Métodos para la selección de prototipos

por

M. C. José Arturo Olvera López

Tesis sometida como requisito parcial
para obtener el grado de

DOCTOR EN CIENCIAS EN LA ESPECIALIDAD DE CIENCIAS COMPUTACIONALES

en el

Instituto Nacional de Astrofísica, Óptica y Electrónica
Marzo 2009
Tonantzintla, Puebla

Supervisada por:

Dr. José Francisco Martínez Trinidad
Investigador titular del INAOE

Dr. Jesús Ariel Carrasco Ochoa
Investigador titular del INAOE

Resumen

En reconocimiento de patrones, los clasificadores supervisados asignan una clase a nuevos objetos o prototipos. Para llevar a cabo este proceso se usa un conjunto de entrenamiento, mediante el cual se proporciona información al clasificador durante su etapa de entrenamiento. En la práctica, no toda la información en los conjuntos de entrenamiento es útil, por lo que es necesario descartar algunos prototipos del conjunto de entrenamiento. A este proceso se le denomina selección de prototipos, la cual corresponde al área en que se ubica el trabajo de investigación de esta tesis.

Mediante la selección de prototipos se reduce el tamaño de un conjunto de entrenamiento y como consecuencia, se reducen los tiempos de ejecución en los procesos de clasificación y/o entrenamiento con una calidad de clasificación aceptable con respecto a la obtenida con los conjuntos originales de entrenamiento. Siendo ésta la principal utilidad de la selección de prototipos.

Se han propuesto diversos métodos para la selección de prototipos, varios de ellos presentan un buen desempeño pero la selección está fuertemente ligada al uso de un clasificador particular, por lo que, cuando se requieren utilizar otros clasificadores, el desempeño de estos métodos se ve afectado. Otra de las características que presentan los métodos del estado del arte es que el tiempo requerido por éstos para llevar a cabo la selección crece cuando el conjunto de entrenamiento es grande, lo cual provoca que sean métodos costosos y, en algunas ocasiones, inaplicables.

La contribución de este trabajo son métodos para la selección de prototipos que solucionan las limitantes de algunos de los métodos existentes, tales como altos tiempos de ejecución y la dependencia del uso de algún clasificador particular en el desempeño de los métodos de selección. En particular, se

proponen cuatro métodos para la selección de prototipos; dos de ellos se basan en la búsqueda secuencial y los restantes en la selección de prototipos borde mediante agrupamientos y relevancia de prototipos, respectivamente.

De acuerdo a los experimentos realizados y resultados obtenidos, los métodos propuestos presentan una solución al problema de la selección de prototipos considerando las limitantes en los métodos relevante existentes. Dos de estos métodos llevan a cabo la selección en un tiempo mucho menor con respecto a otros métodos para el caso específico de grandes conjuntos de datos.

Abstract

In Pattern Recognition, the supervised classifiers assign a label or class to unseen objects or prototypes. For classifying new prototypes a set of prototypes called training set is used, this set provides useful information to the classifiers during the training stage. In practice, not all the information in the training set is useful so it is possible to discard irrelevant prototypes from the training set. This process is known as prototype selection and it is the main topic of this research.

Through prototype selection the training set size is reduced which allows reducing the runtimes in the classification and/or training stages of the classifiers with acceptable classification accuracy, which is the purpose of the prototype selection.

In the literature, several methods have been proposed for selecting prototypes however, their performance is strongly related to the use of a specific classifier and when different classifiers are used, the performance of these methods decreases. In addition, most of the methods spend long time selecting prototypes when large datasets are processed and in some cases, they cannot be applied.

The contribution of this research are four methods for selecting prototypes which solve drawbacks of some methods in the state of the art. The first two methods are based on the sequential search and the remaining methods uses clustering and prototypes relevance for selecting border prototypes.

According to the results reported in this work, the proposed methods are a good option for solving the prototype selection problem. In addition, two of the proposed methods are faster than other methods from the state of the art mainly in the large-training sets case.

Agradecimientos

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo proporcionado para y durante la realización de este trabajo de tesis y también al Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) por permitirme desarrollar en sus instalaciones este trabajo de investigación.

Agradezco de manera especial a dos excelentes guías durante la elaboración de esta investigación, a quienes día a día admiro por su calidad humana y profesional: Dr. José Francisco Martínez Trinidad y Dr. Jesús Ariel Carrasco Ochoa, cuya asesoría fue indispensable en esta tesis doctoral.

Quiero agradecer a: Dr. Josef Kittler, Dr. Jesús Antonio González Bernal, Dr. Eduardo Morales Manzanares, Dr. Carlos Alberto Reyes García y Dr. Luis Enrique Sucar Succar por su tiempo, observaciones y sugerencias realizadas durante el proceso de revisión de este trabajo.

A mis padres y hermanos,

Por su amor, apoyo y presencia durante toda mi vida.

Lista de figuras

Lista de tablas

Contenido

Capítulo 1

Introducción

En este capítulo se presenta una breve introducción referente a este trabajo de investigación. Se describe el área en que se sitúa el problema a resolver, así como un panorama general de la motivación que da pauta a la solución propuesta y el objetivo general de este trabajo de investigación.

1.1 Conceptos preliminares

En Reconocimiento de Patrones, la clasificación supervisada es un proceso mediante el cual se determina la clase a la cual pertenece un prototipo. En el contexto de este trabajo, un prototipo es un conjunto de datos descriptivos, por ejemplo, un registro en una base de datos. La determinación de la clase de los prototipos se lleva a cabo con base en las características descriptivas o atributos de éstos, para lo cual, se requiere de un conjunto de datos previamente etiquetada que se proporciona a los clasificadores. A este conjunto de información se le llama comúnmente conjunto de entrenamiento y contiene información descriptiva de cada uno de sus elementos (prototipos) así como de las clases a las que pertenecen. La muestra o conjunto de entrenamiento, es la base de los clasificadores supervisados para construir sus modelos y clasificar a los nuevos prototipos que se presenten.

Cuando un nuevo prototipo p se presenta para ser clasificado, el objetivo del clasificador es determinar (a partir de la información proporcionada por el conjunto de entrenamiento) la clase o etiqueta que se asignará a p. Una parte importante para el buen desempeño del clasificador es la calidad del conjunto de entrenamiento. Cuando la cantidad de prototipos de una muestra es grande, el tiempo empleado por el clasificador se ve afectado directamente, ya sea en la fase de entrenamiento o de clasificación, principalmente para los clasificadores basados en instancias debido a que éstos, para clasificar un solo prototipo, procesan toda la información del conjunto de entrenamiento. Por otra parte, no se garantiza que todos los elementos de un conjunto de entrenamiento sean útiles para el proceso de clasificación, ya que es común la presencia de elementos superfluos para tal proceso. Este tipo de elementos superfluos pueden ser prototipos ruidosos o redundantes. Los primeros afectan de manera negativa el desempeño del clasificador ya que pueden conducir a clasificaciones erróneas; mientras que los prototipos redundantes son innecesarios debido a que su

información descriptiva puede ser generalizada por algunos otros prototipos en el conjunto de entrenamiento.

En general, estos tipos de prototipos son superfluos para el clasificador, por lo que, la ausencia de éstos en la muestra no afecta en gran medida la calidad de clasificación.

En la práctica, es común la existencia de estos tipos de prototipos en los conjuntos de entrenamiento, por lo que surge la necesidad de descartar aquellos prototipos cuya eliminación afecte poco la calidad de clasificación del conjunto de entrenamiento. De este problema se encarga una rama del reconocimiento de patrones denominada *selección de prototipos*, en la que se sitúa le problema a resolver en este trabajo de investigación.

Existen dos maneras de proceder para llevar a cabo la selección de prototipos para la clasificación supervisada:

- *Filter.* Los prototipos se seleccionan con base en una función independiente del uso de algún clasificador.
- *Wrapper.* Los prototipos se seleccionan con base en los resultados obtenidos al utilizar algún clasificador. Esta estrategia puede ser para un clasificador particular (*SCP*) o para cualquier clasificador (*SCC*).

1.2 Problemática actual

Se han propuesto diversas soluciones para llevar a cabo la selección de prototipos. De las soluciones propuestas en la literatura, la gran mayoría son de tipo *wrapper SCP* y particularmente para el uso del clasificador *k-Nearest Neighbors* (*k-NN*) con lo que, los subconjuntos de prototipos seleccionados únicamente son un buen conjunto de entrenamiento para este clasificador particular.

En problemas de clasificación es común enfrentarse a casos en los que los datos descriptivos de los prototipos son de tipo mezclado, es decir, numéricos y

no numéricos. Una limitante en algunos métodos existentes para la selección de prototipos es que sus criterios de selección son exclusivamente aplicables a datos numéricos sin contemplar el caso de los datos no numéricos y los datos mezclados.

Otra de las características de los métodos existentes es que algunos de ellos presentan un alto costo computacional principalmente para el caso de medianos-grandes conjuntos de datos (del orden de más de 5000 prototipos) mientras que otros métodos son inaplicables para estos casos.

1.3 Motivación

Los conjuntos de entrenamiento son un factor básico e importante en la clasificación supervisada, ya que estos conjuntos proporcionan la información necesaria para que los clasificadores lleven a cabo los procesos de entrenamiento y clasificación.

Puede notarse que después de aplicar un método de selección de prototipos a un conjunto de entrenamiento T ocurre que $\mid S \mid < \mid T \mid$ con lo que, el beneficio de utilizar a S como conjunto de entrenamiento en la clasificación supervisada, es la reducción del tiempo necesario para los procesos de entrenamiento y clasificación. Este beneficio es aún más notorio para los clasificadores supervisados basados en instancias (aquellos que en todo momento utilizan a T para clasificar cada nuevo prototipo), ya que en este tipo de clasificadores el tiempo necesario para el proceso de clasificación de un nuevo prototipo es proporcional a $\mid T \mid$. Este beneficio en la reducción de tiempos de ejecución es la motivación principal para llevar a cabo la investigación descrita en este trabajo. Por otra parte, con base en lo mencionado en la problemática actual de los métodos para la selección de prototipos, en este trabajo de investigación se proponen cuatro soluciones para la selección de prototipos: dos de tipo *wrapper SCC* y dos de tipo *filter*. Los métodos *wrapper* propuestos en este trabajo llevan

a cabo la selección mediante la búsqueda secuencial flotante mientras que los de tipo *filter* basan la selección en agrupamientos y relevancias de los prototipos. De acuerdo a los resultados reportados en capítulos posteriores, estos cuatro métodos seleccionan buenos conjuntos de entrenamiento para varios clasificadores y en el caso específico de los métodos *filter* propuestos, llevan a cabo la selección de manera más rápida con respecto otros métodos de la literatura.

1.4 Objetivo General

El objetivo general del presente trabajo de investigación es:

Proponer métodos para la selección de prototipos de tipo *wrapper* y *filter* tales que permitan trabajar con datos mezclados (numéricos y no numéricos).

Los objetivos específicos de este trabajo de investigación son:

- Explorar el uso de la búsqueda secuencial para la selección de prototipos, en particular, la búsqueda secuencial flotante. En el contexto de la selección de prototipos, estas búsquedas evalúan subconjuntos de prototipos añadiendo o descartando un prototipo a la vez (a partir de un subconjunto inicial de prototipos) de manera repetida hasta que se satisface un criterio que finaliza la secuencia de la búsqueda y se elige el mejor subconjunto de prototipos.

- Proponer métodos para la selección de prototipos frontera para grandes conjuntos de entrenamiento. En el ámbito de la selección de prototipos es recomendable preservar este tipo de prototipos pues son aquellos prototipos que se encuentran en el borde de cada

clase, es decir, delimitan la pertenencia entre prototipos de distintas clases.

Con base en los puntos expuestos en el objetivo general, la principal contribución de este trabajo es el desarrollo de métodos que proporcionan una solución al problema de la selección de prototipos considerando los aspectos de la problemática actual.

En particular, en este trabajo de investigación se presentan cuatro métodos para la selección de prototipos, dos de ellos de tipo *wrapper SCC* y los otros dos de tipo *Filter*. Los métodos *wrapper SCC* están basados en la búsqueda restringida flotante para la selección de prototipos mientras que los métodos *filter* basan su criterio de selección en agrupamientos y la relevancia de prototipos por clase, respectivamente. De acuerdo a los resultados obtenidos, estos métodos superan en precisión (usando distintos clasificadores) y tiempo a otros métodos del estado del arte.

1.5 Descripción del documento

La manera en que está organizado el contenido de este documento es la siguiente:

En el capítulo 2 se define el problema de la selección de prototipos y se describen algunos de los trabajaos más relevantes relacionados a esta área de investigación.

En el capítulo 3 se introducen los métodos que se proponen en este trabajo para la selección de prototipos. Específicamente de tipo *wrapper SCC* y *filter*.

El capitulo 4 muestra los resultados experimentales obtenidos al evaluar el desempeño de los métodos propuestos y una comparación experimental contra otros métodos de selección de prototipos.

Finalmente, se exponen las conclusiones y algunas posibles direcciones a seguir como trabajo futuro.

Capítulo 2

Selección de prototipos

En la clasificación supervisada, se usa un conjunto de datos (conjunto de entrenamiento) para llevar a cabo la clasificación de nuevos casos.

Suele ocurrir que no todos los elementos del conjunto de entrenamiento son útiles para fines de clasificación ya que es común la presencia de ruido y elementos redundantes en tal conjunto, por esta razón es importante descartar del conjunto de entrenamiento aquellos prototipos cuya eliminación no impacte en la calidad de clasificación del conjunto. De este problema se encarga una rama del reconocimiento de patrones denominada *selección de prototipos.*

En este capítulo se describen la clasificación supervisada, posteriormente el problema de la *selección de prototipos* y finalmente algunos trabajos relacionados a la selección de prototipos.

2.1 Clasificación supervisada

En reconocimiento de patrones, la clasificación supervisada (figura 2.1) es un proceso mediante el cual se determina la clase de un nuevo prototipo p_N de acuerdo a sus características descriptivas (atributos), con base en un conjunto de entrenamiento T.

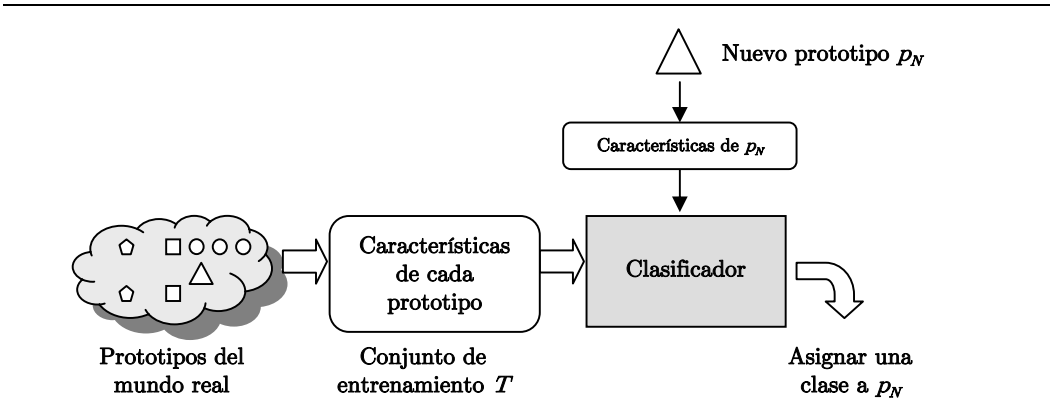


Figura 2.1. Proceso de clasificación supervisada

		Atributos			Clases
		a_1	a_2	a_3	a_4
	p_1	4.5	180	Titanio	Clase 1
	p_2	6.6	250	Oro	Clase 2
$T=$	p_3	4.2	200	Titanio	Clase 1
	p_4	5.9	321.4	Oro	Clase 2
	p_5	6.7	345.4	Platino	Clase 3

Figura 2.2. Ejemplo de un conjunto de entrenamiento con 5 prototipos y cada uno de ellos descrito por tres atributos a_1, a_2, a_3 y una clase a_4

En la clasificación supervisada, para cada $p_j = (a_1, a_2, ..., a_m) \in T$ se conoce la clase a la que el prototipo pertenece. Ejemplos del tipo de descripciones de prototipos en la clasificación supervisada se muestran en la figura 2.2, en la que se describe un conjunto de entrenamiento T de 5 prototipos con 4 atributos $(a_1, ..., a_4)$ de los cuales a_1, a_2 y a_3 corresponden a la descripción de los prototipos y en la última columna se muestra la clase de cada uno de éstos.

Este tipo de conjuntos son la base de los clasificadores supervisados, ya que, estos clasificadores hacen uso de los atributos y clases para decidir a qué clase pertenecen los nuevos prototipos que se presenten.

En reconocimiento de patrones existen diferentes clasificadores supervisados, por ejemplo: *k-Nearest–Neighbors* (*k-NN*), *Locally Weighted Regression* (*LWR*), *Support Vector Machines* (*SVM*), *C4.5* y *Naive Bayes* (*NB*). Los cuales, se usan en los resultados experimentales reportados en este documento y se explican brevemente en los siguientes párrafos. Estos clasificadores fueron elegidos para llevar a cabo experimentos con clasificadores de distintos tipos, como se verá en los siguientes párrafos, cada uno crea de distinta manera sus modelos para clasificar.

2.1.1 k-Nearest Neighbors (*k-NN*)

El clasificador *k-NN* [Covert & Hart, 1967] asigna la clase de un nuevo prototipo con base en la distancia entre éste y los prototipos de un conjunto de entrenamiento.

Dado un conjunto de entrenamiento T y un nuevo prototipo p_N a clasificar, la idea general de *k-NN* es la siguiente: se calcula la distancia entre p_N y cada $p_j \in T$. Posteriormente, con base en las distancias calculadas, se encuentran los k

prototipos en T más cercanos a p_N. Entonces, la clase que se asigna a p_N está en función de las clases de los k prototipos, siendo la función más simple la más frecuente.

Este tipo de clasificador es de tipo "basado en instancias", debido a que cada vez que se requiere clasificar un prototipo, se calcula la distancia entre éste y cada prototipo en T.

2.1.2 Locally Weighted Regression (*LWR*)

LWR [Atkeson, 1997] es una generalización de *k-NN*. *LWR* construye una aproximación a la clase de un nuevo prototipo p_N sobre una región local alrededor de p_N, es decir, considerando solamente prototipos cercanos al que se desea clasificar. Esta aproximación puede ser, por ejemplo, una combinación lineal de atributos con pesos predeterminados [Mitchell, 1997].

2.1.3 Support Vector Machines (*SVM*)

SVM [Vapnik, 1995] representa a los prototipos en T (espacio de entrada) en un espacio denominado espacio de atributos y construye hiper-planos de separación con máximo margen, donde el margen es la distancia entre un hiper-plano y los prototipos más cercanos. Con lo que se crea una frontera de decisión no lineal en el espacio de entrada para clasificar nuevos prototipos. Mediante el uso de funciones denominadas *kernel* es posible calcular los hiper-planos de separación sin explícitamente llevar a cabo la transformación al espacio de atributos. Los hiper-planos de separación se construyen mediante algunos prototipos denominados vectores de soporte [Hearst, 1998].

2.1.4 C4.5

C4.5 [Quinlan, 1993] genera un árbol de decisión, en el que cada nodo representa un atributo, cada rama corresponde a un posible valor del atributo y las hojas del árbol tienen asociada una etiqueta de clase. Dado un nuevo prototipo a clasificar p_N, se sigue la ruta del árbol de acuerdo al valor de los atributos de p_N hasta llegar a una hoja en la que se asigna la clase a p_N. Para generar el árbol, *C4.5* usa una medida de ganancia de información para determinar el atributo con el que se construirán las ramas más homogéneas. El primer nodo del árbol (raíz) corresponde al atributo con mayor ganancia de información y el número de ramas que descienden de éste corresponde al número de posibles valores del atributo. Posteriormente los demás nodos se crean siguiendo el mismo proceso. Cabe mencionar que *C4.5* es una modificación del algoritmo ID3 [Quinlan, 1993]. La diferencia entre ambos es que *C4.5* permite manejar atributos mezclados, ausencia de información y ruido mientras que ID3 solamente puede aplicarse a atributos no numéricos.

2.1.5 Naive Bayes

Este clasificador es de tipo estadístico, se basa en calcular la probabilidad de pertenencia de un prototipo a las distintas clases en el conjunto de entrenamiento [Han & Kamber, 2001]. De manera general, dado un nuevo prototipo p_N, se calcula la probabilidad a posteriori (mediante el teorema de Bayes y considerando como hipótesis a los ejemplos en T) de que p_N pertenezca a cada una de las distintas clases en T. Finalmente, se asigna a p_N la clase correspondiente al máximo valor de las probabilidades calculadas. Este clasificador asume que los atributos son independientes entre sí dada la clase.

2.2 Selección de prototipos

Cuando un nuevo prototipo p_N se presenta al clasificador, el objetivo de éste es determinar (a partir de la información proporcionada por T) la clase o etiqueta que se asignará a p_N. Una parte importante para el buen desempeño del clasificador es la calidad del conjunto de entrenamiento. Cuando la cantidad de prototipos (dimensionalidad) de un conjunto es grande, el tiempo empleado por el clasificador se ve afectado, ya sea en la fase de entrenamiento o de clasificación. Este aspecto es más notorio en los clasificadores basados en instancias.

Por otra parte, no se garantiza que todos los elementos de T sean útiles o proporcionen información relevante para el proceso de clasificación, ya que suelen presentarse elementos superfluos para tal proceso. Este tipo de elementos pueden ser:

- *Prototipos ruidosos.* Son los prototipos que menos información aportan al proceso de clasificación, ya que al ser considerados por el clasificador pueden causar confusión y posiblemente una clasificación errónea de los nuevos prototipos. Los errores producidos durante el proceso de recolección de la información son algunas de las causas que dan origen a este tipo de prototipos [Wilson & Martínez, 2000].
- *Prototipos redundantes.* Son prototipos cuyos atributos descriptivos pueden ser generalizados por algunos otros elementos en la muestra, por lo que resultan ser prototipos innecesarios en T. Este tipo de elementos se presentan en conjuntos en los que prototipos de la misma clase son muy similares [Brighton & Mellish, 2002].

En general estos prototipos son superfluos para los clasificadores, por lo que la ausencia de éstos en T no afecta en gran medida los resultados de clasificación, incluso, en algunos casos, esta ausencia beneficia la calidad de clasificación.

Debido a la existencia de estos tipos de prototipos en un conjunto de entrenamiento, surge la necesidad de seleccionar de entre los elementos de tal conjunto sólo aquellos con los que se preserva o degrada en menor medida la calidad del conjunto de entrenamiento. De este problema se encarga la rama del reconocimiento de patrones denominada *selección de prototipos*. El problema a resolver en esta propuesta de investigación doctoral se ubica en esta área de investigación.

La selección de prototipos (figura 2.3) se define de la siguiente manera:

"Dado un conjunto de entrenamiento T, el proceso de selección de prototipos consiste en elegir (mediante algún criterio de selección) un subconjunto S⊂T, de tal manera que S no contenga elementos superfluos. De aquí en adelante, se utilizará S para denotar al subconjunto de prototipos seleccionado."

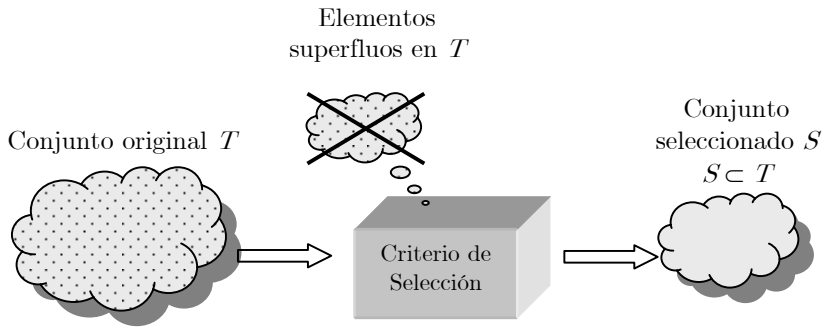


Figura 2.3. Proceso de selección de prototipos. A partir de un conjunto T se obtiene el subconjunto de prototipos S

Idealmente, lo que se busca en la selección de prototipos es algún conjunto S con el que se preserve la calidad de clasificación con respecto a T, es decir, $Acc(S) \cong Acc(T)$ con $|S| < |T|$, donde $Acc(X)$ es la calidad de clasificación (relativa a un conjunto de prueba) obtenida con el conjunto X; pero en la práctica, se buscan subconjuntos con los que se degrada poco la calidad de clasificación.

Según [Bezdek & Kuncheva, 1998, 2001], existen dos maneras para llevar a cabo la selección de prototipos:

- *Selección por extracción.* En el subconjunto obtenido S, para cada $p_i \in S, p_i \in T$, es decir, los prototipos en S son elementos de T (figura 2.4a).
- *Selección por reemplazo.* En el subconjunto obtenido S, se tiene que para cada $p_i \in S, p_i \notin T$, es decir, los elementos de S no son prototipos de T (figura 2.4b).

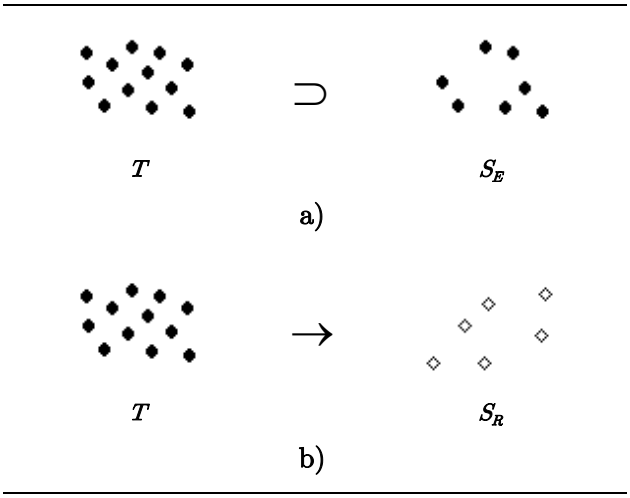


Figura 2.4. a) S_E es obtenido mediante la selección por extracción. **b)** S_R es obtenido mediante la selección por reemplazo. Ambos a partir de T.

El trabajo de investigación en este documento está enfocado en los métodos de selección por extracción.

De manera análoga a la selección de atributos, existen dos estrategias para llevar a cabo la selección de prototipos para la clasificación supervisada:

- *Filter*. Evalúa los subconjuntos de prototipos utilizando una función independiente de algún clasificador.

- *Wrapper*. Evalúa los subconjuntos de prototipos con base en los resultados obtenidos al utilizar algún clasificador. En la estrategia *wrapper*, la selección de prototipos puede ser:
  I) *Selección para un clasificador particular* (*SCP*). Este criterio de selección se basa en un clasificador particular, es decir, utiliza un clasificador específico con base en el cual, se determina cuándo un prototipo es eliminado durante la búsqueda.
  II) *Selección para cualquier clasificador* (*SCC*). En este tipo de selección, a diferencia de la anterior, no se restringe al uso de un clasificador particular para determinar qué prototipos serán descartados durante la búsqueda, sino que, es posible utilizar cualquier clasificador.

2.3 Trabajos relacionados

En esta sección se describen brevemente algunos métodos de tipo *filter* y *wrapper* para la selección de prototipos. Los trabajos descritos en esta sección se muestran en la figura 2.5. La revisión de los trabajos descritos en este capítulo incluye los métodos más relevantes en la literatura al respecto, de acuerdo a los resultados reportados por sus autores.

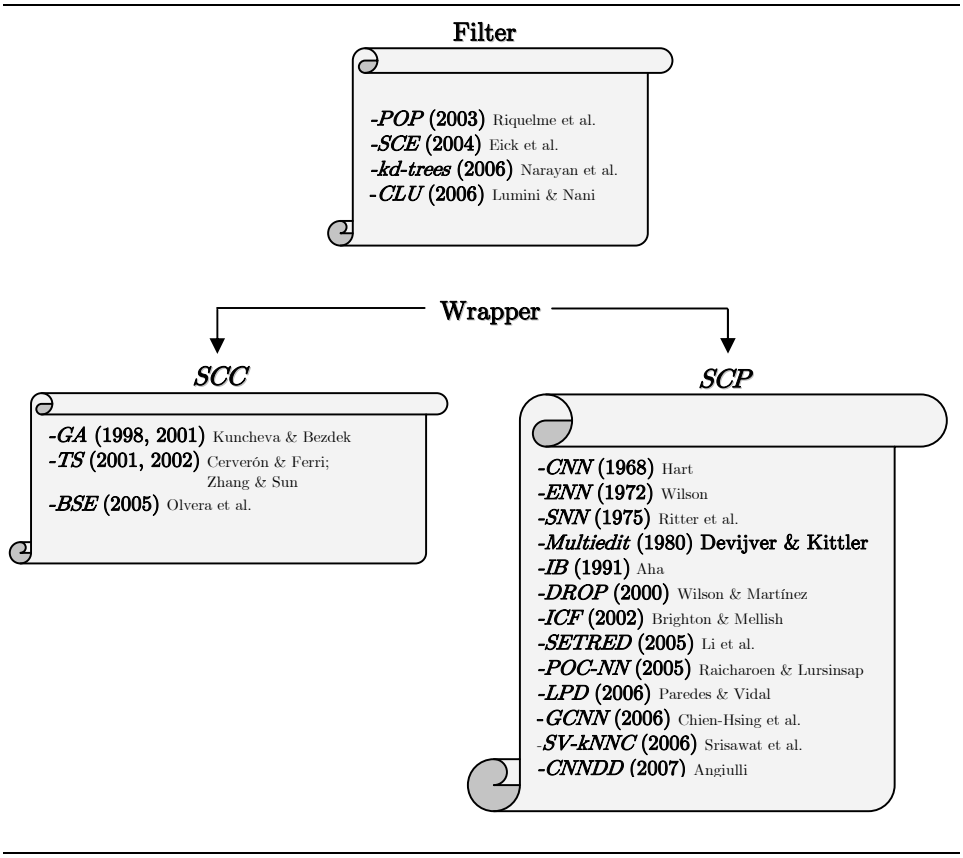


Figura 2.5. Algunos trabajos relacionados a la selección de prototipos

2.3.1 Métodos wrapper SCP

Como se ha mencionado en el capítulo anterior, una gran variedad de los trabajos de tipo *wrapper SCP* se han propuesto con base en la regla del vecino más cercano (*NN*) [Covert & Hart, 1967], y su generalización: *k-NN*. En los párrafos siguientes se describen algunos métodos de este tipo.

En [Hart, 1968] se propone uno de los primeros métodos para la selección de prototipos, la regla *Condensed Nearest Neighbor* (*CNN*), la cual consiste en encontrar de entre los elementos de T un subconjunto S tal que cada prototipo de

T sea clasificado correctamente (con 1-NN) usando a S como conjunto de entrenamiento. Además se asume que en T no existen prototipos cuyos atributos sean idénticos y correspondan a clases distintas. Este método comienza seleccionando de manera aleatoria un prototipo de cada una de las distintas clases y estos prototipos se añaden a S, el cual inicialmente es un conjunto vacío. Posteriormente, cada prototipo en T es clasificado empleando únicamente los prototipos de S. Cuando un prototipo p es clasificado erróneamente entonces éste se añade a S para garantizar que serán correctas las futuras clasificaciones de nuevos prototipos similares a p. El proceso se repite hasta que no existan prototipos en T que sean clasificados de manera errónea.

Esta técnica es sensible al ruido, ya que prototipos ruidosos suelen ser clasificados erróneamente por sus vecinos y de esta manera, los prototipos ruidosos se anexan a S, lo cual provoca dos inconvenientes. El primero es que no se logra una reducción considerable de la muestra, ya que los prototipos ruidosos son innecesarios pero aún siguen presentes. El segundo inconveniente es el efecto negativo que el subconjunto resultante causa en los resultados de clasificación, debido a que los prototipos ruidosos no aportan información relevante al clasificador. Una extensión de *CNN* fue realizada mediante la regla *Selective Nearest Neighbor* (*SNN*) [Ritter et al., 1975], la cual, garantiza encontrar un conjunto pequeño que clasifica correctamente a T.

En [Chien-Hsing et al., 2006] se presenta el método *GCNN* (Generalizad Condensed Nearest Neighbor Rule) que es una extensión de *CNN*. *GCNN* es idéntico a *CNN* pero con la excepción de que se añaden a S prototipos cuya distancia con prototipos similares es menor a un umbral de absorción. Este método selecciona S de tal manera que cada prototipo en T sea absorbido por S, es decir, sea representado por algún elemento en S. En particular, para *CNN* un prototipo p es absorbido si:

$$||p\text{-}x||\text{-}||p\text{-}w|| > 0 \tag{2.1}$$

Donde: x, $w \in S$, x es el prototipo más cercano a p de clase distinta, w es el prototipo más cercano a p de la misma clase y ||z|| es la norma del vector z.

Por otra parte, para GCNN, p es absorbido si:

$$||p\text{-}x||\text{-}||p\text{-}w|| > \delta \tag{2.2}$$

Cuando p satisface (2.1) se dice que p es ligeramente absorbido mientras que si satisface (2.2) entonces p es fuertemente absorbido. En cada iteración de *GCNN* se añaden a S aquellos prototipos que no satisfacen (2). El proceso de selección en *GCNN* finaliza cuando todos los elementos en T han sido fuertemente absorbidos.

Otra de las primeras técnicas de selección de prototipos es el método *Edited Nearest Neighbor* (*ENN*) [Wilson, 1972]. Este método descarta aquellos prototipos cuya clase es distinta a la de la mayoría de sus k vecinos más cercanos. Esta técnica suele emplearse para filtrar el ruido de una muestra, ya que se eliminan aquellos prototipos raros (ruidosos) cuya clase no coincide con la de la mayoría de sus k-vecinos cercanos, nótese que los prototipos seleccionados dependen del valor de k, comúnmente *ENN* utiliza k=3. La regla *RENN* (*Repeated ENN*) [Tomek, 1976] es una variante de *ENN* y consiste en aplicar *ENN* de manera repetida hasta que todos los prototipos en S tengan la misma clase que la mayoritaria de sus k vecinos más cercanos.

En [Devijver & Kittler, 1980] se presenta el método *Multiedit*, el cual crea de manera aleatoria l particiones (P_1, P_2, ..., P_l) a partir de T. Después de generar las particiones, se aplica *ENN* (con *1-NN*) a la partición P_i pero los vecinos se buscan en la siguiente partición, es decir $P_{(i+1) \bmod l}$. Este proceso se repite hasta que no haya eliminación de prototipos en t iteraciones sucesivas.

Otras variantes de *ENN* se presentan en [Sánchez et al., 2003] y [Vázquez et al., 2005] en las que se sigue la idea de *ENN* pero para encontrar a los vecinos cercanos

se usan el algoritmo *k-NCN* (*k-Nearest Centroid Neighborhood* [Chaudhuri, 1996]) y la probabilidad de pertenencia del prototipo a la clase, respectivamente. En el algoritmo *k-NCN*, para encontrar los k vecinos de p se elige al vecino más cercano a p como primer vecino p_{n1} y a partir de éste se encuentra el segundo vecino p_{n2} de manera que la media entre p_{n1} y p_{n2} sea la más cercana a p y así, de manera sucesiva se busca p_{nk}.

La regla *k-NCN* también se utiliza en [Lozano et al., 2003] para la selección de prototipos. Este método encuentra vecindarios con prototipos de la misma clase, de entre los cuales la mayoría de los prototipos en el vecindario son descartados y sólo son representados por algunos prototipos del grupo. El método se basa en calcular de manera repetida los vecinos de cada prototipo p_i (utilizando *k-NCN*) hasta que en el vecindario se detecta un prototipo con clase distinta a la de p_i. Los prototipos representativos p_R de cada vecindario son aquellos con mayor número de vecinos, por lo que se eliminan los vecinos de p_R.

En [Aha, 1991] se proponen una serie de métodos denominados *IB* (*Instance Based*): *IB2, IB3, IB4* e *IB5*, los cuales, para clasificar, utilizan el algoritmo *IB1* que es idéntico a la regla 1-*NN*. *IB2* almacena los prototipos clasificados erróneamente, pues es un método cuya regla a seguir es encontrar en la muestra original un subconjunto que contenga aquellos prototipos que fueron clasificados incorrectamente durante el proceso. *IB2* resulta ser sensible al ruido, pues con base en la regla que utiliza, almacena prototipos ruidosos, ya que, por su naturaleza, este tipo de prototipos suelen clasificarse de manera incorrecta. *IB3* es una extensión de *IB2*, en la cual básicamente se evita almacenar todos los prototipos ruidosos, considerando solamente aquellos que no afecten los resultados de clasificación. *IB3* analiza los resultados de clasificación antes de eliminar un prototipo ruidoso, mantiene un registro de cómo se clasifica con los prototipos que se van almacenando y elimina aquellos con los cuales, estadísticamente se ven

afectados los resultados de clasificación. *IB4* e *IB5* son extensiones de *IB3,* ya que para cada clase determinan un conjunto de pesos que serán asignados a los atributos de los prototipos para fines de cálculo de similitudes.

En [Angiulli, 2007] se presenta el método *CNNDD* (*Condensed NNDD*) que se basa en la regla *k-NNDD* (*k-Nearest Neighbor Domain Description*) referente al concepto de la Descripción del Dominio de los Datos (*Data Domain Description*) o también denominada clasificación de una clase cuyo objetivo es distinguir prototipos pertenecientes a una sola clase y aquellos que no pertenecen a ésta. Este concepto es comúnmente utilizado para la detección de prototipos cuya descripción es significativamente distinta a la del conjunto de entrenamiento. Para determinar la pertenencia de un prototipo a la clase se evalúa si éste se sitúa en una región aceptada (vecindario de radio ϕ) o de rechazo (vecindario de radio$>\phi$). La regla *CNNDD* obtiene un conjunto reducido consistente R a partir de T, es decir, retiene prototipos en R que clasifican correctamente a todo T. Este método comienza por incluir en el conjunto R (inicialmente $R{=}\emptyset$) a los prototipos que se sitúan en regiones aceptadas y posteriormente de entre los prototipos situados en las regiones de rechazo se analiza si alguno de ellos puede añadirse a R.

Otros métodos de selección de prototipos denominados *DROP1, DROP2, DROP3, DROP4* y *DROP5* (*Decremental Reduction Optimization Procedure*) [Wilson & Martínez, 2000] basan su regla de selección de prototipos en términos del concepto de socio. El socio de un prototipo p es aquél prototipo que tiene a p como uno de sus k vecinos más cercanos. *DROP1* elimina un prototipo p de S si sus socios en S se clasifican correctamente sin p, es decir, bajo este criterio, la ausencia de p no afecta la clasificación. *DROP1* comienza calculando las listas de vecinos y socios para cada prototipo en S. Posteriormente, en cada paso, se descarta de S al prototipo p tal que los socios de p en S se clasifican correctamente sin p. Con base en esta regla, puede notarse que *DROP1* elimina prototipos ruidosos, ya que,

comúnmente, los socios de un prototipo ruidoso pueden clasificarse correctamente sin tal prototipo. Puede ocurrir que *DROP1* descarte por completo conjuntos de prototipos de la misma clase antes de descartar prototipos ruidosos, en tal caso, no todos los prototipos ruidosos son eliminados. Para solucionar este problema, *DROP2* verifica el efecto que causa la eliminación del prototipo en T, es decir, *DROP2* elimina p de S si los socios que p tiene en T se clasifican correctamente sin p.

Los prototipos ruidosos pueden situarse en regiones frontera (regiones donde existan prototipos cercanos con distintas clases), por lo que *DROP3*, *DROP4* y *DROP5* aplican un filtrado de ruido como paso previo al proceso de selección (basado en *DROP2*). Este filtrado se lleva a cabo para suavizar las regiones frontera, es decir, eliminar aquellos prototipos muy cercanos pero pertenecientes a distintas clases. La diferencia entre *DROP3, DROP4* y *DROP5* es el criterio empleado en la etapa de filtrado. *DROP3* y *DROP4* utilizan *ENN* como filtro de ruido pero *DROP4* lleva a cabo una etapa previa a la eliminación del prototipo ruidoso, verifica el impacto de clasificación provocado al no considerar tal prototipo para determinar si será o no eliminado. El filtrado utilizado por *DROP5* consiste en eliminar primero a los prototipos cercanos a regiones frontera, los cuales corresponden a prototipos cercanos con distinta clase (enemigos más cercanos).

En [Brighton & Mellish, 2002] se propone el método *Iterative Case Filtering* (*ICF*), cuya regla de selección se basa en los conjuntos *Reachable*(p) y *Coverage*(p) del prototipo p, los cuales se refieren a los conjuntos de vecinos más cercanos y de socios respectivamente. La regla de selección es la siguiente: eliminar aquellos prototipos tales que el tamaño de *Reachable*(p) es mayor que el de *Coverage*(p). Mediante esta regla, un prototipo p será eliminado cuando mediante otros prototipos se generaliza la información que p pudiera proporcionar. Como etapa inicial, *ICF* filtra la muestra de prototipos ruidosos empleando *ENN*.

En [Ke-Ping et al., 2003] se utiliza la idea de los conjuntos *Reachable*(p) y *Coverage*(p) pero se modifica el concepto de *Coverage*(p) de tal manera que los socios de p correspondan únicamente a la misma clase de p, ya que al descartar un prototipo, no se afecta los resultados de clasificación debido a que la eliminación se realizó de entre un conjunto de prototipos con la misma clase. Esta técnica determina si un prototipo es ruidoso, superfluo o crítico, donde un prototipo crítico es aquél cuya eliminación afecta la clasificación de otros prototipos, por lo cual se descartan prototipos ruidosos o aquellos que son superfluos pero no críticos. En este método, p es un prototipo ruidoso si el tamaño de *Reachable*(p) es mayor que el de *Coverage*(p) mientras que p es superfluo cuando éste es clasificado correctamente por *Reachable*(p). Por otra parte, en este método se establecen reglas para determinar el orden en que se descartan los prototipos, ya que si existen dos prototipos p_i, p_j que serán descartados, la decisión se toma con base en el número de los vecinos y enemigos más cercanos de p_i y p_j.

La clasificación semi-supervisada se utiliza en [Li & Zhi-Hua., 2005] con el método *SETRED* (*SElf-TRaining with EDiting*). En este tipo de clasificación, en la fase de entrenamiento de los clasificadores se utilizan prototipos etiquetados (con una clase que se conoce *a priori*) y prototipos no etiquetados (se desconoce la clase a la que éstos pertenecen). *SETRED* divide a T en dos conjuntos: uno etiquetado L y otro no etiquetado U ($L \cup U = T$, $L \cap U = \emptyset$). De manera repetida se elige un conjunto $L' \subset U$ de prototipos confiables. En este método, los prototipos confiables son aquellos más cercanos a cada una de las clases de los ejemplos en L. Posteriormente, de entre los prototipos en L' se descartan aquellos situados en una región de rechazo del vecindario, la cual, se especifica mediante un umbral. Finalmente, $S = L \cup L'$. En este método el número de iteraciones se especifica como parámetro inicial del proceso.

El clasificador *SVM* (*Support Vector Machines*) puede considerarse como otra manera de seleccionar prototipos, ya que de entre todos los elementos en T, sólo el conjunto de vectores de soporte V_s son necesarios para delimitar la separación entre las distintas clases; por tanto, en el ámbito de selección de prototipos, se puede considerar $S = V_s$.

Un método *wrapper SCP* que se basa en considerar la selección de prototipos mediante *SVM* se presenta en [Yuangui et al., 2005], el cual lleva a cabo una doble selección. La primera selección se obtiene al aplicar *SVM* a T, posteriormente, de entre los vectores de soporte obtenidos se lleva a cabo una segunda selección. En particular, para esta segunda selección se utiliza el método *DROP2*. Otro método basado en vectores de soporte es *SV-kNNC* (*Support Vector k-Nearest Neighbor Clustering*) [Srisawat et al., 2006], el cual después de aplicar *SVM,* utiliza el algoritmo *k-means* para agrupar al conjunto de vectores de soporte y preservar grupos homogéneos, es decir, en cada grupo de prototipos se descarta aquellos que no pertenecen a la clase mayoritaria del grupo.

2.3.2 Métodos wrapper SCC

También se han propuesto métodos de tipo *wrapper SCC*, a continuación se describen algunos de estos trabajos.

Una manera de llevar a cabo el proceso de selección de prototipos es mediante búsquedas aleatorias guiadas tales como los algoritmos genéticos (*GA*) [Holland, 1975], las cuales han tenido diversas aplicaciones en problemas referentes a optimización [Goldberg, 1989], [Fogel, 1995]. Los *GA* se basan en la idea de la evolución de las especies. La idea general de los *GA* es la siguiente: dada una población (conjunto de soluciones), y de acuerdo al valor de la función de aptitud que evalúa a los individuos de la población (soluciones), se seleccionan de manera

repetida los mejores individuos (que maximizan la función de aptitud) y se combinan para generar nuevos individuos. Comúnmente, en la selección de prototipos, suele usarse la precisión de clasificación como función de aptitud. Se han presentado diversos métodos para la selección de prototipos mediante *GA*, algunos ejemplos son los presentados en [Kuncheva, 1995, 1997], [Kuncheva & Bezdek, 1998, 2001], [Cano et al., 2003], entre otros.

En [Cerverón & Ferri, 2001], [Zhang & Sun, 2002] se utiliza la búsqueda Tabú (*TS*) [Glover, 1986] para la selección de prototipos. Esta búsqueda (también de tipo aleatoria guiada) se aplica a un subconjunto de la muestra original, denominado solución inicial S_i. Durante la búsqueda, se detectan prototipos que no deben ser excluidos del conjunto solución, en este sentido, son prototipos Tabú. Una vez que se ha obtenido S_i, se busca a partir de éste, algún subconjunto (permitido) $S \subset S_i$ tal que se obtenga una mayor precisión en la clasificación con respecto a la solución inicial. La manera en que se busca S es evaluando todos los subconjuntos vecinos de S_i, es decir los subconjuntos que difieren de S_i solo en un elemento y de manera repetida se reemplaza S_i por el subconjunto vecino con mejor clasificación.

Una característica particular de las búsquedas guiadas como *GA* y *TS* es que su funcionamiento depende en gran medida de parámetros iniciales, ya que es difícil fijar los valores de parámetros con los que se obtengan buenos resultados para cualquier problema. Y además, son métodos con un alto costo computacional.

Otra manera de encontrar soluciones sub-óptimas en problemas de selección es la búsqueda secuencial, la cual ha sido aplicada a problemas como la selección de atributos [Pudil et al., 1994; Blum & Langley, 1997]. Este tipo de búsqueda también ha sido extendida para la selección de prototipos. En [Olvera et al., 2005a] se propone el método *BSE* (*Backward Sequential Edition*) basado en búsqueda secuencial hacia atrás (BSS [Kittler, 1986]) adaptada para la selección de prototipos. *BSE* es un método de selección no exhaustivo que trabaja de la

siguiente manera: dado un conjunto inicial T, en cada paso se descarta o elimina el prototipo que menos información aporta para la calidad o precisión de clasificación del subconjunto parcial, de tal manera que, en el primer paso, después de descartar un prototipo y probar todos los posibles subconjuntos con cardinalidad $|T|-1$, se encuentra el mejor de ellos, en el segundo paso el mejor subconjunto de prototipos de cardinalidad $|T|-2$ es encontrado, y así sucesivamente.

BSE es un método costoso ya que analiza el impacto de eliminación de cada uno de los prototipos del subconjunto parcial, pero en [Olvera et al., 2005b] se proponen métodos denominados *esquemas de edición BSE* con los cuales se reduce el tiempo de ejecución de *BSE*. Estos métodos aplican *BSE* a un subconjunto obtenido mediante el pre-procesamiento del conjunto original, de tal manera que el proceso de selección se lleva a cabo a partir de muestras pequeñas con respecto al tamaño de la original. Los enfoques de pre-procesado de estos esquemas son dos: uno basado en aplicar el proceso de selección a una muestra previamente filtrada de prototipos ruidosos (*ENN+BSE*) y el otro se basa en aplicar el proceso de selección a una muestra previamente reducida (*DROP2+BSE,...,DROP5+BSE*), ya que puede ocurrir que aún existan prototipos redundantes en una muestra a la cual se le ha aplicado previamente algún método de selección de prototipos.

2.3.3 Métodos Filter

En los siguientes párrafos se describen algunos de los métodos de tipo *filter* que han sido propuestos para la selección de prototipos.

En general, mediante un método de selección de prototipos es importante retener prototipos frontera (prototipos cercanos a las fronteras entre clases) ya que

mediante éstos se conserva la separabilidad entre las distintas clases [Wilson & Martínez, 2000], [Brighton & Mellish, 2002].

El método *Pattern by Ordered Projections* (*POP* [Riquelme et al., 2003]) elimina prototipos interiores (lejanos a las fronteras entre clases) de cada clase y preserva algunos prototipos frontera. Este método es una heurística (denominada así por sus autores) que se basa en el concepto de debilidad(p) de cada prototipo, el cual se define como el número de veces que el prototipo p no es frontera o límite en una clase respecto a un atributo (no se encuentra cerca de otra clase). La regla de selección consiste en eliminar a los prototipos irrelevantes, los cuales, según este método, son aquellos prototipos cuya debilidad es igual al número total de atributos que describen a los prototipos, es decir, prototipos que no son bordes de clase. Este método calcula la debilidad de los prototipos con base en ordenamientos crecientes de los valores de atributos, ya que, por ejemplo, para dos dimensiones, representa una región de prototipos que pertenecen a la misma clase con sólo a lo más cuatro prototipos, correspondientes a los valores mínimo y máximo de cada atributo. En este contexto, al considerar la figura 2.6, los prototipos que representan la región de la clase 1 son $\{p_1, p_4\}$ mientras que $\{p_2, p_3\}$ son los prototipos con mayor debilidad, por lo que son descartados. Para la clase 2, ésta es representada por $\{p_6, p_8, p_9\}$.

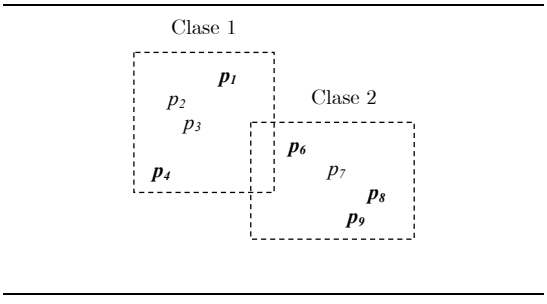


Figura 2.6. Ejemplos de prototipos frontera de acuerdo al método POP.

Puede notarse que esta idea es válida para atributos numéricos, sin embargo, para atributos no numéricos no es trivial el concepto de orden. En [Aguilar-Ruiz et al., 2006] se presenta una extensión de este método, la cual elimina prototipos si la debilidad satisface el valor de un factor correspondiente al número de atributos y el valor de un umbral.

Una técnica propuesta para seleccionar prototipos frontera se presenta en [Raicharoen & Lursinsap, 2005] con el método *POC-NN* (*Pairwise Opossite Class-Nearest Neighbor*) el cual, para detectar prototipos cercanos a las regiones frontera se basa en la cercanía entre los prototipos de una clase y el valor de la media de los prototipos de la clase opuesta. *POC-NN* comienza encontrando los prototipos media de cada clase, posteriormente se encuentra el punto medio entre estos prototipos media y se crea un hiperplano para separar T en regiones. En cada región se buscan los prototipos borde y el proceso se repite dividiendo en dos cada subregión hasta que en cada región obtenida existen prototipos pertenecientes a la misma clase. Para encontrar un prototipo frontera p_B de la región r_1, se calcula la media m_1 de esta región y además el prototipo más cercano p_{N2} a m_1 en la región opuesta r_2. Entonces p_B es un prototipo frontera si es el más cercano a p_{N2} en la región r_1. Finalmente, S es el conjunto de todos los prototipos frontera de cada región.

Un método *filter* basado en árboles *kd* (*k-dimensional trees* [Friedman et al., 1997]) se propone en [Narayan et al., 2006]. Este método consiste en crear un árbol binario (considerando k atributos o dimensiones). Para crear el árbol se parte de la raíz (todos los prototipos en T) y a partir de un atributo pivote se generan los nodos hijos. El pivote utilizado es el i-ésimo atributo para el que exista la máxima diferencia entre los valores de éste en dos prototipos consecutivos (previamente ordenados de manera ascendente con respecto al atributo i). Los nodos se generan de la siguiente manera: en el nodo izquierdo se colocan los prototipos tales que sus

valores del atributo (de acuerdo al atributo pivote del nodo antecesor) sean menores a cierto umbral, mientras que en el nodo derecho se colocan los prototipos cuyos valores del atributo son mayores al umbral. Este proceso se repite hasta que los nodos no pueden ser divididos. Finalmente, S está formado por los prototipos situados en las hojas del árbol.

Algunos autores [Leung et al., 2000; Bezdek & Kuncheva, 2001; Liu & Motoda, 2002; Spillmann et al., 2006] han mencionado la idea de utilizar los métodos de agrupamiento para la selección de prototipos. Esta idea consiste en dividir T en r grupos, entonces, S está formado únicamente por los centros de cada grupo de prototipos. En [Lumini & Nanni, 2006] se propone el método *CLU* (*CLUstering*), el cual se basa en esta idea para seleccionar prototipos en problemas biométricos, específicamente para reconocimiento de las firmas de personas.

Otro método para la selección de prototipos basado en agrupamientos es *SCE* (*Supervised Clustering Editing*) [Eick, et al., 2004] que consiste en, de manera inicial, elegir aleatoriamente un conjunto de prototipos representantes C_R con base en los cuales se crean los agrupamientos. Posteriormente, el proceso de agrupamiento se repite descartando un prototipo en C_R y añadiendo a este conjunto uno de los restantes prototipos de la muestra hasta que la calidad de clasificación disminuye con respecto al mejor resultado parcial obtenido. Este método es costoso ya que analiza los resultados de clasificación considerando cada uno de los prototipos en la muestra.

Dentro del campo del razonamiento basado en casos (*Case-Based Reasoning*) se ha utilizado el concepto de prototipo representativo para determinar los prototipos con los que se tiene la mejor descripción de las distintas clases. En [Rodríguez et al., 2000] se seleccionan prototipos ejemplares para la descripción de conceptos. En este contexto, un prototipo ejemplar es aquél más similar a los de su misma clase y más disimilar a los prototipos ejemplares de otras clases.

2.3.4 Análisis de los trabajos relacionados

En la tabla 2.1, se muestran las características generales de los métodos descritos previamente; el símbolo "√ " indica el tipo de cada método (*filter* o *wrapper*), así como el tipo de atributos al que puede ser aplicado (numéricos, no numéricos o ambos).

Tabla 2.1. Características generales de los métodos descritos en este capítulo.

Método	Wrapper		Filter	Atributos	
	SCP	SCC		Numéricos	No numéricos
CNN (1968)	√			√	√
ENN (1972)	√			√	√
SNN (1975)	√			√	√
Multiedit (1980)	√			√	√
IB (1991)	√			√	√
DROP (2000)	√			√	√
ICF (2002)	√			√	√
SETRED (2005)	√			√	√
GCNN (2006)	√			√	√
SV-*k*NNC (2006)	√			√	
CNNDD (2007)	√			√	√
GA (1998, 2001)		√		√	√
TS (2001, 2002)		√		√	√
BSE (2005)		√		√	√
POP (2003)			√	√	√
SCE (2004)			√	√	
POC-NN (2005)			√	√	
kd-trees (2006)			√	√	
CLU (2006)			√	√	

A partir de las características mostradas en la tabla 2.1, puede notarse que la mayoría de los métodos *wrapper* son para un clasificador particular (*SCP*). De entre éstos, a excepción de *SV-kNN,* todos pueden aplicarse a atributos mezclados y los más exitosos (de acuerdo a los resultados reportados por sus autores) son *DROP* y *GCNN*, pero ambos basados en la regla del vecino más cercano.

Por otra parte, puede observarse que solamente *GA, TS* y *BSE* son métodos *wrapper* para cualquier clasificador (*SCC*) aplicables a atributos mezclados (numéricos, y nominales). Para los métodos basados en *GA* y *TS* es difícil_ajustar los valores óptimos de sus parámetros iniciales y al igual que *BSE*, son métodos con un alto costo computacional.

En lo que respecta a los métodos *filter*, solamente *POP* puede aplicarse para atributos mezclados pero como se comentó en párrafos anteriores, su criterio de selección tiene sentido sólo para atributos numéricos. Los demás métodos *filter* son aplicables exclusivamente para atributos numéricos.

Con base en estas características de los métodos descritos, este trabajo de investigación se ubica en el desarrollo de métodos de tipo *filter* y *wrapper SCC* aplicables a atributos mezclados.

Capítulo 3

Métodos propuestos para la selección de prototipos

En este capítulo se proponen métodos para la selección de prototipos, específicamente de tipo *wrapper* SCC y *filter*.

Los métodos *wrapper* aquí propuestos se basan en la búsqueda secuencial para la selección de prototipos, en particular en la búsqueda secuencial flotante, que ha sido utilizada en la solución de problemas de selección de atributos.

Por otra parte, los métodos *filter* propuestos en este capítulo seleccionan prototipos frontera mediante agrupamientos y la relevancia de los prototipos en cada clase, respectivamente.

En particular, este capítulo presenta los métodos:

- *RFPS* (*Restricted Floating Prototype Selection*)
- *RFPS-Inv* (*Restricted Floating Prototype Selection-Inverse*)
- *PSC* (*Prototype Selection by Clustering*)
- *PSR* (*Prototype Selection by Relevance*)

3.1 RFPS (Restricted Floating Prototype Selection)

El primer método propuesto en este capítulo se basa en adaptar técnicas de búsqueda secuencial a la selección de prototipos, las cuales han sido útiles para resolver problemas de selección de atributos [Blum & Langley, 1997].

El método para la selección de prototipos que se propone en esta sección se basa específicamente en la idea de la búsqueda secuencial flotante (BSF) [Pudil et al., 1994], la cual, puede ser hacia adelante o hacia atrás. La BSF, a diferencia de la búsqueda secuencial simple, permite realizar pasos de inclusión-exclusión o viceversa durante la búsqueda. En [Pudil et al., 1994], se propuso la BSF para la selección de atributos, a continuación se da una breve descripción de la BSF pero en el contexto de la selección de prototipos.

La BSF hacia atrás comienza con $S{=}T'$ y consiste en aplicar después de cada paso hacia atrás (exclusión del peor prototipo en S) pasos consecutivos hacia adelante (inclusión condicional en S de prototipos descartados durante los pasos hacia atrás). La cantidad de pasos de inclusión condicional es controlada mediante la precisión de clasificación, es decir, se incluyen prototipos en S mientras la precisión obtenida al considerar la inclusión sea mejor con respecto a la última mejor obtenida. La BSF hacia adelante comienza con $S = \varnothing$ y corresponde a la contraparte de la BSF hacia atrás, es decir, consiste en aplicar después de cada paso hacia adelante (inclusión en S del mejor prototipo) pasos consecutivos hacia atrás (exclusión condicional de los prototipos en S).

Debido al alto costo computacional de la búsqueda secuencial flotante, en nuestro método proponemos aplicar un paso previo para reducir la cantidad inicial de prototipos en T, además de restringir la búsqueda secuencial flotante hacia atrás a una serie de pasos de exclusión condicional (excluir prototipos mientras la

[1] S es el subconjunto de prototipos seleccionado y T corresponde al conjunto de entrenamiento.

precisión se mantenga o sea mejor) seguidos de una serie de pasos de inclusión condicional (incluir prototipos mientras la precisión sea mejor) y de manera inversa para búsqueda secuencial hacia adelante.

El método para la selección de prototipos propuesto en esta sección es *RFPS* (*Restricted Floating Prototype Selection*) cuyo pseudo-código se muestra en el algoritmo 3.1. *RFPS* comienza con el pre-procesado de la muestra, posteriormente se aplica la exclusión condicional seguida de la inclusión condicional. Un esquema general de *RFPS* se muestra en la figura 3.1.

```
RFPS (Conjunto de entrenamiento T)
    //pre-procesamiento
S= subconjunto obtenido después de aplicar algún método de selección de prototipos a T
Mejor_val =Clasif(S)
Repetir                                  //exclusión condicional
        Peor=null
        Para cada prototipo p en S
          S'=S-{p}
          Si Clasif(S') ≥ Best_val
              Peor=p
              Mejor_val=Clasif(S')
        Si Peor≠null
            S=S-{Peor}
Hasta que Peor==null o |S|==1
D=T-S
Para cada prototipo p_i en D  //inclusión condicional
   S'' = S ∪{p_i}
   Si Clasif(S'') >Mejor_val
       Mejor_val = Clasif(S'')
       S=S ∪{p_i}
Regresar S
```

Algoritmo 3.1. Método *RFPS* para la selección de prototipos

Cada una de las tres etapas del método *RFPS* se detalla a continuación.

- **Pre-procesado.** En esta etapa inicial se reduce el tamaño de T con el objetivo de comenzar el proceso de selección a partir algún subconjunto de menor tamaño que T debido a que aplicar la búsqueda secuencial a todo T es un procedimiento costoso. En particular, para llevar a cabo la reducción inicial de T se utiliza algún método de selección de prototipos.

- **Exclusión condicional.** En esta tapa se descartan prototipos de manera secuencial en el conjunto parcial. Este proceso analiza cada prototipo y en cada paso descarta el prototipo que menos aporta para la calidad de clasificación, en términos de la precisión de un clasificador, la cual es calculada mediante alguna función basada en cualquier clasificador; de manera que *RFPS* es un método para la selección de prototipos de tipo *wrapper SCC*.

  Para ejemplificar este proceso supóngase que después del pre-procesado se tiene un conjunto de n prototipos, $T=\{p_1, p_2, \ldots, p_n\}$ y los m prototipos descartados en el preprocesado son $D=\{ p_{d1}, p_{d2},.., p_{dm}\}$. Considérese también la función $Clasif(X)$, con la cual se obtiene la precisión de clasificación al utilizar como entrenamiento de algún clasificador al conjunto de prototipos X.

  La exclusión condicional analiza la calidad de clasificación al excluir cada prototipo en $S=T$. Para ello, comienza evaluando cada subconjunto de cardinalidad $|S-1|$, es decir $Clasif(S-\{p_1\})$, $Clasif(S-\{p_2\})$, …, $Clasif(S-\{p_n\})$, después se elimina el prototipo que menos calidad aporta en el conjunto de prototipos, es decir, aquel p_i con el que $Clasif(S-\{p_i\})$ es el valor mínimo y p_i se incluye en el conjunto de prototipos descartados $D=D \cup \{p_i\}$. Este proceso se repite para los subconjuntos de cardinalidad $|S-2|, |S-3|, |S-4|, \ldots,$ hasta que $|S|=1$ o hasta que de acuerdo a $Clasif(S)$ se tiene un conjunto subóptimo de clasificación.

- **Inclusión condicional.** Durante las etapas de pre-procesdo y exclusión condicional se eliminan prototipos de S, los cuales se incluyen en $D=\{ p_{d1}, p_{d2}, \ldots, p_k\}$. Los pasos de inclusión condicional en *RFPS* analizan de manera secuencial si la inclusión en S de alguno de los prototipos en D contribuye a

mejorar la calidad de clasificación. Es decir, la inclusión condicional evalúa $Clasif(S \cup \{p_{di}\})$ con cada $p_{di} \in D$ e incluye en S aquellos prototipos tales que $Clasif(S \cup \{p_{di}\}) \geq Clasif(S)$.

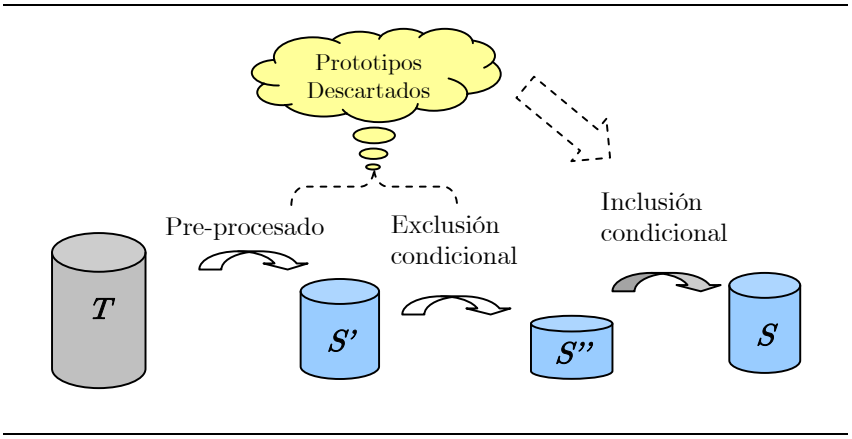


Figura 3.1. Esquema general del método *RFPS* para la selección de prototipos

RFPS es un método de búsqueda flotante restringido debido a que aplica la búsqueda secuencial flotante sólo una vez en cada dirección: la exclusión condicional seguida de la de inclusión condicional, ya que aplicar el método flotante completo sería un proceso muy costoso.

RFPS también puede aplicarse de manera inversa (*RFPS-Inv*), es decir, iniciar con la inclusión condicional y posteriormente la exclusión condicional (Figura 3.2). El método *RFPS-Inv* se muestra en el algoritmo 3.2.

RFPS-Inv, a partir de un conjunto previamente reducido (*S'*) analiza de manera secuencial cada prototipo en *T-S'* (inclusión condicional) para incluir en *S'* sólo aquellos prototipos con los que se mejore la calidad de clasificación. Finalmente, se aplica la exclusión condicional al conjunto obtenido mediante la inclusión condicional.

RFPS-Inv (Conjunto de entrenamiento T)
//pre-procesamiento
S= subconjunto obtenido después de aplicar *ENN* o algún método *DROP* a T
Mejor_ val =*Clasif(S)*
D= *T*-*S*
Para cada prototipo p en D // inclusión condicional
$S' = S \cup \{p\}$
Si *Clasif*(S') >*Mejor_ val*
Mejor_ val = *Clasif*(S')
S=$S \cup \{p\}$
Mejor _ val =*Clasif(S)* //exclusión condicional
Repetir
Peor=*null*
Para cada prototipo p en S
S''=S-$\{p\}$
Si *Clasif(S'')* $\geq$ *Best_ val*
Peor=p
Mejor _ val=*Clasif(S'')*
Si *Peor* $\neq$ *null*
S=S-{*Peor*}
Hasta que *Peor*==*null* o $|S|$==1
Regresar S

Algoritmo 3.2. Método *RFPS-Inv* para la selección de prototipos

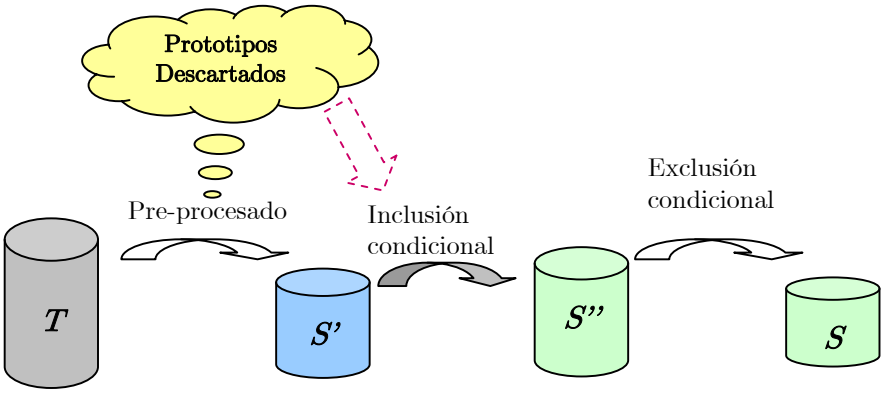


Figura 3.2. Esquema general del método *RFPS-Inv* para la selección de prototipos

3.2 PSC (*Prototype Selection by Clustering*)

Otro de los métodos de los prototipos propuestos en este capitulo es *PSC* cuya descripción se detalla a continuación.

En un conjunto de entrenamiento, los prototipos frontera proporcionan información a los clasificadores para preservar las regiones de discriminación entre las distintas clases. Por otra parte, la mayoría de prototipos interiores de cada clase (prototipos que no son frontera) son superfluos, ya que su ausencia afecta poco la calidad de clasificación del conjunto de entrenamiento [Wilson & Martínez, 2000; Brighton & Mellish, 2002]. A pesar de que los prototipos interiores no son relevantes para los clasificadores, algunos de estos prototipos son necesarios para representar las regiones interiores en el conjunto de entrenamiento.

Varios métodos para la selección de prototipos seleccionan prototipos frontera, a pesar de que sus autores no lo mencionan explícitamente. Como ejemplo de ello, consideremos el conjunto de datos mostrado en la figura 3.3a en la que se muestra un conjunto de datos bidimensional con prototipos pertenecientes a las clases "+" y "•".

En las figuras 3.3b-3.3e se muestran los subconjuntos seleccionados por algunos métodos para la selección de prototipos, en particular *DROP3*, *DROP5* (dos de los métodos más relevantes), *GCNN* (método competitivo contra los métodos *DROP*) y *POC-NN* (método propuesto para seleccionar prototipos frontera). A partir de este ejemplo, puede notarse que efectivamente estos métodos seleccionan prototipos cercanos a la frontera entre clases, lo cual confirma la importancia de retener este tipo de prototipos.

En esta sección, se introduce el método *filter PSC* (*Prototype Selection by Clustering*), el cual, selecciona prototipos frontera y algunos prototipos interiores de cada clase. Este método de selección de prototipos se basa en agrupar el conjunto

de entrenamiento y seleccionar los prototipos frontera a partir de grupos no homogéneos, los cuales se definen a continuación.

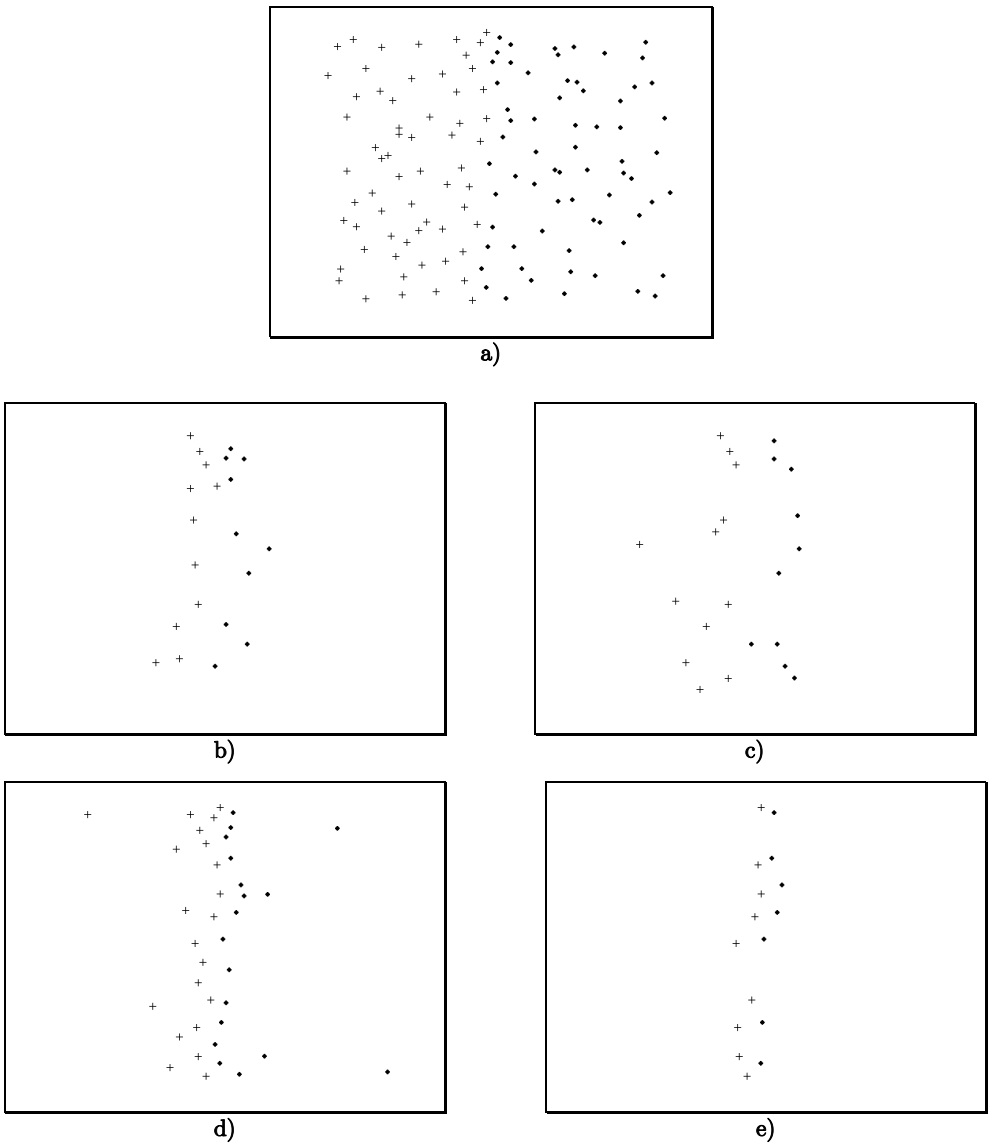


Figura 3.3. a) Conjunto de datos con clases "+" y "•". b) Prototipos seleccionados con DROP3. c) Prototipos seleccionados con DROP5. d) Prototipos seleccionados con GCNN. e) Prototipos seleccionados con POC-NN

DEFINICIÓN (*Grupo homogéneo)*. Un grupo C_x de T es homogéneo cuando todos los prototipos en C_x pertenecen a la misma clase.

DEFINICIÓN (*Grupo no homogéneo)*. Un grupo C_x es no homogéneo cuando los prototipos en C_j pertenecen al menos a dos clases.

Para encontrar prototipos frontera, *PSC* genera c grupos y analiza los aquellos no homogéneos debido a que los prototipos frontera están situados en regiones críticas, es decir, regiones en las que los prototipos pertenecen a diferentes clases.

Como se mencionó antes, *PSC* genera grupos de prototipos a partir de T, por lo que es necesario utilizar algún método de agrupamiento. Comúnmente, los métodos agrupamientos representan los centros de cada grupo mediante la media o valor promedio de éste, concepto que es solamente aplicable a los atributos numéricos. Pero cuando se desea agrupar prototipos descritos mediante atributos no numéricos ¿cómo calcular el promedio de valores no numéricos?.

Para generar los agrupamientos considerando atributos mezclados, *PSC* utiliza el algoritmo de agrupamiento *k-means with similarity functions* (*kMSF*) [García & Martínez; 1999], el cual, es un método de agrupamiento basado en la misma idea del método *k-means*, pero la diferencia entre estos dos métodos se encuentra en la función utilizada para comparar prototipos y la manera de calcular los centros de los grupos. El método *k-means* usa una función de distancia para comparar prototipos mientras que *kMSF* usa una función de similaridad Γ . Por otra parte, *k-means* calcula la media de un grupo para determinar el centro de éste, mientras que *k-MSF* calcula un prototipo representativo de cada grupo.

Para determinar el prototipo representativo p_j^r de un grupo C_j, *kMSF* usa la siguiente expresión:

$$r_{\substack{C_i \\ p_j \in C_i \\ C_q \neq C_i}}(p_j) = \frac{\beta_{C_i}(p_j)}{\alpha_{C_i}(p_j) + (1 - \beta_{C_i}(p_j))} + \eta_{C_q}(p_j) \tag{3.1}$$

Donde:

$$\beta_{C_i}(p_j) = \frac{1}{|C_i| - 1} \sum_{\substack{p_j, p_q \in C_i \\ p_j \neq p_q}} \Gamma(p_j, p_q) \tag{3.2}$$

$$\alpha_{C_i}(p_j) = \frac{1}{|C_i| - 1} \sum_{\substack{p_j, p_q \in C_i \\ p_j \neq p_q}} \left| \beta_{C_i}(p_j) - \Gamma(p_j, p_q) \right| \tag{3.3}$$

$$\eta_{C_k}(p_j) = \sum_{\substack{q=1 \\ i \neq q}}^{c} (1 - \Gamma(p_q^r, p_j)) \tag{3.4}$$

$\Gamma(p_j, p_q)$ es la similaridad entre los prototipos p_j y p_q.

p_q^r es el prototipo representativo del grupo C_q.

c es el número de grupo.

$\beta_{C_i}(p_j)$ es la similaridad promedio de p_j con los demás prototipos en el grupo C_i.

$\alpha_{C_i}(p_j)$ evalúa la varianza entre $\beta_{C_i}(p_j)$ y la similaridad de p_j con los demás prototipos en C_i.

$\eta_{C_k}(p_j)$ es la disimilaridad promedio de p_j con los demás prototipos representativos.

Entonces, el prototipo representativo en C_i es aquél p_i^r tal que:

i) p_i^r es el más similar (en promedio) con los otros prototipos en el grupo.

ii) p_i^r es el más disimilar con respecto a los demás prototipos representativos.

Las propiedades i) y ii) dependen directamente de los valores de $\beta_{C_i}(p_j)$ y $\eta_{C_k}(p_j)$, por lo que p_i^r es aquél prototipo que maximiza la expresión (3.1).

PSC (Algoritmo 3.3) comienza creando c grupos de prototipos a partir de T. Una vez que los grupos $C_1, C_2,\ldots,C_c$ han sido creados, es necesario determinar si cada uno de ellos es homogéneo o no.

PSC (Conjunto de entrenamiento T, número de grupos c):
$S=\emptyset$
Grupos= *kMSF*(T, c) // generar c grupos a partir de T
Para cada grupo C_j en *Grupos*
  Si A_j es no homogéneo entonces
    Encontrar la clase mayoritaria c_M en C_j
    Para cada $p_i \in C_j$. $p_j \notin c_M$
      Encontrar $p_c \in c_M$ el prototipo más similar a p_i con clase c_M
      $S = S \cup \{ p_i \}$
      Encontrar p_M, el prototipo más similar a p_c con clase distinta a c_M
      $S = S \cup \{ p_M \}$
  De lo contrario // C_j es homogéneo
    p_i = prototipo representativo de C_j
    $S = S \cup \{ p_i \}$
Regresar S

Algoritmo. 3.3. Método *PSC* para la selección de prototipos

Si C_j es no homogéneo entonces los prototipos en C_j están situados en una región crítica, por lo que algunos prototipos en C_j son prototipos frontera. Para detectar prototipos frontera, *PSC* encuentra los prototipos que pertenecen a la clase mayoritaria (la clase más frecuente en C_j) ya que estos prototipos son cercanos a una frontera delimitada por las clases minoritarias en C_j. Una vez detectados los prototipos pertenecientes a la clase mayoritaria, los prototipos frontera en la clase mayoritaria son los prototipos más similares a cada prototipo de las clases minoritarias; mientras que los prototipos frontera en las clases minoritarias son los prototipos más similares a cada prototipo frontera de la clase mayoritaria.

Por otra parte, si C_j es homogéneo entonces los elementos en C_j son prototipos interiores de la clase, es decir, estos prototipos no están situados en regiones críticas, por lo que podrían descartarse de T pero algunos de estos prototipos son necesarios para representar las regiones interiores de cada clase. Por lo tanto, *PSC* encuentra el prototipo representativo p_j^r de C_j y descarta los restantes prototipos de tal manera que C_j es representado por p_j^r.

Finalmente, los prototipos seleccionados por el método *PSC* son los prototipos representativos de cada grupo homogéneo y los prototipos frontera detectados en cada grupo no homogéneo mediante el proceso descrito anteriormente.

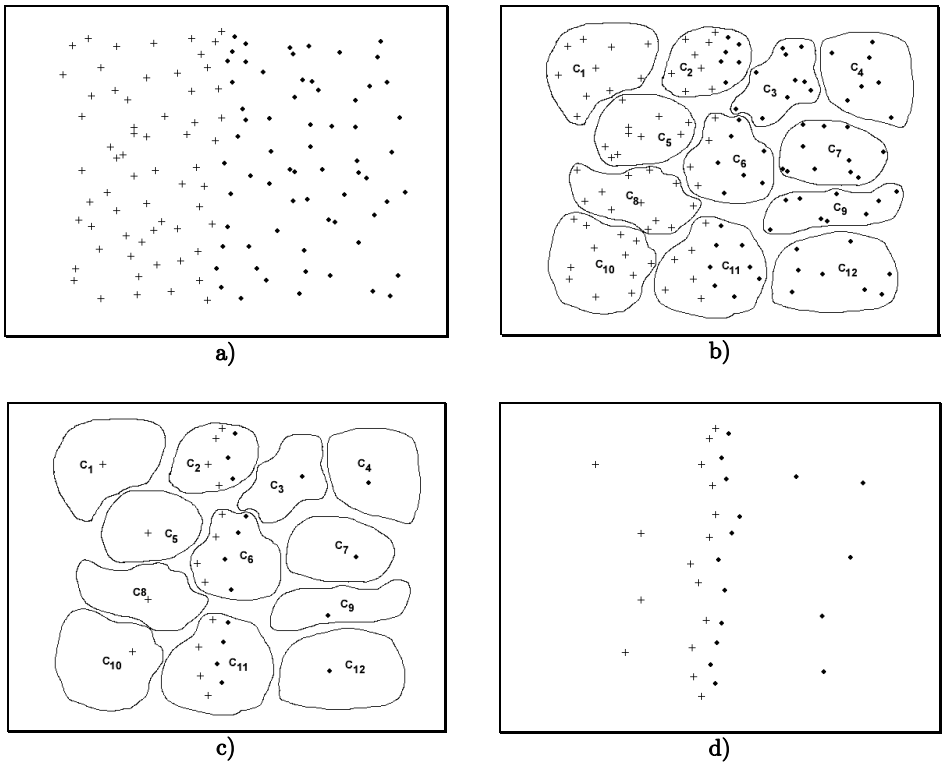


Figura 3.4. **a)** Conjunto de datos con clases "+" y "•". **b)** Grupos creados. **c)** Prototipos seleccionados en cada grupo. **d)** Conjunto de prototipos seleccionado por *PSC*

Para ilustrar el proceso de selección de *PSC*, considérese el ejemplo mostrado en la figura 3.4a, en la que se presenta el mismo conjunto de datos bi-dimensional mostrado en la figura 3.3a. Los grupos $(C_1,\ldots,C_{12})$ generados se muestran en la figura 3.4b en la que los de tipo no homogéneos son C_2, C_6 y C_{11}.

En los grupos C_6 y C_{11}, la clase minoritaria es "+" por lo tanto, los prototipos frontera de la clase mayoritaria (•) son los más similares a cada prototipo de clase minoritaria (+). Por otra parte, los prototipos frontera en la clase "+" son los prototipos más similares (de clase "+") a cada prototipo frontera en la clase "•".

En C_2, la clase minoritaria es "•" y *PSC* sigue el mismo proceso de selección descrito anteriormente. En la figura 3.4c se muestran los prototipos seleccionados en cada grupo y en la figura 3.4d se muestran los prototipos seleccionados por *PSC*.

3.3 PSR (*Prototype Selection by Relevance*)

En un conjunto de entrenamiento algunos prototipos son más relevantes o representativos que otros de su misma clase. De aquí en adelante, como se detallará en párrafos posteriores, se considerará como el prototipo más relevante p_r a aquél que es en promedio más parecido a los demás prototipos de su misma clase. En este sentido, en términos de selección de prototipos, sería deseable seleccionar aquellos prototipos con mayor relevancia o peso informacional en cada clase.

En esta sección, se presenta el método *filter PSR* (*Prototype Selection by Relevance*), el cual, consiste en calcular la relevancia de cada prototipo en T y seleccionar aquellos con mayor relevancia. Para preservar las regiones frontera, *PSR* también selecciona algunos prototipos frontera detectados a partir de los prototipos con mayor relevancia.

La fase inicial de *PSR* consiste en calcular los pesos de relevancia de cada $p_i \in T$. Una vez que se ha calculado la relevancia de cada prototipo, para cada

clase j, *PSR* selecciona los q prototipos con mayor relevancia y a través de ellos, se seleccionan prototipos frontera de manera análoga a *PSC*, es decir, los prototipos frontera son aquellos más similares a cada uno de los q prototipos más relevantes pero con clases distintas. En el algoritmo 3.4 se muestra el correspondiente al método *PSR*.

PSR (Conjunto de entrenamiento T, número de prototipos relevantes q a considerar)
$S = \emptyset$
$R = \emptyset$
Para cada $p_i \in T$ // Calcular la relevancia de cada prototipo
  $RW[i] = Relevancia(p_i)$
Para cada clase c_j // Encontrar los q prototipos más relevantes de cada clase
  $Prot\_C$ = Conjunto de prototipos en T con clase c_j
  Rel_Prot = Los q prototipos en $Prot\_C$ con mayor relevancia en RW
  $R = R \cup Rel\_Prot$
  Para cada $p_i \in Rel\_Prot$ // Encontrar prototipos frontera
    Para cada clase $c_k \neq c_j$
      Encontrar p_s, el prototipo más similar a p_j con clase c_k
      $S = S \cup \{p_s\}$
$S = S \cup R$
Regresar S

Algoritmo. 3.4. Método *PSR* para la selección de prototipos

Para ilustrar el proceso de selección de *PSR* considérese la figura 3.5, en la que se muestra el resultado obtenido al aplicar *PSR* al mismo conjunto de datos sintético utilizado previamente. La figura 3.5b muestra el *30%* de los prototipos más relevantes de cada clase. A partir de los prototipos de las figuras 3.5a y 3.5b, se seleccionan prototipos frontera (figura 3.5c) y finalmente el conjunto seleccionado por *PSR* se muestra en la figura 3.5d.

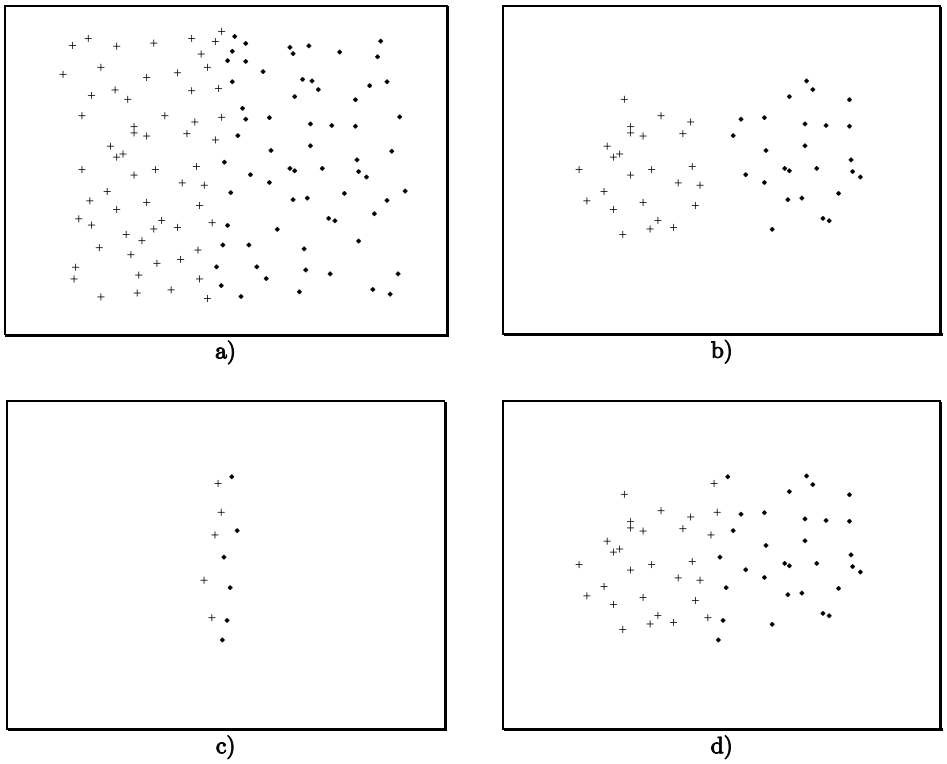


Figura. 3.5. a) Conjunto de datos con clases "+" y "•". **b)** 30% de los prototipos más relevantes **c)** Prototipos frontera seleccionados a partir de los prototipos de **a)** y **b)** . **d)** Conjunto de prototipos seleccionado por *PSR*

En los resultados mostrados en la figura 3.5, la relevancia de un prototipo p está dada en términos de la similaridad promedio que p tiene con los demás prototipos de su misma clase, es decir, el prototipo más similar a todos los de su clase C_j es el más representativo de la clase C_j. Con base en lo anterior, la similaridad promedio (A_s) de p se calcula mediante la siguiente expresión:

$$A_s(p) = \frac{\sum_{p' \in C, p \neq p'} S(p, p')}{\mid C \mid -1} \tag{3.5}$$

Donde:

C es el conjunto de prototipos con la misma clase que p.

$S(p, p')$ es una función de comparación entre prototipos.

Capítulo 4

Resultados experimentales

En este capítulo se presentan los resultados experimentales obtenidos al aplicar los métodos propuestos en el capítulo sobre distintos conjuntos de datos; se detallan los parámetros utilizados en los experimentos y se presenta una comparación entre los métodos propuestos y otros métodos relevantes existentes en la literatura.

4.1 Descripción de experimentos

Los métodos que se proponen en el capítulo anterior fueron aplicados a distintos conjuntos de datos obtenidos del repositorio *Machine Learning Database* de la universidad de California, Irvine [Asunción & Newman, 2007]. En la tabla 4.1 se listan estos conjuntos o bases de datos, especificando el número total de prototipos, atributos y clases correspondientes a cada conjunto de datos. Además se indica, con respecto al número total de atributos, el porcentaje de atributos los cuales son de tipo numérico (%Num) y los de tipo no numérico (%Non).

Tabla 4.1. Características de los conjuntos de datos utilizados en los experimentos

Conjunto de datos	Prototipos	Atributos	%Num	%Non	Clases
Bridges	108	11	9	91	7
Echocardiogram	132	12	69	31	2
Glass	214	10	100	0	7
Hearth Cleveland	303	13	38	62	2
Hepatitis	155	19	32	68	2
Iris	150	4	100	0	3
Letter	20000	16	100	0	26
Segmentation	2100	19	100	0	7
Shuttle Statlog	58000	9	100	0	7
Liver	345	6	100	0	2
UPS	9000	255	100	0	10
Wine	178	13	100	0	3
Zoo	90	16	12	88	7

La manera más común de evaluar resultados de clasificación es mediante conjuntos de prueba y entrenamiento. En todos los experimentos reportados en este capítulo, los conjuntos de prueba y entrenamiento fueron construidos empleando validación cruzada (*k-fold cross stratified validation*), específicamente *10 fold cross stratified validation*.

La validación cruzada consiste en dividir de manera aleatoria cada conjunto de datos en k bloques (de aproximadamente igual tamaño y mutuamente excluyentes), de los cuales k-1 partes se utilizan como conjunto de entrenamiento y la parte

restante se utiliza como conjunto de prueba. Cada una de las k partes resultantes de la división de la base de datos se considera como conjunto de prueba, por lo que se realiza un total de k experimentos por cada base de datos y se reporta el promedio de los k resultados.

Cabe mencionar que en los experimentos realizados en este trabajo, se utilizaron los mismos conjuntos de prueba y entrenamiento para cada método de selección de prototipos, así como el mismo equipo de cómputo[2].

En las tablas mostradas en este capítulo, las columnas *Acc* corresponden a la calidad de clasificación obtenida con S y las columnas *Str* son el porcentaje de retención con respecto a T, es decir:

$$Str = \frac{100 \mid S \mid}{\mid T \mid} \tag{4.1}$$

Además, se incluye la calidad de clasificación obtenida con el conjunto original (*Orig*). Para los resultados de clasificación mostrados en las tablas se llevaron a cabo pruebas estadísticas para determinar si existe diferencia significativa entre los métodos propuestos en este capítulo y los demás métodos. En particular, se utilizó la prueba estadística *k-fold cross validated paired t test* [Dietterich, 1998], en la cual, se calcula la siguiente estadística:

$$t = \frac{\overline{p}\sqrt{n}}{\sqrt{\frac{\sum_{i=1}^{k}(p^{(i)} - \overline{p})^2}{k-1}}} \tag{4.2}$$

Donde:

$$\overline{p} = \frac{1}{k}\sum_{i=1}^{k} p^{(i)} .$$

[2] Los resultados reportados en este capítulo fueron obtenidos utilizando una computadora con procesador Intel Celeron 2.4GHz con 512MB RAM.

$p^{(i)} = p_A^{(i)} - p_B^{(i)}$; $p_A^{(i)}$ y $p_B^{(i)}$ corresponden a la proporción de ejemplos mal clasificados por los métodos *A* y *B*, respectivamente.

En esta prueba, la hipótesis nula corresponde a suponer que los resultados de ambos métodos son iguales. Para determinar si la hipótesis nula se rechaza (los resultados no son iguales) se utiliza la distribución *t* de Student con *k-1* grados de libertad (siendo *k* el número de pliegues de la validación cruzada) y un nivel de confianza *nc*, por lo que si $|t| > t_{k-1,nc}$ se puede concluir que los dos resultados a comparar son significativamente diferentes con un *nc*% de confianza. Para los resultados experimentales mostrados en este capítulo, se utilizó un nivel de confianza de 99%. En cada tabla se especifica con letra negrita el método contra el cual se determina si existe diferencia significativa y de existir, se indica con el símbolo "*".

Para los resultados promedio reportados en cada tabla, se muestra su correspondiente gráfica de dispersión de retención (eje vertical) contra calidad de clasificación (eje horizontal). En este tipo de gráfica, el mejor método con respecto a la clasificación es aquél situado más a la derecha con respecto a los demás, mientras que el método más cercano al eje horizontal es el mejor con respecto a la retención. En estas gráficas se indican (con el símbolo "□") los puntos que forman el frente de Pareto. Este tipo de frente se utiliza para evaluar funciones multi objetivo, en el contexto de selección de prototipos se tienen dos objetivos: clasificación y retención, por lo que el frente de Pareto se forma por los puntos con los mejores valores en ambos objetivos.

4.2 Función de comparación entre prototipos

Para determinar el parecido o similitud entre prototipos es necesario utilizar alguna función mediante la cual se evalúe tal similitud. Para los experimentos

reportados en este trabajo, la función para comparar prototipos fue *HVDM* (*Heterogeneous Value Difference Metric*) [Wilson & Martínez, 2000], la cual permite comparar prototipos descritos por datos mezclados y se define de la siguiente manera:

$$HVDM(X,Y) = \sqrt{\sum_{a=1}^{A} d_a^{\,2}(x,y)} \tag{4.3}$$

Donde:

$d_a(x,y)$ es la distancia para el atributo *a,* y se define como:

$$d_a(x,y) = \begin{cases} 1 & \text{si } x = ?\ o\ y = ? \\ vdm_a(x,y) & \text{si } a \text{ es no numérico} \\ \dfrac{|x-y|}{4\sigma_a} & \text{si } a \text{ es numérico} \end{cases} \tag{4.4}$$

"?" indica un valor faltante del atributo.

σ_a es la desviación estándar para el atributo *a.*

$vdm_a(x,y)$ se define como:

$$vdm_a(x,y) = \sum_{i=1}^{c} \left(\frac{N_{a,x,i}}{N_{a,x}} - \frac{N_{a,y,i}}{N_{a,y}} \right)^2 \tag{4.5}$$

En la expresión mostrada en (4.5), $N_{a,x}$ es el número de veces que *a* tiene valor *x* en *T*; $N_{a,x,i}$ es el número de veces que *a* toma valor *x* en la clase *i*.

4.3 Resultados Experimentales con *RFPS* y *RFPS-Inv*

En esta sección se reportan los resultados obtenidos al aplicar *RFPS* y *RFPS-Inv* sobre diferentes conjuntos de datos.

En las tablas 4.2-4.3 se muestran los resultados de clasificación y retención obtenidos con los métodos *RFPS, RFPS-Inv, DROP3, DROP5* (los mejores

métodos *DROP* reportados en [Wilson & Martínez, 2000]), *GCNN* (de acuerdo a sus autores, éste es competitivo contra los métodos *DROP*), *ENN+BSE* y *DROP+BSE* (métodos basados en búsqueda secuencial). En estas tablas se reportan los resultados obtenidos con *RFPS* utilizando los métodos *ENN* y *DROP* en el paso de pre-procesado. En estas tablas, *ENN+RFPS* corresponde a *RFPS* usando *ENN* para el pre-procesamiento y de manera análoga, para *DROP3+RFPS* y *DROP5+RFPS* se utilizaron los métodos *DROP3* y *DROP5* para el pre-procesamiento. Se incluyen los resultados obtenidos con la búsqueda tabú (*TS*), que también es un método *wrapper* *SCC* y según los experimentos reportados en [Bezdek & Kuncheva, 2001], es de los mejores de este tipo.

El clasificador utilizado en los experimentos mostrados en las tablas 4.2-4.3 fue *k-NN* con *k=3,* valor con el que los métodos *DROP* tienen mejor desempeño. En estas tablas, se indica si existe diferencia significativa (con el símbolo "*") con respecto a *ENN+RFPS*, el cual, en cuanto a clasificación, fue el mejor de los métodos secuenciales.

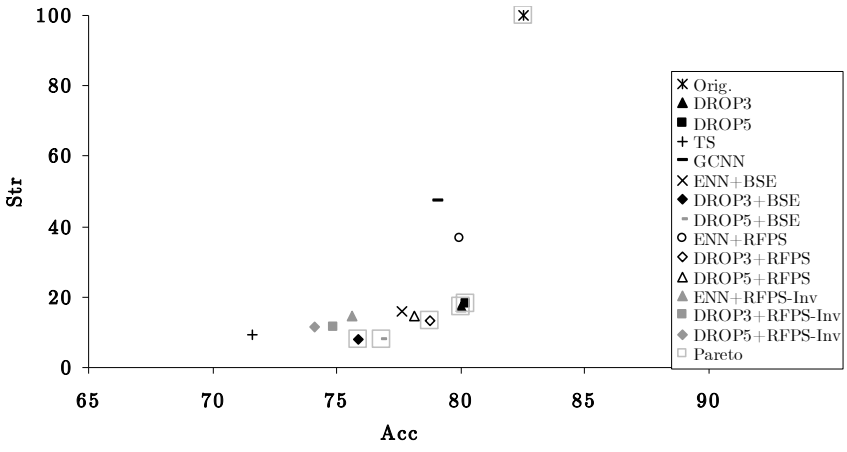


Figura 4.1. Gráfica de dispersión de los resultados mostrados en las tablas 4.2-4.3

Los resultados promedio reportados en las tablas anteriores se muestran en la figura 4.1 en la cual, se presenta la correspondiente gráfica de dispersión.

Con base en los resultados mostrados en las tablas 4.2-4.3 y la figura 4.1, los mejores métodos con respecto a clasificación fueron *DROP3, DROP5* y *ENN+RFPS*, además, *ENN+RFPS* fue mejor que *DROP3* y *DROP5* en los dos conjuntos de datos (*Bridges, Hepatitis*) en que existe diferencia significativa.

Puede observarse que *DROP3+RFPS* forma parte del frente de Pareto y *ENN+RFPS* se sitúa mas cerca del frente con respecto a otros métodos como lo son *TS* y *GCNN*.

Por otra parte, los resultados de clasificación obtenidos con *RFPS* son mejores con respecto a *RFPS-Inv*, lo cual indica que fue mejor realizar primero la exclusión condicional seguida de la inclusión condicional. Esto se debe a que durante la exclusión condicional puede ocurrir que se descarte un conjunto de prototipos que al incluirlos al final (inclusión condicional) ayude a mejorar la calidad de clasificación de *S*. Esto puede también notarse al observar que los resultados obtenidos con *RFPS* son mejores que los obtenidos con *ENN+BSE* y *DROP+BSE*. Nótese que los métodos *ENN+BSE* y *DROP+BSE* corresponden a la fase de exclusión condicional de *RFPS* y al aplicar la inclusión condicional (*RFPS* en su totalidad) la precisión mejora con respecto a solamente aplicar la exclusión condicional, por lo que los subconjuntos obtenidos con este *RFPS* son de mayor tamaño con respecto a *ENN+BSE* y *DROP+BSE*.

Tabla 4.2. Resultados de clasificación (*Acc*) obtenidos con: Conjunto original (*Orig.*), *DROP3, DROP5, ENN+BSE, DROP3+BSE, DROP5+BSE, TS, GCNN, RFPS* y *RFPS-Inv* utiizando *k-NN*

Datos	Orig.	DROP3	DROP5	ENN+BSE	DROP3+BSE	DROP5+BSE	TS	GCNN	ENN+RFPS	DROP3+RFPS	DROP5+RFPS	ENN+RFPS-Inv	DROP3+RFPS-Inv	DROP5+RFPS-Inv
	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc
Bridges	66.09	56.36*	62.82*	58.00	62.90	66.18*	45.90*	68.20	**60.00**	56.45	53.63*	51.54*	55.45*	54.54
Echocardiogram	95.71	92.86	98.42	86.48*	95.43	95.71*	85.71*	93.39	**93.23**	91.96	87.67*	93.21	87.85*	89.10*
Glass	71.42	66.28	62.16	69.41	59.78	54.24	62.59	69.61	**69.43**	64.48	67.74	58.35*	43.50*	55.49*
Heart Cleveland	82.49	78.89	79.87	73.33	73.45*	76.58	72.63*	67.63	**79.52**	77.21*	79.22	78.74	77.90	76.48*
Hepatitis	79.29	78.13*	75.42*	77.00	75.29*	76.66*	74.33	60.66	**79.00**	78.20*	76.66*	77.95	78.45	74.08
Iris	94.66	95.33	94.00	93.00	88.00	89.33*	70.66*	96.00	**93.33**	93.00	93.33	80.66*	92.66	86.00*
Liver	65.22	67.82	63.46	57.67	59.77	57.95	64.13*	66.09	**59.98**	61.70	60.03	58.84	59.75	60.30
Wine	94.44	94.41	93.86	92.74	90.49	91.04	79.44	94.44	**93.63**	94.44	93.85	90.00	88.23*	91.04
Zoo	93.33	90.00	95.56	91.11	77.77*	83.33	88.88	95.55	**91.33**	91.33*	91.11	91.11	90.00	80.00*
Promedio	**82.52**	**80.01**	**80.22**	**77.64**	**75.88**	**76.78**	**71.59**	**79.06**	**79.94**	**78.75**	**78.14**	**75.60**	**74.87**	**74.11**

Tabla 4.3. Resultados de retención correspondientes a la tabla 4.2

Datos	Orig.	DROP3	DROP5	ENN+BSE	DROP3+BSE	DROP5+BSE	TS	GCNN	ENN+RFPS	DROP3+RFPS	DROP5+RFPS	ENN+RFPSInv	DROP3+RFPS-Inv	DROP5+RFPS-Inv
	Str	Str	Str	Str	Str	Str	Str	Str	Str	Str	Str	Str	Str	Str
Bridges	100	14.78	20.66	23.46	6.26	8.11	18.94	88.20	48.10	14.28	23.71	26.80	12.37	19.58
Echocardiogram	100	13.95	14.87	6.00	6.00	6.00	7.46	22.67	86.56	9.85	8.16	11.94	8.95	7.46
Glass	100	24.35	25.91	21.81	14.95	15.21	15.98	61.62	29.34	25.75	26.11	19.47	23.36	16.45
Heart Cleveland	100	11.44	14.59	20.60	7.18	7.80	4.54	9.09	22.41	12.61	12.53	13.60	7.77	9.12
Hepatitis	100	11.47	15.05	16.49	4.22	4.37	5.73	17.75	18.71	8.38	8.67	11.47	6.66	5.23
Iris	100	15.33	12.44	8.00	6.42	6.39	6.50	38.00	10.07	9.92	10.00	5.33	7.18	6.29
Liver	100	26.83	30.59	26.69	10.91	11.75	5.21	83.70	33.68	16.94	19.64	21.80	19.25	19.54
Wine	100	15.04	10.55	8.17	5.05	4.43	6.10	78.89	10.23	8.17	8.30	4.36	5.18	4.42
Zoo	100	20.37	18.77	12.59	11.72	7.76	14.12	26.17	71.14	14.81	14.93	15.92	13.58	14.19
Promedio	**100**	**17.06**	**18.16**	**15.88**	**8.08**	**7.98**	**9.40**	**47.34**	**36.70**	**13.41**	**14.67**	**14.52**	**11.59**	**11.36**

En los resultados anteriores el clasificador utilizado fue *k-NN*, por lo que otro experimento que se llevó a cabo fue evaluar los subconjuntos obtenidos por los métodos comparados en las tablas anteriores utilizando los clasificadores *LWR, SVM, C4.5* y *Naive Bayes* (*NB*)[3]. Cabe mencionar que para *LWR* y *SVM*, de acuerdo a los códigos fuente utilizados, únicamente se usaron los conjuntos de datos para los que se pueden aplicar estos clasificadores (conjuntos de datos descritos por atributos numéricos y sin ausencia de información). Los resultados obtenidos se muestran en las tablas 4.4-4.11 y figuras 4.2-4.5.

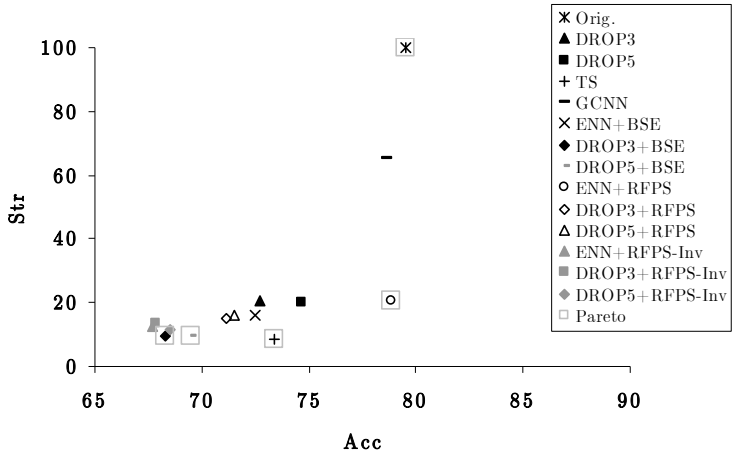


Figura 4.2. Gráfica de dispersión de los resultados obtenidos al utilizar *LWR* (tablas 4.4-4.5)

Puede notarse que, en el caso promedio (figura 4.2), utilizando *LWR*, los mejores métodos en calidad de clasificación fueron *ENN+RFPS* y *GCNN* respectivamente y entre ellos sólo existe diferencia significativa en dos casos (*Iris, Wine*). Sin embargo, *ENN+RFPS* obtuvo mejores resultados de retención con respecto a *GCNN*. Por otra parte, puede notarse que para este clasificador, los métodos *DROP* no forman parte del frente de Pareto, el cual es claramente modificado por *ENN+RFPS*.

[3] El código fuente del clasificador *SVM* fue obtenido de [Vojtech &Václav, 2004], mientras que para *C4.5* y *NB, se utilizó WEKA 3.5.6* [Witten & Frank, 2005].

Tabla 4.4. Resultados de clasificación obtenidos al utilizar los subconjuntos obtenidos por *DROP3, DROP5, ENN+BSE, DROP3+BSE, DROP5+BSE, TS, GCNN, RFPS* y *RFPS-Inv* como entrenamiento para *LWR*

Datos	Orig.	DROP3	DROP5	ENN+BSE	DROP3+BSE	DROP5+BSE	TS	GCNN	ENN+RFPS	DROP3+RFPS	DROP5+RFPS	ENN+RFPS-Inv	DROP3+RFPS-Inv	DROP5+RFPS-Inv
	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc
Glass	57.85	51.66*	54.06	54.54	42.85*	41.10*	50.86*	56.88	**55.47**	45.64*	48.00*	45.65*	42.90*	47.61*
Iris	98.00	92.00*	92.00*	76.66*	78.00*	78.00*	93.33*	95.33*	**98.00**	82.00*	77.33*	73.33*	79.33*	76.00*
Liver	70.13	68.63	68.95	69.79	68.97	70.13	68.40*	70.13	**70.43**	69.98	71.87	68.36	69.84	67.21
Wine	92.15	78.53*	83.63*	88.88*	83.33*	88.88*	80.93*	92.15	**91.50**	87.05*	88.88*	83.33*	79.44*	83.33*
Promedio	**79.53**	**72.71**	**74.66**	**72.47**	**68.29**	**69.53**	**73.38**	**78.62**	**78.85**	**71.17**	**71.52**	**67.67**	**67.88**	**68.54**

Tabla 4.5. Resultados de retención correspondientes a la tabla 4.4

Datos	Orig.	DROP3	DROP5	ENN+BSE	DROP3+BSE	DROP5+BSE	TS	GCNN	ENN+RFPS	DROP3+RFPS	DROP5+RFPS	ENN+RFPSInv	DROP3+RFPS-Inv	DROP5+RFPS-Inv
	Str	Str	Str	Str	Str	Str	Str	Str	Str	Str	Str	Acc	Str	Str
Glass	100	24.35	25.91	21.81	14.95	15.21	15.98	61.62	29.34	25.75	26.11	19.47	23.36	16.45
Iris	100	15.33	12.44	8.00	6.42	6.39	6.5	38.00	10.07	9.92	10.00	5.33	7.18	6.29
Liver	100	26.83	30.59	26.69	10.91	11.75	5.21	83.70	33.68	16.94	19.64	21.80	19.25	19.54
Wine	100	15.04	10.55	8.17	5.05	4.43	6.1	78.89	10.23	8.17	8.30	4.36	5.18	4.42
Promedio	**100**	**20.39**	**19.87**	**16.17**	**9.33**	**9.45**	**8.45**	**65.55**	**20.83**	**15.20**	**16.01**	**12.74**	**13.74**	**11.68**

Tabla 4.6. Resultados de clasificación obtenidos al utilizar los subconjuntos obtenidos por *DROP3, DROP5, ENN+BSE, DROP3+BSE, ROP5+BSE, TS, GCNN, RFPS* y *RFPS-Inv* como entrenamiento para *SVM*

Datos	Orig.	DROP3	DROP5	ENN+BSE	DROP3+BSE	DROP5+BSE	TS	GCNN	ENN+RFPS	DROP3+RFPS	DROP5+RFPS	ENN+RFPS-Inv	DROP3+RFPS-Inv	DROP5+RFPS-Inv
	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc
Glass	72.29	68.48	63.99	61.06*	56.87*	54.71*	66.09*	66.82*	**71.35**	62.47*	63.54*	63.09	54.74*	58.82*
Iris	96.00	94.00*	95.33	94.00	90.00*	95.33	94.66	94.66	**96.00**	92.00*	94.66	93.33	92.66	92.00
Liver	69.91	68.26	68.56	57.97	55.97	57.97	69.73	69.73	**67.97**	58.56*	57.97	57.97	56.56	58.56
Wine	97.18	95.33	91.01	95.61	94.75*	91.11*	83.26	94.83*	**97.18**	95.33*	97.18*	95.05	94.44	94.11
Promedio	**83.85**	**81.52**	**79.72**	**77.16**	**74.40**	**74.78**	**78.44**	**81.51**	**83.13**	**77.09**	**78.34**	**77.36**	**74.6**	**75.87**

Tabla 4.7. Resultados de retención correspondientes a la tabla 4.6

Datos	Orig.	DROP3	DROP5	ENN+BSE	DROP3+BSE	DROP5+BSE	TS	GCNN	ENN+RFPS	DROP3+RFPS	DROP5+RFPS	ENN+RFPSInv	DROP3+RFPS-Inv	DROP5+RFPS-Inv
	Str	Str	Str	Str	Str	Str	Str	Str	Str	Str	Str	Acc	Str	Str
Glass	100	24.35	25.91	21.81	14.95	15.21	15.98	61.62	29.34	25.75	26.11	19.47	23.36	16.45
Iris	100	15.33	12.44	8.00	6.42	6.39	6.5	38.00	10.07	9.92	10.00	5.33	7.18	6.29
Liver	100	26.83	30.59	26.69	10.91	11.75	5.21	83.70	33.68	16.94	19.64	21.80	19.25	19.54
Wine	100	15.04	10.55	8.17	5.05	4.43	6.1	78.89	10.23	8.17	8.30	4.36	5.18	4.42
Promedio	**100**	**20.39**	**19.87**	**16.17**	**9.33**	**9.45**	**8.45**	**65.55**	**20.83**	**15.20**	**16.01**	**12.74**	**13.74**	**11.68**

Tabla 4.8. Resultados de clasificación obtenidos al utilizar los subconjuntos obtenidos por *DROP3, DROP5, ENN+BSE, DROP3+BSE, ROP5+BSE, TS, GCNN, RFPS* y *RFPS-Inv* como entrenamiento para *C4.5*

Datos	Orig.	DROP3	DROP5	ENN+BSE	DROP3+BSE	DROP5+BSE	TS	GCNN	ENN+RFPS	DROP3+RFPS	DROP5+RFPS	ENN+RFPS-Inv	DROP3+RFPS-Inv	DROP5+RFPS-Inv
	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc
Bridges	65.81	47.90*	39.54*	53.27	45.45	40.00	44.90*	52.36*	**54.27**	54.54	56.54*	52.00	46.63	41.27*
Echocardiogram	95.71	84.10	92.85	85.71*	79.82	79.65*	85.00*	91.78	**93.21**	87.67	90.17*	90.46	87.50*	90.00
Glass	67.29	60.19	53.76*	60.64	50.90*	51.45*	48.59*	60.75*	**66.66**	55.19*	62.22*	43.83*	48.67*	47.68*
Heart Cleveland	71.96	68.59	72.16	68.65*	66.90*	70.25	67.67*	66.00	**70.60**	67.61*	70.60	70.00*	66.69*	70.89
Hepatitis	76.70	63.33*	63.41*	75.00	60.66	51.20	52.33*	65.15	**76.04**	71.62*	62.79*	73.33	70.29*	68.20*
Iris	93.99	92.66	90.66	80.00*	80.00*	75.33*	71.33*	88.66	**93.99**	84.00	86.00	89.33	80.66*	84.33*
Liver	63.67	59.48*	63.67	56.58*	58.26*	61.73	57.10*	61.76	**63.08**	57.15*	58.54*	57.74*	58.82	56.76*
Wine	94.44	84.43*	78.88*	88.23	70.58*	66.66*	56.99*	95.55	**91.53**	74.21*	76.86*	78.66*	67.54*	68.56*
Zoo	93.33	81.10	88.88	82.22	61.11*	61.11*	63.33*	81.10	**87.77**	72.22*	67.77*	88.88	60.00*	66.11*
Promedio	**80.32**	**71.31**	**71.53**	**72.26**	**63.74**	**61.93**	**60.80**	**73.68**	**77.46**	**69.36**	**70.17**	**71.58**	**65.20**	**65.98**

Tabla 4.9. Resultados de retención correspondientes a la tabla 4.8

Datos	Orig.	DROP3	DROP5	ENN+BSE	DROP3+BSE	DROP5+BSE	TS	GCNN	ENN+RFPS	DROP3+RFPS	DROP5+RFPS	ENN+RFPSInv	DROP3+RFPS-Inv	DROP5+RFPS-Inv
	Str	Str	Str	Str	Str	Str	Str	Str	Str	Str	Str	Str	Str	Str
Bridges	100	14.78	20.66	23.46	6.26	8.11	18.94	88.20	48.10	14.28	23.71	26.80	12.37	19.58
Echocardiogram	100	13.95	14.87	6.00	6.00	6.00	7.46	22.67	86.56	9.85	8.16	11.94	8.95	7.46
Glass	100	24.35	25.91	21.81	14.95	15.21	15.98	61.62	29.34	25.75	26.11	19.47	23.36	16.45
Heart Cleveland	100	11.44	14.59	20.60	7.18	7.80	4.54	9.09	22.41	12.61	12.53	13.60	7.77	9.12
Hepatitis	100	11.47	15.05	16.49	4.22	4.37	5.73	17.75	18.71	8.38	8.67	11.47	6.66	5.23
Iris	100	15.33	12.44	8.00	6.42	6.39	6.50	38.00	10.07	9.92	10.00	5.33	7.18	6.29
Liver	100	26.83	30.59	26.69	10.91	11.75	5.21	83.70	33.68	16.94	19.64	21.80	19.25	19.54
Wine	100	15.04	10.55	8.17	5.05	4.43	6.10	78.89	10.23	8.17	8.30	4.36	5.18	4.42
Zoo	100	20.37	18.77	12.59	11.72	7.76	14.12	26.17	71.14	14.81	14.93	15.92	13.58	14.19
Promedio	**100**	**17.06**	**18.16**	**15.88**	**8.08**	**7.98**	**9.40**	**47.34**	**36.70**	**13.41**	**14.67**	**14.52**	**11.59**	**11.36**

Tabla 4.10. Resultados de clasificación obtenidos al utilizar los subconjuntos obtenidos por *DROP3, DROP5, ENN+BSE, DROP3+BSE, ROP5+BSE, TS, GCNN, RFPS* y *RFPS-Inv* como entrenamiento para *NB*

Datos	Orig.	DROP3	DROP5	ENN+BSE	DROP3+BSE	DROP5+BSE	TS	GCNN	ENN+RFPS	DROP3+RFPS	DROP5+RFPS	ENN+RFPS-Inv	DROP3+RFPS-Inv	DROP5+RFPS-Inv
	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc
Bridges	64.00	49.61	39.81	51.63*	46.81	38.81*	41.36*	44.36*	**58.18**	50.81	52.90	54.45	51.54*	50.90
Echocardiogram	97.14	78.31*	77.32*	48.39*	67.67*	77.32	81.25	91.78	**97.32**	83.21	79.64*	84.10	83.90	76.25
Glass	48.05	49.56*	47.74*	47.61	40.15*	42.55	50.95	47.57*	**52.33**	43.98*	46.70	35.97*	41.12*	42.64*
Heart Cleveland	83.81	78.20	81.16	80.84	71.25*	78.25	69.66*	75.52	**80.84**	78.17	80.56	78.75	74.23	79.55
Hepatitis	84.58	61.83*	54.20*	83.91	54.29	56.41	79.29	65.87*	**85.29**	75.33	69.87*	76.91*	68.37*	61.41*
Iris	95.33	93.99	95.33	88.66	69.99*	80.66	84.44*	91.99	**95.33**	87.33	90.66	89.99	75.33*	76.00
Liver	56.02	61.50	61.77*	53.11	54.57	51.92	61.11*	56.88	**57.78**	56.33	53.40	55.18	52.21*	48.78
Wine	98.81	61.11*	66.66*	83.33	57.77*	77.77*	61.40*	96.66	**96.66**	78.33	76.66	82.67	72.94	67.41
Zoo	95.55	88.88	83.33*	92.22	85.55	84.44	93.82	93.33	**93.33**	86.66*	91.11	92.22	93.33	87.77*
Promedio	**80.37**	**69.22**	**67.48**	**69.97**	**60.89**	**65.35**	**69.25**	**73.77**	**79.75**	**71.13**	**71.28**	**72.25**	**68.11**	**65.63**

Tabla 4.11. Resultados de retención correspondientes a la tabla 4.10

Datos	Orig.	DROP3	DROP5	ENN+BSE	DROP3+BSE	DROP5+BSE	TS	GCNN	ENN+RFPS	DROP3+RFPS	DROP5+RFPS	ENN+RFPSInv	DROP3+RFPS-Inv	DROP5+RFPS-Inv
	Str	Str	Str	Str	Str	Str	Str	Str	Str	Str	Str	Acc	Str	Str
Bridges	100	14.78	20.66	23.46	6.26	8.11	18.94	88.20	48.10	14.28	23.71	26.80	12.37	19.58
Echocardiogram	100	13.95	14.87	6.00	6.00	6.00	7.46	22.67	86.56	9.85	8.16	11.94	8.95	7.46
Glass	100	24.35	25.91	21.81	14.95	15.21	15.98	61.62	29.34	25.75	26.11	19.47	23.36	16.45
Heart Cleveland	100	11.44	14.59	20.60	7.18	7.80	4.54	9.09	22.41	12.61	12.53	13.60	7.77	9.12
Hepatitis	100	11.47	15.05	16.49	4.22	4.37	5.73	17.75	18.71	8.38	8.67	11.47	6.66	5.23
Iris	100	15.33	12.44	8.00	6.42	6.39	6.50	38.00	10.07	9.92	10.00	5.33	7.18	6.29
Liver	100	26.83	30.59	26.69	10.91	11.75	5.21	83.70	33.68	16.94	19.64	21.80	19.25	19.54
Wine	100	15.04	10.55	8.17	5.05	4.43	6.10	78.89	10.23	8.17	8.30	4.36	5.18	4.42
Zoo	100	20.37	18.77	12.59	11.72	7.76	14.12	26.17	71.14	14.81	14.93	15.92	13.58	14.19
Promedio	**100**	**17.06**	**18.16**	**15.88**	**8.08**	**7.98**	**9.40**	**47.34**	**36.70**	**13.41**	**14.67**	**14.52**	**11.59**	**11.36**

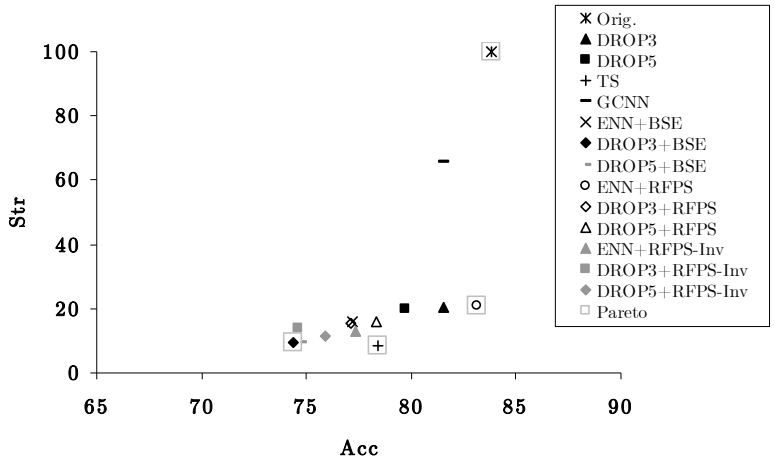


Figura 4.3. Gráfica de dispersión de los resultados obtenidos al utilizar *SVM* (tablas 4.6-4.7)

Para *SVM,* los métodos *ENN+RFPS* y *GCNN* fueron los mejores respectivamente con respecto a clasificación y de igual manera que para *LWR,* los métodos *DROP* y *GCNN* están fuera del frente de Pareto, que es nuevamente modificado por *ENN+RFPS*.

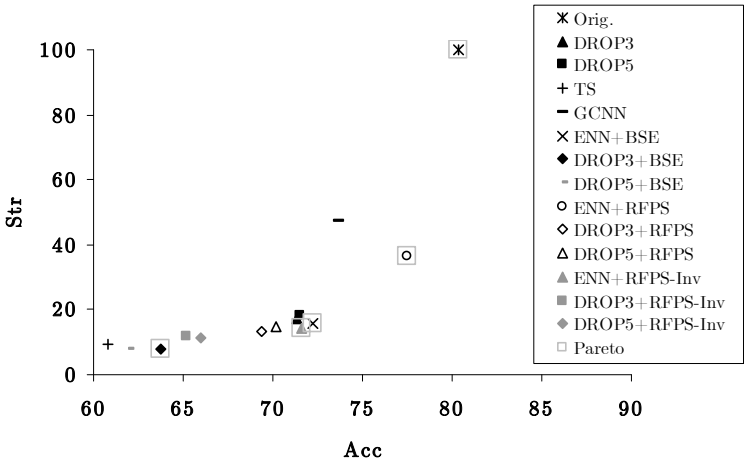


Figura 4.4. Gráfica de dispersión de los resultados obtenidos al utilizar *C4.5* (tablas 4.8-4.9)

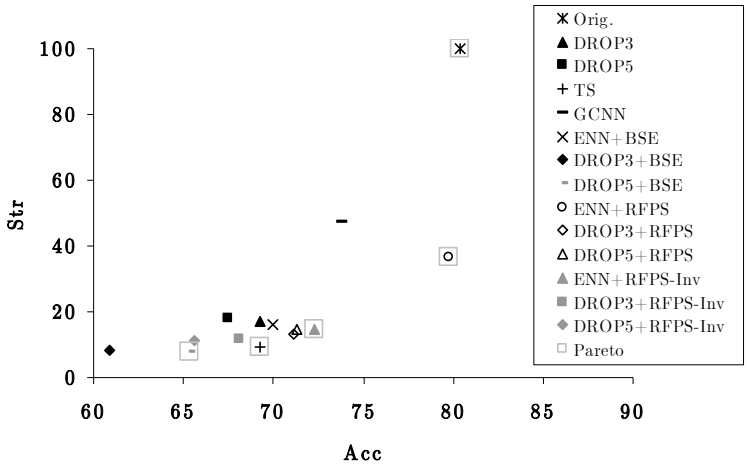


Figura 4.5. Gráfica de dispersión de los resultados obtenidos al utilizar *C4.5* (tablas 410-4.11)

Con base en los resultados obtenidos al utilizar *C4.5* y *NB,* puede notarse que en este experimento el mejor método fue *ENN+RFPS,* superando a *GCNN*, el cual fue de los mejores métodos utilizando *LWR* y *SVM*. En este experimento, de acuerdo al frente de Pareto, *ENN+RFPS* es el mejor método en clasificación con una mejor reducción con respecto a la de *GCNN*.

De estos experimentos, puede notarse que, al evaluar con otros clasificadores la calidad de los subconjuntos obtenidos por los métodos de selección de prototipos, *DROP* y *GCNN* no son los mejores selectores de prototipos, como ocurre cuando se utiliza *k-NN*. Este resultado es de esperarse debido a que el criterio de selección de estos métodos está basado en la regla *k-NN*.

Tabla 4.12. Resultados de clasificación obtenidos por *TS* y *RFPS* utilizando *LWR* durante el proceso de selección

Datos	Orig.	TS	ENN+RFPS	DROP3+RFPS	DROP5+RFPS
	Acc	Acc	Acc	Acc	Acc
Glass	57.85	52.38	**56.84**	50.71	53.72
Iris	98.00	98.00*	**96.66**	93.33	88.67
Liver	70.13	50.00*	**66.34**	68.99*	68.68
Wine	92.15	88.88*	**92.68**	78.51*	85.03*
Promedio	**79.53**	**72.32**	**78.13**	**72.89**	**74.03**

Debido a que *RFPS* (el mejor método restringido flotante en los experimentos anteriores) y *TS* son métodos que permiten utilizar cualquier clasificador durante el proceso de selección, otro experimento consistió en utilizar *LWR, SVM, C4.5* y *NB* durante el proceso de selección de estos

métodos. Los resultados obtenidos se reportan en las tablas 4.12-4.19 y figuras 4.6-4.9.

Tabla 4.13. Resultados de retención correspondientes a la tabla 4.12

Datos	Orig.	TS	ENN+RFPS	DROP3+RFPS	DROP5+RFPS
	Str	Str	Str	Str	Str
Glass	100	6.64	50.26	20.83	21.97
Iris	100	6.22	20.73	10.89	8.15
Liver	100	0.96	36.00	17.13	16.59
Wine	100	2.49	57.51	14.30	8.93
Promedio	**100**	**4.08**	**41.13**	**15.79**	**13.91**

Para *LWR*, los métodos restringidos flotantes fueron mejor en clasificación con respecto a *TS*, aunque este último obtuvo los mejores resultados de retención.

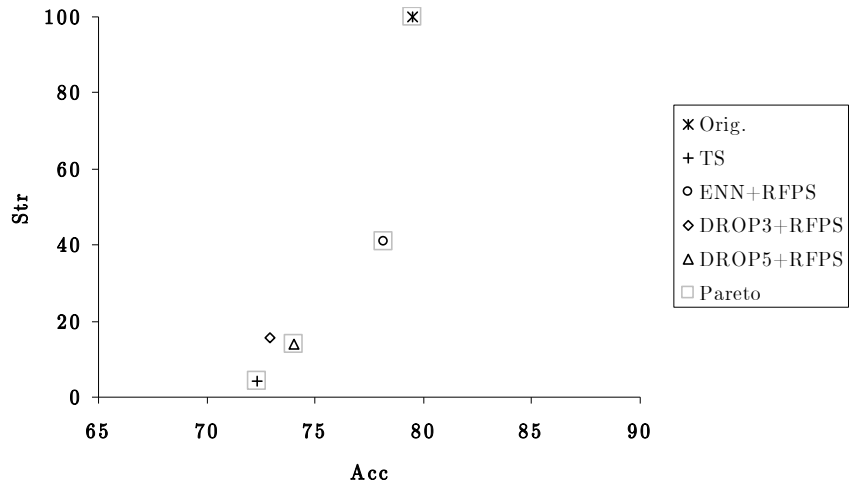


Figura 4.6. Gráfica de dispersión de los resultados de las tablas 4.12-4.13 utilizando *LWR* durante el proceso de selección

Con base en los resultados de las tablas 4.14-4.15 y figura 4.7, utilizando *SVM* durante el proceso de selección, solamente *ENN+RFPS* superó a *TS*, siendo ambos los mejores métodos en este experimento.

Tabla 4.14. Resultados de clasificación obtenidos por *TS* y *RFPS* utilizando *SVM* durante el proceso de selección

Datos	Orig.	TS	ENN+RFPS	DROP3+RFPS	DROP5+RFPS
	Acc	Acc	Acc	Acc	Acc
Glass	72.29	7.75*	**73.82**	68.35	68.90*
Iris	96.00	93.33	**96.00**	93.33	94.67
Liver	69.91	66.87*	**68.67**	62.64	62.66
Wine	97.18	93.88	**96.63**	93.89	92.09
Promedio	**83.85**	**80.46**	**83.78**	**79.55**	**79.58**

Tabla 4.15. Resultados de retención correspondientes a la tabla 4.14

Datos	Orig.	TS	ENN+RFPS	DROP3+RFPS	DROP5+RFPS
	Str	Str	Str	Str	Str
Glass	100	11.36	40.82	17.29	15.42
Iris	100	3.62	8.89	3.70	3.33
Liver	100	3.25	45.84	16.07	18.42
Wine	100	4.93	21.68	3.62	2.75
Promedio	**100**	**5.79**	**29.31**	**10.17**	**9.98**

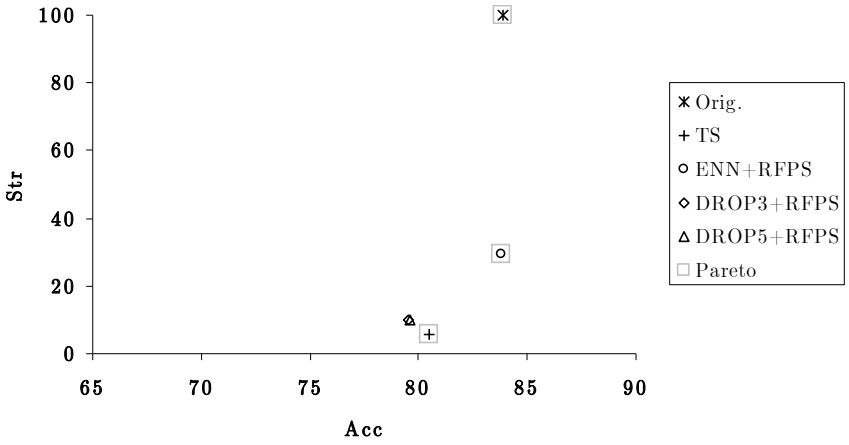


Figura 4.7. Gráfica de dispersión de los resultados de las tablas 4.14-4.15 utilizando *SVM* durante el proceso de selección

Tabla 4.16. Resultados de clasificación obtenidos por *TS* y *RFPS* utilizando *C4.5* durante el proceso de selección

Datos	Orig.	TS	ENN+RFPS	DROP3+RFPS	DROP5+RFPS
	Acc	Acc	Acc	Acc	Acc
Bridges	65.81	54.54	**54.54**	57.36	52.81
Echocardiogram	95.71	91.42	**95.71**	82.50	92.85
Glass	67.29	62.07	**65.92**	58.28	63.36
Heart Cleveland	71.96	69.96	**71.96**	67.04	71.50
Hepatitis	76.70	69.32*	**76.07**	62.13	73.33
Iris	93.99	83.33*	**93.99**	82.00	93.99
Liver	63.67	57.68*	**63.59**	60.57	63.07
Wine	94.44	82.51*	**94.44**	78.89	87.64*
Zoo	93.33	84.44	**88.88**	79.67	78.88
Promedio	**80.32**	**72.81**	**78.34**	**69.83**	**75.27**

Tabla 4.17. Resultados de retención correspondientes a la tabla 4.16

Datos	Orig.	TS	ENN+RFPS	DROP3+RFPS	DROP5+RFPS
	Str	Str	Str	Str	Str
Bridges	100	12.37	32.10	10.27	17.15
Echocardiogram	100	7.96	11.41	6.31	6.90
Glass	100	6.59	25.53	13.02	21.19
Heart Cleveland	100	2.09	28.36	7.55	7.29
Hepatitis	100	2.15	17.71	3.80	5.87
Iris	100	5.48	59.40	4.89	6.96
Liver	100	1.15	42.98	17.45	22.34
Wine	100	4.62	18.54	14.57	7.80
Zoo	100	14.19	20.98	4.93	15.67
Promedio	**100**	**6.29**	**28.56**	**9.20**	**12.35**

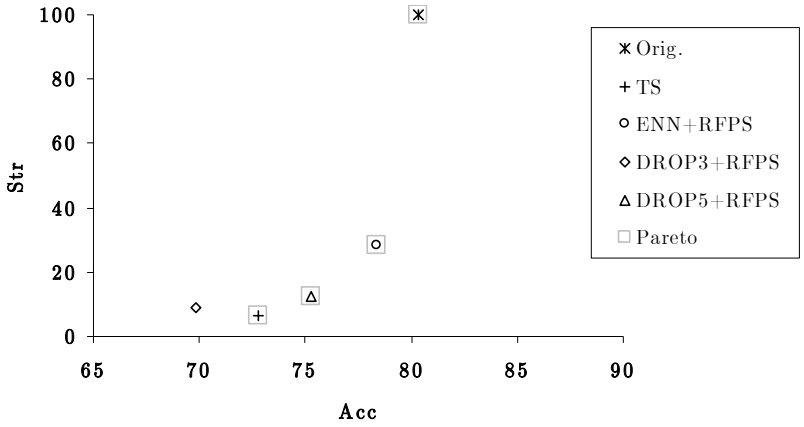


Figura 4.8. Gráfica de dispersión de los resultados de la tablas 4.16-4.17 utilizando *C4.5* durante el proceso de selección

Tabla 4.18. Resultados de clasificación obtenidos por *TS* y *RFPS* utilizando *NB* durante el proceso de selección

Datos	Orig.	TS	ENN+RFPS	DROP3+RFPS	DROP5+RFPS
	Acc	Acc	Acc	Acc	Acc
Bridges	64.00	58.54*	**62.09**	51.52	41.90*
Echocardiogram	97.14	86.42	**94.82**	77.14*	82.67
Glass	48.05	49.11*	**64.59**	56.47*	59.80*
Heart Cleveland	83.81	71.89	**84.16**	71.61	80.51
Hepatitis	84.58	77.33*	**87.20**	70.08	57.95*
Iris	95.33	59.15*	**97.99**	93.33	91.99
Liver	56.02	60.04*	**62.66**	65.57	64.63*
Wine	98.81	86.37*	**96.11**	83.33	79.70*
Zoo	95.55	94.00*	**92.22**	88.88	87.77
Promedio	**80.37**	**71.43**	**82.43**	**73.10**	**71.88**

Tabla 4.19. Resultados de retención correspondientes a la tabla 4.18

Datos	Orig.	TS	ENN+RFPS	DROP3+RFPS	DROP5+RFPS
	Str	Str	Str	Str	Str
Bridges	100	15.83	32.84	9.90	15.70
Echocardiogram	100	9.00	14.86	9.00	7.81
Glass	100	5.45	55.65	20.40	21.96
Heart Cleveland	100	3.22	16.71	7.84	9.24
Hepatitis	100	3.79	28.71	4.44	4.15
Iris	100	1.93	37.85	5.70	6.07
Liver	100	2.09	59.96	24.34	28.69
Wine	100	3.68	49.68	7.99	5.68
Zoo	100	11.11	9.75	8.39	8.14
Promedio	**100**	**6.23**	**34.00**	**10.89**	**11.94**

En los resultados obtenidos con *C4.5* y *NB* durante el proceso de selección (tablas 4.16-4.17, 4.18-4.19; figuras 4.8-4.9), en cuanto a precisión, *TS* fue superado por los métodos restringidos flotantes (excepto por *DROP3+RFPS* para *C4.5*).

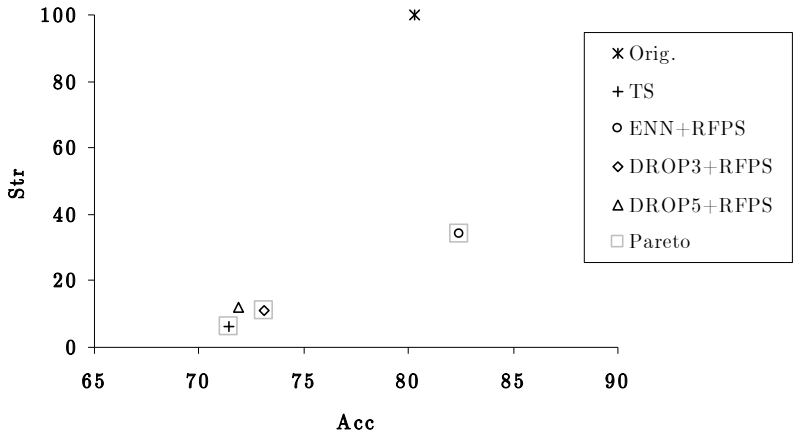


Figura 4.9. Gráfica de dispersión de los resultados de la tablas 4.18-4.19 utilizando *NB* durante el proceso de selección

De acuerdo a estos resultados, los métodos de selección de prototipos restringidos flotantes propuestos en esta sección presentan un buen desempeño cuando se requiere utilizar clasificadores distintos a *k-NN*, por lo que estos métodos restringidos flotantes son una buena opción de selección de prototipos del tipo *wrapper* SCC.

4.3.1 Tiempos de ejecución del método *RFPS*

En esta sección se reportan los tiempos de ejecución del método *RFPS* y de los demás métodos para la selección de prototipos. Los resultados se muestran en la tabla 4.20, la cual muestra el tamaño de cada conjunto de datos y el tiempo (en segundos) que tarda cada método en seleccionar prototipos.

A partir de estos tiempos de ejecución puede notarse que ambos métodos *wrapper SCC* son muy lentos con respecto a los demás métodos. Esta limitante característica de los métodos *wrapper* restringe su uso a problemas con grandes conjuntos de datos. Cabe mencionar que a pesar de que *RFPS* y *TS* son métodos costosos, éste último requiere de un mayor tiempo para llevar a cabo la selección con respecto a *RFPS* y en particular para *DROP3+RFPS* y *DROP5+RFPS*.

Tabla 4.20. Tiempos de ejecución (en segundos) de los métodos *DROP3, DROP5, GCNN, TS* y *RFPS*

Datos	Tamaño			Tiempo						
	Prototipo	Atributos	Clases	DROP3	DROP5	GCNN	TS	ENN+RFPS	DROP3+RFPS	DROP5+RFPS
Echocardiogram	74	9	2	0.19	0.08	1.02	489.12	428.12	4.62	5.18
Zoo	90	16	7	0.26	0.27	1.15	826.58	624.04	13.27	12.97
Bridges	108	11	7	0.16	0.14	15.27	778.96	786.31	27.04	33.08
Iris	150	4	3	0.19	0.16	1.25	514.06	434.61	7.86	6.52
Hepatitis	155	19	2	0.46	0.50	5.33	839.62	836.14	34.61	38.07
Wine	178	4	3	0.84	0.59	3.03	1093.75	903.82	14.06	9.33
Glass	214	9	6	0.46	0.47	4.14	639.10	423.18	26.49	28.38
Heart Cleveland	303	13	5	1.32	1.34	12.62	1369.90	1105.43	193.29	198.57
Liver	345	6	2	0.76	0.76	19.45	1847.38	1386.35	65.4	78.90

4.4 Resultados Experimentales con *PSC*

En esta sección se presentan los resultados obtenidos con *PSC* al aplicarlo a distintos conjuntos de datos. Se reporta una comparación experimental contra los métodos *DROP3, DROP5, CLU* (método basado en agrupamientos) y *GCNN*.

Para *PSC* y *CLU* es necesario generar c grupos. Con el objetivo de determinar el valor de c se hicieron experimentos con diferentes valores para este parámetro. En la tabla 4.21 se muestran los resultados de clasificación (utilizando *k-NN, k=3*) obtenidos utilizando los valores *c=4t, 6t, 8t, 10t* y *12t*,

donde t es el numero de clases en cada conjunto de datos. En este experimento se ha omitido el conjunto de datos *Zoo* debido a que para el caso de *c=12t* este número de grupos excede la cantidad de prototipos en el conjunto de entrenamiento.

Tabla 4.21. Calidad de clasificación obtenida con *PSC* y *CLU* creando diferente número de grupos

Datos		Número de grupos									
		c=4t		*c=6t*		*c=8t*		*c=10t*		*c=12t*	
	Orig.	CLU	PSC	CLU	PSC	CLU	PSC	CLU	PSC	CLU	PSC
Bridges	66.09	51.63	51.63	53.54	56.54	58.38	59.45	61.27	61.09	61.07	60.03
Echocardiogram	95.71	85.90	86.42	90.71	86.42	94.10	91.42	90.71	85.53	88.10	85.67
Glass	71.42	53.26	56.21	52.83	59.09	55.58	56.16	56.08	63.52	57.63	56.72
Heart	82.49	74.61	71.26	73.00	73.27	75.29	72.26	76.33	74.00	75.18	73.54
Hepatitis	79.29	77.50	73.12	75.00	75.37	75.87	79.29	75.66	75.54	76.45	76.95
Iris	94.66	84.00	84.66	86.66	94.66	87.33	88.88	89.00	90.00	85.43	94.00
Liver	65.22	51.57	55.32	52.18	55.36	52.58	54.54	51.87	54.78	53.97	55.36
Wine	94.44	85.91	88.30	88.11	92.67	87.37	88.41	88.30	88.19	89.52	91.00
Promedio	**81.17**	**70.55**	**70.87**	**71.50**	**74.17**	**73.31**	**73.80**	**73.65**	**74.08**	**73.42**	**74.16**

De acuerdo a los resultados mostrados en la tabla 4.21, para el método *CLU,* en promedio, los mejores resultados de clasificación fueron obtenidos creando *c=10t*, mientras que para *PSC,* los mejores resultados se obtuvieron con *c=6t*, por lo que, estos valores de c se utilizaron en los experimentos reportados en este documento. Puede notarse que en estos experimentos, para ambos métodos no ocurre que los mejores valores de clasificación se obtienen creando un mayor número de grupos.

Tabla 4.22. Resultados de clasificación obtenidos con: Conjunto original (*Orig.*), *DROP3, DROP5, GCNN, CLU* y *PSC* utilizando *k-NN*, *k*=3.

Datos	Orig.	DROP3	DROP5	GCNN	CLU	PSC
	Acc	Acc	Acc	Acc	Acc	Acc
Bridges	66.09	56.36	62.82*	68.20*	61.27	**56.54**
Echocardiogram	95.71	92.86	98.42	93.39	90.71	**86.42**
Glass	71.42	66.28*	62.16*	67.74*	56.08*	**59.09**
Heart Cleveland	82.49	78.89	79.87	67.63	76.33	**73.27**
Hepatitis	79.29	78.13	75.42	60.66*	75.66	**75.37**
Iris	94.66	95.33*	94.00*	95.00*	89.00*	**94.66**
Liver	65.22	67.82*	63.46	66.09*	51.87*	**55.36**
Wine	94.44	94.41*	93.86*	94.44*	88.30*	**92.67**
Zoo	93.33	90.00	95.56	95.55	90.00	**92.22**
Promedio	**82.52**	**80.01**	**80.62**	**78.74**	**75.47**	**76.18**

Tabla 4.23. Resultados de retención correspondientes a la tabla 4.22.

Datos	Orig.	DROP3	DROP5	GCNN	CLU	PSC
	Str	Str	Str	Str	Str	Str
Bridges	100	14.78	20.66	88.20	63.68	42.86
Echocardiogram	100	13.95	14.87	22.67	30.03	19.82
Glass	100	24.35	25.91	61.62	31.15	38.45
Heart Cleveland	100	11.44	14.59	9.09	18.33	22.51
Hepatitis	100	11.47	15.05	17.75	14.33	7.95
Iris	100	15.33	12.44	38.00	22.22	20.45
Liver	100	26.83	30.59	83.70	6.44	45.87
Wine	100	15.04	10.55	78.89	37.03	37.15
Zoo	100	20.37	18.77	26.17	76.41	41.48
Promedio	**100**	**17.06**	**18.16**	**47.34**	**33.29**	**30.73**

Una vez que se fijó el valor del número de grupos a crear, los resultados obtenidos con los distintos métodos (utilizando *k-NN, k=3*) se muestran en las tablas 4.22-4.23 y figura 4.10. En las tablas mostradas en esta sección, se indica (*) si estadísticamente existe diferencia significativa con respecto a *PSC*.

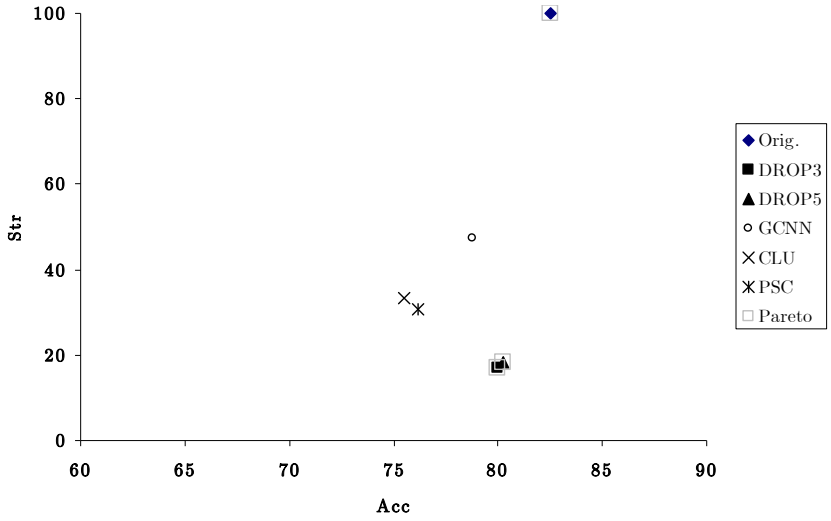


Figura 4.10. Gráfica de dispersión de los resultados de las tablas 4.22-4.23

En este experimento, en promedio, los mejores métodos fueron *DROP3, DROP5* y *GCNN* respectivamente. En lo que respecta al método *PSC*, éste obtuvo mejores resultados de clasificación que *CLU*, el cual es el método más lejano al frente de Pareto (figura 4.10), formado por los métodos *DROP*. Los subconjuntos obtenidos con *PSC* son de mayor dimensionalidad con respecto a

los obtenidos con *CLU* debido a que *CLU* sólo retiene a los prototipos representantes de cada grupo, mientras que *PSC* además de los prototipos representantes, retiene los prototipos frontera de los grupos no homogéneos.

En el experimento reportado en las tablas 4.22-4.23, se utilizó *k-NN* (*k=3*), razón por la cual, *DROP3, DROP5* y *GCNN* obtuvieron mejores resultados de clasificación, debido a que su funcionamiento está basado en este clasificador. Al igual que en experimentos anteriores, los subconjuntos obtenidos por los métodos mostrados en la tabla 4.22 fueron evaluados como entrenamiento para los clasificadores *LWR, SVM, C4.5* y *NB*. Los resultados obtenidos se reportan en las tablas 4.24-4.31 y figuras 4.11-4.14.

Tabla 4.24. Resultados de clasificación obtenidos al evaluar los subconjuntos obtenidos con: *DROP3, DROP5, GCNN, CLU* y *PSC* como entrenamiento para *LWR*

Datos	Orig.	DROP3	DROP5	GCNN	CLU	PSC
	Acc	Acc	Acc	Acc	Acc	Acc
Glass	57.85	50.09*	52.38	56.88	52.40*	**56.06**
Iris	98.00	92.00*	92.00*	95.33	94.00*	**96.00**
Liver	70.13	68.26	69.54	70.13	64.70*	**68.57**
Wine	92.15	62.75*	60.78*	92.15*	68.88*	**88.72**
Promedio	**79.53**	**68.26**	**68.68**	**78.62**	**70.00**	**77.34**

Tabla 4.25. Resultados de retención correspondientes a la tabla 4.24

Datos	Orig.	DROP3	DROP5	GCNN	CLU	PSC
	Str.	Str	Str	Str	Str	Str
Glass	100	20.66	14.87	6.50	31.15	38.45
Iris	100	8.00	26.69	22.67	22.22	20.45
Liver	100	6.26	6.00	9.09	6.44	45.87
Wine	100	6.42	10.91	38.00	37.03	37.15
Promedio	**100**	**20.39**	**19.87**	**65.55**	**24.21**	**35.48**

Con base en los resultados obtenidos con *LWR* y *SVM* (tablas 4.24-4.25, 4.26-4.27 y figuras 4.11-4.12), *GCNN* fue mejor que *PSC* en lo que respecta a clasificación, pero este ultimo superó a los demás métodos. En ambos casos, *PSC* y *GCNN* forman el frente de Pareto.

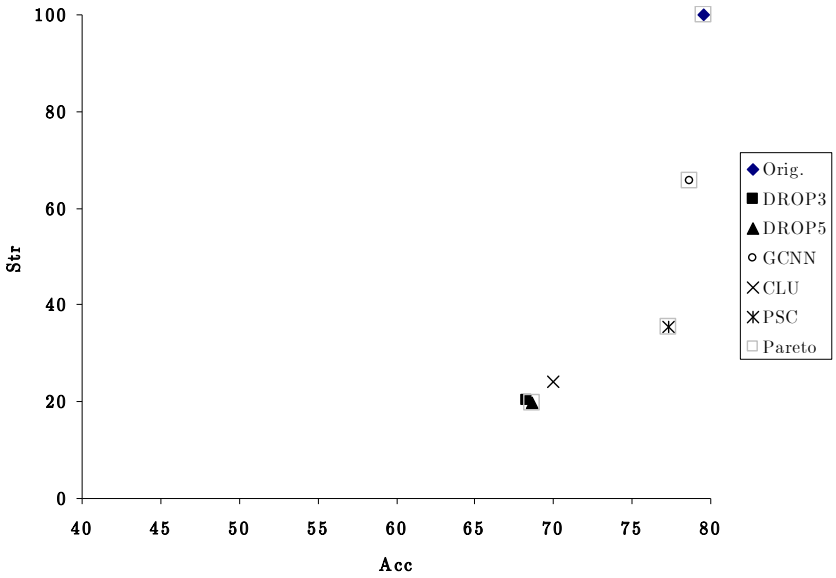


Figura 4.11. Gráfica de dispersión de los resultados de las tablas 4.24-4.25

Tabla 4.26. Resultados de clasificación obtenidos al evaluar los subconjuntos obtenidos con: *DROP3, DROP5, GCNN, CLU* y *PSC* como entrenamiento para *SVM*

Datos	Orig.	DROP3	DROP5	GCNN	CLU	PSC
	Acc	Acc	Acc	Acc	Acc	Acc
Glass	72.29	59.74	63.50	66.82*	61.90*	**65.84**
Iris	96.66	91.33	93.33	94.66	94.00	**93.33**
Liver	70.72	58.26	56.78	69.73	54.25*	**64.07**
Wine	97.18	94.93	92.71*	94.83*	88.88*	**95.52**
Promedio	**84.21**	**76.07**	**76.58**	**81.51**	**74.76**	**79.69**

Tabla 4.27. Resultados de retención correspondientes a la tabla 4.26

Datos	Orig.	DROP3	DROP5	GCNN	CLU	PSC
	Str.	Str	Str	Str	Str	Str
Glass	100	20.66	14.87	6.50	31.15	38.45
Iris	100	8.00	26.69	22.67	22.22	20.45
Liver	100	6.26	6.00	9.09	6.44	45.87
Wine	100	6.42	10.91	38.00	37.03	37.15
Promedio	**100**	**20.39**	**19.87**	**65.55**	**24.21**	**35.48**

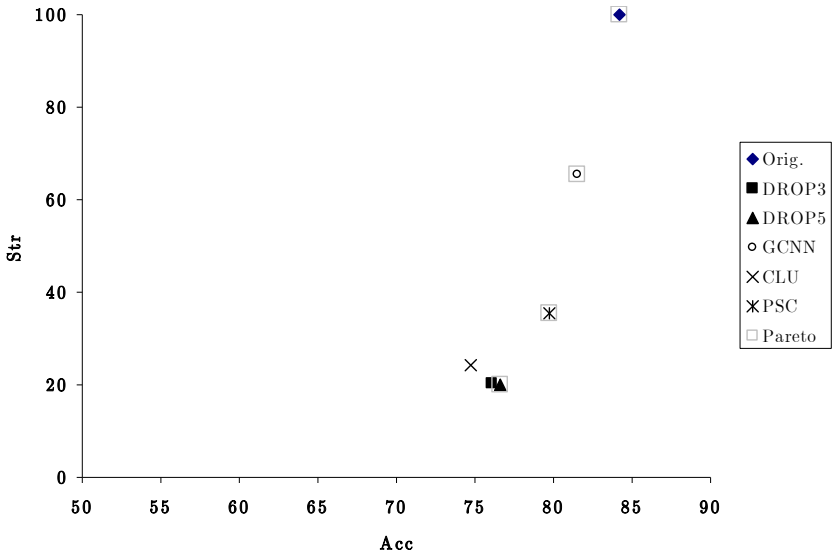


Figura 4.12. Gráfica de dispersión de los resultados de las tablas 4.26-4.27

Tabla 4.28. Resultados de clasificación obtenidos al evaluar los subconjuntos obtenidos con: *DROP3, DROP5, GCNN, CLU* y *PSC* como entrenamiento para *C4.5*

Datos	Orig.	DROP3	DROP5	GCNN	CLU	PSC
	Acc	Acc	Acc	Acc	Acc	Acc
Bridges	65.81	47.90*	39.54*	52.36*	55.45*	**65.81**
Echocardiogram	95.71	84.10	92.85	91.78	93.21	**94.46**
Glass	67.29	60.19	53.76	60.75	58.35	**60.58**
Heart Cleveland	71.96	68.59	72.16*	66.00	76.57	**69.89**
Hepatitis	76.70	63.33*	63.41*	65.16	84.61	**73.50**
Iris	93.99	92.66*	90.66	90.66	71.58*	**90.66**
Liver	63.67	59.48	63.67	61.76*	44.13*	**63.67**
Wine	94.44	84.43	78.88	95.55	49.49*	**90.77**
Zoo	93.33	81.10*	88.88	81.10*	86.65*	**93.33**
Promedio	**80.32**	**71.31**	**71.53**	**73.90**	**68.89**	**78.07**

Tabla 4.29. Resultados de retención correspondientes a la tabla 4.28

Datos	Orig.	DROP3	DROP5	GCNN	CLU	PSC
	Str.	Str	Str	Str	Str	Str
Bridges	100	14.78	20.66	88.20	63.68	42.86
Echocardiogram	100	13.95	14.87	22.67	30.03	19.82
Glass	100	24.35	25.91	61.62	31.15	38.45
Heart Cleveland	100	11.44	14.59	9.09	18.33	22.51
Hepatitis	100	11.47	15.05	17.75	14.33	7.95
Iris	100	15.33	12.44	38.00	22.22	20.45
Liver	100	26.83	30.59	83.70	6.44	45.87
Wine	100	15.04	10.55	78.89	37.03	37.15
Zoo	100	20.37	18.77	26.17	76.41	41.48
Promedio	**100**	**17.06**	**18.16**	**47.34**	**33.29**	**30.73**

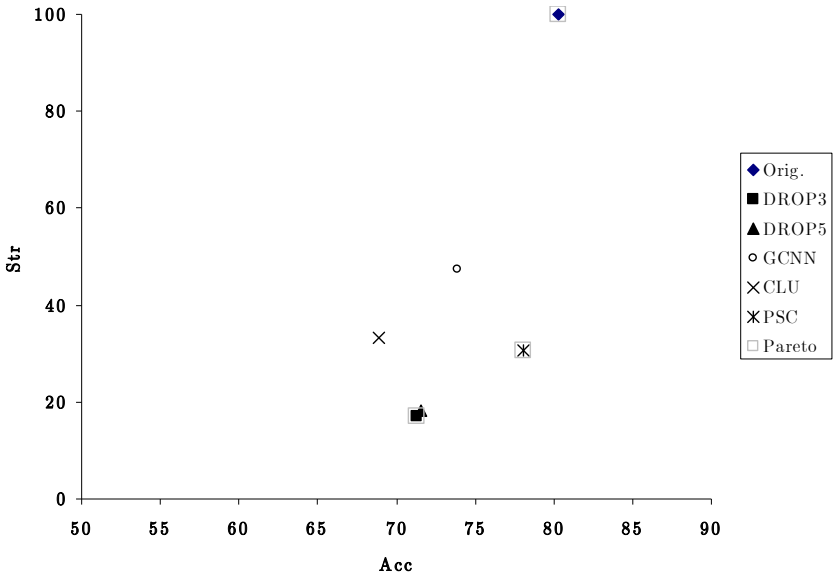


Figura 4.13. Gráfica de dispersión de los resultados de la tablas 4.28-4.29

Tabla 4.30. Resultados de clasificación obtenidos al evaluar los subconjuntos obtenidos con: *DROP3, DROP5, GCNN, CLU* y *PSC* como entrenamiento para *NB*

Datos	Orig.	DROP3	DROP5	GCNN	CLU	PSC
Bridges	64.00	49.61	39.81*	44.36	56.54	**58.63**
Echocardiogram	97.14	78.31*	77.32	91.78	78.03	**84.82**
Glass	48.05	49.56	47.74	47.57	41.78*	**47.14**
Heart Cleveland	83.81	78.20	81.16	75.52	78.17	**78.20**
Hepatitis	84.58	61.83	54.20	65.87*	66.79	**79.29**
Iris	95.33	91.99	93.99	95.33	91.99*	**93.33**
Liver	56.02	61.50	61.77	56.88	50.44*	**56.88**
Wine	98.81	61.11*	66.66	96.66	90.00*	**97.77**
Zoo	95.55	88.88*	83.33	93.33	93.33	**95.55**
Promedio	**80.37**	**69.00**	**67.33**	**74.14**	**71.90**	**76.85**

Tabla 4.31. Resultados de retención correspondientes a la tabla 4.30

Datos	Orig.	DROP3	DROP5	GCNN	CLU	PSC
	Str.	Str	Str	Str	Str	Str
Bridges	100	14.78	20.66	88.20	63.68	42.86
Echocardiogram	100	13.95	14.87	22.67	30.03	19.82
Glass	100	24.35	25.91	61.62	31.15	38.45
Heart Cleveland	100	11.44	14.59	9.09	18.33	22.51
Hepatitis	100	11.47	15.05	17.75	14.33	7.95
Iris	100	15.33	12.44	38.00	22.22	20.45
Liver	100	26.83	30.59	83.70	6.44	45.87
Wine	100	15.04	10.55	78.89	37.03	37.15
Zoo	100	20.37	18.77	26.17	76.41	41.48
Promedio	**100**	**17.06**	**18.16**	**47.34**	**33.29**	**30.73**

Por otra parte, en los resultados obtenidos con *C4.5* y *NB* (tablas 4.28-4.29, 4.30-4.31 y figuras 4.13-4.14), en promedio, el mejor método fue *PSC*, que también obtuvo los mejores resultados de clasificación en la mayoría de los conjuntos de datos. En ambos casos, *PSC* modifica el frente de Pareto en lo que respecta a clasificación.

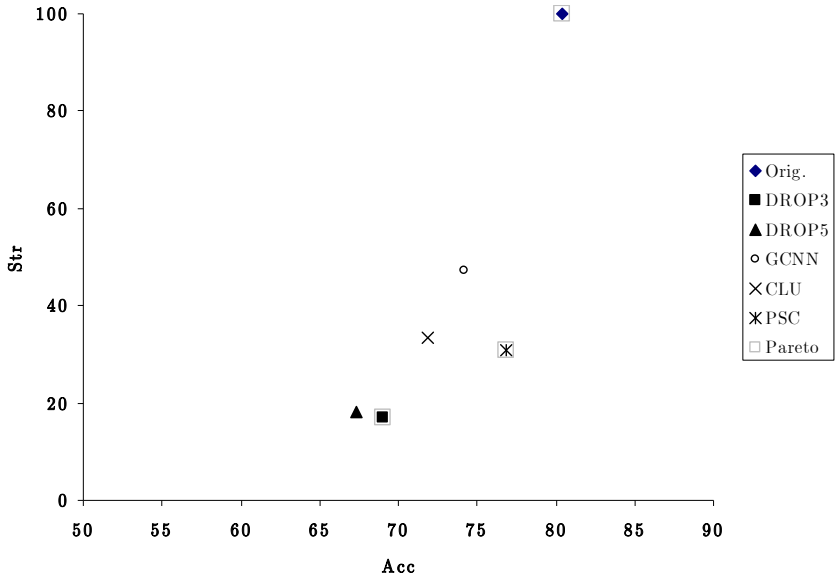


Figura 4.14. Gráfica de dispersión de los resultados de las tablas 4.30-4.31

Puede notarse que, para los distintos clasificadores utilizados, los métodos *PSC* y *GCNN* presentan un desempeño mejor comparado con los métodos *DROP*.

PSC es un método que selecciona prototipos frontera, por lo que se realizaron experimentos para compararlo contra el método *POC-NN,* que también fue diseñado para seleccionar prototipos frontera. Los resultados obtenidos por *PSC* y *POC-NN* para los distintos clasificadores utilizados en experimentos previos se muestran en las tablas 4.32-4.33. En estas tablas sólo se indican los resultados de retención para el clasificador *k-NN* debido a que éstos son los mismos que para el resto de los resultados con los otros clasificadores.

Cabe mencionar que este experimento solamente se aplicó sobre conjuntos de datos numéricos ya que *POC-NN* sólo puede ser aplicado a este tipo de datos.

Tabla 4.32. Resultados de clasificación obtenidos con *PSC* y *POC-NN* utilizando los subconjuntos seleccionados como entrenamiento para *k-NN* (k=3), *LWR* y *SVM*.

Datos	Clasificador										
	k-NN			LWR		SVM		C4.5		NB	
	Orig.	PSC	POC-NN	PSC	POC-NN	PSC	POC-NN	PSC	POC-NN	PSC	POC-NN
	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc
Glass	71.42	**59.09**	71.47*	**56.06**	50.41	**65.84**	57.46*	**60.58**	63.56*	**47.14**	46.66
Iris	94.66	**94.66**	77.33*	**96.00**	85.33	**93.33**	94.00	**90.66**	73.99	**93.33**	76.66
Liver	65.22	**55.36**	55.68	**68.57**	68.68	**64.07**	58.56	**63.67**	54.46	**56.88**	56.57
Wine	94.44	**92.67**	94.93*	**88.72**	86.40	**95.52**	95.52	**90.77**	90.32*	**97.77**	91.11
Promedio	**86.53**	**75.45**	**74.85**	**77.34**	**72.71**	**79.69**	**76.39**	**76.42**	**70.58**	**73.78**	**67.75**

Tabla 4.33. Resultados de retención correspondientes a las tablas 4.32-4.33.

Datos	POC-NN	PSC
	Str	Str
Glass	57.42	38.45
Iris	12.88	20.45
Liver	58.09	45.87
Wine	40.82	37.15
Promedio	**42.30**	**35.48**

A partir de estos resultados, puede observarse que de entre estos dos métodos, para todos los clasificadores, *PSC* obtuvo mejores resultados tanto en clasificación como en reducción.

4.5 Resultados Experimentales *PSR*

En esta sección se presentan los resultados experimentales obtenidos al aplicar los métodos *DROP3, DROP5, GCNN* y *PSR* a distintos conjuntos de datos.

Debido a que en la fase inicial de *PSR* es necesario elegir los q prototipos más relevantes, para determinar el valor de q se realizaron experimentos con diferentes valores de este parámetro. En la tablas 4.34-4.35 se muestran los resultados de clasificación obtenidos al elegir q=10%, 20%, 30%, 40% y 50% de los prototipos más relevantes en cada clase.

Tabla 4.34. Resultados de clasificación obtenidos con *PSR* eligiendo diferente número de prototipos relevantes por clase

Datos	Porcentaje (q) de prototipos por clase				
	q=10%	*q=20%*	*q=30%*	*q=40%*	*q=50%*
	Acc	*Acc*	*Acc*	*Acc*	*Acc*
Bridges	48.90	50.90	57.63	57.54	61.18
Echocardiogram	83.21	83.92	90.53	90.53	89.28
Glass	61.12	64.37	64.85	65.80	69.17
Heart Cleveland	75.81	75.84	79.18	78.20	77.21
Hepatitis	74.95	82.58	83.16	81.91	79.91
Iris	86.66	88.66	91.33	89.33	92.00
Liver	61.96	62.85	63.77	64.92	64.61
Wine	94.00	92.12	92.18	92.74	93.00
Zoo	93.33	93.33	93.33	93.33	93.33
Promedio	**75.55**	**77.17**	**79.55**	**79.37**	**79.97**

Tabla 4.35. Resultados de retención correspondientes a la tabla 4.34

Datos	Porcentaje (q) de prototipos por clase				
	q=10%	*q=20%*	*q=30%*	*q=40%*	*q=50%*
	Str	*Str*	*Str*	*Str*	*Str*
Bridges	26.31	37.62	47.79	55.65	63.83
Echocardiogram	14.26	26.57	37.68	48.05	56.75
Glass	23.67	32.81	42.36	51.24	59.60
Heart Cleveland	14.22	25.70	36.96	49.39	61.05
Hepatitis	13.11	23.58	33.40	43.22	52.47
Iris	14.88	25.70	38.07	47.11	56.59
Liver	14.36	24.66	35.55	44.15	53.84
Wine	18.10	30.96	42.94	52.24	61.98
Zoo	29.50	39.75	51.11	59.13	69.01
Promedio	**18.71**	**29.71**	**40.65**	**50.02**	**54.96**

Al analizar los resultados obtenidos, puede notarse que la clasificación obtenida con *PSR* será mejor cuando la cantidad de prototipos representativos seleccionados en la fase inicial se acerca al 100%, lo cual, para fines de selección de prototipos no sería un resultado deseable.

Con base en los resultados obtenidos, en el caso promedio, el valor de q con el que *PSR* obtuvo mejores resultados fue q=30%, por lo que se utilizó este valor en los resultados mostrados en secciones posteriores.

Después de establecer el valor del parámetro q para *PSR*, los resultados de clasificación obtenidos con éste y los métodos mencionados al principio de esta sección (utilizando *k-NN, k=3*) se muestran en las tablas 4.36-4.37 y figura 4.15.

Tabla 4.36. Resultados de clasificación obtenidos con: Conjunto original (*Orig.*), *DROP3, DROP5, GCNN* y *PSC* utilizando *k-NN*, *k*=3.

Datos	Orig.	DROP3	DROP5	GCNN	PSR
	Acc	Acc	Acc	Acc	Acc
Bridges	66.09	56.36	62.82*	68.20	**57.63**
Echocardiogram	95.71	92.86	94.82	93.39	**90.53**
Glass	71.42	66.28	62.16	67.74	**64.85**
Heart Cleveland	82.49	78.89	79.87	67.63*	**79.18**
Hepatitis	79.29	78.13	75.42*	60.66*	**83.16**
Iris	94.66	95.33	94.00	95.00	**91.33**
Liver	65.22	67.82	63.46	66.09	**63.77**
Wine	94.44	94.41	93.86	94.44	**92.18**
Zoo	93.33	90.00	95.56	95.55	**93.33**
Promedio	**82.52**	**80.01**	**80.62**	**78.74**	**79.55**

Tabla 4.37. Resultados de retención correspondientes a la tabla 4.36.

Datos	Orig.	DROP3	DROP5	GCNN	PSR
	Str	Str	Str	Str	Str
Bridges	100	14.78	20.66	88.20	47.79
Echocardiogram	100	13.95	14.87	22.67	37.68
Glass	100	24.35	25.91	61.62	42.36
Heart Cleveland	100	11.44	14.59	9.09	36.96
Hepatitis	100	11.47	15.05	17.75	33.40
Iris	100	15.33	12.44	38.00	38.07
Liver	100	26.83	30.59	83.70	35.55
Wine	100	15.04	10.55	78.89	42.94
Zoo	100	20.37	18.77	26.17	51.11
Promedio	**100**	**17.06**	**18.16**	**47.34**	**40.65**

Con base en estos resultados, puede observarse que, en promedio, los mejores resultados se obtuvieron con *DROP3, DROP5* y *PSR,* para los cuales, sólo en dos casos existe diferencia significativa con respecto a la clasificación.

Los subconjuntos obtenidos por los métodos del experimento anterior se evaluaron como entrenamiento para *LWR, SVM, C4.5* y *NB*. Los resultados de clasificación se muestran en las tablas 4.38-4.41 y figuras 4.16-4.19. Para estos resultados, los porcentajes de retención son los mismos mostrados en la tabla 4.37.

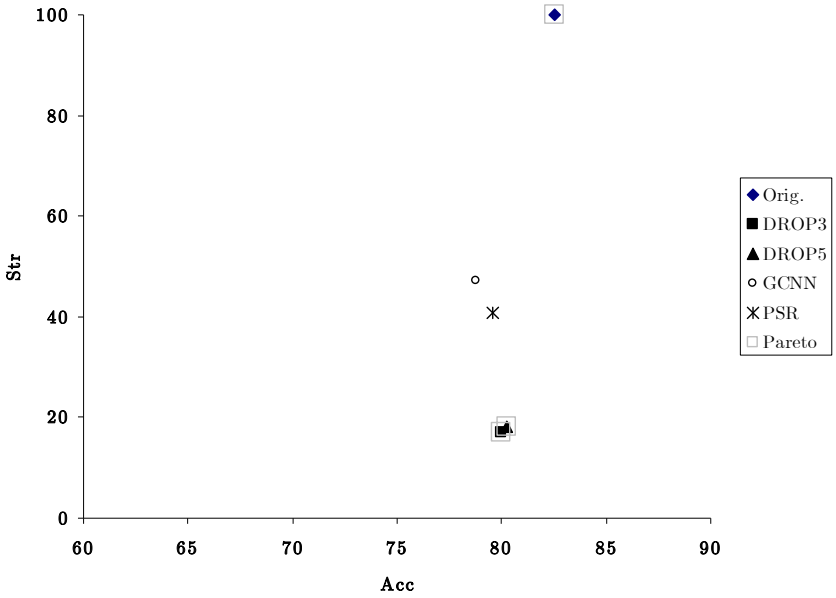


Figura 4.15. Gráfica de dispersión de los resultados de las tablas 4.36-4.37

Tabla 4.38. Resultados de clasificación obtenidos al evaluar los subconjuntos obtenidos con: *DROP3, DROP5, GCNN, CLU* y *PSR* como entrenamiento para *LWR*

Datos	Orig.	DROP3	DROP5	GCNN	PSR
	Acc	Acc	Acc	Acc	Acc
Glass	57.85	50.09	52.38	56.88	**56.54**
Iris	98.00	92.00*	92.00*	95.33	**95.33**
Liver	70.13	68.26	69.54	70.13	**70.58**
Wine	92.15	62.75	60.78*	92.15	**85.98**
Promedio	**79.53**	**68.28**	**68.68**	**78.62**	**65.55**

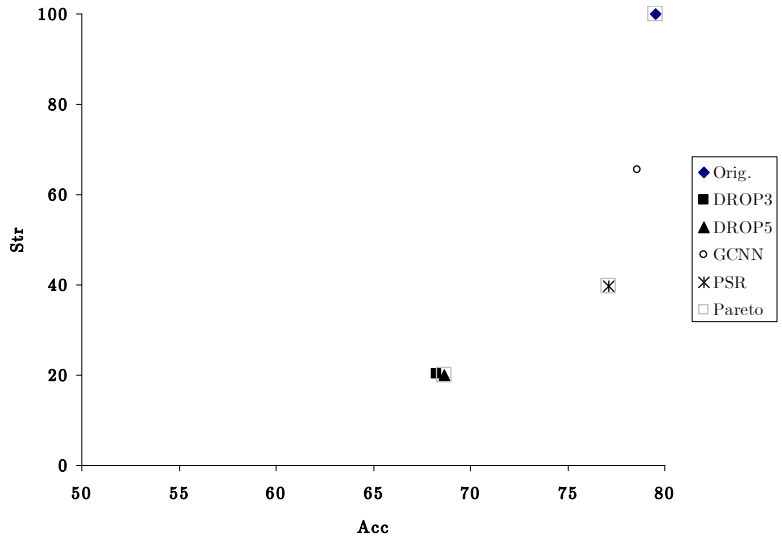


Figura 4.16. Gráfica de dispersión de los resultados de la tabla 4.38

Tabla 4.39. Resultados de clasificación obtenidos al evaluar los subconjuntos obtenidos con: *DROP3, DROP5, GCNN, CLU* y *PSC* como entrenamiento para *SVM*

Datos	Orig.	DROP3	DROP5	GCNN	PSR
	Acc	Acc	Acc	Acc	Acc
Glass	72.29	59.74*	63.50	66.82*	**61.66**
Iris	96.66	91.33*	93.33*	94.66	**95.33**
Liver	70.72	58.26*	56.78*	69.73	**69.02**
Wine	97.18	94.93	92.71*	94.83*	**94.96**
Promedio	**84.21**	**76.07**	**76.58**	**81.51**	**80.24**

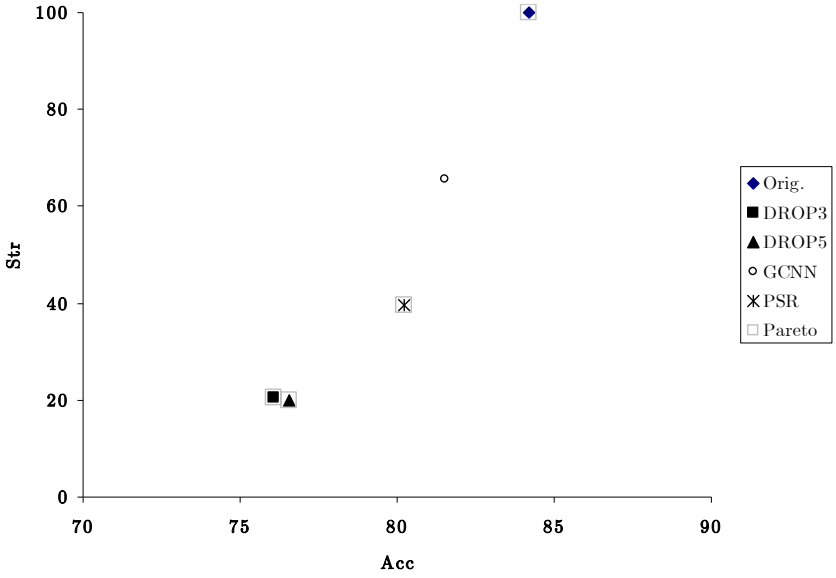


Figura 4.17. Gráfica de dispersión de los resultados de la tabla 4.39

Tabla 4.40. Resultados de clasificación obtenidos al evaluar los subconjuntos obtenidos con: *DROP3, DROP5, GCNN* y *PSR* como entrenamiento para *C4.5*

Datos	Orig.	DROP3	DROP5	GCNN	PSR
	Acc	Acc	Acc	Acc	Acc
Bridges	65.81	47.90	39.54*	52.36	**51.09**
Echocardiogram	95.71	84.10*	92.85	91.78	**95.71**
Glass	67.29	60.19	53.76	60.75	**63.48**
Heart Cleveland	71.96	68.59	72.16	66.00	**71.35**
Hepatitis	76.70	63.33*	63.41*	65.16*	**83.20**
Iris	93.99	92.66	90.66	90.66	**93.33**
Liver	63.67	59.48*	63.67*	61.76*	**65.21**
Wine	94.44	84.43*	78.88*	95.55*	**94.44**
Zoo	93.33	81.10*	88.88*	81.10*	**95.55**
Promedio	**80.32**	**71.31**	**71.53**	**73.90**	**79.26**

De acuerdo a los resultados obtenidos, para *LWR* y *SVM* los mejores métodos fueron *GCNN* y *PSR* respectivamente, y en la mayoría de los casos no existe diferencia significativa de clasificación entre ellos.

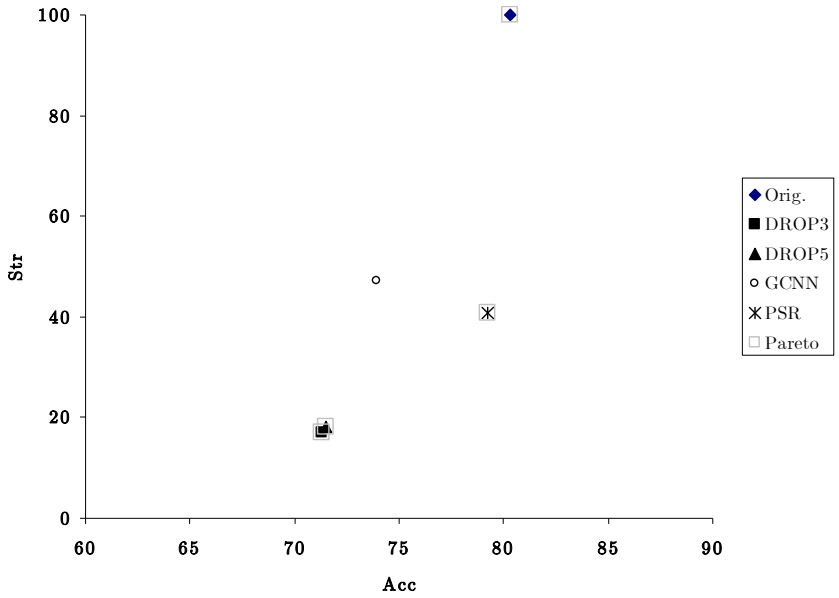


Figura 4.18. Gráfica de dispersión de los resultados de la tabla 4.40

Tabla 4.41. Resultados de clasificación obtenidos al evaluar los subconjuntos obtenidos con: *DROP3, DROP5, GCNN* y *PSR* como entrenamiento para *NB*

Datos	Orig.	DROP3	DROP5	GCNN	PSR
Bridges	64.00	49.61	39.81*	44.36*	**50.90**
Echocardiogram	97.14	78.31	77.32	91.78	**90.71**
Glass	48.05	49.56*	47.74*	47.57*	**60.71**
Heart Cleveland	83.81	78.20	81.16	75.52	**81.86**
Hepatitis	84.58	61.83*	54.20*	65.87*	**79.37**
Iris	95.33	91.99	93.99	95.33*	**91.99**
Liver	56.02	61.50*	61.77*	56.88	**66.94**
Wine	98.81	61.11*	66.66*	96.66*	**92.22**
Zoo	95.55	88.88*	83.33*	93.33	**95.55**
Promedio	**80.37**	**69.00**	**67.33**	**74.14**	**78.92**

Por otra parte, con *C4.5* y *NB,* el mejor método en clasificación fue *PSR* con diferencia significativa en varios casos con respecto a los demás métodos. Para los cuatro clasificadores, de entre los dos mejores métodos (*PSR* y *GCNN*) los mejores resultados de retención se obtuvieron con *PSR*.

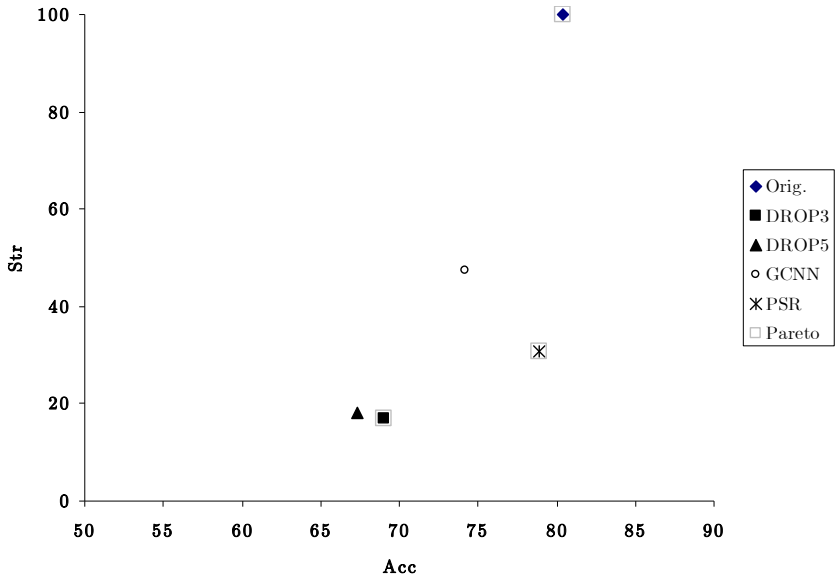


Figura 4.19. Gráfica de dispersión de los resultados de la tabla 4.41

4.5.1 Tiempos de ejecución de los métodos *PSC* y *PSR*

En esta sección se reportan los tiempos de ejecución de los métodos *PSC, PSR* y los métodos evaluados en las secciones 4.5-4.6.

En la tabla 4.42 se muestran los tiempos de ejecución (en segundos) así como el tamaño de cada conjunto de datos. En este experimento se incluyen tres conjuntos de datos adicionales (*Segmentation, Letter* y *UPS*) con mayor cantidad de prototipos con respecto a los conjuntos de datos utilizados en los experimentos de secciones anteriores. Los símbolos "♦", "♦♦" y "♦♦♦" indican que el tiempo de ejecución corresponde a más de 10, 20 y 30 horas respectivamente.

Puede notarse que para los conjuntos de datos de menor tamaño, los métodos *DROP* son los más rápidos, pero, para los conjuntos más grandes, *CLU, PSC* y *PSR* son mucho más rápidos que los demás métodos, lo cual es un aspecto importante porque la selección de prototipos es particularmente útil para este tipo de conjuntos de datos, en los que se requiere reducir la cantidad

de los mismos y, como consecuencia de ello, tener una reducción de los tiempos de entrenamiento y/o clasificación.

Tabla 4.42. Tiempos de ejecución (en segundos) de los métodos *DROP3, DROP5,GCNN, CLU, PSC* y *PSR*

Datos	Tamaño			Tiempo					
	Prototipos	Atributos	Clases	DROP3	DROP5	GCNN	CLU	PSC	PSR
Echocardiogram	74	9	2	0.19	0.08	1.02	1.13	1.34	0.42
Zoo	90	16	7	0.26	0.27	1.15	2.31	4.24	0.75
Bridges	108	11	7	0.16	0.14	15.27	7.94	8.33	1.15
Iris	150	4	3	0.19	0.16	1.25	0.74	0.79	0.37
Hepatitis	155	19	2	0.46	0.50	5.33	10.03	13.35	2.42
Wine	178	4	3	0.84	0.59	3.03	1.62	1.82	0.46
Glass	214	9	6	0.46	0.47	4.14	0.43	0.54	0.87
Heart Cleveland	303	13	5	1.32	1.34	12.62	19.09	19.88	6.57
Liver	345	6	2	0.76	0.76	19.45	1.62	1.70	0.73
Segmentation	2100	19	7	208.39	208.91	716.28	15.84	16.52	59.49
Letter	20000	16	26	15213.13	14339.61	♦	769.08	613.58	912.28
UPS	9000	255	10	♦	♦♦	♦♦♦	743.47	793.94	3125.42

Otro experimento referente a los tiempos de ejecución se aplicó al conjunto de datos *Shuttle Statlog* cuyo número de prototipos (58000 prototipos, 9 atributos, 3 clases) es mayor al de los conjuntos utilizados previamente. A partir de los prototipos de esta base de datos se crearon conjuntos con distinto número de prototipos. A estos conjuntos de datos se aplicaron los métodos *DROP, GCNN, CLU, PSC, PSR* y se incluye el método *POC-NN,* el cual, puede ser aplicado a este conjunto de datos cuyos prototipos son descritos por atributos numéricos en su totalidad.

Los tiempos de ejecución de cada método se reportan en la tabla 4.43[4] y la figura 4.20a, en la cual, algunos puntos no fueron graficados debido a que exceden la escala del eje vertical. En la figura 4.20b se muestran solamente los tiempos de ejecución de los métodos más rápidos: *CLU, PSC* y *PSR,* con el objetivo de apreciar mejor la diferencia entre ellos. Los resultados de clasificación obtenidos en este experimento se muestran en la figura 4.21.

Los métodos *CLU, PSC* y *PSR* son los métodos más rápidos respectivamente (figuras 4.20a-4.20b) pero de entre estos tres, *PSC* y *PSR* son mejores en clasificación que *CLU.* Por tanto, ambos métodos de selección son la mejor

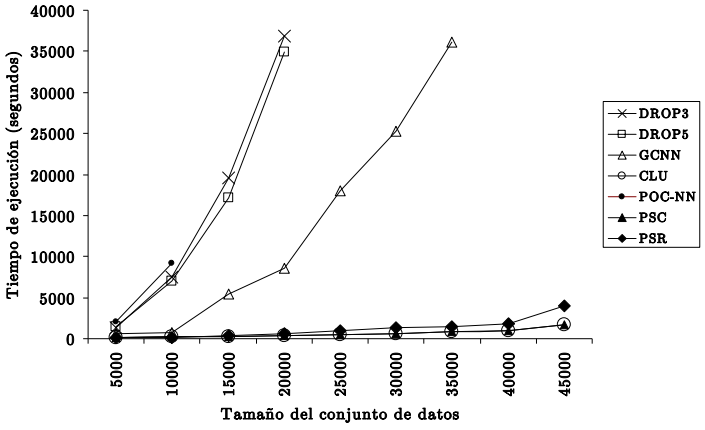
[4] Con la implementación del método *POC-NN* utilizada no fue posible aplicar este método ("n.a." en la tabla 4.43) a conjuntos de más de 10000 prototipos.

opción para problemas con muchos prototipos de entrenamiento. Sin embargo, de las figura 4.20b y 4.21 se aprecia que para los conjuntos más grandes *PSC* es más rápido que *PSR,* pero *PSR* obtiene mejor calidad.

Tabla 4.43. Tiempos de ejecución (en segundos) de los métodos *DROP3, DROP5, GCNN, POC-NN, PSC* y *PSR* para los distintos conjuntos de datos creados a parir de *Shuttle Statlog*

Datos (Número de prototipos)	Tiempo						
	DROP3	DROP5	GCNN	CLU	POC-NN	PSC	PSR
5000	1385.16	1391.94	569.21	68.25	2079.12	70.68	94.65
10000	7435.28	6956.44	680.04	276.35	9124.50	286.64	169.96
15000	19578.34	17173.80	5398.64	294.81	n.a.	295.38	360.32
20000	36800.69	34882.73	8640.25	362.85	n.a.	378.51	619.25
25000	63949.24	58913.58	18005.23	483.25	n.a.	487.39	940.65
30000	94360.05	91646.41	25200.31	632.80	n.a.	633.40	1333.46
35000	133964.02	130303.20	36120.08	826.05	n.a.	838.73	1509.09
40000	198886.48	184613.48	43120.61	948.69	n.a.	957.03	1789.92
45000	272846.47	259687.56	72180.34	1650.28	n.a.	1660.19	3955.81

a)



b)

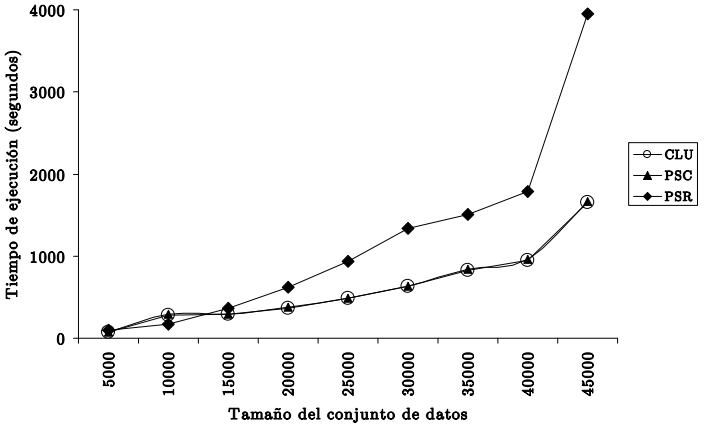


Figura 4.20. a) Gráfica de los tiempos de ejecución mostrados en la tabla 4.44. b) Gráfica de los tiempos de ejecución de los métodos *CLU, PSC* y *PSR*

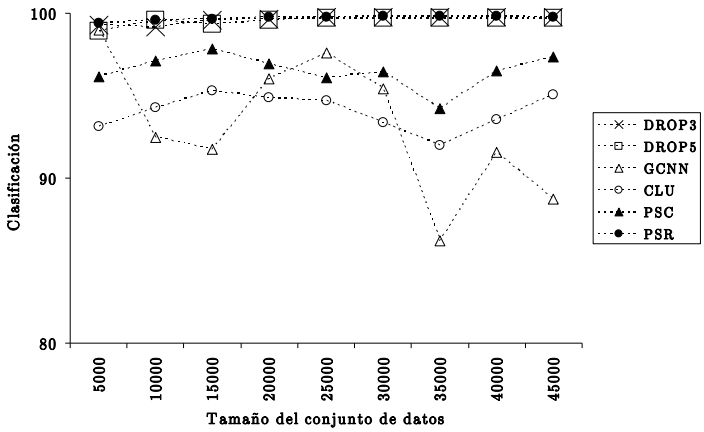


Figura 4.21. Resultados de clasificación obtenidos con los conjuntos de datos creados a partir de *Shuttle Statlog*

En los experimentos anteriores se reportan solamente los tiempos de selección. Otro experimento realizado consistió en medir los tiempos totales de cada método, es decir el tiempo de selección y el de entrenamiento/clasificación utilizando el conjunto obtenido por cada método como entrenamiento de los clasificadores. Este experimento se llevó a cabo para determinar cómo son los tiempos totales con respecto a clasificar con todo el conjunto de entrenamiento sin hacer selección de prototipos, para lo cual se consideraron los cuatro conjuntos de datos más grandes utilizados en experimentos previos (*Segmentation, Letter, UPS, Shuttle*). Los resultados se muestran en las tablas 4.44 a 4.48. En cada una de ellas se muestran los tiempos totales para cada clasificador: *k-NN, LWR, SVM, C4.5* y *NB* respectivamente. Se reporta el tiempo de entrenamiento y clasificación con el conjunto de de datos sin previa selección para cada clasificador (*Torig.*) y los tiempos totales de cada método. Para el caso de *k-NN,* este tiempo corresponde a clasificación ya que en este clasificador no hay una fase de entrenamiento. En esta tabla, el tiempo indicado a la izquierda del símbolo "+" corresponde al tiempo de selección, mientras que el tiempo a la derecha de tal símbolo es el tiempo de entrenamiento/clasificación con el subconjunto seleccionado. Los valores en los que no se específica la unidad de tiempo corresponden a segundos.

De acuerdo a estos resultados, puede notarse que para los clasificadores *k-NN, C4.5* y *NB,* con todos los métodos, en ningún caso el tiempo total es menor con respecto al tiempo usando todo el conjunto de entrenamiento. Este aspecto es aún más notorio para *C4.5* y *NB*, ya que son clasificadores muy rápidos. Por otra parte, para *LWR* y *SVM* los tiempos totales de los métodos *CLU, PSC* y *PSR* son menores (y notoriamente menores para los conjuntos de datos *UPS, Shuttle*) con respecto a entrenar con todo el conjunto de entrenamiento.

Tabla 4.44. Tiempos totales de ejecución de los métodos *DROP3, DROP5, GCNN, CLU, PSC* y *PSR* utilizando *k-NN*.

Datos	Tiempo						
	Torig.	DROP3	DROP5	GCNN	CLU	PSC	PSR
Segmentation	1.34	208.39+0.21	208.91+0.18	716.28+0.17	15.84+0.04	16.52+0.21	59.49+0.52
Letter	99.71	15213.13+16.29	14339.61+13.5	10hrs+33.96	769.08+3.65	613.58+17.23	912.28+45.21
UPS	219.71	10hrs+22.55	20hrs+17.49	30hrs+75.8	743.47+4.6	793.94+33.9	3125.42+100.03
Shuttle	403.6	75hrs+2.83	72hrs+2.36	20hrs+11.35	1650.28+0.24	1660.19+6.34	3955.81+162.04

Tabla 4.45. Tiempos totales de ejecución de los métodos *DROP3, DROP5, GCNN, CLU, PSC* y *PSR* utilizando *LWR*.

Datos	Tiempo						
	Torig.	DROP3	DROP5	GCNN	CLU	PSC	PSR
Segmentation	21.34	208.39+3.47	208.91+3.07	716.28+2.96	**15.84+0.78**	**16.52+3.23**	59.49+8.15
Letter	836.87	15213.13+136.66	14339.61+114.08	10hrs+285.31	**769.08+12.03**	**613.58+153.37**	912.28+410.25
UPS	2647.23	10hrs+568.92	20hrs+210.84	30hrs+914.12	**743.47+31.42**	**793.94+409.51**	3125.42+1211.35
Shuttle	3243.42	75hrs+22.07	72hrs+19.21	20hrs+91.42	**1650.28+1.94**	**1660.19+50.84**	3955.81+1300.11

Tabla 4.46. Tiempos totales de ejecución de los métodos *DROP3, DROP5, GCNN, CLU, PSC* y *PSR* utilizando *SVM*.

Datos	Tiempo						
	Torig.	DROP3	DROP5	GCNN	CLU	PSC	PSR
Segmentation	112.76	208.39+17.97	208.91+16.15	716.28+15.56	**15.84+4.15**	**16.52+17.07**	**59.49+43.23**
Letter	2548.41	15213.13+417.12	14339.61+348.31	10hrs+868.23	**769.08+36.64**	**613.58+467.12**	**912.28+1053.56**
UPS	13876.85	10hrs+398.18	20hrs+308.62	30hrs+1338.28	**743.47+46.12**	**793.94+599.74**	**3125.42+1796.14**
Shuttle	14128.37	75hrs+30.31	72hrs+24.37	20hrs+91.43	**1650.28+2.47**	**1660.19+64.86**	**3955.81+1657.54**

Tabla 4.47. Tiempos totales de ejecución de los métodos *DROP3, DROP5, GCNN, CLU, PSC* y *PSR* utilizando *C4.5*.

Datos	Tiempo						
	Torig.	DROP3	DROP5	GCNN	CLU	PSC	PSR
Segmentation	1.40	208.39+0.24	208.91+0.2	716.28+0.19	15.84+0.005	16.52+0.21	59.49+0.53
Letter	26.35	15213.13+4.31	14339.61+4.01	10hrs+8.99	769.08+0.37	613.58+4.82	912.28+12.96
UPS	137.56	10hrs+14.15	20hrs+10.92	30hrs+47.48	743.47+1.63	793.94+21.28	3125.42+63.88
Shuttle	29.18	75hrs+0.23	72hrs+0.17	20hrs+0.82	1650.28+0.017	1660.19+0.45	3955.81+11.71

Tabla 4.48. Tiempos totales de ejecución de los métodos *DROP3, DROP5, GCNN, CLU, PSC* y *PSR* utilizando *NB*.

Datos	Tiempo						
	Torig.	DROP3	DROP5	GCNN	CLU	PSC	PSR
Segmentation	0.43	208.39+0.06	208.91+0.05	716.28+0.05	15.84+0.01	16.52+0.06	59.49+0.16
Letter	3.39	15213.13+0.55	14339.61+0.46	10hrs+0.14	769.08+0.048	613.58+0.62	912.28+1.66
UPS	18.18	10hrs+1.89	20hrs+1.44	30hrs+6.27	743.47+0.21	793.94+2.81	3125.42+8.44
Shuttle	9.37	75hrs+0.06	72hrs+0.05	20hrs+0.26	1650.28+0.005	1660.19+0.14	3955.81+3.76

La utilidad de la selección de prototipos en estos experimentos fue notoria para los clasificadores *LWR* y *SVM* debido que, con respecto a los tiempos totales, se requiere de más tiempo al entrenar con todo el conjunto de entrenamiento.

En las tablas 4.44 a 4.48 los tiempos mostrados se obtuvieron al clasificar conjuntos cuyo tamaño aproximado es $|T|/10$ debido a que se clasifica el bloque de prueba de la validación cruzada. Supongamos el caso particular en que a partir de los conjuntos de entrenamiento de las tablas 4.44 a 4.48 se requiere clasificar con *k-NN* conjuntos más grandes; consideremos este clasificador, pues es de los más costosos en su fase de clasificación. Para este caso supongamos que se requiere clasificar conjuntos cuyo tamaño es 10 veces el tamaño de cada uno de los conjuntos de entrenamiento de las tablas 4.44 a 4.48. Los resultados de los respectivos tiempos (en segundos) de clasificación para *k-NN* y los tiempos totales de los métodos *PSC, PSR* se muestran en la tabla 4.49.

Tabla 4.49. Tiempos totales de ejecución de los métodos *PSC* y *PSR* utilizando *k-NN* al clasificar conjuntos cuyo tamaño es 10 veces con respecto a los conjuntos de las tablas 4.44 a 4.48

Datos a clasificar	Tiempo		
	Torig.	PSC	PSR
10|Segmentation|	136.12	**16.52+51.06**	**59.49+51.82**
10|Letter|	9968.45	**613.58+1786.32**	**912.28+4658.14**
10|UPS|	21885.14	**793.94+3401.04**	**3125.42+10031.34**
10|Shuttle|	40535.21	**1660.19+628.25**	**3955.81+16193.28**

A partir de estos resultados puede notarse que el tiempo total de *PSC* y *PSR* es mucho menor con respecto a clasificar con todo el conjunto de datos. Por

tanto, debido a la reducción de tiempos de ejecución, en este tipo de casos es donde la selección de prototipos es particularmente útil.

4.6 Comparación experimental entre los métodos propuestos

En esta sección se presenta una comparación experimental entre los diferentes métodos propuestos en este trabajo. De manera análoga a secciones anteriores, esta comparación se realizó usando los clasificadores: *k-NN, LWR, SVM, C4.5* y *NB*. En particular, los métodos a comparar son: *RFPS* (el mejor método restringido flotante), *PSC* y *PSR*. También se ha incluido el muestreo aleatorio (*RS*) para la selección de prototipos. Esta técnica consiste en seleccionar de manera aleatoria un conjunto de n prototipos durante un tiempo t_f y después se elige el conjunto (de entre los resultantes de la selección aleatoria) con la mejor precisión de clasificación.

Los resultados obtenidos se reportan en las tablas 4.51-4.56 y figuras 4.22-4.26. En estas tablas, se indica la diferencia significativa en clasificación con respecto a *ENN+RFPS*.

Tabla 4.50. Descripción de los parámetros usados para *RS* de los resultados reportados en las tablas 4.51-4.56

Método	Tamaño del subconjunto seleccionado (n)	Tiempo del muestreo aleatorio (t_f)
***RS* (1)**	Tamaño del subconjunto seleccionado por *DROP3* (el mejor resultado de retención de entre los métodos *DROP*).	Tiempo de selección del método *PSC* (el método más rápido de los propuestos en este documento).
***RS* (2)**	Tamaño del subconjunto seleccionado por *PSC*	Tiempo de selección del método *DROP3*
***RS* (3)**	Tamaño del subconjunto seleccionado por *PSC*	Tiempo de selección del método *PSC*
***RS* (4)**	Tamaño del subconjunto seleccionado por *PSR*	Tiempo de selección del método *PSR*

Para el muestreo aleatorio es necesario fijar los dos parámetros iniciales n y t_f. En estas tablas se reportan los resultados obtenidos con *RS* usando cuatro distintos parámetros (tabla 4.50). El valor de estos parámetros corresponde a combinaciones de los mismos resultados de retención y tiempo de los métodos *DROP3* (mejor método en retención), *PSC* y *PSR*.

Tabla 4.51. Resultados de clasificación (*Acc*) y retención (*Str*) obtenidos con: Conjunto original (*Orig.*), *RFPS*, *PSC* y *PSR* utilizando *k-NN*, *k*=3.

Datos	Orig.	ENN+RFPS	DROP3+RFPS	DROP5+RFPS	PSC	PSR	RS(1)	RS(2)	RS(3	RS(4)
	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc
Bridges	66.09	**60.00**	56.45	53.63*	56.54	57.63	42.45*	58.45	53.81*	56.36*
Echocardiogram	95.71	**93.23**	91.96	87.67*	86.42*	90.53	87.85	87.21*	86.21	86.96
Glass	71.42	**69.43**	64.48	67.74	59.09	64.85	57.55*	58.02*	56.23*	59.51
Heart Cleveland	82.49	**79.52**	77.21*	79.22	73.27	79.18	80.19	80.15	76.50	76.17*
Hepatitis	79.29	**79.00**	78.20*	76.66*	75.37*	83.16	79.87	81.16*	74.25*	80.50*
Iris	94.66	**93.33**	93.00	93.33	94.66	91.33	85.33*	89.33	90.66*	91.33*
Liver	65.22	**59.98**	61.70	60.03	55.36	63.77*	57.15*	61.47*	60.88*	55.17
Wine	94.44	**93.63**	94.44	93.85	92.67	92.18	80.00*	90.00*	91.18*	90.00
Zoo	93.33	**91.33**	91.33*	91.11	92.22*	**93.33**	93.33	88.88*	88.88	91.11
Promedio	**82.52**	**79.94**	**78.75**	**78.14**	**76.18**	**79.55**	**73.75**	**77.19**	**75.40**	**76.35**

Tabla 4.52. Resultados de retención correspondientes a la tabla 4.50.

Datos	Orig.	ENN+RFPS	DROP3+RFPS	DROP5+RFPS	PSC	PSR	RS(1)	RS(2)	RS(3	RS(4)
	Str	Str	Str	Str	Str	Str	Str	Str	Str	Str
Bridges	100	48.12	14.28	23.71	42.86	47.79	14.78	42.86	42.86	47.79
Echocardiogram	100	86.56	9.85	8.16	19.82	37.68	13.95	19.82	19.82	37.68
Glass	100	29.34	25.75	26.11	38.45	42.36	24.35	38.45	38.45	42.36
Heart Cleveland	100	22.41	12.61	12.53	22.51	36.96	11.44	22.51	22.51	36.96
Hepatitis	100	18.71	8.38	8.67	7.95	33.40	11.47	7.95	7.95	33.40
Iris	100	10.07	9.92	10.00	20.45	38.07	15.33	20.45	20.45	38.07
Liver	100	33.68	16.94	19.64	45.87	35.55	26.83	45.87	45.87	35.55
Wine	100	10.23	8.17	8.30	37.15	42.94	15.04	37.15	37.15	42.94
Zoo	100	71.14	14.81	14.93	41.48	51.11	20.37	41.48	41.48	51.11
Promedio	**100**	**36.70**	**13.41**	**14.67**	**30.73**	**40.65**	**17.06**	**30.73**	**30.73**	**40.65**

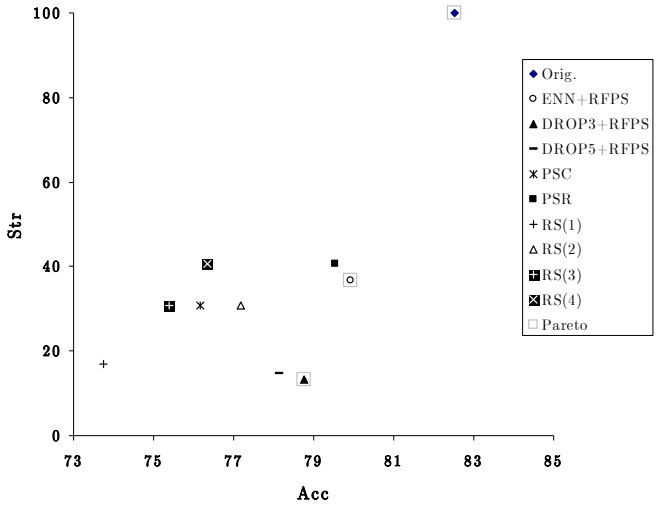


Figura 4.22. Gráfica de dispersión de los resultados de las tablas 4.51-4.52

Tabla 4.53. Resultados de clasificación obtenidos al evaluar los subconjuntos obtenidos con: Conjunto original (*Orig.*), *RFPS, PSC, PSR* y *RS* como entrenamiento para *LWR*

Datos	Orig.	ENN+RFPS	DROP3+RFPS	DROP5+RFPS	PSC	PSR	RS(1)	RS(2)	RS(3	RS(4)
	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc
Glass	57.85	**55.47**	45.64*	48.00*	56.06	56.54	49.97*	55.35	51.73*	51.77*
Iris	98.00	**98.00**	82.00*	77.33*	96.00*	95.33	91.33	94.00*	90.66*	96.00*
Liver	70.13	**70.43**	69.98	71.87	68.57	70.58	67.52*	68.94	68.73	70.18
Wine	92.15	**91.50**	87.05*	88.88*	68.57*	85.98*	65.55*	72.22*	86.66*	86.88*
Promedio	**79.53**	**78.85**	**71.17**	**71.52**	**72.30**	**77.11**	**68.59**	**72.63**	**74.45**	**76.21**

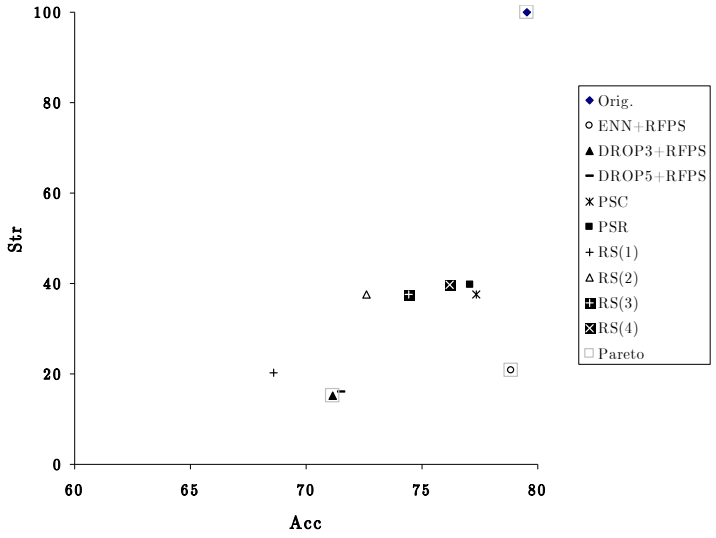


Figura 4.23. Gráfica de dispersión de los resultados de la tabla 4.52

Tabla 4.54. Resultados de clasificación obtenidos al evaluar los subconjuntos obtenidos con: *ENN+RFPS, DROP3+RFPS...DROP5+RFPS, PSC, PSR* y *RS* como entrenamiento para *SVM*

Datos	Orig.	ENN+RFPS	DROP3+RFPS	DROP5+RFPS	PSC	PSR	RS(1)	RS(2)	RS(3)	RS(4)
	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc
Glass	72.29	**71.35**	62.47*	63.54*	65.84*	61.66	61.66	57.87*	62.50	65.34*
Iris	96.00	**96.00**	92.00*	94.66	93.33	95.33	95.33	90.66	93.33	94.00
Liver	69.91	**67.97**	58.56	57.97	64.07	69.02	69.02	58.26*	58.56*	58.84*
Wine	97.18	**97.18**	95.33*	97.18*	95.52	94.96	94.96	65.55*	60.00*	65.55*
Promedio	**83.85**	**83.13**	**77.09**	**78.34**	**79.69**	**80.24**	**80.24**	**68.09**	**68.60**	**70.93**

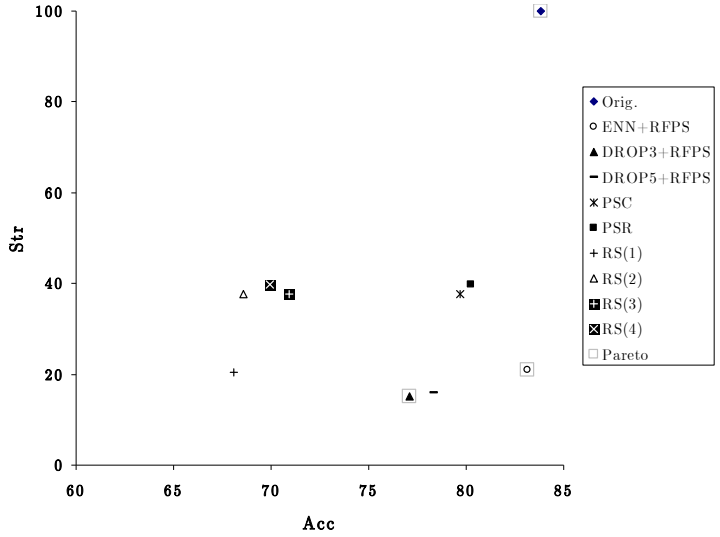


Figura 4.24. Gráfica de dispersión de los resultados de la tabla 4.53

Tabla 4.55. Resultados de clasificación obtenidos al evaluar los subconjuntos obtenidos con: *RFPS, PSC, PSR* y *RS* como entrenamiento para *C4.5*

Datos	Orig.	ENN+RFPS	DROP3+RFPS	DROP5+RFPS	PSC	PSR	RS(1)	RS(2)	RS(3)	RS(4)
	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc
Bridges	65.81	**60.00**	54.54	56.54*	65.81	51.09*	42.36*	61.18	50.54*	58.18*
Echocardiogram	95.71	**93.23**	87.67	90.17*	94.46	95.71	91.96*	94.46*	94.89	94.28
Glass	67.29	**69.43**	55.19*	62.22*	60.58	63.48	61.68*	62.05*	62.53	59.84*
Heart Cleveland	71.96	**79.52**	67.61*	70.60	69.89	71.35*	67.96*	70.56	72.89*	71.32*
Hepatitis	76.70	**79.00**	71.62*	62.79*	73.50	83.20*	73.49	80.62	79.29	80.66
Iris	93.99	**93.33**	84.00	86.00	90.66	93.33	87.33*	91.33*	92.66*	93.33
Liver	63.67	**59.98**	57.15*	58.54*	63.67	65.21*	56.53*	60.81	62.33*	59.15*
Wine	94.44	**93.63**	74.21*	76.86*	90.77	94.44	70.00*	90.00	90.00*	87.77*
Zoo	93.33	**91.33**	72.22*	67.77*	93.33	95.55*	75.55*	85.55*	85.55*	90.00*
Promedio	**80.32**	**79.94**	**69.36**	**70.17**	**78.07**	**79.26**	**69.65**	**77.40**	**76.74**	**77.17**

De acuerdo a estos resultados, se observa que para la selección de prototipos, con los parámetros iniciales utilizados para *RS,* este método presenta un comportamiento competente de clasificación con respecto a los métodos propuestos pero en todos los casos *RS* es superado por *ENN+RFPS* y *PSR*.

Puede notarse que el mejor de entre los métodos propuestos con respecto a la calidad de clasificación fue *ENN+RFPS* seguido de *PSR* y luego *PSC*. Por otra parte, con respecto a retención, el método que mejores resultados obtuvo fue *DROP3+RFPS*.

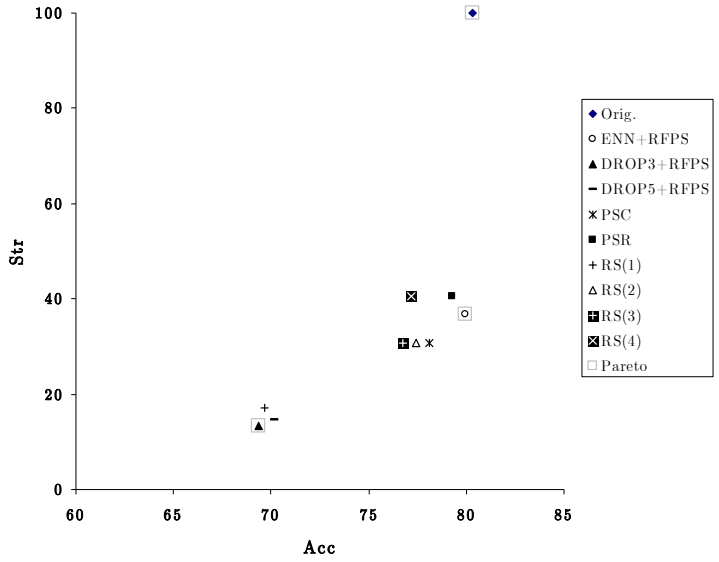


Figura 4.25. Gráfica de dispersión de los resultados de la tabla 4.55

Tabla 4.56. Resultados de clasificación obtenidos al evaluar los subconjuntos obtenidos con: *RFPS, PSC, PSR* y *RS* como entrenamiento para *NB*

Datos	Orig.	ENN+RFPS	DROP3+RFPS	DROP5+RFPS	PSC	PSR	RS(1)	RS(2)	RS(3)	RS(4)
	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc
Bridges	65.81	**58.18**	50.81	52.90	58.63	50.90	49.90*	54.63	52.45*	60.09
Echocardiogram	95.71	**97.32**	83.21	79.64*	84.82*	90.71*	82.14*	82.32*	83.75*	85.35*
Glass	67.29	**52.33**	43.98*	46.70	47.14	60.71*	43.05*	46.42*	48.13*	46.21*
Heart Cleveland	71.96	**80.84**	78.17	80.56	78.20	81.86	77.52	81.49	81.83	78.83
Hepatitis	76.70	**85.29**	75.33	69.87*	79.29*	79.37*	78.62	77.29	78.58	80.66*
Iris	93.99	**95.33**	87.33	90.66	93.33	91.99	92.00	93.33*	91.33*	93.33*
Liver	63.67	**57.78**	56.33	53.40	56.88*	66.94	53.95	53.97	59.47	53.59*
Wine	94.44	**96.66**	78.33	76.66	97.77	92.22	85.55*	94.44*	97.77	95.55*
Zoo	93.33	**93.33**	86.66*	91.11	95.55	95.55	86.66*	86.66*	93.33	92.22*
Promedio	**80.32**	**79.67**	**71.13**	**71.28**	**76.85**	**78.92**	**72.15**	**74.51**	**76.52**	**76.20**

Cabe mencionar que algunos métodos de selección de prototipos no pueden ser aplicados a grandes conjuntos de datos debido al tiempo que requieren para llevar a cabo la selección, pero esta característica afecta poco a *PSC* y *PSR* ya que, como lo mostraron los experimentos, para los conjuntos de datos con mayor dimensionalidad, *PSC* y *PSR* fueron más rápidos que los demás métodos.

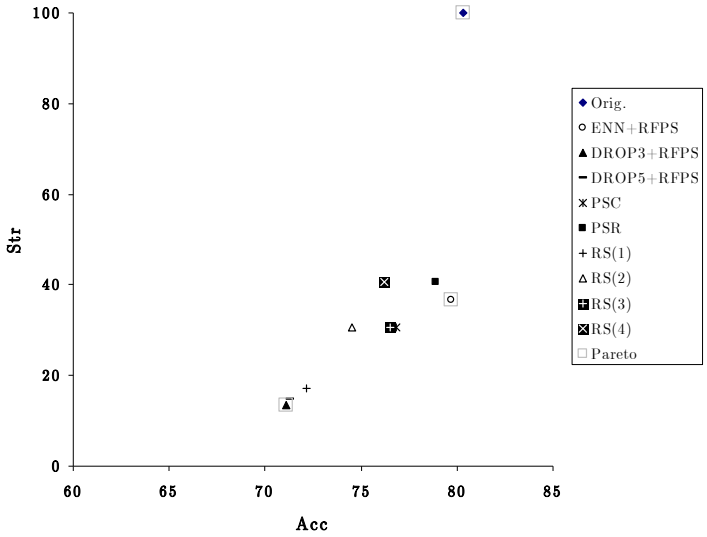


Figura 4.26. Gráfica de dispersión de los resultados de la tabla 4.55

El método con mejor desempeño en calidad de clasificación es *ENN+RFPS* pero debido a que es un método costoso, resulta útil solamente para conjuntos pequeños de datos.

Cuando se requiere procesar conjuntos grandes de datos, la mejor opción es *PSR* cuyo desempeño en calidad de clasificación es similar al de *ENN+RFPS* pero con un tiempo mucho menor de ejecución. Por otra parte, *PSC* obtiene resultados similares a *PSR* pero en menos tiempo de ejecución con respecto a *PSR,* con lo cual, *PSC* puede ser aplicado a conjuntos aún más grandes.

Conclusiones

En la clasificación supervisada, no siempre toda la información en un conjunto de entrenamiento es útil en el proceso de aprendizaje de los clasificadores; por otra parte, el tiempo requerido en los procesos de aprendizaje/clasificación es proporcional al tamaño del conjunto de entrenamiento. Por estas razones surge la necesidad de elegir un subconjunto de prototipos del conjunto de entrenamiento, es decir, aplicar un método de selección de prototipos previo a la etapa de clasificación.

Con base en el estudio del estado del arte del capítulo 2, puede notarse que se han propuesto diversos métodos para la selección de prototipos pero la mayoría de ellos son de tipo *wrapper SCP* (específicamente propuestos para *k-NN*), por lo que los subconjuntos que obtienen son buenos sólo cuando se utiliza el clasificador particular para el que han sido propuestos. En lo que respecta a los métodos *filter*, se han propuesto pocos métodos de este tipo, además de que todos ellos son aplicables solamente a datos numéricos.

En este trabajo se propusieron, evaluaron y compararon los métodos *RFPS, RFPS-Inv* (de tipo *wrapper SCC*) y *PSC, PSR* (de tipo *filter*). Los primeros dos métodos se basan en aplicar de manera restringida la búsqueda secuencial flotante para la selección de prototipos; los restantes dos métodos seleccionan prototipos frontera mediante agrupamientos y relevancia de prototipos, respectivamente. De acuerdo a los resultados reportados, con estos métodos se alcanzó el objetivo de este trabajo de investigación (métodos competentes de tipo *wrapper SCC* y *filter* aplicables a datos mezclados) y con base en los experimentos reportados en este documento, se concluye lo siguiente:

- Si se requiere procesar conjuntos pequeños de datos y utilizar el clasificador *k-NN,* los mejores métodos para seleccionar prototipos son *DROP3* y *DROP5.*
- Si se requiere usar otros clasificadores la mejor opción, de entre los métodos propuestos en este trabajo, en cuanto a precisión es *ENN+RFPS.*
- Para conjuntos grandes de datos las mejores opciones son *PSC* y *PSR,* siendo mejor *PSR* si lo que se busca es una mejor calidad de clasificación o *PSC* si lo que se busca es mayor velocidad.

Como se mostró en los experimentos referentes a los tiempos de ejecución, los tiempos totales de los métodos *PSC* y *PSR* son menores con respecto a entrenar y clasificar con los conjuntos de entrenamiento sin previa selección, principalmente para los clasificadores *k-NN, LWR* y *SVM.* Por lo que, la utilidad de la selección de prototipos es notoria para este tipo de clasificadores.

De acuerdo a los experimentos reportados en este trabajo, el mejor método en cuanto a la calidad de clasificación es *RFPS,* específicamente *ENN+RFPS,* pero, sólo es conveniente utilizar este método en conjuntos pequeños de datos, debido a su alto costo computacional.

Aportaciones del trabajo de investigación

La aportación de este trabajo de investigación son cuatro métodos para la selección de prototipos:

1) El método *RFPS* de tipo *wrapper*, el cual está basado en la búsqueda restringida flotante (exclusión condicional seguida de inclusión

condicional) para la selección de prototipos. Este método permite el uso de cualquier clasificador durante el proceso de selección.

2) El método *RFPS-Inv* de tipo *wrapper*, el cual también se basa en la búsqueda restringida flotante pero en dirección inversa a *RFPS*, es decir, la inclusión condicional seguida de la exclusión condicional. Las características de este método son análogas a las de *RFPS,* pero este último es mejor que *RFPS-Inv.*

3) El método *PSC* de tipo *filter*, basado en agrupamiento para seleccionar prototipos frontera y algunos prototipos interiores de cada clase. Este método genera grupos de prototipos y para seleccionar prototipos borde analiza los grupos no homogéneos debido a que este tipo de grupos corresponden a regiones en las que se encuentran prototipos similares pero pertenecientes a distintas clases, es decir regiones borde. Los prototipos interiores seleccionados por este método corresponden a los prototipos representativos de cada grupo homogéneo.

4) El método *PSR* de tipo *filter*, el cual, selecciona los prototipos más relevantes de cada clase y a partir de estos prototipos selecciona prototipos frontera. Este método considera la relevancia de los prototipos con base en la similaridad promedio que tienen los prototipos de la misma clase, en este sentido, el prototipo más relevante de la clase es el más similar a los demás.

Trabajo Futuro

Los métodos *filter* propuestos en este trabajo requieren de un parámetro inicial, en particular, el número de grupos a crear en *PSC* y la cantidad de prototipos relevantes a seleccionar en la fase inicial de *PSR*. Por lo que, como

trabajo futuro, se propondrán técnicas rápidas de selección de prototipos en las que se pueda ajustar de manera automática los parámetros iniciales. Estos parámetros podrían ajustarse, por ejemplo, analizando el grado de homogeneidad de los grupos generados (para el caso de *PSC*) y el grado de redundancia de prototipos de la misma clase (par el caso de *PSR*).

Anexo

Trabajos Publicados

Los artículos publicados derivados de este trabajo de investigación son los siguientes:

Congresos:

1) J. Arturo Olvera-López et al. **Restricted Sequential Floating Search applied to Object Selection**. In: P. Perner (Ed.): MLDM 2007, LNAI 4571, pp. 694–702, 2007. Springer-Verlag.

2) J. Arturo Olvera-López et al. **Object Selection Based on Clustering and Border Objects**. In: M. Kurzynski et al. (Eds.): Computer Recognition Systems 2, ASC 45, pp. 27–34, 2007. Springer-Verlag.

3) J. Arturo Olvera-López et al. **Mixed Data Object Selection Based on Clustering and Border Objects**. In: L. Rueda, D. Mery, and J. Kittler (Eds.): CIARP 2007, LNCS 4756, pp. 674–683, 2007. Springer-Verlag.

4) J. Arturo Olvera-López et al. **Prototype Selection Via Prototype Relevance**. In: J. Ruiz-Shulcloper and W.G. Kropatsch (Eds.): CIARP 2008, LNCS 5197, pp. 153–160, 2008. Springer-Verlag.

Revistas JCR (*Journal Citation Reports*):

5) J. A. Olvera-López et al. **"Prototype Selection based on Sequential Search".** To appear in *Intelligent Data Analysis* vol. 13(4).

6) J. Arturo Olvera-López et al. **"A new Fast Prototype Selection Method based on Clustering".** *Pattern Analysis and Applications*, special issue. (Accepted paper).

Referencias

Aguilar-Ruiz J. S., Nepomuceno J. A., Díaz-Díaz N., and Nepomuceno I. (2006). "A Measure for Data Set Editing by Ordered Projections". LNAI 4031. M. Alí and R. Dapaoigny (Eds.): IEA/AIE 2006, pp. 1339-1348.

Aha D. W. (1991). "Instance based learning algorithms". Machine Learning 6 (1), pp, 37-66.

Angiulli F. (2007). "Condensed Nearest Neighbor Data Domain Description". IEEE Transaction on Pattern Analysis and Machine Intelligence, 29 (10), pp. 1746-1758.

Asuncion A., Newman D.J. (2007). UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html] Irvine CA: University of California, School of Information and Computer Science.

Atkeson C. G., Moorel A. W., Schaal S. (1997). "Locally Weighted Learning". In Artificial Intelligence Review, 11(1-5), pp. 11-73.

Bezdek James C., Kuncheva L. I. (2001). "Nearest Prototype Classifier Designs: An Experimental Study". International Journal of Intelligent Systems, 16 (12), pp. 1445-1473.

Blum A. L., Langley P. (1997). "Selection of relevant features and examples in machine learning". Artificial Intelligence 97, pp. 245-271.

Brighton H., Mellish. C. (2002). "Advances in Instance-Based Learning Algorithms". Data Mining and Knowledge Discovery, 6, pp. 153-172.

Cano J. R., Herrera F., Lozano M. (2003). "Using Evolutionary Algorithms as Instance Selection for Data Reduction in KDD: an experimental study". IEEE Transactions on Evolutionary Computation, Vol. 7, No. 6, pp. 561-575.

Cerverón V., Ferri F. J. (2001). "Another move toward the minimum consistent subset: a tabu search approach to the condensed nearest neighbour rule". IEEE Transactions on Systems. Man and Cybernetics, Part B, Vol. 31, No.3, pp. 408-413.

Chaudhuri B. B. (1996). "A new definition of neighborhood of a point in multi-dimensional space". Pattern Recognition Letters, Vol. 17, pp. 11-17.

Chien-Hsing C., Bo-Han K., and Fu C. (2006). The Generalized Condensed Nearest Neighbor Rule as A Data Reduction Method. 18th International Conference on Pattern Recognition, Vol. 2 pp. 556-559.

Covert T. M. and Hart P.E. (1967). "Nearest Neighbor Pattern Classification". IEEE Transactions on Computers, 13, pp. 21-27.

Devijver P. A., Kittler J. (1980). On the edited nearest neighbor rule. Proceedings of the Fifth International Conference on Pattern Recognition, pp. 72-80.

Dietterich T. G. (1998). „Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms". Neural Computation, 10 (7), pp. 1895-1924.

Eick C. F., Zeidat N., and Vilalta R. (2004). "Using Representative-Based Clustering for Nearest Neighbor Dataset Editing". 4th IEEE International Conference on Data Mining (ICMD04), pp. 375-378.

Fogel D. B. (1995). *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. Ed. IEEE press.

Friedman J. H., Bentley J. L., and Finkel R. A. (1997). "An algorithm for finding best matches in logarithmic expected time". ACM Trans. Math. Software 3 (3), pp. 209-226.

García S. J. R. and Martínez T. J. F. (1999). "Extension to C-means algorithm for the use of similarity functions". LNAI 1704. J.M. Zytkow and J. Rauch (Eds.): PKDD'99, LNAI 1704, pp. 354-359.

Glover F. (1986). "The General Employee Scheduling Problem: An Integration of Management Science and Artificial Intelligence". Computers and Operations Research, Vol. 13, No. 4, pp. 563-593.

Goldberg D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning.* Ed. Addison-Wesley.

Han J., Kamber M. (2001). *Data mining: Concepts and Techniques*. Ed. Morgan Kaufmann publishers.

Hart P. E. (1968). "The Condensed Nearest Neighbor Rule". IEEE Transactions on Information Theory, 14, pp. 515-516.

Hearst M. A. (1998). "Trends & Controversies. Support Vector Machines", IEEE Intelligent Systems, pp. 18-21.

Holland J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press.

Ke-Ping Z., Shui-Geng Z., Ji-Hong G., and Ao-Ying A. (2003). "C-Pruner: A new instance pruning algorithm". Proceedings of 2nd IEEE International Conference on Machine Learning and Cybernetics, Vol. 1, pp. 94-99.

Kittler J. (1986). "Feature selection and extraction". In Young and Fu (Eds.). Handbook of pattern recognition and image processing. New York: Academic Press, pp. 203-217.

Kuncheva L. I. (1995). "Editing for the *k*-nearest neighbors rule by a genetic algorithm". Pattern Recognition Letters, Vol. 16, pp. 809-814.

Kuncheva L. I. (1997). "Fitness functions in editing k-NN referent set by genetic algorithms". Pattern Recognition, Vol. 30, pp. 1041-1049.

Kuncheva L. I., Bezdek J. C. (1998). "Nearest prototype classification: clustering. Genetic algorithms. Or random search?".IEEE Transactions on Systems, Man and Cybernetics, Part C, Vol. 28, No. 1, pp. 160 – 164.

Leung Y., Zhang. J.S.. Xu. Z.B. (2000). "Clustering by scale-space filtering". IEEE Trans. on Pattern Analysis and Machine Intelligence, 22, 1396-1410.

Li M. and Zhi-Hua Z. (2005) "SETRED: Self-training with Editing". LNAI 3518. T.B. Ho, D. Cheung, and H. Liu (Eds.): PAKDD 2005, pp. 611–621.

Liu H., Motoda H. (2002). "On Issues of Instance Selection". Data Mining and Knowledge Discovery, 6, pp. 115-130.

Lozano M., Sánchez J. S., and Pla F. (2003). "Reducing Training Sets by NCN-Based Exploratory Procedures". LNCS 2652. F.J. Perales et al. (Eds.): IbPRIA 2003, pp. 453-461.

Lumini A., Nanni L. (2006). "A clustering method for automatic biometric template selection". Pattern Recognition 39, pp. 495-497.

Mitchell T. M. (1997). *Machine Learning*. Ed. WCB/McGraw-Hill.

Narayan B. L., Murthy C. A. and Pal S. K. (2006). "Maxdiff kd-trees for data condensation". Pattern Recognition Letters 27, pp. 187-200.

Olvera-López J. A., Carrasco-Ochoa J. A. and Martínez-Trinidad J. F. (2005a). "Sequential Search for Decremental Edition". LNCS 3578. M. Gallagher, J. Hogan, and F. Maire (Eds.): IDEAL 2005, pp. 280-285.

Olvera-López J. A., Martínez-Trinidad J. F. and Carrasco-Ochoa J. A. (2005b). "Edition Schemes based on BSE". LNCS 3773. M. Lazo and A. Sanfeliu (Eds.): CIARP 2005, pp. 360-367.

Pudil P., Ferri F.J., Novovicová J. and Kittler J. (1994). "Floating Search Methods for Feature Selection with Nonmonotonic Criterion Functions". In: Proceedings of the 12th International Conference on Pattern Recognition. IEEE Computer Society Press, pp. 279-283. 1994.

Quinlan J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Raicharoen T. and Lursinsap C. (2005). "A divide-and-conquer approach to the pairwise opposite class-nearest neighbor (POC-NN) algorithm". Pattern Recognition Letters, 26(10), pp. 1554-1567.

Riquelme J.C., Aguilar-Ruíz J.S., Toro M. (2003). "Finding representative patterns with ordered projections". Pattern Recognition, 36, pp. 1009-1018.

Ritter G. L., Woodruff H.B., Lowry S. R. and Isenhour T. L. (1975). "An Algorithm for a Selective Nearest Neighbor Decision Rule". IEEE Transactions on Information Theory, 21-6, pp. 665-669.

Rodríguez A. F., Vadera S., and Sucar L. E. (2000). "A Probabilistic Exemplar-Based Model for Case-Based Reasoning". LNAI 1793. O. Cairo, L. E. Sucar, and F. J. Cantu (Eds.): MICAI 2000, pp. 40-51.

Sánchez J. S., Barandela R., Marqués A. I., Alejo R., Badenas J. (2003). "Analysis of new techniques to obtain quality training sets". Pattern Recognition Letters, 24, pp. 1015-1022.

Spillmann B., Neuhaus. M., Bunke. H., Pekalska E. and Duin. R.P.W. (2006). "Transforming Strings to Vector Spaces Using Prototype Selection". LNCS 4109. D.-Y. Yeung et al. (Eds.): SSPR&SPR 2006, pp. 287–296.

Srisawat A., Phienthrakul T., and Kijsirikul B. (2006). "SV-kNNC: An Algorithm for Improving the Efficency of k-Nearest Neighbr". LNAI 4099. Q. Yang and G. Webb (Eds.): PRICAI 2006, pp. 975-979.

Tomek I. (1976). "An Experiment with the Edited Nearest-Neighbor Rule". IEEE Transactions on Systems, Man, And Cybernetics, 6-6, pp. 448-452.

Vapnik V. (1995). *The Nature of Statistical Learning Theory.* Springer-Verlag. New York 1995.

Vázquez F., Sánchez S. and Pla F. (2005). "A Stochastic Approach to Wilson's Editing Algorithm". LNCS 3523. J.S. Marques et al. (Eds.): IbPRIA 2005, pp. 35-42.

Vojtech F. and Václav H. (2004). "Statistical Pattern Recognition Toolbox for Matlab". Research report. Center for Machine Perception Department of Cybernetics. Faculty of Electrical Engineering. Czech Technical University.

Wilson D. R. and Martínez T. R. (2000). "Reduction Techniques for Instance-Based Learning Algorithms". Machine Learning, 38, pp. 257-286.

Wilson D. L. (1972). "Asymptotic Properties of Nearest Neighbor Rules Using Edited Data". IEEE Transactions on Systems, Man, And Cybernetics, 2-3, pp. 408-421.

Witten I. H., Frank E. (2005). "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco.

Yuangui L., Zhonghui H., Yunze C. and Weidong Z. (2005). "Support Vector Based Prototype Selection Method for Nearest Neighbor Rules". LNCS 3610. L. Wang, K. Chen, and Y.S. Ong (Eds.): ICNC 2005, pp. 528-535.

Zhang H., Sun G. (2002). "Optimal reference subset selection for nearest neighbor classification by tabu search". Patter Recognition 35, pp. 1481-1490.