CFS-0.8	6	C-SVC	0.711	0.642	68
			nu-SVC	0.644	0.638	65
4	AG-CFS-0.9	46	C-SVC	1	1	100
			nu-SVC	0.92	0.893	90
5	AG-Consistencia-0.6	142	C-SVC	1	1	100
			nu-SVC	0.97	0.964	97
6	AG- Consistencia-0.7	84	C-SVC	1	1	100
			nu-SVC	0.935	0.917	93
7	AG- Consistencia-0.8	138	C-SVC	1	1	100
			nu-SVC	0.99	0.988	99
8	AG- Consistencia-0.9	202	C-SVC	1	1	100
			nu-SVC	0.99	0.988	99

Los modelos de clasificación obtenidos después de reducir las variables con AH y ambas medidas de evaluación se muestran en la **Tabla 26**. Los valores de clasificación se comportan entre 68-97% y 65-85% para C-SVC y nu-SVC respectivamente. Los modelos 2,3 y 4 tienen los mejores resultados con 7 y 3 variables respectivamente.

Tabla 24: Clasificación de la selección de variables por AH

No. Modelo	Modelo	No. Variables	MSV	Precisión	Área ROC	% Clasificación
1	AH-CFS-0.6	4	C-SVC	0.84	0.82	83
			nu-SVC	0.77	0.754	77
2	AH-CFS-0.7	7	C-SVC	0.97	0.964	97
			nu-SVC	0.858	0.844	85
3	AH-CFS-0.8	7	C-SVC	0.97	0.964	97
			nu-SVC	0.858	0.844	85
4	AH-CFS-0.9	3	C-SVC	0.809	0.787	80
			nu-SVC	0.818	0.799	81
5	AH-Consistencia-0.6-0.9	2	C-SVC	0.711	0.642	68
			nu-SVC	0.646	0.64	65

Resultados del Enfriamiento Simulado

En la siguiente tabla se muestran los modelos obtenidos al reducir las variables con ES, los resultados se encuentran entre un 74 y 99% de clasificación. El mejor modelo según la cantidad de variables y los valores de precisión, Área y porciento de buena clasificación es el 1 para ambas MSV; aunque los modelos 5 y 6 con 27 y 21 variables respectivamente tienen loa mejores resultados, teniendo el inconveniente de la elevada cantidad de variables con respecto a la cantidad de instancias de la muestra.

Tabla 25: Clasificación de la selección de variables por ES

No. Modelo	Modelo	No. Variables	MSV	Precisión	Área ROC	% Clasificación
1	ES-CFS-0.7	13	C-SVC	0.88	0.887	89
			nu-SVC	0.88	0.887	89
2	ES-CFS-0.8	5	C-SVC	0.765	0.728	75
			nu-SVC	0.852	0.815	84
3	ES-CFS-0.9	4	C-SVC	0.802	0.705	74
			nu-SVC	0.831	0.808	83

4	ES-Consistencia-0.7	43	C-SVC	0.99	0.988	99
			nu-SVC	0.952	0.94	95
5	ES- Consistencia-0.8	27	C-SVC	0.979	0.979	98
			nu-SVC	0.938	0.937	94
6	ES- Consistencia-0.9	21	C-SVC	0.99	0.991	99
			nu-SVC	0.901	0.886	90

Al culminar las pruebas de este clúster, comparando los modelos obtenidos en la muestra inicial presentes en la **Tabla 11** y la muestra clusterizada (**Tabla 19**) con los resultados obtenidos una vez seleccionadas las variables con mayor relevancia se logran modelos con mejores resultados.

3.4 Procedimiento para el desarrollo de modelos de clasificación empleando máquinas de soporte vectorial.

Como consecuencia de los resultados de los experimentos realizados, se propone el siguiente procedimiento para el desarrollo de modelos de clasificación empleando máquinas de soporte vectorial, a partir de la reducción del espacio de variables independientes.

1) Aplicarle el clasificador MSV a la muestra completa para determinar los parámetros de los modelos con nu-SVC y C-SVC. Variando C entre 1 y 1000, nu y gamma de 0.01 a 0.99 y la validación cruzada de 2 a 100.
2) Seleccionar las variables teniendo como métodos de búsqueda AG, ES y AH. Variando la Pc de 0.6 a 0.9 y la Pm en 0.01 para AH y AG, mientras que para ES se varía alfa de 0.7 a 0.9.
3) Aplicarle a los subconjuntos seleccionados las medidas de evaluación CFS y Consistencia para obtener el subconjunto final de variables.
4) A la muestra de variables seleccionadas se le aplica el clasificador MSV con los parámetros determinados en el paso 1.
5) Verificar la calidad de los modelos a partir de los resultados de precisión, exactitud y Área ROC. Si la muestra no es heterogénea se escoge el modelo con mejores

resultados, en caso contrario se clusteriza la muestra y se comienza nuevamente el procedimiento para cada clúster.

Conclusiones Generales

1. Se proponen modelos de clasificación de antibióticos del tipo de las cefalosporinas y de compuestos activos frente a cáncer de próstata, empleando Máquinas de Soporte Vectorial como clasificador y Enfriamiento Simulado como método de reducción de variables para las medidas de evaluación propuestas. Los porcientos de acierto oscilaron entre 89 y 100% para cefalosporinas y antitumorales respectivamente.
2. Se propone un procedimiento para el desarrollo de modelos de clasificación a partir de un gran número de datos estructurales, empleando Máquinas de Soporte Vectorial y algoritmos de búsqueda y evaluación para la reducción del número de variables.
3. Se diseñó, implementó y evaluó un nuevo algoritmo híbrido que combina algoritmo genético con algoritmos de búsqueda secuencial, que resultó más potente que los restantes empleados en el trabajo pero menos eficiente por tener mayor pérdida de información.

Recomendaciones

- ✓ Buscar nuevos métodos y medidas de evaluación que contribuyan al mejoramiento de los resultados obtenidos con los algoritmos implementados.
- ✓ Desarrollar modelos QSAR empleando MSV.

Referencias Bibliográficas

1. Kohavi, R. "*Wrappers for Performance Enhancement and Oblivious Decision Graphs*". 1995 Stanford University, Computer Science Department. [Consultado el 22 de marzo de 2008].

2. Balmaseda, J. C. L. "*El Cáncer de los Cubanos*". [Consultado el 22 de junio de 2009]. Disponible en: http://medicinacubana.blogspot.com/2007/04/el-cncer-de-los-cubanos.html.

3. Ruiz Sánchez, R "Selección de Atributos mediante proyecciones" España, Sevilla: s.n., 2005. [Consultado el 6 de noviembre de 2009]

4. [Consultado el 1 de noviembre de 2009] Disponible en: http://www.eumed.net/libros/2006a/rmss/a5.htm

5. Garey, M. R.; Johnson, D. S., "*Computers and Intractability: a Guide to the Theory of NP-Completeness"*, W. H. Freeman and Co., San Francisco, California, 1979. [Consultado el 8 de noviembre de 2009]

6. Kohavi, R.; George H. J., *"Wrappers for feature subset selection. Artificial Intelligence"*, 1997 [Consultado el 8 de noviembre de 2009]

7. Langley, P., "*Selection of relevant features in machine learning."* In Procedings of the AAAI Fall Symposium on Relevance, páginas 1–5, New Orleans, LA, USA, 1994. AAAI Press. [Consultado el 8 de noviembre de 2009]

8. Dash, M.; Liu, H., "*Feature selection for classification. Intelligent Data Analysis*", 1(1-4):131–156, 1997. [Consultado el 6 de octubre de 2009]

9. Cano de Amo, JR. *"Reducción de Datos basada en Selección Evolutiva de Instancias para Minería de Datos"* Granada: s.n., 2004. [Consultado el 6 de octubre de 2009]

10. Glover, F., "*Tabu Search: A Tutorial",* Interfaces, Vol 20, No. 4, pp. 74-94, 1990. [Consultado el 10 de enero de 2009]

11. Holland, J. H., *"Adaptation in Natural And Artificial Systems"*, University of Michigan Press, Ann Arbor, 1975. [Consultado el 7 de enero de 2009]

12. Cerny, V., *"Thermodynamical Approach to the Traveling Salesman Problem: An efficient Simulated Algorithm",* Journal of Optimization Theory and Applications, Vol. 45, 41-45, 1985. [Consultado el 7 de enero de 2009]

13. Kirpatrick, S., Gelatt, C. and Vecchi, M. "*Optimization by Simulated Annealing, Science"*, vol. 220, pp. 672-680, 1983. [Consultado el 7 de enero de 2009]

14. Liu, H.; Setiono, R., *"A probabilistic approach to feature selection - a filter solution"*. In International Conference on Machine Learning, páginas 319–327, 1996. [Consultado el 9 de enero de 2009]

15. Liu, H.; Setiono, R., "*Feature selection and classification - aprobabilistic wrapper approach*". [Consultado el 9 de enero de 2009]

16. [Consultado el 12 de octubre de 2009] Disponible en: http://www.gsi.dit.upm.es/~anto/tesis/html/evalits.html

17. Coello Coello, Carlos A. "*Classification and regression trees*." Chapman & Hall, 1998. [Consultado el 18 de marzo de 2009].

18. Kira, K; Rendell, L. A., "A *practical approach to feature selection."* In Proceedings of the Ninth International Conference on Machine Learning, páginas 249–256, Aberdeen, Scotland, 1992. Morgan Kaufmann. [Consultado el 18 de febrero de 2008]

19. Kononenko, I., "*Estimating attributes: Analysis and extensions of RELIEF."* In European Conference on Machine Learning, páginas 171–182, 1994. [Consultado el 22 de febrero de 2008]

20. Schlimmer, J.C., *"Efficiently inducing determinations: A complete and efficient search algorithm that uses optimal pruning."* In Proceedings of the Tenth International Conference on Machine Learning, pages 284–290, New Brunswick, NJ, 1993. Morgan Kaufmann. [Consultado el 12 de marzo de 2008]

21. Almuallim, H.; Dietterich, G.T., *"Learning with many irrelevant features."* In Proceedings of the Ninth National Conference on Artificial Intelligence, volume 2, pages 547–552, San Jose,CA, 1991.AAAI Press. [Consultado el 12 de diciembre de 2009]

22. Huan Liu, Hiroshi Motoda, and Manoranjan Dash. *"A monotonic measure for optimal feature selection"*. In European Conference on Machine Learning, paginas 101–106, 1998. [Consultado el 23 de octubre de 2009]

23. Pawlak, Z. "*Rough Sets, Theoretical aspects of reasoning about data."* Kluwer Academic Publishers, 1991. [Consultado el 23 de octubre de 2009]

24. Cover, T. M.; Thomas, Joy A. "Elements of Information theory." Wiley-Interscience, 1991. [Consultado el 14 de marzo de 2008]

25. Sheinvald, J.; Dom, B.; Niblack, W. "*A modelling approach to feature selection*." In 10th International Conference on Pattern Recognition, volume i, pages 535–539, 1990. [Consultado el 14 de marzo de 2008]

26. Press, W.H.; Flannery, B.P.; Teukolsky, S.A.; Vetterling, W.T. ".*Numerical recipes in C. Cambridge University Press*", 1998. [Consultado el 14 de marzo de 2008]

27. Patrenahalli, M.; Fukunaga, N.; Fukunaga, K. "A *branch and bound algorithm for feature subset selection."* IEEE Transactions on Computers, 26(9):917–922, sep 1977. [Consultado el 14 de marzo de 2008]

28. Deekers, A.; Aarts, E., "Global *Optimization and Simulated Annealing, Mathematical Programming*" 50, 1991, pp. 367-393. [Consultado el 12 de abril de 2008]

29. Carreras, X. y Romero, E. Márquez L. "*Máquinas de Vectores Soporte*". [aut. libro] J., Ramírez, M. y Ferri, C Hernández. "Introducción a la Minería de Datos". España : Pearson, 2004. [Consultado el 10 de abril de 2008]

30. Colectivo de Autores. "*Perfeccionamiento de la matriz de confusión que resulta de un clasificador, en dependencia del dominio de aplicación*". Cuba, Santa Clara, 2007. [Consultado el 6 de marzo de 2008]

31. Durán Acevedo, C. M., "Diseño y optimización de los subsistemas de un sistema de olfato electrónico para aplicaciones agroalimentarias e industriales." pág 40.(2000) [Consultado el 6 de marzo de 2008]

32. Metrópolis, N; Rosenbluth, A.; Teller, A.; Teller, E. *"Equations of state calculations by fast computing machines*". The journal of chemical physics, Vol. 21, No. 6.1953. [Consultado el 14 de mayo de 2009].

33. Kirkpatrick, S.; Gelatt, C. D. and Vecchi, M. P. *"Optimization by Simulated Annealing"*. Science 220, pp. 671-680. 1983. [Consultado el 14 de mayo de 2009].

34. Gutiérrez, M. Á.; De los Cobos, S. G.; Pérez Salvador, B. R. *"Optimización con recocido simulado para el problema de conjunto independiente"*. Revista En Línea. Universidad Autónoma Metropolitana. México, 1998. [Consultado el 14 de noviembre de 2009]. Disponible en: http://www.azc.uam.mx/publicaciones/enlinea2/3-2.html

35. [Consultado el 12 de octubre de 2009] Disponible: http://www.cimat.mx/~horebeek/cursus/node40.html

36. Quinlan, J. R., "*Programs for Machine Learning."* C 4.5 Morgan Kaufmann, 1993. [Consultado el 12 de enero de 2008]

37. Kirpatrick, S., Gelatt, C. and Vecchi, M. *"Optimization by Simulated Annealing, Science"*, vol. 220, pp. 672-680, 1983.[Consultado el 16 de enero de 2008]

38. Carrasco Velar, R. "*Nuevos descriptores atómicos y moleculares para estudios de estructura-actividad: Aplicaciones."* Ciudad de La Habana: s.n., 2003. [Consultado el 13 de septiembre de 2009]

39. Pedemonte, M.; Nesmachnow, S. *"Estudio Empírico de Operadores deCruzamiento en un Algoritmo Genético Aplicado al Problema de Steiner Generalizado"*, Universidad de Málaga, España, 2003. [Consultado el 16 de noviembre de 2009] Disponible en: http://www.fing.edu.uy/~sergion/gp/documentos/propios/EEOGSP.pdf

40. Dowsland, K.A.; Adenso Díaz, B. *"Heuristic design and fundamentals of the Simulated Annealing"*, Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No.19, pp. 93-102, ISSN: 1137-3601, 2003. [Consultado el 16 de noviembre de 2009] Disponible en: http://www.mac.cie.uva.es/~arratia/cursos/UVA/Enfriamiento-simulado.pdf

Bibliografía

- ✓ Escalona Arranz, JC; Carrasco Velar, R; Padrón García, JA. *"Introducción al diseño de fármacos*". Ciudad de La Habana: Editorial Universitaria, 2008. ISBN 978-959-16-0647-1.
- ✓ Garey, M. R.; Johnson, D. S., *"Computers and Intractability : a Guide to the Theory of NP-Completeness".* W. H. Freeman and Co., San Francisco, California, 1979
- ✓ Kohavi, R.; George H. J., *"Wrappers for feature subset selection. Artificial Intelligence*", 1997
- ✓ Langley, P., *"Selection of relevant features in machine learning".* In Procedings of the AAAI Fall Symposium on Relevance, páginas 1–5, New Orleans, LA, USA, 1994
- ✓ Dash, M.; Liu, H*., "Feature selection for classification. Intelligent Data Analysis",* 1(1-4):131–156, 1997
- ✓ Kohavi, R., *"Feature subset selection as search with probabilistic estimates."* In AAAI Fall Symposium on Relevance, páginas 122–126, November 1994
- ✓ Davies, S.; Russell, S. "*Np-completeness of searches for smallest possible feature sets*". In AAAI Press, editor, Proceedings of the 1994 AAAI Fall Symposium on Relevance, páginas 37–39, 1994
- ✓ Brassard, G.; Bratley,T., "*Fundamentos de Algoritmia".* Prentice Hall, 1997
- ✓ Liu, H.; Setiono, R. , "*A probabilistic approach to feature selection - a filter solution"*. In International Conference on Machine Learning, páginas 319–327, 1996
- ✓ Liu, H.; Setiono, R., *"Feature selection and classification - aprobabilistic wrapper approach."*
- ✓ Durán Acevedo, C. M., *"Diseño y optimización de los subsistemas de un sistema de olfato electrónico para aplicaciones agroalimentarias e industriales."* pág 40., 2000

- ✓ Coello Coello, CA. *"Introducción a los Algoritmos Genéticos".* Disponible en: http://www.redcientifica.com/doc/doc199904260011.html
- ✓ Quinlan, J. R., "Induction of decision trees. Machine Learning", 1:81–106, 1986.
- ✓ Metrópolis, N; Rosenbluth, A.; Teller, A.; Teller, E. *"Equations of sta85te calculations by fast computing machines".* The journal of chemical physics, Vol. 21, No. 6,1953
- ✓ Gutiérrez, M. Á.; De Los Cobos, S. G.; Pérez Salvador, B. R. "*Optimización con recocido simulado para el problema de conjunto independiente*". Revista En Línea. Universidad Autónoma Metropolitana. México, 1998. Disponible en: http://www.azc.uam.mx/publicaciones/enlinea2/3-2.html
- ✓ Breiman, L., "*Classification and regression trees."* Chapman & Hall, 1998
- ✓ Kira, K; Rendell, L. A., *"A practical approach to feature selection".* In Proceedings of the Ninth International Conference on Machine Learning, páginas 249–256, Aberdeen, Scotland, 1992
- ✓ Kononenko, I., "*Estimating attributes: Analysis and extensions of RELIEF*". In European Conference on Machine Learning, páginas 171–182, 1994.
- ✓ Schlimmer, J.C., "*Efficiently inducing determinations: A complete and efficient search algorithm that uses optimal pruning"*. In Proceedings of the Tenth International Conference on Machine Learning, pages 284–290, New Brunswick, NJ, 1993
- ✓ Almuallim, H.; Dietterich, G.T., "*Learning with many irrelevant features*." In Proceedings of the Ninth National Conference on Artificial Intelligence, volume 2, pages 547–552, San Jose,CA, 1991
- ✓ Liu, H; Motoda, H; Dash, M . "*A monotonic measure for optimal feature selection."* In European Conference on Machine Learning, páginas 101–106, 1998.
- ✓ Pawlak, Z. "*Rough Sets, Theoretical aspects of reasoning about data*". Kluwer Academic Publishers, 1991.

- ✓ Cover, T. M.; Thomas, J. A. *"Elements of Information theory".* Wiley-Interscience, 1991.
- ✓ Sheinvald, J.; Dom, B.; Niblack, W. "*A modelling approach to feature selection".* In 10th International Conference on Pattern Recognition, volume i, pages 535–539, 1990.
- ✓ Glover, F. "*Future Paths for Integer Programming and Links to Artificial Intelligence, Computers and Operations Research"*. 13, 533-549, 1986.
- ✓ Craiman, L., *"UML y Patrones Introducción al análisis y al diseño orientado a objetos*". Prentice hall, 1999.
- ✓ Kohavi, R., *"Feature subset selection as search with probabilistic estimates*." In AAAI Fall Symposium on Relevance, páginas 122–126, November 1994.
- ✓ Davies, S.; Russell, S. *"Np-completeness of searches for smallest possible feature sets"*. In AAAI Press, editor, Proceedings of the 1994 AAAI Fall Symposium on Relevance, paginas 37–39, 1994
- ✓ Brassard, G.; Bratley,T., *"Fundamentos de Algoritmia*". Prentice Hall, 1997
- ✓ Quinlan, J. R., "*Induction of decision trees. Machine Learning*", 1:81–106, 1986
- ✓ Moreno Pérez, J A.; Melián Batista, B. "*Metaheurísticas para la planificación logística"*, Grupo de Computación Inteligente. Universidad de La Laguna, 2005
- ✓ Glover, F. *"Future Paths for Integer Programming and Links to Artificial Intelligence, Computers and Operations Research"*, 1986
- ✓ Coello Coello, CA., *"Introducción a los Algoritmos Genéticos*" Disponible en: http://www.redcientifica.com/doc/doc199904260011.html
- ✓ Press, W.H.; Flannery, B.P.; Teukolsky, S.A.; Vetterling, W.T. "*Numerical recipes in C. Cambridge University Press"*, 1998
- ✓ Rosales García, A. R.; Marrero López, Y. *"Propuesta del diseño arquitectónico de la plataforma GRAph-TOol".*

- ✓ Carrasco Velar, R. "*Nuevos descriptores atómicos y moleculares para estudios de estructura-actividad: Aplicaciones."* Ciudad de La Habana: s.n., 2003.
- ✓ Pedemonte, M.; Nesmachnow, S. *"Estudio Empírico de Operadores deCruzamiento en un Algoritmo Genético Aplicado al Problema de Steiner Generalizado"*, Universidad de Málaga, España, 2003. Disponible en: http://www.fing.edu.uy/~sergion/gp/documentos/propios/EEOGSP.pdf
- ✓ 40. Dowsland, K.A.; Adenso Díaz, B. *"Heuristic design and fundamentals of the Simulated Annealing"*, Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No.19, pp. 93-102, ISSN: 1137-3601, 2003. Disponible en: http://www.mac.cie.uva.es/~arratia/cursos/UVA/Enfriamiento-simulado.pdf

Anexo 1: Búsqueda bibliográfica en google y EBSCO

Palabra Clave	No. Registros Google	No. Registros EBSCO
Structure Activity Relationship	60 000 000	9300
Variable reduction	912 000	9164
Genetic Algorithms	94 600	8000
Hybrid Greddy	5 190	7324

Anexo 2: Resultados para costo 100 y Cross Validation 10

nu	gamma	C-SVC	nu-SVC
0,5	0,1	65	71
0,1	0,1	65	65
0,5	0,5	65	65
0,1	0,5	65	65
0,01	0,1	65	65
0,01	0,5	65	65
0,01	0,99	65	65
0,01	0,01	65	65
0,5	0,01	65	65
0,99	0,01	65	65
0,8	0,01	65	65
0,9	0,01	65	65
0,95	0,01	65	65
0,88	0,01	65	65
0,1	0,9	65	65
0,5	0,9	68	68
0,3	0,9	67	53
0.9	**0.5**	**71**	**71**

Anexo 3: Resultados para nu 0.9, gamma 0.5 y cross validation 10

Costo	C-SVC	nu-SVC
1	74	71
10	71	71
100	71	71
1000	71	71

Anexo 4: Resultados de la Clasificación para costo 100,nu 0.9 y ganma 0.5

Valor de Validación	MSV	% clasificación
2	C-SVC	54
	nu-SVC	56
5	C-SVC	52
	nu-SVC	57
10	C-SVC	52
	nu-SVC	57
20	C-SVC	55
	nu-SVC	56
100	C-SVC	57
	nu-SVC	56

Glosario de términos

A

Actividad biológica: Actividad que caracteriza el comportamiento biológico en compuestos químicos (Molécula o Fragmento).

B

Bioinformática: Es la aplicación de los ordenadores y los métodos informáticos en el análisis de datos experimentales y simulación de los sistemas biológicos.

C

Compuestos Orgánicos: Compuestos cuya composición fundamental es sobre la base del elemento químico carbono.

D

Descriptor: Número que caracteriza estructuralmente la molécula.

G

GPL: Acrónimo de General Public Licence (Licencia pública general de GNU).

N

NP-completo: En teoría de la complejidad computacional, la clase de complejidad NP-completo es el subconjunto de los problemas de decisión en NP tal que todo problema en NP se puede reducir en cada uno de los problemas de NP-completo. Se puede decir que los problemas de NP-completo son los problemas más difíciles de NP (no polinomial) y muy probablemente no formen parte de la clase de complejidad.