SISTEMA DE RAZONAMIENTO BASADO EN CASOS COMO SOPORTE AL DIAGNÓSTICO MÉDICO MEDIANTE CLASIFICACIÓN DE DATOS MULTI-CLASE

MABEL XIMENA ORTEGA ADARME
DIANA MARCELA VIVEROS MELO

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
SAN JUAN DE PASTO
2017

SISTEMA DE RAZONAMIENTO BASADO EN CASOS COMO SOPORTE AL DIAGNÓSTICO MÉDICO MEDIANTE CLASIFICACIÓN DE DATOS MULTI-CLASE

MABEL XIMENA ORTEGA ADARME
DIANA MARCELA VIVEROS MELO

Trabajo de grado para optar por el título de Ingenieros Electrónicos

ASESOR
PhD. DIEGO HERNÁN PELUFFO ORDÓÑEZ
INGENIERO ELECTRÓNICO

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
SAN JUAN DE PASTO
2017

NOTA DE RESPONSABILIDAD

“Las ideas y conclusiones aportadas en el siguiente trabajo de grado son responsabilidad exclusiva de los autores.”

Acuerdo 1. Artículo 324. Octubre 11 de 1966, emanado del Honorable Consejo Directivo de la Universidad de Nariño.

Nota de aceptación

_____

_____

_____

_____

_____

_____

Firma del presidente del jurado

_____

Firma del jurado

_____

Firma del jurado

San Juan de Pasto, 11 de febrero de 2017

RESUMEN

El razonamiento basado en casos (CBR) es un proceso que intenta imitar el comportamiento de un humano experto en la forma de tomar decisiones con respecto a un caso y aprender de la experiencia de casos anteriores. En particular, CBR ha demostrado ser una metodología adecuada para aplicaciones en el ámbito médico. La mayoría de enfoques basados en CBR disponibles están destinados a diagnosticar pacientes, clasificándolos en dos categorías: La presencia o ausencia de una patología; sin considerar que pueden existir subcategorías e incluso surgir categorías nuevas. Esto conlleva a que la clasificación se realice de manera forzada y a veces imprecisa en presencia de casos extraños. Otra desventaja de los sistemas de CBR convencionales radica en que la asignación de los casos se hace de forma discreta sin determinar el grado o la probabilidad de pertenencia para facilitar la toma de decisiones en el soporte diagnóstico.

En este trabajo, se presenta una extensión del esquema convencional de CBR en escenarios multi-clase. Para este fin, se lleva a cabo un estudio comparativo del desempeño de algunos clasificadores multi-clase, y así, identificar el mejor de ellos para ser integrado en aplicaciones de CBR. Una contribución importante de este trabajo es la estimación de probabilidades de pertenencia para los nuevos casos. La metodología de CBR propuesta mejora la usabilidad con respecto a enfoques convencionales y, además, proporciona información más significativa al experto en la etapa de revisión.

ABSTRACT

Case-based reasoning (CBR) is a process that attempts to mimic the behavior of a human expert in decision-making task regarding a single case and to learn from the experience of previous cases. In particular, CBR has proven to be a suitable tool for applications in the medical field. The majority of available CBR-based approaches are intended to help in the diagnosis of patients. By classifying them into two categories: The presence or absence of a pathology; without considering that subcategories may exist and even new categories arise. This means that the classification is forced and sometimes imprecise in the presence of strange cases. Another disadvantage of conventional CBR systems is the typical assignment of cases to classes, which is done discreetly with no consideration of the degree or likelihood of membership to facilitate decision making in the diagnostic support.

In this paper, we present an extension of the conventional CBR scheme to multi-class scenarios. For this purpose, a comparative study of the performance of some multi-class classifiers is performed, and thus, identify the best of them to be integrated in CBR applications. An important contribution of this work is the estimation of probabilities of membership for the new cases. The proposed CBR methodology improves usability with respect to conventional approaches and, in addition, provides more meaningful information to the expert in the review stage.

TABLA DE CONTENIDO

LISTA DE TABLAS

LISTA DE FIGURAS

LISTA DE ANEXOS

GLOSARIO

Razonamiento: Es el proceso de organizar y estructurar las ideas para obtener respuestas y resoluciones a los problemas de cualquier índole.

Clasificación: Asignación de un objeto a una de las diversas categorías o clases especificadas.

Clase: Agrupación de objetos que tiene características comunes.

Base de casos: Es la materia prima del sistema de predicción. Es el histórico de casos que se usa para entrenar al sistema que detecta los patrones. El conjunto de casos se compone de instancias o muestras, y las instancias de factores, características o propiedades.

Instancia: Es cada uno de los datos de los que se disponen para hacer un análisis.

Características: Son los atributos que describen cada una de las instancias del conjunto de datos.

La inteligencia artificial: Es un área multidisciplinaria que combina ramas de la ciencia como la lógica, la computación y la filosofía que se encarga de diseñar y crear entidades artificiales que son capaces de resolver problemas o realizar tareas por sí mismos, utilizando algoritmos y paradigmas de comportamiento humano.

Algoritmo: Conjunto definido de reglas o procesos que llevan a la solución de un problema en un número determinado de pasos.

Aprendizaje automático: Es la rama de la inteligencia artificial que se dedica al estudio de los agentes/programas que aprenden o evolucionan basados en su experiencia, para realizar una tarea determinada cada vez mejor. El objetivo principal de todo proceso de aprendizaje es utilizar la evidencia conocida para poder crear una hipótesis y poder dar una respuesta a nuevas situaciones no conocidas.

Diagnóstico médico: Parte de la medicina que tiene por objetivo identificar una enfermedad basándose en los síntomas que presenta el paciente, el historial clínico y los exámenes complementarios.

Pre-procesamiento de los datos: Se refiere a la selección, la limpieza, el enriquecimiento, la reducción y la transformación de las bases de datos.

Diagrama de dispersión: Un diagrama de dispersión o gráfica de dispersión o gráfico de dispersión es un tipo de diagrama matemático que utiliza las coordenadas cartesianas para mostrar los valores de dos o tres variables para un conjunto de datos.

Selección de características: Hace referencia al proceso de reducir las entradas para su procesamiento y análisis, o de encontrar las entradas más significativas.

Desbalance de clases: Se presenta cuando existen conjuntos de datos que tienen una cantidad grande de datos de cierto tipo (clase mayoritaria), mientras que el número de datos del tipo contrario es considerablemente menor (clase minoritaria).

ACRÓNIMOS

CBR Case-based reasoning (Razonamiento basado en casos)

SVM Support Vector Machine (Máquinas de soporte vectorial)

K-NN Nearest Neighbour (Vecinos cercanos)

ANN Artificial Neural Networks (Redes neuronales artificiales)

PC Parzen classifier (Clasificador de Parzen)

SMOTE Syntetic Minority Over-sampling Technique (Técnica de sobremuestreo sintético minoritario)

CfsSubsetEval Criterio de evaluación de subgrupos basado en correlación

MSE Mean Squared Error (Error cuadrático medio)

FCF Frecuencia cardíaca fetal

CU Contracción uterina (CU),

Se Sensibilidad

Sp Especificidad

CP Porcentaje de clasificación

ROC Receiver Operating Characteristic (Característica operativa del receptor).

INTRODUCCIÓN

El aprendizaje a partir de la experiencia es un proceso que se da de forma natural en los seres humanos, y el conocimiento generado con dicho proceso se convierte en la base para establecer soluciones a problemas cotidianos. Pretendiendo emular esta habilidad del ser humano, ha surgido el razonamiento basado en casos (*Case-based reasoning - CBR*), que es una metodología utilizada para el procesado en computadores, que intenta imitar el comportamiento de un ser humano experto en la toma de decisiones con respecto a alguna temática y aprender de la experiencia de casos pasados [1].

En el contexto de CBR, cuando un sistema se enfrenta a un problema, recuerda soluciones que funcionaron bien con problemas similares y las utiliza como punto de partida en la solución. Este método resuelve problemas nuevos adaptando soluciones que ya fueron utilizadas con éxito en problemas anteriores similares. Para hacer esto, se debe comparar el problema al que se enfrenta actualmente, con aquellos que se han resuelto satisfactoriamente en el pasado. Y una vez que se hayan recordado problemas parecidos al actual, se realiza algún tipo de adaptación a la solución propuesta para que funcione en la situación actual [2].

Es importante conocer cómo realizar analogías y usar el razonamiento porque permite extender resultados y conclusiones de un dominio a otro diferente o comprender una situación basándose en otra.

La metodología de CBR ha sido estudiada en diferentes campos y la medicina ha encontrado aplicaciones interesantes. De esta manera, el CBR es una herramienta eficaz para la solución de problemas existentes en el campo del diagnóstico médico [3], debido a que es una metodología que tiene como objetivo fundamental servir de apoyo al trabajo del médico en determinadas circunstancias cuando los síntomas representan el problema y el diagnóstico o el tratamiento son la solución. Este modelo requiere de gran precisión debido a la trascendencia que puede llevar consigo una decisión mal tomada. Por lo tanto, dicha actividad es realizada por médicos con una cierta cantidad de experiencia en el área en la que se esté realizando el diagnóstico. Para este propósito, se han venido aplicando diferentes ramas de la inteligencia artificial dentro de las cuales se pueden destacar CBR, sistemas expertos, redes neuronales, minería de datos, agentes Inteligentes, entre otros [4].

El CBR sugiere un modelo de razonamiento que incorpora los aspectos ya mencionados de resolución de problemas, mediante el entendimiento y el aprendizaje, e integra todo ello en procesos de memoria. En resumen, estas son las premisas subyacentes al modelo [5]:

- La referencia a casos pasados es de gran utilidad para tratar situaciones que vuelven a darse. La referencia a situaciones similares es necesaria a menudo para tratar la complejidad de una nueva situación. Por ello, recordar un caso para usarlo en un problema futuro (e integrar ambos) es necesariamente un proceso de aprendizaje.

- Debido a que las descripciones de los problemas son a menudo incompletas es necesario una etapa de entendimiento o interpretación. Ya que no puede llevarse a cabo un razonamiento, sin una resolución adecuada de una nueva situación, si ésta no se entiende con cierta completitud, se puede considerar que esta etapa es un prerrequisito y una parte del ciclo de razonamiento, pues el entendimiento de las situaciones mejora conforme progresa el razonador. No obstante, cualquier forma de razonamiento necesita que la situación sea elaborada con suficiente detalle y representada con suficiente claridad y con el vocabulario apropiado para que el razonador reconozca el conocimiento que necesita (sea conocimiento general o casos) para razonar a partir de él.

- La práctica demuestra que no suele existir un caso pasado exactamente igual a un caso nuevo. Por ello, es muy usual requerir de una adaptación que se debe aplicar a la solución pasada para que se ajuste a la nueva situación.

- El aprendizaje es una consecuencia natural del razonamiento. Si se halla un nuevo procedimiento en el curso de la resolución de un problema complejo y su ejecución resulta positiva, entonces se aprende el nuevo procedimiento para resolver esta nueva clase de situaciones.

La revisión de la solución propuesta y el análisis de dicha revisión son dos partes necesarias para completar el ciclo de razonamiento/aprendizaje. Este análisis de la revisión (habitualmente llevada a cabo por un agente externo) puede conllevar una reparación de posibles fallos.

En este trabajo, se propone desarrollar un sistema de razonamiento basado en casos multi-clase para asistencia diagnóstica, que brinde información útil a los especialistas en la salud, y así proporcionar a los usuarios, diagnósticos de múltiples clases y más cercanos a la realidad de acuerdo con el análisis histórico de los pacientes.

1. DESCRIPCION DE PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

Típicamente, en un problema de clasificación, se parte de un conjunto finito de casos y para cada caso se tiene un conjunto de observaciones de algunas características relevantes, de forma que la clasificación correcta se basa en la relación que existe entre cada una de ellas.

La metodología de CBR toma como modelo la forma de razonar del humano, y se basa sobre todo en el uso de la experiencia previa para afrontar problemas y situaciones nuevas. El objetivo es relacionar las observaciones y las clases y así poder determinar la clase a la que pertenece cualquier caso dado a partir de los valores de sus atributos [2]. Se han desarrollado algunos métodos de clasificación mediante CBR, no obstante, para aplicaciones de diagnóstico médico la mayoría de ellos han sido diseñados para clasificar entre dos clases: La presencia y ausencia de una patología, sin proveer información adicional que facilite al experto la toma de decisiones, especialmente, en casos extraños. En la literatura científica, se encuentra que el área de clasificación multi-clase en CBR es poco explorada, por tanto el diseño de las etapas de revisión y adaptación multi-clase es aún un problema abierto**.**

1.2. JUSTIFICACIÓN

Se ha visto que los expertos en un área determinada parecen seguir un patrón cuando encuentran un nuevo problema. Normalmente, tratan de recordar casos similares que se hayan visto en el pasado, recordando los resultados de dichos casos y en algunas ocasiones el razonamiento que llevó a dichos resultados. Algunos argumentan que es la base de los procesos cognitivos y el mecanismo por el cual los seres humanos solucionan problemas y aprenden del mundo circundante. Precisamente, la solución de problemas utilizando CBR consiste en desarrollar los sistemas basados en conocimiento, imitando la conducta de expertos humanos. Los últimos años del anterior siglo y las primeras décadas del presente, indican una tendencia en la evolución de las investigaciones en el campo de la salud, que jerarquiza el papel de los recursos humanos y la información en el análisis y evaluación de las estrategias científicas y sus resultados [6].

Actualmente, el CBR ha demostrado ser apropiado para aplicar estrategias de analogía en dominios poco estructurados y en aquellos donde la adquisición de conocimiento es difícil. Por lo tanto, la elección de esta metodología es ideal para

el desarrollo de sistemas de apoyo diagnósticos, particularmente, en dominios de alta complejidad conceptual debido al alto número de conceptos, interrelaciones, terminología e interdependencias entre sus elementos. Por ejemplo, en servicios médicos multidisciplinares, los casos clínicos pueden describir aspectos muy diferentes de la evolución de un paciente. Esta metodología está despertando bastante interés debido a que su uso es muy intuitivo, permite realizar procesos de aprendizaje, y en la comunidad científica se han construido algunos sistemas con bastante éxito en dominios complejos donde otras técnicas no han generado buenos resultados [2].

Lo que se busca con el desarrollo de este proyecto es una modificación de algunas etapas del CBR con el fin de realizar un proceso aprendizaje de casos complejos y multi-clase. Para ello, se realiza la integración de tres áreas: Representación de datos, clasificadores multi-clase y razonamiento basado en casos. Bajo el supuesto de que los datos son complejos (alta dimensión y estructura compleja), se propone utilizar en la etapa de pre-procesado técnicas adecuadas de representación de datos, es decir, de selección de características y balanceo de clases. Asimismo, en las etapas recuperación y adaptación, incorporar clasificadores multi-clase de forma que se realice la recuperación de casos con múltiples clases y de acuerdo con la naturaleza de los datos se obtenga la probabilidad o valor de pertenencia del nuevo caso con respecto a cada una de las clases o los casos conocidos. De esta forma, se obtiene una respuesta que ayuda al personal médico que se enfrenta ante un nuevo caso, a tomar una decisión de acuerdo con los posibles diagnósticos.

1.3. CONTRIBUCIONES DE ESTA TESIS

La medicina es un campo que, sin duda, necesita de todo tipo de aportaciones tecnológicas y científicas, ya que existen problemas enormes que han tomado años ser entendidos y tratados por parte de los especialistas de la salud. Una herramienta muy útil aplicada en este medio es el CBR debido a que facilita la toma de decisiones al experto para brindar un diagnóstico médico utilizando información de casos que han sido resueltos exitosamente. Generalmente, se ha desarrollado sistemas de CBR bi-clase en donde sólo puede identificarse la presencia o ausencia de una patología, por lo cual, se propone una metodología que consiste en realizar una extensión de CBR a escenarios multi-clase en donde se pueda dar un diagnóstico más preciso y obtener información del caso a tratar por medio de información significativa, como lo es, la estimación de probabilidades de pertenencia del nuevo caso a cada categoría.

En este trabajo de grado se desarrolla una interfaz gráfica de un CBR para asistencia diagnostica, con el objetivo de facilitar la visualización de información a los expertos en el campo de la medicina, y así, poder diagnosticar con mayor

precisión a los pacientes más cercanos a la realidad, teniendo en cuenta el análisis histórico de los mismos.

En el campo de investigación de CBR, este trabajo representa un aporte en el desarrollo de nuevas exploraciones, con ayuda de los algoritmos desarrollados.

1.4. ORGANIZACIÓN DEL DOCUMENTO

Este trabajo está dividido en 8 secciones principales nombradas de la siguiente manera: Introducción, descripción del problema, objetivos, marco teórico, metodología, marco experimental, resultados y conclusiones.

En la sección 2, se presenta el planteamiento del problema, la justificación de este trabajo y las contribuciones científicas de esta investigación.

En la sección 3, se presenta los objetivos que fueron planteados como logros con el desarrollo de esta investigación.

En la sección 4, se presenta una revisión bibliográfica que incluyen conceptos sobre el CBR y las etapas que lo conforman, además de las técnicas de pre-proceso de datos.

En la sección 5, se describe la metodología propuesta, así como la implementación del CBR.

En la sección 6, se describe las bases de datos, medidas de desempeño y pruebas realizadas.

En la sección 7, se da a conocer los resultados obtenidos de la investigación elaborada. El desarrollo de un CBR multi-clase con la estimación de las probabilidades para nuevos casos y su visualización por medio de una interfaz gráfica.

Finalmente, en la sección 8, se presenta las conclusiones que se obtuvieron a partir de este trabajo, así como los trabajos futuros que pueden mejorar la metodología de CBR propuesta en esta investigación.

2. OBJETIVOS

En esta sección se plantea los objetivos esperados con el desarrollo de esta tesis.

2.1. OBJETIVO GENERAL

Desarrollar un sistema de razonamiento basado en casos para múltiples categorías usando clasificadores multi-clase con el fin de dar soporte al diagnóstico médico.

2.2. OBJETIVOS ESPECÍFICOS

- Realizar un estudio comparativo de técnicas supervisadas de reconocimiento de patrones para clasificar datos multi-clase en entornos de razonamiento basado en casos.

- Proponer un enfoque para la estimación de probabilidades de pertenencia de los nuevos casos con respecto de cada clase, considerando la naturaleza de los datos, con el fin de brindar información útil para facilitar el diagnóstico.

- Integrar clasificadores multi-clase en el ciclo de vida de un sistema de razonamiento basado en casos para proporcionar una herramienta del soporte de asistencia médica.

3. MARCO TEÓRICO

3.1. RAZONAMIENTO BASADO EN CASOS

Un sistema de razonamiento basado en casos (CBR) es una metodología capaz de emplear el conocimiento específico adquirido en situaciones previas y utilizarlo en el presente. Para enfrentar el nuevo problema se realiza una búsqueda en la memoria, de casos similares resueltos en el pasado. Además, incrementa el conocimiento almacenando el nuevo caso para ser aplicado en situaciones futuras. Esto permite que el conocimiento se mantenga actualizado en todo momento [6]. Por lo tanto, un sistema de CBR necesita de una colección de experiencias, llamadas casos, almacenadas en una base de casos, donde cada caso se compone generalmente de una descripción del problema y la solución que se aplicó. Así, un sistema CBR se basa en dos grandes hipótesis: En primer lugar, la premisa de que un sistema puede ser un buen solucionador eficiente y efectivo de problemas sin necesidad de poseer un conocimiento completo de la relación que existe entre un problema y su solución, y, en segundo lugar, el hecho de que los problemas tienden a repetirse y por ello la experiencia es un recurso útil [7].

La metodología de CBR ha logrado buenos resultados en muchos campos de aplicación, entre ellos: Determinar el estado de operación de una planta industrial realizando procesos de planificación, diagnóstico y mantenimiento [8], [9], [10]; análisis de la solvencia empresarial [11]; generación de soluciones para negocios laborales y argumentos legales [2], entre otros.

En el ámbito del diagnóstico médico, los resultados han sido no solamente de interés práctico, sino también en el plano teórico, en donde el gran volumen de información que se ha registrado en las bases de casos ha sido una contribución importante. De esta manera, el razonamiento basado en casos es una herramienta poderosa para la solución de problemas existentes en este campo. En particular, desde esta perspectiva conceptual es posible [12]:

- Recordar la experiencia previa, que es particularmente útil para evitar la repetición de errores cometidos en el pasado, pues es posible alertar al razonador de forma que tome algunas medidas para evitar la repetición de errores anteriores.

- Lograr el aprendizaje que tiene lugar a partir de la información almacenada correspondiente a casos que fueron previamente resueltos exitosamente o no.

- Ofrecer soluciones a nuevos casos a partir del análisis de un razonador que interactúa con bases de casos.

- Focalizar el razonamiento hacia partes importantes del problema señalando los rasgos más significativos del asunto analizado.

El proceso de solución de problemas se inicia con el reconocimiento de la existencia de un problema, que puede manifestarse como una discrepancia entre el estado actual y el estado deseado, una contradicción entre lo que cabría esperar y lo que se observa o una carencia de conocimientos para explicar un fenómeno dado. Esta clarificación tiene lugar en un momento posterior en el que se alcanza una representación más adecuada de los aspectos críticos del problema, evidenciándose un grado superior de comprensión de su naturaleza y estructura [6].

3.1.1. Definición de caso y Base de Casos

El principal componente de un sistema CBR es la base de casos. Un caso es una porción de conocimiento que representa una experiencia concreta y el contexto en que sucedió. Es importante almacenar el contexto en que se produjo la experiencia porque ayuda a determinar cuándo es aplicable ese conocimiento [2]. El problema es la identificación de atributos que caracterizan al contexto y detectar cuándo dos contextos son similares.

Un caso es una experiencia que enseña algo, de tal manera puede haber experiencias que no aporten ninguna nueva información al sistema, con lo cual se plantea el problema de identificar cuándo dos casos son superficialmente distintos. Generalmente, un caso se compone de las siguientes partes [7]:

- La descripción de un problema, ya sea la situación a interpretar o el problema de planificación a resolver. Esta es la característica de los casos que se utiliza para determinar su similitud.

- La descripción de una solución. Además del plan aplicado o la interpretación asignada. En la solución de un caso se puede guardar información adicional, sobre todo con el objetivo de facilitar futuras adaptaciones. Información acerca del proceso que llevó a la obtención de la solución, qué alternativas se consideraron, cuáles se eligieron y cuáles se descartaron y el motivo de la elección o el rechazo de dicho caso.

- Resultados obtenidos, en donde se determina si ha sido un éxito o un fracaso, si ha cumplido o no las expectativas, si es una estrategia de reparación o una referencia a la solución del problema [2].

3.1.2. Tareas y sistemas representativos

Las aplicaciones de CBR se clasifican principalmente en dos tipos: Tareas de clasificación y tareas de síntesis [5].

En las tareas de clasificación, el objetivo es relacionar las observaciones y las clases, y así poder determinar la clase a la que pertenece cualquier objeto dado a partir de los valores de sus atributos. De esta forma, cuando se presenta un caso nuevo al sistema, éste tiene como información el conjunto de valores que presentan los atributos de ese caso. Y a partir de ese conjunto de observaciones, el CBR es capaz de determinar correctamente la clase a la que pertenece el caso [2].

La mayoría de las herramientas de CBR disponible dan un soporte aceptable para las tareas de clasificación, que suelen estar relacionadas con la recuperación de casos. Existe una gran variedad de campos de acción, entre ellos [5]:

- Diagnosis: Médica o de fallos de equipos.
- Predicción: Pronóstico de fallos de equipos o actuación sobre las acciones o participaciones de un mercado.
- Valoración: Análisis de riesgos para bancos o seguros o estimación de costes de proyectos.
- Control de procesos: Control de fabricación de equipos.
- Planificación: Reutilización de planos de viaje o planificadores de trabajo.

Las tareas de síntesis intentan crear una nueva solución combinando partes de soluciones previas. Éstas son inherentemente complejas a causa de las restricciones de los elementos usados durante la síntesis. Los sistemas de CBR que realizan tareas de síntesis deben realizar adaptación y son normalmente sistemas híbridos que combinan CBR con otras técnicas. Algunas de las tareas que realizan estos sistemas son [5]:

- Diseño: Consiste en la creación de un nuevo artefacto adaptando elementos de otros existentes.
- Planificación: Es la creación de nuevos planes a partir de otros previos.
- Configuración: Se refiere a la creación de nuevos planificadores a partir de otros previos.

3.1.3. Ciclo del Razonamiento Basado en Casos

De acuerdo con Aamodt y Plaza [13], el ciclo general de CBR puede ser descrito en cuatro procesos: Recuperación *(Retrieve)*, Reutilización o Adaptación (*Reuse)*, Revisión *(Revise)* y Aprendizaje (*Retain)*. En la figura 1, se ilustra este ciclo.

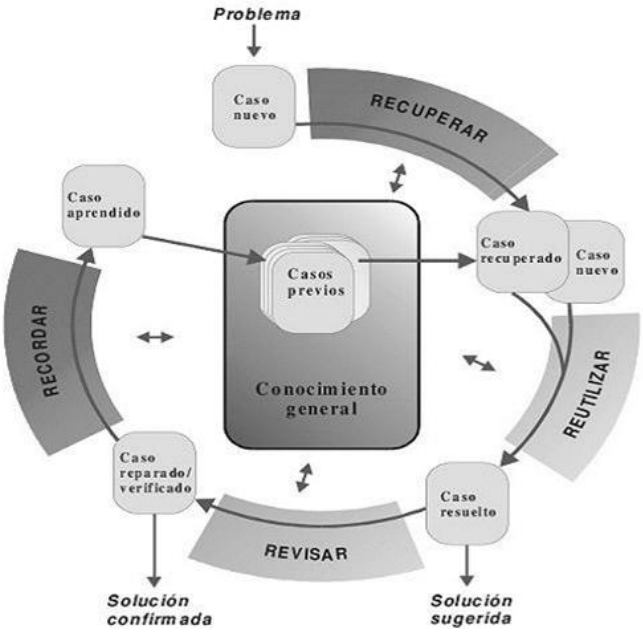


Figura 1. Ciclo de Razonamiento Basado en Casos. Un nuevo problema inicia con la recuperación de casos similares, posteriormente se llega a una etapa de adaptación, seguida de la revisión y finalmente el aprendizaje de la nueva experiencia.
Fuente: [14]

- **Recuperación *(Retrieve)***

En este paso se recuerdan las experiencias pasadas que pueden ser útiles. Se consulta la base de casos y se recupera uno o varios casos similares al problema actual, comparando dos casos y determinando el grado de similitud entre los mismos.

Los algoritmos de recuperación más utilizados son: k-vecinos más cercanos, árboles de decisión y sus derivados. Estas técnicas necesitan de una medida de similitud para determinar la proximidad entre casos por ejemplo la distancia euclídea, de hamming o de levenshtein [5]. En el caso de clustering se utilizan métricas como *gain* o entropía.

- **Reutilización o Adaptación *(Reuse)***

Las soluciones recuperadas en la etapa anterior, corresponden a aquellos problemas vividos en el pasado más parecidos al problema inicial. Aquí aparece el concepto adaptación, que se hace necesario cuando la solución recuperada no es directamente aplicable al problema en curso. En consecuencia las soluciones, en ocasiones han de ser adaptadas al nuevo problema para su correcto funcionamiento a la hora de ser reutilizadas [7].

Para identificar qué debe ser corregido se pueden usar varios métodos [2]:

- Diferencias entre las especificaciones del problema recuperado y el nuevo.
- Usar una lista de comprobaciones que se deben realizar.
- Detectar inconsistencias entre la solución recuperada y los objetivos y restricciones del problema nuevo.
- Prever los efectos que va a tener la solución, por ejemplo utilizando modelos, casos o simulación.
- Llevar a cabo la solución y analizar el resultado.

➢ **Revisión *(Revise)***

Los sistemas de CBR, como se ha dicho anteriormente, aprenden de nuevas experiencias, nuevos casos que aportan un enriquecimiento del sistema. Después de reutilizar una posible solución, esta puede ser correcta o no, si es correcta la nueva solución será almacenada en el sistema, pero si la solución no ha sido satisfactoria entonces se tendrá que revisar nuevamente [7].

De acuerdo con [7], la revisión de casos comprende básicamente dos fases:

- Evaluar la solución. Es decir, decidir si la solución dada es la correcta al problema planteado. Esta fase normalmente se realizará por algún método externo al sistema CBR, como por ejemplo el usuario. A veces es necesario realizar simulaciones.
- Reparar los fallos. Si no es correcta la solución se detecta los fallos y se corrigen.

➢ **Aprendizaje *(Retain)***

Una de las principales características de un sistema CBR es poder recordar los nuevos casos y su solución aplicada, para ello es fundamental poder almacenar estos casos en lo que se llama la base de casos. Dicha base de casos cada vez irá aumentando y enriqueciéndose gracias a las soluciones de problemas basados en la experiencia. La forma de estructuración de la base de casos y las políticas de aprendizaje del sistema de CBR facilitarán el buen funcionamiento. Por ello, el primer problema que debe tratar un sistema de aprendizaje es decidir de qué casos se aprenden. La eficiencia de un sistema CBR se puede degradar cuando el número de casos crece excesivamente y por tanto, se debe evitar incluir casos que no aporten información nueva al sistema. El rango de posibilidades va desde los sistemas que, de forma autónoma deciden qué casos deben incluir hasta los que delegan esta posibilidad en el mismo usuario. El segundo problema relacionado con el aprendizaje es la que se refiere a la organización de la estructura de casos. Dependiendo de la complejidad de la estructura utilizada, este proceso puede ser más o menos complicado [7]:

- Si la organización es lineal, basta con añadir un nuevo elemento a la lista.
- Si la estructura se induce a partir de los casos, será necesario redefinir la periodicidad de la indexación. Normalmente este proceso se realiza fuera de línea para no perturbar la interacción del usuario con el sistema.
- En los modelos más complejos donde se presentan generalizaciones de los casos, es necesario aplicar, técnicas de aprendizaje más sofisticadas, similares a las aplicadas en otros campos de inteligencia artificial.

3.1.4. Diagnóstico médico

Tal vez el mayor problema en el campo médico es realizar el diagnóstico de una enfermedad, uno de los inconvenientes más importantes en éste proceso es la subjetividad del especialista que lo realiza; este hecho se hace notar en particular en actividades de reconocimiento de patrones, donde la experiencia del profesional está directamente relacionada con el diagnóstico final, esto es debido al hecho de que el resultado no depende de una solución sistematizada sino de la interpretación de los síntomas del paciente [15].

En términos generales el diagnóstico médico es fundamentalmente el proceso de identificar la enfermedad que está sufriendo un paciente, con el fin de poder determinar cuál es la mejor forma para tratarla; siendo éste uno de los temas más explotados en la ciencia de la computación desde hace ya un tiempo, especialmente en el campo de la inteligencia artificial [4].

Para el diagnóstico en general y el diagnóstico médico particularmente la integración rápida y fácil de conocimiento que pueda reemplazar al conocimiento adquirido anteriormente es un factor fundamental, dado que el diagnóstico médico es un proceso muy complejo, que requiere la recopilación de los datos del paciente, un profundo entendimiento de la literatura médica alrededor del tema y muchos años de experiencia clínica [16]. Sin embargo, no se puede formular un diagnóstico preciso sin antes tener en cuenta diferentes opciones o alternativas o puede darse el caso en que varios especialistas en la salud brinden distintos diagnósticos para un caso particular, ya que cada uno de ellos puede tener un proceso de decisión diferente.

Con ello, la educación médica en su constante perfeccionamiento, requiere la introducción de técnicas avanzadas para preparar a un individuo capaz de mantenerse actualizado en su especialidad durante toda su vida [6] y el CBR es una herramienta que permite facilitar el diagnóstico de los pacientes, ya que permite desarrollar sistemas basados en conocimiento, emulando la capacidad natural que posee el hombre en la toma de decisiones.

3.2. APRENDIZAJE DE MÁQUINA (*Machine Learning*)

El aprendizaje de máquina se basa en la creación de programas que sean capaces de generalizar comportamientos en una máquina, en base a una información que no está estructurada y se suministra a modo de ejemplos [17]. El objetivo del aprendizaje de máquina, es que los computadores sean capaces de procesar y ejecutar algoritmos teniendo en cuenta cualquier tipo de variable [18]. Este aprendizaje, es posible gracias a la detección de patrones dentro de un conjunto de datos, de manera que es el sistema computacional es el que predice qué situaciones podrían darse o no. Finalmente, a partir de este aprendizaje es posible tomar decisiones y resultados fiables [19].

3.2.1. Clasificación Supervisada

Este tipo de clasificación cuenta con un conocimiento a priori, es decir para la tarea de clasificar un objeto dentro de una categoría o clase contamos con modelos ya clasificados. Podemos diferenciar dos fases dentro de este tipo de clasificación. La primera fase tenemos un conjunto de entrenamiento o de aprendizaje (para el diseño del clasificador) y otro llamado de *test* o de validación (para clasificación), que servirán para construir un modelo o regla general para la clasificación. En la segunda fase, es el proceso en sí de clasificar los objetos o muestras de las que se desconoce la clase a las que pertenecen [20].

Ejemplos de clasificación supervisada son: el diagnóstico de enfermedades, predicción de quiebra o bancarrota en empresas, reconocimiento de caracteres escritos a mano, etc.

Entre algunos de los clasificadores se encuentran: **Máquinas de soporte vectorial (SVM -** *Support Vector Machine***)**, ***K*- Vecinos más cercanos (*K*-NN -** *Nearest Neighbour***)**, **Redes neuronales artificiales (ANN -** *Artificial Neural Networks***)**, **Clasificador de parzen (PC –** *Parzen classifier***).**

3.2.2. Clasificación No Supervisada

En este tipo de clasificación se cuenta con muestras que tienen un conjunto de características, de las cuales no se conoce a qué clase o categoría pertenece. La finalidad es el descubrimiento de grupos de “objetos” cuyas características sea muy similares (semejanza intragrupo) y a la vez encontrar elementos distintos de otro grupo (distancia entre grupos), con el fin de separar las diferentes clases [20]. Estos mecanismos de clasificación basan su efecto en la búsqueda de clases con suficiente separabilidad espectral como para conseguir diferenciar unos elementos de otros [21]. Esta clasificación se puede hacer usando fórmulas de distancia espectral mínima, utilizando los promedios arbitrarios o el promedio de las firmas espectrales existentes [22].

Entre algunos métodos de esta clasificación se encuentran los siguientes:

- **K-medias:** El propósito es minimizar una función objetivo que defina qué tan buena es la solución del agrupamiento. La idea principal es definir k centroides (uno para cada grupo) y luego tomar cada punto de la base de datos y situarlo en la clase de su centroide más cercano. El próximo paso es recalcular el centroide de cada grupo y volver a distribuir todos los objetos según el centroide más cercano. El proceso se repite hasta que ya no hay cambio en los grupos de un paso al siguiente [23].

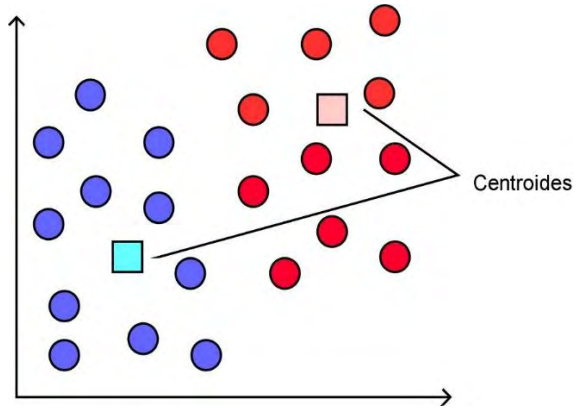


Figura 2. Agrupación de datos con k-medias.

- **Agrupamiento paramétrico basado en densidades (PDBC):** Estos métodos identifican *clusters* de formas arbitraria y son robustos frente a la presencia de ruido. El agrupamiento se realiza de tal manera que los grupos que se forman tienen una alta densidad de puntos en su interior mientras que entre ellos aparecen zonas de baja densidad [24].

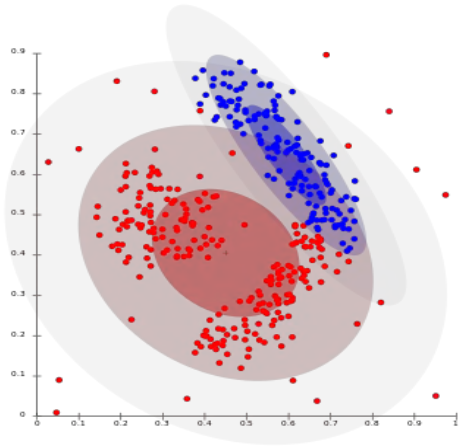


Figura 3. Agrupamiento de datos con PDBC. Los grupos se definen como áreas de densidad mayor que el resto del conjunto de datos. Por lo general se considera que los puntos de ruido y de frontera.
Fuente: [25]

3.3. PRE-PROCESAMIENTO DE DATOS

Generalmente, los datos reales pueden ser impuros o redundantes por diferentes razones, entre las cuales se encuentran: datos ruidosos, inconsistentes o incompletos. Lo que conduce a extracción de patrones erróneos o poco útiles. Con el pre-procesamiento de datos se pretende que los datos que van a ser utilizados en tareas de análisis o descubrimiento de conocimiento conserven su coherencia.

Con frecuencia, el pre-procesamiento de los datos tiene un impacto significativo en el desempeño general de los algoritmos de aprendizaje supervisado. Aplicar algunas técnicas de pre-procesamiento permite que los algoritmos de aprendizaje sean más eficientes [26].

Las técnicas más comúnmente utilizadas en la solución de los inconvenientes con los datos son: Limpieza, reducción, integración, selección y transformación de datos [27], [28], [29].

Figura 4. Diagrama de bloques de pre-procesamiento de datos, en donde se obtiene los datos que van a ser utilizados para posteriores tareas.

3.3.1. Normalización

El atributo es escalado a un rango específico, normalmente de -1 a 1, o de 0 a 1.

La normalización es empleada cuando se tienen atributos con órdenes de magnitud muy diferentes. Gracias a la normalización se evita que los atributos con valores más altos ganen un peso significativamente más importante en el modelo final que aquellos con valores más bajos [30].

3.3.2. Selección de características

Las técnicas de minería de datos que extraen modelos a partir de ejemplos tienden a obtener modelos complejos conforme crece el volumen de datos del conjunto sobre el cual se aplican. El elevado tamaño de los conjuntos de datos provoca inconvenientes adicionales tales como: Aumento en el tiempo de respuesta de los modelos, aumento en la sensibilidad al ruido y la posibilidad de sobreajuste de los modelos sobre el conjunto de entrenamiento. Al extraerse modelos de gran tamaño, la solución obtenida es poco comprensible para la

mente humana. Se hace por tanto necesario un pre-procesamiento previo que disminuya el tamaño del conjunto almacenado [27].

Debido al efecto negativo de atributos irrelevantes en la mayoría de esquemas de aprendizaje automático, es común que se lleve a cabo un proceso de selección de atributos previo al aprendizaje. La selección de características, consiste en escoger las muestras más representativas de un conjunto determinado. Disminuyendo el conjunto inicial de datos, se consigue reducir tanto la complejidad en tiempo de cálculo, como los recursos de almacenamiento. La eliminación de instancias no produce una degradación de los resultados, ya que podemos estar eliminando ejemplos repetidos o ruidosos y por tanto, evitar el sobre aprendizaje.

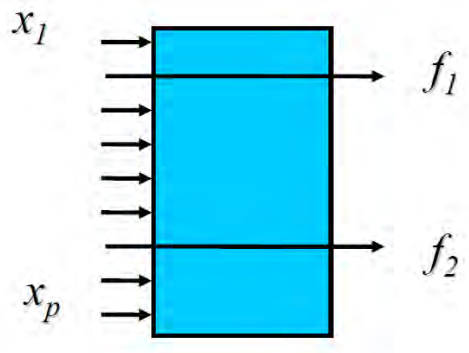


Figura 5. Selección de características. Se parte de un conjunto de x_n atributos y con el proceso de selección se obtienen f_n atributos.

El proceso de selección se realiza dependiendo del tipo de variables con las que se está trabajando. Pueden ser:

Variables ordinales: Corresponde a variables cuyos valores no son números, pero se pueden ordenar. Los atributos que se registran pueden mantener entre si una relación de jerarquía, pero estas relaciones no permiten más que una cantidad determinada de análisis, como por ejemplo puede ser el año de escolaridad, la pertenencia a un grupo socioeconómico o un grupo etario. Cuando elaboramos instrumentos que contemplen rotulaciones de atributos o estimaciones cualitativas estamos en presencia de variables que poseen jerarquía.

Variables cardinales: Son aquellas variables en donde su valor tiene pleno significado numérico, es decir que no sólo presentan las propiedades ordinales de los números, sino también las cardinales, dichas variables se dividen en:

Continuas: Variables que pueden tomar cualquier valor dentro de cualquier intervalo (edad, salarios, estatura, producción anual, etc.)

Discretas: Aquellas que toman solo algunos valores dentro de algún intervalo (hijos por familia, número de huelgas anuales, producción mensual de automóviles)

3.3.3. Balanceo de datos

En clasificación, el problema de desbalance de clases se presenta de manera natural en diversos dominios del mundo real [31]. En medicina, se ha observado que ciertas enfermedades afectan a un número reducido de personas; por lo que el número de expedientes de personas que las padecen es reducido, comparado con el total de expedientes médicos. En este tipo de escenarios, es de vital importancia que los sistemas de reconocimiento automático puedan predecir correctamente instancias de clase minoritaria y, al mismo tiempo, que no dañen la precisión de las predicciones para la clase mayoritaria [32].

La complejidad de los datos juega un papel importante en este tipo de problemas. Que puede entenderse como el traslape entre clases, la falta de datos representativos en algunas regiones del espacio de entrada o la existencia de subconceptos [33]. En estas circunstancias, los clasificadores presentan una tendencia de clasificación hacia la clase mayoritaria, maximizando de ésta manera el error de clasificación y clasificando correctamente instancias de clase mayoritaria en detrimento de instancias de clase minoritaria [34].

Para mejorar el desempeño de sistemas de reconocimiento de patrones en conjuntos de datos desbalanceados, se han propuesto soluciones que intentan balancear o limpiar los datos antes de aplicarlos a métodos de clasificación existentes. Estas soluciones son llamadas métodos externos y trabajan con los datos en una etapa de pre-procesamiento. En otras propuestas, se modifican los algoritmos de clasificación con la finalidad de incluir en ellos un mecanismo para hacer que las instancias de la clase minoritaria sean consideradas de mayor importancia que el resto [32].

Entre los métodos para el tratamiento de datos desbalanceados se encuentran:

- **Sobremuestreo *(Oversampling)***

Se refiere a balancear la distribución de las clases añadiendo muestras de la clase minoritaria, algunos algoritmos representativos son:

SMOTE: (*Syntetic Minority Over-sampling Technique)* Genera nuevas instancias de la clase minoritaria interpolando los valores de las instancias minoritarias más cercanas a una dada [34].

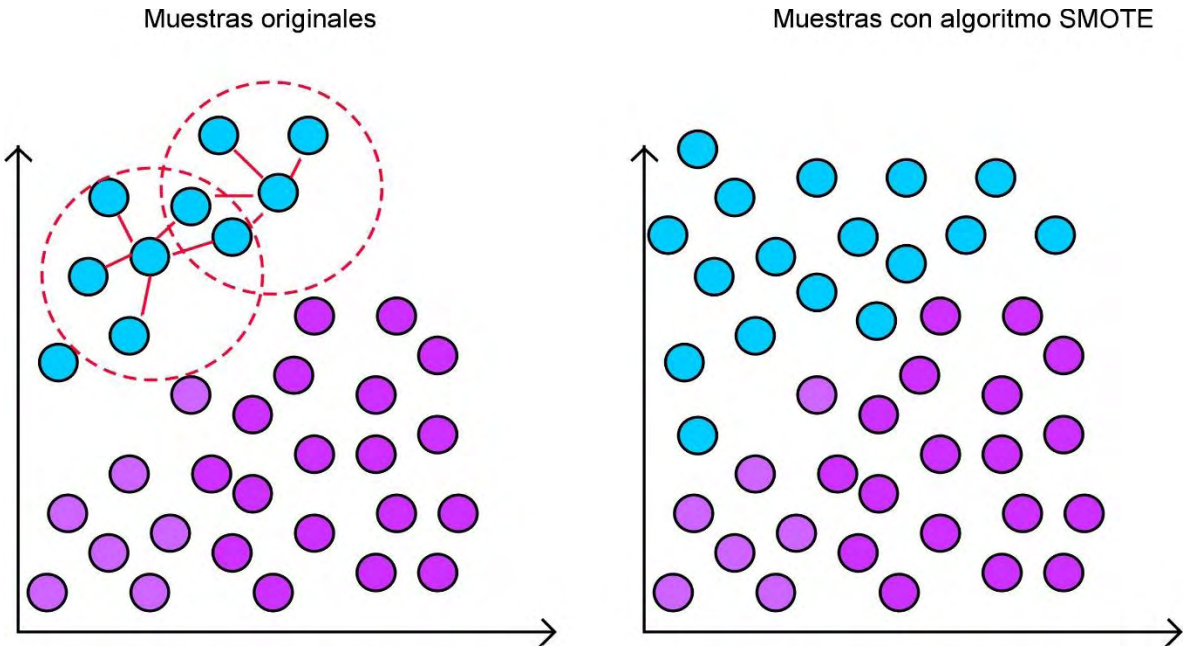


Figura 6. Algoritmo SMOTE. Esta técnica genera muestras sintéticas de la clase minoritaria, teniendo en cuenta los vecinos cercanos a una muestra dada.

Remuestreo *(Resampling)*: Duplica al azar instancias de la clase minoritaria.

El sobremuestreo tiene la ventaja de no perder información pero puede repetir muestras con ruido además de aumentar el tiempo necesario para procesar el conjunto de datos [34].

- **Submuestreo *(Undersampling)***

Consiste en eliminar muestras de la clase mayoritaria de manera aleatoria hasta balancear la distribución de las clases, entre estos algoritmos se destacan:

Submuestreo aleatorio *(Random undersampling)*: Elimina al azar instancias de la clase mayoritaria.

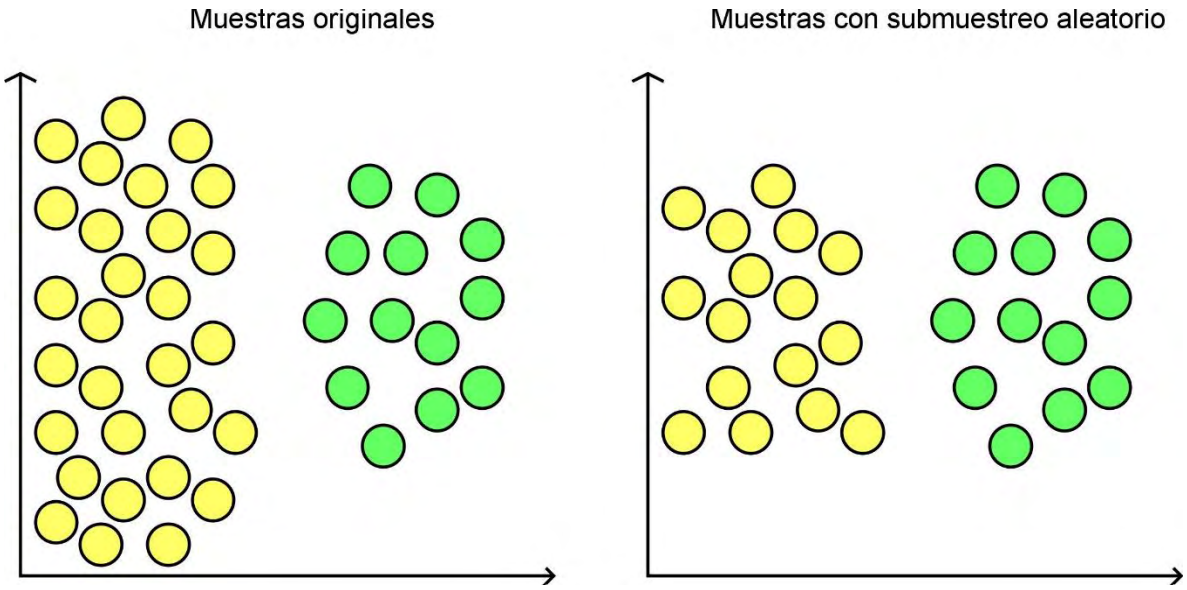


Figura 7. Submuestreo aleatorio. Esta técnica elimina muestras al azar de la clase mayoritaria.

Tomek Links: Elimina sólo instancias de la clase mayoritaria que sean redundantes o que se encuentren muy cerca de instancias de la clase minoritaria.

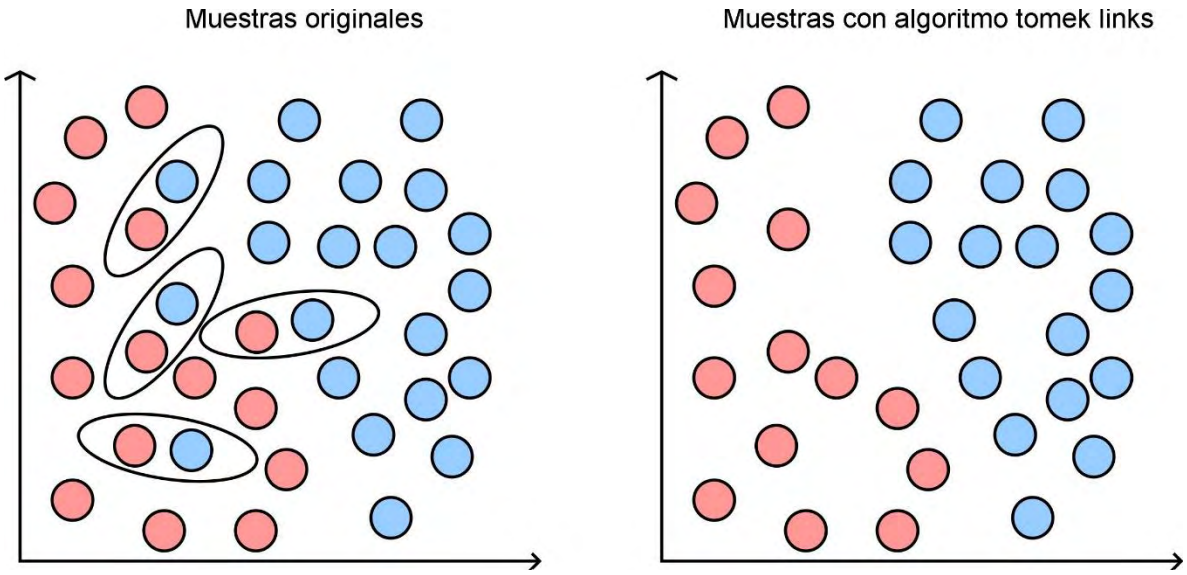


Figura 8. Tomek Links. Elimina muestras de la clase mayoritaria que sean redundantes o estén cerca de las muestras de la clase minoritaria.

Wilson Editing: También conocido como ENN (*Editing Nearest Neighbor*) elimina aquellas instancias donde la mayoría de sus vecinos pertenecen a otra clase.

Entre los inconvenientes del submuestreo, está la pérdida de información que se produce al eliminar instancias de la muestra. Sin embargo, tiene la ventaja de reducir el tiempo de procesado del conjunto de datos [34].

- ***Boosting***

Consiste en asociar pesos a cada instancia que se van modificando en cada iteración del clasificador. Inicialmente todas las instancias tienen el mismo peso y después de cada iteración, en función del error cometido en la clasificación se reajustan los pesos con objeto de reducir dicho error [34]:

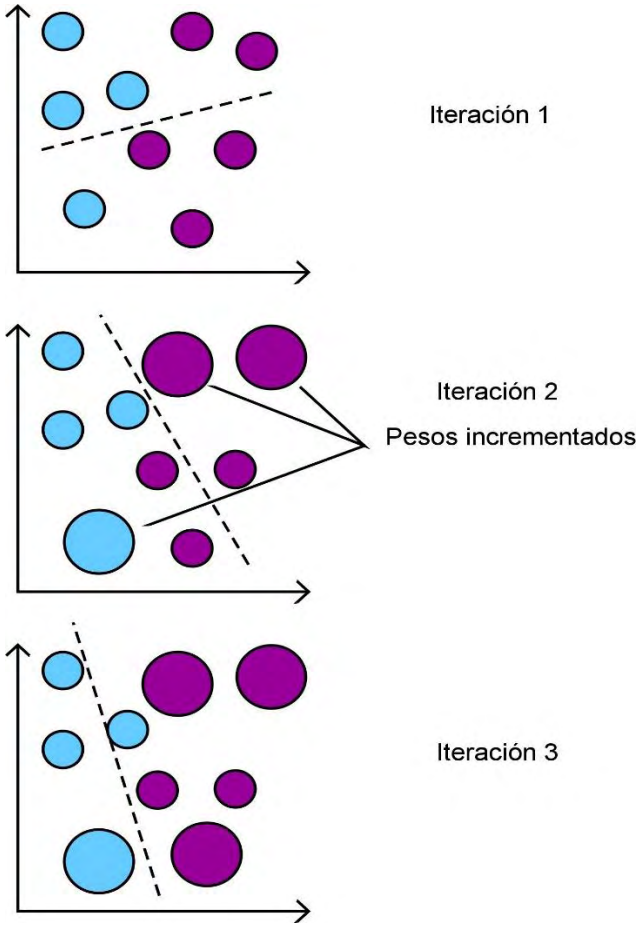


Figura 9. Funcionamiento del boosting. Modifica los pesos de los objetos en cada iteración en función del error cometido en la clasificación.

AdaBoost: Implementa el algoritmo de *Boosting* descrito. En cada iteración AdaBoost genera nuevas instancias utilizando remuestreo.

SMOTEBoost: Es similar a *AdaBoost* pero usa SMOTE en lugar del remuestreo para generar nuevas instancias.

RUSBoost: Aplica *AdaBoost* pero en cada iteración utiliza el submuestreo aleatorio que reducen el tamaño de la muestra de datos y simplifican y aumentan el rendimiento del clasificador.

3.4. ESTIMACIÓN DE DENSIDADES DE PROBABILIDAD

Se basa en un modelo generativo probabilístico para los datos observados. La estimación de densidad se refiere al proceso de estimar la función de densidad subyacente de tal manera que el modelo pueda describir mejor los datos. El modelo aprendido se utiliza para detectar nuevos patrones basados en algunos criterios derivados de medidas estadísticas, como la probabilidad [35].

La disponibilidad requerida de la densidad de probabilidad condicional y las probabilidades a priori se obtienen por medio del conocimiento general del proceso físico y sensorial del sistema en términos de modelos matemáticos. En el aprendizaje supervisado, dos enfoques se discuten usualmente, el aprendizaje paramétrico y el aprendizaje no paramétrico [31].

- **Aprendizaje paramétrico**

En el aprendizaje paramétrico, la suposición base hace referencia a que los únicos factores desconocidos son los parámetros de las densidades de probabilidad involucradas. Por lo tanto, el aprendizaje a partir de muestras se reduce hasta encontrar los valores adecuados de estos parámetros [31].

- **Aprendizaje No paramétrico**

En el aprendizaje no paramétrico, el conocimiento previo acerca de la forma funcional de las distribuciones de probabilidad condicional no está disponible o no se utiliza de forma explícita. Por su parte, la estimación no paramétrica no hace uso de ningún supuesto acerca de la forma de las funciones, por el contrario permite que cada observación contribuya a la construcción de su propio modelo.

3.4.1. Estimadores de Parzen

Es una técnica para la estimación no paramétrica de la densidad, que también se puede utilizar para la clasificación. Normalmente, usa una función *Kernel* y la técnica se aproxima a una distribución de conjunto de entrenamiento dado a través de una combinación lineal de los *Kernels* centrados en los puntos observados.

Las ventanas de Parzen pueden considerarse como una generalización de la técnica de *k* - vecinos más cercanos. En lugar de elegir los *k* vecinos más cercanos de un punto de prueba y etiquetarlo de acuerdo a la mayor ponderación de los votos de sus vecinos, se puede considerar todos los puntos del esquema de

votación y asignar su peso por medio de la función *Kernel*. Con los núcleos de *Gauss*, el peso disminuye exponencialmente con el cuadrado de la distancia. La anchura σ de la gaussiana determina la ponderación relativa de los puntos cercanos y lejanos. Sintonizando este parámetro controla la capacidad de predicción del sistema [36].

4. METODOLOGÍA

La metodología propuesta para el desarrollo del CBR multi-clase se resume en la figura 10 y se compone de las siguientes etapas: Pre-procesamiento que incluye normalización, selección de características y balanceo de datos; recuperación de los casos similares; adaptación, que contiene la clasificación multi-clase y la estimación de probabilidades de pertenencia y, finalmente, la revisión y aprendizaje.

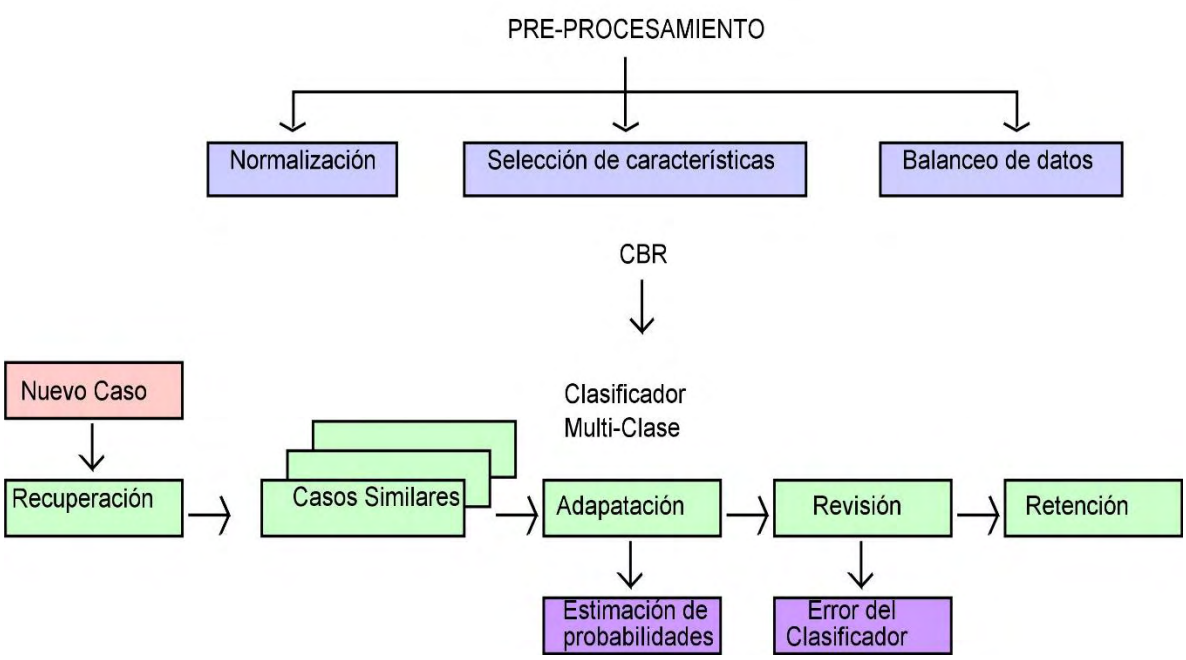


Figura 10. Metodología para el desarrollo de CBR multi-clase. En ella se muestra las etapas que lo conforman iniciando con el pre-procesamiento de los datos. La recuperación de casos similares, adaptación que incluye estimación de probabilidades, la revisión y aprendizaje.

4.1. PRE-PROCESAMIENTO

- **Selección de características**

Cfs-SubsetEval elabora una jerarquización de subconjuntos de atributos de acuerdo a su correlación basada en una función de evaluación heurística. Dicha función de evaluación se basa en el cálculo de la correlación estadística, buscando atributos que están muy poco correlacionados entre ellos, pero tienen una buena correlación con la clase. Las características irrelevantes por

tanto son ignoradas, ya que ellas mantendrán una muy baja o nula correlación con la clase. La información redundante por otra parte será penalizada ya que el atributo redundante tendrá una alta correlación con una o varias de las características restantes. La inclusión de una característica por tanto depende de si esta es capaz de explicar la clase en fragmentos del espacio de instancias que no han sido ya explicadas por otros atributos. La función de evaluación utilizada es la siguiente:

$$Ms = \frac{k\,\overline{ref}}{\sqrt{k+k(k-1)\overline{r_{ff}}}}\,, \qquad (1)$$

donde Ms es el mérito heurístico del subconjunto S conteniendo k características, $\overline{ref}$ es el valor de la correlación media entre la clase y la característica $f(f \subset S)$ y $\overline{r_{ff}}$ es la mejor correlación entre dos características del conjunto S. El método CFS asume que las características son condicionalmente independientes dada la clase, esto puede ser una simplificación aceptable en algunos casos, pero si existe una fuerte interacción entre distintos atributos, entonces CFS no puede garantizar que los atributos seleccionados sean relevantes [37]. En este trabajo se utiliza como método de búsqueda el algoritmo de *BestFirst.*

BestFirst

Es una búsqueda en profundidad, pero aplicando vuelta atrás (en inglés *backtraking*) hasta un límite de retrocesos. Básicamente la búsqueda se desarrolla usando un árbol y consiste en ir eliminando atributos hasta llegar a un cierto número de atributos (predeterminados por el usuario). El subconjunto de atributos es evaluado usando una métrica monotónica y el valor obtenido es guardado como una cota. A continuación, se procede a quitar otros atributos del conjunto original, siguiendo un esquema ordenado de eliminación de atributos (esquema de enumeración); cada subconjunto así obtenido es evaluado. Si algún subconjunto obtiene una evaluación igual o peor que la cota, se detiene la exploración de esa rama (es decir, se realiza una poda), puesto que continuar la exploración es inútil pues no conduce a una mejor solución que la que ya se tiene actualmente. Por otro lado, si todos los subconjuntos evaluados resultan mejor que la cota, se actualiza la cota con el nuevo valor, y se repite el procedimiento hasta que no haya más ramas que explorar. Con este procedimiento, se logra ahorros en tiempo de procesamiento, y al mismo tiempo garantizando que la solución es óptima (usando la métrica correcta) [37].

- **Balanceo de datos**

Se utiliza el algoritmo **SMOTE,** es una técnica de sobremuestreo que genera instancias “sintéticas” o artificiales para equilibrar la muestra de datos basado en

la regla del vecino más cercano. La generación se realiza extrapolando nuevas instancias en lugar de duplicarlas como hace el algoritmo de remuestreo [38].

La generación se realiza extrapolando nuevas instancias en lugar de duplicarlas como hace el algoritmo de remuestreo. Para cada una de las instancias minoritarias se buscan las instancias minoritarias vecinas (más cercanas) y se crean N instancias entre la línea que une la instancia original y cada una de las vecinas. El valor de N depende del tamaño de sobremuestreo deseado. Para un caso del 200% por cada instancia de la clase minoritaria deben crearse dos nuevas instancias genéricas [34]. El algoritmo de expica en la figura 11.

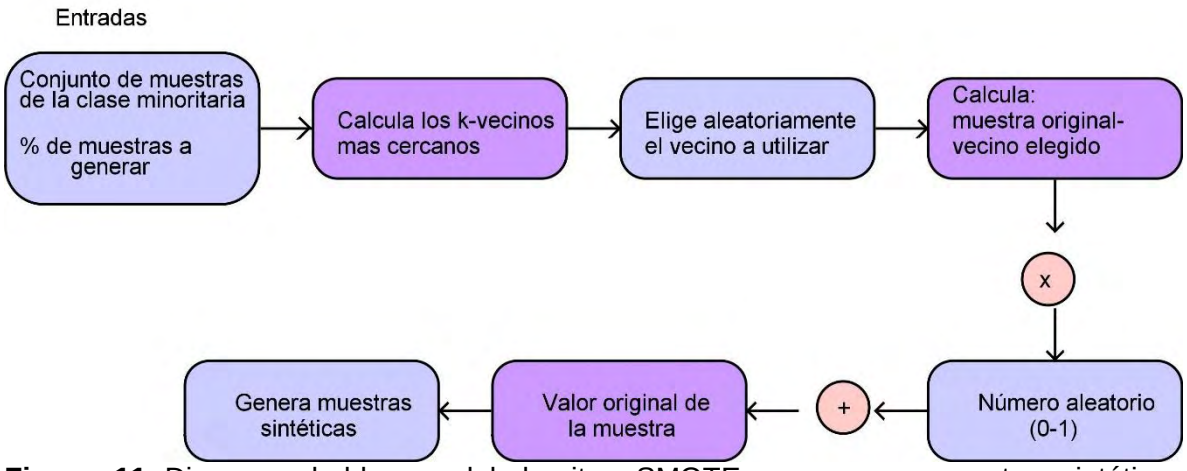


Figura 11. Diagrama de bloques del algoritmo SMOTE para generar muestras sintéticas de la clase minoritaria.

4.2. RECUPERACIÓN DE CASOS SIMILARES

En esta etapa se utiliza el conocimiento adquirido de la base de casos para realizar la recuperación de los casos similares a la situación actual y así tener una base para poder resolverla o clasificarla.

La técnica utilizada es los $k-$ vecinos más cercanos, donde los ejemplos de entrenamiento son vectores en un espacio característico multidimensional, cada ejemplo está descrito en términos de d atributos considerando q clases para la clasificación. Los valores de los atributos del $i-esimo$ ejemplo $(donde\ 1 \leq i \leq N)$ se representan por el vector d- dimensional.

$$x_i = (x_1, x_2, \ldots, x_d,) \epsilon\, X.$$

El espacio es particionado en regiones por localizaciones y etiquetas de los ejemplos de entrenamiento. Un punto en el espacio es asignado a la clase C si

esta es la clase más frecuente entre los k ejemplos de entrenamiento más cercano. Generalmente se usa la distancia euclidiana.

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{d} (x_{ri} - x_{rj})^2} \quad (2)$$

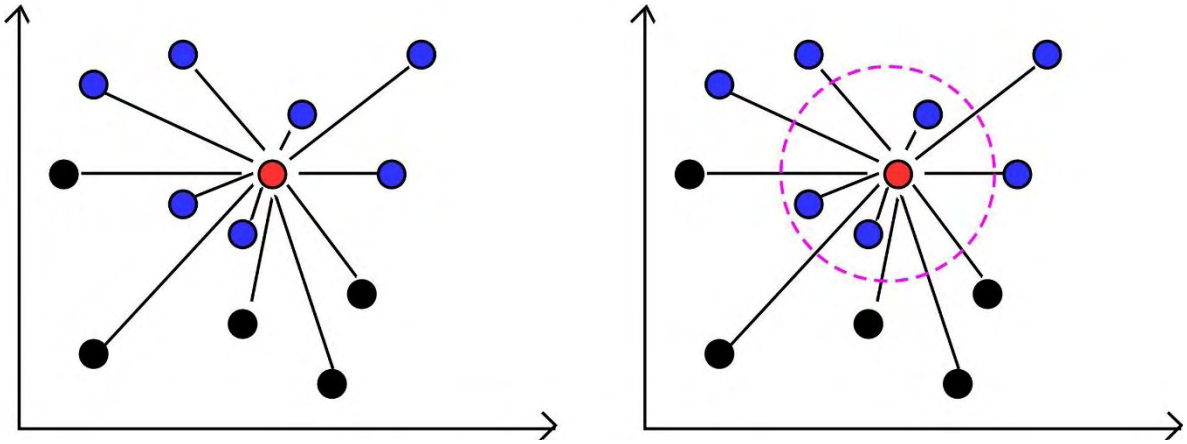


Figura 12. Método de los k- Vecinos más cercanos. Se calcula los vecinos cercanos de la muestra por medio de la distancia euclidiana. El punto rojo representa la nueva muestra y los conjuntos de cada clase están representados por los puntos azules y negros.

4.3. ADAPTACIÓN Y ESTIMACIÓN DE PROBABILIDADES

4.3.1. Máquinas de Soporte Vectorial

La idea es seleccionar un hiperplano de separación que equidista de los ejemplos más cercanos de cada clase para, de esta forma, conseguir lo que se denomina un margen máximo a cada lado del hiperplano. Además, a la hora de definir el hiperplano, sólo se consideran los ejemplos de entrenamiento de cada clase que caen justo en la frontera de dichos márgenes. Estos ejemplos reciben el nombre de vectores soporte [39].

Para clasificación binaria de ejemplos linealmente separables, dado un conjunto de ejemplos $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$, donde $x_i \in \mathbb{R}^d$ e $y_i \in \{+1, -1\}$ se puede definir un hiperplano de separación como una función lineal que es capaz de separar dicho conjunto sin error. Como se observa en la figura 13.

$$D(x) = (w_1 x_1 + \cdots + w_d x_d) + b = w^T x + b \ , \quad (3)$$

donde w y b son coeficientes reales.

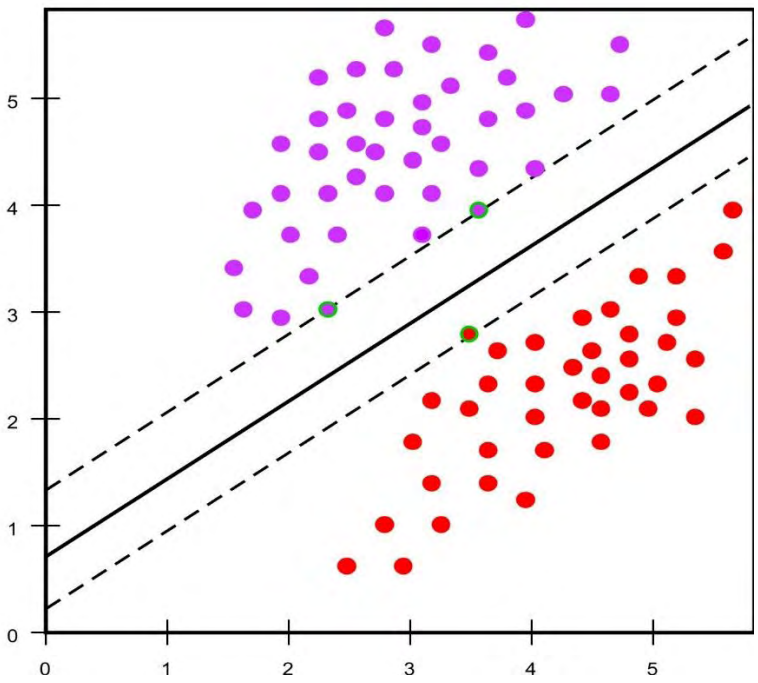


Figura 13. Hiperplano de separación en un espacio bidimensional de un conjunto de ejemplos separables en dos clases de entre los infinitos posibles.

El hiperplano de separación cumplirá la siguiente restricción para todo x_i del conjunto de ejemplos:

$$y_i(< w, y_i > + b) \geq 0, \qquad i \in \{1, \ldots, n\}.$$

El hiperplano que permite separar los ejemplos no es único, es decir, existen infinitos hiperplanos separables, representados por todos aquellos hiperplanos que son capaces de cumplir las restricciones impuestas. La definición de hiperplano de separación óptimo es que éste equidista del ejemplo más cercano de cada clase.

Se define el concepto de margen de un hiperplano de separación, denotado por τ, como la mínima distancia entre dicho hiperplano y el ejemplo más cercano de cualquiera de las dos clases figura 14.

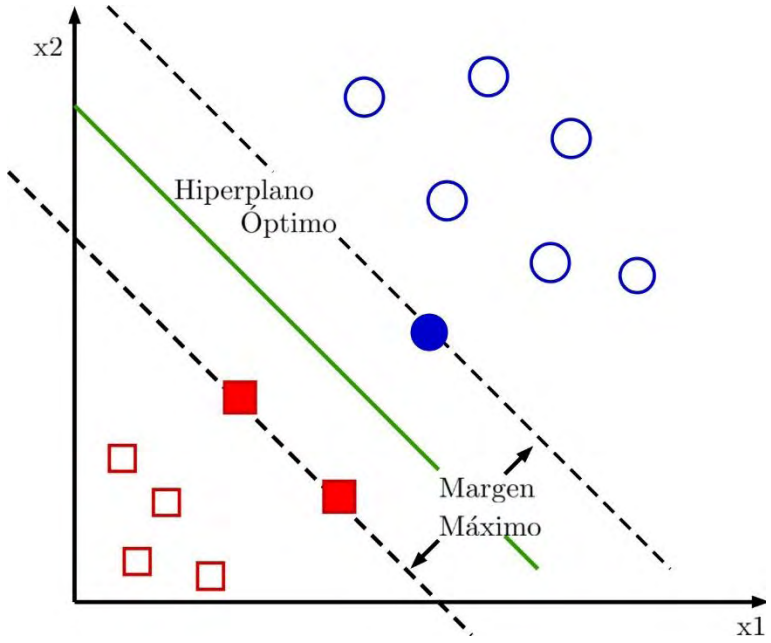


Figura 14. Margen de un hiperplano de separación. Hiperplano de separación óptimo y su margen asociado (máximo).

Por geometría, se sabe que la distancia entre un hiperplano de separación $D(x)$ y un ejemplo x' viene dada por:

$$\frac{|D(x')|}{\|w\|}\ , \tag{4}$$

donde $|.|$ el operador valor absoluto, $\|.\|$ el operador norma de un vector y w el vector que, junto con el parámetro b, define el hiperplano $D(x)$. Todos los ejemplos de entrenamiento cumplirán que:

$$\frac{y_i D(x_i)}{\|w\|} \geq \tau, \quad i \in \{1, \dots, n\}, \tag{5}$$

de (5) se deduce que encontrar el hiperplano óptimo es equivalente a encontrar el valor de w que maximiza el margen. Para limitar el número de soluciones a una sola, y teniendo en cuenta que (5) se puede expresar también como:

$$y_i D(x_i) \geq \tau \|w\|, \quad i \in \{1, \dots, n\}. \tag{6}$$

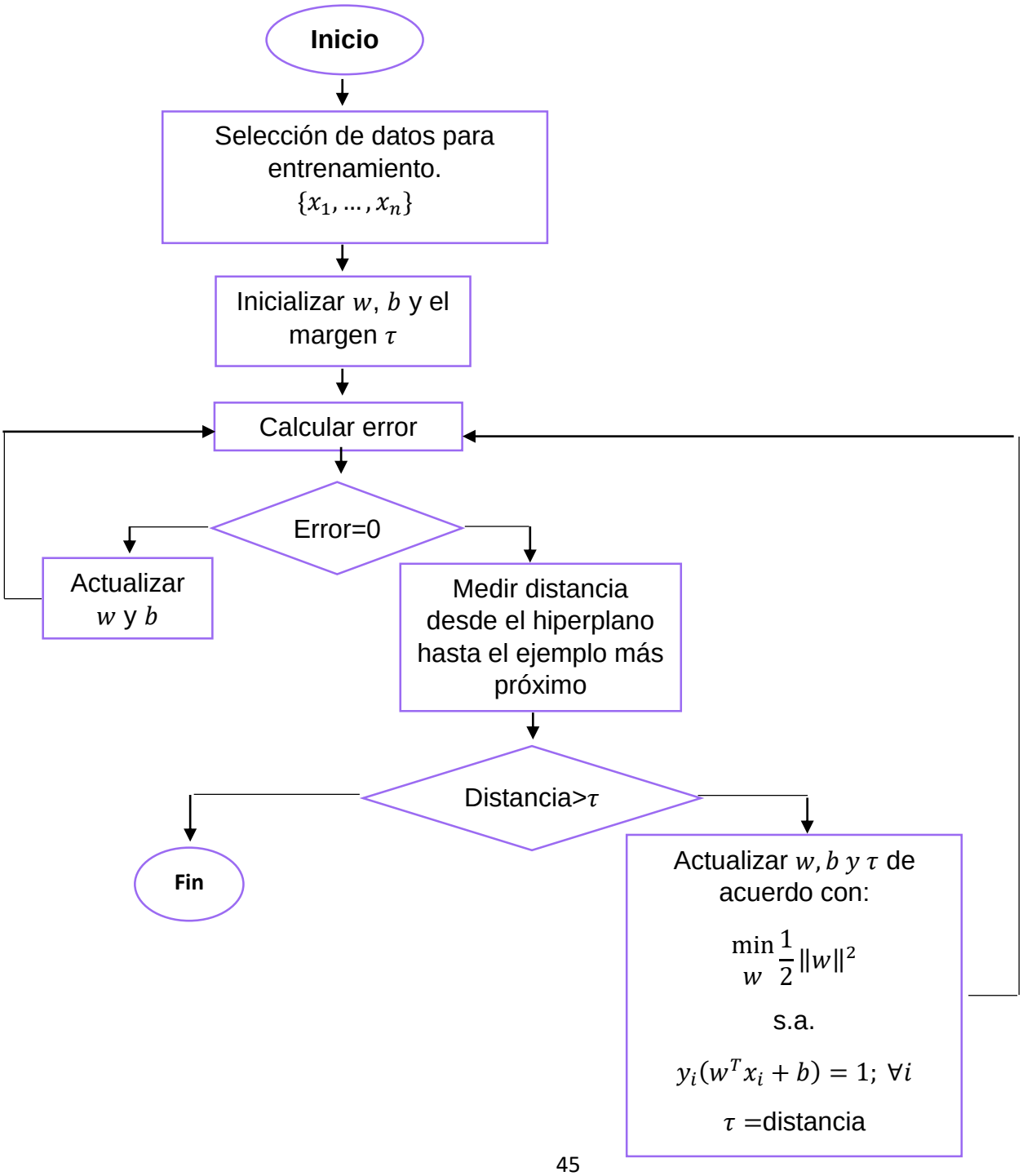
La escala del producto de τ y la norma de w se fija, de forma arbitraria, a la unidad, es decir:

$$\|w\| = 1.$$

Llegando a la conclusión final de que aumentar el margen es equivalente a disminuir la norma de w, ya que la expresión anterior se puede expresar como:

$$\tau = \frac{1}{\|w\|}.$$

A continuación, se explica el procedimiento anterior mediante un diagrama de flujo:



4.3.2. Redes Neuronales artificiales

Las neuronas son células muy especializadas con una morfología característica y unas propiedades funcionales que les permite la recepción, generación y propagación de impulsos nerviosos. Además poseen dispositivos específicos de contacto intercelular como las sinapsis, para la transferencia interneuronal de señales nerviosas en los circuitos neuronales [40].

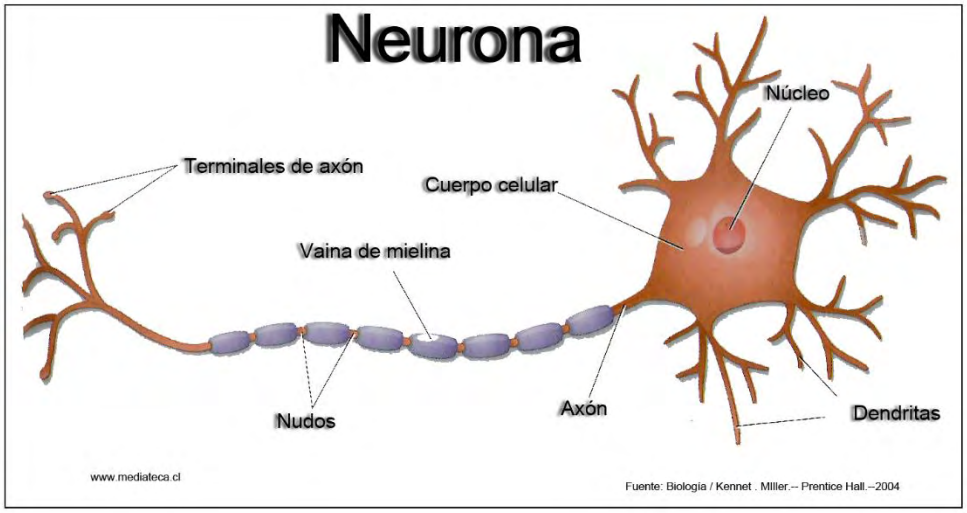


Figura 15. La neurona es la unidad estructural y funcional del sistema nervioso. En su estructura se puede distinguir: Núcleo, dendritas, axón y terminales de axón.
Fuente: [41]

En las neuronas, la información se transmite mediante cambios de polaridad en las membranas de las células debido a la presencia de neurotransmisores que alteran la concentración iónica del interior celular.

En el interior de la neurona existen proteínas e iones con carga negativa. Esta diferencia de concentración de iones produce también una diferencia de potencial entre el exterior de la membrana y el interior celular. El valor que se alcanza es de unos -70 milivoltios (negativo el interior con respecto al valor de cargas positivas del exterior).

Esta variación entre el exterior y el interior se alcanza por el funcionamiento de la bomba de sodio/potasio (Na+/K+). Durante este proceso se expulsa tres iones de sodio que se encontraban en el interior de la neurona e introduce dos iones de potasio que se encontraban en el exterior. Los iones sodio no pueden volver a entrar en la neurona, debido a que la membrana es impermeable al sodio. Por ello, la concentración de iones sodio en el exterior es elevada. Además, se pierden 3 cargas positivas cada vez que funciona la bomba de Na+/K+, aunque entren dos

cargas de potasio. Esto hace que en el exterior haya más cargas positivas que en el interior, creando una diferencia de potencial [42]. Figura 16.

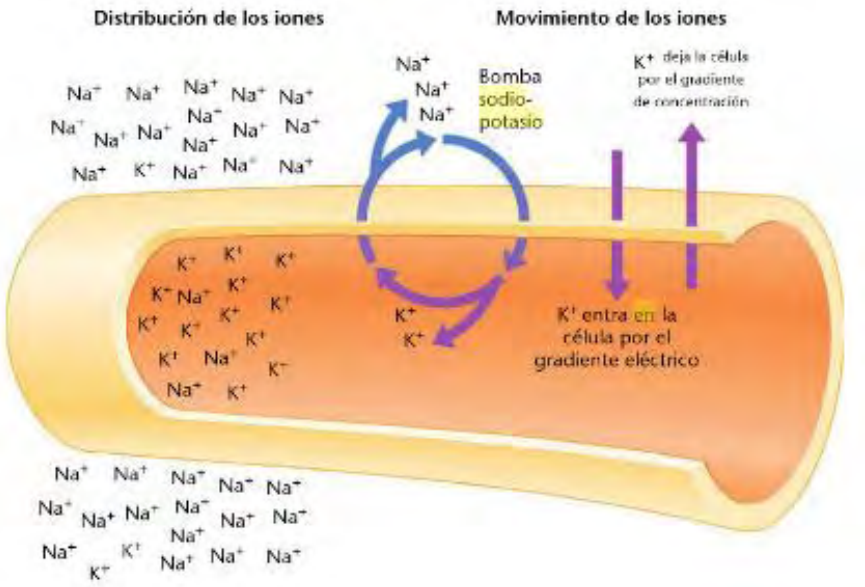


Figura 16. Gradiente de sodio potasio de una membrana en reposo: El ion sodio (Na) está más concentrado fuera de la neurona: El ion potasio (k) está más concentrado en su interior.
Fuente: [43]

La red neuronal es un procesador paralelo distribuido constituido por unidades simples de procesamiento que tienen una disposición natural para almacenamiento de conocimiento experimental. Imitan al cerebro en dos aspectos:

1. El conocimiento es adquirido por la red desde su ambiente a través de un proceso de aprendizaje.

2. La fuerza de conexión entre las neuronas (conocido como los pesos sinápticos) son usados para almacenar el conocimiento adquirido [44].

Las redes neuronales se caracterizan principalmente por:

- Tener una inclinación natural a adquirir el conocimiento a través de la experiencia, el cual es almacenado, al igual que en el cerebro.
- Poseen un alto nivel de tolerancia a fallas, es decir pueden sufrir un daño considerable y continuar teniendo un buen comportamiento, al igual como ocurre en los sistemas biológicos.
- Tener un comportamiento altamente no lineal, lo que les permite procesar información procedente de otros fenómenos no-lineales [45].

Las neuronas biológicas interactúan dinámicamente entre ellas y cambian sus relaciones en el tiempo. Las interacciones son bastante complejas y dependen de

la estructura de las neuronas. En una estructura simple, se puede observar la interconexión tipo "*feedforward*". La información de una célula se pasa a otra y puede ser más fuerte que otras conexiones [44].

El modelo de una red neuronal se indica en la figura 6. En este modelo se tiene que:

- $\boldsymbol{x} = [x_1\ x_2\ \ldots x_d]^T$ Vector de entradas
- $\boldsymbol{w} = [w_1\ w_2\ \ldots w_d]^T$ Vector de pesos
- Un factor de desplazamiento (bias) b
- Una función de activación $f(\bar{x})$
- Salida y

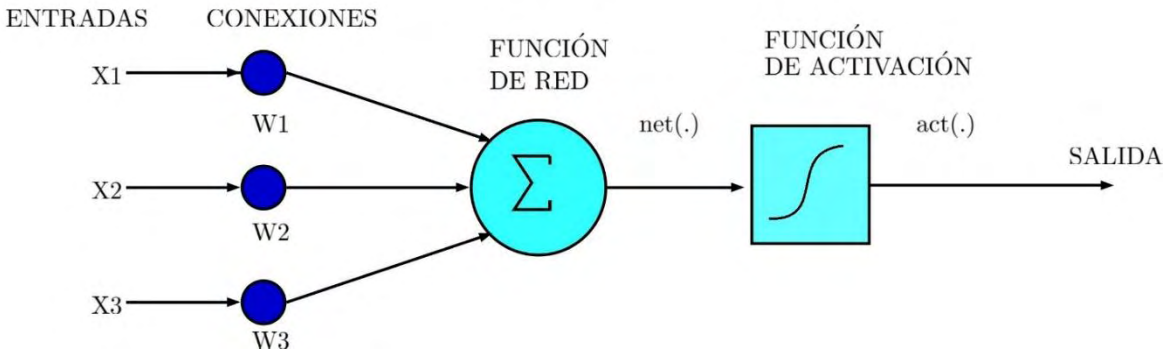


Figura 17. Modelo de una red neuronal. Esta red neuronal de una sola capa cuenta con 3 entradas, una función de red y una función de activación.

Para este modelo se tiene que:

- El potencial de activación está dado por:

$$\bar{x} = \sum_{i=1}^{n} w_i x_i + b = x^T w + b \qquad (7)$$

- La función de activación determina la señal de salida

$$y = f(\bar{x}) = f(\sum_{i=1}^{n} w_i x_i + b) = f(x^T w + b) \qquad (8)$$

- La salida y generalmente se normaliza en un rango $y \in [0,1]$ o $y \in [-1,1]$. Las funciones de activación pueden ser la función de umbral, función sigmoidal, función tangente hiperbólica, lineal, entre otras. La función de activación utilizada en la red neuronal se describe a continuación:

Función sigmoidal

Es una función matemática que aparece en diversos modelos de crecimiento de poblaciones, propagación de enfermedades epidémicas y difusión en redes sociales. Dicha función constituye un refinamiento del modelo exponencial para el crecimiento de una magnitud. Modela la función sigmoidea de crecimiento de un conjunto *P*.

El estudio inicial de crecimiento es aproximadamente exponencial; al cabo de un tiempo, aparece la competencia entre algunos miembros de *P* por algún recurso crítico y la tasa de crecimiento disminuye; finalmente, en la madurez, el crecimiento se detiene. Figura 18

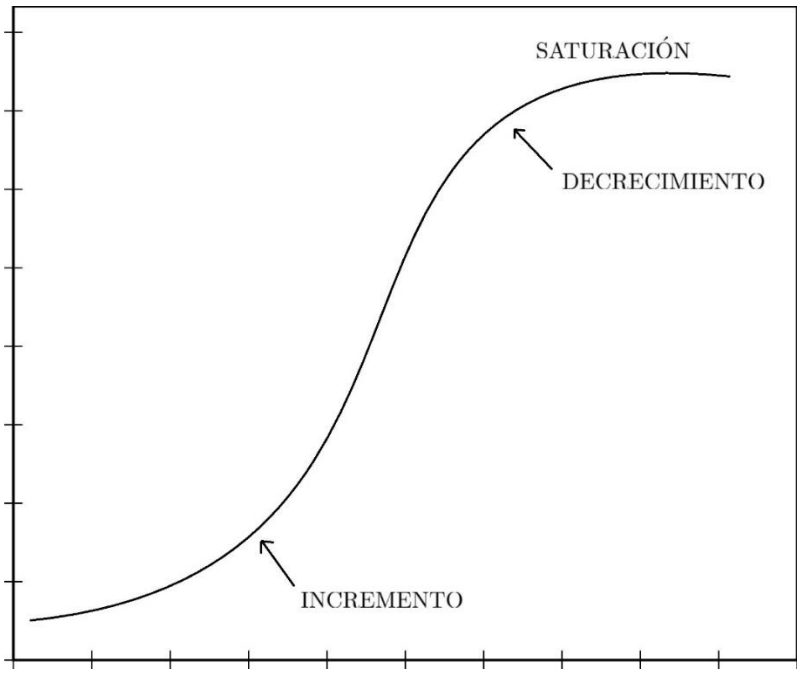


Figura 18. Función sigmoidal. La curva varía en el tiempo indicando un instante en el que presenta un crecimiento seguido de un leve decrecimiento y finalmente la saturación.

La función logística simple se define mediante la expresión matemática:

$$f(\bar{x}) = \frac{1}{1+exp(-\bar{x})} \quad . \qquad (9)$$

Las diferentes clases de ANN se distinguen entre sí por los siguientes elementos:

- Las neuronas o nodos que constituyen elementos básicos de procesamiento.
- La arquitectura de la red descrita por las conexiones ponderadas entre los nodos.

- El algoritmo de entrenamiento, usado para encontrar los parámetros de la red.

Redes Neuronales hacia adelante (*Feedforward*)

En este tipo de redes se empieza con un vector de entradas el cual es equivalente en magnitud al número de neuronas de la primera capa de la red, las cuales procesan dicho vector elemento por elemento en paralelo. La información, modificada por los factores multiplicativos de los pesos en cada neurona, e transmitida hacia delante por la red pasando por las capas ocultas (si hay) para finalmente ser procesada por la capa de salida.

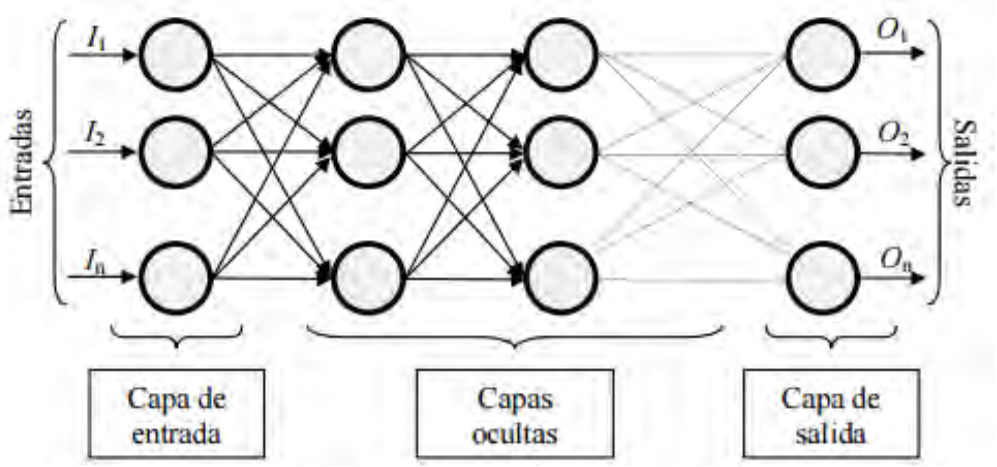


Figura 19. Red neuronal Los datos ingresan por medio de la “capa de entrada”, pasan a través de la “capa oculta” y salen por la “capa de salida”.
Fuente: [46]

Redes de propagación hacia atrás (B*ackpropagation*)

El nombre de *backpropagation* resulta de la forma en que el error es propagado hacia atrás a través de la red neuronal, en otras palabras, el error se propaga hacia atrás desde la capa de salida. Esto permite que los pesos sobre las conexiones de las neuronas ubicadas en las capas ocultas cambien durante el entrenamiento. El cambio de los pesos en las conexiones de las neuronas además de influir sobre la entrada global, influye en la activación y por consiguiente en la salida de una neurona. Por lo tanto, es de gran utilidad considerar las variaciones de la función activación al modificarse el valor de los pesos. Esto se llama sensibilidad de la función activación, de acuerdo al cambio en los pesos [46].

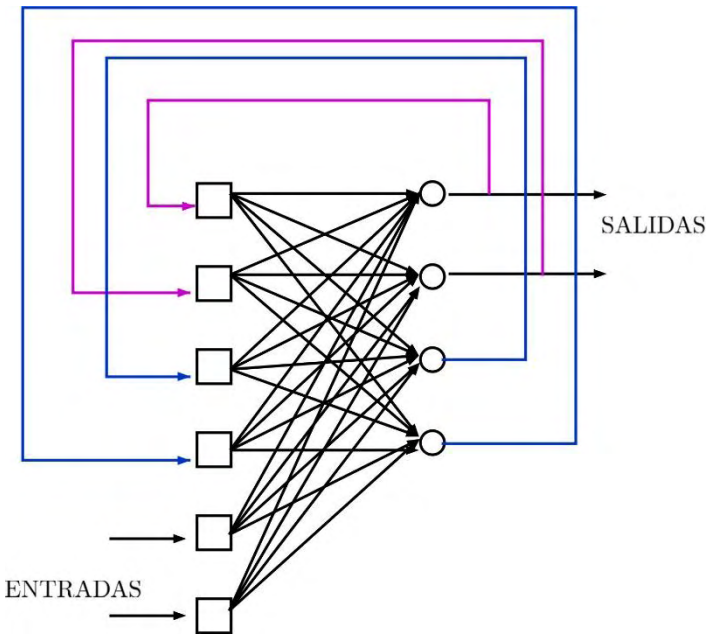


Figura 20. Red neuronal de propagación hacia atrás (backpropagation).

4.3.3. K-vecinos más cercanos

El método de los k-vecinos o k-NN es un método supervisado, cuyo argumento principal es la distancia entre instancias. El método básicamente consiste en comparar la nueva instancia a clasificar con los datos k más cercanos conocidos, y dependiendo del parecido entre los atributos el nuevo caso se ubicará en la clase que más se acerque al valor de sus propios atributos [47].

- **Métodos basados en vecindad**

Los métodos basados en vecindad son fundamentalmente dependientes de la distancia y en consecuencia poseen características propias de ésta como la cercanía, la lejanía y la magnitud de longitud, entre otras. Los métodos basados en vecindad, además de servir para tareas de clasificación, también se usan para agrupación de datos. Existen dos grupos de métodos de vecindad, según la forma en que se realiza el aprendizaje. El grupo de los métodos retardados y los no retardados. En los métodos retardados como k-NN, cada vez que se va a clasificar un dato, en la fase de entrenamiento, se elabora un modelo específico para cada nuevo dato, y una vez que éste se clasifica sirve como un nuevo caso de entrenamiento para clasificar una nueva instancia. En los métodos no retardados se generaliza un solo modelo (también a partir de casos conocidos) para todos los nuevos datos que se desean clasificar, y éstos únicamente son tomados en cuenta

como datos de entrenamiento cuando se vuelve a construir un nuevo modelo general [47].

Las métricas, alternativas, usadas para medir la distancia son:

- Distancia de Manhattan.
- Distancia de Chebychev.
- Distancia del coseno.
- Distancia de Mahalanobis.
- Distancia usando la función delta.
- Distancia entre dos conjuntos.

- **Distancia Euclidiana**

Se trata de una función no negativa usada en diversos contextos para calcular la distancia entre dos puntos, primero en el plano y luego en el espacio. También sirve para definir la distancia entre dos puntos en otros tipos de espacios de tres o más dimensiones. Y para hallar la longitud de un segmento definido por dos puntos de una recta, del plano o de espacios de mayor dimensión [47]. La distancia euclidiana entre dos puntos se define en la ecuación:

$$(\boldsymbol{X}_1, \boldsymbol{X}_2) = \sqrt{\sum_{i=1}^{d} (x_{1i} - x_{2i})^2}. \quad (10)$$

4.3.4. Estimación de probabilidades

El objetivo de la estimación de Parzen es obtener estimaciones de densidades de probabilidad condicional $p(x)$, Teniendo un conjunto $D = \{x_1, x_2 \dots x_N\}$ de n muestras independientes e idénticamente distribuidas. La estimación $\hat{p}(x)$ de la ventana de Parzen $p(x)$ basada en las n muestras en D está dada por:

$$\hat{p}(x) = \frac{1}{n} \sum_{i=1}^{n} \delta_n (x - x_i), \quad (11)$$

donde $\delta_n(\cdot)$ es la función Kernel con soporte localizado y su forma exacta depende de n.

Se opta en utilizar las funciones de núcleo gaussiano por dos razones. En primer lugar, la función gaussiana es suave y, por tanto, la función de densidad estimada $\hat{p}(x)$ también varía suavemente. En segundo lugar, si asumimos una forma especial de la familia gaussiana en la cual la función es radialmente simétrica, la función puede ser completamente especificada por un parámetro de varianza solamente. Así $\hat{p}(x)$ puede expresarse como una mezcla de núcleos Gaussianos radialmente simétricos con una varianza común σ^2.

$$\hat{p}(x) = \frac{1}{n(2\pi)^{2/d}\sigma^d} \sum_{i=1}^{N} exp\left\{-\frac{\|x - x_i\|^2}{2\sigma^2}\right\}, \quad (12)$$

donde d es la dimensión del espacio de características.

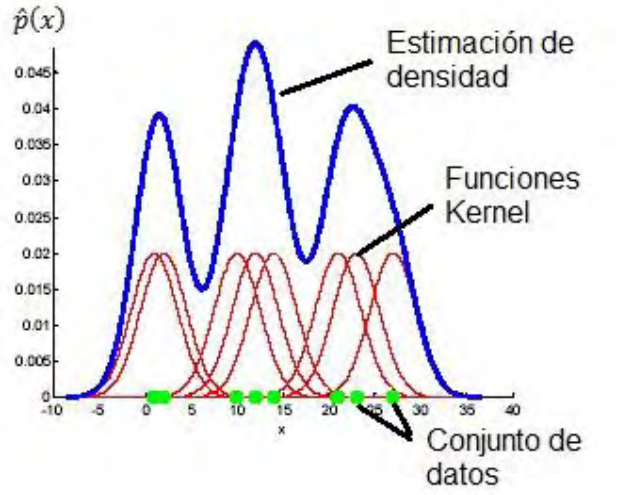


Figura 21. La estimación ventana Parzen puede ser considerada como una suma de gaussianas centradas en los puntos de datos. La función de *Kernel* determina la forma de las gaussianas. El parámetro σ, también llamado el parámetro de suavizado o ancho de banda, determina su anchura.
Fuente: [48]

4.4. REVISIÓN

Para la etapa de revisión se utiliza el **error cuadrático medio** (MSE- *Mean Squared Error*), que mide el promedio de los errores al cuadrado, es decir, la diferencia entre el estimador y lo que se estima. El MSE equivale a la suma de la varianza y la desviación al cuadrado del estimador. Un estimador es usado para deducir el valor de un parámetro desconocido en un modelo estadístico. La desviación es la diferencia entre el valor esperado del estimador y el valor real del parámetro que se quiere estimar.

Al calcular la raíz cuadrada del MSE se obtiene la raíz cuadrada de la desviación media, que es una buena medida de precisión y también es conocida como la media cuadrática.

Si $\hat{Y}$ es un vector de N predicciones y Y es el vector de los verdaderos valores, entonces el (estimado) MSE del predictor es:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{Y}_i - Y_i\right)^2. \qquad (13)$$

Se considera el cuadrado de ese error para evitar que las diferencias positivas se compensen con las negativas.

5. MARCO EXPERIMENTAL

Para los experimentos, se utilizan las bases de datos disponibles públicamente de la *UCI Machine Learning Repository* de la Universidad de California [49]. Se evalúa el desempeño de los clasificadores multi-clase con métricas de desempeño como matriz de confusión y curvas ROC con el objetivo de identificar el mejor de ellos e integrarlo a la etapa de adaptación del CBR.

5.1. BASES DE DATOS

La primera base de datos, es la de cleveland, contiene 303 instancias. Se refiere a la presencia de enfermedad cardíaca en el paciente. Es un valor entero de 0 (sin presencia) a 4. Los experimentos con la base de datos de Cleveland se han concentrado en intentar distinguir la presencia (valores 1, 2, 3, 4) de la ausencia (valor 0). Esta base de datos consta de 13 atributos que se explican en la tabla 1.

Tabla 1. Información de atributos de la base de datos cleveland

1	Age	Edad
2	Sex	Sexo
3	CP	Tipo de dolor en el pecho (angina típica = 1, angina atípica = 2, dolor no angina = 3, asintomática = 4)
4	Trestbps	Presión arterial en reposo
5	CHOL	Colesterol
6	FBS	Azúcar en sangre en ayunas (True = 1; Falso = 0)
7	RestECG	resultados electrocardiográficos en reposo (Normal = 1; Anormalidad de la onda ST-T (inversión de la onda T y / o elevación o depresión del ST superior a> 0,05 mV) = 2; Mostrar hipertrofia ventricular izquierda probable o definida por los criterios de estrés = 3),
8	Thalach	Frecuencia cardiaca máxima
9	Exang	Angina inducida por el ejercicio (Sí = 1, No = 0)
10	Oldpeak	Pico antiguo (depresión del segmento ST inducida por el ejercicio en relación a descansar)
11	Slope	Pendiente del segmento ST del ejercicio de pico (Ascendiente = 1, plana = 2, Descendiente = 3)
12	CA	Número de vasos principales (0-3) coloreados por fluoroscopia
13	Thal	(Normal = 3, Detección fija = 6, Detección reversible = 7)

La segunda base de datos, es de cadiotocografía, este conjunto de datos consta de las mediciones de la frecuencia cardíaca fetal (FCF) y la contracción uterina

(CU), cuenta con características del cardiotocograma clasificadas por obstetras expertos.Contiene 2126 cardiotocogramas fetales pertenecientes a diferentes clases. La clasificación de esta base de datos se realizó con respecto a un patrón morfológico (A, B, C. …) y a un estado fetal (N, S, P). Por lo tanto, el conjunto de datos se puede utilizar ya sea para los experimentos de 10 o 3 clases. Este conjunto de datos consta de 23 atributos que indican en la tabla 2.

Tabla 2. Información de atributos de la base de datos cardiotocografía

1	LB – FCF	línea de base (latidos por minuto)
2	AC	Número de aceleraciones por segundo
3	FM	Número de movimientos fetales por segundo
4	UC	Número de contracciones uterinas por segundo
5	DL	Número de desaceleraciones de luz por segundo
6	DS	Número de deceleraciones graves por segundo
7	DP	Número de desaceleraciones prolongadas por segundo
8	ASTV	Porcentaje de tiempo con variabilidad anormal a corto plazo
9	MSTV	Valor medio de la variabilidad a corto plazo
10	ALTV	Porcentaje de tiempo con variabilidad anormal a largo plazo
11	MLTV	Valor medio de la variabilidad a largo plazo
12	Width	Ancho del histograma de FCF
13	Min	Mínimo del histograma de FCF
14	Max	Máximo de la FCF histograma
15	Nmax	Número de picos en el histograma
16	Nzeros	Número de ceros en el histograma
17	Mode	Moda del histograma
18	Mean	Media del histograma
19	Median	Mediana del histograma
20	Variancia	Varianza del histograma
21	Tendency	Tendencia del histograma
22	Clase FHR	Código de clase de patrón (1 a 10)
23	NSP	Código de la clase de estado fetal (Normal = 1, Sospechoso = 2, Patológico = 3)

5.2. ERROR DE LOS CLASIFICADORES

Por medio del MSE es posible hacer estimaciones de rendimiento de los clasificadores entrenados, utilizando un conjunto de datos de prueba que debe contener muestras de todas las clases de la base de casos. MSE es usado para determinar la medida cuando el clasificador no se ajusta a la información, o si eliminando ciertos términos es posible mejorar el rendimiento del mismo.

El MSE proporciona una forma para elegir el mejor clasificador. Un MSE mínimo a menudo, pero no siempre, indica una variación mínima, y por lo tanto indica una buena estimación en el proceso de clasificación.

5.3. MEDIDAS DE DESEMPEÑO

Cada nuevo caso puede ser dividido en dos grupos: Uno conformado por los casos pertenecientes a la clase de interés (CI) y otro conformado por los casos diferentes de la clase de interés (NCI). De esta manera se utilizan las siguientes medidas de desempeño: sensibilidad (Se), especificidad (Sp) y porcentaje de clasificación (CP).

$$Se = \frac{VP}{VP+FN} * 100 \quad (14)$$

$$Sp = \frac{VN}{VN+FP} * 100 \quad (15)$$

$$CP = \frac{VN+VP}{VN+VP+FN+FP} * 100 \quad (16)$$

Donde:

- VP Son los verdaderos positivos o casos de la clase de interés clasificados correctamente.
- VN Son los verdaderos negativos o casos diferentes de la clase de interés clasificados correctamente.
- FP Son los falsos positivos o casos diferentes de la clase de interés clasificados como casos de la clase de interés.
- FN Son los falsos negativos o casos de la clase de interés clasificados como casos diferentes de la clase de interés.

La sensibilidad y especificidad miden la proporción de casos NCI y CI que son clasificados correctamente respectivamente. Estas medidas se usan para medir el desempeño del sistema, pero no tienen implicación en la sintonización de los parámetros del proceso de clasificación.

5.4. MATRÍZ DE CONFUSIÓN

En el campo de la inteligencia artificial una matriz de confusión es una herramienta que permite la visualización del desempeño de un algoritmo que se emplea en aprendizaje supervisado, es decir la exactitud de una clasificación. También es conocida como matriz de error o de contingencia. Esta es una matriz cuadrada de $n\,x\,n$, donde n representa el número de clases. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. Uno de los beneficios de las matrices de confusión es que facilitan ver si el sistema está confundiendo dos clases.

En el caso bi-clase la matriz se representa como la figura 22.

		Clasificación	
		Verdadero	**Negativo**
Clase Real	**Verdadero**	Verdaderos positivos	Falsos positivos
	Negativo	Verdaderos negativos	Falsos negativos

Figura 22. Matriz de confusión para el caso bi-clase. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real.

Para el caso multi-clase la matriz de confusión se explica en la figura 23. Donde:

VP son los casos de la clase de interés clasificados correctamente, es decir los que se ubican en la diagonal.

FN corresponden a la suma de valores de la fila de la clase de interés (Excluyendo VP).

FP corresponden a la suma de valores de la columna de la clase de interés (Excluyendo VP).

VN corresponden a la suma de valores de todas las filas y columnas (Excluyendo la fila y columna de la clase de interés).

		Clasificación			
		Clase 1	**Clase 2**	**Clase 3**	**Clase 4**
Clase Real	**Clase 1**	$\boldsymbol{VP_1}$	E_{12}	E_{13}	E_{14}
	Clase 2	E_{21}	$\boldsymbol{VP_2}$	E_{23}	E_{24}
	Clase 3	E_{31}	E_{32}	$\boldsymbol{VP_3}$	E_{34}
	Clase 4	E_{41}	E_{42}	E_{43}	$\boldsymbol{VP_4}$

Figura 23. Matriz de confusión para el caso multi-clase.

5.5. CURVAS ROC (*Receiver-Operating Characteristic)*

La característica operativa del receptor ROC es una técnica de visualización y selección de clasificadores basado en su desempeño. Presenta la sensibilidad de

una prueba diagnóstica que produce resultados continuos en función de los falsos positivos (complementario de la especificidad), para distintos puntos de corte. Otra interpretación de ésta curva es la representación de la razón o ratio de verdaderos positivos (VPR) frente a la razón o ratio de falsos positivos (FPR).

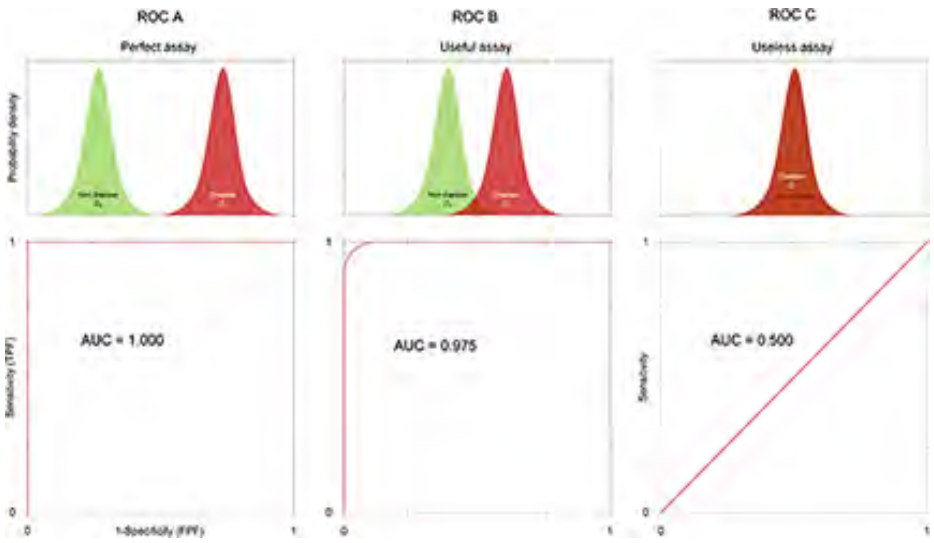


Figura 24. Tipos de Curvas ROC. Representa una herramienta para seleccionar los modelos posiblemente óptimos de acuerdo a los valores de sensibilidad y especificidad.
Fuente: [50]

6. RESULTADOS

- **Error de los clasificadores**

Los resultados obtenidos del rendimiento de los clasificadores para la base de datos Cleveland se muestran en las tablas 3 y 4. En la primera, se realiza las pruebas sin la etapa de pre-procesamiento de datos. En ella se observa que la tarea de clasificación es un desafío ya que el rendimiento es bajo para todos los clasificadores. En la tabla 4, se evalúa la clasificación con la etapa de pre-procesamiento. Se observa que el desempeño de los clasificadores Parzen y *k*-NN mejora significantemente. Sin embargo, los clasificadores SVM y ANN mejoran su desempeño pero no es relevante. Estos resultados se muestran en la figura 25.

Tabla 3. Errores de los clasificadores con la Base de datos Cleveland (sin Pre-Procesamiento)

Prueba	SVM	ANN	Parzen	*k*-NN
1	0.73	0.66	0.67	0.71
2	0.71	0.66	0.66	0.60
3	0.71	0.66	0.68	0.73
4	0.68	0.75	0.64	0.64
5	0.73	0.77	0.77	0.73
6	0.68	0.86	0.75	0.75
Promedio	0.70	0.72	0.69	0.69

Tabla 4. Errores de los clasificadores con la Base de datos Cleveland (con Pre-Procesamiento)

Prueba	SVM	ANN	Parzen	*k*-NN
1	0.62	0.41	0.28	0.26
2	0.62	0.56	0.26	0.24
3	0.61	0.42	0.32	0.33
4	0.63	0.45	0.32	0.30
5	0.62	0.52	0.25	0.24
6	0.61	0.54	0.26	0.26
Promedio	0.61	0.48	0.28	0.27

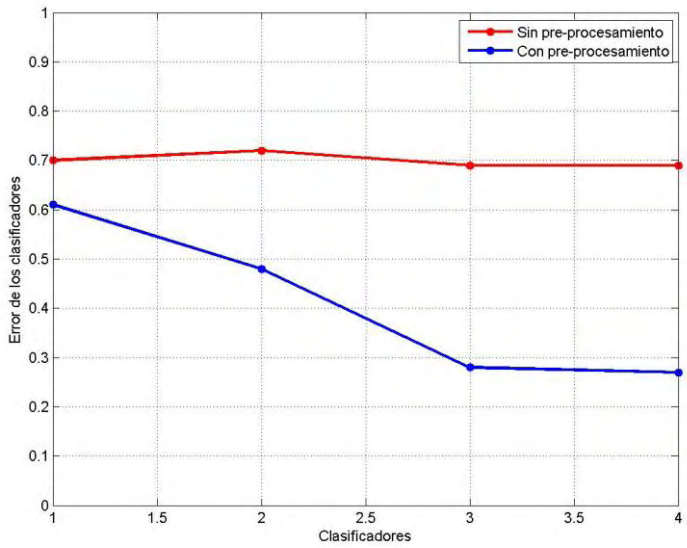


Figura 25. Error de los clasificadores para la base de datos de Cleveland. La línea roja indica el error sin etapa de pre-procesamiento y la línea azul con la etapa de pre-procesamiento. (1) SVM, (2) ANN, (3) Parzen, (4) k-NN.

El desempeño de los clasificadores con la base de datos de cardiotocografía se muestra en las tablas 5 y 6. En la primera se evalúan los clasificadores sin realizar una etapa de pre-procesamiento de datos, se observa que los errores de los clasificadores son pequeños. En la tabla 6 se realiza las pruebas con la etapa de pre-procesamiento. Se observa, que el desempeño de todos los clasificadores mejora, obteniendo muy buenos resultados para esta base de datos. Estos resultados se muestran en la figura 26**.**

Tabla 5. Errores de los clasificadores con la Base de datos de cardiotocografía (sin Pre-Procesamiento)

Prueba	SVM	ANN	Parzen	*k*-NN
1	0.033	0.016	0.033	0.016
2	0.016	0.016	0.033	0.016
3	0.033	0.016	0.050	0.016
4	0.066	0.100	0.050	0.066
5	0.050	0.050	0.050	0.050
6	0.033	0.050	0.050	0.033
Promedio	0.038	0.041	0.044	0.032

Tabla 6. Errores de los clasificadores con la Base de datos de cardiotocografía (con Pre-Procesamiento)

Prueba	SVM	ANN	Parzen	*k*-NN
1	0.016	0.016	0.016	0
2	0.016	0.033	0.033	0.016
3	0.016	0.050	0.033	0.066
4	0.050	0.033	0.033	0.050
5	0.050	0.033	0.050	0.033
6	0	0	0	0
Promedio	0.024	0.027	0.027	0.027

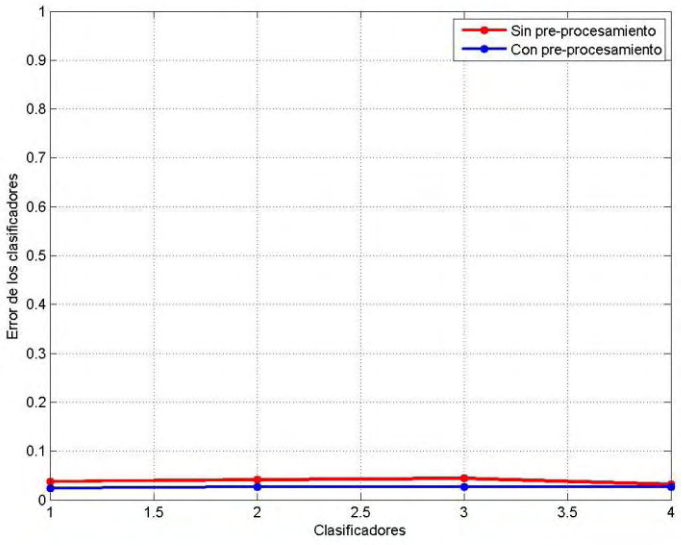


Figura 26. Error de los clasificadores para la base de datos de cardiotocografía.. La línea roja indica el error sin etapa de pre-procesamiento y la línea azul con la etapa de pre-procesamiento. (1) SVM, (2) ANN, (3) Parzen, (4) k-NN.

- **Medidas de desempeño**

Para la base de datos de Cleveland, se obtuvieron las medidas de desempeño utilizando 10 muestras de cada clase, para un total de 50 muestras. Los resultados de estas medidas para los clasificadores SVM, ANN, Parzen y k-NN, se indica en las tablas 7, 8, 9 y 10 respectivamente, En ellas se observa que el mejor clasificador es k-NN con el valor de 98% de sensibilidad, 99.5% de especificidad y 98% de efectividad.

Tabla 7. Medidas de desempeño para la base de datos cleveland-clasificador SVM

Medida	Valor					Promedio
	Clase 0	**Clase 1**	**Clase 2**	**Clase 3**	**Clase 4**	
Se	100	0	100	0	0	40
Sp	97.5	100	27.5	100	100	85
CP	40					40

Tabla 8. Medidas de desempeño para la base de datos cleveland-clasificador ANN

Medida	Valor					Promedio
	Clase 0	**Clase 1**	**Clase 2**	**Clase 3**	**Clase 4**	
Se	30	40	70	100	30	54
Sp	100	97.5	55	97.5	92.5	88.5
CP	54					54

Tabla 9. Medidas de desempeño para la base de datos cleveland-clasificador Parzen

Medida	Valor					Promedio
	Clase 0	**Clase 1**	**Clase 2**	**Clase 3**	**Clase 4**	
Se	90	100	100	100	70	92
Sp	100	92.5	97.5	100	100	98
CP	92					92

Tabla 10. Medidas de desempeño para la base de datos cleveland-clasificador k-NN

Medida	Valor					Promedio
	Clase 0	**Clase 1**	**Clase 2**	**Clase 3**	**Clase 4**	
Se	100	90	100	100	100	98
Sp	100	100	100	100	97.5	99.5
CP	98					98

Los resultados anteriores se pueden apreciar en la figura 27. Es posible concluir que el clasificador con las mejores medidas de desempeño es el clasificador de los k-vecinos más cercanos.

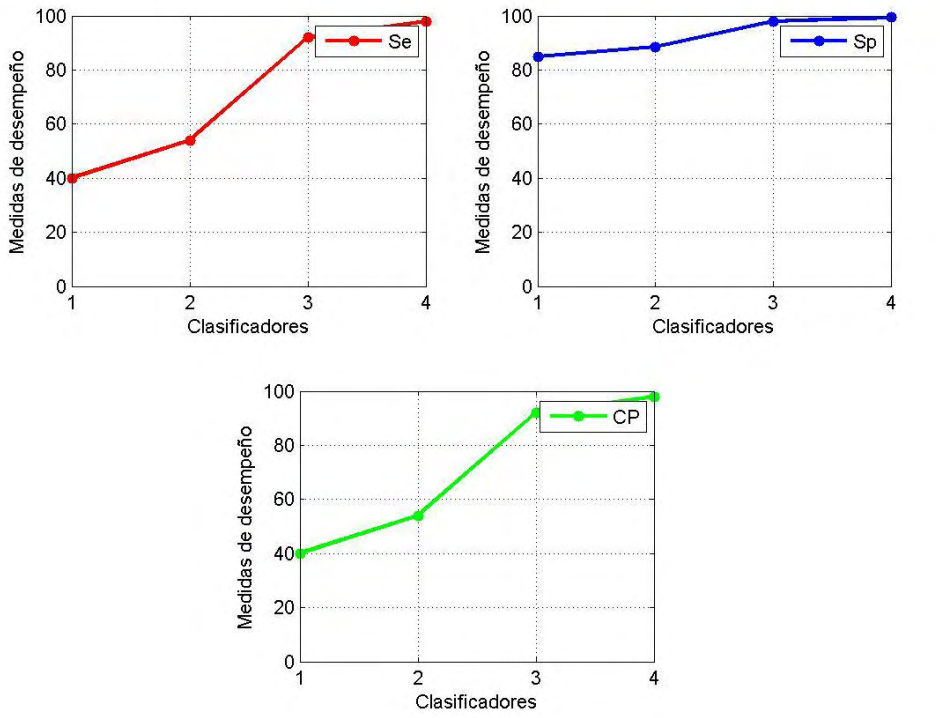


Figura 27. Medidas de desempeño de los clasificadores (1) SVM, (2) ANN, (3) Parzen y (4) *k*-NN para la base de datos Cleveland.

Para la base de datos de cardiotocografía se obtuvieron las medidas de desempeño utilizando 15 muestras de cada clase, para un total de 45 Los resultados de estas medidas para los clasificadores SVM, ANN, Parzen y *k*-NN, se indican en las tablas 11, 12, 13 y 14 respectivamente. En ellas se observa que la mayoría de los clasificadores tienen medidas de sensibilidad, especificidad y efectividad del 100% excepto las redes neuronales (ANN) con el valor de 97.77% de sensibilidad, 98.88% de especificidad y 97.77% de efectividad.

Tabla 11. Medidas de desempeño para la base de datos cardiotocografía-clasificador SVM

Medida	Valor			Promedio
	Clase 1	**Clase 2**	**Clase 3**	
Se	100	100	100	100
Sp	100	100	100	100
CP	100			100

Tabla 12. Medidas de desempeño para la base de datos cardiotocografía-clasificador ANN

Medida	Valor			Promedio
	Clase 1	**Clase 2**	**Clase 3**	
Se	100	93.33	100	97.77
Sp	96.66	100	100	98.88
CP	97.77			97.77

Tabla 13. Medidas de desempeño para la base de datos cardiotocografía-clasificador de Parzen

Medida	Valor			Promedio
	Clase 1	**Clase 2**	**Clase 3**	
Se	100	100	100	100
Sp		100	100	
CP	100			100

Tabla 14. Medidas de desempeño para la base de datos cardiotocografía-clasificador k-NN

Medida	Valor			Promedio
	Clase 1	**Clase 2**	**Clase 3**	
Se	100	100	100	100
Sp	100	100	100	100
CP	100			100

Los resultados anteriores se pueden apreciar en la figura 28. De ella, se puede observar que todos los clasificadores tienen buenas medidas de desempeño, la mayoría de ellos alcanza valores de 100%.

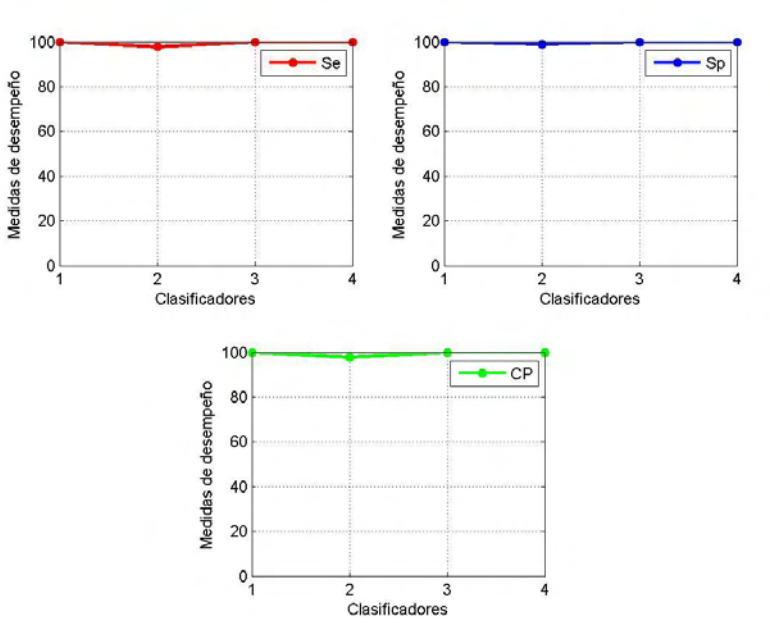


Figura 28. Medidas de desempeño de los clasificadores (1) SVM, (2) ANN, (3) Parzen y (4) *k*-NN para la base de datos cardiotocografía.

Teniendo en cuenta lo anterior, es el *k*-NN, ya que tuvo el mejor desempeño para las dos bases de datos.

- **Matriz de confusión**

Para la base de datos de cleveland, se obtuvieron las matrices de confusión de los clasificadores SVM, ANN, Parzen y k-NN, las cuales se muestran en las tablas 15, 16, 17 y 18 respectivamente. En la primera se observa que el clasificador SVM acertó correctamente en la asignación de las clases 0 y 2, pero la clasificación de las demás clases fue incorrecta, debido a que a los casos de las clases 1 y 4 los asignó a clase 2 y casos de la clase 3 los asignó a clase 1 y 2.

Tabla 15. Matriz de confusión para la base de datos de cleveland con el clasificador SVM

	Clase 0	**Clase 1**	**Clase 2**	**Clase 3**	**Clase 4**
Clase 0	10	0	0	0	0
Clase 1	0	0	10	0	0
Clase 2	0	0	10	9	0
Clase 3	1	0	9	0	0
Clase 4	0	0	10	0	0

El clasificador ANN clasifico correctamente la totalidad de casos de la clase 3, en las otras clases no asignó correctamente todos los casos como se indica en la tabla 16.

Tabla 16. Matriz de confusión para la base de datos de cleveland con el clasificador ANN

	Clase 0	**Clase 1**	**Clase 2**	**Clase 3**	**Clase 4**
Clase 0	3	0	7	0	0
Clase 1	0	4	6	0	0
Clase 2	0	0	7	0	3
Clase 3	0	0	0	10	0
Clase 4	0	1	5	1	3

En la tabla 17 se muestra los resultados de la matriz de confusión con el clasificador de Parzen, en ella se observa que clasificó correctamente todos los casos de la clase 1, 2 y 3, y presentó pocos errores en la asignación de casos en clases 0 y 4.

Tabla 17. Matriz de confusión para la base de datos de cleveland con el clasificador de Parzen

	Clase 0	**Clase 1**	**Clase 2**	**Clase 3**	**Clase 4**
Clase 0	9	1	0	0	0
Clase 1	0	10	0	0	0
Clase 2	0	0	10	0	0
Clase 3	0	0	0	10	0
Clase 4	0	2	1	0	7

El clasificador k-NN acertó en todos los casos de todas las clases, excepto en la clase 4, en donde clasificó un caso como clase 1.

Tabla 18. Matriz de confusión para la base de datos de cleveland con el clasificador k-NN

	Clase 0	**Clase 1**	**Clase 2**	**Clase 3**	**Clase 4**
Clase 0	10	0	0	0	0
Clase 1	0	10	0	0	0
Clase 2	0	0	10	0	0
Clase 3	0	0	0	10	0
Clase 4	0	1	0	0	9

Para la base de datos de cardiotocografía, se obtuvieron las matrices de confusión de los clasificadores SVM, ANN, Parzen y k-NN, las cuales se muestran en las tablas 19, 20, 21 y 22 respectivamente. En ellas se observa que todos los

clasificadores asignaron los casos en la clase adecuada, excepto el clasificador de Parzen que falló en la asignación de un caso en clase 1, el cual pertenecía a clase 2.

Tabla 19. Matriz de confusión para la base de datos de cardiotocografía con el clasificador SVM

	Clase 1	**Clase 2**	**Clase 3**
Clase 1	15	0	0
Clase 2	0	15	0
Clase 3	0	0	15

Tabla 20. Matriz de confusión para la base de datos de cardiotocografía con el clasificador ANN

	Clase 1	**Clase 2**	**Clase 3**
Clase 1	15	0	0
Clase 2	1	14	0
Clase 3	0	0	15

Tabla 21. Matriz de confusión para la base de datos de cardiotocografía con el clasificador de Parzen

	Clase 1	**Clase 2**	**Clase 3**
Clase 1	15	0	0
Clase 2	0	15	0
Clase 3	0	0	15

Tabla 22. Matriz de confusión para la base de datos de cardiotocografía con el clasificador k-NN

	Clase 1	**Clase 2**	**Clase 3**
Clase 1	15	0	0
Clase 2	0	15	0
Clase 3	0	0	15

Con los anteriores resultados, se observa que para la base de datos de cleveland, el clasificador k-NN presentó el mejor desempeño en la asignación de casos a la clase correcta y para la base de casos de cardiotocografía, la mayoría de clasificadores acertó en su tarea.

- **Curvas ROC**

Con los clasificadores SVM, ANN, Parzen y k-NN se obtuvieron las curvas ROC que se muestran en las figuras 29, 30, 31 y 32 respectivamente. Las curvas ilustran la sensibilidad y 1-especificidad de cada uno de los posibles puntos de corte de cada clase en un diagnóstico. La curvas ROC se construyen en base a la unión de distintos puntos de corte, correspondiendo el eje Y a la sensibilidad y el eje X a (1-especificidad). Ambos ejes incluyen valores entre 0 y 1, es decir del 0% a 100%.

De acuerdo a lo anterior, el mejor desempeño lo obtuvo el cuarto clasificador, debido a que realiza una mejor separación entre las clases.

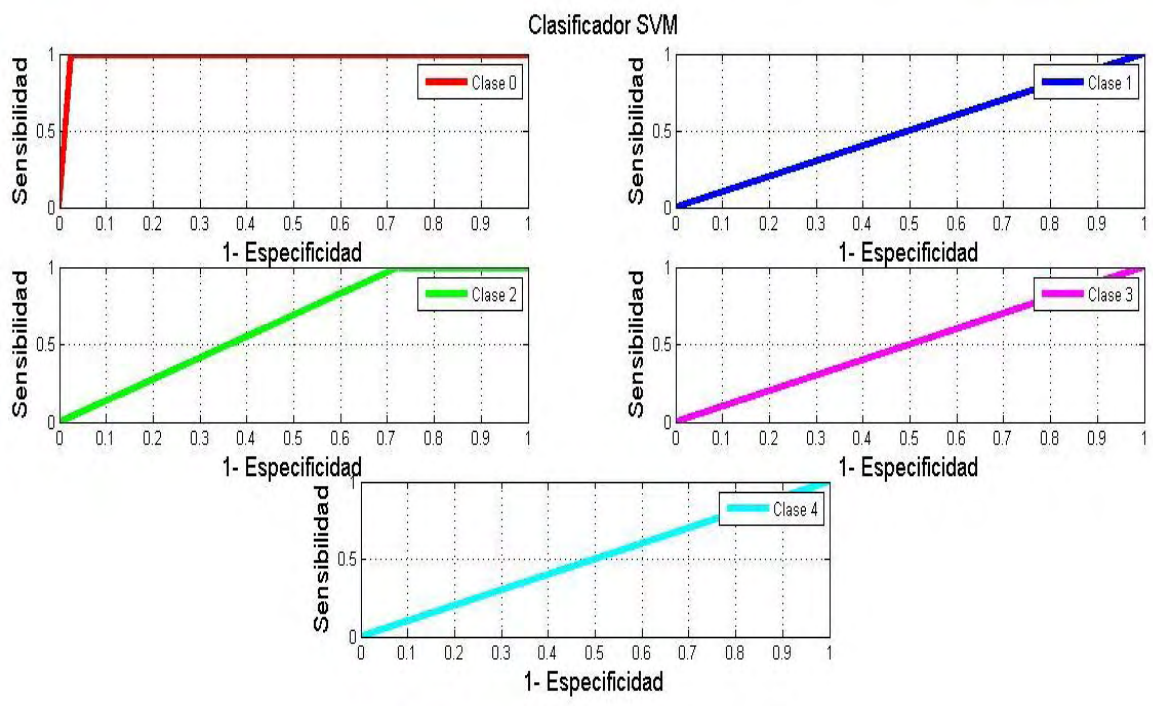


Figura 29. Curvas ROC del clasificador SVM con la base de datos de cleveland.

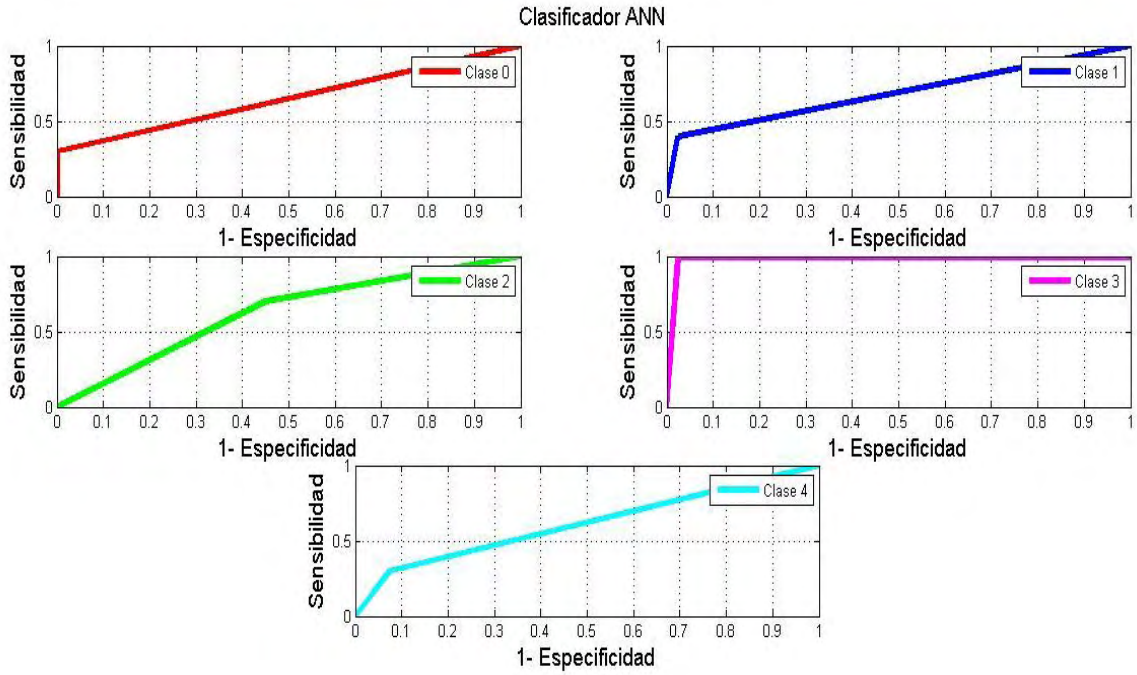


Figura 30. Curvas ROC del clasificador ANN con la base de datos de cleveland.

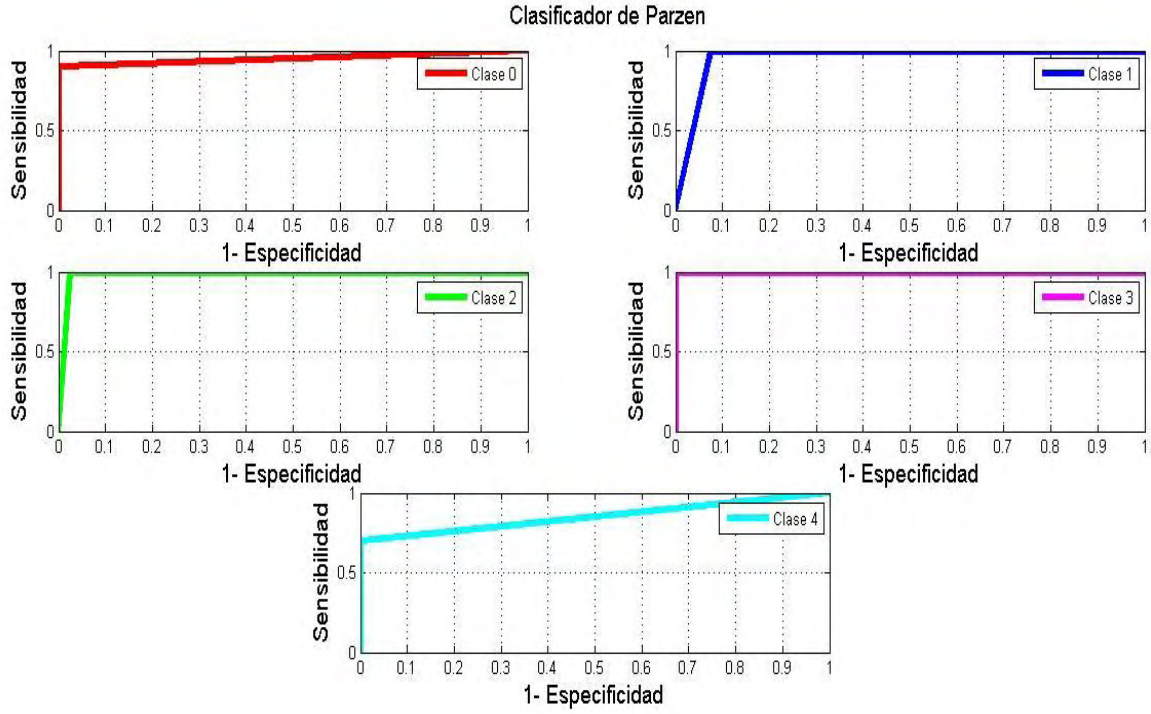


Figura 31. Curvas ROC del clasificador de Parzen con la base datos de cleveland.

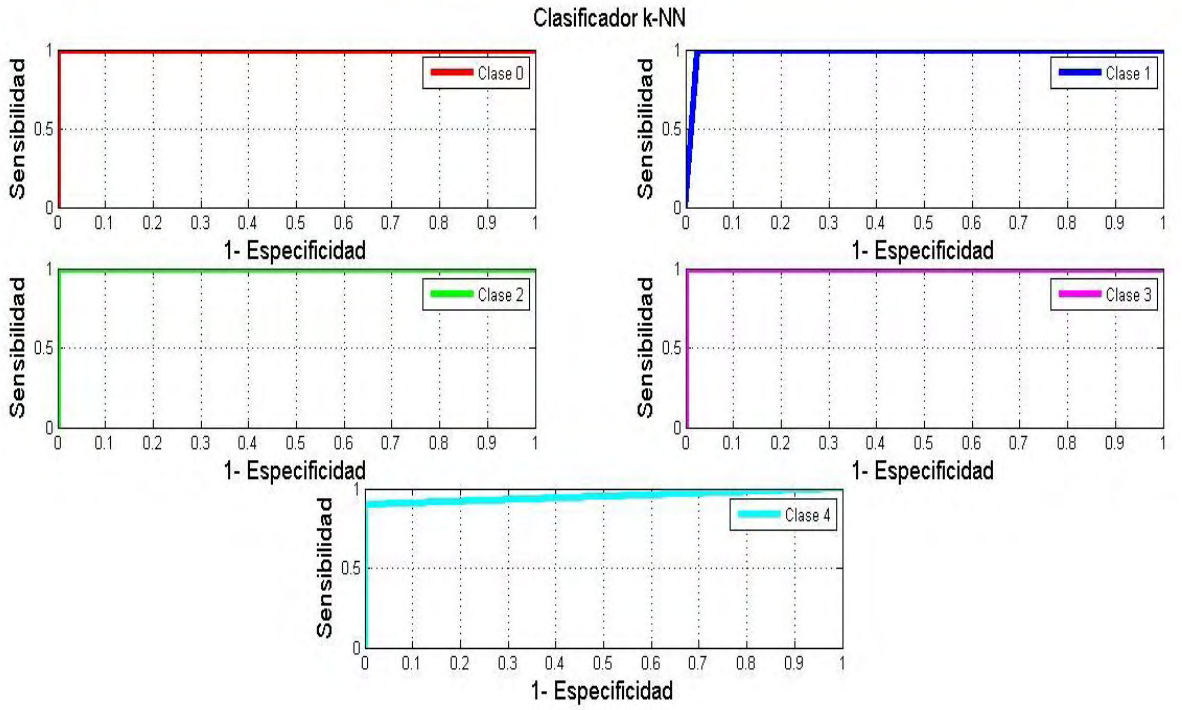


Figura 32. Curvas ROC del clasificador k-NN con la base de datos de cleveland.

Para la base de datos de cardiotocografia se obtuvieron las curvas ROC con los clasificadores SVM, ANN, Parzen y k-NN. Todas las curvas muestran un buen indicador de la precisión de los clasificadores como se observa en las figuras 33, 34 35 y 36.

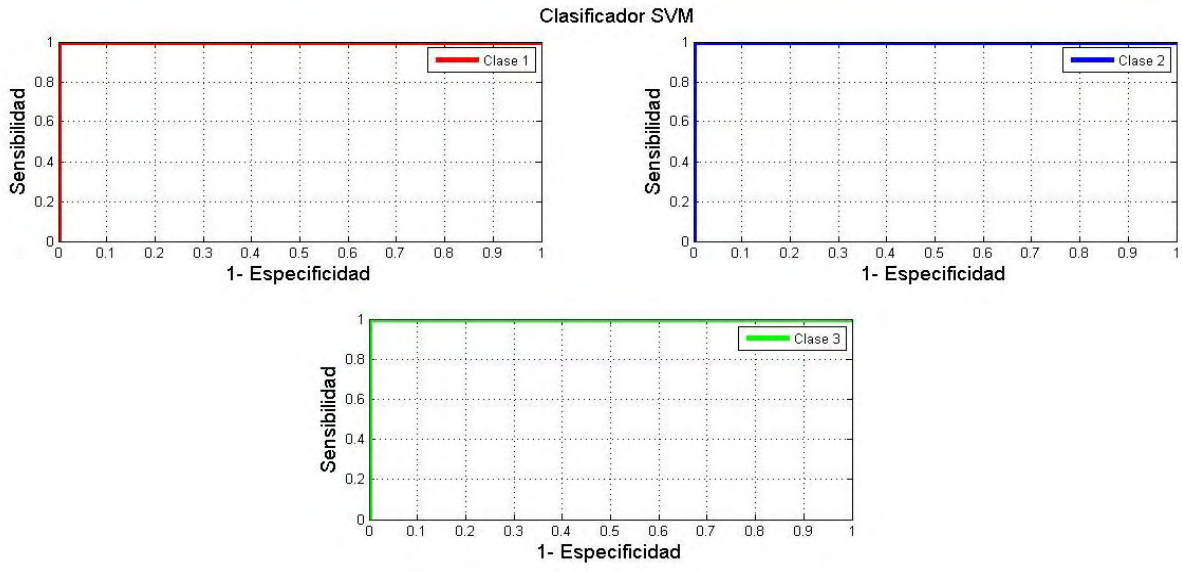


Figura 33. Curvas ROC del clasificador SVM con la base de datos de cardiotocografía.

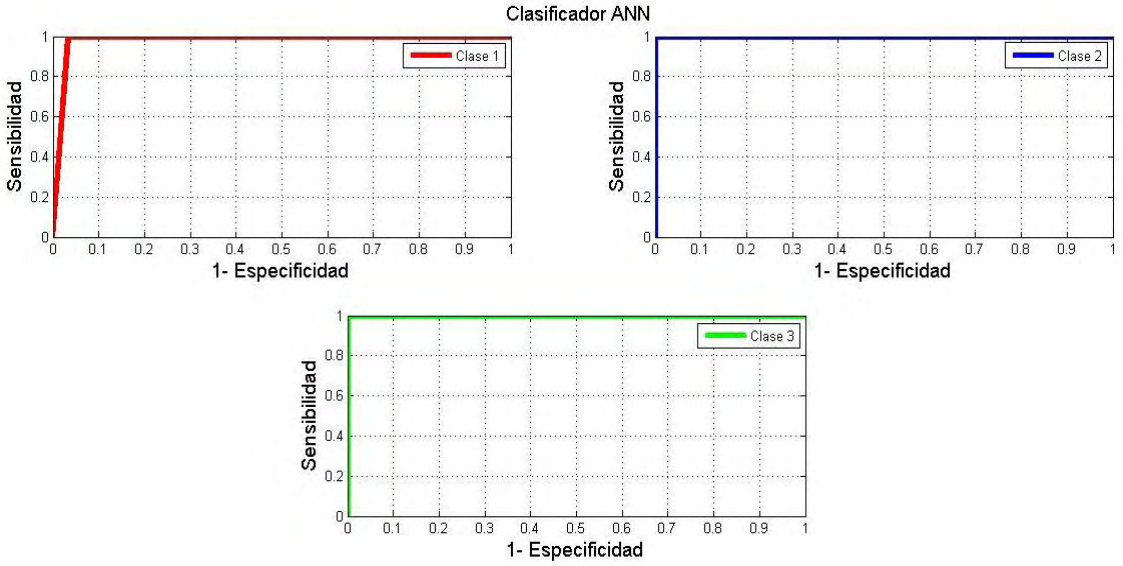


Figura 34. Curvas ROC del clasificador ANN con la base de datos de cardiotocografía.

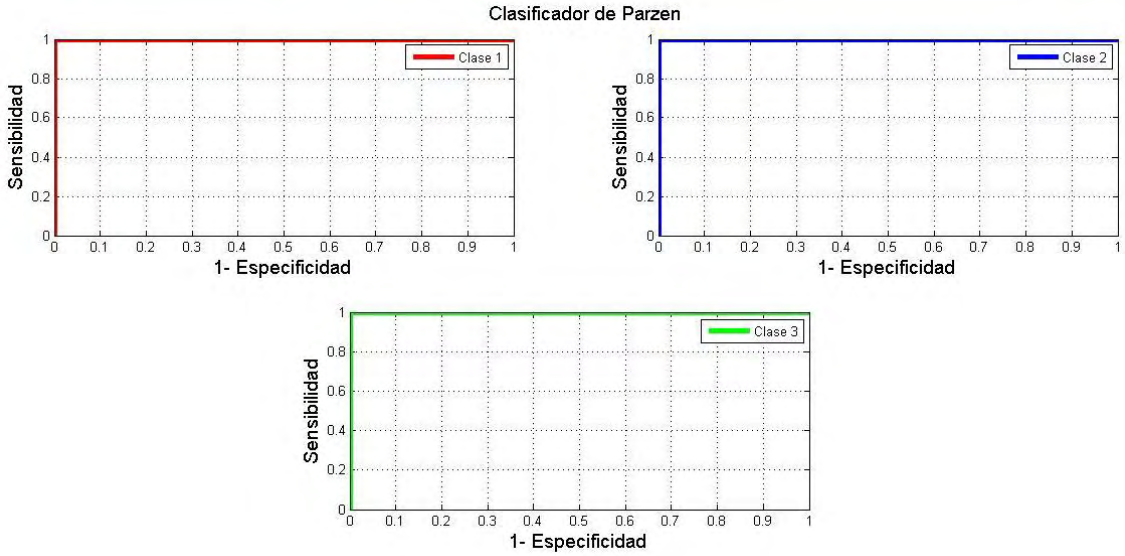


Figura 35. Curvas ROC del clasificador Parzen con la base de datos de cardiotocografía.

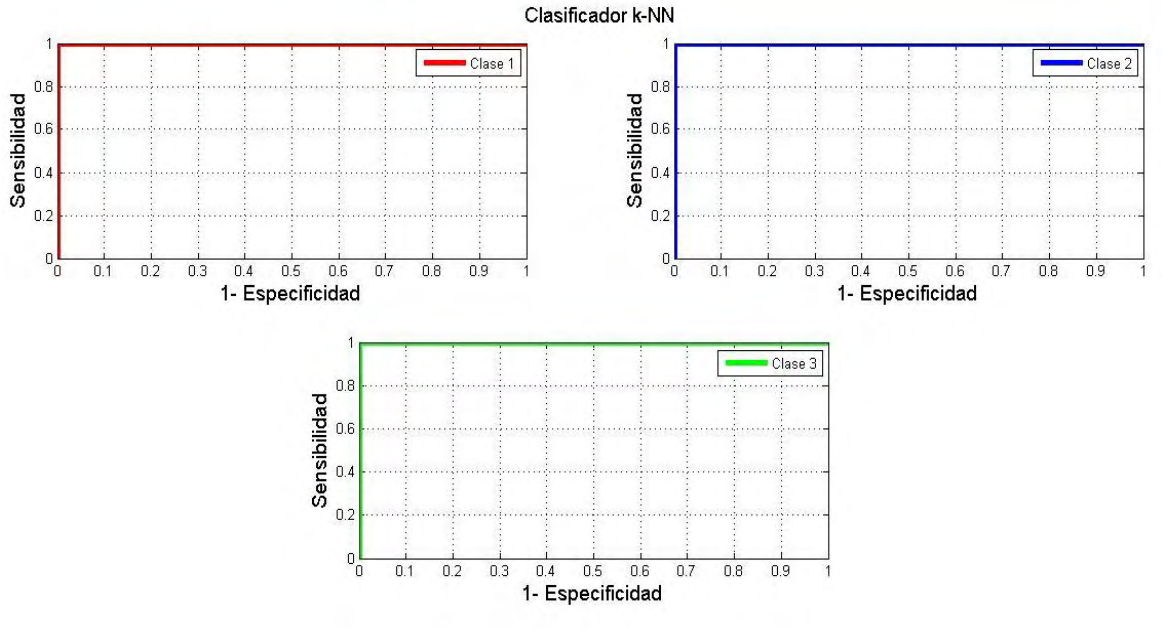


Figura 36. Curvas ROC del clasificador k-NN con la base de datos de cardiotocografía.

Con los resultados obtenidos anteriormente se concluye que el mejor clasificador para integrarse en la etapa de adaptación de CBR es el k- vecinos más cercanos, ya se fue el más sobresaliente en todas las pruebas realizadas con las dos bases de datos.

- **Probabilidades**

El sistema de apoyo al diagnóstico médico integrado en el clasificador multi-clase k-NN fue probado con dos bases de datos. Los resultados adquiridos para la base de datos de cleveland se muestran en la tabla 23, en ella se observa que el CBR acierta en todos los casos tanto en la asignación de la clase correcta como en la estimación de las probabilidades de pertenencia.

Tabla 23. Resultados de clasificación del sistema propuesto para la base de datos de cleveland

Prueba	**Clase asignada por K-NN**	**Probabilidad**					**Clase Real**
		Clase 0	**Clase 1**	**Clase 2**	**Clase 3**	**Clase 4**	
1	**0**	0.90	0.02	0	0	0.06	0
2	**0**	0.82	0.13	0	0	0.04	0
3	**1**	0	0.72	0.07	0.04	0.14	1
4	**1**	0	0.71	0.07	0.05	0.14	1
5	**2**	0	0.01	0.8	0.01	0.16	2
6	**2**	0	0	0.79	0	0.19	2

7	**3**	0	0	0	0.86	0.12	3
8	**3**	0	0	0	0.70	0.29	3
9	**4**	0	0.02	0.16	0.01	0.79	4
10	**4**	0	0.21	0.11	0.09	0.58	4

En la tabla 24 se muestra los resultados para la base de datos de cardiotocografía, se puede visualizar que el clasificador presenta un porcentaje de acierto del 100% para las pruebas realizadas. Además, las probabilidades obtenidas se relacionan con la clase real del caso en la medida que la probabilidad es mayor para la clase a la que pertenece.

Tabla 24. Resultados de clasificación del sistema propuesto para la base de datos de cardiotocografía

Prueba	**Clase asignada por K-NN**	**Probabilidad**			**Clase real**
		Clase 1	**Clase 2**	**Clase 3**	
1	**1**	0.91	0.05	0.02	1
2	**1**	0.92	0.04	0.02	1
3	**1**	0.91	0.05	0.02	1
4	**2**	0.01	0.96	0.02	2
5	**2**	0.01	0.96	0.02	2
6	**2**	0.05	0.92	0.02	2
7	**3**	0	0	1	3
8	**3**	0	0	1	3
9	**3**	0	0	1	3
10	**3**	0	0	1	3

- **Interfaz de CBR**

En esta sección se muestra la interfaz realizada en donde el usuario puede trabajar de forma más interactiva y establecer un contacto más fácil e intuitivo con el sistema CBR. La interfaz se desarrolló en el software de MATLAB. En la figura 37 se visualiza la ventana que contiene los comandos básicos para su uso.

La interfaz ofrece los siguientes beneficios para el usuario:

- Seleccionar la base de datos que desea usar.
- Establecer el número de datos con el que desea entrenar y validar el clasificador multi-clase.
- Agregar un nuevo caso para ser sometido al CBR

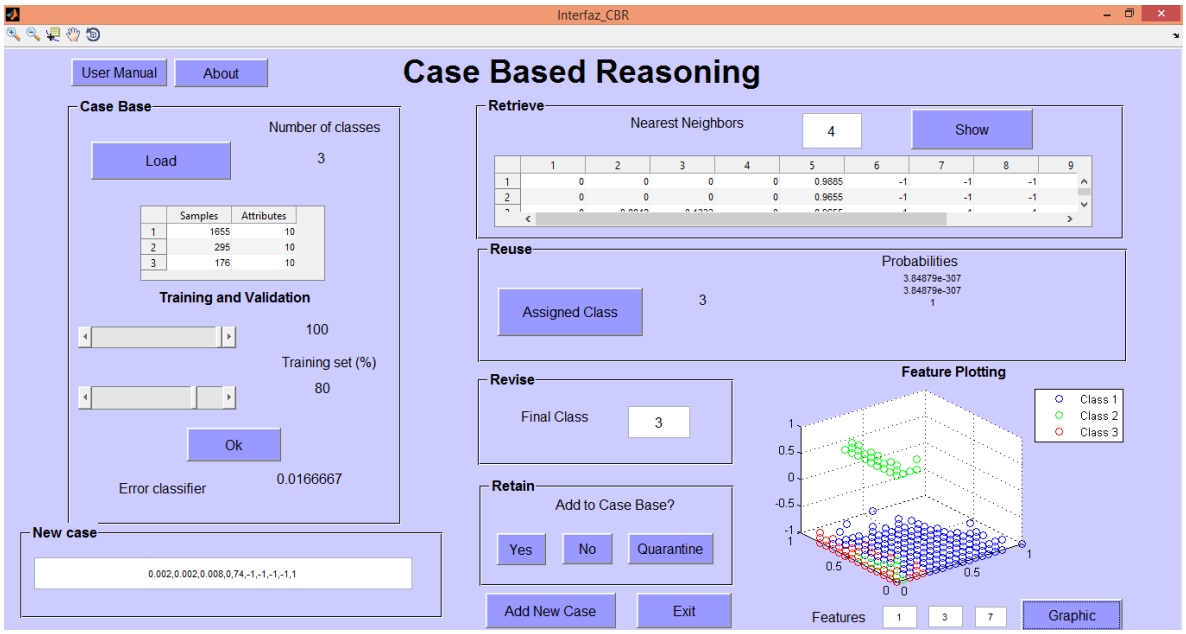


Figura 37. Interfaz de CBR desarrollada. Contiene un botón de manual de usuario explicando su funcionamiento. Permite elegir la base de casos, el ingreso de características del nuevo caso, el número de vecinos cercanos que desea seleccionar y la clase asignada por el CBR.

Como resultados la interfaz arroja:

- El número de clases que posee la base de datos seleccionada.
- Número de atributos de las muestras.
- El error del clasificador.
- Los casos similares al caso de estudio.
- Estimación de probabilidades de pertenencia del nuevo caso a cada clase.
- Un documento en formato Excel en donde se almacena la información de los casos que han sido agregados a la base de casos y enviados a cuarentena.

La interfaz le permite al usuario:

- Visualizar el resultado de clasificador y las probabilidades arrojadas por el sistema.
- Asignar la clase a la que corresponde el nuevo caso.
- Decidir si el caso que se está tratando debe añadirse a la base de casos, no se lo tiene en cuenta o se lo envía a un conjunto de cuarentena, para ser revisado por otro experto.
- Gráfico de las características que se desea observar.

Para visualizar el funcionamiento de la interfaz de una forma más interactiva se muestra un ejemplo:

- Inicialmente con el botón ***Load*** se selecciona la base de datos con la que el usuario quiere trabajar.
- Una vez cargada la base de datos, se muestra el número de clases con las que cuenta dicha base de casos. Para el ejemplo de la figura 12 son 3 clases.
- En la tabla de la figura 38 se visualiza el número de muestras y atributos con los que cuenta la base de casos, por cada clase.

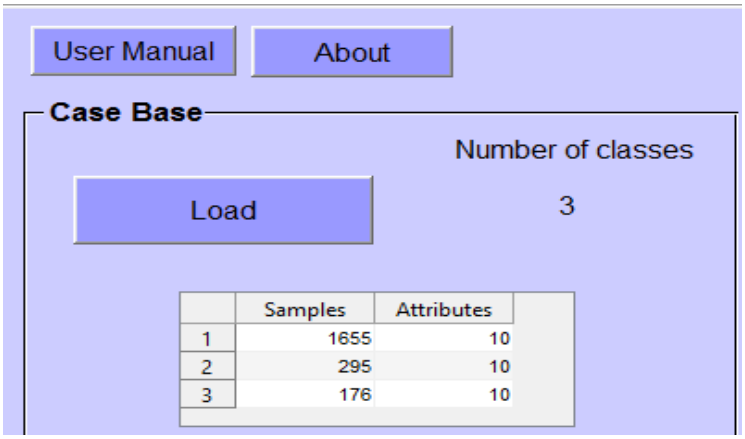


Figura 38. Ejemplo de funcionamiento de la interfaz y explicación de los botones auxiliares: En esta parte de la interfaz se observan los botones **User Manual**: carga un manual de usuario y **About**: información acerca de los autores. Con el botón **Load** se carga la base de datos y en la tabla se señala el número de muestras y los atributos por cada clase. Para este ejemplo el número de clases son 3 y los atributos para cada muestra son 10.

- Mediante los *sliders* se selecciona el porcentaje de datos con el que el usuario desea realizar el proceso de entrenamiento y validación del clasificador. Figura 39.
- Al aceptar los parámetros con el botón ***OK***, el programa muestra el error del clasificador y el usuario procede a agregar el nuevo caso en el espacio ***New case***. Figura 39.

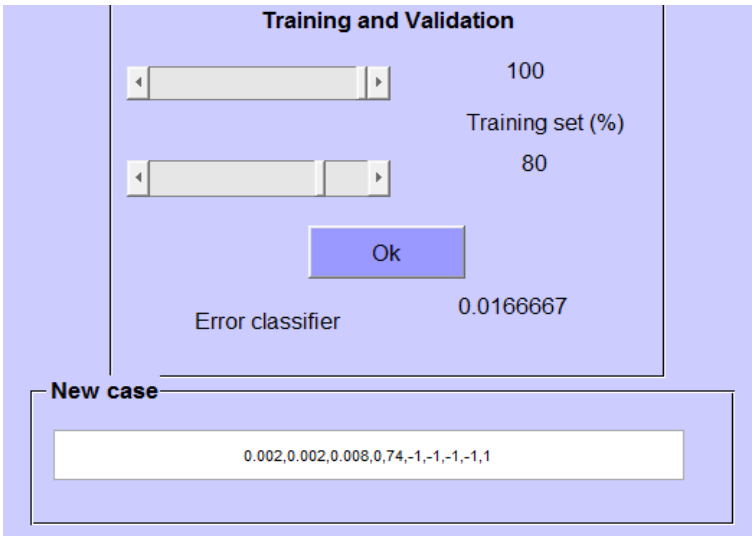


Figura 39. Ejemplo de funcionamiento de la interfaz en la primera parte de ejecución: En esta parte de la interfaz se selecciona el porcentaje de datos para entrenamiento y validación. Una vez presionado el botón **Ok** se conoce el error del clasificador. En el espacio ***New case*** el usuario puede agregar un nuevo caso para ser analizado.

En la etapa de ***Retrieve*** el usuario selecciona el número de casos similares que desea conocer. Se indica además la clase a la que ha sido asignado el nuevo caso y las probabilidades de pertenencia a cada clase. Figura 40.

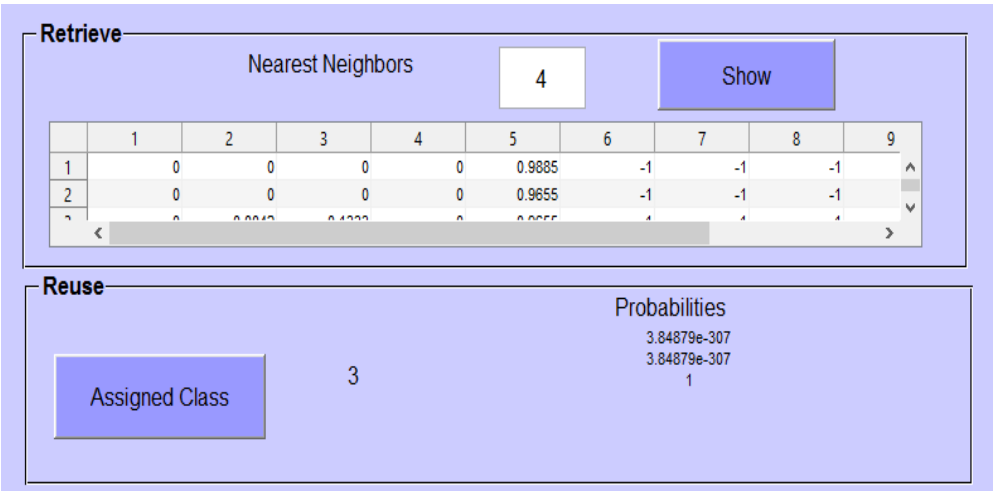


Figura 40. Ejemplo de funcionamiento de la interfaz en las etapas *Retrieve* y *Reuse*: En esta etapa de la interfaz el usuario selecciona el número de casos similares del nuevo caso, Además de ello, se observa la clase del nuevo caso a la que ha sido asignado por el CBR y las probabilidades de pertenencia a cada clase.

- Con la información suministrada, el usuario digita la clase final para ser asignado el nuevo caso, y decide si el caso examinado debe ser agregado a base de casos, no se agrega, o se va aun estado de cuarentena.
- El usuario puede seleccionar 3 atributos del conjunto de la base de casos, para que sean graficados en el *scatter plot*. Figura 41.
- Finalmente, el usuario puede agregar un nuevo caso para repetir el proceso o salir de la aplicación.

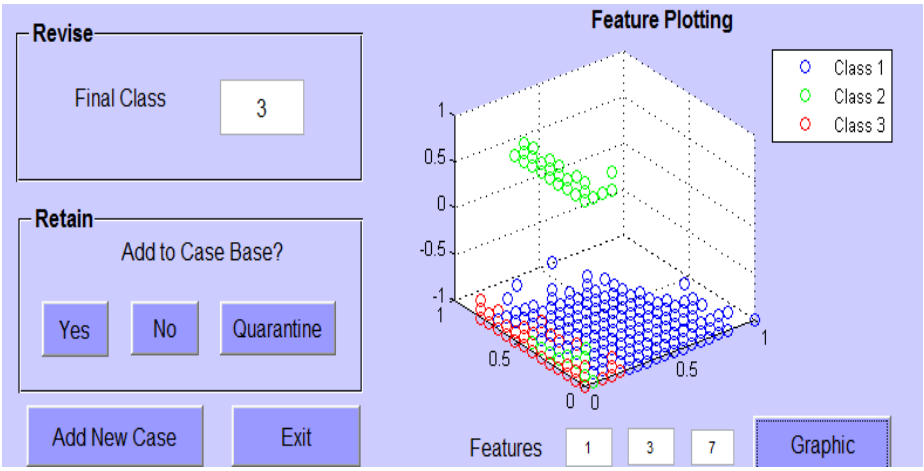


Figura 41. Ejemplo de funcionamiento de la interfaz en las etapas *revise* y *retain*: En la casilla ***final class*** el usuario digita la clase a la que pertenece el caso estudiado y en la etapa de ***retain*** decide si la información es útil para enviarlo a la base de casos o si requiere que sea enviado a cuarentena. En las casillas ***features*** el usuario selecciona los atributos que desea graficar.

7. CONCLUSIONES

De acuerdo a las revisiones realizadas, gran parte de los sistemas encontrados eran bi-clase, limitando el resultado a la presencia o a la ausencia de una enfermedad. Con el desarrollo de este trabajo, el especialista contaría con una variedad de posibilidades que le ayudaría a discernir o a ampliar el proceso de toma de decisiones.

Se realizaron pruebas con múltiples clasificadores multi-clase para determinar el que mejor se adapte al sistema. El resultado mostró que el k-NN fue el mejor candidato para integrarlo a la etapa de adaptación del CBR debido a los resultados de las medidas de desempeño realizadas.

Un gran aporte de nuestro trabajo es la obtención de probabilidades de pertenencia a cada clase mediante la distribución y naturaleza de los datos utilizando los estimadores de Parzen, éste recurso matemático y la información que suministra, permitirán al especialista justificar una respuesta y entregar con mayor exactitud un diagnóstico médico.

Los resultados obtenidos muestran que el sistema CBR arroja resultados satisfactorios en la clasificación de ambas bases de datos de pruebas. Llegando a obtener más de un 90% de asertividad en la clasificación de las muestras.

Se demostró la importancia de la etapa de pre-procesamiento que se debe realizar a los datos y que es una tarea necesaria para su preparación, ya que los resultados varían significativamente cuando no se ha realizado una selección o un balanceo de clases.

La interfaz desarrollada permite al usuario interactuar con los datos y con el sistema de CBR. Entre las posibilidades que tiene se encuentran la de determinar si el caso a examinar es relevante para que sea enviado al conjunto de conocimiento o es un caso extraño que debe ser enviado a cuarentena y ser revisado por otros expertos o simplemente no considerarlo. Esta metodología, brinda al experto un manejo sobre la base de casos y por tanto permite el aprendizaje del sistema.

El algoritmo programado y la interfaz obtenida ofrecen la posibilidad de ajustarse a cualquier base de datos, sin importar el número de clases o de atributos. Lo que indica la versatilidad y funcionabilidad del sistema, ya que es capaz de adaptarse a cualquier situación que de sesea analizar.

8. RECOMENDACIONES

Se recomienda estudiar en la etapa de adaptación la posibilidad de incorporar múltiples expertos al enfoque de CBR, así como también el estudio de otras técnicas que clasificación multi-clase con el objetivo de mejorar los resultados y proporcionar soluciones con un mayor grado de acierto.

Es importante explorar diferentes alternativas para la estimación de probabilidades de pertenencia y diferentes técnicas de pre-procesamiento de datos para mejorar el desempeño de los clasificadores y proporcionar al especialista información útil para facilitar el diagnóstico de los pacientes.

Se propone realizar la interfaz de usuario en un lenguaje de programación de alto nivel con el objetivo de mejorar la interactividad de los usuarios y capacidad de procesamiento.

BIBLIOGRAFÍA

[1] J. M. J. Herrero, «Una Aproximación Multimodal al Diagnóstico temporal mediante razonamiento basado en casos y razonamiento basado en modelos. Aplicaciones en la medicina.,» *Revista Iberoamericana de Inteligencia Artificial,* pp. 77-80, 2007.

[2] M. L. Bonillo, Razonamiento Basado en Casos aplicado a problemas de clasificación, Tesis de grado Doctoral. Universidad de Granada, España, 2003.

[3] Y. Lemos, «Redes Neuronales Y Sistemas de Razonamiento Basado en Casos,» 23 Noviembre 2012. [En línea]. Available: http://nancyyesly.blogspot.com.co/. [Último acceso: 15 01 2017].

[4] E. A. B. López, «Inteligencia Artificial Aplicada al Diagnóstico Médico. Ingeniería de Sistemas y Computación, Universidad Nacional de Colombia, Sede Bogotá,» pp. 1-2.

[5] L. Lozano y J. Fernandez, «Razonamiento Basado en Casos: Una Visión General. Ingeniería Informática, Universidad de Vlladolid, España,» 1-3.

[6] J. P. Febles Rodríguez y V. Estrada Sentí, «Uso del razonamiento basado den casos para la enseñanza de temas médicos,» *Ingeniería Industrial,* vol. 23, nº 1, p. 10, 2002.

[7] M. M. Arjona Giménez, I. López Arévalo y A. Valls Mateu, «Estudio para la implementación de un sistema de razonamiento basado en casos. Proyecto final de carrera. Ingeniería Técnica en Informática de Gestión, Universidad Técnica Superior de Ingeniería, Valencia.,» 2005.

[8] D. Brann, D. A. Thurman y C. M. Mitchell, «Case-Based Reasoning as a Methodology for Accumulating Human Expertise for Discrete System Control,» de *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on*, 1995, pp. 4219-4223.

[9] J. Colomer, J. Melendez y F. Gamero, «Qualitative representation of process trends for situation assessment based on cases.,» *IFAC Proceedings Volumes,* vol. 35, nº 1, pp. 103-108, 2002.

[10] J. Britanik y M. Marefat, «Case-Based Manufacturing Process Planning with integrated support for knowledge sharing,» de *Assembly and Task Planning, 1995. Proceedings., IEEE International Symposium on*, 1995, pp. 107-1112.

[11] J. D. A. Suárez, «Técnicas de Inteligencia Artificial aplicadas al análisis de la solvencia empresarial,» *Documento de Trabajo núm. 206, Universidad de Oviedo, Facultad de Ciencias Económicas,* pp. 1-5, 2000.

[12] R. Rodriguez, G. Fernández y G. López, «Razonamiento Basado en Casos en Ciencias Médicas Sobre plataforma web. Centro de Cibernética Aplicada a la Medicina (CECAM). Instituto Superior de Ciencias Medicas de la Habana. Municipio playa. Ciudad de la Habana. Cuba».

[13] A. Aamodt y E. Plaza, «Case-based reasoning: Foundational issues, methodological variations, and system approaches,» *AI communications},* vol. 7, nº 1, pp. 39-59, 1994.

[14] Y. Sánchez Corales, Y. Pérez Romero, S. Salas Hechavarria y F. Dávila Hernández, «Herramienta informática para la determinación de acciones de salud relacionadas con la hipertensión arterial,» *Revista Cubana de Informática Médica,* vol. 6, nº 1, pp. 87-98, 2014.

[15] A. Abu Hanna y P. Lucas, «Pronostic models in medicine-AI and Statical Approaches,» *Methods of information in medicine,* vol. 40, pp. 6-11, 2001.

[16] M. Mahfouf, M. F. Abbod y D. A. Linkers, «A survey of fuzzy logic monitoring and control utilisation in medicine,» *Artificial intelligence in medicine,* vol. 21, nº 1, pp. 27-42, 2001.

[17] G. R. Lorbada, «¿Qué es el Machine Learning y por qué va a ser clave en el futuro?,» [En línea]. Available: http://lorbada.com/es/que-es-el-machine-learning-y-por-que-va-a-ser-clave-en-el-futuro. [Último acceso: 12 01 2017].

[18] Philips. [En línea]. Available: http://www.comparteinnovacion.philips.es/ rtículos -en-healthtech/ rtículos/machine-learning-inteligencia-artificial-aplicada-al-diagnostico-medico. [Último acceso: 15 01 2017].

[19] Intelygenz, «¿Qué es Machine Learning y qué aplicaciones tiene en nuestro día a día?,» [En línea]. Available: http://www.intelygenz.es/que-es-machine-learning-y-que-aplicaciones-tiene-dia-a-dia/. [Último acceso: 14 01 2017].

[20] A. G. d. T. A. e. C.-. L. Ecuador, «Advanced Tech Computing Group UTPL,» [En línea]. Available: https://advancedtech.wordpress.com/. [Último acceso: 10 01 2017].

[21] Universidad de Jaén, «Prácticas de Teledetección. Departamento de ingenirería Cartográfica, Geodésica y Fotogrametría.».

[22] Telecentro Regional en Tecnologías Geoespaciales, «Fundamentos de percepción remota,» [En línea]. Available: http://geoservice.igac.gov.co/contenidos_telecentro/fundamentos_pr_semana4/index.php?id=31. [Último acceso: 12 01 2017].

[23] R. O. Duda, P. E. Hart y D. G. Stork, Pattern classification, John Wiley & Sons, 2012.

[24] J. L. Rodríguez Sotelo, D. Peluffo Ordoñez y G. Castellanos Rodriguez, «Segment clustering methodology for unsupervised Holter recordings analysis,» *Tenth International Symposium on Medical Information Processing and Analysis,* 2015.

[25] M. Hermann, N. Martinez Madrid y R. Seepold, «Detection of variations in holter ECG recordings based on dynamic cluster analysis,» de *Intelligent Decision Technologies*, Springer, 2015, pp. 209-217.

[26] C. Hernández y J. Rodríguez, «Preprocesamiento de datos estructurados,» *Revista Vínculos,* vol. 4, nº 2, pp. 27-48, 2013.

[27] F. Herrera y J. Cano, «Técnicas de reducción de datos en KDD. El uso de Algoritmos Evolutivos para la Selección de Instancias.,» *Actas del I*

Seminario Sobre Sistemas Inteligentes (SSI'06), Universidad Rey Juan Carlos, Madrid, pp. 165-181, 2006.

[28] P. Langley y and others, «Selection of relevant features in machine learning,» de *Proceedings of the AAAI Fall symposium on relevance*, 1994, pp. 245-271.

[29] J. C. Riquelme Santos, R. Ruiz y K. Gilbert, «Minería de datos: Conceptos y tendencias,» *Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial,* vol. 10, nº 29, pp. 11-18, 2006.

[30] C. Mongay Fernandéz, Estadistica Avanzada. Quimiometría., Valencia: Universidad de Valencia, 2011.

[31] G. Batista, R. Prati y M. C. Monard, «A study of the behavior of several methods for balancing machine learning training data,» *ACM Sigkdd Explorations Newsletter,* vol. 6, nº 1, pp. 20-29, 2004.

[32] L. Puente-Maury, C. Asdrúbal López , W. Cruz-Santos y L. López-García, «Método rápido de preprocesamiento para clasificación en conjuntos de datos no balanceados,» *Research in Computing Science,* vol. 73, pp. 129-142, 2014.

[33] H. He y E. A. Garcia, «Learning from imbalanced data,» *IEEE Transactions on knowledge and data engineering,* vol. 21, nº 9, pp. 1263-1284, 2009.

[34] J. Moreno, D. Rodriguez , M. Sicilia, J. Riquelme y R. Ruiz, «SMOTE-I: mejora del algoritmo SMOTE para balanceo de clases minoritarias,» *Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos,* vol. 3, nº 1, 2009.

[35] D.-Y. Yeung y C. Chow, «Parzen-window network intrusion detectors,» de *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 4, 2002, pp. 385-388.

[36] «UCSC Bioinformatics (Computational Biology),» [En línea]. Available: https://compbio.soe.ucsc.edu/genex/genexTR2html/node11.html. [Último

acceso: 29 12 2016].

[37] «Comenzando con Weka: Filtrado y selección de subconjuntos de atributos basada en su relevancia descriptiva para la clase,» [En línea]. Available: http://www.academia.edu/28515475/Comenzando_con_Weka_Filtrado_y_selecci%C3%B3n_de_subconjuntos_de_atributos_basada_en_su_relevancia_descriptiva_para_la_clase. [Último acceso: 11 01 2017].

[38] J. O. Prieto-Entenza, M. Pupo-Merino y R. Carrasco-Velar, «Modelos de predicción de actividad citotóxica en células SK-N-SH mediante técnicas de softcomputing en una muestra heterogénea de compuestos,» *Revista CENIC Ciencias Biológicas,* vol. 42, nº 3, pp. 111-118, 2011.

[39] E. Camona Suárez, «Tutorial sobre Máquinas de Vectores de Soporte (SVM),» 2014.

[40] M. Á. Lafarga Coscojuela, Biología celular de la neurona y de la sinapsis, Ed. Universidad de Cantabria, 1994.

[41] K. R. Miller y J. S. Levine, Prentice hall biology, Recording for the Blind & Dyslexic, 2009.

[42] M. d. e. G. d. España, «El sistema nervioso,» [En línea]. Available: recursos.cnice.mec.es/biosfera/alumno/1bachillerato/animal/contenidos16.htm. [Último acceso: 12 02 2017].

[43] J. W. Kalat, Células nerviosas y los impulsos nervioso. Psicología biológica, Editorial Paraninfo, 2004.

[44] S. Haykin y N. Network, «A comprehensive foundation,» *Neural Networks,* vol. 2, p. 41, 2004.

[45] F. Izaurieta y C. Saavedra, «Redes Nerunales Artificiales. Departamento de Física, Universidad de Concepción Chile.,» 2000.

[46] D. J. Matich, «Redes Neuronales: Conceptos básicos y aplicaciones,» *Cátedra de informática aplicada a la ingeniería de procesos - orientación I,*

2001.

[47] J. E. Rodríguez, E. A. Blanco y R. O. Camacho, «Clasificación de datos usando el método k-nn,» *Revista Vínculos,* vol. 4, nº 1, pp. 4-18, 2013.

[48] N.-P. Techniques, «Byclb.com,» 2017. [En línea]. Available: http://www.byclb.com/TR/Tutorials/neural_networks/ch11_1.htm. [Último acceso: 24 01 2017].

[49] D. Aha, «The UCI Machine Learning Repository,» The University of California, Irvine, 1987. [En línea]. Available: http://archive.ics.uci.edu/ml/. [Último acceso: 2017 02 11].

[50] P. Pereira, «Evaluation of Rapid Diagnostic Test Performance,» *Proof and concepts in rapid diagnostic tests and technologies. Rijeka: InTech,* 2016.

ANEXOS

Esta sección ha sido destinada a los resultados tangibles logrados con el trabajo realizado en esta tesis. Estos anexos contienen una descripción más ampliada de la metodología, marco experimental y resultados mencionados en la sección 4, 5 y 6 respectivamente donde se expone los detalles más relevantes. Asimismo, las publicaciones y participaciones en eventos que se han realizado por medio de los resultados obtenidos de este trabajo de investigación.

Anexo 1. Pseudocódigo de algoritmo SMOTE

Algoritmo SMOTE (*T, N, k*)

Entradas: Número de muestras de la clase minoritaria *T*; Cantidad de SMOTE *N%*; Número de vecinos más cercanos *k*
Salida: (*N / 100*)**T* Muestras sintéticas de la clase minoritaria

1. **si** *N* < 100
2. **entonces** Aleatorizar las muestras de clase minoritaria T
3. *T = (N / 100) * T*
4. *N = 100*
5. **termina si**
6. *N = (int)(N / 100)(* La cantidad de SMOTE es asumida que está en múltiplos enteros de 100 *)*
7. *k* = Numero de vecinos cercanos
8. *numattrs* = Número de atributos
9. *Sample* [ ] [ ]: Arreglo para muestras minoritarias originales
10. *newindex:* Guarda un recuento del número de muestras sintéticas generadas, inicializado a 0
11. *Synthetic* [ ] [ ]: Arreglo para las muestras sintéticas *(* Calcula k vecinos más cercanos para cada muestra de la clase minoritaria solamente *)*
12. **desde** *i* ← *1* **hasta** *T* **hacer**
13. Calcula los *k* vecinos mas cercanos para cada muestra de la clase minoritaria solamente
14. Populate(*N, i, nnarray*)
15. **termina desde**

Populate(*N, i, nnarray) (* Función para generar las muestras sintéticas *)*

16. **mientras** *N ≠ 0*
17. Escoge un número aleatorio entre *1* y *k*, llamado *nn.* Este paso escoge uno de los *k* vecinos más cercanos de *i*
18. **desde** attr ← *1* **hasta** *numattrs* **hacer**

19. Calcular: *dif = Sample[nnarray[nn]][attr] – Sample[i][attr]*
20. Calcular: *gap* = número aleatorio entre 0 y 1
21. *Synthetic[newindex][attr] = Sample[i][attr] + gap*dif*
22. **termina desde**
23. *newindex++*
24. *N = N-1*
25. **termina mientras**
26. **devuelve** (* *Final de Populate* *)

Final de Pseudocódigo

Anexo 2. Pseudocódigo de k- Vecinos más cercanos

K- Vecinos más cercanos, Clasificar (***X, Y***, *x, k*)

Entradas: Conjunto de entrenamiento *X*; Etiquetas de *X;* Muestra a evaluarle los vecinos cercanos *x*; Número de vecinos cercanos *k*

Salida: Etiquetas de *k* Vecinos más cercanos de *x*

1. **desde** *i = 1* **hasta** *m* **hacer**
2. Calcula la distancia $d_i = d(\boldsymbol{X}_i, x)$
3. **termina desde**
4. Calcula el conjunto I conteniendo los índices para las k distancias más cortas $d(\boldsymbol{X}_i, x)$
5. Ordenar $d_i (i = 1, \ldots, N)$ en orden ascendente
6. Quedarnos con los k casos más cercanos a x
7. **devuelve** etiqueta de$\{\boldsymbol{Y}_i\}\ donde\ i \in I$

Final de Pseudocódigo

Anexo 3. Pseudocógico de CBR

Case Base Reasoning CBR

Entradas: Conjunto de entranamiento (Base de Casos) X_BC; Etiquetas de X_BC, Y_BC; Nuevo Caso NC

Salidas: Casos Similares X_cer ; Probabilidades de pertenencia a cada clase $\Pr\_NC$, Clase del nuevo caso $Classf$; Error del clasificador $error$.

1. Crear conjunto de datos
   A = dataset(X_BC,Y_BC)
2. Seleccionar conjunto de entrenamiento y validación
   C = Entrenamiento
   D = Validación
3. Entrenamiento del clasificador

4. Estimación de error

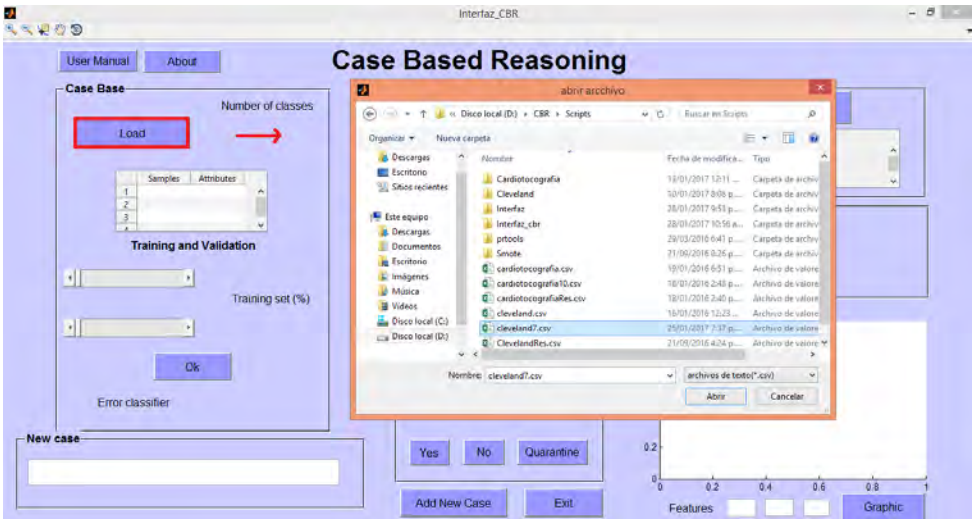
Nuevo Caso

5. **mientras** flag ==1
6. Recuperar casos similares y ordenarlos en forma ascendente
   $X_{ce} = sort(dist(NC, X_{BC}))$
7. Identificar la clase de NC que fue asignada por el clasificador
8. Estimación de probabilidades
9. $X_P = [X_{BC}; NC]$
10. $Y_P = [Y_{BC}; Class(n)]$
11. $X_{ds} = dataset(X_P, Y_P)$
12. $Prob = Parzen\ (X_{ds})$
13. Clase asignada
14. Mostrar $Classf$
15. Retener información?
    0 Añadir a base de casos
    1 No añadir
    2 Enviar a cuarentena
16. Agregar nuevo caso?
    0 Salir → flag = 0
    1 Continuar → flag = 1 → Limpiar variables
17. **termina mientras**
    Final de Pseudocódigo

Anexo 4. Manual de Usuario de la interfaz CBR

- **Cargar base de casos**

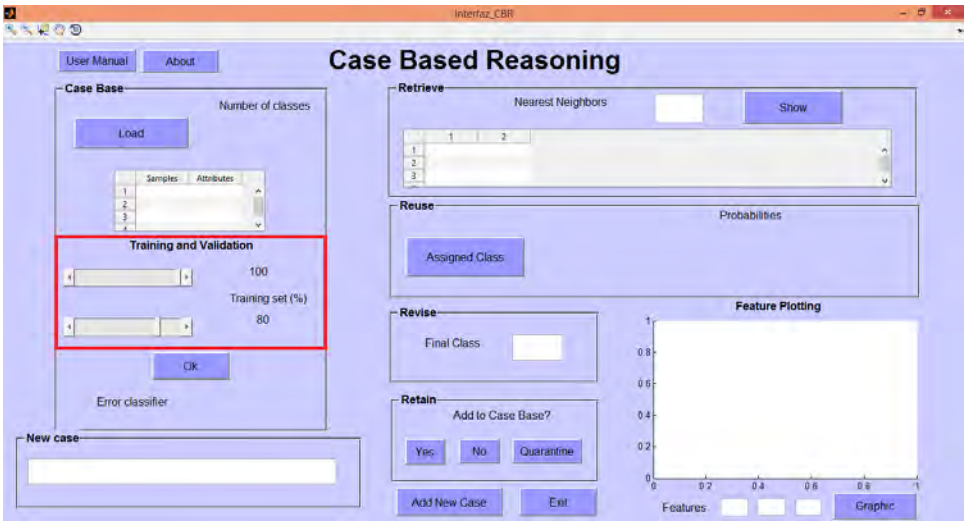
Para cargar la base de casos que servirá como entrenamiento de CBR, se presiona el botón **Load** y se abrirá una ventana para seleccionar los datos y sus etiquetas en cualquier formato (por ejemplo, .csv, .xlsx).



- **Entrenamiento y validación**

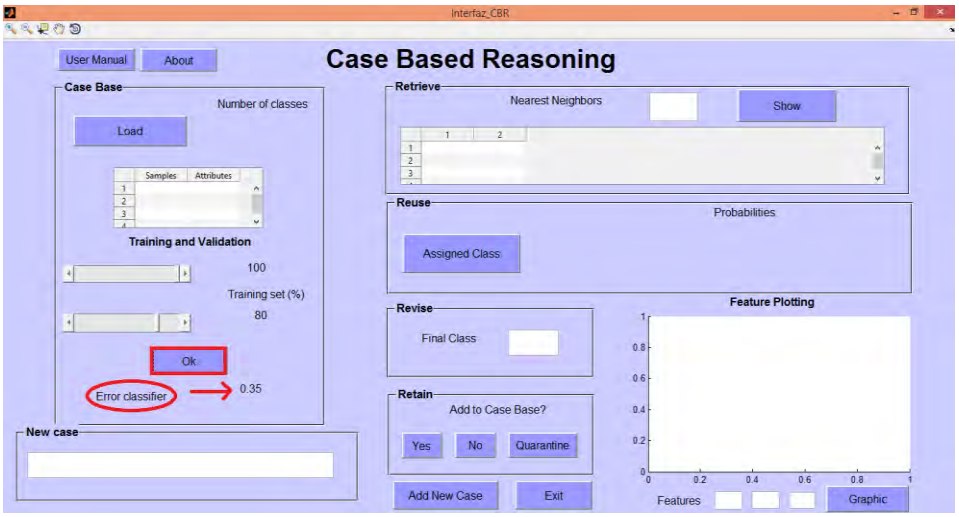
Con el primer **slider** se selecciona el número de muestras de cada clase para los procesos de entrenamiento y validación de los clasificadores.

El segundo **slider** selecciona el porcentaje de muestras que serán utilizadas en la etapa de entrenamiento y el porcentaje restante se utilizará en la etapa de validación.



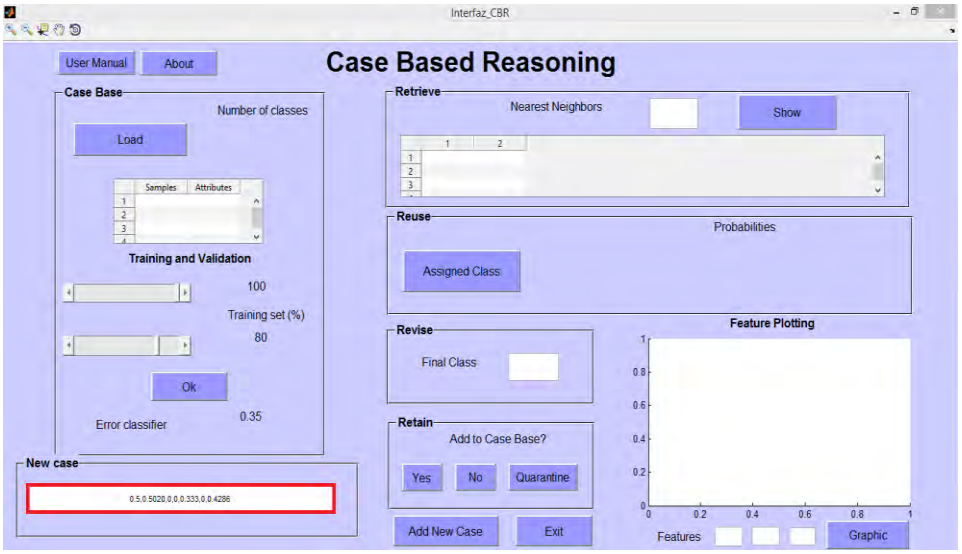
- **Error de los clasificadores**

Al presionar el botón **Ok** mostrará el error del clasificador, esto indicará el grado de fallo que puede tener el CBR al proporcionar un diagnóstico para el nuevo caso.



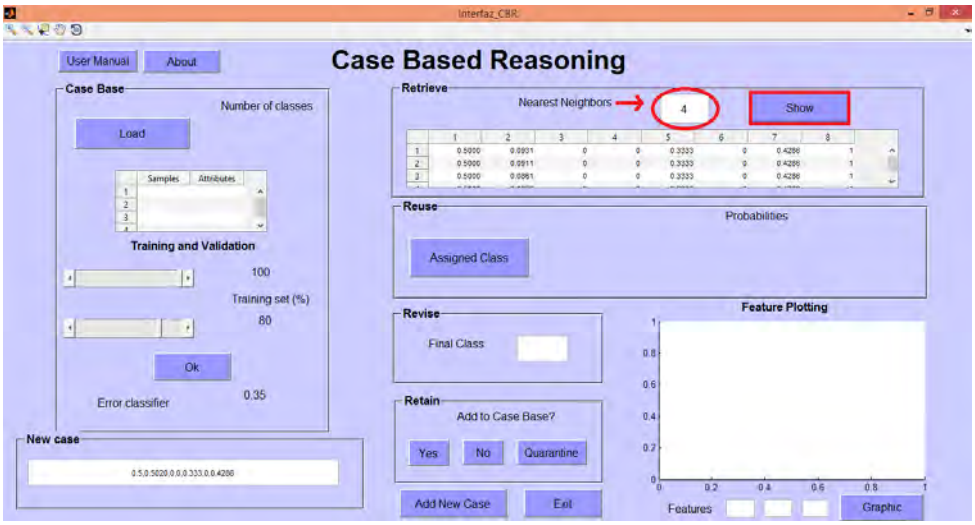
- **Nuevo caso**

En esta parte, se digita las características del nuevo caso separados por comas (,). Cada nuevo caso debe contener el mismo número de atributos de las muestras utilizadas en las etapas de entrenamiento y validación.



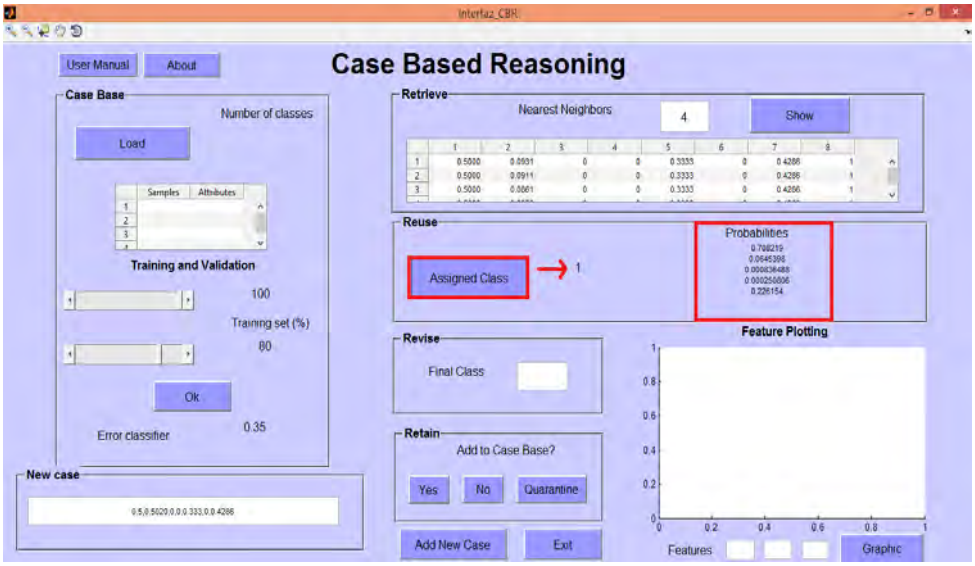
- **Recuperación**

Para recuperar el número de casos similares, se digita el número deseado en la casilla de **nearest neighbors** y luego presione el botón **show** para mostrar los resultados. La última columna de la tabla muestra la clase a la que pertenecen los casos recuperados y en las demás muestran las características de los casos recuperados.



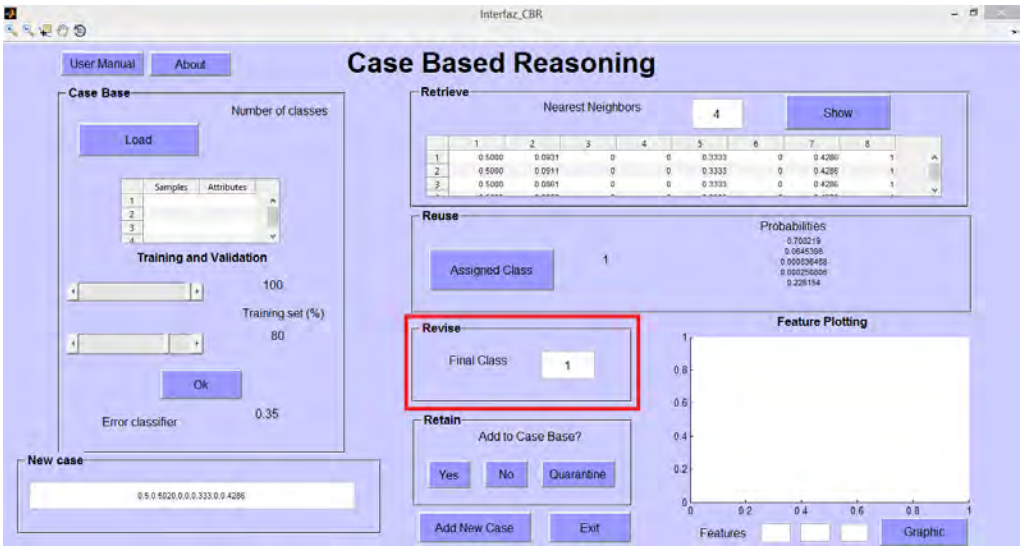
- **Adaptación**

Al presionar el botón **assigned class**, es posible visualizar la clase asignada por el CBR, además de las probabilidades de pertenencia a cada clase.



- **Revisión**

Esta etapa debe ser realizada por el experto médico, ya que verifica y asigna el diagnóstico final del nuevo caso. Para ello, el nuevo caso debe ser digitado en la casilla **final class**.



- **Aprendizaje**

Si desea agregar el nuevo caso a la **base de casos** debe presionar **yes**, con ello se creará un documento en formato **excel** con la información agregada, si no desea agregarlo presione **no** y si el nuevo caso requiere una segunda revisión o es un caso con características extrañas puede enviarlo a la base de casos llamados **quarantine**, asimismo se creará un documento en formato **Excel** con esta información.

Con el fin de mostrar gráficamente el proceso de clasificación de CBR en el gráfico de dispersión, puede introducir el número de las características a mostrar y presionar el botón **graphic**.

Para agregar nuevos casos, presione el botón **add new case** y complete la información para repetir el proceso.

Para cerrar la interfaz, pulse el botón **exit.**

Anexo 5. Artículo de Conferencia Internacional INCISCOS *(International Conference on Information Systems and Computer Science).*

Este anexo contiene el artículo seleccionado en el evento INCISCOS para sustentación en modalidad ponente.

A multi-class extension for case-based reasoning applied to medical problems: A first approach

D. Viveros-Melo*
M. Ortega-Adarme+
Universidad de Nariño
Pasto, Colombia
Email: *dianavive.77@udenar.edu.co
+mabel12-02@udenar.edu.co

X. Blanco Valencia
BISITE Research Group
Universidad de Salamanca, Spain
Email: xiopepa@usal.es

A. E. Castro-Ospina
Research Center of the Instituto
Tecnológico Metropolitano
Medellín, Colombia
Email: andrescastro@itm.edu.co

S. Murillo Rendón
Universidad Autónoma de Manizales
Manizales, Colombia
Email: smurillo@autonoma.edu.co

D. H. Peluffo-Ordóñez
Universidad Técnica del Norte
Ibarra, Ecuador
E-mail: dhpeluffo@utn.edu.ec

***Abstract*—Case-based reasoning (CBR) is a problem solving approach that uses past experience to tackle current problems. CBR has demonstrated to be appropriate for working with unstructured domains data or difficult knowledge acquisition situations, as it is the case of the diagnosis of many diseases. Some of the trends and opportunities that may be developed for CBR in the health science are oriented to reduce the number of features in highly dimensional data, as well as another important focus on how CBR can associate probabilities and statistics with its results by taking into account the concurrence of several ailments. In this paper, in order to adequately represent the database and to avoid the inconveniences caused by the high dimensionality, a number of algorithms are used in the preprocessing stage for performing both variable selection and dimension reduction procedures. Subsequently, we make a comparative study of multi-class classifiers. Particularly, four classification techniques and two reduction techniques are employed to make a comparative study of multi-class classifiers on CBR.**

(a) Keywords— case based reasoning; high dimensionality; variable selection.

I. INTRODUCTION

Case-based Reasoning (CBR) solves new problems by retrieving previously solved problems and reusing the corresponding solutions. In the past twenty years, CBR methodology has attracted much attention, showing its usability in applications usually focused on open and weak theory domains, such as medical diagnosis, design, corporate planning and many engineering domains [1]. The core of the CBR is the case, which usually indicates a problem situation. From another point of view, a case is prior learning experience, which has been captured and can be reused to solve future problems. The life cycle for solving a problem using CBR is mainly carried out in four phases: to identify the current problem and find a past case similar to the new case (retrieve), using the case and suggest a solution to the current problem (reuse/adaptation), evaluate the proposed solution (revise), and update the system to learn from experience (retain) [2].

The CBR has demonstrated to be an appropriate methodology for:

- Working with unstructured domains data or difficult knowledge acquisition situation, for example, many diseases are not well understood by formal models or universally applicable guidelines [3], [4].
- Making tasks in the medical domain. These tasks cover diagnosis, therapy planning, interacting with patients, identifying medical errors etc. Among these tasks, medical diagnosis has been one of the most popular research subjects in both medical informatics and computer science communities. [5]
- When guidelines are available, they provide a general framework to guide clinicians, but require consequent background knowledge to become operational, which is precisely the kind of information recorded in practice cases; cases complement guidelines very well and help to interpret them [4].
- Highly data intensive field in medicine, where it is advantageous to develop a system capable of reasoning from pre-existing cases from an electronic medical record, for instance, or from cases mined from the data. [4].

So, the CBR, is a reasoning process, which is medically accepted and also getting increasing attention from the medical domain. A number of benefits of applying CBR in the medical domain have already been identified [4], [6], [7]. However, the medical applications offer a number of challenges for the CBR researchers and drive advances in research [8].

In order to adequately represent data and to avoid the inconveniences caused by its high dimensionality, we propose the use of variable selection and dimension reduction

techniques in a preprocessing stage for CBR tasks, finally, we make a comparative study of multi-class classifiers to assess processed data performance.

The rest of this paper is structured as follows: Section II describes the proposed methodology, as well as the pattern recognition procedures used in this work. Section III presents the proposed experimental setup. Results and discussion are gathered in section IV. Finally, some concluding remarks and future works are drawn in Section V.

II. MATERIAL AND METHODS

This section outlines the proposed framework to assess the feasibility of using multi-class schemes within CBR approaches. Particularly, we resort to the adaptation of a pattern recognition stages into the CBR life cycle.

In the CBR scheme, the recovery is the most important stage, since in this phase the system finds the most similar cases to the current unknown case, simulating an efficient memory as a human expert would [9]. By combining the CBR methodology with classifiers, a cost function would be used to find the nearby cases.

The next stage where we adapt classifiers would be in the adaptation stage, because we want to show the answer in terms of probabilities. With the classifier we can find the membership degree of the new case in each of the classes, which would be helpful for medical staff.

To that end, we propose to carry out a comparative study of multi-class classifiers within preprocessing, recovery and adaptation CBR stages. Fig. 1 depicts the proposed methodology to perform the comparison of multi-class classifiers.

A. Preprocessing

Variable selection: First, as preprocessing stage a variable selection procedure is employed. In this work, we use the so-called correlation based feature subset (CfsSubsetEval) algorithm, which evaluates the relevance of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy among them. And as search method the bestfirst algorithm, to reduce the number of parameters per instance of a dataset with a backtracking. It starts with the whole set of attributes and search backward to reduce the number of parameters per instance of a dataset.

Dimensionality Reduction: After performing variable selection and aiming to improve both visual inspection and classification performance, a dimensionality reduction stage is employed by using well known methods, namely Laplacian Eigenmaps (LE) and t-distributed stochastic neighbor embedding (t-SNE).

B. Adaptation and recovery

Here, with the aim of accomplishing a multi-class case recovery, representative multi-class classifiers are considered. Due to their characteristics, we select the following classifiers: *K* Nearest Neighbor Classifier (*K*-NN) being a geometric-distance-based-approach, artificial neural networks (ANN) being a heuristic-search-based approach, support vector machines (SVM) being a model-based classifier, and Parzen's Classifier (PC) being a non-paramatric density-based classfier.

III. EXPERIMENTAL SETUP

A. Database

For evaluating the proposed methodology, we used two databases from UCI Machine Learning Repository. The first one, named Cardiotocograms, contains 2126 fetal cardiotocograms belonging to different classes. This data set consists of 21 attributes which include LB - FHR baseline (beats per minute), AC of accelerations per second, FM of fetal movements per second, UC of uterine contractions per second, DL of light decelerations per second, DS of severe decelerations per second, DP of prolongued decelerations per second, ASTV percentage of time with abnormal short term variability, MSTV mean value of short term variability, ALTV percentage of time with abnormal long term variability, MLTV mean value of long term variability, Width width of FHR histogram, Min minimum of FHR histogram, Max Maximum of FHR histogram, Nmax of histogram peaks, Nzeros of histogram zeros, Mode - histogram mode, Mean histogram mean, Median histogram median, Variance histogram variance, Tendency histogram tendency, CLASS FHR pattern class code (1 to 10) and NSP fetal state class code (Normal=1; Suspect=2; Pathologic=3).

The second database, named Cleveland, contains 303 instances. Consisting of 13 attributes which include age, sex, chest pain type, resting blood pressure, cholesterol, fasting blood sugar, resting ECG, maximum heart rate, exercise induced angina, oldpeak, slope, number of vessels coloured, thal and the classification values from 0 no presence to 4 types of heart diseases.

B. Parameter settings and procedures

As outcomes of the preprocessing stage, we obtain that Cardiotocograms database is reduced to 10 features, and Cleveland database to 7 features. Subsequently, as part of the same stage, by using dimensionality reduction techniques Cardiotocogram database is reduced to a 2-, 3-, 5-, 8-dimensional space. Likewise, Cleveland database is reduced to 2-, 3-, 5-dimensional space. As well, the whole subset of selected variables is considered for both databases.

For classification techniques, it should be stated out that a 20-fold cross-validation was performed to achieve unbiased results. Particularly, the following setup is established:

- *K-NN*: This instance-based classification technique needs a value for the number of neighbors (*K*), such parameter is optimized by means of a leave-one-out strategy.
- *ANN*: The heuristic-based classification technique requires a number of units per hidden layer. In this work, a back-propagation trained feed-forward neural net is used with a single hidden layer. The number of units is computed from the data itself as the half of the instances divided by feature size

plus the number of classes. The weight initialization consists of setting all weights to be zero, as well as the dataset is used as a tuning set.

- *SVM*: This instance-based classification method takes advantage of the kernel trick to compute the most discriminative non-linear hyperplane between classes. Therefore, its performance heavily depends on the selection and tuning of the kernel type. For this work a Gaussian kernel is selected given its ability of generalization and its band-width parameter was fixed by the Silverman's rule [10].
- *PC*: This probabilistic-based classification method requires a smoothing parameter for the Gaussian distribution computation, which is optimized.

As a performance measure, it is used the standard mean classification error.

IV. RESULTS AND DISCUSSION

Achieved results for different number of dimensions as well as different classifiers are shown in Table I as the mean and standard deviation over the 20 folds runs. It can be seen how Cleveland dataset is a challenge task since performance is poor for all classifiers. It should be stated also that dimensionality reduction does not necessarily improves classification performance for both dimensionality reduction techniques. Nevertheless, by reducing dimensionality there is a gain in visual analysis of data as can be appreciated in Figure 1, particularly it can be seen how in 2D (Figures 2(a) and 2(c)) and 3D (Figures 2(b) and 2(d)) Cleveland data is highly overlapped which is consistent with achieved results. It should be noted that the error for SVM classifier is 0.397 ± 0.07, which is not far from the result obtained in [11], where the classification accuracy with 7 attributes is of 70.36%.

For Cardiotocograms dataset classes separability is evident in lower dimensions, i.e. 2D and 3D, as depicted in Figures 3(a) to 3(d) leading to outstanding results as shown in Table I,

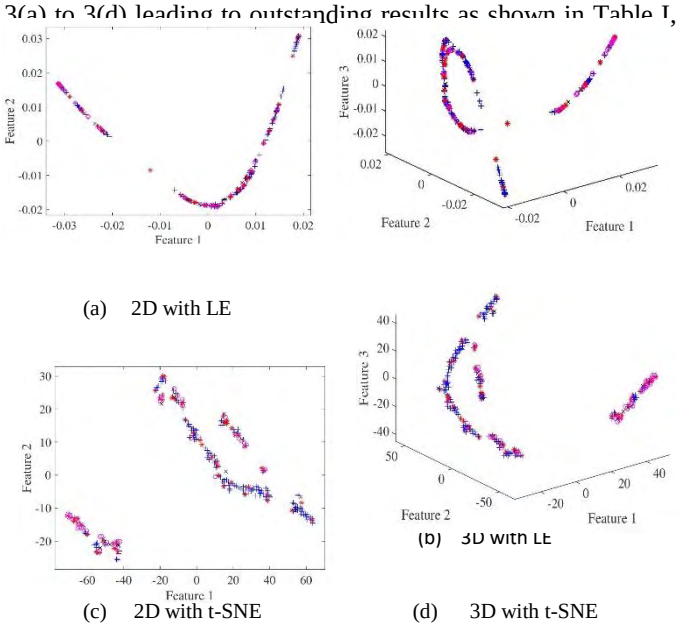


Fig. 2. Low-dimensional scatterplots for Cleveland database. Figures (a), (c) show the first two features from database. Figures (b), (d) show the first three features from database.

however, as for Cleveland dataset, dimensionality reduction does not substantially improves classification performance on Cardiotocograms dataset even though it enhances data visualization. We can see that for the Cardiotocograms database the best result was using the SVM classifier the error is 0.028 ± 0.016, improving the results obtained in [12] where they achieved an average accuracy of 0.9328.

TABLE I

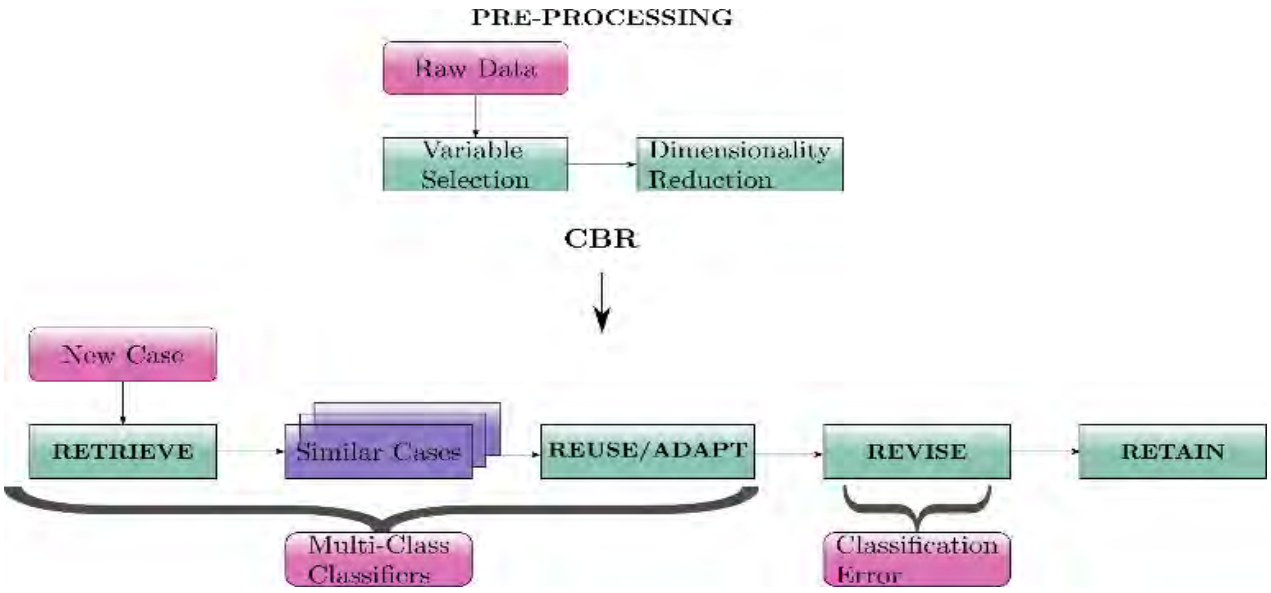
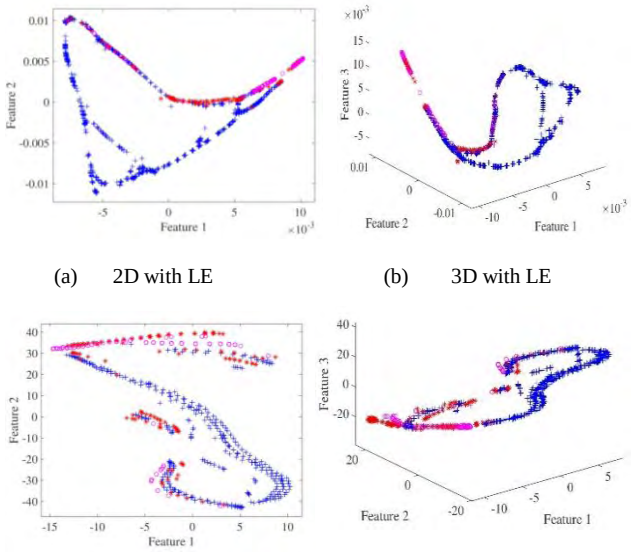


Fig. 1. Block diagram of proposed methodology. The aim of the comparative study is assessing the possibility of incorporating multi-class classifiers into CRB approaches design, as well as identifying the best classifier for this task.

ACHIEVED CLASSIFICATION PERFORMANCE OVER 20-FOLD CROSS VALIDATION FOR CONSIDERED DATABASES AND DIMENSIONALITY REDUCTION TECHNIQUES

DB	Reduction Technique	# dimd	*K*-NN	ANN	SVM	PC
Cleveland	t-SNE	2	0.381 ± 0.08	0.389 ± 0.067	0.389 ± 0.013	0.393 ± 0.093
		3	0.382 ± 0.06	0.367 ± 0.09	0.389 ± 0.013	0.393 ± 0.069
		5	0.397 ± 0.07	0.362 ± 0.089	0.389 ± 0.028	0.4 ± 0.087
		7	0.397 ± 0.07	0.347 ± 0.062	0.401 ± 0.029	0.393 ± 0.069
	LE	2	0.408 ± 0.069	0.393 ± 0.077	0.389 ± 0.013	0.393 ± 0.041
		3	0.397 ± 0.066	0.397 ± 0.075	0.389 ± 0.013	0.374 ± 0.047
		5	0.389 ± 0.067	0.404 ± 0.085	0.412 ± 0.036	0.389 ± 0.067
		7	0.389 ± 0.065	0.382 ± 0.065	0.397 ± 0.07	0.404 ± 0.064
Cardiotocograms	t-SNE	2	0.037 ± 0.015	0.084 ± 0.038	0.071 ± 0.017	0.077 ± 0.017
		3	0.036 ± 0.016	0.073 ± 0.02	0.054 ± 0.019	0.076 ± 0.018
		5	0.032 ± 0.017	0.088 ± 0.017	0.039 ± 0.019	0.075 ± 0.016
		8	0.035 ± 0.016	0.079 ± 0.016	0.033 ± 0.017	0.075 ± 0.019
		10	0.031 ± 0.017	0.082 ± 0.036	0.028 ± 0.016	0.076 ± 0.019
	LE	2	0.045 ± 0.014	0.078 ± 0.016	0.086 ± 0.017	0.102 ± 0.023
		3	0.054 ± 0.018	0.072 ± 0.015	0.061 ± 0.016	0.09 ± 0.02
		5	0.042 ± 0.014	0.075 ± 0.031	0.048 ± 0.014	0.09 ± 0.016
		8	0.039 ± 0.015	0.067 ± 0.019	0.038 ± 0.013	0.065 ± 0.016
		10	0.381 ± 0.25	0.06 ± 0.017	0.038 ± 0.016	0.063 ± 0.016

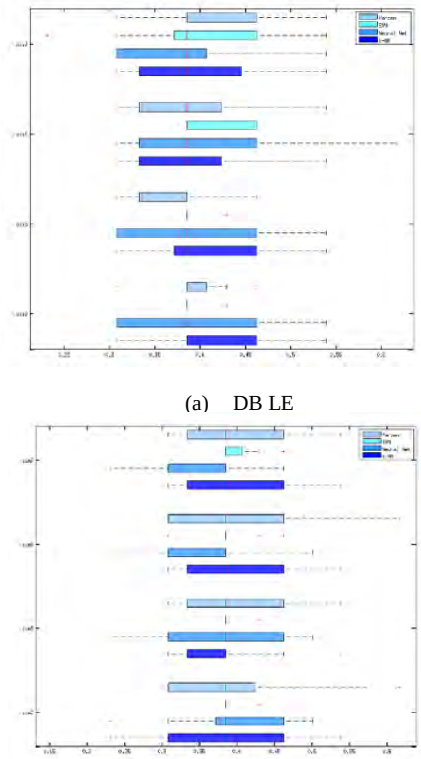
By performing a stability assessment, it could be seen from Figures 4,5 by the width of the error boxplots how SVM and *K*-NN classifiers achieves the best results for considered Cardiotocogram and Cleveland databases. Moreover, it should be noted how SVM classification results are the most stable of the considered classification techniques.



(a) 2D with LE (b) 3D with LE

(c) 2D with t-SNE (d) 3D with t-SNE

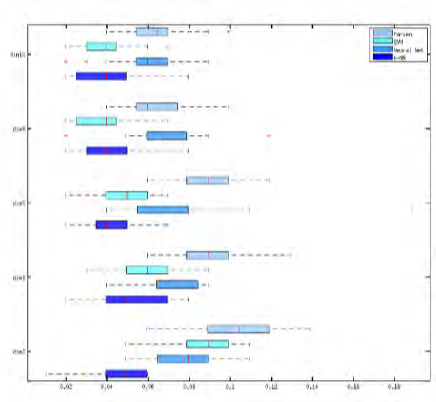
Fig. 3. Low-dimensional scatterplots for Cardiotocograms database. Figures (a), (c) show the first two features from database. Figures (b), (d) show the first three features from database.



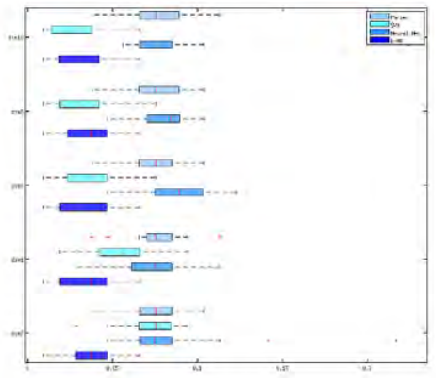
(a) DB LE

(b) DB t-SNR

Fig. 4. Classification error boxplots for considered classification techniques on Cleveland databases.



(a) DB LE



(b) DB t-SNE

Fig. 5. Classification error boxplots for considered classification techniques on Cardiotocograms databases.

V. CONCLUSIONS AND FUTURE WORK

This work presents a feasibility evaluation of the use of techniques from the field of pattern recognition into CBR frameworks, so that conventional CBR can be extended to multi-class scenarios.

Experimentally we prove that the SVM classifier is a good candidate for integration with the CBR approach to create a generic system to assist physicians in the diagnosis of patients and is capable of working with databases multiclass associating probabilities each class, responding to one of the challenges of [4], [13].

As a future work, we will explore the possibility to design a case recovery stage for CBR able to deal with mult-class cases while providing users with class membership (probabilities to belong) estimates for a new case.

ACKNOWLEDGMENTS

Authors would like to thank to the Facultad de Ingeniería en Ciencias Aplicadas as well as electronic engineering and telecommunications program from Universidad Técnica del Norte.

REFERENCES

L. Huan, X. Li, D, Hu, T, Hao, L. Wenyin and X, Chen. Adaptation Rule Learning for Case-Based Reasoning. " *Concurrency and Computation: Practice and Experience* ", 21(5), 673-689, 2009.

J. Kolodner, Case-based Reasoning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

J. M. Juárez Herrero, "Una aproximación multimodal al diagnóstico temporal mediante razonamiento basado en casos y razonamiento basado en modelos. aplicaciones en medicina*," Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, vol. 11, pp. 77–80, 2007.

I. Bichindaritz, "Case-based reasoning in the health sciences: What's next?" *Artificial Intelligence in Medicine*, vol. 36, no. 2, pp. 127–135, feb 2006.

HT. Wang and AU Tansel. MedCase: A Template Medical Case Store for Case-Based Reasoning in Medical Decision Support. *In Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 962-967. ACM, 2013.

L. Gierl and R. Schmidt, "CBR in medicine," in Case-Based Reasoning Technology, From Foundations to Applications. Springer-Verlag: New-York, 1998, pp. 273–298

S. Montani, "Exploring new roles for case-based reasoning in heterogeneous AI systems for medical decision support," Appl. Intell., pp. 275–285, 2007.

S. Begum, M. Uddin, P. Funk, N. Xiong and M. Folke. Case-based reasoning systems in the health sciences: a survey of recent trends and developments. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 41(4), 421-434. (2011).

J. L. Kolodner, "Maintaining organization in a dynamic longterm memory," *Cognitive Science*, vol. 7, no. 4, pp. 243–280, 1983.

S. J. Sheather *et al*., "Density estimation," *Statistical Science*, vol. 19, no. 4, pp. 588–597, 2004.

S. Bhatia, P. Prakash, and G. N. Pillai, "Svm based decision support system for heart disease classification with integer-coded genetic algorithm to select critical features," 2008.

Sundar. C, M. Chitradevi, and G. Geetharamani, "Article: Classification of cardiotocogram data using neural network based machine learning technique," *International Journal of Computer Applications*, vol. 47, no. 14, pp. 19–25, June 2012, full text available.

M. Kwiatkowska and S. Atkins, "Case representation and retrieval in the diagnosis and treatment of obstructive sleep apnea: A semiofuzzy approach," 2004.

Anexo 6. Ponencia en AUNAR DataVis Day

En la Corporación Universitaria Autónoma De Nariño (AUNAR) se realizó una ponencia sobre la investigación del presente trabajo.

CORPORACIÓN UNIVERSITARIA
AUTÓNOMA
DE NARIÑO

OTORGA EL PRESENTE
CERTIFICADO A:

Mabel Ximena Ortega Adarme

Con documento de identidad No.

PARTICIPO COMO PONENTE CON:

Sistema de razonamiento basado en casos como soporte al diagnóstico médico mediante clasificación de datos multi-clase

EN AUNAR DATAVIS DAY

Realizado en las instalaciones de AUNAR, del 25 al 27 de Agosto
Se firma en San Juan de Pasto a los 27 días del mes de Agosto de 2016

Dra. Ingreed Colunge
Rectora

Mg. Constanza Colunge
Vicerrectora Académica

Ing. Milton Cabrera Alvarez
Coordinador Programa
Ingeniería Informática

OTORGA EL PRESENTE
CERTIFICADO A:

Diana Marcela Viveros Melo

Con documento de identidad No.

PARTICIPO COMO PONENTE CON:

Sistema de razonamiento basado en casos como soporte al diagnóstico médico mediante clasificación de datos multi-clase

EN AUNAR DATAVIS DAY

Realizado en las instalaciones de AUNAR, del 25 al 27 de Agosto
Se firma en San Juan de Pasto a los 27 días del mes de Agosto de 2016

Dra. Ingreed Colunge
Rectora

Mg. Constanza Colunge
Vicerrectora Académica

Ing. Milton Cabrera Alvarez
Coordinador Programa
Ingeniería Informática

Anexo 7. Poster en ISCB-LA *(International Society for Computational Biology Latin America Bioinformatics Conference).*

Se participó en el evento ISCB-LA en Buenos Aires, Argentina con el poster titulado “Multi-class based reasoning for medical applications: An exploratory study”.

On behalf of the ISCB and A2B2C Organizing Committee, I thank you for your participation in ISCB-Latin America 2016

This document certifies that

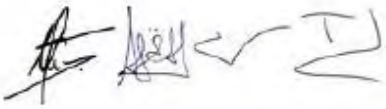
Santiago Murillo Rendón
Universidad Autónoma de Manizales

has presented

"Multi-class case-based reasoning for medical applications: An exploratory study"

At the 2016 ISCB-Latin America Conference, 21-23 November 2016.

Yours sincerely,



Fernán Agüero, Universidad de San Martín, Argentina
Gustavo Parisi, A2B2C President, Universidad de Quilmes, Argentina
Alfonso Valencia, ISCB President, Structural and Computational Biology Programme
Spanish National Cancer Research Centre (CNIO)

International Society for Computational Biology
9650 Rockville Pike
Bethesda, MD, USA 20814

Anexo 8. Artículo en revista ADCAIJ *(Advances in Distributed Computing and Artificial Intelligence Journal).*

Se publicó un artículo en la revista ADCAIJ de la Universidad de Salamanca, España.

From: **María NAVARRO CÁCERES** <revistaseusal3@gmail.com>
Date: 2017-01-17 18:02 GMT+01:00
Subject: [ADCAIJ] Editor Decision
To: Señora Xiomara Patricia Blanco Valencia <xiopepa@usal.es>

Señora Xiomara Patricia Blanco Valencia:

We have reached a decision regarding your submission to ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal, "Kernel-based framework for spectral dimensionality reduction and clustering formulation: A theoretical study".

Our decision is to: Accept Publication Under Changes.

Please, follow carefully the reviewers' recommendation attached below in order to prepare your final manuscript. Likewise, follow carefully the template of our journal.

The deadline for the final draft submission will be 28th January. Please, contact us for any question.

Yours sincerely,

María NAVARRO CÁCERES
Universidad de Salamanca, Salamanca
maria90@usal.es

Kernel-based framework for spectral dimensionality reduction and clustering formulation: A theoretical study

X. Blanco-Valencia[a], M. A. Becerra[b,c], A. E. Castro-Ospina[c], M. Ortega-Adarme[d], D. Viveros-Melo[d], J. C. Alvarado-Pérez[a,e], and D. H. Peluffo-Ordóñez[f]

[a]Universidad de Salamanca, Spain
[b]Institución Universitaria Salazar y Herrera, Colombia
[c]Research Center of the Instituto Tecnológico Metropolitano, Colombia
[d]Universidad de Nariño, Colombia
[e]Coorporación Universitaria Autónoma de Nariño, Colombia
[f]Universidad Técnica del Norte, Ecuador – dhpeluffo@utn.edu.ec

KEYWORD

Kernel PCA; Spectral clustering; Support vector machine.

ABSTRACT

This work outlines a unified formulation to represent spectral approaches for both dimensionality reduction and clustering. Proposed formulation starts with a generic latent variable model in terms of the projected input data matrix. Particularly, such a projection maps data onto an unknown high-dimensional space. Regarding this model, a generalized optimization problem is stated using quadratic formulations and a least-squares support vector machine. The solution of the optimization is addressed through a primal-dual scheme. Once latent variables and parameters are determined, the resultant model outputs a versatile projected matrix able to represent data in a low-dimensional space, as well as to provide information about clusters. Particularly, proposed formulation yields solutions for kernel spectral clustering and weighted-kernel principal component analysis.

1. Introduction

In pattern recognition, the term kernel is used to define a function that establishes the similarity among given input elements. Therefore, a kernel function enables learning methods to use similarities for representing the samples or data points, instead of using explicitly the input data matrix [Belanche Muñoz, 2013]. Kernel-based methods have been widely exploited for both supervised and unsupervised learning approaches showing their usability and versatility in several applications [Aldrich and Auret, 2013], such as image segmentation [Wu et al., 2015,Binol et al., 2017], time-varying data analysis [Langone et al., 2013] and complex data clustering [Peluffo-Ordóñez et al., 2014a], and hypothesis testing [Harchaoui et al., 2013], among others. This article explores the benefit of using a kernel model within the design of spectral formulations of clustering and unsupervised dimensionality reduction methods.

On one side, kernel methods are of interest since they allow to incorporate prior knowledge into the clustering procedure [Filippone et al., 2008]. In case of unsupervised clustering methods (that is to say, when clusters are naturally formed by following a given partition criterion), a set of initial parameters should be properly selected to avoid any local optimum solution distant from the desired global optimum. Indeed, in spectral clustering

(SC), such initial parameters are traditionally the number of clusters and the input kernel matrix itself. On the other side, the aim of dimensionality reduction (DR) is to extract a lower dimensional, relevant information from high-dimensional data, being then a key stage for the design of pattern recognition systems. Indeed, when using adequate DR stages, the system performance can be enhanced as well as the data visualization can become more intelligible [Alvarado-Pérez et al., 2015]. Recent methods of DR are focused on the data topology preservation [Peluffo-Ordóñez et al., 2014b]. Mostly such a topology is driven by graph-based approaches where data are represented by a similarity matrix, and it is then susceptible to be expressed in terms of a kernel matrix [Ham et al., 2004, Alvarez-Meza et al., 2017], which means that a wide range of methods can be set within a kernel principal component analysis (KPCA) framework [Peluffo-Ordonez et al., 2014]. At the moment to choose a method for either SC or DR, aspects such as nature of data, complexity, aim to be reached and problem to be solved should be taken into consideration. In this regard, it must be quoted that there exists a variety of spectral methods making then the selection of a method a nontrivial task. In fact, some problems may require the combination of methods so that the properties of different methods are simultaneously exploited [Peña-Unigarro et al., 2016]. Some works have studied the benefit of taking advantage simultaneously of DR and SC techniques. For instance, in [Peluffo-Ordóñez et al., 2014a], a DR approach (linear feature extraction) is used to enhance the clustering performance by performing the grouping process over the projected data rather than over the original data. Other works are focused on generating variable relevance [Wolf and Bileschi, 2005,Peluffo Ordoñez et al., 2015] or data representation [Wolf and Shashua, 2005] criteria from conventional spectral clustering formulations.

In this work, we outline a unified formulation able to explain kernel approaches for both spectral clustering (SC) and unsupervised DR. Such a formulation starts with a latent variable model of a high-dimensional representation of the input data matrix, involving implicitly a mapping function. The model is incorporated within a quadratic functional, which along with an orthonormal constraint constitutes our optimization problem being a non-supervised version of a least-square-support-vector-machine (LS-SVM) formulation. Its solution is accomplished by relaxing the problem, and following a primal-dual scheme, which readily leads to a kernel representation given the quadratic nature of the functional. The proposed formulation represents a framework to easily understand the relationship between kernel-based approaches for SC and unsupervised DR. Also, the resultant model yields explicitly the solution of two well-known methods, namely the so-called kernel spectral clustering (KSC) proposed in [Alzate and Suykens, 2010], and a version of weighted kernel PCA (WKPCA) [Peluffo-Ordonez et al., 2014].

The rest of this paper is organized as follows: Section 2 presents a brief overview on kernels. Section 3 describes our unified formulation, and explains the SC and DR perspectives. Finally, some concluding remarks are drawn in section 4.

2. Overview on kernels

For following statements, let us consider the following notation: Let $\boldsymbol{Y} \in \mathbb{R}^{D \times N}$ be the input data matrix formed by N samples (data points), denoted by $\boldsymbol{y}_i \in \mathbb{R}^D$ with $i \in \{1, \ldots, N\}$. As well, from another point of view, it is conformed by D variables such that $\boldsymbol{y}^{(\ell)} \in \mathbb{R}^N$ is the ℓ-th variable, with $\ell \in \{1, \ldots, D\}$. Mathematically, kernels involve a mapping process from a d-dimensional input space representing a data set to a (d_h-) high-dimensional space, where $d_h >> D$. In terms of pattern recognition, the advantage of mapping the original data space onto a higher one lies in the fact that the latter space may provide a better data representation regarding cluster separability. Furthermore, it must be taken into account that the mapping is done before carrying out any clustering process. Then, the success of the data clustering task can be partly attributed to the

kernel-matrix-building function when grouping algorithms are directly associated with the chosen kernel.

Currently, kernels with special structure aimed to attend particular interests have been proposed. For instance, in [Seeland et al., 2012], a structural clustering kernel is introduced by incorporating similarities induced by a structural clustering algorithm to improve graph kernels recommended by literature. Mercer kernels have been used for solving multi-cluster problems [Domeniconi et al., 2011]. In [Belanche Muñoz, 2013], different kernels (generative, convolution, and covariance kernels, among others) are explained as well as important developments on how to construct kernels from a generating function are described.

In terms of human learning theory, one of the fundamental problems is the discrimination among elements or objects. Consider the following instance: We have a set of objects formed by two different classes; then, when a new object appears the classification and/or visualization task is to determine to which class such an object belongs. This is usually done by taking into account the object's properties as well as similarities and differences with regards to the two previously known classes. According to the above, and regarding kernel theory, we need to create or choose a similarity or affinity measure to compare the data. Since such similarities are non-negative, kernel functions are positive-definite. A kernel function can be defined in the form:

$$\begin{aligned} \mathcal{K}(\cdot,\cdot) : \mathbb{K}^D \times \mathbb{K}^D &\longrightarrow \mathbb{K} \\ \boldsymbol{y}_i, \boldsymbol{y}_j &\longmapsto \mathcal{K}(\boldsymbol{y}_i, \boldsymbol{y}_j), \end{aligned} \tag{1}$$

where $\mathbb{K} = \mathbb{C}$ or $\mathbb{R}$. Note that in this case we have assumed elements $\boldsymbol{y}_i$ to be real and D-dimensional. Then, for a total of N data points, we can arrange the kernel function values into a $N \times N$ matrix $\boldsymbol{K}$ with entries $k_{ij} = \mathcal{K}(\boldsymbol{y}_i, \boldsymbol{y}_j)$, called Gram matrix or kernel matrix as well. Such a matrix is positive-semidefinite, i.e., a $N \times N$ complex matrix satisfying $\sum_{i=1}^{N}\sum_{j=1}^{N} c_i \bar{c}_j k_{ij} \geq 0$, for all $c_i \in \mathbb{C}$, being $\bar{c}_i$ the complex conjugate of c_i. Similarly, a real symmetric $N \times N$ matrix $\boldsymbol{K}$ satisfying the same condition given for all $c_i \in \mathbb{R}$ is also called positive-semidefinite. In terms of spectral matrix analysis, a symmetric matrix is positive-semidefinite if and only if all its eigenvalues are non-negative. In the literature, a number of different terms are used for positive-definite kernels, such as reproducing kernel, Mercer kernel, admissible kernel, support vector kernel, non-negative definite kernel and covariance.

2.1 Kernel trick

Now, let us consider a function to map from the D-dimensional space to that d_h dimensional one is in the form $\phi(\cdot)$, such that: $\phi(\cdot) : \mathbb{R}^D \longrightarrow \mathbb{R}^{d_h}, \boldsymbol{y}_i \longmapsto \phi(\boldsymbol{y}_i)$. The matrix $\boldsymbol{\Phi} = \left[\phi(\boldsymbol{y}_1)^\top, \ldots, \phi(\boldsymbol{y}_N)^\top\right]$, $\boldsymbol{\Phi} \in \mathbb{R}^{d_h \times N}$, is a high dimensional representation of the input data matrix $\boldsymbol{Y}$. A sagittal diagram of the mapping function is shown in Figure 1.

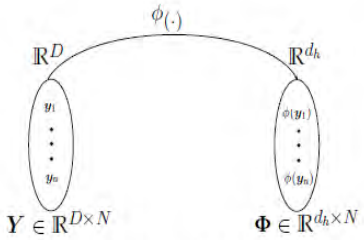


Figure 1: Mapping function to a high dimensional space.

An interesting property of the kernel functions is the so-called *kernel trick*. In topology, a kernel function can be seen as an inner product in the domain of Hilbert space $\mathcal{H}$, as follows: $\mathcal{K}(\boldsymbol{y}_i, \boldsymbol{y}_j) = \langle \phi(\boldsymbol{y}_i), \phi(\boldsymbol{y}_i) \rangle_{\mathcal{H}}$. Kernel trick allows for performing the mapping and the inner product simultaneously by defining an associated kernel function. Then, we can estimate the kernel matrix without knowing the mapping function. This property gains importance in kernel theory, since it permits to replace a positive-definite kernel with another kernel that is finite and approximately positive-definite. For instance, from a given algorithm formulated in terms of a positive-definite kernel $\mathcal{K}$, we can construct an alternative algorithm by replacing it by another positive-definite kernel $\tilde{\mathcal{K}}$ [Schölkopf and Smola, 2002], in such a manner that $\boldsymbol{\Phi}\boldsymbol{\Phi}^\top = \boldsymbol{K}$. Then, in this case, kernel trick has served to estimate $\boldsymbol{\Phi}\boldsymbol{\Phi}^\top$ as $\boldsymbol{K}$. In the domain of $\mathcal{H}$, $\boldsymbol{K}$ holds the inner product of the mapped data points (rows of matrix $\boldsymbol{\Phi}$), or -from another point of view- the outer product of the mapped variables (columns of matrix $\boldsymbol{\Phi}$).

2.2 Types of kernel functions

Radial basis function (RBF) kernels are those that can be written in terms of similarity or dissimilarity measure, in the form:

$$\mathcal{K}(\boldsymbol{y}_i, \boldsymbol{y}_j) = f(\mathrm{d}(\boldsymbol{y}_i, \boldsymbol{y}_j)), \tag{2}$$

where $\mathrm{d}(\cdot,\cdot)$ is a measure on the domain of $\boldsymbol{Y}$, in this case $\mathbb{R}^D$, so:

$$\begin{aligned} \mathrm{d}(\cdot,\cdot) : \mathbb{R}^D \times \mathbb{R}^D &\longrightarrow \mathbb{R}^+ \\ \boldsymbol{y}_i, \boldsymbol{y}_j &\longmapsto \mathrm{d}(\boldsymbol{y}_i, \boldsymbol{y}_j) \end{aligned} \tag{3}$$

and f is a function defined on $\mathbb{R}^+$. Usually, such measure arises from the inner product; $\mathrm{d}(\boldsymbol{y}_i, \boldsymbol{y}_j) = \|\boldsymbol{y}_i - \boldsymbol{y}_j\| = \sqrt{\langle \boldsymbol{y}_i - \boldsymbol{y}_j, \boldsymbol{y}_i - \boldsymbol{y}_j \rangle}$. In Table 1, some common kernels recommended by the state of the art are described.

Kernel name	Definition	Domain
lineal	$\langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle$	$\mathbb{R}^D$
Polynomial	$\langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle^D$	$\mathbb{R}^D$
Rational quadratic	$1 - \frac{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2}{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 + \sigma}, \sigma \in \mathbb{R}^+$	$\mathbb{R}^d$
Exponential	$\exp\left(-\frac{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|}{2\sigma^2}\right), \sigma \in \mathbb{R}^+$	$\mathbb{R}^D$
Gaussian	$\exp\left(-\frac{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2}{2\sigma^2}\right), \sigma \in \mathbb{R}^+$	$\mathbb{R}^D$

Table 1: Some kernel functions.

2.2.1 Special kernels

- Scaled Gaussian kernel matrix

An alternative to the Gaussian kernel is a local scaled version regarding the data point neighborhood as follows:

$$k_{ij} = \exp\left(-\frac{||\boldsymbol{y}_i - \boldsymbol{y}_j||^2}{\sigma_i \sigma_j}\right), \tag{4}$$

where σ_i is the scaling parameter defined as $\sigma_i = ||\boldsymbol{y}_i - \boldsymbol{y}_i(m)||$ being $\boldsymbol{y}_i(m)$ the m-th nearest neighbor to data point $\boldsymbol{y}_i$. The parameter m is established regarding the nature of the input data. This kernel is widely explained in [Zelnik-manor and Perona, 2004].

3. Generalized kernel formulation

This section is aimed at formulating a model and cost function for a multipurpose data representation.
To establish our model, let us consider an output data matrix $\boldsymbol{X} \in \mathbb{R}^{d \times N}$, being $d \leq D$ formed by N data points denoted by $\boldsymbol{x}_i \in \mathbb{R}^d$, with $i \in \{1, \ldots, N\}$, as well as by d variables denoted as $\boldsymbol{x}^{(\ell)} \in \mathbb{R}^N$ with $\ell \in \{1, \ldots, d\}$. Also, let us assume an orthonormal projection matrix $\boldsymbol{W} \in \mathbb{R}^{D_h \times d}$, such that $\boldsymbol{W} = [\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(d)}]$ and $\boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{I}_d$, where $\boldsymbol{w}^{(\ell)} \in \mathbb{R}^{D_h}$ and $\boldsymbol{I}_d$ is a d-dimensional identity matrix. Since $\boldsymbol{W}$ is orthonormal, elements $\boldsymbol{w}^{(\ell)}$ represent a d-dimensional base and can then generate a new space by means of a linear combination in the form: $\boldsymbol{x}^{(\ell)} = \boldsymbol{w}^{(\ell)} \boldsymbol{\Phi}$. So, the output matrix becomes $\boldsymbol{X} = \boldsymbol{W}^\top \boldsymbol{\Phi}$. Here, in order to add an offset effect, we consider a whole latent variable model as $\boldsymbol{x}^{(\ell)} = \boldsymbol{w}^{(\ell)} \boldsymbol{\Phi} + b_\ell \boldsymbol{1}_N$. Such a model can be expressed in matrix terms as:

$$\boldsymbol{X} = \boldsymbol{W}^\top \boldsymbol{\Phi} + \boldsymbol{b} \otimes \boldsymbol{1}_N^\top, \tag{5}$$

where b_l is a bias term, and $\boldsymbol{b} = [b_1, \ldots, b_d]$, $\otimes$ denotes Kronecker product, and $\boldsymbol{1}_N$ accounts for a N-dimensional all ones vector. Both PCA and SVM, in their simplest formulations, involve an energy term regarding the data matrix. Unlike conventional formulations that starts with a known input matrix, we pose a latent variable model, being unknown both variables (output and mapped data matrix) as well parameters (bias term and projection matrix). By incorporating a weighting matrix $\boldsymbol{\Delta} = \text{Diag}(\delta_1, \ldots, \delta_N)$, the energy term regarding $\boldsymbol{X}$ can be written as $\boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}^\top$. Then, a functional in terms of the generalized matrix M-norm [Peluffo Ordoñez et al., 2015] can be expressed as:

$$\frac{1}{N} \text{tr}(\boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}^\top) = ||\boldsymbol{X}||^2_{(1/N)\boldsymbol{\Delta}}. \tag{6}$$

From another point of view, if we define a weighted output data matrix as $\widetilde{\boldsymbol{X}} \in \mathbb{R}^{d \times N}$ as

$$\widetilde{\boldsymbol{X}} = \boldsymbol{X}\ \text{Diag}(\delta_1^{1/2}, \ldots, \delta_N^{1/2}), \tag{7}$$

the functional $\text{tr}(\boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}^\top)$ can also be directly seen as an energy term, so: $\text{tr}(\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{X}}^\top)$. Our model can be determined by means of a primal-dual formulation as described below.

Primal formulation: Recalling the functional given in equation (6) and the othonormality condition of projection matrix, we can write the following optimization problem:

$$\max_{\boldsymbol{X},\boldsymbol{W},\boldsymbol{b}} \quad \frac{1}{N} \text{tr}(\boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}^\top), \quad \text{s. t.} \quad \boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{I}_d, \quad \boldsymbol{X} = \boldsymbol{\Phi}\boldsymbol{W} + \boldsymbol{b} \otimes \boldsymbol{1}_N^\top, \tag{8}$$

which can be relaxed as

$$\max_{\boldsymbol{X},\boldsymbol{W},\boldsymbol{b}} \quad \frac{1}{2N} \text{tr}(\boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}^\top\boldsymbol{\Gamma}) - \frac{1}{2} \text{tr}(\boldsymbol{W}^\top \boldsymbol{W}), \text{s. t.} \quad \boldsymbol{X} = \boldsymbol{W}^\top \boldsymbol{\Phi} + \boldsymbol{b} \otimes \boldsymbol{1}_N^\top, \tag{9}$$

where $\boldsymbol{\Gamma} = \text{Diag}([\gamma_1, \ldots, \gamma_d])$ is a diagonal matrix holding regularization parameters.

Dual formulation: To solve problem (9), we form the corresponding Lagrangian, as follows:

$$\mathcal{L} = \frac{1}{2N}\,\mathrm{tr}(\boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}^\top\boldsymbol{\Gamma}) - \frac{1}{2}\,\mathrm{tr}(\boldsymbol{W}^\top\boldsymbol{W}) - \mathrm{tr}(\boldsymbol{A}^\top(\boldsymbol{X} - \boldsymbol{W}^\top\boldsymbol{\Phi} - \boldsymbol{b}\otimes\mathbf{1}_N^\top)), \tag{10}$$

where matrix $\boldsymbol{A} \in \mathbb{R}^{N\times n_e}$ holds the Lagrange multiplier vectors, that is, $\boldsymbol{A} = [\boldsymbol{\alpha}^{(1)}, \cdots, \boldsymbol{\alpha}^{(n_e)}]$, being $\boldsymbol{\alpha}^{(l)} \in \mathbb{R}^N$ the l-th vector of Lagrange multipliers. Solving the Karush-Kuhn-Tucker (KKT) conditions on (10), we get:

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{X}} = \mathbf{0} \Rightarrow \boldsymbol{X} = N\boldsymbol{\Delta}^{-1}\boldsymbol{A}\boldsymbol{\Gamma}^{-1}, \quad \frac{\partial\mathcal{L}}{\partial\boldsymbol{W}} = \mathbf{0} \Rightarrow \boldsymbol{W} = \boldsymbol{\Phi}\boldsymbol{A},$$

Therefore, by applying Lagrange multipliers and eliminating the primal variables from the initial problem (8), the following eigenvector-based dual solution is obtained:

$$\boldsymbol{A}\boldsymbol{\Lambda} = \boldsymbol{A}\boldsymbol{\Delta}(\boldsymbol{I}_N + (\mathbf{1}_N\otimes\boldsymbol{b}^\top)(\boldsymbol{K}\boldsymbol{A})^{-1})\boldsymbol{K},$$

where $\boldsymbol{\Lambda} = \mathrm{Diag}(\boldsymbol{\lambda})$, $\boldsymbol{\Lambda} \in \mathbb{R}^{N\times N}$, $\boldsymbol{\lambda} \in \mathbb{R}^N$ is the vector of eigenvalues with $\lambda_l = N/\gamma_l$, $\lambda_l \in \mathbb{R}^+$. Again, $\boldsymbol{K} \in \mathbb{R}^{N\times N}$ is a given kernel matrix, satisfying the Mercer's theorem such that $\boldsymbol{\Phi}^\top\boldsymbol{\Phi} = \boldsymbol{K}$.

In order to pose a quadratic dual formulation satisfying the condition $\boldsymbol{b}^\top\mathbf{1}_N = 0$ by centering vector $\boldsymbol{b}$ (i.e. with zero mean), the bias term is chosen in the form $b_l = -1/(\mathbf{1}_N^\top\boldsymbol{\Delta}\mathbf{1}_N)\mathbf{1}_N^\top\boldsymbol{\Delta}\boldsymbol{K}\boldsymbol{\alpha}^{(l)}$. Therefore, the solution of problem (9) is reduced to the following eigenvector-related problem:

$$\boldsymbol{A}\boldsymbol{\Lambda} = \boldsymbol{\Delta}\boldsymbol{H}\boldsymbol{K}\boldsymbol{A}, \tag{11}$$

where matrix $\boldsymbol{H} \in \mathbb{R}^{N\times N}$ is the centering matrix that is defined as $\boldsymbol{H} = \boldsymbol{I}_N - (1/(\mathbf{1}_N^\top\boldsymbol{V}\mathbf{1}_N))\mathbf{1}_N\mathbf{1}_N^\top\boldsymbol{\Delta}$. Imposing a linear independency constraint on Lagrangian vector multipliers, $\boldsymbol{A}$ might be chosen as an orthonormal matrix. In consequence, a feasible solution is to estimate $\boldsymbol{A}$ and Λ as the spectral decomposition of a centered weighted kernel matrix $\boldsymbol{\Delta}\boldsymbol{H}\boldsymbol{K}$ -eigenvector and eigenvalue diagonal matrix, respectively. Finally, the output data matrix can be calculated as follows:

$$\boldsymbol{X} = \boldsymbol{A}^\top\boldsymbol{K} + \boldsymbol{b}\otimes\mathbf{1}_N^\top. \tag{12}$$

Given this, the solution is determined by the spectrum of a centered weighted kernel matrix and a bias vector defined so that the centering condition is ensured. In the following sections, we show how this solution can be applied for both dimensionality reduction and spectral clustering.

3.1 Dimensionality reduction perspective

Latent data matrix $\boldsymbol{X}$ is given by the linear model $\boldsymbol{W}^\top\boldsymbol{\Phi} + \boldsymbol{b}\otimes\mathbf{1}_N^\top$, which clearly involves a linear combination. If we seek for a low-dimensional representation of input data $\boldsymbol{Y}$, it just suffices by estimating $\boldsymbol{X}$ with a low-rank version of $\boldsymbol{W}$. Such an estimation of the reduced matrix can be performed on the dual problem solution by using some eigenvectors from $\boldsymbol{A}$. According to the nature of the quadratic formulation, the dimensionality reduction leads to a weighted, kernelized version of PCA.

Weighted kernel PCA: Given that the optimization is done under a maximization criterion, the eigenvectors associated with the largest eigenvalues should be selected. In this sense, final dimension d indicates how many

eigenvectors are to be considered. Indeed, the eigenvalues of the centered weighted kernel defines the explained variance, so that the final dimension can be estimated with respect to it. Then, our generalized kernel model represents a weighted kernel PCA formulation when using a low-rank representation of matrix $\boldsymbol{W}$, being then able to embed a D-dimensional data matrix $\boldsymbol{Y}$ into a low-dimensional resulting matrix $\boldsymbol{X}$.

Kernel PCA: To yield the conventional kernel PCA outcomes, the initial model should be considered as a linear projection (with no bias term) in the form $\boldsymbol{X} = \boldsymbol{W}^\top \boldsymbol{\Phi}$. Since d is clearly less than d_h, a low-rank version of $\boldsymbol{\Phi}$ is then $\tilde{\boldsymbol{\Phi}} = \boldsymbol{W}\boldsymbol{X}$. So, we can write a functional to be minimized as $\frac{1}{N}||\boldsymbol{\Phi} - \tilde{\boldsymbol{\Phi}}||_F^2$, which has a dual problem given by:

$$\max_{\boldsymbol{X}} \ \text{tr}(\boldsymbol{X}^\mathsf{T}\boldsymbol{K}\boldsymbol{X}), \quad \text{s. t.} \quad \boldsymbol{X}^\top\boldsymbol{X} = \boldsymbol{I}_d, \tag{13}$$

as widely explained in [Peluffo-Ordonez et al., 2014]. Therefore, a feasible solution is when $\boldsymbol{X}$ are the eigenvectors associated with the d largest eigenvalues. As well, this formulation can be seen as a generalized Weighted PCA when using a Mahalanobis distance regarding any positive-semidefinite matrix [Peluffo-Ordóñez et al., 2014, Peluffo-Ordonez et al., 2014]. Since kernel PCA is derived under the assumption that matrix $\boldsymbol{\Phi}$ has zero mean, centering becomes necessary. To satisfy this condition, we can normalize the kernel matrix with:

$$\begin{aligned}\boldsymbol{K} \leftarrow & \boldsymbol{K} - \frac{1}{N}\boldsymbol{K}\boldsymbol{1}_N\boldsymbol{1}_N^\top - \frac{1}{N}\boldsymbol{1}_N\boldsymbol{1}_N^\top\boldsymbol{K} + \frac{1}{N^2}\boldsymbol{1}_N\boldsymbol{1}_N^\top\boldsymbol{K}\boldsymbol{1}_N\boldsymbol{1}_N^\top \\ & = (\boldsymbol{I}_N - \boldsymbol{1}_N\boldsymbol{1}_N^\top)\boldsymbol{K}(\boldsymbol{I}_N - \boldsymbol{1}_N\boldsymbol{1}_N^\top).\end{aligned} \tag{14}$$

3.2 Clustering perspective

Notice that the primal formulation given in (9) can be seen as a least-squares SVM. Then, our model should be able to provide information about the clusters immersed in data matrix. Since no supervised information is used, grouping process is fully unsupervised.

Kernel spectral clustering: Suppose that the output holds non-encoded information about centroids or prototypes for each cluster. Then, output data points should be represented in low dimension $d = K - 1$, being K the assumed number of clusters. Because each cluster is represented by a single point in the $K-1$-dimensional eigenspace, such that those single points are always in different orthants due also to the KKT conditions, we can encode the eigenvectors considering that two points are in the same cluster if they are in the same orthant in the corresponding eigenspace [Alzate and Suykens, 2010]. Then, a code book can be obtained from the rows of the matrix containing the $K-1$ binarized leading eigenvectors in the columns, by using $\text{sign}(\boldsymbol{x}^{(\ell)})$. Then, matrix $\overline{\boldsymbol{X}} = \text{sgn}(\boldsymbol{X})$ is the code book being each row a codeword. Finally, clusters are formed according to the minimal Hamming distance between codewords within the space of $\overline{\boldsymbol{X}}$. This clustering approach is so-called kernel spectral clustering (KSC), introduced in [Alzate and Suykens, 2010]. Figure 2 depicts graphically the effect of cluster assignment when using a Hamming encoding.

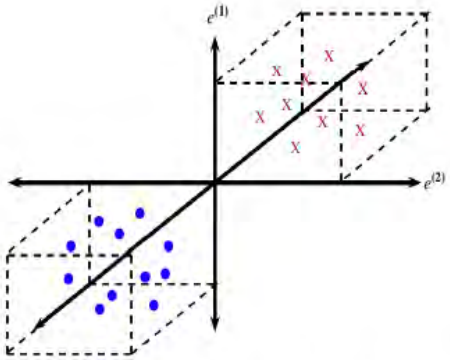


Figure 2: Encoding $\overline{\boldsymbol{E}} = \text{sign}(\boldsymbol{E})$. The example shows a two clusters problem. Since each single cluster is located in a different orthant, a feasible encoding is by using the sign function and Hamming distance becomes a proper measure to assign elements to a cluster according to the minimal distance.

Out-of-samples extension: The big advantage of this approach is that it can be extended to out-of-samples analysis without re-clustering the whole data to determine the assignment cluster membership for new testing data [Alzate and Suykens, 2010]. In particular, defining $z \in \mathbb{R}^d$ as the projection vector of a testing data point y_{test}, and by taking into consideration the training clustering model, the testing projections can be computed as

$$z = \boldsymbol{A}^\top \boldsymbol{K}_{\text{test}} + \boldsymbol{b},$$

where $\boldsymbol{K}_{\text{test}} \in \mathbb{R}^d$ is the kernel vector such that $\boldsymbol{K}_{\text{test}} = [K_{\text{test}_1}, \ldots, K_{\text{test}_N}]^\top$, being

$$K_{\text{test}_i} = K(\boldsymbol{y}_i, \boldsymbol{y}_{\text{test}}).$$

Once, the test projection vector z is computed, a decoding stage is carried out that consists of comparing the binarized projections with respect to the codewords in the code book $\overline{\boldsymbol{X}}$ and assigning cluster membership based on the minimal Hamming distance [Alzate and Suykens, 2010].

4. Conclusions

The aim of this paper is to state a generalized formulation able to explain the close relationship between spectral clustering and dimensionality reduction, within a kernel-based framework. Specifically, it has been shown that a least-square-support-vector-machine optimization problem, involving a latent variable model in terms of a high-dimensional representation of input data matrix, yields solutions containing information for encoding cluster assignment, and in turn for representing data matrix embedded in a lower-dimensional space. Furthermore, our formulation provides researchers on spectral, unsupervised pattern recognition methods with a fully matrix notation and formulation to easily understand kernel-based approaches such as KSC and KPCA.

The benefit of the proposed generalized formulation is that allows for an easy understanding of methods so that fair comparison among them as well as decision making on what methods are the most suitable ones can be readily performed. As a future work, ways to represent a wider range of (not only spectral) methods for data representation and classification (from supervised and unsupervised inferences) are to be explored and/or designed.

5. References

Aldrich, C. and Auret, L., 2013. Statistical Learning Theory and Kernel-Based Methods. In *Unsupervised Process Monitoring and Fault Diagnosis with Machine Learning Methods*, pages 117–181. Springer.

Alvarado-Pérez, J. C., Peluffo-Ordóñez, D. H., and Therón, R., 2015. Bridging the gap between human knowledge and machine learning. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 4(1):54–64.

Alvarez-Meza, A., Lee, J., Verleysen, M., and Castellanos-Dominguez, G., 2017. Kernel-based dimensionality reduction using Renyi's α-entropy measures of similarity. *Neurocomputing*, 222:36 – 46. ISSN 0925-2312. doi:http://dx.doi.org/10.1016/j.neucom.2016.10.004.

Alzate, C. and Suykens, J. A. K., 2010. Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(2):335–347.

Belanche Muñoz, L. A., 2013. Developments in kernel design. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2013), Bruges, Belgium*, pages 369–378.

Binol, H., Ochilov, S., Alam, M. S., and Bal, A., 2017. Target oriented dimensionality reduction of hyperspectral data by Kernel Fukunaga-Koontz Transform. *Optics and Lasers in Engineering*, 89:123 – 130. ISSN 0143-8166. doi:http://dx.doi.org/10.1016/j.optlaseng.2016.03.009. 3DIM-DS 2015: Optical Image Processing in the context of 3D Imaging, Metrology, and Data Security.

Domeniconi, C., Peng, J., and Yan, B., 2011. Composite kernels for semi-supervised clustering. *Knowledge and information systems*, 28(1):99–116.

Filippone, M., Camastra, F., Masulli, F., and Rovetta, S., 2008. A survey of kernel and spectral methods for clustering. *Pattern recognition*, 41(1):176–190.

Ham, J., Lee, D. D., Mika, S., and Schölkopf, B., 2004. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the twenty-first international conference on Machine learning*, page 47. ACM.

Harchaoui, Z., Bach, F., Cappé, O., and Moulines, E., 2013. Kernel-based methods for hypothesis testing: a unified view. *IEEE Signal Processing Magazine*, 30(4):87–97.

Langone, R., Alzate, C., and Suykens, J. A., 2013. Kernel spectral clustering with memory effect. *Physica A: Statistical Mechanics and its Applications*.

Peluffo-Ordóñez, D. H., Lee, J. A., Verleysen, M., Rodriguez, J. L., and Castellanos-Dominguez, G., 2014. Unsupervised relevance analysis for feature extraction and selection. In *ICPRAM 2014*, pages 310–315.

Peluffo-Ordonez, D. H., Aldo Lee, J., and Verleysen, M., 2014. Generalized kernel framework for unsupervised spectral methods of dimensionality reduction. In *2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 171–177. IEEE.

Peluffo-Ordóñez, D. H., Alzate, C., Suykens, J. A., and Castellanos-Dominguez, G., 2014a. Optimal Data Projection for Kernel Spectral Clustering. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 553–558.

Peluffo-Ordóñez, D. H., Lee, J. A., and Verleysen, M., 2014b. Recent methods for dimensionality reduction: A brief comparative analysis. In *European Symposium on Artificial Neural Networks (ESANN)*. Citeseer.

Peluffo Ordoñez, D. H., Lee, J. A., Verleysen, M., Rodriguez, J. L., Castellanos-Dominguez, G. et al., 2015. Unsupervised relevance analysis for feature extraction and selection. A distance-based approach for feature relevance. In *3rd International Conference on Pattern Recognition Applications and Methods (ICPRAM 2014)*.

Peña-Unigarro, D. F., Salazar-Castro, J. A., Peluffo-Ordóñez, D. H., Rosero-Montalvo, P. D., Oña-Rocha, O. R., Isaza, A. A., Alvarado-Pérez, J. C., and Theron, R., 2016. Interactive visualization methodology of high-dimensional data with a color-based model for dimensionality reduction. In *2016 XXI Symposium on*

Signal Processing, Images and Artificial Vision (STSIVA), pages 1–7. doi:10.1109/STSIVA.2016.7743318.

Schölkopf, B. and Smola, A. J., 2002. *Learning with Kernels*.

Seeland, M., Karwath, A., and Kramer, S., 2012. A structural cluster kernel for learning on graphs. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 516–524. ACM.

Wolf, L. and Bileschi, S., 2005. Combining variable selection with dimensionality reduction. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 801–806. IEEE.

Wolf, L. and Shashua, A., 2005. Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. *Journal of machine learning*, 6:1855 – 1887.

Wu, Y., Ma, W., Gong, M., Li, H., and Jiao, L., 2015. Novel Fuzzy Active Contour Model With Kernel Metric for Image Segmentation. *Applied Soft Computing*.

Zelnik-manor, L. and Perona, P., 2004. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems 17*, pages 1601–1608. MIT Press.

Anexo 9. Página Web

Dentro del desarrollo de este proyecto, uno de los productos esperados era una diseñar una página web en la que se pueda encontrar información relacionada a la interfaz, algoritmos, ejecutables, datos y otros productos adicionales como los artículos y videos. La página web fue creada en Google Sites y se puede acceder mediante el siguiente vínculo:

https://sites.google.com/site/degreethesisdiegopeluffo/case-base-reasoning-system-for-medical-aplications-1

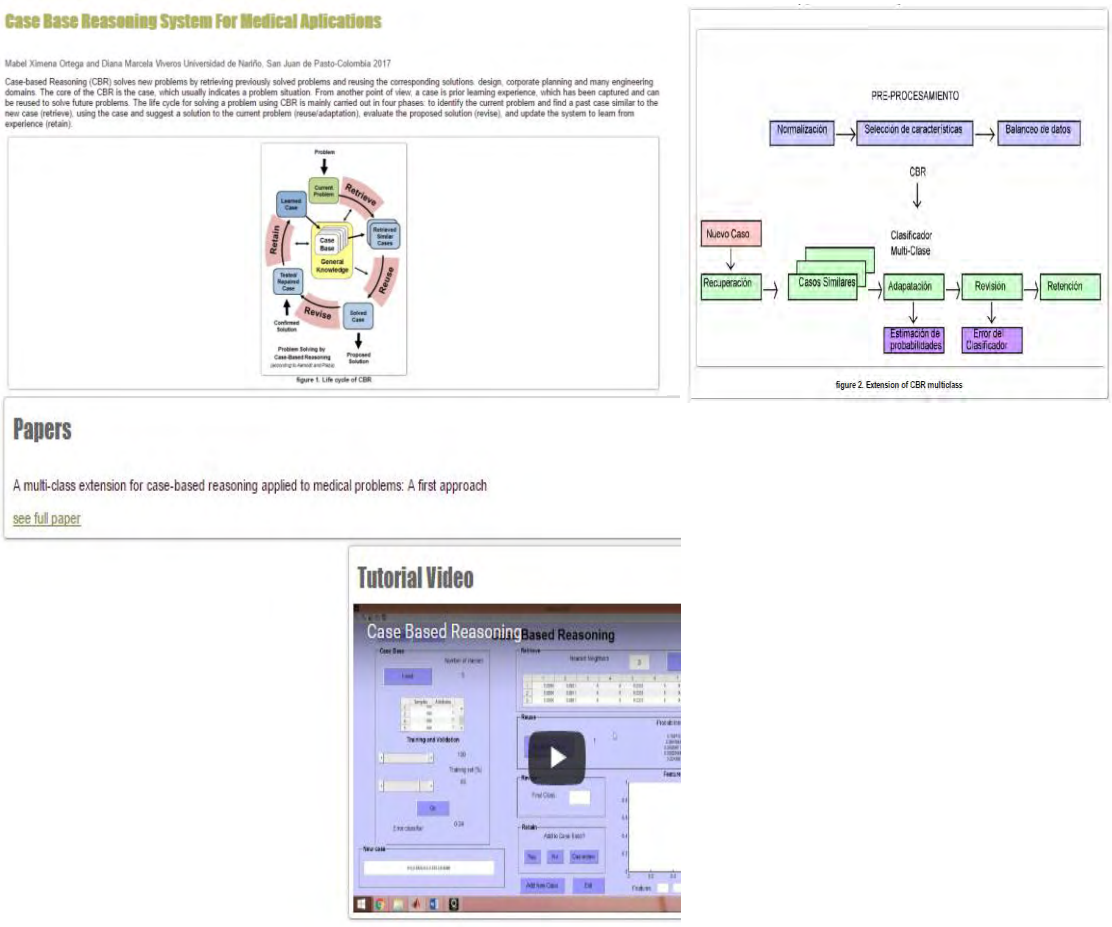


Figura 42. Diseño de la página web en donde se puede encontrar una amplia información acerca de la interfaz desarrollada así como scripts, tutoriales y manuales.

Anexo 10. Versión extendida del artículo "*A multi-class extension for case-based reasoning applied to medical problems: A first approach*". Para la revista científica Enfoque UTE (En proceso de evaluación).

Enfoque UTE, Artículo sometido a revisión
http://ingenieria.ute.edu.ec/enfoqueute/
e-ISSN: 1390-6542 / p-ISSN: 1390-9363
Recibido (Received): 2015/xx/yy
Aceptado (Accepted): Pendiente

Razonamiento basado en casos aplicado al diagnóstico médico utilizando clasificadores multi-clase: Un estudio preliminar

Case based reasoning applied to medical diagnosis using multi-class classifier: A preliminary study

D. Viveros-Melo [1], M. Ortega-Adarme [2], X. Blanco Valencia [3], A. E. Castro-Ospina [4]
S. Murillo Rendón [5], D. H. Peluffo-Ordóñez [6]

Resumen:
El razonamiento basado en casos (CBR) es un proceso utilizado para el procesado de los ordenadores que trata de imitar el comportamiento de un experto humano en la toma de decisiones con respecto a un tema y aprender de la experiencia de casos pasados. CBR ha demostrado ser apropiado para trabajar con datos de dominios poco estructurados o situaciones donde es difícil la adquisición de conocimiento, como es el caso del diagnóstico médico, donde es posible identificar enfermedades como: cáncer, predicción de epilepsia y diagnóstico de apendicitis. Algunas de las tendencias que se pueden desarrollar para CBR en la ciencia de la salud están orientadas a reducir el número de características en datos de gran dimensión. Una contribución importante puede ser la estimación de probabilidades de pertenencia a cada clase para los nuevos casos. Con el fin de representar adecuadamente la base de datos y evitar los inconvenientes causados por la alta dimensión, ruido y redundancia de los mimos, en este trabajo, se utiliza varios algoritmos en la etapa de pre-procesamiento para realizar una selección de variables y reducción de dimensiones. Además, se realiza una comparación del rendimiento de algunos clasificadores multi-clase representativos para identificar el más eficaz e incluirlo en un esquema CBR. En particular, se emplean cuatro técnicas de clasificación y dos técnicas de reducción para hacer un estudio comparativo de clasificadores multi-clase sobre CBR

Palabras clave: Razonamiento basado en casos; alta dimensionalidad; selección de variables.

Abstract:
Case-based reasoning (CBR) is a process used for computer processing that tries to mimic the behavior of a human expert in making decisions regarding a subject and learn from the experience of past cases. CBR has demonstrated to be appropriate for working with unstructured domains data or difficult knowledge acquisition situations, such as medical diagnosis, where it is possible to identify diseases such as: cancer diagnosis, epilepsy prediction and appendicitis diagnosis. Some of the trends that may be developed for CBR in the health science are oriented to reduce the number of features in highly dimensional data. An important contribution may be the estimation of probabilities of belonging to each class for new cases. In this paper, in order to adequately represent the database and to avoid the inconveniences caused by the high dimensionality, noise and redundancy, a number of algorithms are used in the preprocessing stage for performing both variable selection and dimension reduction procedures. Also, a comparison of the performance of some representative multi-class classifiers is carried out to identify the most effective one to include within a CBR scheme. Particularly, four classification techniques and two reduction techniques are employed to make a comparative study of multi-class classifiers on CBR.

[1,2] Universidad de Nariño, Pasto – Colombia (dianavive.77@udenar.edu.co, mabel12-02@udenar.edu.co)
[3] Universidad de Salamanca, Salamanca – España (xiopepa@usal.es)
[4] Tecnológico Metropolitano, Medellín – Colombia (andrescastro@itm.edu.co)
[5] Universidad Autónoma de Manizales, Manizales – Colombia (smurillo@autonoma.edu.co)
[6] Universidad Técnica del Norte, Ibarra – Ecuador (dhpeluffo@utn.edu.ec)

Keywords: Case based reasoning; High dimensionality; Variable selection.

1. Introduction

Case-based Reasoning (CBR) solves new problems by retrieving previously solved problems and reusing the corresponding solutions. The specificity of the case-based approach of reasoning lies in its focus on the inseparability of reasoning from memory and from learning (Bichindaritz & Conlon, 1996). In CBR terminology, a case usually denotes a problem situation. A previously experienced situation, which has been captured and learned in a way that it can be reused in the solving of future problems, is referred to as a past case, previous case, stored case, or retained case. Correspondingly, a new case or unsolved case is the description of a new problem to be solved. Case-based reasoning is in effect a cyclic and integrated process of solving a problem, learning from this experience, solving a new problem, etc. (Aamodt & Plaza, 1994).

The common life cycle for solving a problem using CBR is mainly carried out in four-step process (T rendowicz & Jeffery, 2014).

1. Retrieve: One or more problems that are similar to the new problem are retrieves from the base of previously solved problems, and one attempts to modify them to fit the new problem parameters.
2. Reuse: The solutions of the selected previous problems are reused to solve the new one.
3. Revise: The solved new problem is then revised against the actual solution.
4. Retain: When successfully tested, it is added to the base of previous problems to be reused for solving future problems.

In particular, CBR has demonstrated to be an appropriate methodology for:

- Working with unstructured domains data or difficult knowledge acquisition situation, for example, many diseases are not well understood by formal models or universally applicable guidelines (Herrero, 2007), (Bichindaritz & Marling, 2006).

- Help Desk systems used in the area of customer service to solve problems with products or services (Jenal, Gonzales, Alejo, & Ramos López, 2006).

- Predictions of the possible success of a proposed solution when the information is stored taking into account the level of success of the solutions, the case-based reasoner may be able to predict proposed solution to the current problem. Clearly, the reasoner will have in not only those levels of success stored but also the differences between the Case or cases recovered and the current situation (Lozano & Fernández, 2008).

- Automatic Acquisition of Subjective Knowledge because CBR systems exhibit an incremental knowledge acquisition, and knowledge can be abstracted by generalizing cases ((Phuong, Hoang, Prasad, Hung, & Drake, 2001).

- Medical diagnosis based on the similarity of the symptoms of the base of cases with the current case select the one that best suits to make a diagnosis ((Jenal, Gonzales, Alejo, & Ramos López, 2006).

Among these tasks, medical diagnosis has been one of the most popular research subjects in both medical informatics and computer science communities. Medical diagnosis is the process aiming at identifying diseases based on findings (such as symptoms, lab reports, patient's complaints, and other environmental factors) Adapting computer aided decision support systems to the diagnosis process requires a database-like medical knowledge base, and a problemsolving strategy applied to the knowledge base. A single problem solving strategy may be sufficient for simple diagnosis problems However, some difficulties may arise when the diagnosis problem becomes complex (Wang, Hsien-Tseng, & Tansel, 2013).

So, the CBR, is a reasoning process, which is medically accepted and also getting increasing attention from the medical domain. A number of benefits of applying CBR in the medical domain have already been identified (Bichindaritz & Marling, 2006), (Gierl, Bull, & Schmidt, 1998), (Montani, 2008). However, the medical applications offer a number of challenges for the CBR researchers and drive advances in research (Begum, Ahmed, Funk, Xiong, & Folke, 2011).

In order to adequately represent data and to avoid the inconveniences caused by its high dimensionality, we propose the use of variable selection and dimension reduction techniques in a preprocessing stage for CBR tasks, finally, we make a comparative study of multi-class classifiers to assess processed data performance.

The rest of this paper is structured as follows: Section II describes the proposed methodology, as well as the pattern recognition procedures used in this work. Section III presents the proposed experimental setup. Results and discussion are gathered in section IV. Finally, some concluding remarks and future works are drawn in Section V.

2. Material and Methods

This section outlines the proposed framework to assess the feasibility of using multi-class schemes within CBR approaches. Particularly, we resort to the adaptation of a pattern recognition stages into the CBR life cycle.

In the CBR scheme, the recovery is the most important stage, since in this phase the system finds the most similar cases to the current unknown case, simulating an efficient memory as a human

expert would (Kolodner, 1983). By combining the CBR methodology with classifiers, a cost function would be used to find the nearby cases.

The next stage where we adapt classifiers would be in the adaptation stage, because we want to show the answer in terms of probabilities. With the classifier we can find the membership degree of the new case in each of the classes, which would be helpful for medical staff.

To that end, we propose to carry out a comparative study of multi-class classifiers within preprocessing, recovery and adaptation CBR stages. Fig. 1 depicts the proposed methodology to perform the comparison of multi-class classifiers.

A. *Preprocessing*

Variable selection: First, as preprocessing stage a variable selection procedure is employed. In this work, we use the so-called correlation based feature subset (CfsSubsetEval) algorithm, which evaluates the relevance of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy among them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred. And as search method the bestfirst algorithm, to reduce the number of parameters per instance of a dataset with a backtracking. It starts with the whole set of attributes and search backward to reduce the number of parameters per instance of a dataset.

Dimensionality Reduction: After performing variable selection and aiming to improve both visual inspection and classification performance, a dimensionality reduction stage is employed by using well known methods, namely Laplacian Eigenmaps (LE), that uses spectral techniques to perform dimensionality reduction. This technique relies on the basic assumption that the data lies in a low-dimensional manifold in a high-dimensional space [50] and t-distributed stochastic neighbor embedding (t-SNE), it is a nonlinear dimensionality reduction technique that is particularly well-suited for embedding high-dimensional data into a space of two or three dimensions, which can then be visualized in a scatter plot.

B. *Adaptation and recovery*

Here, with the aim of accomplishing a multi-class case recovery, representative multi-class classifiers are considered. Due to their characteristics, we select the following classifiers: *K* Nearest Neighbor Classifier (*K*-NN) being a geometric-distance-based-approach, artificial neural networks (ANN) being a heuristic-search-based approach, support vector machines (SVM) being a model-based classifier, and Parzen's Classifier (PC) being a non-paramatric density-based classfier.

3. Experimental Setup

A. *Database*

For evaluating the proposed methodology, we used two databases from UCI Machine Learning Repository. The first one, named Cardiotocograms, contains 2126 fetal cardiotocograms belonging to different classes. The dataset consists of measurements of fetal heart rate (FHR) and uterine contraction (UC) features on cardiotocograms classified by expert obstetricians. This data set consists of 21 attributes which include LB - FHR baseline (beats per minute), AC of accelerations per second, FM of fetal movements per second, UC of uterine contractions per second, DL of light decelerations per second, DS of severe decelerations per second, DP of prolongued decelerations per second, ASTV percentage of time with abnormal short term variability, MSTV mean value of short term variability, ALTV percentage of time with abnormal long term variability, MLTV mean value of long term variability, Width width of FHR histogram, Min minimum of FHR histogram, Max Maximum of FHR histogram, Nmax of histogram peaks, Nzeros of histogram zeros, Mode - histogram mode, Mean histogram mean, Median histogram median, Variance histogram variance, Tendency histogram tendency, CLASS FHR pattern class code (1 to 10) and NSP fetal state class code (Normal=1; Suspect=2; Pathologic=3).

The second database, named Cleveland, contains 303 instances. The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4. Experiments with the Cleveland database have concentrated on simply attempting to distinguish presence (values 1,2,3,4) from absence (value 0). Consisting of 13 attributes which include age, sex, chest pain type (Typical angina=1; Atypical angina= 2; Non-anginal pain=3; Asymptomatic=4), resting blood pressure, cholesterol, fasting blood sugar (True=1; False=0), resting electrocardiographic results (Normal=1; Having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)=2; Showing probable or definite left ventricular hypertrophy by Estes' criteria=3), maximum heart rate, exercise induced angina (Yes=1; No=0), oldpeak, slope (Upsloping=1; Flat=2; Downsloping=3), number of vessels coloured, thal (Normal=3; Fixed defect=6; Reversable defect=7) and the classification values from 0 no presence to 4 types of heart diseases.

B. Parameter settings and procedures

As outcomes of the preprocessing stage, we obtain that Cardiotocograms database is reduced to 10 features, and Cleveland database to 7 features. Subsequently, as part of the same stage, by using dimensionality reduction techniques Cardiotocogram database is reduced to a 2-, 3-, 5-, 8-dimensional space. Likewise, Cleveland database is reduced to 2-, 3-, 5-dimensional space. As well, the whole subset of selected variables is considered for both databases.

For classification techniques, it should be stated out that a 20-fold cross-validation was performed to achieve unbiased results. Particularly, the following setup is established:

- *K-NN:* Is a nonparametric supervised classification method based on distances. This instance-based classification technique needs a value for the number of neighbors (K), such parameter is optimized by means of a leave-one-out strategy, is K-fold cross validation taken to its logical extreme, with K equal to N, the number of data points in the set. That means that N separate times, the approximate function is trained on all the data except for one point and a prediction is made for that point.

- *ANN*: The heuristic-based classification technique requires a number of units per hidden layer. In this work, a back-propagation trained feed-forward neural net is used with a single hidden layer. The number of units is computed from the data itself as the half of the instances divided by feature size plus the number of classes. The weight initialization consists of setting all weights to be zero, as well as the dataset is used as a tuning set.

- *SVM*: This instance-based classification method takes advantage of the kernel trick to compute the most discriminative non-linear hyperplane between classes. Therefore, its performance heavily depends on the selection and tuning of the kernel type. For this work a Gaussian kernel is selected given its ability of generalization and its band-width parameter was fixed by the Silverman's rule (Sheather, 2004).

- *PC*: This probabilistic-based classification method requires a smoothing parameter for the Gaussian distribution computation, which is optimized.

As a performance measure, it is used the standard mean classification error.

4. Results and Discussion

Achieved results for different number of dimensions as well as different classifiers are shown in Table I as the mean and standard deviation over the 20 folds runs. It can be seen how Cleveland dataset is a challenge task since performance is poor for all classifiers. It should be stated also that dimensionality reduction does not necessarily improves classification performance for both dimensionality reduction techniques. Nevertheless, by reducing dimensionality there is a gain in visual analysis of data as can be appreciated in Figure 1, particularly it can be seen how in 2D (Figures 2(a) and 2(c)) and 3D (Figures 2(b) and 2(d)) Cleveland data is highly overlapped which is consistent with achieved results. It should be noted that the error for SVM classifier is 0.397 ± 0.07, which is not far from the result obtained in (Bhatia, Praveen , & Pillai, 2008), where the classification accuracy with 7 attributes is of 70.36%.

TABLE I
ACHIEVED CLASSIFICATION PERFORMANCE OVER 20-FOLD CROSS VALIDATION FOR CONSIDERED DATABASES AND DIMENSIONALITY REDUCTION TECHNIQUES

DB	Reduction Technique	# dimd	*K*-NN	ANN	SVM	PC
Cleveland	t-SNE	2	0.381 ± 0.08	0.389 ± 0.067	0.389 ± 0.013	0.393 ± 0.093
		3	0.382 ± 0.06	0.367 ± 0.09	0.389 ± 0.013	0.393 ± 0.069
		5	0.397 ± 0.07	0.362 ± 0.089	0.389 ± 0.028	0.4 ± 0.087
		7	0.397 ± 0.07	0.347 ± 0.062	0.401 ± 0.029	0.393 ± 0.069
	LE	2	0.408 ± 0.069	0.393 ± 0.077	0.389 ± 0.013	0.393 ± 0.041
		3	0.397 ± 0.066	0.397 ± 0.075	0.389 ± 0.013	0.374 ± 0.047
		5	0.389 ± 0.067	0.404 ± 0.085	0.412 ± 0.036	0.389 ± 0.067
		7	0.389 ± 0.065	0.382 ± 0.065	0.397 ± 0.07	0.404 ± 0.064
Cardiotocograms	t-SNE	2	0.037 ± 0.015	0.084 ± 0.038	0.071 ± 0.017	0.077 ± 0.017
		3	0.036 ± 0.016	0.073 ± 0.02	0.054 ± 0.019	0.076 ± 0.018
		5	0.032 ± 0.017	0.088 ± 0.017	0.039 ± 0.019	0.075 ± 0.016
		8	0.035 ± 0.016	0.079 ± 0.016	0.033 ± 0.017	0.075 ± 0.019
		10	0.031 ± 0.017	0.082 ± 0.036	0.028 ± 0.016	0.076 ± 0.019
	LE	2	0.045 ± 0.014	0.078 ± 0.016	0.086 ± 0.017	0.102 ± 0.023
		3	0.054 ± 0.018	0.072 ± 0.015	0.061 ± 0.016	0.09 ± 0.02
		5	0.042 ± 0.014	0.075 ± 0.031	0.048 ± 0.014	0.09 ± 0.016
		8	0.039 ± 0.015	0.067 ± 0.019	0.038 ± 0.013	0.065 ± 0.016
		10	0.381 ± 0.25	0.06 ± 0.017	0.038 ± 0.016	0.063 ± 0.016

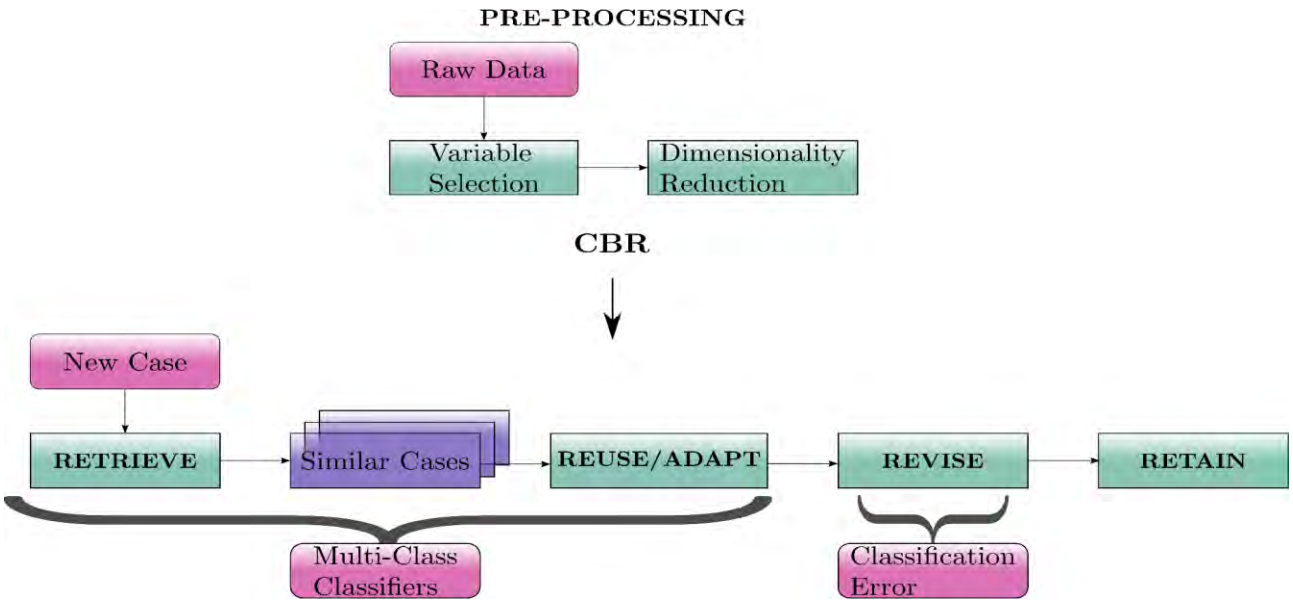


Figure 1. Block diagram of proposed methodology. The aim of the comparative study is assessing the possibility of incorporating multi-class classifiers into CRB approaches design, as well as identifying the best classifier for this task

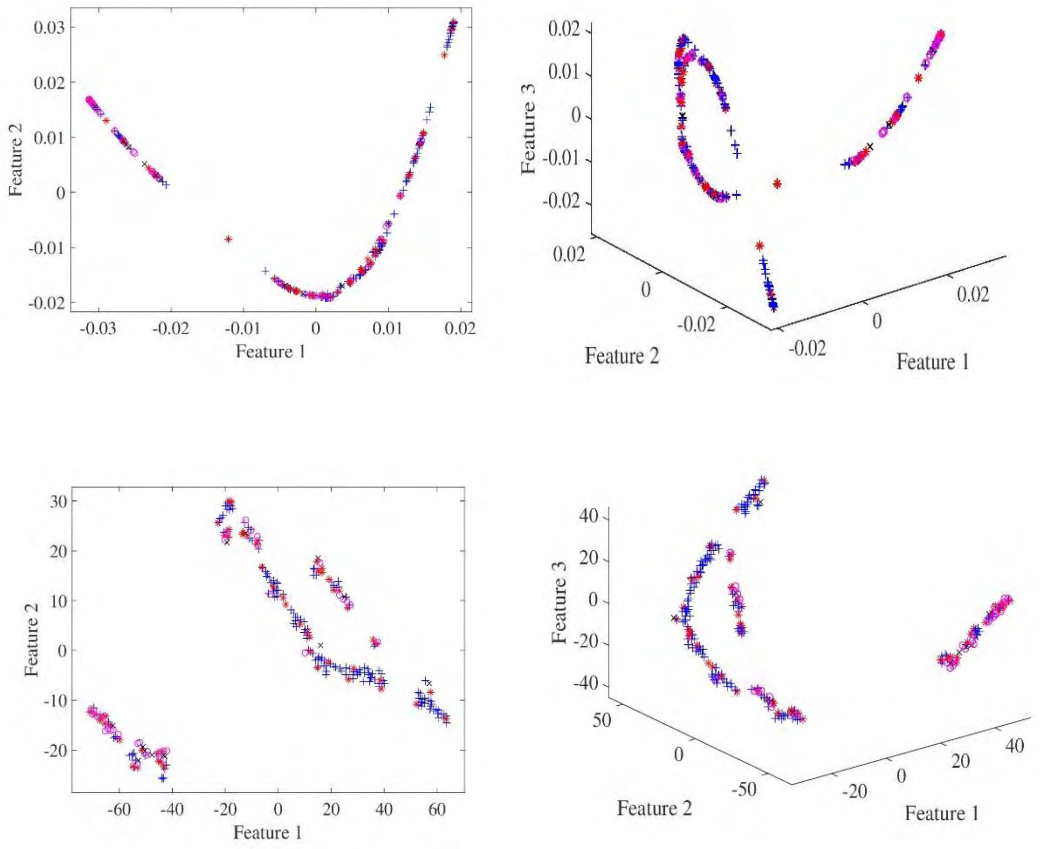


Figure 2. Low-dimensional scatterplots for Cleveland database. Figures (a), (c) show the first two features from database. Figures (b), (d) show the first three features from database.

For Cardiotocograms dataset classes separability is evident in lower dimensions, i.e. 2D and 3D, as depicted in Figures 3(a) to 3(d) leading to outstanding results as shown in Table I, however, as for Cleveland dataset, dimensionality reduction does not substantially improves classification performance on Cardiotocograms dataset even though it enhances data visualization. We can see that for the Cardiotocograms database the best result was using the SVM classifier the error is 0.028 ± 0.016, improving the results obtained in (Sundar, Chitradevi, & G. , 2012), where they achieved an average accuracy of 0.9328.

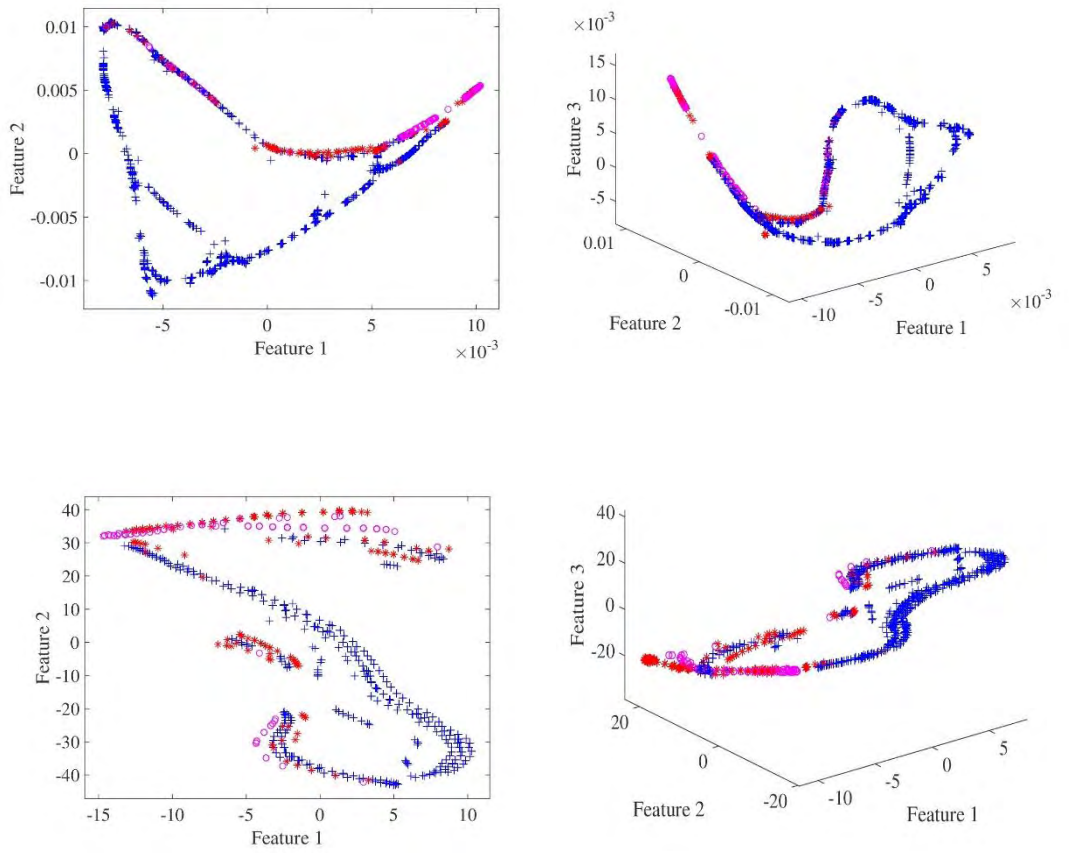


Figure 3. Low-dimensional scatterplots for Cardiotocograms database. Figures (a), (c) show the first two features from database. Figures (b), (d) show the first three features from database.

By performing a stability assessment, it could be seen from Figure 4 by the width of the error boxplots how SVM and *K*-NN classifiers achieves the best results for considered Cardiotocogram and Cleveland databases. Moreover, it should be noted how SVM classification results are the most stable of the considered classification techniques.

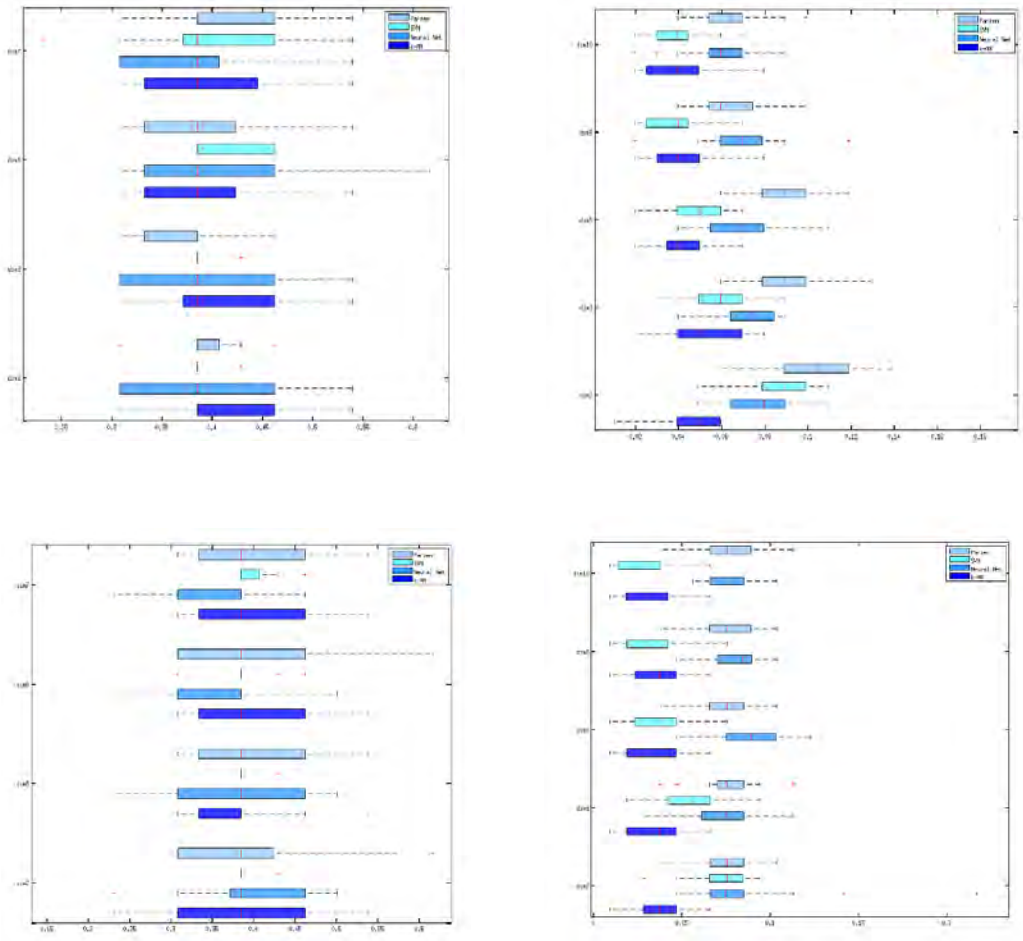


Figure 4. Classification error boxplots for considered classification techniques on Cleveland and Cardiotocograms databases

5. Conclusions and future work

This work presents a feasibility evaluation of the use of techniques from the field of pattern recognition into CBR frameworks, so that conventional CBR can be extended to multi-class scenarios. The above, with the aim of facilitating decision making in the diagnostic support, especially in situations where subcategories may exist and even emerging categories according to the condition of the patients.

Experimentally we prove that the SVM classifier is a good candidate for integration with the CBR approach to create a generic system to assist physicians in the diagnosis of patients and is capable of working with databases multiclass associating probabilities each class, responding to one of the challenges of [51], [52].

As a future work, we will explore the possibility to design a case recovery stage for CBR able to deal with multi-class cases while providing users with class membership (probabilities to belong) estimates for a new case, improving usability with respect to conventional approaches and, in addition, providing more meaningful information to the expert at the review stage in order to provide a more accurate diagnosis.

To improve the performance of the classifiers, we will study new techniques of variable selection and reduction of dimensions, as well as techniques of data balancing for databases that require it as is the case of Cleveland, due to problems such as class overlapping or the lack of data representative of input data.

Acknowledgments

Authors would like to thank to the Facultad de Ingeniería en Ciencias Aplicadas as well as electronic engineering and telecommunications program from Universidad Técnica del Norte.

References

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications 7*, 39-59.

Begum, S., Ahmed, M. U., Funk, P., Xiong, N., & Folke, M. (2011). Case-based reasoning systems in the health sciences: a survey of recent trends and developments. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 421-434.

Belkin, M. (2003). Problems of learning on manifolds. *The University of Chicago*.

Bhatia, S., Praveen , P., & Pillai, G. (2008). SVM based decision support system for heart disease classification with integer-coded genetic algorithm to select critical features. En *Proceedings of the World Congress on Engineering and Computer Science, WCECS* (págs. 22-24).

Bichindaritz, I. &. (1996). Temporal knowledge representation and organization for case-based reasoning.

Bichindaritz, I., & Conlon, E. (1996). Temporal knowledge representation and organization for case- based reasoning. *Temporal Representation and Reasoning, 1996.(TIME'96), Proceedings., Third International Workshop on*, 152-159.

Bichindaritz, I., & Marling, C. (2006). Case-based reasoning in the health sciences: What's next? *Artificial intelligence in medicine*, 127-135.

Gierl, L., Bull, M., & Schmidt, R. (1998). CBR in Medicine. En *Case-Based Reasoning Technology*

(págs. 273-297). pringer Berlin Heidelberg.

Herrero, J. M. (2007). *Una aproximación multimodal al diagnostico temporal mediante razonamiento basado en casos y razonamiento basado en modelos. Aplicaciones en la medicina.*

Jenal, Gonzales, M., Alejo, S. M., & Ramos López, R. (2006). Sistema CBR para presentación de entrenamientos físicos personalizados en Internet.

Kolodner, J. (1983). Maintaining organization in a dynamic long-term memory. *Cognitive science 7*, 243-280.

Kwiatkowska, M., & Atkins, M. (2004). Case representation and retrieval in the diagnosis and treatment of obstructive sleep apnea: a semio-fuzzy approach. En *Proceedings of the 7th European Conference on Case-Based Reasoning* (págs. 5-35).

Lozano, L., & Fernández, J. (2008). Razonamiento basado en casos: Una visión general.

Montani, S. (2008). Exploring new roles for case-based reasoning in heterogeneous AI systems for medical decision support. *Applied Intelligence 28*, 275-285.

Phuong, Hoang, N., Prasad, N., Hung, D. H., & Drake, J. (2001). pproach to combining case based reasoning with rule based reasoning for lung disease diagnosis. En *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th* (págs. 883-888). IEEE.

Sheather, S. (2004). Density estimation. *Statistical Science 19*, 588-597.

Sundar, C., Chitradevi, M., & G. , G. (2012). Classification of cardiotocogram data using neural network based machine learning technique. *International Journal of Computer Applications 47*.

Trendowicz, A., & Jeffery, R. (2014). *Software project effort estimation: Foundations and best practice guidelines for success.* Springer.

Wang, Hsien-Tseng, & Tansel, A. U. (2013). MedCase: a template medical case store for case- based reasoning in medical decision support. En *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (págs. 962-967). ACM.