

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTA DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**LS-SVM BASADO EN OPTIMIZACIÓN POR
ENJAMBRES DE PARTÍCULAS PARA
CLASIFICACIÓN DE ACCIDENTES DE TRÁNSITO**

FÉLIX HERNÁN CASTRO FUENTES

INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO CIVIL EN INFORMÁTICA

DICIEMBRE 2010

Pontificia Universidad Católica de Valparaíso
Faculta de Ingeniería
Escuela de Ingeniería Informática

**LS-SVM BASADO EN OPTIMIZACIÓN POR
ENJAMBRES DE PARTÍCULAS PARA
CLASIFICACIÓN DE ACCIDENTES DE TRÁNSITO**

FÉLIX HERNÁN CASTRO FUENTES

Profesor Guía: **Nibaldo Rodríguez Agurto**

Carrera: **Ingeniería Civil Informática**

DICIEMBRE 2010

Dedicado especialmente a mi familia, a Natalia, a mis amigos más cercanos, a mis compañeros de universidad y a quienes me guiaron y apoyaron durante la carrera para llegar a ser un profesional.

Índice

Glosario de Términos	iii
Lista de Abreviaturas	iv
Lista de Figuras	v
Lista de Tablas	vi
Resumen.....	vii
Abstract	viii
1 Introducción	1
2 Marco General del Proyecto.....	3
2.1 Objetivo General.....	3
2.2 Objetivos Específicos	3
2.3 Planteamiento del Problema	3
2.4 Alcance del Estudio	3
3 Marco Teórico	4
3.1 Máquinas de Soporte Vectorial (SVM).....	4
3.1.1 Definición Formal de SVM.....	4
3.1.2 Descripción del Modelo	5
3.1.3 Funciones de decisión	6
3.1.4 Clasificación SVM, Caso Linealmente Separable	7
3.1.5 Clasificación SVM, Caso Linealmente No Separable	9
3.1.6 Clasificación SVM, Caso No lineal	10
3.1.7 Características claves de SVM.....	12
3.1.8 Algoritmos para SVM	12
3.2 Mínimos Cuadrados SVM (LS-SVM).....	13
3.2.1 Definición Formal	13
3.3 Optimización por enjambres de partículas	15
3.3.1 Terminología y fundamentos de PSO	16
3.3.2 Definición Formal	17
3.3.3 Algoritmo	18
3.3.4 Variantes.....	21
4 Estado del Arte Accidentes de Tránsito	30
5 Descripción de los datos en estudio	33
5.1 Obtención	33

5.2	Estructura de los datos	33
6	Modelo LS-SVM – PSO	37
6.1	Descripción del Modelo General	37
6.2	Métricas de rendimiento para evaluar el clasificador.....	41
7	Implementación y Experimentación del Trabajo	45
7.1	Pre proceso de los Datos.....	45
7.1.1	Integración de los datos.....	45
7.1.2	Selección y codificación de los datos.....	45
7.1.3	Transformación de los datos	47
7.2	Software y Hardware Utilizado	47
7.3	Clasificación LS-SVM con PSO	48
7.3.1	PSO con Factor de Inercia Lineal	49
7.3.2	IPSO (Improved Particle Swarm Optimization)	54
7.3.3	PSO con Adaptación Dinámica (DAPSO)	58
8	Análisis de Resultados	64
8.1	Comparación entre variantes de PSO	64
8.2	Comparación con trabajos relacionados	68
9	Conclusión.....	70
10	Bibliografía.....	72
Anexos	74
10.1	A: Planilla SIEC 2.....	74

Glosario de Términos

Algoritmo Evolutivo: constituyen una técnica general de resolución de problemas de búsqueda y optimización inspirada en la teoría de la evolución de las especies y la selección natural.

Hiperplano: espacio vectorial de dimensión n en relación a un origen fijo. En SVM es utilizado para separar y diferenciar clases.

Kernel: función que permite mapear los datos de un espacio cualquiera a un espacio de mayor dimensión.

Lagrange: método para trabajar con funciones de varias variables en donde se desea maximizar o minimizar, sujeto a ciertas restricciones.

Metaheurística: consiste en una estrategia de alto nivel que guía a otras heurísticas para buscar soluciones que sean factibles en dominios donde la tarea es compleja.

Minería de datos: conjunto de técnicas y tecnologías que permiten explorar grandes bases de datos, de manera automática o semiautomática, con el objetivo de encontrar patrones repetitivos, tendencias o reglas que expliquen el comportamiento de los datos en un determinado contexto.

Programación Cuadrática: problema de optimización, en donde se debe minimizar o maximizar una función cuadrática de varias variables sujetas a restricciones lineales sobre dichas variables.

Lista de Abreviaturas

CONASET	: Comisión Nacional de Seguridad de Tránsito.
DAPSO	: Particle Swarm Optimizer with Dynamic Adaptation.
IPSO	: Improved Particle Swarm Optimization.
LS-SVM	: Least Squares Support Vector Machines.
PSO	: Particle Swarm Optimization.
QP	: Quadratic Programming.
RBF	: Radial Basis Function.
ROC	: Receiver Operating Characteristic.
SRM	: Structural Risk Minimization.
SVM	: Support Vector Machines.
UAC	: Area Under ROC curve.
VPP	: Valor Predictivo Positivo.
VPN	: Valor Predictivo Negativo.

Lista de Figuras

Figura 3.1 Hiperplano de separación óptima.....	5
Figura 3.2 Funciones de decisión	6
Figura 3.3 (a) Caso Linealmente Separable (b) Caso Linealmente no Separable	7
Figura 3.4 Caso no Lineal	10
Figura 3.5 Swarm en la Naturaleza	15
Figura 3.6 Actualización de una Partícula.....	19
Figura 3.7 Diagrama de Flujo PSO Convencional	20
Figura 3.8 Ejemplo jerarquía definida mediante un árbol regular con $h=3$, $d=4$, $m=21$	25
Figura 3.9 (a) Circular con $k=2$. (b) Circular con conexiones variantes y 2 subvecindarios. (c) Rueda. (d) Ruedas con conexiones variantes	28
Figura 3.10 (a) Arcos aleatorios. (b) Piramidal. (c) Malla. (d) Toroidal.....	28
Figura 5.1 Entidades base de datos CONASET	34
Figura 6.1 Modelo General de la solución	38
Figura 6.2 Esquema para validación cruzada para 4 subconjuntos	40
Figura 6.3 Curva ROC.....	43
Figura 7.1 Costo vs Iteración, PSO F. Inercia Lineal, fitness costo validación cruzada.....	50
Figura 7.2 Costo vs Iteración, PSO F. Inercia Lineal, fitness error absoluto medio	52
Figura 7.3 Gráfico Comparativo Exactitud entre Fitness, PSO F. Inercia Lineal	53
Figura 7.4 Costo vs Iteración, IPSO, fitness costo validación cruzada	55
Figura 7.5 Costo vs Iteración, IPSO, fitness error absoluto medio	56
Figura 7.6 Gráfico Comparativo Exactitud entre Fitness, IPSO	58
Figura 7.7 Costo vs Iteración, DAPSO, fitness costo validación cruzada	59
Figura 7.8 Costo vs Iteración, DAPSO, fitness error absoluto medio.....	61
Figura 7.9 Gráfico Comparativo Exactitud entre Fitness, DAPSO.....	62
Figura 8.1 Gráfico Exactitud LS-SVM, variantes de PSO.....	65
Figura 8.2 Gráfico Sensibilidad LS-SVM, variantes de PSO.....	65
Figura 8.3 Gráfico Especificidad LS-SVM, variantes de PSO.....	66
Figura 8.4 Gráfico VPP LS-SVM, variantes de PSO	66
Figura 8.5 Gráfico VPN LS-SVM, variantes de PSO	67
Figura 8.8 Exactitud SVM vs LS-SVM PSO F. Inercia Lineal.....	68
Figura 8.9 Redes Bayesianas vs LS-SVM PSO con Factor de Inercia Lineal	69

Lista de Tablas

Tabla 4.1 Exactitud SVM usando kernel función de base radial	31
Tabla 4.2 Sensibilidad y especificidad etapa de validación	32
Tabla 6.1 Codificación Partícula	39
Tabla 6.2 Matriz de Confusión.....	41
Tabla 7.1 Atributos seleccionados para el modelo	46
Tabla 7.2 Formación de clases	47
Tabla 7.3 Parámetros de entrada PSO - kernel RBF	49
Tabla 7.4 Resultados PSO F. Inercia Lineal, fitness costo validación cruzada.....	49
Tabla 7.5 Resultados para varias muestras PSO F. Inercia Lineal, fitness costo validación cruzada.....	51
Tabla 7.6 Resultado PSO F. Inercia Lineal, fitness error absoluto medio	51
Tabla 7.7 Resultados testing para varias muestras PSO F. Inercia Lineal, fitness error absoluto medio	53
Tabla 7.8 Resultados IPSO, fitness costo validación cruzada.....	54
Tabla 7.9 Resultados para varias muestras IPSO, fitness costo validación cruzada	55
Tabla 7.10 Resultados IPSO, fitness error absoluto medio	56
Tabla 7.11 Resultados testing para varias muestras IPSO, fitness error absoluto medio.....	57
Tabla 7.12 Resultados DAPSO, fitness costo validación cruzada	58
Tabla 7.13 Resultados para varias muestras DAPSO, fitness costo validación cruzada.....	60
Tabla 7.14 Resultados DAPSO, fitness error absoluto medio.....	60
Tabla 7.15 Resultados testing para varias muestras DAPSO, fitness error absoluto medio	62
Tabla 8.1 Comparación training LS-SVM con variantes PSO.....	64
Tabla 8.2 Comparación ganancia de exactitud entre modelos	68

Resumen

Los accidentes de tránsito han sido analizados mediante diversas técnicas y algoritmos de minería de datos, tales como redes neuronales artificiales, árboles de decisión y redes bayesianas, sin embargo, LS-SVM no ha sido utilizada con frecuencia en los siniestros de tránsito. Si bien, LS-SVM es una técnica bastante robusta, presenta un problema combinatorial en la estimación y ajuste de sus parámetros de entrada, ya que el rendimiento del modelo de clasificación depende en gran parte de estos. Dado lo anterior, este trabajo tiene por objetivo principal desarrollar y evaluar un modelo de clasificación utilizando máquinas de soporte vectorial con algoritmos de optimización de enjambre de partículas, para clasificar el grado de severidad (lesionado o ileso) en el cual resultan las personas involucradas en accidentes de tránsito de la región de Valparaíso. Para ello, se confeccionan distintos modelos de clasificación, utilizando variantes de PSO para el ajuste de los parámetros de la máquina. El análisis de resultados indica que PSO con factor de inercia lineal obtuvo un mejor rendimiento, alcanzado un 82,04% de exactitud promedio de clasificación, lo que supera a trabajos similares realizados en la Escuela de Ingeniería en Transportes de la PUCV y trabajos internacionales. Finalmente se concluye que las máquinas de soporte vectorial si son capaces de presentar un buen grado de generalización para clasificar el estado en el cual resultan las personas (lesionado o ileso) de la región de Valparaíso

Palabras Claves: *Clasificación, Máquinas de Soporte Vectorial, Mínimos cuadrados -SVM, Optimización por Enjambre de Partículas, Accidentes de Tránsito.*

Abstract

Traffic accidents have been analyzed through many techniques and data mining algorithms such as artificial neuronal networks, decision trees and Bayesian networks, however LS-SVM haven't been used frequently on traffic accidents. Although LS-SVM is a quite robust technique, it presents combinatorial problems with the estimation of its entry parameters, since the classification parameters performance depends, in great measure, on this. Given this, the main aim of this paper work is to develop and assess a classification model using vector support machines with particle swarm optimization algorithms, this is for classify the severity grade (injured or uninjured) of people involved in the traffic accident in the Valparaíso region. For this, we are going to elaborate different classification models, using variants of the PSO for the machine parameters adjustment. The results analysis indicates that PSO with linear inertia factor obtained better performance, reaching 82,04% of classification average exactitude, which overcomes similar investigations developed in the PUCV Transportation Engineering school and international investigations. Finally, we conclude that vector support machines are able to generalize in a good range for classify the severity grade (injured or uninjured) of people involved in the traffic accidents in the Valparaiso region.

Key Words: *Classification, Support Vector Machine, Least Squares Support Vector Machines, Particle Swarm Optimization, Traffic Accidents.*

1 Introducción

Según la Organización Mundial de la Salud (OMS), todos los años fallecen más de 1,2 millones de personas en las vías de tránsito del mundo, y entre 20 y 50 millones sufren traumatismos no mortales, siendo la novena causa de mortalidad mundial y se estima que para el año 2030 escale hasta la quinta posición [1].

En Chile la situación no es menor, los siniestros de tránsito se han posicionado como una verdadera epidemia social, alcanzando cifras alarmantes y complejas de abordar. Según la Comisión Nacional de Seguridad de Tránsito (CONASET), solamente en 2009 se contabilizaron un total de 56.330 accidentes, siendo la colisión el tipo de siniestro más recurrente. La región de Valparaíso es la segunda región con más siniestro a nivel nacional y el 2009 contribuyó con un total de 6.662 siniestros, en los cuales se perdieron 134 vidas humanas [2].

Muchos investigadores han propuesto modelos predictivos acerca de las circunstancias en las cuales es más probable que ocurra algún siniestro de tránsito, sin embargo, ninguna ha sido absolutamente certera para poder controlarlos o reducirlos. Mediante técnicas de Minería de Datos o *Data Mining*, es posible obtener patrones o identificar las variables más significativas que ayuden y faciliten a esclarecer las condiciones en que ocurran los accidentes, y así poder controlar y reducir los niveles de accidentes de tránsito.

Dentro de la minería de datos existen diversas técnicas para el análisis. Una de ellas corresponde a las Maquinas de Soporte Vectorial o *Support Vector Machines* (SVM), la cual es una técnica de reconocimientos de patrones basada en la metodología de aprendizaje supervisado usada para regresión y clasificación. Actualmente SVM para clasificación binaria es una técnica bastante robusta, la que se ha consolidado dentro de la última década por sobre técnicas más antiguas de minería de datos, como las redes neuronales. Sin embargo, presenta algunas desventajas importantes como la resolución de un sistema dual con programación cuadrática y el problema combinatorial en la estimación y ajuste de los parámetros de entrada de su modelo. Generalmente, dichos parámetros deben ser ajustados por el investigador en base a su experiencia, lo que no siempre conlleva a asignar valores óptimos afectando directamente al rendimiento del modelo. Los parámetros a estimar son:

- C : Parámetro que controla el *trade off* entre la maximización del margen y la minimización del error de entrenamiento.
- Parámetros del *Kernel*: dependen del tipo de *kernel* a utilizar, estos son
 - d, t : Polinomial
 - σ^2 : Función de Base Radial
 - s, θ : Perceptrón de dos capas o Sigmoidal.

El método de los mínimos cuadrados para SVM o *Least Squares Support Vector Machines* (LS-SVM) es una reformulación de SVM, el cual elimina la desventaja de la resolución de un sistema dual con programación cuadrática, vía un sistema lineal de ecuaciones, lo que sin duda disminuye costos de procesamiento pudiendo abarcar una cantidad más considerable de datos de entrenamiento. Sin embargo, el rendimiento de la clasificación con LS-SVM sigue dependiendo en gran parte de los parámetros nombrados anteriormente.

Para salvar el problema anterior se ha escogido en este trabajo de investigación el algoritmo de Optimización por Enjambres de Partículas o *Particle Swarm Optimization* (PSO), el cual es una metaheurística evolutiva y de búsqueda basada en el comportamiento social de seres vivos en la naturaleza, tales como enjambres de insectos, cardúmenes de peces y bandada de pájaros.

Por lo tanto, este trabajo de investigación desarrolla y evalúa un modelo de clasificación utilizando máquinas de soporte vectorial con algoritmos de optimización de enjambre de partículas, para clasificar el grado de severidad (lesionado o ileso) en el cual resultan las personas involucradas en accidentes de tránsito de la región de Valparaíso, y así esclarecer las características más significativas e influyentes en los accidentes de tránsito. Para ello, se crean distintos modelos de clasificación utilizando variantes de PSO.

En los siguientes capítulos se presenta en detalle el trabajo de investigación realizado a lo largo de proyecto 1 y 2. Es así que en el comienzo del documento se presentan el marco general del proyecto, describiendo los objetivos y planteamiento del problema. Más adelante se presenta un marco conceptual de las técnicas de minería de datos SVM y LS-SVM para clasificación, de manera de introducir los fundamentos básicos para su comprensión. También se analiza la metaheurística optimización por enjambres de partículas PSO y algunas de sus variantes. Luego se presenta el diseño del modelo general de clasificación, el cual integra LS-SVM con PSO, y junto con este las métricas para evaluar su rendimiento. Posteriormente se implementan los distintos modelos, se contrastan y analizan los resultados entre las variantes implementadas y trabajos relacionados. Finalmente se concluye sobre lo realizado.

2 Marco General del Proyecto

2.1 Objetivo General

Desarrollar y evaluar un modelo de clasificación utilizando máquinas de soporte vectorial con algoritmos de optimización de enjambre de partículas, para clasificar el grado de severidad en el cual resultan las personas involucradas en accidentes de tránsito de la región de Valparaíso.

2.2 Objetivos Específicos

- Explicar el funcionamiento de las Máquinas de Soporte Vectorial y Optimización por Enjambre de Partículas (PSO).
- Diseñar la estructura del modelo y estimar los parámetros de LS-SVM mediante el uso de PSO con sus variantes.
- Implementar y contrastar resultados de los modelos propuestos.

2.3 Planteamiento del Problema

Si bien la técnica de LS-SVM es bastante robusta, presenta un problema combinatorial en la estimación y ajuste de sus parámetros de entrada, ya que su rendimiento depende en gran parte de estos. Es por esto, que se desea estimar la mejor combinación de parámetros, mediante la metaheurística Optimización de Enjambre de Partículas.

Por otra parte, el departamento de investigación de la Escuela de Ingeniería en Transporte de la Pontificia Universidad Católica de Valparaíso han utilizado técnicas de minería de datos para el análisis de accidentes de tránsito, obteniendo buenos resultados, sin embargo, SVM y LS-SVM no han sido utilizadas, más aún, estudios e investigaciones alrededor del mundo relacionados con SVM y LS-SVM aplicados a accidentes de tráfico, son bastantes escasos.

En base a lo anterior, surge la idea de aplicar estas técnicas de clasificación y optimización a de accidentes de tránsito con fin de construir un modelo de clasificación que logre diferenciar lo más exactamente posible el estado en el cual resultan las personas involucradas en accidentes de tránsito.

2.4 Alcance del Estudio

En vista de los estudios realizados por la Escuela de Ingeniería en Transporte de la PUCV, el estudio se enmarca en accidentes de tránsito de la región de Valparaíso durante el periodo 2003 al 2009. Por ello es que se pretende crear un modelo de clasificación para determinar el estado en el cual resultan las personas, estos son lesionados o ilesos.

3 Marco Teórico

3.1 Máquinas de Soporte Vectorial (SVM)

Actualmente los datos almacenados en bases de datos crecen de manera acelerada en todo el mundo. Por otra parte, se necesitan técnicas que ayuden y faciliten al análisis y comprensión de dichos datos. Estas técnicas son provistas por la minería de datos, las cuales pueden ser supervisadas o no supervisadas. La clasificación de datos en el caso de SVM es descrita como una técnica de aprendizaje supervisado.

Un proceso de clasificación supervisado incluye dos fases: entrenamiento y prueba. En la fase de entrenamiento o *training* un conjunto de datos inicial es usado para decidir que parámetros deberán ser ponderados y combinados con el objetivo de separar las clases, de esta manera construir un clasificador para la fase siguiente. El aprendizaje intenta descubrir una representación óptima a partir del conjunto de datos cuya etiqueta de clase es conocida por el investigador. En la fase de prueba o *testing*, el clasificador determinado en la fase de entrenamiento es aplicado a un conjunto de datos u objetos (conjunto de prueba) cuyas etiquetas de clase se desconoce. De esta forma clasificar los elementos y comparar con las etiquetas reales para determinar la efectividad del modelo.

El presente capítulo introduce el marco conceptual de la técnica de Maquinas de Soporte Vectorial.

3.1.1 Definición Formal de SVM

Las Maquinas de Soporte Vectorial (SVM) son una técnica de reconocimientos de patrones basada en la metodología de aprendizaje, generando resultados robustos y satisfactorios. Fueron desarrolladas como una herramienta robusta y sólida para regresión y clasificación en dominios complejos, por Vladimir Vapnik y su equipo en los laboratorios AT&T. Su proceso de aprendizaje es supervisado, es decir, del ámbito predictivo. En clasificación supervisada los casos pertenecientes al conjunto de datos tienen asignada una clase o etiqueta a priori, siendo el objetivo encontrar patrones o tendencias de los casos pertenecientes a una misma clase. Además su teoría está basada en la idea de minimización del riesgo estructural (SRM) [3].

Las Maquinas de Soporte Vectorial se diferencian de otras técnicas como redes neuronales y programación genética, ya que las SVM no son afectadas por el problema de los mínimos locales, debido a que su entrenamiento se basa en problemas de optimización convexa.

La clasificación mediante SVM permite obtener clasificadores lineales y no lineales. En los siguientes puntos se analiza el clasificador lineal para datos separables y no separables, y luego el clasificador no lineal, en donde se introduce el concepto de *Kernel*. Además cabe destacar que en los siguientes subcapítulos solo se presentará los casos para clasificación binaria, ya que es éste el tipo de clasificación a utilizar en este trabajo de investigación.

3.1.2 Descripción del Modelo

Básicamente la técnica de Máquinas de Soporte Vectorial para clasificación se basa en encontrar un hiperplano que divida y maximice el margen entre las clases en un espacio, como se aprecia en la Figura 3.1.

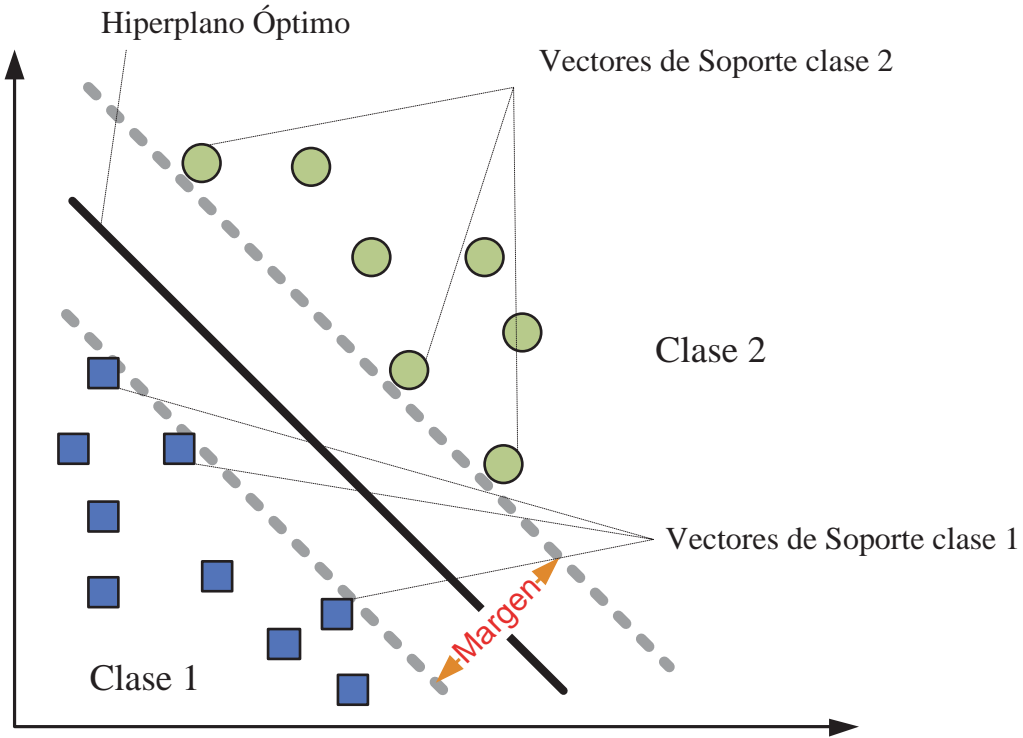


Figura 3.1 Hiperplano de separación óptima

Por un lado, los hiperplanos que están más alejados de las fronteras de las clases de objetos corresponden a mayores márgenes de separación. Por otro, hiperplanos que aciertan más en la asignación de objetos a las clases a las que efectivamente pertenecen, tienen un menor error de clasificación. Por lo tanto, un hiperplano de separación ideal debe maximizar el margen de separación y minimizar el error de clasificación. Sin embargo, no siempre es posible cumplir los dos objetivos simultáneamente. Para salvar esta dificultad se plantea un problema de optimización cuya función objetivo combina ambos propósitos.

Este problema de optimización resulta ser un problema de minimización cuadrático convexo. En el caso que el número de objetos a clasificar es mayor que el número de atributos o características de cada objeto, lo que usualmente no sucede, se dice que este problema tiene una única solución óptima y que las clases son linealmente separables. Sin embargo, el estudio de los SVM se ha extendido al caso de clases que no son linealmente separables mediante la introducción de variables de pérdida u holgura. Más aún, cuando no es posible encontrar un hiperplano que separe los elementos en el espacio de solución de entrada, se introduce el concepto del *Kernel*, el cual trabaja en un espacio (espacio de características) de mayor dimensión, en donde si es posible encontrar un hiperplano o una superficie plana que logre separar los elementos en dos clases. A este caso se le llaman clasificadores no lineales.

3.1.3 Funciones de decisión

Considere el problema de clasificación de un punto cuyas características están dadas por el vector x tal que $x = (x_1, \dots, x_p)^T$ pertenece a una de dos clases posibles. Suponiendo que se tienen las funciones $f_1(x)$ y $f_2(x)$ que definen las clases 1 y 2 (clasificación binaria) y clasificando al punto x dentro de la clase 1 si $f_1(x) > 0, f_2(x) < 0$, o clasificando al punto x dentro de la clase 2 si $f_1(x) < 0, f_2(x) > 0$, se les llaman funciones de decisión.

Si la función de decisión $f_1(x)$ o la función $f_2(x)$ que se mueven hacia la línea punteada de su propio lado como se aprecia en la Figura 3.2, el conjunto de datos de entrenamiento aún siguen siendo correctamente clasificado, dando la certeza de que es posible encontrar un conjunto infinito de hiperplanos que clasifiquen correctamente los datos de entrenamiento. Sin embargo, es claro que la precisión de clasificación será directamente afectada por la posición de las funciones de decisión.

SVM a diferencia de otros métodos de clasificación consideran esta desventaja y encuentra la función de decisión de tal forma que la distancia entre los datos de entrenamiento es maximizada. Esta función de decisión es llamada función de decisión óptima o hiperplano de decisión óptima.

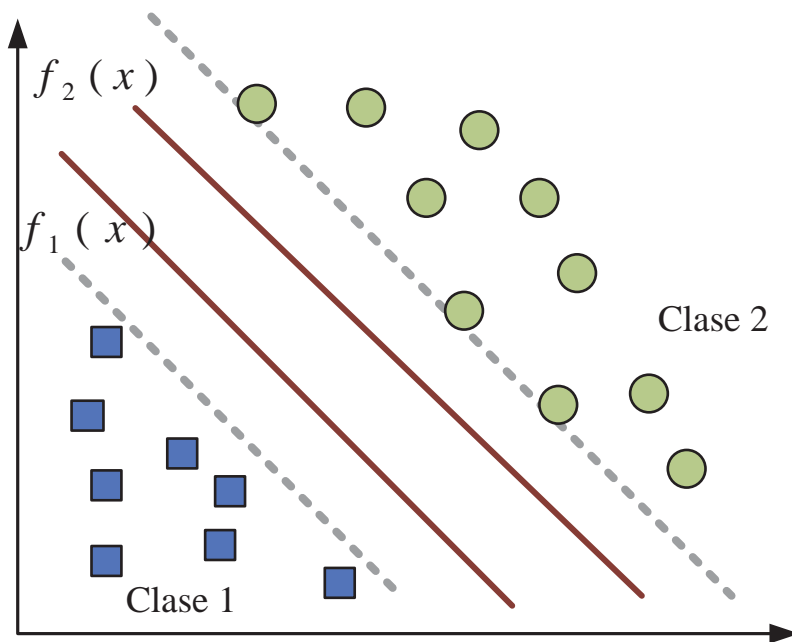


Figura 3.2 Funciones de decisión

3.1.4 Clasificación SVM, Caso Linealmente Separable

Considere el problema de separar el conjunto de vectores de entrenamiento $(x_1, y_1), \dots, (x_l, y_l) \in R^n$ pertenecientes a dos clases separadas $y_i = \{+1, -1\}$.

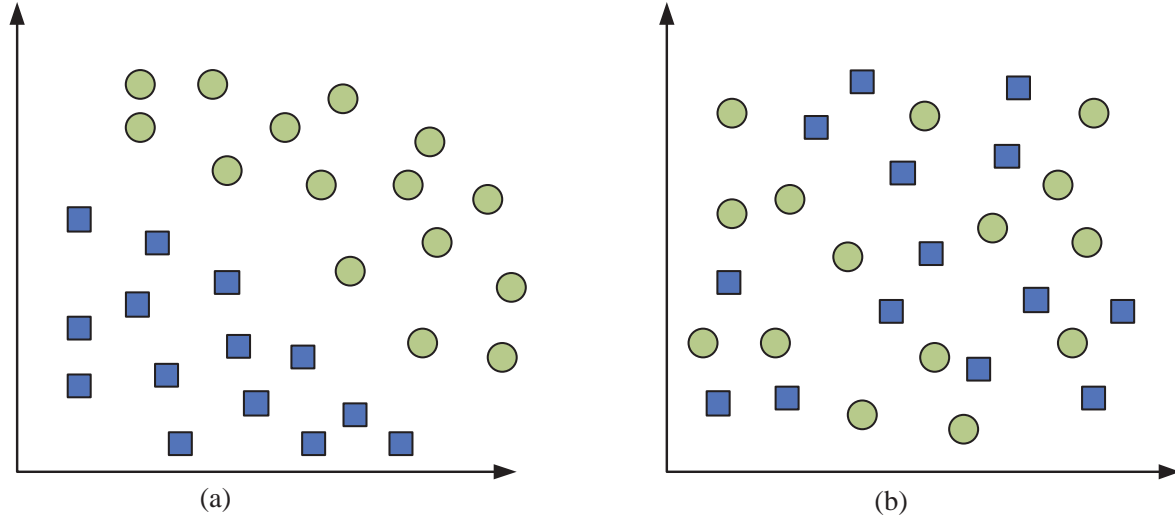


Figura 3.3 (a) Caso Linealmente Separable (b) Caso Linealmente no Separable

En este problema el objetivo es separar los vectores de entrenamiento en dos clases mediante un hiperplano:

$$(w \cdot x) + b = 0, w \in R^n, b \in R \quad (3.1.1)$$

Donde w y b son parámetros que se inducen a partir de la función de decisión:

$$f(x) = \text{signo}(w \cdot x + b) \quad (3.1.2)$$

Existen muchos posibles clasificadores lineales que pueden separar los datos, sin embargo, hay uno solo que maximiza el margen de separación. El hiperplano $(w \cdot x_i) + b = 0$ satisface las siguientes condiciones:

$$\begin{aligned} (w \cdot x_i) + b &> 0, y_i = +1 \\ (w \cdot x_i) + b &< 0, y_i = -1 \end{aligned} \quad (3.1.3)$$

Combinando las dos expresiones en (3.1.3) y escalando w y b , con un factor apropiado, una superficie de decisión equivalente se puede formular como aquella que satisface la siguiente restricción:

$$y_i [(w \cdot x_i) + b] \geq 1, i = 1, \dots, l \quad (3.1.4)$$

Además, se puede demostrar que el hiperplano que separa óptimamente los datos en dos clases es aquel que minimiza el funcional:

$$\tau(w) = \frac{1}{2} \|w\|^2 \quad (3.1.5)$$

Por lo tanto, el problema de optimización cuadrático puede ser reformulado como un problema de optimización no restringida, usando multiplicadores de Lagrange y su solución estaría dada por la identificación de los puntos de silla del funcional de Lagrange:

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^l \alpha_i \{y_i [(w \cdot x_i) + b] - 1\} \quad (3.1.6)$$

Donde α_i son los multiplicadores de Lagrange. El Lagrangiano tiene que ser minimizado respecto a w y b , es decir:

$$\frac{\partial L}{\partial b} = 0 \rightarrow w = \sum_{i=1}^l \alpha_i y_i = 0 \quad (3.1.7)$$

$$\frac{\partial L}{\partial w} = 0 \rightarrow w - \sum_{i=1}^l \alpha_i y_i x_i = 0 \rightarrow w = \sum_{i=1}^l \alpha_i y_i x_i = 0 \quad (3.1.8)$$

Reemplazando (3.1.7) y (3.1.8) en (3.1.6) se obtiene el problema de optimización dual:

$$W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (3.1.9)$$

Debiendo ser maximizado sujeto a las restricciones $\alpha_i \geq 0$.

Encontrar la solución de la ecuación (3.1.9) requerirá la aplicación de técnicas de optimización de programación cuadrática (QP) y métodos numéricos. Es aquí donde se encuentra una de las desventajas de esta técnica ya que el costo de procesamiento es muy alto para los datos de entrenamiento.

Una vez hallada la solución en la forma de un vector $\alpha^0 = (\alpha_1^0, \alpha_2^0, \dots, \alpha_l^0)$, el hiperplano de separación óptima está dado por:

$$\begin{aligned} w_0 &= \sum y_i \alpha_i^0 x_i, \\ b_0 &= -\frac{1}{2} w_0 \cdot [x_r + x_s] \end{aligned} \quad (3.1.10)$$

Donde x_r, x_s es cualquier Vector de Soporte de alguna de las dos clases. Por lo tanto el clasificador puede ser construido como:

$$f(x) = \text{signo}(w_0 \cdot x + b_0) = \text{signo}\left(\sum_{i=1}^l y_i \alpha_i^0 (x_i \cdot x) + b_0\right) \quad (3.1.11)$$

Solamente los puntos x_i que tienen multiplicadores de Lagrange α_i^0 diferentes de cero, son llamados Vectores de Soporte. Nótese que si los datos son linealmente separables, todos los vectores de soporte estarán sobre el margen, por lo tanto, el número de Vectores de Soporte puede ser muy pequeño.

3.1.5 Clasificación SVM, Caso Linealmente No Separable

En la práctica no es habitual trabajar con conjuntos de datos separables. Existen casos en donde se encuentran vectores de una clase dentro de la región correspondiente a los vectores de otra clase y los cuales nunca podrán ser separados por medio de hiperplanos, como se muestra en la Figura 3.3(b). En estas situaciones se dirá que el conjunto es linealmente no separable.

Ante estos casos, el problema de optimización (3.1.5) sujeto a la restricción (3.1.4), no encuentra una solución posible. Sin embargo, no es difícil ampliar las ideas generales del caso separable al caso no separable. Para ello, se introduce una variable ξ de holgura en las restricciones y se plantea un nuevo conjunto de restricciones:

$$\begin{aligned} (w \cdot x_i) + b &\geq +1 - \xi_i, y_i = +1 \\ (w \cdot x_i) + b &\leq -1 + \xi_i, y_i = -1 \end{aligned} \quad (3.1.12)$$

Se tiene ahora que para que se produzca un error en la clasificación de un vector de entrenamiento (una entrada no es ubicada en la clase correcta) es necesario que el valor correspondiente a ξ_i sea superior a la unidad. Así, si en el vector x_i se comete un error entonces $\xi_i \geq 1$ y por tanto $\sum \xi_i$ es una cota superior del número de errores que se cometen dentro del conjunto de entrenamiento.

Ya que en el caso no separable, necesariamente se han de cometer errores, se debe asignar a la función objetivo un coste extra que en cierto modo penalice los errores o el ruido. En consecuencia se introduce la constante C , resultando la siguiente ecuación:

$$\tau(w) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \quad (3.1.13)$$

La constante C es un parámetro que controla el *trade-off* entre la maximización del margen y la minimización del error de entrenamiento. Así, cuando C es pequeño, se permite equivocarse (clasificar mal) muchas veces, pero a cambio se obtiene un margen grande. Por otro lado, cuando C es grande, no se permite equivocarse (se pena altamente por cada error) y se obtiene un margen pequeño. Cabe destacar que este es el primer parámetro que debe ser ajustado.

Por lo tanto, el problema de optimización que se plantea es:

$$\min_{w \in R^n} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \quad (3.1.14)$$

Sujeto a:

$$y_i [(w \cdot x_i) + b] - 1 + \xi_i \geq 0, \forall i = 1, \dots, l; \xi_i \geq 0 \forall i$$

De nuevo se quiere maximizar la función representada en la ecuación (3.1.9) sujeto a:

$$0 \leq \alpha_i \leq C \quad \forall i = 1, \dots, n;$$

$$\sum_{i=1}^l \alpha_i y_i = 0$$

Y cuya solución final viene dada por:

$$w = \sum_{i=1}^{NSV} \alpha_i y_i s_i \quad (3.1.15)$$

Donde NSV denota el número de vectores soporte y por s_i los vectores soporte del conjunto $\{x_1, \dots, x_n\}$. Claramente $NSV \leq l$ y una de las características más interesante de estos modelos es que eligiendo adecuadamente los parámetros es posible conseguir que NSV sea muy inferior a l con lo que se consigue una representación corta de la solución en función de los vectores de entrada sin perder capacidad de generalización.

3.1.6 Clasificación SVM, Caso No lineal

En una SVM, el hiperplano óptimo es determinado para maximizar su habilidad de generalización. Pero, si los datos de entrenamiento no son linealmente separables, es decir, no se pueden separar las clases dentro del espacio de solución original, el clasificador obtenido puede no tener una alta habilidad de generalización, aun cuando los hiperplanos sean determinados óptimamente. Por este motivo, para poder maximizar el espacio entre clases (hiperplano óptimo), el espacio de entrada original es transformado dentro de un espacio de mayor dimensión llamado “espacio de características”, como se muestra en la Figura 3.4. La técnica de SVM puede crear una hipersuperficie de decisión no lineal, capaz de clasificar datos separables no linealmente. Generalmente, para patrones de entrada n-dimensionales, en lugar de una curva no lineal, SVM creará una hipersuperficie de separación no lineal.

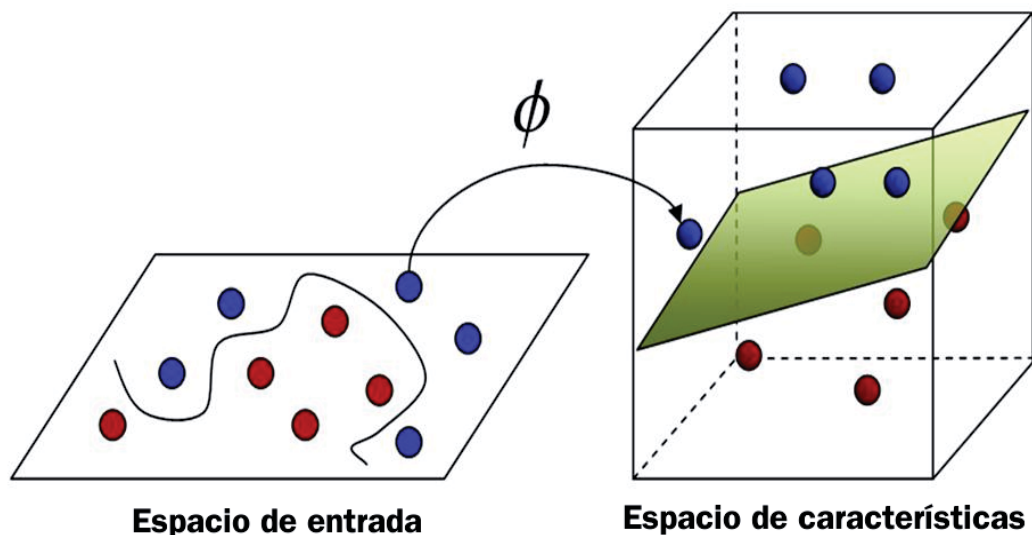


Figura 3.4 Caso no Lineal

En este caso, el problema de optimización de la ecuación (3.1.9) se convierten en:

$$W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j k(x_i, x_j) \quad (3.1.16)$$

Donde $k(x_i, x_j)$ es la función *kernel* que realiza el mapeo no lineal en el espacio de características, manteniéndose con las mismas restricciones. Por su parte, el clasificador que implementa el hiperplano de separación óptima en el espacio de característica está dado por:

$$f(x) = \text{signo} \left(\sum_{i=1}^l y_i \alpha_i k(x_i \cdot x) + b \right) \quad (3.1.17)$$

Consecuentemente, todo lo que se ha derivado para el caso lineal es también aplicable para un caso no lineal usando un *kernel*. Además, usando diferentes funciones *kernel*, el algoritmo de SVM puede construir una variedad de máquinas de aprendizaje. Algunos tipos de *kernels* se aprecian a continuación:

- Kernel Lineal

$$\varphi(x, x_k) = x_k^T x$$

- Kernel Polinomial

$$\varphi(x, x_k) = (x_k^T x + 1)^d$$

- Kernel Función de Base Radial

$$\varphi(x, x_k) = \exp \left\{ \frac{-\|x - x_k\|^2}{\sigma^2} \right\}$$

- Kernel Sigmoidal

$$\varphi(x, x_k) = \tan \left[s x_k^T x + \theta \right]$$

Por lo tanto, dependiendo del tipo de *kernel* a utilizar, se deben ajustar los parámetros asociados a cada *kernel*, por ejemplo:

- d, T : K. Polinomial
- ρ : K. Función de Base Radial
- s, θ : K. Perceptrón de dos capas o Sigmoidal.

Nótese que para el kernel Lineal no posee parámetros a ajustar, en este caso solo se debe optimizar la variable de penalización de errores C .

En resumen, para aplicar la técnica de SVM se deben ajustar los parámetros asociados al *kernel* y el parámetro de penalización de errores (C). He aquí unas de las desventajas a resolver en este trabajo, ajustar los parámetros de la mejor forma con el algoritmo de Optimización por Enjambre de Partículas, el cual se verá en los capítulos siguientes.

3.1.7 Características claves de SVM

Según [4] y [3], la técnica de SVM posee varias ventajas sobre otros algoritmos de minería de datos, como las redes neuronales. Estas se detallan a continuación:

- Combina aspectos geométricos fáciles de visualizar y elementos teóricos de importante fundamento, como es Dualidad, funciones *Kernel* y el Riesgo Estructural.
- Permite la construcción de separadores no lineales mediante funciones de *Kernel*.
- La función discriminante depende sólo de una cantidad reducida de objetos (los Vectores de Soporte) que están más cerca de la superficie clasificadora, a diferencia de métodos estadísticos como el Análisis Discriminante, en el cual valores extremos determinan el hiperplano construido.
- Tiene un desempeño computacional más eficiente que otros algoritmos de Inteligencia Artificial (Redes Neuronales por ejemplo).
- Minimiza el problema de sobreajuste (*overfitting*), debido a que se preocupa de no obtener modelos muy ajustados a los datos, mediante la maximización del margen entre los hiperplanos paralelos al hiperplano separador. Para esto, minimiza la norma del vector normal al hiperplano separador.

3.1.8 Algoritmos para SVM

Una de las principales desventajas que presenta esta técnica, corresponde al alto costo de procesamiento que se produce al utilizar programación cuadrática para resolver la ecuación (3.1.16). En base a esto, algunos algoritmos han sido propuestos para salvar esta dificultad. A continuación se muestran de manera general algunos de los algoritmos más utilizados actualmente.

3.1.8.1 SMO

El algoritmo SMO (Optimización Mínima Secuencial o *Sequential Minimal Optimization*) fue propuesto por el científico John Platt. Nace a partir de la idea del método de descomposición a su extremo, al optimizar un subconjunto mínimo de únicamente dos puntos en cada iteración. Básicamente el fuerte de esta técnica reside en el hecho de que el problema de optimización para dos puntos admite una solución analítica, eliminando la necesidad de usar un optimizador de programación cuadrática.

El requisito de que la condición $\sum_{i=1}^l \alpha_i y_i = 0$, obliga en todo momento a que el número de multiplicadores que puede ser optimizado en cada paso sea dos. Cada vez que un multiplicador es actualizado, por lo menos otro multiplicador necesita ser ajustado con el propósito de mantener la condición verdadera. En cada paso, SMO elige dos elementos α_i y α_j para optimizarlos y encuentra el valor óptimo de esos dos parámetros, dado que los demás se encuentran fijos. Además actualiza el vector α . Por otra parte, la elección de los dos puntos es determinada por una heurística y la optimización de los dos multiplicadores es realizada analíticamente.

El rendimiento de SMO es muy bueno cuando se está en el caso de SVM no lineal, ya que el tiempo de computación del *kernel* puede ser reducido. Además, a pesar de que este

algoritmo necesita más iteraciones para converger, cada iteración usa pocas operaciones, por lo tanto converge muy rápido. Otra característica del algoritmo radica en que este no necesita almacenar la matriz del *kernel* en la memoria, ya que no se involucran operaciones matriciales.

3.1.8.2 SSVM

El algoritmo *Simple Support Vector Machines* fue propuesto por Vishwanathan y Murty. El algoritmo inicializa con un pequeño conjunto de datos llamado conjunto candidato de vectores soporte. Este conjunto es obtenido mediante un algoritmo que detecta y obtiene los datos más cercanos entre clases opuestas. Una vez que el algoritmo encuentra un punto que viola las condiciones en el conjunto de datos, el algoritmo lo aparta y lo adiciona al conjunto candidato de vectores soporte. Luego los datos que se encuentran ya presentes en el conjunto candidato son eliminados. Este procedimiento es repetido en todo el conjunto de datos hasta que todos los puntos hayan sido evaluados y ningún otro punto que viole las condiciones haya sido encontrado.

3.1.8.3 LIBSVM

El algoritmo LIBSVM está basado en el algoritmo SMO, sin embargo, posee un algoritmo de selección del conjunto de trabajo mucho más avanzado. LIBSVM emplea un algoritmo de dirección de búsqueda que maximiza el incremento la función objetivo en cada iteración.

3.1.8.4 LS-SVM

Este algoritmo es el utilizado en este trabajo de investigación. Su fuerte radica en que la resolución de la ecuación (3.1.16) se realiza a través de un conjunto de ecuaciones lineales. De esta forma reduce considerablemente el tiempo de procesamiento, pudiendo tratar una cantidad más considerable de datos de entrenamiento. En el subcapítulo 3.2 se muestra con más detalle.

3.2 Mínimos Cuadrados SVM (LS-SVM)

La técnica de los mínimos cuadrados SVM o *Least Squares Support Vector Machine* (LS-SVM) fue propuesta y creada por Suykens y Vandewalle [5] en 1999. Esta técnica es una reformulación de las SVM y soluciona el problema nombrado anteriormente a través de un conjunto de ecuaciones lineales. De esta forma, este método puede tratar una cantidad más considerable de datos de entrenamiento y a la vez el costo de procesamiento de los datos de entrenamiento disminuye considerablemente. Una característica particular de SVM es el hecho de que la mayoría de los multiplicadores de Lagrange, asociados a los vectores de entrenamiento, son nulos. Esto no ocurre en LS-SVM, los valores se distribuyen, sin predominancia de valores nulos [3].

3.2.1 Definición Formal

En el método LS-SVM se plantea el siguiente problema de minimización:

$$\min_{w,b,\ell} J(w, b, \ell) = \frac{1}{2} w^T w + C \frac{1}{2} \sum_{k=1}^N \ell_k^2 \quad (3.2.1)$$

Sujeto a la restricción:

$$y_k [w^T \varphi(x_k + b)] = 1 - \ell_k, k = 1, \dots, N \quad (3.2.2)$$

Donde C es el factor de regularización o penalización de errores y ℓ_k es la diferencia entre el output y_k deseado con el output actual. El Lagrangiano para el problema es:

$$L(w, b, \ell; \alpha) = J(w, b, \ell) - \sum_{k=1}^N \alpha_k \{y_k [w^T \varphi(x_k) + b] - 1 + \ell_k\} \quad (3.2.3)$$

Donde α_k son los multiplicadores de Lagrange (pueden ser positivos o negativos debido a las restricciones de igualdad como se derivan de las condiciones de *Kuhn-Tucker*).

Calculando las derivadas parciales del Lagrangiano, con respecto de todas las variables e igualando a cero se obtienen las condiciones de optimización, esto debido a que la función es convexa:

$$\begin{aligned} \frac{\partial L}{\partial w} = 0 &\rightarrow w = \sum_{k=1}^N \alpha_k y_k \varphi(x_k) \\ \frac{\partial L}{\partial b} = 0 &\rightarrow \sum_{k=1}^N \alpha_k y_k = 0, \\ \frac{\partial L}{\partial \ell_k} = 0 &\rightarrow \alpha_k = C \ell_k, k = 1, \dots, N, \\ \frac{\partial L}{\partial \alpha_k} = 0 &\rightarrow y_k [w^T \varphi(x_k) + b] - 1 + \ell_k = 0, k = 1, \dots, N \end{aligned} \quad (3.2.4)$$

Se puede escribir inmediatamente la solución como el siguiente conjunto de ecuaciones lineales:

$$\begin{pmatrix} I & 0 & 0 & -Z^T \\ 0 & 0 & 0 & -Y^T \\ 0 & 0 & C^I & -I \\ Z & Y & I & 0 \end{pmatrix} \begin{pmatrix} w \\ b \\ \ell \\ \alpha \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vec{1} \end{pmatrix} \quad (3.2.5)$$

Dónde $Z = [\varphi(x_1)^T y_1; \dots; \varphi(x_N)^T y_N]$, $Y = [y_1; \dots; y_N]$, $\vec{1} = [1; \dots; 1]$, $\ell = [\ell_1; \dots; \ell_N]$, $\alpha = [\alpha_1; \dots; \alpha_N]$.

La solución también está dada por:

$$\begin{pmatrix} 0 & -Y^T \\ Y & ZZ^T + C^{-II} \end{pmatrix} \begin{pmatrix} b \\ \alpha \end{pmatrix} = \begin{pmatrix} 0 \\ \vec{1} \end{pmatrix} \quad (3.2.6)$$

La condición de Mercer puede ser aplicada a la matriz $\Omega = ZZ^T$, donde:

$$\begin{aligned} \Omega_{kl} &= y_k y_l \varphi(x_k)^T \varphi(x_l) \\ &= y_k y_l \psi(x_k, x_l) \end{aligned} \quad (3.2.7)$$

Por lo tanto, el clasificador se encuentra resolviendo las ecuaciones (3.2.6) y (3.2.7) en lugar de programación cuadrática. Los valores soporte α_k son proporcionales a los errores en los puntos de datos (3.2.4), mientras que en el caso de SVM los valores son iguales a cero.

Hasta aquí gracias a LS-SVM se ha salvado la desventaja de la resolución de un problema de optimización de programación cuadrática, gracias a la resolución del problema mediante ecuaciones lineales, de esta forma disminuir considerablemente el costo de procesamiento y aumentar tamaño de datos a entrenar.

3.3 Optimización por enjambres de partículas

La optimización por enjambre de partículas, más conocida como PSO del inglés *Particle Swarm Optimization*, es una metaheurística evolutiva y de búsqueda desarrollado por Kennedy y Eberhart [6], inspirada en el comportamiento social de algunas comunidades de organismos presentes en la naturaleza, tales como cardúmenes de peces, bandadas de aves, enjambre de insectos, entre otros (ver Figura 3.5). Este comportamiento social se basa en la transmisión del suceso de cada individuo a los demás individuos del grupo, lo cual resulta en un proceso sinérgico que permite a los individuos satisfacer de la mejor manera posible sus necesidades más inmediatas, tales como la localización de alimentos o de algún lugar de cobijo.



Figura 3.5 Swarm en la Naturaleza

El enjambre que se pone de ejemplo para explicar este método es uno de abejas. Las abejas a la hora de buscar polen buscan la región del espacio en la que existe más densidad de flores, ya que es ahí donde existe más polen. Una sola abeja vuela de modo errático sobre el espacio, lo único que recuerda una abeja es la región donde ha visto más densidad de flores. A su vez, el enjambre de abejas sabe de algún modo la región del espacio donde el enjambre ha encontrado más densidad de flores. Cada abeja ante la duda variará individualmente su movimiento como una composición de estas dos direcciones volando hacia un lugar intermedio. Es de esperar que la abeja en ese sobrevuelo encuentre una región con más densidad de flores de la que ha visto hasta ese momento o incluso que la que ha visto el enjambre, en tal caso la dirección de exploración del enjambre variará explorando esa nueva

zona. Pasado el tiempo, si se encuentra una región con mayor densidad el enjambre explorará esa región.

La optimización por enjambre de partículas llega a una solución próxima a la global, en lugar de converger hacia soluciones locales como puede suceder con otros métodos tradicionales. Otros aspectos, tales como el hecho de realizar la búsqueda utilizando un conjunto de puntos sobre el espacio de soluciones en lugar de un único punto, el utilizar una función objetivo para dirigir la búsqueda en lugar de derivadas, o que los nuevos puntos a explorar se determinen de acuerdo a reglas de decisión estocásticas en lugar de deterministas, reafirman la superioridad de PSO sobre técnicas de optimización local.

Muchos estudios han optado por utilizar PSO, debido a características tan sencillas que presenta el modelo, estas son:

1. Pocos parámetros a ajustar.
2. Normalmente trabaja con poblaciones pequeñas.
3. Una convergencia más rápida.

3.3.1 Terminología y fundamentos de PSO

PSO sigue una analogía de partículas en un enjambre, en base a esto se definen algunos términos claves para su entendimiento, estos son:

- **Partícula o agente:** cada individuo en el enjambre es representado como una partícula o agente. Todas las partículas en el enjambre actúan individualmente bajo el mismo principio: acelerar hacia la mejor posición personal y global mientras que constantemente verifican el valor de su posición y velocidad
- **Posición:** es representada por coordenadas en el plano $x-y$, pero en general, puede extender esta idea a cualquier espacio D-dimensional de acuerdo al problema a tratar.
- **Velocidad:** refleja el cambio de dirección que sufre la partícula dentro del espacio de búsqueda.
- **Función *fitness* o función desempeño:** como en todas las técnicas de cómputo evolutivo, debe existir alguna función o método para evaluar la calidad de la solución. Por esta razón, el *fitness* en el caso de PSO debe tomar la posición en el espacio de solución y retornar un número que representa el valor de esa posición. El *fitness* provee la interfaz entre el problema físico y el algoritmo de optimización.
- **Mejor personal (*pbest*):** es la posición con el mejor valor de *fitness* encontrado por la partícula hasta el momento.
- **Mejor global (*gbest*):** es la posición con el mejor valor de *fitness* encontrado del enjambre total. Este valor está presente si se utiliza un modelo global, es decir, un modelo en el que los individuos son influenciados por el mejor de toda la población.
- **Mejor global (*lbest*):** es la posición con el mejor valor de *fitness* encontrado en el vecindario de una partícula. Este valor está presente si se utiliza un modelo local, es decir, un modelo en el que los individuos son influenciados por el mejor de un grupo

pequeño de individuos (vecindario). La determinación del vecindario depende de la forma en la que se efectúa el agrupamiento de individuos de la población.

Básicamente, PSO consiste en un algoritmo iterativo basado en una población de individuos denominada enjambre, en la que cada individuo, llamado agente o partícula, que representa una posible solución dentro del espacio de soluciones multidimensional, se dice que “sobrevuela” junto con el resto de sus congéneres el espacio de decisión en busca de una solución óptima, ajustando su posición de acuerdo a su propia experiencia y la de sus vecinos.

PSO está sustentado en lo que se conoce como “inteligencia de grupo”, más comúnmente referido en la literatura como *swarm intelligence* [7], fundamentada en el comportamiento colectivo de sistemas auto-organizados y descentralizados. Dentro de la inteligencia de grupo se deben respetar cinco principios básicos: proximidad, calidad, diversidad de respuesta, estabilidad y adaptabilidad.

- **Proximidad:** la población debiera ser capaz de realizar cálculos sencillos de espacio y tiempo, lo cual se traduce en PSO a movimientos en N dimensiones llevados a cabo durante una serie de intervalos de tiempo que coinciden con movimientos de la población a una determinada velocidad
- **Calidad:** promueve la capacidad de la población para responder a factores de calidad en el espacio de soluciones, lo que se consigue en PSO con la memoria de la partícula y con la historia o conocimiento social que comparten entre sí todos los congéneres.
- **Diversidad de respuesta:** representadas por las tendencias marcadas por la memoria personal de cada partícula y por la historia de la mejor posición visitada por el conjunto.
- **Estabilidad y adaptabilidad:** resaltan aspectos contrapuestos, por un lado la población sólo cambia su comportamiento como grupo cuando se actualiza la mejor posición históricamente visitada por alguno de los miembros que lo integran, respetando así el principio de estabilidad. Además, la población debe de ser a su vez adaptativa, es decir, debe ser capaz de modificar su comportamiento y movimiento cuando hay alguna señal que así lo recomienda desde el punto de vista de ahorro computacional o de mejora en la precisión. Esta premisa se consigue fácilmente en PSO, dado que la población en su conjunto cambia su rumbo cuando alguna de las partículas alcanza una solución próxima a la global que mejora el resultado.

3.3.2 Definición Formal

Una partícula se define como un vector formado por la posición, velocidad y memoria. Sea $i = 1, 2, \dots, n$ la i -ésima partícula y $d = 1, 2, \dots, D$ las dimensiones (número de variables). Dado un espacio de búsqueda D -dimensional, cada partícula i del enjambre conoce su posición actual $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$, la velocidad $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$ con la cual ha llegado a dicha posición y la mejor posición $P_i = (p_{i1}, p_{i2}, \dots, p_{id})$ en la que se ha encontrado, denominada mejor personal (*pbest*). Además todas las partículas conocen la mejor posición encontrada dentro del enjambre, a esta se le llama mejor global (*gbest*).

En PSO convencional la velocidad y posición se actualizan de acuerdo a las siguientes ecuaciones:

$$v_{id}(t+1) = v_{id}(t) + c_1 \cdot r_1 \cdot (p_{id}(t) - x_{id}(t)) + c_2 \cdot r_2 \cdot (p_{gd}(t) - x_{id}(t)) \quad (3.3.1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (3.3.2)$$

Donde c_1 es el parámetro cognitivo que indica la influencia máxima de la mejor experiencia individual de la partícula y c_2 es el parámetro social que indica la influencia máxima de la información social. Los parámetros r_1 y r_2 son dos números aleatorios que varían entre $[0,1]$, cuyo objetivo es emular el comportamiento estocástico y un tanto impredecible que exhibe la población del enjambre.

El movimiento de las partículas sobre el espacio de soluciones y, en consecuencia, el rendimiento del algoritmo, está condicionado por el grado de influencia de las tres componentes de la velocidad mostradas en la ecuación (3.3.1). Estas componentes corresponden a:

- **Momento, hábito o inercia** considera la tendencia de la partícula.
- **Memoria, factor de cognición o auto aprendizaje** incluye la experiencia de la propia partícula.
- **Cooperación, conocimiento social, conocimiento de grupo o información compartida**, refleja el intercambio de información y el comportamiento social como grupo.

Para acotar la velocidad de la partícula se especifica un valor máximo v_{max} , que restringe la velocidad en cada dimensión al intervalo $[-v_{max}, v_{max}]$. Si el valor v_{max} es demasiado grande, las partículas pueden sobrepasar e ignorar continuamente la zona con la solución global. Por el contrario, si v_{max} toma valores extremadamente pequeños las partículas explorarán el espacio de soluciones muy lentamente y podrán quedar atrapadas alrededor de soluciones locales, incapaces de librarse de la base de atracción. Con el objetivo de reducir el efecto de v_{max} y perfeccionar el control del alcance de la búsqueda sobre el espacio de soluciones, surgen las versiones mejoradas del algoritmo.

3.3.3 Algoritmo

A continuación se describen cada uno de los pasos que conlleva este algoritmo.

- Paso 1: Inicializar una población de partículas de tamaño n con posiciones y velocidades aleatorias en un espacio D -dimensional.
- Paso 2: Para cada partícula, evaluar la función de costo o *fitness* en base a algún criterio de optimización.
- Paso 3: Comparar el *fitness* actual de cada partícula con el *fitness* de su mejor posición personal, *pbest*. Si el valor actual es mejor que el de *pbest*, entonces

asignarle a p_{best} al valor actual, y $P_i = X_i$, es decir, la localización actual pasa a ser la mejor hasta el momento.

- Paso 4: Identificar la partícula en la enjambre o grupo (puede ser todo el conjunto de partículas, o un grupo de ellos) con la mejor tasa de éxito hasta el momento, y asignar su índice a la variable g_{best} .
- Paso 5: Ajustar la velocidad y posición de la partícula () de acuerdo a las ecuaciones (3.3.1) y (3.3.2). Se debe verificar que las partículas no sobrepasen la velocidad máxima definida v_{max} .

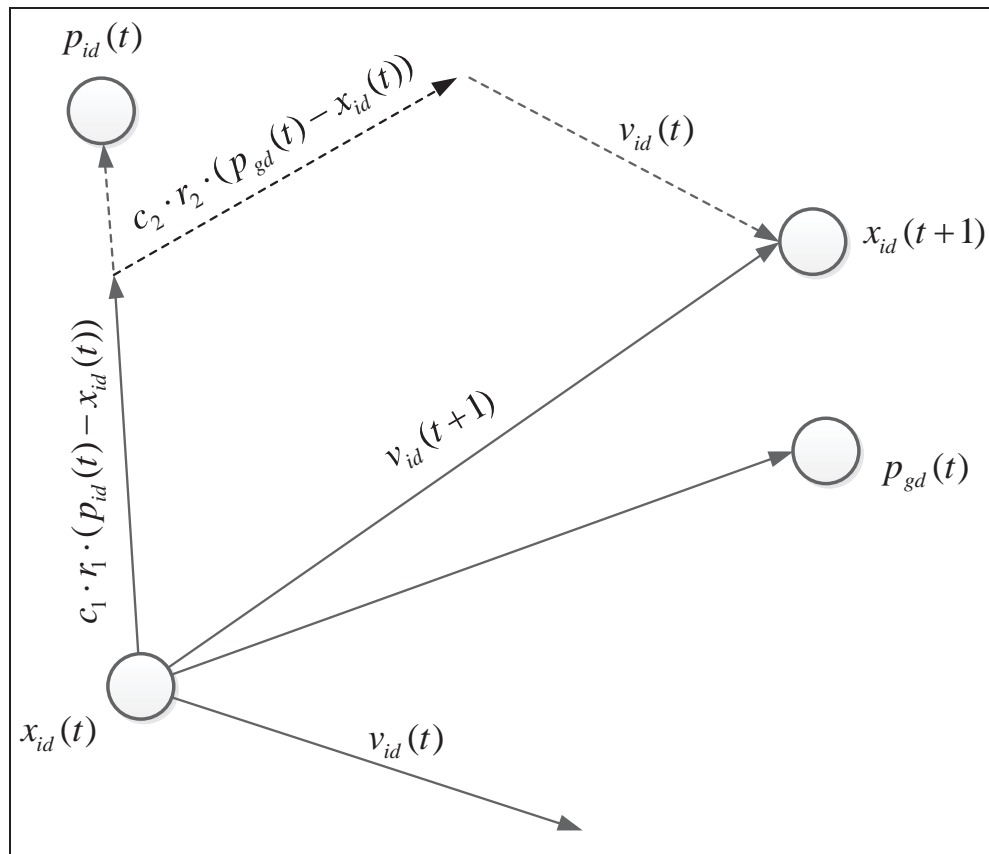


Figura 3.6 Actualización de una Partícula

- Paso 6: Verificar si se cumple el criterio de detención (máximo número de iteraciones, número de iteraciones sin mejoras o valor de $fitness$ alcanzado), si no ir al Paso 2.

Adicionalmente para entender con mayor claridad el algoritmo se presenta el diagrama de flujo asociado a esta metaheurística, representado en la Figura 3.7.

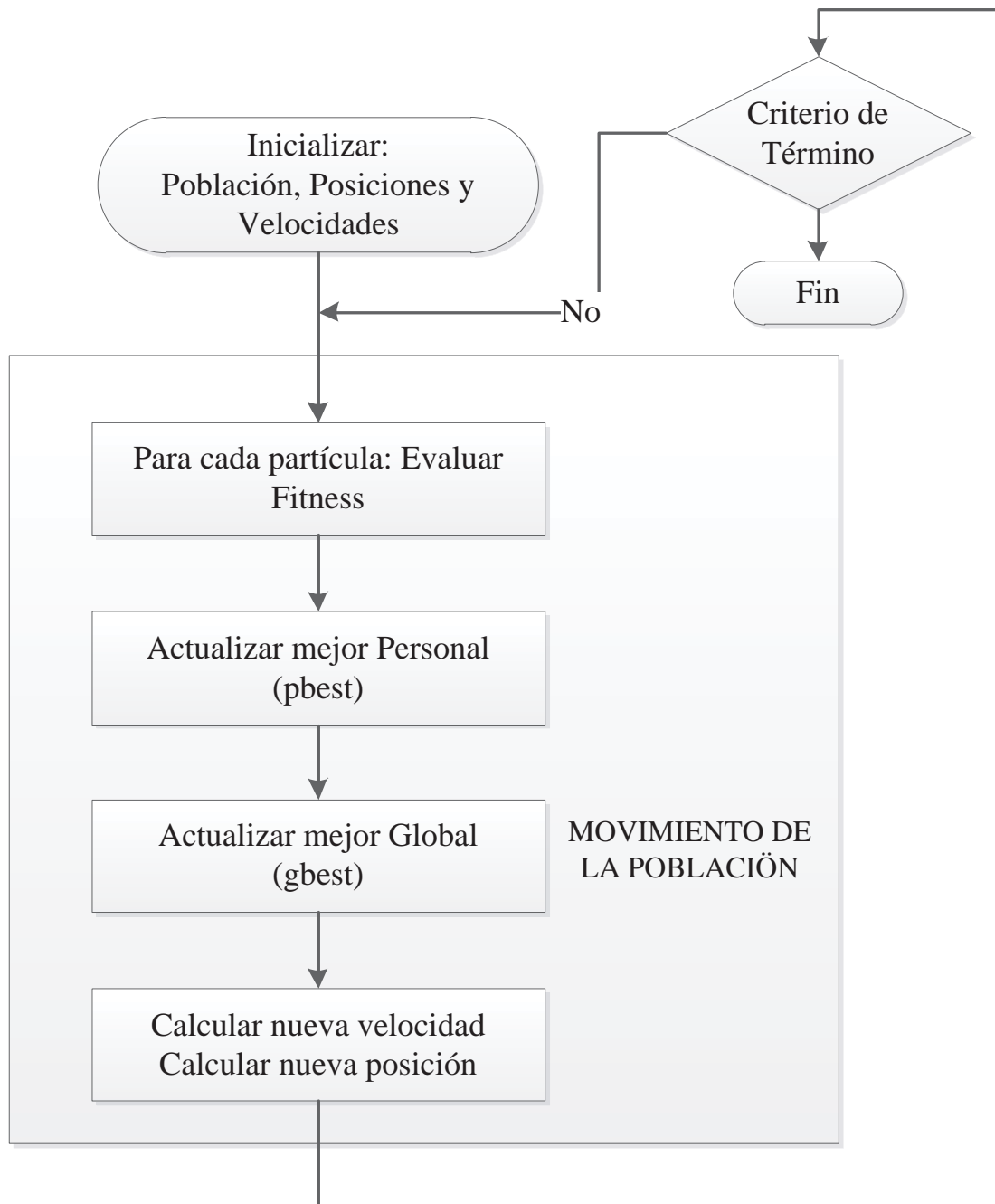


Figura 3.7 Diagrama de Flujo PSO Convencional

3.3.4 Variantes

Tras el auge de la metaheurística PSO, muchos investigadores han realizado modificaciones al algoritmo enfocándose principalmente en la velocidad de convergencia y en la calidad de las soluciones encontradas. A continuación, se presentan algunas de las principales variantes de PSO.

3.3.4.1 PSO con Factor de Inercia

En [8], Shi y Eberhart introdujeron el factor de inercia w en la ecuación que determina la velocidad. Esta modificación surgió como mecanismo de control de las habilidades de exploración y explotación del enjambre y además para que la velocidad no excediera los límites establecidos. Numerosos estudios concluyeron que como mecanismo de control de exploración/explotación funciona correctamente, aunque no logró eliminar completamente la necesidad del chequeo de la velocidad.

El factor w multiplica a la velocidad previa en la fórmula básica de velocidad y, puede permanecer fijo o ser linealmente decrementado en cada ciclo de vuelo. Luego, la nueva velocidad es actualizada de la misma manera que en las otras variantes de PSO. Un factor de inercia con valor de 1 introduce una preferencia para que las partículas continúen moviéndose en la misma dirección. De otro modo, se cambia la dirección.

De esta forma la ecuación de velocidad (3.3.1) queda de la siguiente manera:

$$v_{id}(t+1) = w \cdot v_{id}(t) + c_1 \cdot r_1 \cdot (p_{id}(t) - x_{id}(t)) + c_2 \cdot r_2 \cdot (p_{gd}(t) - x_{id}(t)) \quad (3.3.3)$$

Las primeras versiones de PSO con factor de inercia utilizaban un valor constante durante toda la búsqueda. Estudios posteriores demostraron que para algunos problemas, el decremento del factor de inercia w en cada iteración permite combinar la exploración de una búsqueda global con la explotación de una búsqueda local. El factor inercia w , es multiplicado por la velocidad actual en la ecuación de actualización de la velocidad y se va reduciendo gradualmente a lo largo del tiempo (medido en iteraciones del algoritmo). De esta forma, en cada iteración el factor de inercia es calculado de la siguiente forma:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \cdot iter \quad (3.3.4)$$

Donde w_{\max} es el peso inicial y w_{\min} el peso final, $iter_{\max}$ es el número máximo de iteraciones y $iter$ es la iteración actual. w debe mantenerse entre 0.9 y 1.2. Valores altos provocan una búsqueda exhaustiva (más diversificación) y valores bajos una búsqueda más localizada (más intensificación).

Otra posibilidad es la reinicialización aleatoria de w en cada iteración utilizando, por ejemplo, un número aleatorio generado con una distribución Gaussiana:

$$w \sim N(0.72; \sigma)$$

con σ lo suficientemente pequeño como para asegurar que w sea menor que 1.

Recientemente Bin, Zhigang y Xingsheng en [9] propusieron IPSO (*Improved Particle Swarm Optimization*), el cual plantea un peso de inercia dinámico que es decreciendo de acuerdo a las iteraciones. Si bien es bastante parecido al método nombrado en los párrafos anteriores, este algoritmo calcula el peso de inercia de la siguiente forma:

$$w' = w * \mu^{-k} \quad (3.3.5)$$

Dónde:

$$\begin{aligned} w &\in [0,1] \\ \mu &\in [1.0001, 1.005] \\ k &= \text{num iteración} \end{aligned}$$

3.3.4.2 PSO con adaptación Dinámica DAPSO

Esta variante propuesta en [10], plantea una modificación de PSO con adaptación dinámica. Esta se sustenta en la base de que la velocidad de las partículas tiende a cero rápidamente, lo que causa que las partículas queden atrapadas en óptimos locales. Este fenómeno es llamado “similitud” del enjambre de partículas, el cual restringe el área de búsqueda de los individuos.

Para ampliar el área de búsqueda puede recurrirse a dos opciones incrementar el número de partículas o debilitar la capacidad de las partículas para seguir al valor de la mejor solución del enjambre *gbest*. Sin embargo, la primera opción implica un mayor coste computacional y la segunda implica una lenta convergencia.

Una de las premisas de esta modificación, trata de que la velocidad de las partículas al momento de converger no deben tender a cero. Es por ello que se introduce el factor de inercia w , el cual varía en cada iteración. Los autores plantean que un peso de inercia que decrece linealmente puede no reflejar realmente el proceso actual de búsqueda. Por lo tanto, en DAPSO w es afectado por el estado evolutivo del algoritmo y determinado por dos factores, el factor de velocidad evolutivo y el grado de agregación del enjambre.

El factor de velocidad evolutivo se define de la siguiente forma:

$$h_i^t = \left| \frac{\min(F(pbest_i^{t-1}), F(pbest_i^t))}{\max(F(pbest_i^{t-1}), F(pbest_i^t))} \right| \quad (3.3.6)$$

Donde $F(pbest_i^t)$ es el fitness de $pbest_i^t$. Bajo este supuesto se obtiene que $0 < h \leq 1$. Este parámetro toma en cuenta la historia recorrida de cada partícula. Cuando menor sea el valor de h será más rápida la velocidad.

Por otra parte, el grado de agregación se define de la siguiente forma:

$$s = \left| \frac{\min(F_{ibest}, \overline{F}_i)}{\max(F_{ibest}, \overline{F}_i)} \right| \quad (3.3.7)$$

Donde \bar{F}_t es el promedio del fitness de todas las partículas en la t-ésima iteración. Cabe destacar que $F(gbest_t)$ no puede ser sustituido por F_{tbest} , ya que F_{tbest} representa el valor encontrado en la iteración t-ésima, mientras que $F(gbest_t)$ denota el valor óptimo que todo el enjambre ha encontrado hasta la iteración t-ésima. De esta forma, el factor de inercia puede ser escrito en función de los parámetros h, s .

$$w'_i = f(h'_i, s)$$

El propósito de la variación de w es dar al algoritmo una mayor capacidad para buscar rápidamente y salir de los óptimos locales. Además esta variante dice obtener mejores soluciones que PSO estándar, ya que en el trayecto de búsqueda de una partícula si la posibilidad de encontrar la solución incrementa, el individuo no se precipita en la siguiente posición con aceleración, sino que desacelera (disminuye el factor de inercia) para volar hacia el valor óptimo, lo que conduce a aumentar la intensidad de búsqueda en un área pequeña. De lo contrario, se aumenta la velocidad y la intensidad de búsqueda en un área grande.

Mientras tanto, con el fin de evitar el efecto de similitud del enjambre, la capacidad de salir de óptimos locales debiese ser mejorada, es decir, cuando el gado de agregación se hace más grande, el peso de inercia debería incrementar proporcionalmente.

En consecuencia w se define de la siguiente forma:

$$w'_i = w_{ini} - \alpha(1 - h'_i) + \beta s \quad (3.3.8)$$

Donde w_{ini} es el valor inicial de w y es inicializado en 1. Puesto que $0 \leq h < 1$ y $0 \leq s \leq 1$, se puede obtener $1 - \alpha \leq w \leq 1 + \beta$. La elección de los valores de α y β típicamente están dentro del rango $[0, 1]$.

3.3.4.3 PSO con Factor de Constricción

Introducida por Clerc [11] en 1999, este factor consta de un conjunto de ecuaciones lineales que, al menos teóricamente, derivan en la convergencia óptima de las partículas. En este modelo, la velocidad está restringida por una constante χ , para asegurar la convergencia a un punto estable del espacio de solución, sin la necesidad de controlar que la velocidad exceda un valor máximo. La ecuación (3.3.1) de la velocidad se reemplaza por la siguiente:

$$v_{id}(t+1) = \chi \cdot [v_{id}(t) + c_1 \cdot r_1 \cdot (p_{id}(t) - x_{id}(t)) + c_2 \cdot r_2 \cdot (p_{gd}(t) - x_{id}(t))] \quad (3.3.9)$$

El factor de constricción χ se define de la siguiente forma:

$$\chi = \frac{2K}{|2 - \phi - \sqrt{\phi \cdot (\phi - 4)}|} \quad (3.3.10)$$

Donde

$$\phi = \phi_1 + \phi_2$$

$$\phi \geq 4$$

$$\phi_1 \geq c_1 \cdot r_1$$

$$\phi_2 \geq c_2 \cdot r_2$$

$$K \in [0;1]$$

El parámetro κ controla la exploración y explotación del enjambre. Para valores de κ cercanos a 0, se obtiene una rápida convergencia con explotación local. Para valores de κ cercanos a 1, la convergencia se produce más lentamente y se experimenta un grado mayor de exploración. Aunque el valor de κ es constante, algunas variaciones comienzan con valores altos y, a medida que transcurren los ciclos, se va decrementando a fin de aumentar la explotación. Aunque este factor es bastante similar al de inercia, el de constricción presenta algunas ventajas:

- No es necesario controlar que la velocidad se mantenga inferior a un valor máximo.
- El modelo de constricción garantiza la convergencia con los parámetros antes indicados.
- Los cambios de dirección son regulados por las constantes ϕ_1 y ϕ_2 del factor de constricción.

3.3.4.4 PSO Completamente Informado

Esta variante de PSO fue propuesta por Mendes [12] en 2004, en donde cada individuo recibe información completa de todos sus vecinos topológicos. Está basada en el modelo del factor de constricción, argumentando que el valor de ϕ no debería ser dividido en sólo dos valores, sino que se deberían tomar en cuenta todos los individuos del vecindario y, que el valor ϕ sea proporcional a éstos. Para una partícula dada, ϕ debe ser descompuesto en:

$$\phi_k = \frac{\phi}{|N|} \forall k \in N$$

Donde N es el vecindario de la partícula. Por lo tanto, la ecuación de la velocidad queda de la siguiente forma:

$$v_{id}(t+1) = X \cdot \left[v_{id}(t) + \sum_{k \in N} \phi_k \cdot W(k) \cdot r_k (p_{id}(t) - x_{id}(t)) \right] \quad (3.3.11)$$

W es una función de peso establecida previamente.

3.3.4.5 PSO Jerárquico

Presentado por Janson et al. [80], consiste de un algoritmo que dinámicamente adapta la topología de la población. Los individuos se encuentran organizados en una estructura de árbol, donde aquellos con mejor valor de fitness se encuentran en los nodos superiores. La jerarquía está definida por altura h , el grado de ramificación d o número máximo de hijos de nodos interiores y el total de número de nodos m del árbol.

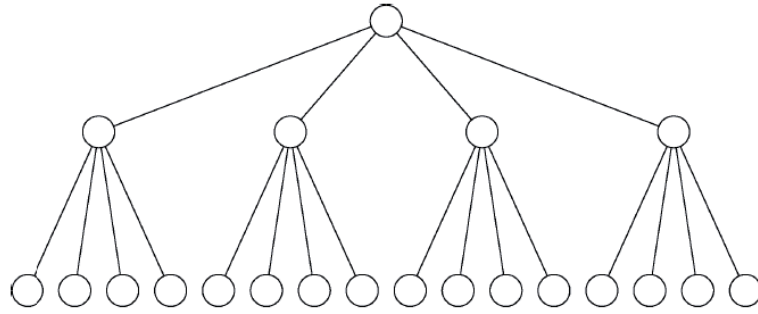


Figura 3.8 Ejemplo jerarquía definida mediante un árbol regular con $h=3$, $d=4$, $m=21$

En cada iteración las partículas hijas actualizan las velocidades considerando su mejor posición y el mejor valor obtenido por su padre. Luego, el desempeño de los hijos se compara con el de los padres, y en caso de que el primero sea mejor que el de sus progenitores, se intercambian las posiciones de los individuos en la jerarquía del árbol. El grado de ramificación que se utilice para el árbol influye en el balance de la exploración y explotación del algoritmo.

3.3.4.6 Modelo Puramente Cognitivo

Esta variante provee una tendencia a que las partículas regresen a las mejores posiciones alcanzadas en su pasado, ya que excluye la componente social de la fórmula de actualización de velocidad de PSO convencional. De esta forma, la ecuación queda de la siguiente manera:

$$v_{id}(t+1) = v_{id}(t) + c_1 \cdot r_1 \cdot (p_{id}(t) - x_{id}(t)) \quad (3.3.12)$$

El factor de inercia puede ser incluido en la ecuación si se considera necesario. En varios estudios se ha comprobado que este modelo es levemente más vulnerable a las fallas, comparado con el modelo completo (el que incluye la componente social). Esta variante tiende a explotar áreas locales donde las partículas fueron inicializadas. El número de iteraciones que requiere es mayor que el modelo completo y, falla cuando los coeficientes de aceleración y velocidad son valores pequeños

3.3.4.7 Modelo Puramente Social

Al contrario del modelo Puramente Cognitivo, esta variante excluye la parte de la componente cognitivo. De esta forma, la ecuación queda de la siguiente manera:

$$v_{id}(t+1) = v_{id}(t) + c_2 \cdot r_2 \cdot (p_{gd}(t) - x_{id}(t)) \quad (3.3.13)$$

El valor p_{gd} puede ser reemplazado por p_{id} si se utilizan vecindarios. Las partículas siguen la atracción producida por el mejor del enjambre o el mejor del vecindario. No considera de manera directa la posibilidad de retornar a un lugar visitado previamente. Se ha demostrado que este modelo es más rápido y efectivo que el modelo cognitivo y, en algunos casos, que el modelo completo.

3.3.4.8 PSO con Actualizaciones Síncronas y Asíncronas

La diferencia entre una actualización asíncrona con una síncrona, radica en el instante en el cual se realiza la actualización de la memoria de cada partícula y el conocimiento social del grupo.

En el modelo síncrono, todas las partículas se mueven en paralelo. El Fitness de las partículas se evalúa en cada iteración, se actualiza su memoria p_{id} y el conocimiento social p_{gd} . Por otro lado, en el PSO asíncrono al momento de desplazarse, cada partícula aprovecha la información actualizada por sus inmediatos predecesores, es decir, en cada iteración k , la i -ésima partícula se desplaza hacia un nuevo punto, utilizando la información de los vectores p_{id} y p_{gd} , actualizados por las $i-1$ partículas previas. Seguidamente, la partícula evalúa la calidad del nuevo punto y actualiza, si procede, las variables p_{id} y p_{gd} .

Esta información se transmite a las restantes partículas. Al actualizar la información partícula a partícula, el modelo asíncrono acelera la optimización, aunque la naturaleza del modelo síncrono lo hace susceptible de ser ejecutado en paralelo, sobre múltiples procesadores.

3.3.4.9 De acuerdo al tipo de vecindario

Una de las principales características que posee PSO es la interacción social que se observa entre los individuos del enjambre. Por ende, es importante el estudio de las posibles estructuras sociales que pueden incluirse en PSO. Todos los individuos pertenecientes a la misma estructura social o topología de vecindad comparten información, aprenden y siguen a los mejores dentro de su propia estructura.

Diversos estudios han mostrado que la comunicación entre los individuos de un grupo o población así como su desempeño, están fuertemente ligados a la topología o estructura social que posee el grupo. Además, dichas interacciones se observan particularmente entre los vecinos inmediatos adyacentes, y son estas interacciones las que provocan que el algoritmo funcione, ya que cada partícula por sí misma no podría evolucionar demasiado. Dicho de otro modo, dependiendo de la topología que adquiera la población, la transmisión de la información entre individuos puede acelerarse o ralentizarse, lo cual está relacionado con la velocidad de convergencia y con la capacidad del algoritmo para escapar de soluciones locales.

Una de las estructuras más simple que se plantea es aquella donde todos los individuos son influenciados exactamente de la misma forma, por la mejor solución encontrada por algún miembro de la población. Este tipo de modelo recibe el nombre de *gbest* o mejor global, y corresponde a una estructura donde todos los individuos están conectados entre sí.

Otra de las principales topologías corresponde a *lbest* o mejor local. En esta cada individuo es afectado por el mejor desempeño de sus k vecinos inmediatos en la población topológica (comúnmente llamada vecindario).

Como se nombró anteriormente la elección de la topología es uno de los principales inconvenientes que presenta la implementación de PSO, debido a la importancia y fuerte

influencia que esta ejerce en el desempeño del algoritmo. La topología mejor global tiende a converger prematuramente, debido a la pérdida de la diversidad. Por otra parte, la topología mejor local es más lenta de converger, pero conserva la diversidad. Los resultados obtenidos con algoritmos que incluyen vecindarios suelen ser, generalmente, mejores cuando son comparados con los obtenidos con topologías de mejor global.

Existen diversas estructuras de mejor local o *lbest* definidas en [13], estas son:

- Circular o Anular: cada individuo está conectado con sus k vecinos inmediatos, intentando imitar al mejor de ellos. En la Figura 3.9(a) se observa un ejemplo con $k = 2$. Aquí cada vecindario se encuentra solapado para facilitar el intercambio de información y, de esta manera, converger a una única solución al final del proceso de búsqueda. La información entre vecindarios fluye lentamente, lo cual provoca que la convergencia sea más lenta pero al mismo tiempo más segura. En funciones multimodales (aquellas que poseen varios óptimos) este tipo de estructura suele obtener un buen desempeño con respecto a otras estructuras. Muchas variantes de este tipo de estructura pueden ser creadas, en donde la conexión de las partículas puede ser elegida aleatoriamente o siguiendo algún criterio, dependiendo del problema. La Figura 3.9(b) muestra un ejemplo.
- Rueda: un único individuo (el central) está conectado a todos los demás, y todos estos están conectados solamente al individuo central, como se muestra en Figura 3.9(c). En esta estructura una sola partícula sirve de punto focal, y toda la información entre los individuos es canalizada por ese punto. La partícula central compara el desempeño de todas las partículas en el vecindario para ajustar las posiciones a la mejor. Tan pronto como la mejor posición sea detectada, la información es enviada a todas las partículas para que efectúen el cambio. La estructura de rueda propaga las buenas soluciones en forma demasiado lenta lo cual puede no ser del todo bueno, porque el proceso de búsqueda se tornaría demasiado lento. Algunas variantes pueden ser introducidas, como desconectar algunas partículas de la partícula central y conectarlas a otras partículas en el vecindario (Figura 3.9(d)).
- Arcos Aleatorios: para n partículas, n conexiones simétricas son asignadas entre pares de individuos. Como su nombre lo indica, las conexiones son asignadas de forma aleatoria entre las n partículas. La Figura 3.10(a) muestra esta estructura para un tamaño de población de 12 individuos.
- Pirámide: la idea de esta topología es la creación de una estructura tridimensional que comunique a las partículas. Se asemeja a un modelo de alambre tridimensional, como se observa en la Figura 3.10(b).
- Toroidal: esta topología puede utilizarse como estructura de vecindario. Es una superficie cerrada con base en la topología de malla (Figura 3.10(c)) pero a diferencia de esta última donde cada partícula posee dos vecinos, en la toroidal posee 4 vecinos directos (Figura 3.10(d)).

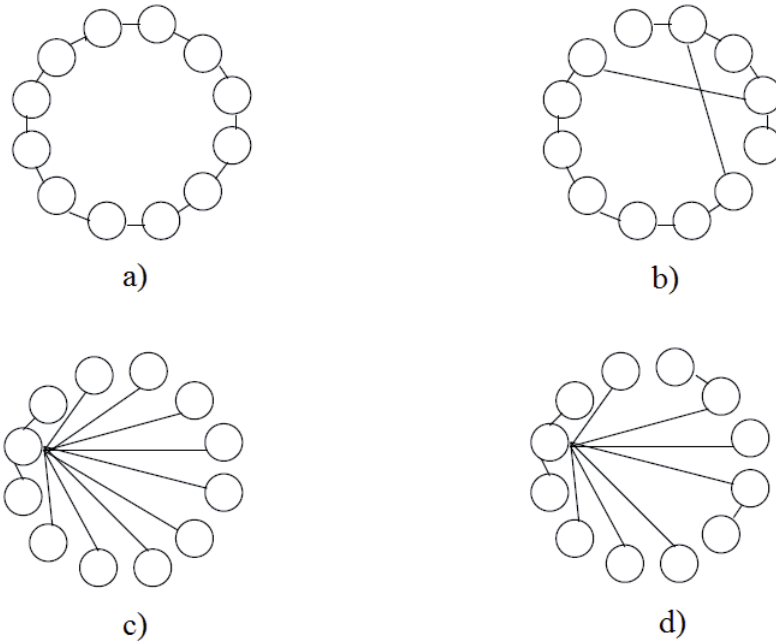


Figura 3.9 (a) Circular con $k=2$. (b) Circular con conexiones variantes y 2 subvecindarios. (c) Rueda. (d) Ruedas con conexiones variantes

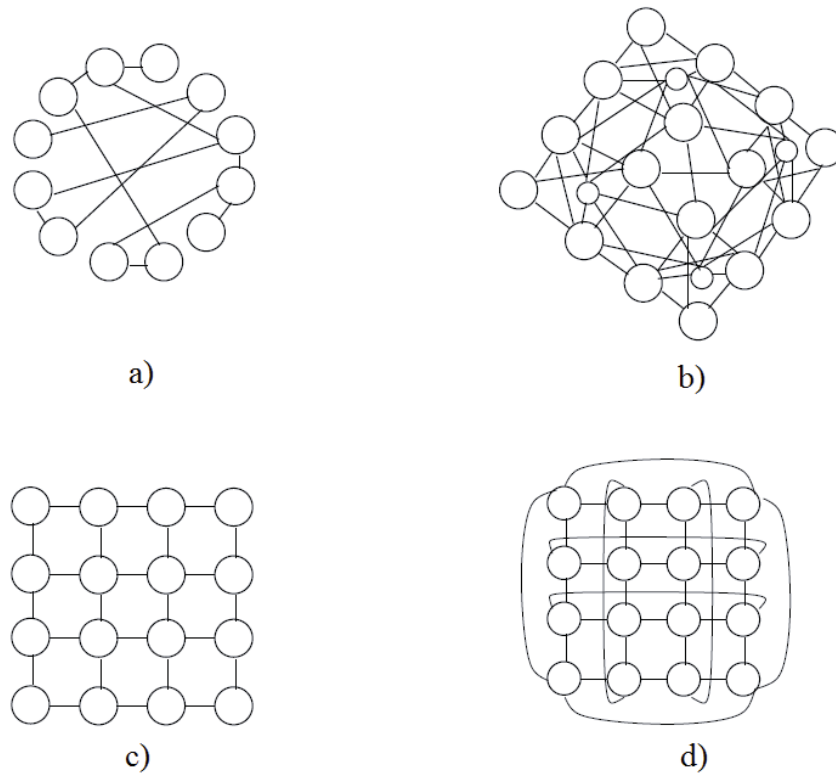


Figura 3.10 (a) Arcos aleatorios. (b) Piramidal. (c) Malla. (d) Toroidal

Cada una de estas topologías, puede variar algunas de sus conexiones, para constituir subvecindarios. Lo importante es lograr una topología que mejore el desempeño del algoritmo. Actualmente, no existe una topología que sea buena para todo tipo de problema. Generalmente, las estructuras completamente conectadas son mejores para problemas unimodales (un único óptimo), mientras que las menos conectadas funcionan mejor para problemas multimodales, dependiendo del grado de interconexión entre las partículas.

El objetivo de utilizar estructuras de vecindarios es evitar la convergencia prematura del algoritmo hacia óptimos locales. Esto es posible porque en una topología de vecindario, cada individuo es influenciado por el mejor valor encontrado por grupos más pequeños de partículas, y no por el mejor valor hallado por algún individuo de la población entera. Con una estructura de vecindario, el algoritmo tiende a explorar mejor el espacio de búsqueda y, de esta manera, efectúa una convergencia más lenta posibilitando aumentar la calidad de las soluciones encontradas.

4 Estado del Arte Accidentes de Tránsito

Desde hace más de una década se han estudiado y analizado accidentes de tránsito a través de herramientas de minería de datos en todo el mundo, algunos de ellos se detallan a continuación.

Se han analizado accidentes, como en 1997 por Dia y Rose [14]. Utilizaron datos del mundo real para comparar la técnica red neuronal MLP, con un modelo de detección de accidentes en la autopista de Melbourne. Los resultados mostraron que el modelo de la red neuronal era más fiable, ya que podía detectar más rápido los accidentes sobre el modelo que estaba en operación en las autopistas de Melbourne. En 1999 Yang y otros [15], utilizaron redes neuronales para detectar patrones seguros de conducción que tuviesen menos probabilidad de causar la muerte y lesiones cuando se produce un accidente de tráfico. Concluyeron que mediante el control de una sola variable (velocidad de conducción o condiciones de luz), podrían reducir las muertes y lesiones hasta en un 40%.

Por su parte, Mussone, Ferrari y Oneta en 1999 [16], utilizaron redes neuronales para analizar accidente ocurridos en intersecciones en Milán, Italia. Eligieron las técnicas de *feed-forward* (MLP) con *Back Propagation* (BP). El modelo tenía 10 nodos de entrada para ocho variables (día o noche, flujos de tráfico en las intersecciones, número de puntos virtuales conflictivos, número de puntos reales conflictivos, tipo de intersección, tipo de accidente, estado de superficie de las carreteras y condiciones climáticas). El nodo de salida fue el índice de accidentes, que se calculó como el cociente entre el número de accidentes en una intersección dada y el número de accidentes en la intersección más peligrosa. Los resultados mostraron que el mayor índice de accidentes de atropellos en peatones era en intersecciones no señalizadas y en la noche. En el mismo año Evanco [17], realizó un análisis estadístico multivariado, para determinar la relación entre las muertes y los tiempos de notificación de accidentes. El análisis demostró que el tiempo de notificación de accidentes era un factor importante en el número de víctimas mortales en carreteras rurales.

En el año 2002 Bedard y otros [18], aplicaron una regresión logística para determinar la independencia entre la contribución del conductor, choque y características de los vehículos. Los autores encontraron que la prevención de fatalidades radicaba en el mayor uso del cinturón de seguridad, en la reducción velocidad y la reducción de impactos del lado del conductor. En el mismo año, Ossiander y Cummings [19], usaron modelos de regresión de Poisson para analizar la asociación entre el índice de accidentes mortales (accidentes mortales por cada milla recorrida del vehículo) y el aumento en el límite de velocidad. Concluyeron que al aumentar el límite de velocidad, este se asociaba con un mayor índice de accidentes mortales, por ende aumentaban las muertes en las carreteras del estado de Washington.

Por otra parte, Abdelwahab y otros en 2003 [20], estudiaron los accidentes de tránsito en Florida. El análisis se centró en los accidentes de vehículos producidos en intersecciones con semáforos. Se comparó el desempeño de las técnicas de Perceptron Multicapa (MLP) con Fuzzy ARTMAP, concluyendo que la precisión de clasificación de MLP era superior. En el mismo año, Sohn y Lee [21], aplicaron clustering para mejorar la precisión de clasificadores individuales para dos categorías de gravedad (lesiones corporales y daños a la propiedad). Para

ello utilizaron redes neuronales y árboles de decisión. Concluyeron que los algoritmos de *clustering* trabajan mejor para clasificar accidentes de tránsito.

En el mismo año 2003 Chong, Abraham y Paprzycki [22], utilizaron redes neuronales, árboles de decisión y un modelo híbrido, para construir modelos que pudiesen predecir la gravedad de las lesiones. Además incluyeron una breve aplicación con Maquinas de Soporte Vectorial. Concluyeron que para los casos en que las personas resultan ilesas o posiblemente lesionadas, se comportaba mejor el modelo híbrido que las redes neuronales, y para el caso no lesionado y posiblemente lesionado se pudo modelar mejor haciendo uso de árboles de decisión. La técnica de SVM no obtuvo muy buenos resultados, no obstante, en la Tabla 4.1 se aprecia el porcentaje de exactitud del estado lesionado e ileso, utilizando el tipo de *kernel* función de base radial. Cabe destacar que los parámetros de entrada σ^2 , C , fueron ajustados a juicio del investigador, sin utilizar alguna técnica que optimizara la estimación de estos. Además, se puede apreciar que el mayor porcentaje de exactitud obtenido fue de un 60,3%.

Tabla 4.1 Exactitud SVM usando kernel función de base radial

Parámetros	Exactitud
$\sigma^2 = 0.0001$; $C = 42.8758$	59,95%
$\sigma^2 = 0.001$; $C = 4.6594$	59,97%
$\sigma^2 = 0.5$; $C = 0.5$	59,38%
$\sigma^2 = 1.2$; $C = 0.5$	56,64%
$\sigma^2 = 1.5$; $C = 2$	54,67%
$\sigma^2 = 2$; $C = 10$	54,82%
$\sigma^2 = 0.00001$; $C = 100$	60,11%
$\sigma^2 = 0.0001$; $C = 100$	59,95%
$\sigma^2 = 0.001$; $C = 100$	60,3%

Por otra parte, miembros de la Escuela de Ingeniería en Transportes de la PUCV han trabajado en el análisis de accidentes de tránsito de las principales regiones de Chile mediante técnicas estadísticas y algunas técnicas de minería de datos. Por ejemplo en el 2006 Montt y otros [23], analizaron los siniestros mediante estadísticas descriptivas y árboles de decisión, sin embargo, no se utilizó una metodología que ameritase un trabajo con gran cantidad de datos. Más tarde, Montt, Zúñiga y Chacón en el 2009 [24], trabajaron con técnicas de minería de datos basadas principalmente en estructuras de reyes bayesianas. Procedieron a aplicar algoritmos de aprendizaje paramétrico y propagación de evidencia. Obtuvieron que la probabilidad de salir afectado en un accidente de tránsito era de un 65%, y que el tipo de atropello era el accidente con mayor probabilidad de ser afectado con un 99,52%. Además,

concluyeron que la mayor probabilidad de salir ileso era la colisión y que en zonas rurales existía mayor probabilidad de salir afectado con un 79%, y el 21% restante correspondía a la probabilidad de salir afectado en una zona urbana. Finalmente, el porcentaje de exactitud del modelo general que consistía en el estado (lesionado o ileso) en el cual resultan las personas involucradas de tránsito, fue de un 78%. En la Tabla 4.2 se muestra la sensibilidad y especificidad de este modelo.

Tabla 4.2 Sensibilidad y especificidad etapa de validación

	Sensibilidad	Especificidad
Red General	71%	87%

En el mismo año Montt, Musso y Chacón [25], utilizaron métodos de agrupamiento e índices de Calinski y Harabasz para encontrar agrupaciones que representaran mejor la información. Lograron caracterizar los distintos tipos de accidentes, como choques con objeto, volcadura, caídas y los atropellos, encontrando características particulares en cada uno de los tipos de accidentes.

El último informe de seguridad vial realizado por la organización mundial de la salud en el 2009 [1], sostuvo como conclusión que la tasa más alta de letalidad por cien mil habitantes correspondían a países de ingresos bajos y medios. Sin embargo, este estudio no utilizó herramientas de minería de datos sino que solo técnicas y herramientas estadísticas.

A través de una recolección de la literatura de análisis de accidentes de tránsito en el mundo y en Chile, específicamente en estudios realizados por la escuela de Ingeniería de Transportes de la PUCV, se puede inferir que la minería de datos ha aportado bastante ayuda y conocimiento al análisis en los accidentes de tránsito. Redes Neuronales, Arboles de Decisión, Redes Bayesianas y Clustering son técnicas predominantes en este contexto, sin embargo, las técnicas de clasificación de SVM y LS-SVM no han sido muy utilizadas por el hecho de que es una técnica más nueva y compleja. No obstante, tras la buena utilidad que ha prestado la minería de datos en accidentes de tránsito se deduce que es posible utilizar la técnica de clasificación LS-SVM.

5 Descripción de los datos en estudio

En el presente capítulo se detalla la estructura de los datos de accidentes de tránsito, así como también las fuentes de donde se extrajeron para el desarrollo de este trabajo de investigación.

5.1 Obtención

Los accidentes de tránsito en Chile son registrados por Carabineros. Ellos registran manualmente en una planilla llamada SIEC2 (ver anexo A: Planilla SIEC 2) cada accidente ocurrido, siguiendo un instructivo preestablecido. Posteriormente los datos son ingresados a una base de datos central de la CONASET.

Los datos fueron suministrados por la Escuela de Ingeniería en Transportes de la PUCV, cuya escuela es la que tiene el nexo directo con la CONASET. Los datos recolectados venían en planillas Excel y separados en distintos archivos, cada uno por año los que varían entre 2003 y 2009. Cada archivo estaba separado en 3 pestañas con información respecto a las tres entidades definidas por la CONASET, estas son:

1. **Accidente.** Información de las distintas características o atributos de un accidente en particular, estas son: identificador del accidente, fecha, hora de accidente, comuna, causas, tipo de accidente, estado atmosférico, condición calzada, tipo de calzada, estado calzada, urbano/rural, y la información que indica la cantidad de ilesos, leves, graves, menos graves y muertos.
2. **Persona.** Información de las características o atributos de las personas involucradas en los accidentes, estas son: el identificador del accidente, sexo, calidad, edad, y resultado.
3. **Vehículo.** Información de las características o atributos de los vehículos involucrados en los accidentes, estos son el identificador del accidente, servicio, y el tipo de vehículo.

5.2 Estructura de los datos

Los datos originarios de la base de datos de la CONASET como se nombró anteriormente cuentan con tres entidades, estas son: Accidente, Persona y Vehículo. (Figura 5.1)

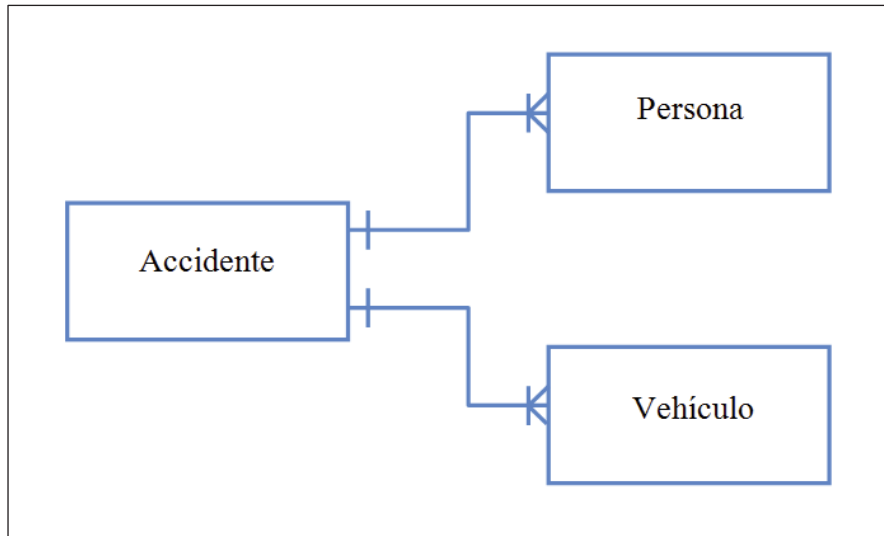


Figura 5.1 Entidades base de datos CONASET

Mediante la interacción de estas tres entidades, se aprecia que en cada accidente hay por lo menos una o muchas personas y uno o muchos vehículos involucrados. Además cada entidad es definida por atributos o características propias. Estas se detallan a continuación:

1. **Accidentes.** Entidad correspondiente a los accidentes ocurridos en una determinada fecha y lugar geográfico. Posee una serie de atributos, estos son:
 - Idaccidente: identificador único y correlativo por cada accidente ocurrido en todo Chile.
 - Fecha: fecha del suceso.
 - Hora: hora del suceso.
 - Región: región del suceso.
 - Comuna: comuna de la región en donde ocurrió el accidente.
 - Tipo de accidente: existen diez tipos accidentes, estos son:
 - Atropello
 - Caída
 - Colisión: dentro de estas se presentan las colisiones frontal, lateral, por alcance y perpendicular.
 - Impacto con animal
 - Choque con objeto: dentro de estos se presentan los choques con objeto frontal, lateral y posterior.
 - Choque con vehículo detenido: dentro de estos se presentan los choque con vehículo detenido frente/frente, frente/lado, frente/posterior, lado/frente, lado/lado, lado/posterior, posterior/frente, posterior/lado y posterior/posterior.
 - Volcadura
 - Incendio
 - Descarrilamiento
 - Otro tipo
 - Causa: corresponde al origen del accidente, existiendo dieciocho tipos, estas son:

- Fallas mecánicas: dentro de estas se encuentran las fallas del tipo frenos, dirección, eléctrica, suspensión, neumáticos, motor y carrocería.
- Adelantamiento: dentro de estas se encuentran los adelantamientos sin el espacio o tiempo suficiente, sin efectuar la señal respectiva, por la berma, sobrepasando línea continua, en cruce, curva, cuesta, puente, conducción, bajo la influencia del alcohol, bajo la influencia de drogas o estupefacientes, contra el sentido del tránsito, en estado de ebriedad, físicas deficientes (cansancio, sueño), por izquierda eje calzada, no atento condiciones del tránsito momento, sin mantener distancia razonable ni prudente y cambiar sorpresivamente pista de circulación.
- No respetar derecho a paso: dentro de estos se encuentran los derechos al peatón y vehículo.
- Pasajero: dentro de estos se presentan las situaciones como si sube o desciende de vehículo movimiento, viaja en pisadera de vehículo, imprudencia y ebriedad.
- Peatón: dentro de estos se presentan las situaciones: permanece sobre la calzada, cruza calzada forma sorpresiva o descuidada, imprudencia, ebriedad, cruza calzada fuera paso peatones, y cruza camino o carretera sin precaución.
- Señalización: dentro de ellas están: mal instalada o mantenida forma defectuosa, desobedecer luz roja de semáforo, desobedecer indicación carabinero servicio, desobedecer señal ceda el paso, desobedecer señal pare, desobedecer otra, semáforo mal estado o deficiente y desobedecer luz intermitente semáforo.
- Velocidad: dentro de ellas se encuentran: mayor que máxima permitida, no razonable ni prudente, no reducir cruce de calles, cumbre, curva, etc., exceso en zona restringida, menor que mínima establecida,
- Carga: dentro de ellas se encuentran: mayor que la autorizada vehículo, obstruye visual conductor, escurre a la calzada, sobresale estructura vehículo.
- Virajes indebidos.
- Animales sueltos en vía pública.
- Vehículos en retroceso, conducir
- Vehículos en panne sin señalización o deficiente
- Pérdida control vehículo
- Suicidio
- Causas no determinadas
- Otras causas
- Fuga por hecho delictual
- Ubicación relativa: Existen diecisiete ubicaciones, estos son:
 - Cruce con semáforo funcionando, cruce sin señalización, enlace a desnivel, plaza de peaje, acera o berma, enlace a nivel, tramo de vía curva vertical, túnel, cruce con señal "pare", cruce regulado por carabinero, tramo de vía recta, acceso no habilitado, tramo de vía curva horizontal, rotonda, otros no considerados, cruce con señal "ceda el paso", cruce con semáforo apagado y puente.
- Estado Atmosférico: Existen seis estados atmosféricos, estos son:

- Despejado, nublado, lluvia, llovizna, neblina, nieve.
 - Tipo calzada: Existen seis tipos de calzada, estos son:
 - Concreto, asfalto, adoquín, mixto, ripio y tierra.
 - Estado Calzada: estas son:
 - Bueno, regular, malo.
 - Condición Calzada: estos son nueve:
 - Seco, húmedo, mojado, con barro, con nieve, con aceite, escarcha, gravilla y otros.
 - Muertos: cantidad de muertos involucrados en el accidente.
 - Graves: cantidad de graves involucrados en el accidente.
 - Menos Graves: cantidad de menos graves involucrados en el accidente.
 - Leves: cantidad de leves involucrados en el accidente.
 - Ilesos: cantidad de ilesos involucrados en el accidente.
- 2. Personas:** Entidad asociada a las personas involucradas en un accidente en particular. Un accidente por lo menos posee una persona involucrada, ya sea conductor, pasajero o peatón. Los atributos relacionados a esta entidad son:
 - Calidad: se subdividen en:
 - Peatón, conductor, pasajero.
 - Sexo
 - Resultado: corresponden a cinco estados en los que pueden quedar las personas:
 - Muerto, grave, menos grave, leve, ileso.
- 3. Vehículos:** Entidad asociada a los vehículos involucrados en un accidente en particular. Un accidente por lo menos presenta uno o más vehículos. Los atributos y características asociadas a esta entidad son:
 - Tipo vehículo: dentro de los tipos se encuentran treinta tipos de vehículos, estos son:
 - Bus/ taxi bus, minibús, trolebús, automóvil, camioneta, jeep, furgón, ambulancia, camión simple, camión simple con remolque, tracto-camión, tracto-camión con remolque, carro bomba, carro transporte de valores, remolque/semi remolque, motocicleta, motoneta/bicimoto, moto arenera, bicicleta, tracción animal, carro tracción humana, tractor, maquinaria agrícola, maquinaria movimiento tierras, maquinaria industrial, patín/patineta, patín motorizado, ferrocarril, dado a la fuga, otros no clasificados.
 - Servicio: dentro de los tipos de servicios encontramos 17 tipos de servicio, estos son:
 - Carabineros, fiscal, particular, transporte escolar, taxi básico, taxi colectivo urbano, taxi colectivo rural, bomberos, salud, locomoción colectivo urbano, locomoción colectivo rural, servicio interurbano, servicio internacional, carga normal, carga peligrosa, dado a la fuga, otros sin especificar.

6 Modelo LS-SVM – PSO

En esta sección se presenta el modelo base para la construcción del clasificador. Para ello, se utiliza la máquina clasificadora (LS-SVM) a la que se agrega la metaheurística PSO para la selección y estimación de los parámetros de esta. Además, se presentan las métricas más utilizadas para evaluar el rendimiento de un clasificador binario, la cuales corresponden a la exactitud, sensibilidad, especificidad y la curva ROC.

6.1 Descripción del Modelo General

El modelo representado en la Figura 6.1, representa el modelo base para la estimación de los parámetros de LS-SVM mediante el uso de PSO. Específicamente abarca los siguientes puntos:

1. Estimar los parámetros de LS-SVM mediante la metaheurística PSO (Compatible con cualquier variante de PSO), para lograr un mayor grado de generalización del modelo, es decir, que al momento de clasificar un nuevo elemento sea capaz de diferenciar correctamente a que clase corresponde.
2. Construir un clasificador para el estado en el cual resultan las personas involucradas en accidentes de tránsito. Específicamente clasificar si una persona termina en estado Lesionado o Ileso, dada las características que definen el suceso.

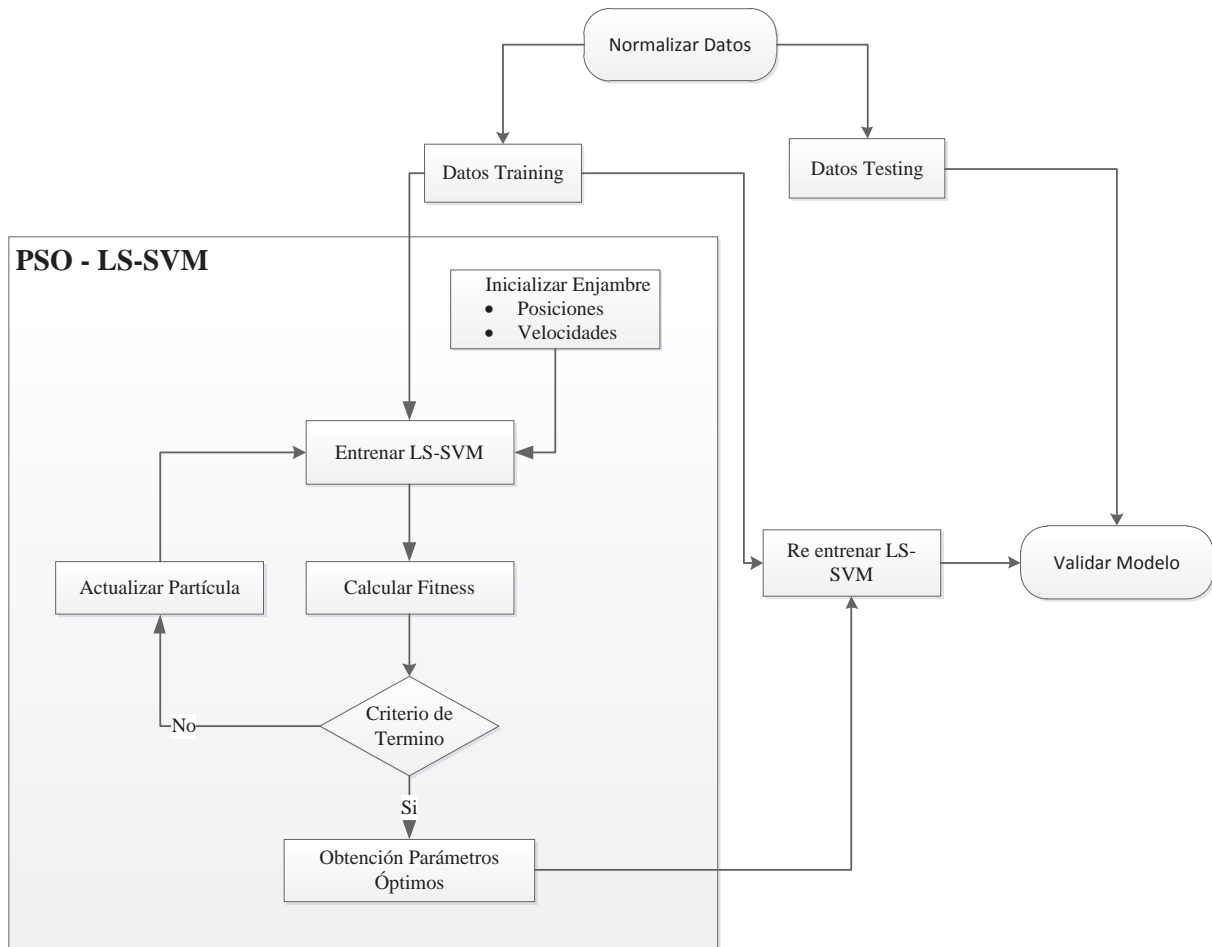


Figura 6.1 Modelo General de la solución

Los pasos que conlleva este modelo se describen a continuación.

Normalizar Datos. En primera instancia los datos son normalizados en un rango específico y son divididos para cada etapa: entrenamiento y validación.

Inicializar Enjambre. Corresponde a inicializar las posiciones y velocidades de las partículas, por ende se debe saber a priori cómo representar o codificar los parámetros en la metaheurística PSO. Esta codificación debe incluir el parámetro de penalización de errores y los parámetros del *kernel* a utilizar, ya que se pretende buscar la combinación de parámetros que proporcione un mejor rendimiento al clasificador.

De esta forma, la codificación de los parámetros para ser representados algoritmo de optimización por enjambre de partículas depende del tipo de *kernel* escogido para el entrenamiento. Por ende los parámetros o partículas se representan de la siguiente forma:

Tabla 6.1 Codificación Partícula

Tipo de Kernel	Codificación Vector Partículas
Kernel RBF	Vector: $[C, \sigma^2]$
Kernel Polinomial	Vector: $[C, d, T]$
Kernel Perceptrón de dos capas	Vector: $[C, s, \theta]$

Actualizar Partículas. En este paso se actualizan las posiciones y velocidades de las partículas dependiendo del tipo de PSO a utilizar. Luego se seleccionan los parámetros que serán ingresados a la etapa de entrenamiento de la máquina. Dependiendo de la t -ésima iteración, se escogen las partículas correspondientes y se entregan a LS-SVM para entrenar y calcular el *fitness*.

Entrenar LS-SVM y Evaluar Rendimiento. Se entrena LS-SVM con los parámetros seleccionados anteriormente y se retorna el valor *fitness* de cada partícula en la t -ésima iteración. Luego se realizan las respectivas comparaciones en base a la función de desempeño, la cual debe minimizar el costo de entrenamiento de los datos. El *fitness* o función de desempeño puede calcularse de distintas formas, aquí se presentan las siguientes:

- **Ratio de elementos mal clasificados.** Corresponde a 1-exactitud, es decir, al porcentaje de datos mal clasificados durante el entrenamiento. Operacionalmente, consiste en comparar las etiquetas originales de los datos de entrenamiento, con las etiquetas predichas por el modelo, y así comprobar el total de elementos mal clasificados. Para ello, se puede utilizar la función del error cuadrático medio.
- **Costo Validación Cruzada.** Corresponde al costo de entrenamiento de la validación cruzada. Básicamente consiste en dividir los datos de entrenamiento aleatoriamente en L -partes. En la i -ésima ($i=1, \dots, L$) iteración, la i -ésima parte de los datos es usada para testear o validar (datos de testing) y las $L-1$ partes son asignadas para datos de entrenamiento, midiendo el costo de elementos mal clasificados. De la misma forma, se itera hasta la última parte del conjunto de datos, rescatando el costo en cada iteración. Finalmente se realiza un promedio de los costos de las distintas iteraciones y se retorna.

Una representación de dicha validación se hace a continuación:

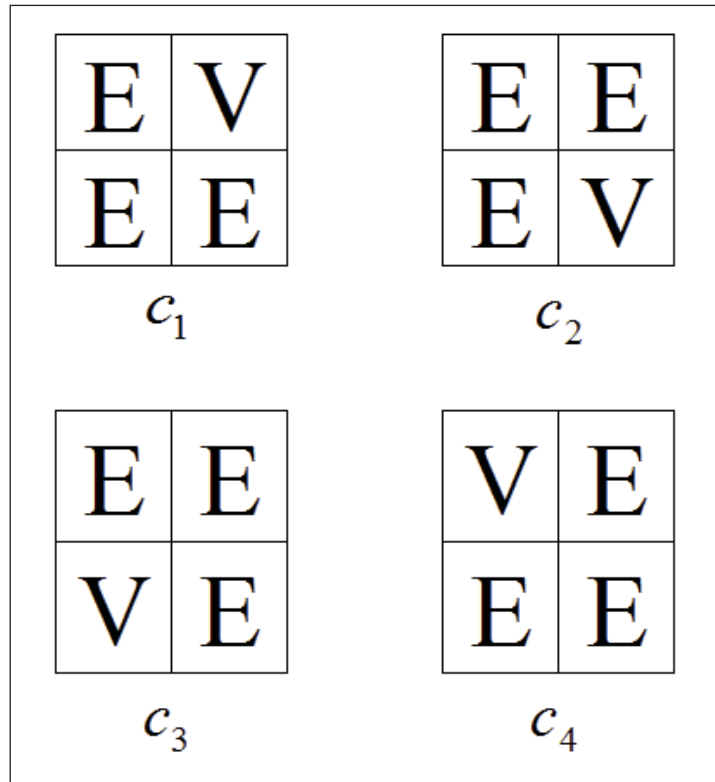


Figura 6.2 Esquema para validación cruzada para 4 subconjuntos

E = Datos de Entrenamiento

V = Datos de Validación

c_i = Costo de elementos mal clasificados del subconjunto i

La suma de todos los costos c_i divididos por el total de subconjuntos, da lugar al costo promedio utilizado como *fitness* en el modelo.

Criterio de término. Si no se ha alcanzado el número total de iteraciones o no se ha alcanzado algún valor óptimo definido a priori, se actualizarán las posiciones y velocidades de cada partícula y se entrenará nuevamente. Por el contrario si se ha llegado a algún valor óptimo el cual se haya definido a priori, o se haya alcanzado el número total de iteraciones el ciclo de búsqueda de los parámetros finaliza.

Obtención de los parámetros óptimos y re entrenamiento. Una vez encontrado los parámetros óptimos, el modelo es reentrenado con dichos parámetros.

Validar Modelo. Finalmente el modelo es validado con la data de testing. Para ello, se utilizan diversas métricas, entre ellas exactitud, sensibilidad y especificidad.

Cabe destacar que la utilización de PSO y sus variantes para estimar los parámetros óptimos del modelo se basa en mezclar sus cualidades de exploración y explotación, de manera que se pueda abarcar todo el espacio de búsqueda y encontradas las mejores regiones donde poder encontrar un óptimo local que se acerque lo más posible a una solución óptima de la red.

6.2 Métricas de rendimiento para evaluar el clasificador

Para construir un modelo de clasificación se debe pasar por dos etapas: entrenamiento y prueba. Cada una de estas etapas debe ser medida para conocer el grado de exactitud de la clasificación. Si la etapa de entrenamiento logra un alto porcentaje de elementos clasificados, se dice que el modelo obtuvo un buen aprendizaje, de lo contrario se dice que se está en presencia de sobreajuste o *overffiting*, es decir cuando un modelo se entrena demasiado (se sobreentrena) o se entrena con datos extraños, por lo que el algoritmo de aprendizaje puede quedar ajustado a unas características muy específicas de los datos de entrenamiento que no tienen relación causal con la función objetivo. Por otra parte, si la etapa de prueba obtiene un alto porcentaje de elementos bien clasificados, se dice que el modelo de clasificación generaliza bien, de lo contrario se dice que el modelo no logra clasifica bien los nuevos elementos. Sin duda, lo que se pretende buscar es que el modelo de clasificación tenga un buen aprendizaje para que de esta forma generalice de la mejor manera y el rendimiento sea el óptimo.

En base a lo anterior, se debe medir cada etapa para comprobar la exactitud de cada una de ellas. Para ello, existen muchos métodos estadísticos para evaluar y comprobar el rendimiento de un clasificador.

En particular, para este trabajo de investigación se consideran las métricas de exactitud, sensibilidad, especificidad y área de la curva ROC. Dichos métricas son creadas en base a los siguientes errores de prueba:

- Verdaderos Positivos (VP): número de éxitos. En este contexto corresponden al número de personas detectadas lesionadas correctamente.
- Verdaderos Negativos (VN): número de rechazos correctos. En este contexto corresponde a las personas detectadas ilesas correctamente.
- Falsos Positivos (FP): número de falsas alarmas. En este contexto corresponden al número de personas detectadas lesionadas, siendo que en realidad resultaron ilesas.
- Falsos Negativos (FN): En este contexto corresponde al número de personas detectadas ilesas, siendo que en realidad resultaron lesionadas.

Los VP, VN, FP y FN son resumidos en la siguiente matriz de confusión o tabla de contingencia:

Tabla 6.2 Matriz de Confusión

		Resultado Real	
		Lesionadas	Ileso
Resultado Obtenido	Lesionadas	VP	FP
	Ileso	FN	VN

A partir de los Verdaderos Positivos (VP), Verdaderos Negativos (VN), Falsos Positivos (FP) y Falsos Negativos (FN) se construyen los siguientes ratios de error, los cuales corresponden a las métricas bases para este trabajo:

- **Exactitud:** corresponde al total de personas bien clasificadas, ya sea con lesión o sin lesión, dentro del total de personas de personas clasificadas. Es representada de la siguiente forma:

$$Exactitud = \left(\frac{VP + VN}{VP + VN + FN + FP} \right) \quad (6.2.1)$$

- **Sensibilidad:** corresponde a la probabilidad de que una persona realmente lesionada sea detectada como tal por la prueba. Es representada por la siguiente ecuación:

$$Sensibilidad = \left(\frac{VP}{VP + FN} \right) \quad (6.2.2)$$

- **Especificidad:** corresponde a la probabilidad de que una persona ilesa sea detectada como tal por la prueba. Es representada por la siguiente ecuación:

$$Especificidad = \left(\frac{VN}{VN + FP} \right) \quad (6.2.3)$$

- **Valor Predictivo Positivo:** corresponde a la probabilidad de padecer la lesión si se obtiene un resultado positivo en el test. Por ejemplo si se tiene un VPP de 80%, significa que el 80% de las personas que se detecten como lesionadas van a estar realmente lesionadas. Es representada por la siguiente ecuación:

$$VPP = \left(\frac{VP}{VP + FP} \right) \quad (6.2.4)$$

- **Valor Predictivo Negativo:** corresponde a la probabilidad de que una persona con un resultado negativo en la prueba esté realmente ileso. Por ejemplo si se tiene un VPN de 90%, significa que el 90% de las personas que se detecten como ilesos van a estar realmente ilesos. Es representada por la siguiente ecuación:

$$VPN = \left(\frac{VN}{VN + FN} \right) \quad (6.2.5)$$

La exactitud, sensibilidad y especificidad son condiciones necesarias para medir grado de **validez** del modelo, es decir, el grado en que un test mide lo que se supone que debe medir. Por otra parte, el valor predictivo positivo y el valor predictivo negativo están relacionados con la **seguridad** del modelo, es decir, con qué seguridad el clasificador predecirá la presencia o ausencia de la lesión. Además, se debe medir el grado de **reproductividad** del modelo, la cual corresponde a la capacidad del clasificador para ofrecer los mismos resultados cuando se repite su aplicación en circunstancias similares, es decir, validando el modelo con distintos tamaños de muestras.

Otra de las métricas más usadas en clasificación binaria corresponde a la curva ROC (*Receiver Operating Characteristic*). El ROC es un gráfico en el que se observan todos los pares sensibilidad/especificidad resultantes de la variación continua de los puntos de corte en todo el rango de resultados observados. En el eje y de coordenadas se sitúa la sensibilidad o fracción de verdaderos positivos, definida como se muestra en la ecuación (6.2.2). En el eje x 1-especificidad.

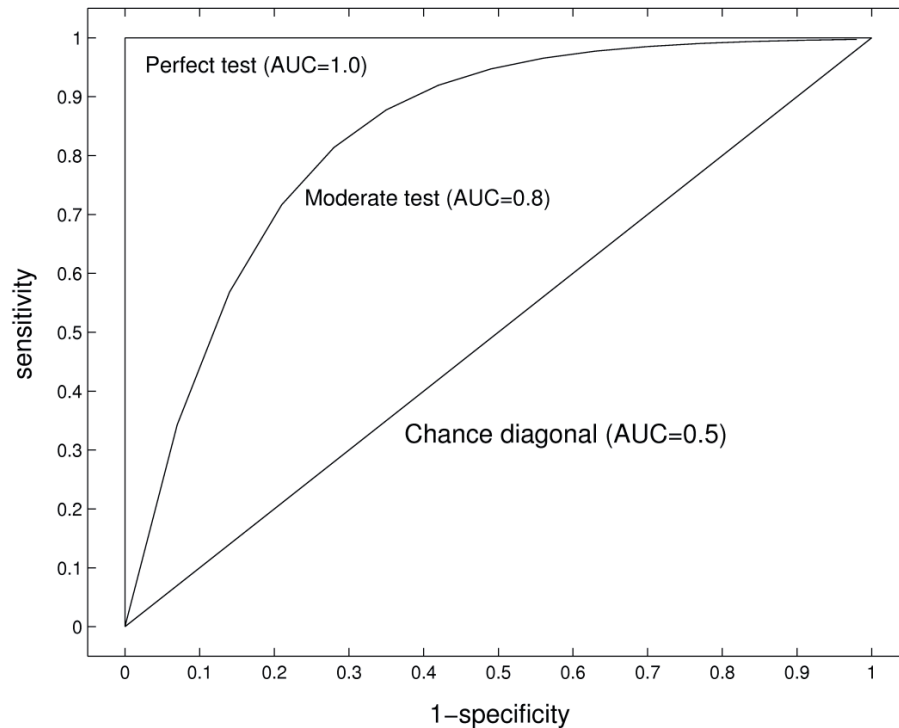


Figura 6.3 Curva ROC

Cada punto de la curva representa un par S/1-E correspondiente a un nivel de decisión determinado (ver Figura 6.3). Una prueba con discriminación perfecta, sin solapamiento de resultados en las dos poblaciones, tiene una curva ROC que pasa por la esquina superior izquierda, donde S y E toman valores máximos (S y $E = 1$). Una prueba sin discriminación, con igual distribución de resultados en los dos subgrupos, da lugar a una línea diagonal de 45° , desde la esquina inferior izquierda hasta la superior derecha. La mayoría de las curvas ROC caen entre estos dos extremos. Si cae por debajo de la diagonal de 45° se corrige cambiando el criterio de positividad de “mayor que” a “menor que” o viceversa.

Cualitativamente, cuanto más próxima es una curva ROC a la esquina superior izquierda, más alta es la exactitud global de la prueba. De la misma forma, si se dibujan en un mismo gráfico las curvas obtenidas con distintas pruebas, aquella que esté situada más hacia arriba y hacia la izquierda tiene mayor exactitud. De esta forma, por simple observación se obtiene una comparación cualitativa.

A partir de la curva ROC se deriva el “Área Bajo la Curva” (UAC). El área bajo la curva ROC es el mejor indicador global de la precisión de una prueba ya que hace factible expresar el desempeño de una prueba mediante un número simple. Esta área es siempre mayor o igual a

0,5. El rango de valores se mueve entre 1 (discriminación perfecta) y 0,5 (no hay diferencias en la distribución de los valores de la prueba entre los 2 grupos).

Para explicar la interpretación de esta medida, [26] ponen el siguiente ejemplo: un área de 0.8 significa que un individuo seleccionado al azar del grupo de lesionados tiene un valor mayor del marcador que el de un individuo elegido al azar del grupo no-enfermo un 80% de las veces. Esto no significa que un resultado positivo ocurre con una probabilidad de 0.80, ni que un resultado positivo se asocia con la enfermedad el 80% de las veces.

7 Implementación y Experimentación del Trabajo

A continuación se presenta la implementación de los distintos modelos de clasificación basados en el modelo general propuesto en el capítulo anterior. Asimismo, se estiman los parámetros de LS-SVM con las variantes de PSO y se evalúa el rendimiento de cada uno de los modelos para el *kernel* RBF. Se implementaron tres variantes de PSO: PSO con factor de inercia lineal (sección 3.3.4.1), IPSO (sección 3.3.4.1) y DAPSO (sección 3.3.4.2).

Lo nombrado anteriormente requiere que los datos ya estén formateados para ser utilizados por cada modelo. Es por esto, que se presenta una descripción completa acerca del pre procesamiento realizado a los datos.

7.1 Pre proceso de los Datos

7.1.1 Integración de los datos

Los datos recolectados desde el año 2003 al 2009 fueron entregados en formato Excel, los cuales fueron traspasados a un motor de base de datos MySQL para su posterior uso. Para ello, se crearon las mismas entidades representadas en la Figura 5.1 . Solamente se insertaron aquellos registros que poseían la información completa de cada entidad, descartándose aquellos que poseían valores nulos. Además solamente se ingresaron a la base de datos aquellos registros correspondientes a la región de Valparaíso, debido a que el alcance de esta investigación se limita a clasificar los accidentes en dicha región del país. De esta forma, se pudieron rescatar los datos mediante simples consultas sql para luego ser guardados en Matlab.

7.1.2 Selección y codificación de los datos

Como el modelo a clasificar corresponde al estado en el cual resultan las personas (lesionadas o ilesas) involucradas en los accidentes de tránsito, se procedió a identificar aquellas características más representativas. En base a esto, se analizaron y escogieron los atributos propuestos en el trabajo [24] realizado por la Escuela de Ingeniería en Transporte de la PUCV. Para ello se rescataron de la base de datos la información con los siguientes atributos:

Tabla 7.1 Atributos seleccionados para el modelo

Número	Característica
1	Comuna
2	Urbano/Rural
3	Estado Atmosférico
4	Hora
5	Causa
6	Tipo Calzada
7	Estado Calzada
8	Condición Calzada
9	Tipo de Accidente
10	Calidad
11	Sexo
12	Edad
13	Resultado

Cabe destacar que solamente se utilizan códigos por cada atributo. Por ejemplo el atributo de calidad (10) posee 3 tipos:

- (Código 1): Conductor,
- (Código 2): Pasajero,
- (Código 3): Peatón.

De la misma forma que el atributo calidad, son codificados los atributos restantes. Otro punto a destacar corresponde al atributo Resultado (13), el cual es utilizado para extraer la etiqueta en la clasificación. Por ejemplo, el atributo resultado se divide en los siguientes tipos:

- (Código 1): Muerto
- (Código 2): Grave
- (Código 3): Menos Grave
- (Código 4): Leve
- (Código 5): Ileso

Estos atributos sirven para identificar y formar las dos clases a utilizar (personas lesionadas y personas ilesas). Ambas clases son formadas como se muestran en la Tabla 7.2:

Tabla 7.2 Formación de clases

Clase	Atributos	Etiqueta
Persona Lesionada	Grave, Menos Grave y Leve	1
Persona Ilesa	Ileso	-1

Como se aprecia en la tabla anterior las personas que resultan en estado de Muerto, no son consideradas para el modelo. Finalmente, el total de características para realizar la clasificación son doce debido a que el atributo resultado es utilizado para conformar la clase y para establecer la etiqueta.

7.1.3 Transformación de los datos

La normalización de los datos suele proporcionar mejores resultados que la introducción directa de datos crudos en los algoritmos. Es por ello que los datos son normalizados por columna. La normalización está dada por la siguiente fórmula:

$$z(i) = \left(\frac{2x(i) - (b + a)}{b - a} \right)$$

Dónde:

$x(i)$: elemento actual

b : valor máximo de la columna

a : valor mínimo de la columna

7.2 Software y Hardware Utilizado

Para la clasificación se utilizó el Toolbok LS-SVM versión 1.7 desarrollada en Matlab [27]. Esta herramienta fue desarrollada por Suykens y otros, quienes son los creadores de LS-SVM. Es de código abierto y está disponible bajo la licencia general pública GNU. Además, es una de las herramientas más robustas a nivel mundial de LS-SVM para regresión y clasificación. Por otra parte, el algoritmo de optimización por enjambres de partículas y sus variantes también fueron programados en Matlab. En cuanto a Matlab, se trabajó bajo la versión 7.10.0 de 64 bit, en Windows 7 x64 bit.

El hardware utilizado para la implementación de LS-SVM y PSO fue el siguiente:

- Procesador Intel Core 2 Duo
- Velocidad del Procesador 2.2 GHz
- Memoria RAM 4 Gb

7.3 Clasificación LS-SVM con PSO

A continuación se presenta la clasificación de los distintos modelos propuestos e implementados, en base a las variantes del algoritmo de optimización por enjambre de partículas. Para ello, se escogió y utilizó el tipo de *kernel* Función de Base Radial (RBF), ya que según la literatura y trabajos relacionados presenta mejores resultados que los demás. Además, las variantes de PSO implementadas fueron las siguientes:

- PSO con Factor de Inercia Lineal
- IPSO (*Improved Particle Swarm Optimization*)
- DAPSO (*Dynamic Adaptation Particle Swarm Optimization*)

Por otra parte, la elección de la función de costo o *fitness* en PSO, juega un papel fundamental en la generalización del modelo. En base a esto, se utilizan dos tipos de *fitness*:

- Costo de elementos mal clasificados utilizando validación cruzada para 10 subconjuntos.
- Costo de elementos mal clasificados utilizando la función del error absoluto medio.

El total de la población con la que se cuenta para el estudio es alrededor de 70.000 datos, de los cuales en primera instancia se extraen 3000 para cada modelo de PSO. Estos se dividen en 2.000 para la etapa de entrenamiento y los 1000 restante para testing. En segunda instancia se extraen distintas muestras para volver a testear el modelo y así comprobar el comportamiento del clasificador con distintos tamaños de muestra, es decir, advertir que tan preciso es el clasificador. Es de importancia recalcar que las muestras se toman aleatoriamente del conjunto de datos, comprendidos entre los años 2003 al 2009.

Por otra parte, el *kernel* del tipo RBF requiere solo un parámetro a ajustar, este corresponde a σ^2 . También se debe ajustar el parámetro C que controla el *trade-off* entre la maximización del margen y la minimización del error de entrenamiento. De esta forma, para cualquier variante de PSO la representación de la partícula para este tipo de *kernel*, es representada por el siguiente vector: $[C, \sigma^2]$.

Los valores de los parámetros de inicialización de PSO con LS-SVM comunes para las 3 variantes utilizadas, se presentan en la siguiente tabla:

Tabla 7.3 Parámetros de entrada PSO - kernel RBF

Parámetros de Entrada	Valores
Tamaño de la Población	24
Rango posición partícula C	[1–5000]
Rango posición partícula σ^2	[0–20]
Componente cognitiva	2
Componente social	1.5
Velocidad máxima	2
Número máximo de iteraciones	500 iter (<i>fitness</i> Costo Validación Cruzada) 100 iter (<i>fitness</i> Error Absoluto Medio)

Como se aprecia en la Tabla 7.3 el número máximo de iteraciones corresponde al criterio de término o parada del algoritmo.

7.3.1 PSO con Factor de Inercia Lineal

EL factor de inercia lineal w se caracteriza por ir decreciendo en un intervalo preestablecido, a medida que aumentan las iteraciones. El intervalo escogido para dicho factor es entre [0.9-0.4].

7.3.1.1 Fitness Costo Validación Cruzada

En primera instancia se utilizó la muestra de 3.000, los que se dividieron en 2.000 para training y 1.000 para testing. El tiempo total de estimación de parámetros con esta variante fue exactamente de 32.247 segundos, equivalentes a 8,95 hrs aprox. Los parámetros óptimos obtenidos fueron $C = 167.8857$ y $\sigma^2 = 7.9708$, en tanto, el mejor *fitness* fue de: 0.1795. El resumen general para la etapa de testing se resume en la siguiente tabla.

Tabla 7.4 Resultados PSO F. Inercia Lineal, fitness costo validación cruzada

ETAPA	Exactitud	Sensibilidad	Especificidad	VPP	VPN	Mal Clasificados
Training	90,15	88,7	92,22	94,22	85,1	197
Testing	85,9	85	87,39	91,9	77,61	141

Como se muestra en la tabla anterior, esta variante obtuvo un 85,9% de exactitud en la etapa de testing, esto quiere decir que el modelo ha clasificado de manera correcta en un 85,9% las personas que resultan en el estado lesionado o ileso.

Por otra parte, la sensibilidad obtenida fue de un 85%, cifra correspondiente al porcentaje de personas bien clasificadas en el estado lesionado. Asimismo, la especificidad resultó ser un 87,39% correspondiendo al porcentaje de personas bien clasificadas en el estado ileso. En cuanto a la seguridad del resultado, se tiene un valor predictivo positivo de 91,8%, este significa que un 91,8% de las personas detectadas como lesionadas están realmente lesionadas, en tanto, el valor predictivo negativo fue de 72,8%, este significa que un 72,8% de las personas detectadas como ilesas están realmente ilesas.

Finalmente se muestra la variación del costo en el entrenamiento (*fitness*) por cada iteración:

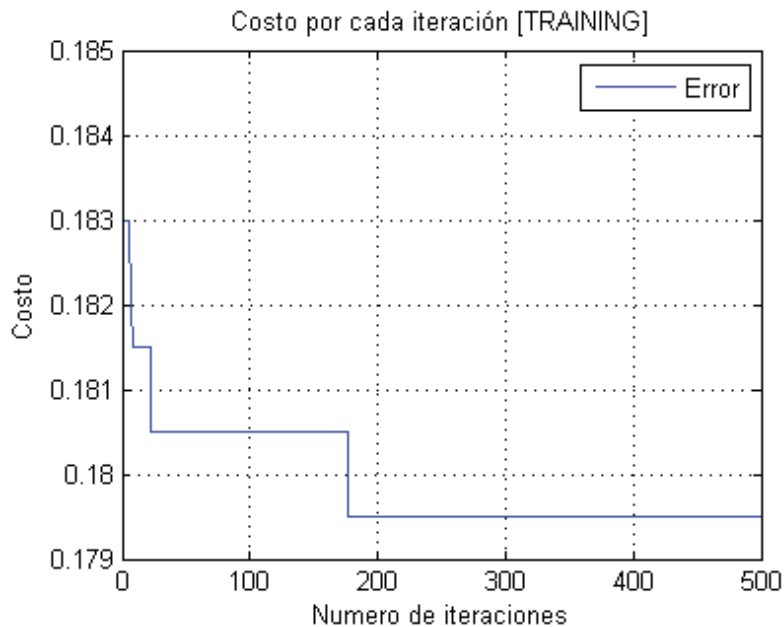


Figura 7.1 Costo vs Iteración, PSO F. Inercia Lineal, fitness costo validación cruzada

Se aprecia que el costo disminuye gradualmente hasta la iteración 25. Luego se mantiene hasta la iteración 180 en donde el costo disminuye bruscamente hasta el valor de 0.1775. Este se mantiene hasta la iteración 500.

En segunda instancia se procedió a testear el modelo con distintos tamaños de muestras a partir del modelo entrenado anteriormente, para comprobar el comportamiento del clasificador. En la siguiente tabla se aprecian los resultados.

Tabla 7.5 Resultados para varias muestras PSO F. Inercia Lineal, fitness costo validación cruzada

Tamaño Muestra	Exactitud	Sensibilidad	Especificidad	PPV	NPV
1000	85,9	85,0	87,4	91,9	72,8
2000	84,9	80,2	91,3	92,7	76,9
3000	83,7	79,2	90,1	91,9	75,3
4000	81,8	76,9	88,5	90,2	73,6
5000	80,6	76,9	85,8	88,2	72,7
6000	80,2	77,2	84,4	87,4	72,5
7000	81,3	77,0	86,4	87,0	76,0
8000	81,4	77,8	85,5	86,4	76,6
9000	81,8	78,1	86,1	86,6	77,5
10000	81,8	77,5	86,6	86,6	77,5
15000	81,1	76,2	86,7	86,7	76,0
20000	80,1	75,5	85,4	85,6	75,2
Promedio	82,0	78,1	87,0	88,4	75,2

A través de la tabla anterior se aprecia que al variar el tamaño de la muestra para testing, la exactitud del clasificador se mantiene entre el rango 80.1% y 85.9%, obteniendo como promedio un 82%. Asimismo ocurre con las demás métricas, las cuales se mantienen dentro de un rango aceptable.

7.3.1.2 Fitness Error Absoluto Medio

El tiempo total de estimación de parámetros con esta variante para un total de 100 iteraciones fue exactamente de 8.069 segundos, equivalentes a 2,24 hrs aprox. Los parámetros óptimos obtenidos fueron $C = 4531.7848$ y $\sigma^2 = 0.00039067$, en tanto, el mejor *fitness* fue de: 0.00021178. Los resultados se detallan en la siguiente tabla

Tabla 7.6 Resultado PSO F. Inercia Lineal, fitness error absoluto medio

ETAPA	Exactitud	Sensibilidad	Especificidad	VPP	VPN	Mal Clasificados
Training	100	100	100	100	100	0
Testing	94,6	99,2	86,9	92,7	98,5	54

Esta variante obtuvo un 94,6% de exactitud en la etapa de testing (1.000 datos), esto quiere decir que el modelo ha clasificado de manera correcta en un 94,6% las personas que resultan en el estado lesionado o ileso.

Por otra parte, la sensibilidad obtenida fue de un 86,9%, cifra correspondiente al porcentaje de personas bien clasificadas en el estado lesionado. Asimismo, la especificidad resultó ser un 92,7% correspondiendo al porcentaje de personas bien clasificadas en el estado ileso. En cuanto a la seguridad del resultado, se tiene un valor predictivo positivo de 92,7%, este significa que un 92,7% de las personas detectadas como lesionadas están realmente

lesionadas, en tanto, el valor predictivo negativo fue de 98,5%, este significa que un 98,5% de las personas detectadas como ilesas están realmente ilesas.

Finalmente se muestra la variación del costo en el entrenamiento (fitness) por cada iteración:

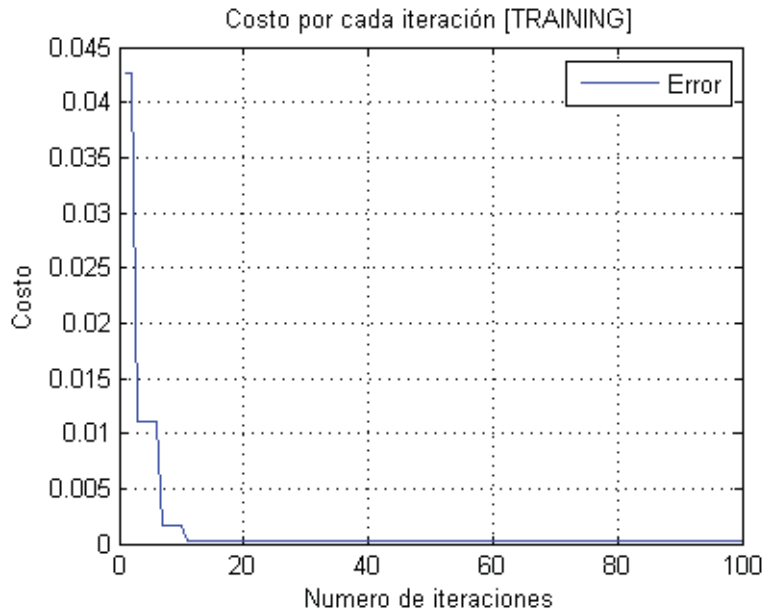


Figura 7.2 Costo vs Iteración, PSO F. Inercia Lineal, fitness error absoluto medio

Se aprecia que el costo disminuye gradualmente hasta la iteración 12, llegando a un valor cercano a cero. Aunque no se aprecie detalladamente en el gráfico después de la iteración 12, igualmente el costo sigue disminuyendo tendiendo a un valor cercano a 0.

En segunda instancia se procedió a testear el modelo con distintos tamaños de muestras a partir del modelo entrenado anteriormente, para comprobar el comportamiento del clasificador. En la siguiente tabla se aprecian los resultados.

Tabla 7.7 Resultados testing para varias muestras PSO F. Inercia Lineal, fitness error absoluto medio

Tamaño Muestra	Exactitud	Sensibilidad	Especificidad	VPP	VPN
1000	94,6	99,2	86,9	92,7	98,5
2000	64,1	98,1	17,0	62,0	86,7
3000	63,0	98,4	12,7	61,5	84,4
4000	61,8	98,4	11,3	60,4	83,8
5000	61,5	98,6	9,9	60,3	83,1
6000	61,1	98,7	8,5	60,2	81,9
7000	57,7	98,8	8,9	56,2	86,4
8000	57,3	98,9	8,3	55,9	86,4
9000	56,7	98,9	8,5	55,3	87,3
10000	55,5	99,0	8,7	53,9	88,9
15000	56,9	99,0	8,7	55,4	88,8
20000	56,6	99,0	7,9	55,3	87,2
Promedio	62,2	98,7	16,4	60,8	87,0

A través de la tabla anterior se aprecia que al variar el tamaño de la muestra para testing, la exactitud del clasificador se mantiene entre el rango 55.5% y 94.6%, obteniendo como promedio un 62.2%. Claramente se está presente a un modelo sobre ajustado, ya que al aumentar el tamaño de muestra el porcentaje de clasificación disminuye considerablemente. Además, viendo el promedio de especificidad solo se obtiene un 16,4% alcanzando un mínimo de 7,9% para un tamaño de 20.000 datos de testing. En resumen se está en presencia de un pésimo clasificador.

7.3.1.3 Comparación entre Fitness

A modo de resumen se presenta un gráfico comparativo de la exactitud entre los *fitness* utilizados, para distintos tamaños de muestra.

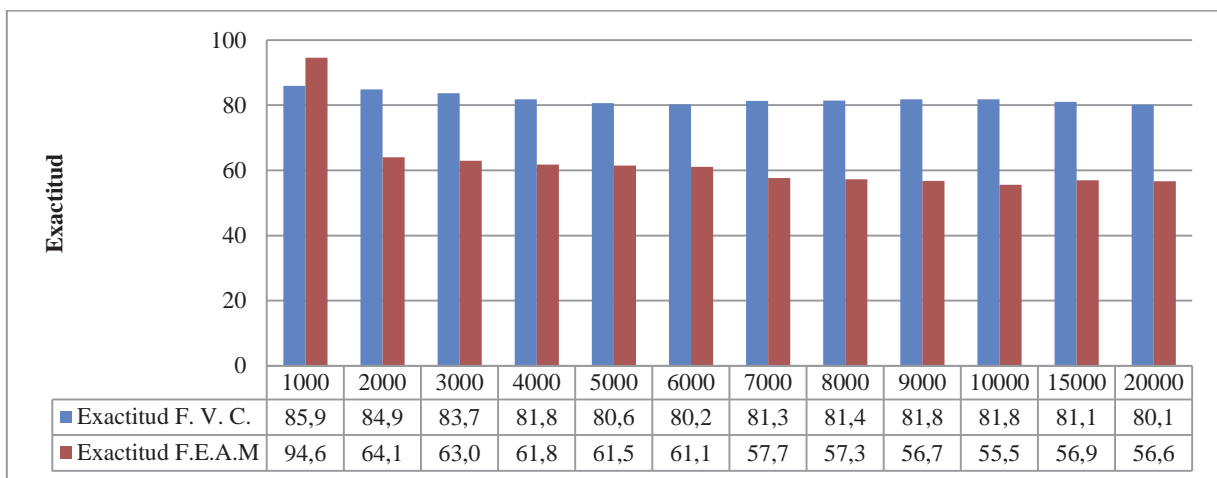


Figura 7.3 Gráfico Comparativo Exactitud entre Fitness, PSO F. Inercia Lineal

Claramente se aprecia que utilizando como *fitness* el costo de la validación cruzada para 10 subconjuntos, la exactitud se mantiene en un rango bien acotado al aumentar el tamaño de la muestra. Es así que en promedio se obtiene un 82% de exactitud en la clasificación. Sin embargo, el *fitness* error absoluto medio obtiene un pésimo clasificador ya que al variar el tamaño de la muestra la exactitud disminuye desde un 94,6% a un 55,5%. Esto quiere decir que el modelo está sobre ajustado. Por lo tanto, se comprueba que el modelo con mayor generalización es aquel que utiliza el *fitness* de costo de validación cruzada.

7.3.2 IPSO (Improved Particle Swarm Optimization)

Esta variante posee 2 parámetros de entrada. Estos son inicializados como siguen:

- $W_{inicial} = 0.9$
- $\mu = 1.0002$

7.3.2.1 Fitness Costo Validación Cruzada

En primera instancia se utilizó la muestra de 3.000, los que se dividieron en 2.000 para training y 1.000 para testing. El tiempo total de estimación de parámetros con esta variante fue exactamente de 40.062 segundos, equivalentes a 11,12 hrs aprox. Los parámetros óptimos obtenidos fueron $C = 457.92$ y $\sigma^2 = 13.85$, en tanto, el mejor *fitness* fue de: 0.178. El resumen general para la etapa de testing se resume en la siguiente tabla.

Tabla 7.8 Resultados IPSO, fitness costo validación cruzada

ETAPA	Exactitud	Sensibilidad	Especificidad	VPP	VPN	Mal Clasificados
Training	86,9	85,21	89,3	91,93	80,85	262
Testing	84,5	81,8	89	92,59	74,43	155

Como se muestra en la tabla anterior, esta variante obtuvo un 84,5% de exactitud en la etapa de testing, esto quiere decir que el modelo ha clasificado de manera correcta en un 84,5% las personas que resultan en el estado lesionado o ileso.

Por otra parte, la sensibilidad obtenida fue de un 81,8%, cifra correspondiente al porcentaje de personas bien clasificadas en el estado lesionado. Asimismo, la especificidad resultó ser un 89% correspondiendo al porcentaje de personas bien clasificadas en el estado ileso. En cuanto a la seguridad del resultado, se tiene un valor predictivo positivo de 92,59%, este significa que un 92,59% de las personas detectadas como lesionadas están realmente lesionadas, en tanto, el valor predictivo negativo fue de 74,43%, este significa que un 74,43% de las personas detectadas como ilesas están realmente ilesas.

Finalmente se muestra la variación del costo en el entrenamiento (*fitness*) por cada iteración:

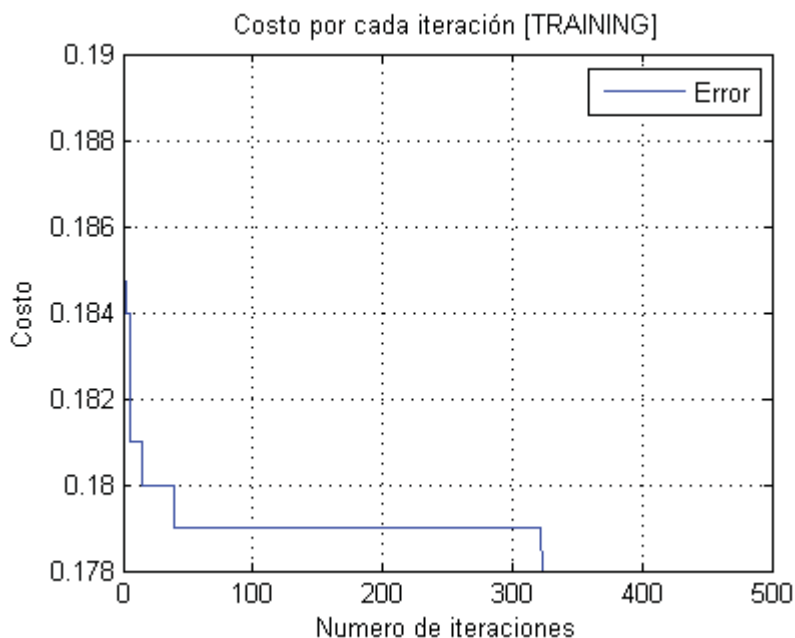


Figura 7.4 Costo vs Iteración, IPSO, fitness costo validación cruzada

Se aprecia que el costo disminuye gradualmente hasta cerca de la iteración 40. Luego se mantiene pasada la iteración 300 en donde el costo disminuye hasta el valor de 0.178, el cual se mantiene hasta la iteración 500.

En segunda instancia se procedió a testear el modelo con distintos tamaños de muestras a partir del modelo entrenado anteriormente, para comprobar el comportamiento del clasificador. En la siguiente tabla se aprecian los resultados.

Tabla 7.9 Resultados para varias muestras IPSO, fitness costo validación cruzada

Tamaño Muestra	Exactitud	Sensibilidad	Especificidad	VPP	VPN
1000	84,5	81,8	89,0	92,6	74,4
2000	83,6	83,5	83,7	87,9	78,1
3000	83,6	83,5	83,7	87,9	78,1
4000	81,6	82,1	80,9	85,6	76,6
5000	79,7	82,2	76,3	82,8	75,5
6000	79,0	83,2	73,0	81,2	75,7
7000	79,8	83,2	75,8	80,3	79,2
8000	79,3	83,4	74,6	79,4	79,2
9000	79,7	83,6	75,3	79,4	80,0
10000	79,6	83,5	75,2	79,0	80,3
15000	79,1	82,2	75,4	79,6	78,5
20000	79,9	82,9	76,7	79,2	80,7
Promedio	80,4	83,0	77,3	82,0	78,4

A través de la tabla anterior se aprecia que al variar el tamaño de la muestra para testing, la exactitud del clasificador se mantiene entre el rango 79.1% y 84.5%, obteniendo como promedio un 80,4%. Asimismo ocurre con las demás métricas, las cuales se mantienen dentro de un rango aceptable.

7.3.2.2 Fitness Error Absoluto Medio

El tiempo total de estimación de parámetros con esta variante para un total de 100 iteraciones fue exactamente de 8.069 segundos, equivalentes a 2,24 hrs aprox. Los parámetros óptimos obtenidos fueron $C = 4531.7848$ y $\sigma^2 = 0.0002975$, en tanto, el mejor *fitness* fue de: 0.00021178. Los resultados se detallan en la siguiente tabla

Tabla 7.10 Resultados IPSO, fitness error absoluto medio

ETAPA	Exactitud	Sensibilidad	Especificidad	VPP	VPN	Mal Clasificados
Training	100	100	100	100	100	0
Testing	93,8	99,8	83,6	91,1	99,6	62

Esta variante obtuvo un 93,8% de exactitud en la etapa de testing (1.000 datos), esto quiere decir que el modelo ha clasificado de manera correcta en un 93,8% las personas que resultan en el estado lesionado o ileso.

Por otra parte, la sensibilidad obtenida fue de un 99,8%, cifra correspondiente al porcentaje de personas bien clasificadas en el estado lesionado. Asimismo, la especificidad resultó ser un 83,6% correspondiendo al porcentaje de personas bien clasificadas en el estado ileso. Finalmente se muestra la variación del costo en el entrenamiento (fitness) por cada iteración.

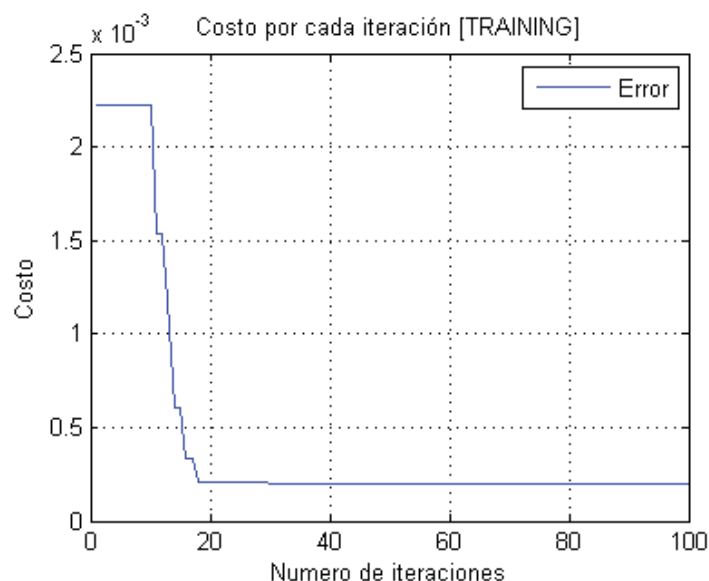


Figura 7.5 Costo vs Iteración, IPSO, fitness error absoluto medio

En segunda instancia se procedió a testear el modelo con distintos tamaños de muestras a partir del modelo entrenado anteriormente, para comprobar el comportamiento del clasificador. En la siguiente tabla se aprecian los resultados.

Tabla 7.11 Resultados testing para varias muestras IPSO, fitness error absoluto medio

Tamaño Muestra	Exactitud	Sensibilidad	Especificidad	VPP	VPN
1000	93,8	99,8	83,6	91,1	99,7
2000	60,0	99,7	5,1	59,2	93,5
3000	59,9	99,7	3,4	59,5	89,4
4000	58,8	99,7	2,6	58,5	84,6
5000	59,0	99,6	2,5	58,7	82,8
6000	59,2	99,7	2,4	58,9	83,6
7000	55,1	99,7	2,2	54,7	85,5
8000	54,9	99,7	2,2	54,5	85,1
9000	54,3	99,7	2,4	53,9	88,5
10000	53,6	99,7	2,1	53,2	87,1
15000	54,3	99,7	2,3	53,9	87,9
20000	54,3	99,7	2,0	53,9	86,3
Promedio	59,8	99,7	9,4	59,2	87,8

A través de la tabla anterior se aprecia que al variar el tamaño de la muestra para validación, la exactitud del clasificador se mantiene entre el rango 53.6% y 93.8%, obteniendo como promedio un 59,8%. Al igual que el modelo anterior (PSO con F. Inercia Lineal) se está presente a un modelo sobre ajustado, ya que al aumentar el tamaño de muestra el porcentaje de clasificación disminuye considerablemente. Además, viendo el promedio de especificidad solo se obtiene un 9,4% alcanzando un mínimo de 2% para un tamaño de 20.000 de testing. En resumen se está en presencia de un pésimo clasificador.

7.3.2.3 Comparación entre Fitness

A modo de resumen se presenta un gráfico comparativo de la exactitud entre los *fitness* utilizados, para distintos tamaños de muestra.

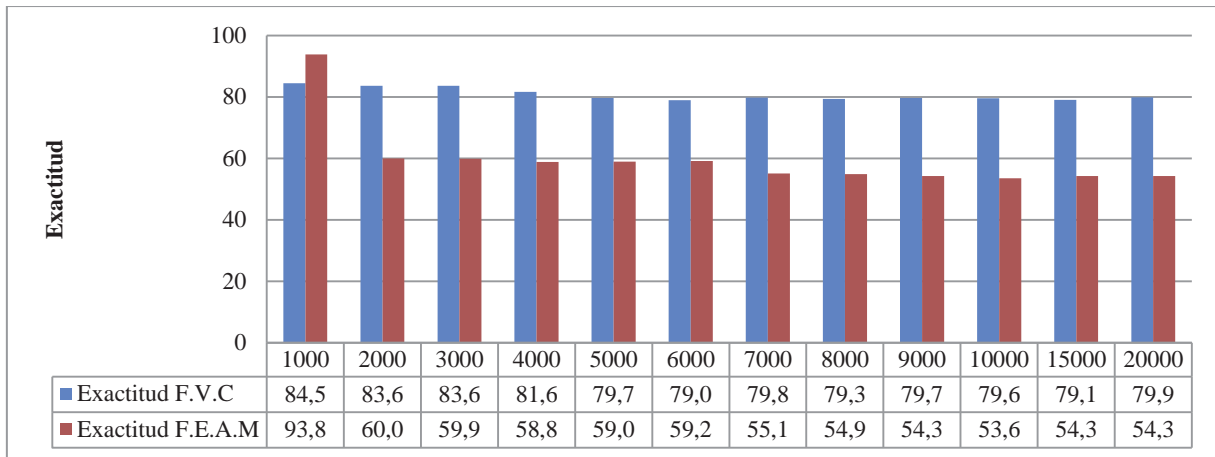


Figura 7.6 Gráfico Comparativo Exactitud entre Fitness, IPSO

Claramente se aprecia que utilizando como fitness el costo de la validación cruzada para 10 subconjuntos, la exactitud se mantiene en un rango bien acotado al aumentar el tamaño de la muestra. Es así que en promedio se obtiene un 80,4% de exactitud en la clasificación. Sin embargo, el fitness error absoluto medio obtiene un pésimo clasificador ya al variar el tamaño de la muestra la exactitud disminuye desde un 93,8% a un 53,6%. Esto quiere decir que el modelo está sobre ajustado. Por lo tanto, se comprueba que el modelo con mayor generalización es aquel que utiliza el fitness de costo de validación cruzada.

7.3.3 PSO con Adaptación Dinámica (DAPSO)

Esta variante posee 3 parámetros propios de entrada. Estos son inicializados como siguen:

- $W_{inicial} = 0,9$
- $\alpha = 0,4$
- $\beta = 0,8$

7.3.3.1 Fitness Costo Validación Cruzada

En primera instancia se utilizó la muestra de 3.000, los que se dividieron en 2.000 para training y 1.000 para testing. El tiempo total de estimación de parámetros con esta variante fue exactamente de 34.442 segundos, equivalentes a 9,56 hrs aprox. Los parámetros óptimos obtenidos fueron $C = 272.8791$ y $\sigma^2 = 15.4345$, en tanto, el mejor *fitness* fue de: 0.1755. El resumen general para la etapa de testing se resume en la siguiente tabla.

Tabla 7.12 Resultados DAPSO, fitness costo validación cruzada

ETAPA	Exactitud	Sensibilidad	Especificidad	VPP	VPN	Mal Clasificados
Training	88,2	82,3	94,1	93,31	84,16	236
Testing	84,8	80,4	90,8	92,4	77,1	160

Como se muestra en la tabla anterior, esta variante obtuvo un 84,8% de exactitud en la etapa de testing, esto quiere decir que el modelo ha clasificado de manera correcta en un 84,8% las personas que resultan en el estado lesionado o ileso.

Por otra parte, la sensibilidad obtenida fue de un 80,4%, cifra correspondiente al porcentaje de personas bien clasificadas en el estado lesionado. Asimismo, la especificidad resultó ser un 90,8% correspondiendo al porcentaje de personas bien clasificadas en el estado ileso. Además, se tiene un valor predictivo positivo de 92,4%, este significa que un 92,4% de las personas detectadas como lesionadas están realmente lesionadas, en tanto, el valor predictivo negativo fue de 77,1%, este significa que un 77,1% de las personas detectadas como ilesas están realmente ilesas.

Finalmente se muestra la variación del costo en el entrenamiento (*fitness*) por cada iteración:

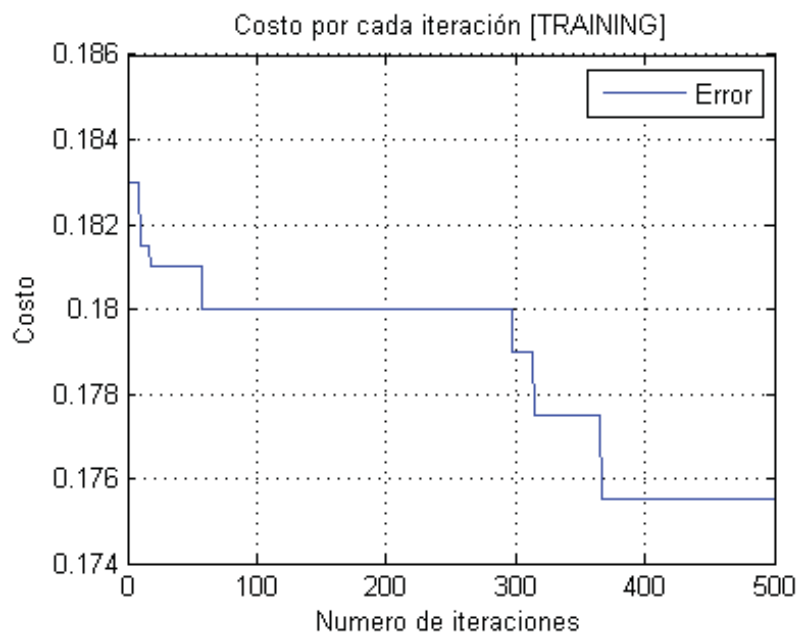


Figura 7.7 Costo vs Iteración, DAPSO, fitness costo validación cruzada

Se aprecia que el costo disminuye gradualmente hasta cerca de la iteración 60. Luego se mantiene hasta la iteración 300, en donde el costo disminuye gradualmente cercano a la iteración 380 en donde obtiene el valor mínimo de 0.1775, y se mantiene hasta la iteración 500.

En segunda instancia se procedió a testear el modelo con distintos tamaños de muestras a partir del modelo entrenado anteriormente, para comprobar el comportamiento del clasificador. En la siguiente tabla se aprecian los resultados.

Tabla 7.13 Resultados para varias muestras DAPSO, fitness costo validación cruzada

Tamaño Muestra	Exactitud	Sensibilidad	Especificidad	VPP	VPN
1000	84,8	80,4	90,8	92,4	77,1
2000	84,0	77,7	94,6	96,1	71,6
3000	84,2	79,7	90,6	92,4	75,8
4000	82,0	77,4	88,4	90,1	73,9
5000	80,4	77,0	85,0	87,7	72,7
6000	81,8	76,9	87,0	86,5	77,8
7000	81,0	77,2	85,5	86,3	76,0
8000	81,1	78,0	84,7	85,7	76,6
9000	81,6	78,2	85,5	86,0	77,4
10000	81,5	77,4	86,0	86,0	77,3
15000	81,2	77,0	85,9	86,2	76,6
20000	80,3	75,9	85,2	85,3	75,7
Promedio	81,7	77,5	87,1	88,0	75,6

A través de la tabla anterior se aprecia que al variar el tamaño de la muestra para testing, la exactitud del clasificador se mantiene entre el rango 80.3% y 84.8%, obteniendo como promedio un 81,7%. Asimismo ocurre con las demás métricas, las cuales se mantienen dentro de un rango aceptable.

7.3.3.2 Fitness Error Absoluto Medio

El tiempo total de estimación de parámetros con esta variante para un total de 100 iteraciones fue exactamente de 9.537 segundos, equivalentes a 2,6 hrs aprox. Los parámetros óptimos obtenidos fueron $C = 4512.3847$ y $\sigma^2 = 0.0003547$, en tanto, el mejor *fitness* fue de: 0.00021271. Los resultados se detallan en la siguiente tabla

Tabla 7.14 Resultados DAPSO, fitness error absoluto medio

ETAPA	Exactitud	Sensibilidad	Especificidad	VPP	VPN	Mal Clasificados
Training	100	100	100	100	100	0
Testing	88,8	99,52	70,77	85,13	98,87	112

Esta variante obtuvo un 88,8% de exactitud en la etapa de testing (1.000 datos), esto quiere decir que el modelo ha clasificado de manera correcta en un 88,8% las personas que resultan en el estado lesionado o ileso.

Por otra parte, la sensibilidad obtenida fue de un 99,52%, cifra correspondiente al porcentaje de personas bien clasificadas en el estado lesionado. Asimismo, la especificidad resultó ser un 70,77% correspondiendo al porcentaje de personas bien clasificadas en el estado ileso. Además, se tiene un valor predictivo positivo de 85,13%, este significa que un 85,13% de las personas detectadas como lesionadas están realmente lesionadas, en tanto, el valor

predictivo negativo fue de 98,87%, este significa que un 98,87% de las personas detectadas como ilesas están realmente ilesas. Finalmente se muestra la variación del costo en el entrenamiento (fitness) por cada iteración:

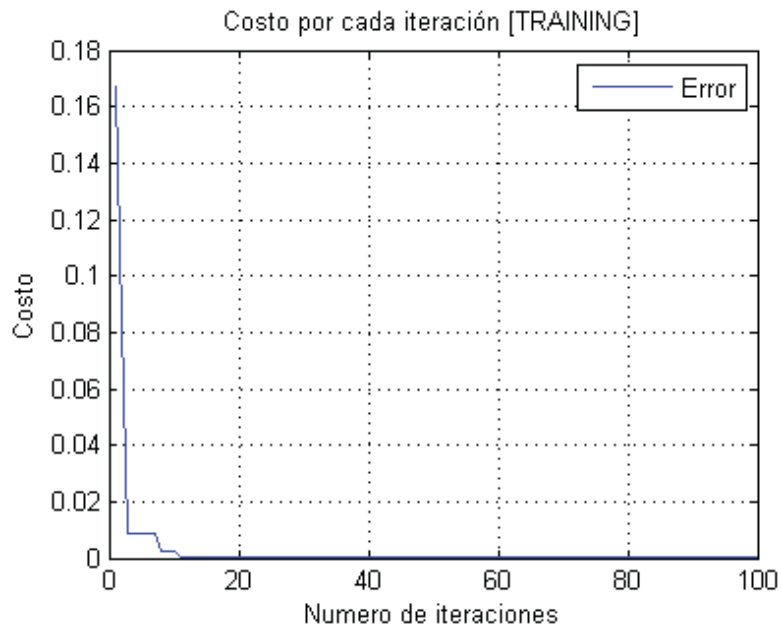


Figura 7.8 Costo vs Iteración, DAPSO, fitness error absoluto medio

Se aprecia que el costo disminuye gradualmente hasta la iteración 12. Aunque no se aprecie detalladamente en el grafico después de la iteración 12, igualmente el costo sigue disminuyendo tendiendo a un valor cercano a 0.

En segunda instancia se procedió a testear el modelo con distintos tamaños de muestras a partir del modelo entrenado anteriormente, para comprobar el comportamiento del clasificador. En la siguiente tabla se aprecian los resultados.

Tabla 7.15 Resultados testing para varias muestras DAPSO, fitness error absoluto medio

Tamaño Muestra	Exactitud	Sensibilidad	Especificidad	PPV	NPV
1000	88,8	99,5	7,1	85,1	98,9
2000	60,7	99,1	7,6	59,7	86,5
3000	60,6	99,3	5,5	59,9	85,0
4000	59,3	99,4	4,1	58,8	82,1
5000	59,4	99,4	36,8	58,9	81,9
6000	59,4	99,3	3,4	59,0	78,5
7000	55,4	99,4	3,2	54,9	82,9
8000	55,2	99,5	3,0	54,7	83,8
9000	54,6	99,6	3,2	54,0	86,9
10000	53,9	99,6	3,0	53,3	87,0
15000	54,7	99,7	3,2	54,1	88,9
20000	54,7	99,6	2,9	54,1	87,7
Promedio	59,7	99,5	6,9	58,9	85,9

A través de la tabla anterior se aprecia que al variar el tamaño de la muestra para validación, la exactitud del clasificador varía entre el rango 53.9% y 88.8%, obteniendo como promedio un 59.7%. Al igual que los modelos anteriores (PSO con F. Inercia Lineal, IPSO) se está presente a un modelo sobre ajustado, ya que al aumentar el tamaño de muestra el porcentaje de clasificación disminuye considerablemente. Además, el porcentaje promedio de especificidad es demasiado bajo, este no alcanza a superar un 7%. En resumen se está en presencia de un pésimo clasificador.

7.3.3.3 Comparación entre Fitness

A modo de resumen se presenta un gráfico comparativo de la exactitud entre los *fitness* utilizados, para distintos tamaños de muestra.

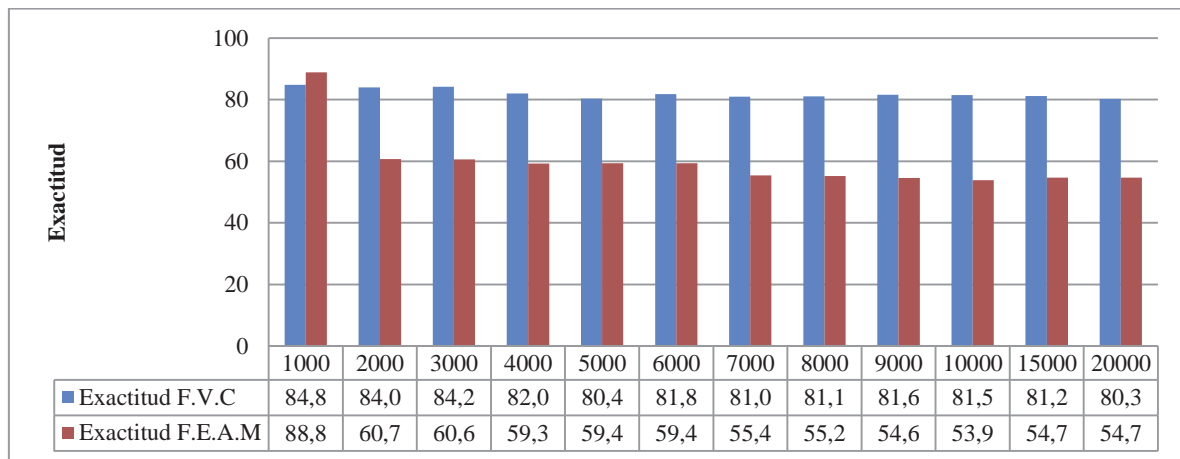


Figura 7.9 Gráfico Comparativo Exactitud entre Fitness, DAPSO

Claramente se aprecia que utilizando como *fitness* el costo de la validación cruzada para 10 subconjuntos, la exactitud se mantiene en un rango bien acotado al aumentar el tamaño de la muestra. Es así que en promedio se obtiene un 82% de exactitud en la clasificación. Sin embargo, el *fitness* error absoluto medio obtiene un pésimo clasificador ya al variar el tamaño de la muestra la exactitud disminuye desde un 94,6% a un 55,5%. Esto quiere decir que el modelo está sobre ajustado. Por lo tanto, se comprueba que el modelo con mayor generalización es aquel que utiliza el *fitness* de costo de validación cruzada.

8 Análisis de Resultados

En este capítulo se busca contrastar los modelos implementados en el capítulo anterior, en base a los resultados obtenidos por cada modelo. Es por esto, que en primer lugar se contrastan los modelos a partir de las tres variantes de PSO utilizadas, eligiendo la mejor entre ellas. En segundo lugar, el mejor modelo obtenido es comparado con los trabajos relacionados descritos en [24] y [22]. Cabe destacar que el *fitness* error absoluto medio no es considerado, ya que como se vio en el capítulo anterior, este presentó un muy bajo desempeño en cada variante al variar el tamaño de muestra para testing, por ende es descartado y solo se utilizan los experimentos realizados con el *fitness* de validación cruzada para 10 subconjuntos.

8.1 Comparación entre variantes de PSO

Aunque lo primordial es analizar la etapa de testing para comprobar el rendimiento del modelo, es conveniente tratar de analizar la etapa de entrenamiento, ya que cada variante de PSO trabaja directamente con la fase de training. Cabe recordar que dicha etapa fue entrenada con una cantidad de 2.000 datos. La siguiente tabla resume dichos resultados:

Tabla 8.1 Comparación training LS-SVM con variantes PSO

Variantes	Exactitud	Sensibilidad	Especificidad	VPP	VPN	Mal Clasif.	Tiempo Entrenamiento
PSO F.I.L	90,15	88,7	92,22	94,22	85,1	197	8,96 hr
IPSO	86,9	85,21	89,3	91,93	80,85	262	11,13 hr
DAPSO	88,2	82,3	94,1	93,31	84,16	236	9,57 hr

La variante con menor tiempo de entrenamiento fue PSO con factor de inercia lineal con 8,96 hr aprox. En tanto, IPSO fue la que tardó más con 11,13 hr. Aunque contextualizando al problema tratado en este trabajo el tiempo no es un factor crítico, igualmente hay considerar que es demasiado alto para cada modelo. Esto se debe a que la técnica LS-SVM si bien trabaja con ecuaciones lineales, igualmente el costo de procesamiento es bastante alto. Otro aspecto importante a destacar corresponde al total elementos mal clasificados, en donde PSO con factor de inercia lineal obtuvo la menor tasa de mal clasificados, lo que se traduce en un 90,15% de exactitud.

Pasando al análisis de la etapa de testing, en primer lugar se realiza una comparación en cuanto a la validez del modelo. Para ello, se presenta la exactitud, sensibilidad y especificidad de los tres modelos implementados. Además, es importante recalcar que se presentan dichos resultados para distintos tamaños de muestra, es decir, que a la misma vez se comprueba la reproductividad de los resultados.

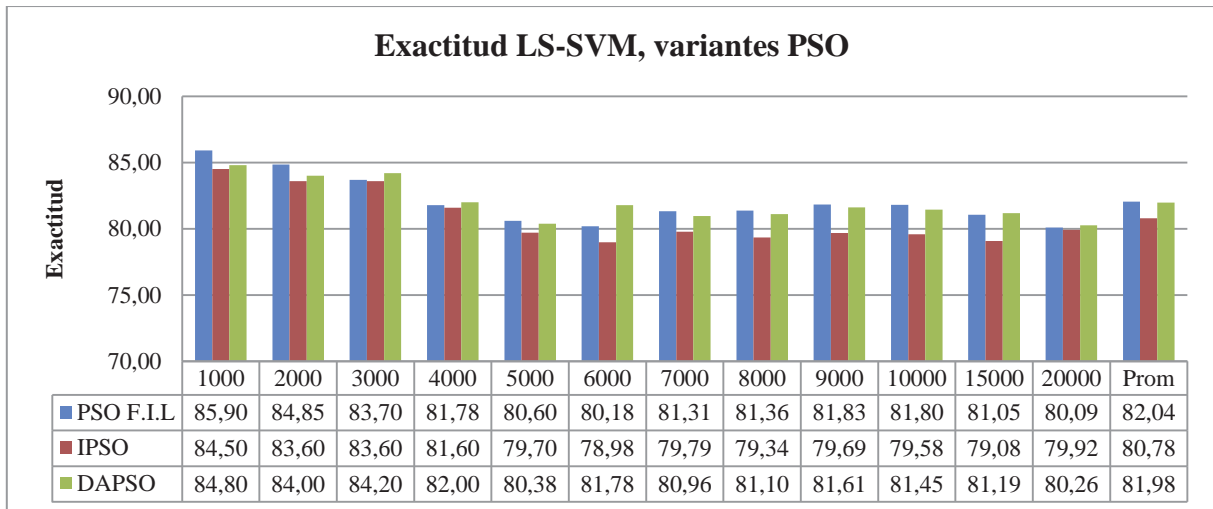


Figura 8.1 Gráfico Exactitud LS-SVM, variantes de PSO

Al comparar la exactitud para los tres modelos se aprecia que la variante de PSO con factor inercia lineal obtuvo un mayor porcentaje en promedio, sin embargo, para algunos tamaños de muestra como 3.000, 4.000 y 15.000 fue superada por PSO con adaptación dinámica. Además, sólo supero a DAPSO en un 0,06% en el promedio de exactitud lo que significa que no hubo una gran diferenciación entre estas dos variantes. Por otra parte, IPSO obtuvo la puntuación más baja en exactitud llegando a promediar un 80,78%, y en diferencia con la mejor variante (PSO con factor de inercial) fue de un 1,26%. También, se aprecia el buen grado de generalización del modelo, ya que al aumentar el tamaño de muestra el porcentaje de exactitud se mantiene casi regular para las dos mejores variantes.

En la misma línea de validez del modelo, se presentan los gráficos comparativos de sensibilidad y especificidad de las tres variantes:

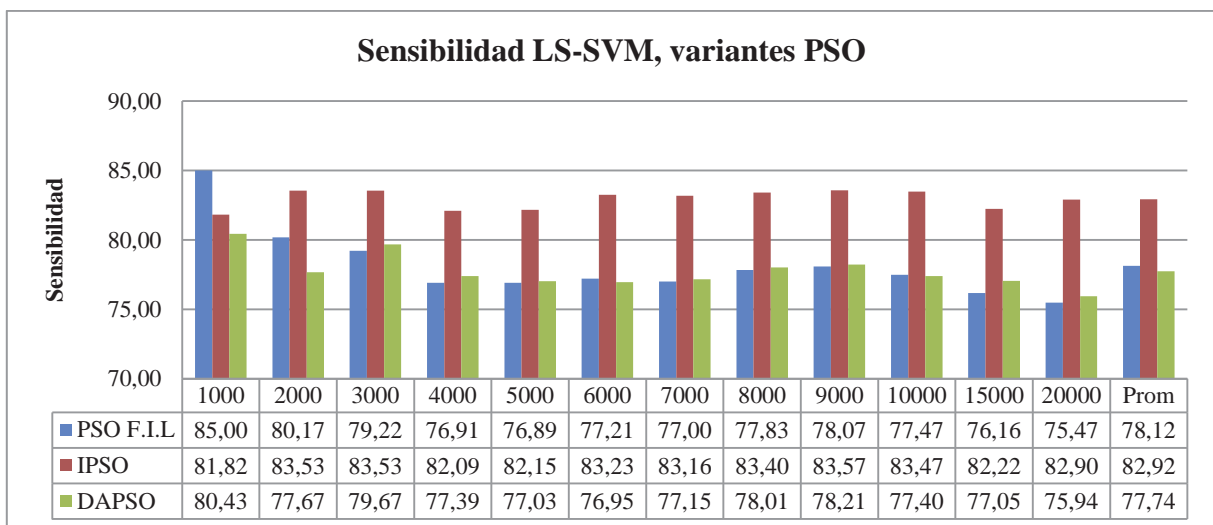


Figura 8.2 Gráfico Sensibilidad LS-SVM, variantes de PSO

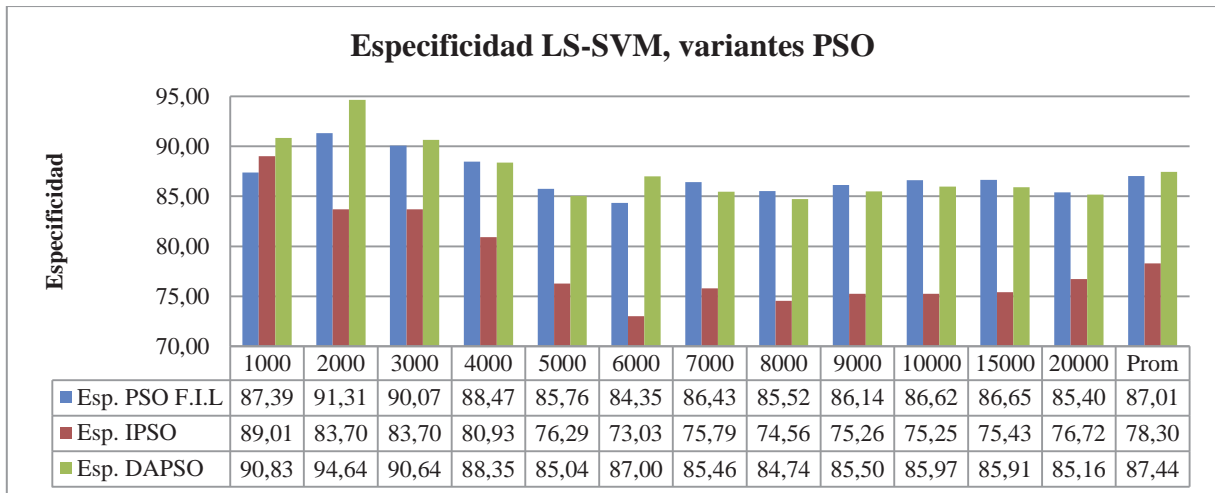


Figura 8.3 Gráfico Especificidad LS-SVM, variantes de PSO

Mediante el gráfico comparativo de sensibilidad se aprecia que IPSO presenta el mejor promedio con un 82,92% superando en 4,8% a PSO con factor de inercia lineal y en 5,18% a DAPSO. En cuanto a la especificidad DAPSO y PSO con factor de inercia lineal obtuvieron sobre un 87% superando en casi 9% a IPSO. Además, en ambas métricas al variar el tamaño de muestra para validación, los porcentajes se mantuvieron en un rango aceptable, lo que comprueba la buena precisión del modelo.

La sensibilidad y especificidad no responden a: si el resultado es lesionado, ¿cuál es la probabilidad de que el individuo esté realmente lesionado?, o si el resultado es ileso, ¿cuál es la probabilidad de que de que el individuo esté realmente ileso?. Para responder estas preguntas se utilizan las métricas asociadas a la seguridad del modelo, estas se representan en los siguientes gráficos:

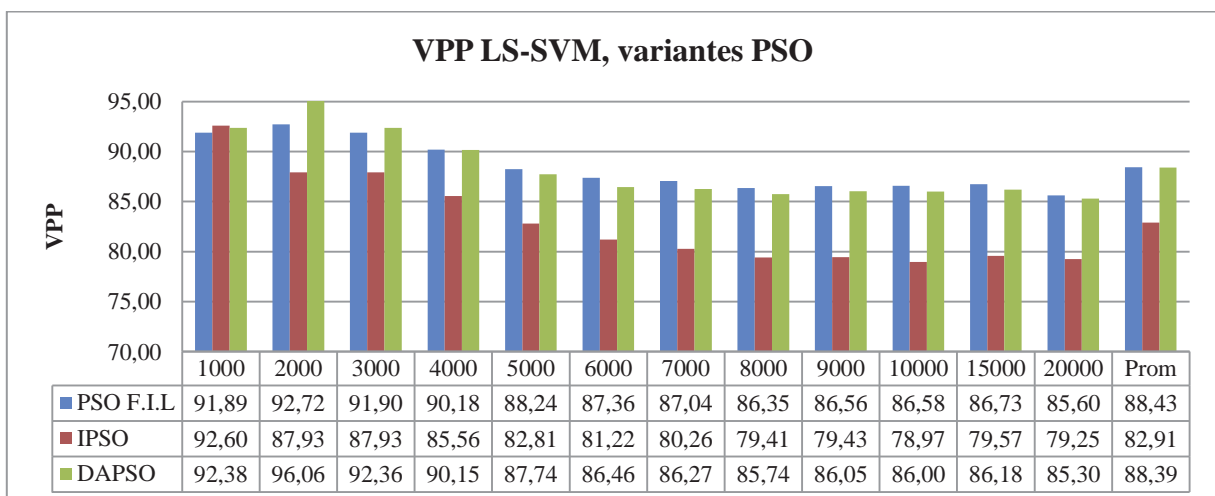


Figura 8.4 Gráfico VPP LS-SVM, variantes de PSO

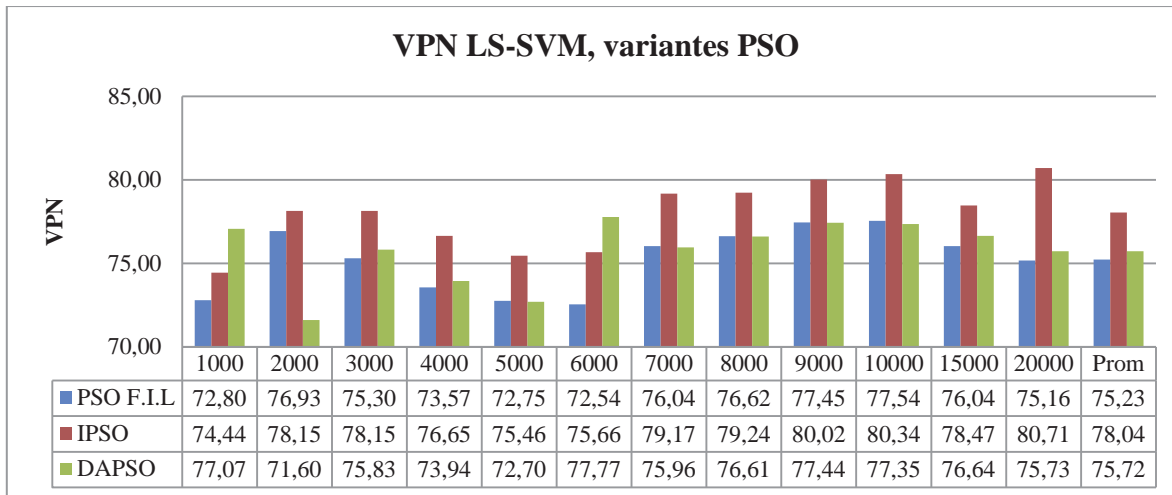


Figura 8.5 Gráfico VPN LS-SVM, variantes de PSO

Los gráficos comparativos de valor predictivo positivo (VPP) y valor predictivo negativo (VPN) denotan el grado de seguridad del modelo. Las variantes PSO con factor de inercia lineal y DAPSO presentaron los mejores resultados sobre un 88%, esta cifra significa que el 88% de las personas que se detecten como ilesos van a estar realmente ilesos. Por otra parte, IPSO obtuvo el mejor promedio de valor predictivo negativo con un 78,04%, significa que el 78,04% de las personas que se detecten como ilesas van a estar realmente ilesas. Además, ambas métricas al variar el tamaño de muestra para validación, los porcentajes se mantuvieron en un rango aceptable, lo que comprueba la buena precisión del modelo.

En resumen al visualizar y contrastar los resultados de los tres modelos se aprecia que no hubo una gran diferenciación entre ellos, sin embargo, PSO con factor de inercia lineal y DAPSO superaron a IPSO en la exactitud por casi 2 puntos porcentuales. Además, IPSO obtuvo el mayor tiempo de entrenamiento lo que no es menor. Aunque entre los modelos de PSO con factor de inercia lineal y DAPSO hubo una diferencia mínima, PSO con factor de inercia lineal es la que obtuvo un mayor rendimiento en la exactitud, es decir, el mejor porcentaje de personas bien clasificadas en el estado lesionado e ileso.

Para corroborar lo anterior se presenta un contraste entre los tres modelos mediante la ganancia, de esta forma es posible determinar si el modelo permitió una ganancia positiva o negativa (perdida). La fórmula de ganancia está dada por la siguiente ecuación:

$$GA = \frac{A - B}{A} * 100 \quad (6.2.6)$$

Donde G representa la ganancia de A sobre la B de manera porcentual.

En las siguientes tablas se puede apreciar la ganancia de exactitud de cada modelo sobre los demás.

Tabla 8.2 Comparación ganancia de exactitud entre modelos

A \ B	LS-SVM con PSO F.I.L	LS-SVM con IPSO	LS-SVM con DAPSO
LS-SVM con PSO F.I.L	0%	1,5358%	0,0731%
LS-SVM con IPSO	-1,5598%	0%	-1,4855%
LS-SVM con DAPSO	-0,0732%	1,4638%	0%

Como se aprecia en la tabla anterior el modelo de LS-SVM con PSO Factor de Inercia Lineal fue quien obtuvo ganancia sobre los demás modelos. Es así que obtuvo una ganancia de 1,5358% sobre LS-SVM con IPSO y un 0,0731% sobre LS-SVM con DAPSO.

Por lo tanto, se utiliza LS-SVM con PSO factor de inercia lineal en el siguiente subcapítulo para contrastar resultados con trabajos relacionados.

8.2 Comparación con trabajos relacionados

En primer lugar se contrastan los resultados obtenidos en [22], y detallados en el capítulo 4. Cabe destacar que en dicho trabajo se utilizó SVM convencional y la estimación de parámetros fue realizada a juicio del investigador. También es de considerar que se utilizaron otras características (variables de entrada del modelo) para la clasificación. Por otra parte, el trabajo citado solo presentó como métrica la exactitud para evaluar el clasificador, por ende solo se utilizara dicha métrica para la comparación.

De un total de 9 entrenamientos realizados al variar los valores de los parámetros de entrada utilizando el tipo de *kernel* RBF, se obtuvo como mejor resultado un 60,3%. En contraste con el modelo LS-SVM PSO con factor de inercia lineal realizado en este trabajo, hubo una gran diferencia entre ambos, esta es representada en el siguiente gráfico:

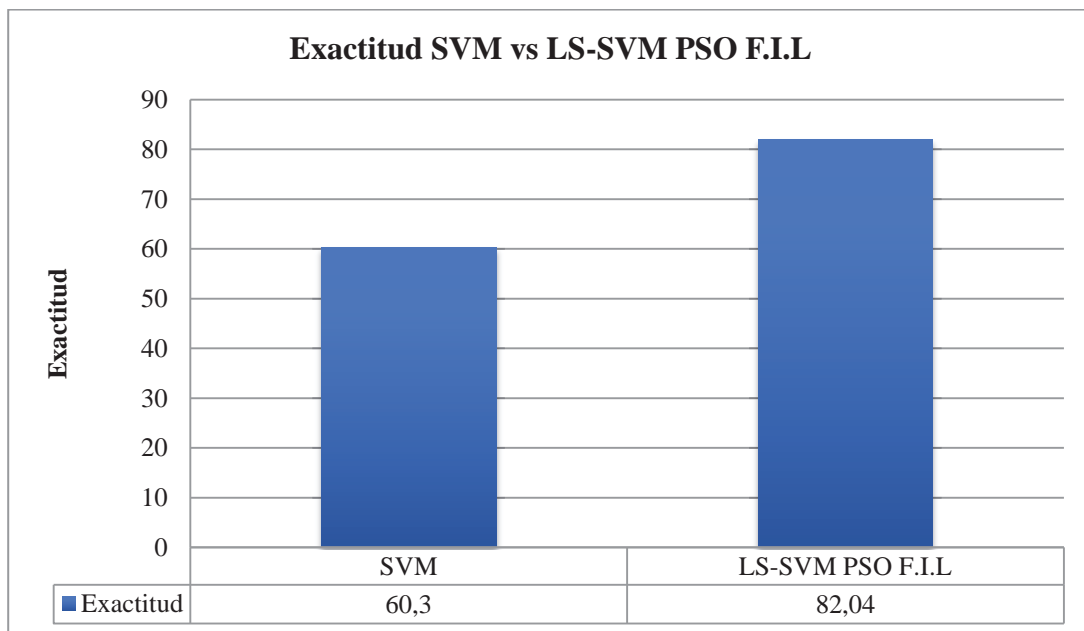


Figura 8.6 Exactitud SVM vs LS-SVM PSO F. Inercia Lineal

Al comparar la exactitud de ambos trabajos se concluye que el modelo LS-SVM con factor de inercia lineal obtuvo una amplia superioridad sobre SVM convencional, superándolo sobre un 20%. A su vez la ganancia fue de un 24,5 %. Cabe destacar que el 60,3% es el mejor resultado obtenido por dicho trabajo, no así el 82,04% el cual es un valor promedio.

En segundo lugar se realizó el contraste de resultados con el trabajo realizado en la Escuela de Ingeniería en Transportes de la PUCV y miembros de la Universidad USACH. La clasificación realizada en dicho trabajo fue realizada con redes bayesianas y como métricas se utilizaron la exactitud, sensibilidad y especificidad. EL contraste de los modelos es representado en la siguiente figura:

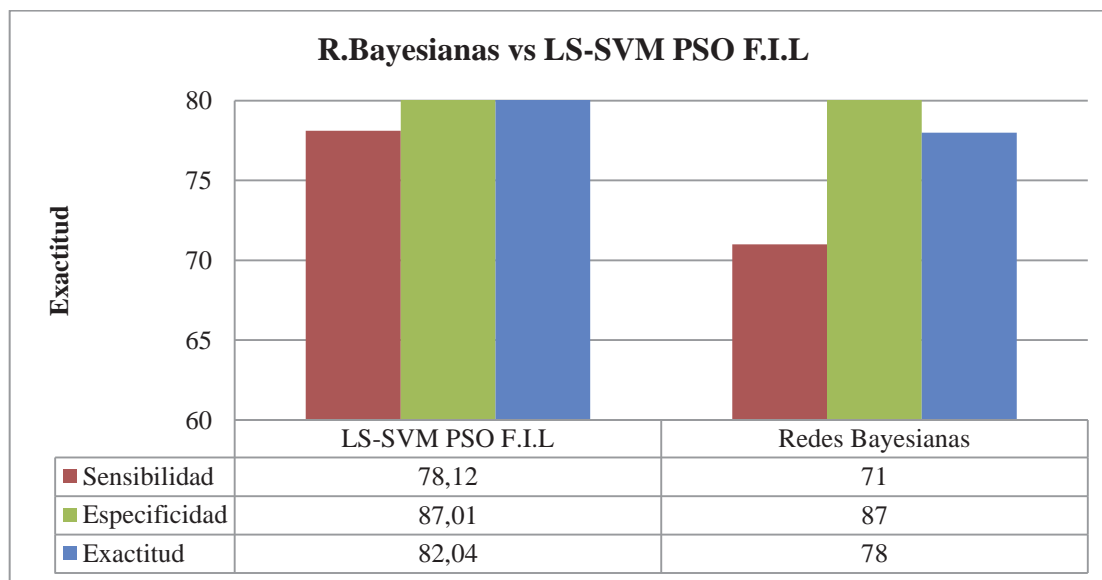


Figura 8.7 Redes Bayesianas vs LS-SVM PSO con Factor de Inercia Lineal

Al comparar estas tres métricas se observa que PSO con factor de inercia lineal también obtuvo mejores resultados que las redes bayesianas, superando en más de 4 puntos porcentuales en la exactitud del modelo. A su vez la ganancia fue de un 4,92%.

Finalmente es de importancia destacar y reiterar que estas comparaciones son realizadas en base al promedio de distintas muestras de los resultados obtenidos con PSO con factor de inercia lineal, no así en los trabajos relacionados que no utilizan un promedio de varias muestras, sino más bien, el mejor resultado para una muestra en particular. Por ende, las diferencias podrían ser aún más elevadas con el modelo obtenido en este trabajo de investigación.

9 Conclusión

En la presente memoria se logró establecer y explicar el marco conceptual general del estudio de SVM y PSO, el cual fue base para la implementación del trabajo de clasificación de accidentes de tránsito de la región de Valparaíso. Se presentaron los fundamentos y definiciones de cada técnica, principalmente aquellas que fueron clave en la obtención del clasificador, estas fueron: LS-SVM, las variantes de PSO y las funciones *fitness* utilizadas en cada una de estas. Con esto se cumplió uno de los primeros objetivos establecidos al comienzo de esta investigación.

Gracias al análisis y estudio realizado sobre LS-SVM y PSO se logró diseñar la estructura del modelo de clasificación, el cual fue base para la implementación de estimación de parámetros de cada modelo. De un total de 70.000 datos aprox. de accidentes de tránsito (12 variables de entrada cada uno) recolectados entre los años 2003 y 2009, se escogieron al azar en primera instancia 2.000 para la etapa de training y 1000 para la etapa de testing. En segunda instancia se tomaron 12 tamaños de muestras para validar cada modelo. En base a lo anterior, se utilizaron tres variantes de PSO: factor de inercia lineal, IPSO y DAPSO, en donde cada una de ellas se evaluó en dos tipos de *fitness*: costo de elementos mal clasificados utilizando validación cruzada para 10 subconjuntos y costo de elementos mal clasificados utilizando la función del error absoluto medio. De estos dos, el costo de la validación cruzada para 10 subconjuntos de 200 registros cada uno, obtuvo mejores resultados en comparación con el *fitness* error absoluto medio. Esto ocurrió para cada variante de PSO implementada.

A partir del análisis de los distintos modelos propuestos utilizando como *fitness* el costo de la validación cruzada, se concluye que no hubo una gran diferenciación entre los tres modelos, sin embargo, PSO con factor de inercia lineal (82,04%) fue quien obtuvo mejores resultados en exactitud, superando al peor modelo IPSO (80,78%) en casi 2 puntos porcentuales de diferencia. A su vez, la ganancia entre PSO con factor de inercia lineal sobre IPSO fue de 1,53% en exactitud. Es de importancia destacar que estos porcentajes de exactitud de cada modelo corresponden a un promedio de 12 muestras de validación.

Asimismo, al comparar los resultados de la mejor variante obtenida en este trabajo de investigación con los resultados obtenidos en [22] y [24] se establece que PSO con factor de inercia lineal superó ampliamente a dichos trabajos, en exactitud, sensibilidad y especificidad. La diferencia en exactitud sobre [22] fue de 21,74% con una ganancia de 26,49% y sobre [24] la diferencia fue de 4,04% con una ganancia de 4,92%. Es de importancia destacar que las comparaciones fueron realizadas en base al promedio de 12 muestras de los resultados obtenidos en PSO con factor de inercia lineal, no así en los trabajos relacionados que no utilizan un promedio de varias muestras, sino más bien, el mejor resultado para una muestra en particular. Por ende, las diferencias podrían ser aún más elevadas al utilizar como referencia de comparación el modelo obtenido en este trabajo (LS-SVM con PSO factor de inercia lineal).

Por otra parte, gracias a esta memoria se realizaron dos papers en conjunto con la escuela de Ingeniería en Transportes de la Pontificia Universidad Católica de Valparaíso, los cuales fueron presentados y publicado en los anales del congreso OPTIMA 2011, Pucón,

Chile, 26-29 de Octubre 2011 [28] y XV Congreso Chileno de Ingeniería de Transporte, Santiago, Chile, 3-6 de Octubre 2011 [29].

Finalmente se concluye que las máquinas de soporte vectorial si son capaces de presentar un buen grado de generalización para clasificar el estado en el cual resultan las personas (lesionadas o ilesas) de la región de Valparaíso, presentándose el modelo PSO con factor de inercia lineal con un 82,04% de exactitud en rendimiento.

10 Bibliografía

- [1] Organización Mundial de la Salud. *Informe sobre la situación mundial de seguridad vial: es hora de pasar a la acción*. Ginebra : s.n., 2009. http://www.who.int/violence_injury_prevention/road_safety_status/2009.
- [2] Comisión Nacional de Seguridad de Tránsito. *Tipos de Tránsito 2000-2009*. CONASET. Santiago : s.n., 2009. http://www.conaset.cl/portal/portal/default/estadisticas_generales..
- [3] Vapnik., V. *The nature of statistical learning theory*. New York: Springer-Verlag : s.n., 1995.
- [4] Vapnik, V. *Statistical Learning Theory*. New York : John Wiley and Sons, 1998.
- [5] Suykens, J.A.K y Vandewalle, J. *Least Squares Support Vector Machine Classifiers*. s.l. : Neur.Proc.Lett, 1999.
- [6] Eberhart, R y Kennedy, J. *A new optimizer using particle swarm theory*. s.l. : Proceedings of the sixth International Symposium on Micro Machine and Human Science MHS'95, 1995. págs. 39-43.
- [7] Eberhart, R y Kennedy, J. *Particle swarm optimization*. Australia : Proceedings of the IEEE International Conference on Neural Networks-ICNN'95, 1995. págs. 1942-1948.
- [8] Shi, Y y Eberhart, R. *A modified particle swarm optimizer*. s.l. : Proc. of the IEEE Int., 1998. págs. 69-73.
- [9] Jiao, Bin, Lian, Zhigang y Gu, Xingsheng. *A dynamic inertia weight particle swarm optimization algorithm*. China : s.n., 2006.
- [10] Xueming, Yang, y otros. *A modified PSO with dynamic adaptation*. China : s.n., 2006.
- [11] Clerc, M. *The Swarm and the Queen: Towards a Deterministic and Adaptive Particle*. s.l. : En Proceeding of IEEE Congress on Evolutionary Computation, 1999. págs. 1957–1999.
- [12] Mendes, R, Kennedy, J y Neves, J. *The Fully Informed Particle Swarm: Simpler maybe Better*. s.l. : En IEEE Transactions on Evolution Computation, 2004. págs. 204-210.
- [13] Kennedy, J. *Small World and Mega-Minds: Effects of Neighborhood Topology on Particle Swarm Performance*. s.l. : En Proceedings of the 1999 Congress on Evolutionary Computation, 1999. págs. 1931–1938.
- [14] Dia, H y Rose, G. *Development and Evaluation of Neural Network Freeway Incident Detection Models Using Field Data*. s.l. : Transportation Research C, 1997. págs. 313-331.
- [15] Yang, W.T, Chen, H. C y Brown, D. B. *Detecting Safer Driving Patterns By A Neural*. 1999.

- [16] Mussone, L, Ferrari, A y Oneta, M. *An analysis of urban collisions using an artificial intelligence model*. Milán : Accident Analysis and Prevention, 1999. págs. 705-718.
- [17] Evanco, W. M.,. *The Potential Impact of Rural Mayday Systems on Vehicular Crash Fatalities*. s.l. : Accident Analysis and Prevention, Vol. 31, 1999. págs. 455-462.
- [18] Bedard, M., Guyatt, G. H., Stones, M. J., & Hireds, J. P. *The Independent Contribution of Driver, Crash, and Vehicle Characteristics to Driver Fatalities*. s.l. : Accident analysis and Prevention, 2002. págs. 717-727.
- [19] Ossiander, E. M., & Cummings, P.,. *Freeway speed limits and Traffic Fatalities in Washington State*. s.l. : Accident Analysis and Prevention, Vol. 34, 2002. págs. 13-18.
- [20] Abdelwahab, M y Abdel, Aty. *Analysis and Prediction of Traffic Fatalities Resulting From Angle Collisions Including the Effect of Vehicles' Configuration and Compatibility*. s.l. : Accident Analysis, 2003.
- [21] Sohn, So Young y Lee, Sung Ho Lee. *Data fusion, ensemble and clustering to improve the classification accuracy for the severity of road traffic accidents in Korea*. s.l. : Safety Science, 2003.
- [22] Chong, Miao. Ajit, Abraham. Paprzycki, Marcin. *Accident Data Mining Using Machine Learning Paradigms*. Computer Science Department. Oklahoma State University, USA : s.n., 2003.
- [23] Montt, C. y Otros. *Conductas y actitudes de los usuarios del sistema de transporte en la seguridad vial*. . Valparaíso : Proyecto de Investigación Universidad Católica de Valparaíso., 2006.
- [24] Montt, Cecilia, Zúñiga, Alejandro y Chacón, Max. *Estructura Bayesianas en Accidentes de Tránsito*. 2009.
- [25] Montt, Cecilia, Musso, Reynaldo y Chacón, Max. *Análisis de Accidentes de Tránsito con Métodos de Agrupamiento*. 2009.
- [26] Zweig, M y Campbell, G. *Receiver-Operating Characteristic (ROC) Plots: A Fundamental Evaluation Tool in Clinical Medicine*. s.l. : Clinical Chemistry, 1993. págs. 561-577. Vol. 39.
- [27] MATLAB/C LS-SVMlab toolbox v1.7. [En línea] <http://www.esat.kuleuven.be/sista/lssvmlab/>.
- [28] Castro F, Félix, Montt, Cecilia y Rodríguez, Nivaldo. *SVM BASADO EN OPTIMIZACIÓN POR ENJAMBRES DE PARTÍCULAS PARA CLASIFICACIÓN DE ACCIDENTES DE TRÁNSITO*. Pucon : s.n., 2011.
- [29] Castro F, Félix, Montt, Cecilia y Rodríguez, Nivaldo *SVM BASADO EN OPTIMIZACIÓN POR ENJAMBRES DE PARTÍCULAS PARA CLASIFICACIÓN DE ACCIDENTES DE TRÁNSITO*. Santiago : s.n., 2011.

Anexos

10.1A: Planilla SIEC 2

REGISTRO DE ACCIDENTES EN EL TRANSITO Y FERROVIARIOS

Membrete Comisaría
o Destacamento

SIEC 2

Cód. Unidad

Nº. Formulario

IDENTIFICACION

CLASE ACCIDENTE → TRANSITO 1 FERROVIARIO 2 Fecha → DIA MES AÑO Hora → HORA MINUTO

SUBSECTOR (Si sucedió FUERA DEL SECTOR JURISDICCIONAL, anote "99" y Nombre Comuna en línea siguiente)

COMUNA

TIPO DE ACCIDENTE		(Marque SOLO una alternativa)	
ATROPELLO	10		
CAIDA	20		
COLISION		CHOQUE CON OTRO VEHICULO DETENIDO	
		Fronte/Posterior	61
		Lado/Posterior	66
		Posterior/Posterior	69
Fronte/Lado	62	Posterior/Frente	67
Fronte/Posterior	63	Posterior/Lado	68
Lado/Frente	64	(Causante/Pasivo)	
Lado/Lado	65		
IMPACTO C/ ANIMAL	40	VOLCADURA	70
CHOQUE CON OBJETO		INCENDIO	80
Frontal	51	DESCARRILAMIENTO	90
Lateral	52		
Por Alcance	53		
Perpendicular	34	OTRO TIPO	99

UBICACION RELATIVA (Marque ALTERNATIVA y complete datos que correspondan) URBANA 1 RURAL 2 VIA FERREA 3

URBANA O RURAL

TRAMO DE VIA RECTA	01	CRUCE CON SEMAFORO FUNCIONANDO	11	ENLACE A NIVEL	21
TRAMO DE VIA CURVA HORIZONTAL	02	CRUCE CON SEMAFORO APAGADO	12	ENLACE A DESNIVEL	22
TRAMO DE VIA CURVA VERTICAL	03	CRUCE REGULADO POR CARABINERO	13	ACCESO NO HABILITADO	23
ACERA O BERMA	04	CRUCE CON SEÑAL "PARE"	14	ROTONDA	24
PUENTE	05	CRUCE CON SEÑAL "CEDA EL PASO"	15	PLAZA DE PEAJE	25
TUNEL	06	CRUCE SIN SEÑALIZACION	16	OTROS NO CONSIDERADOS	99

CALLE RUTA ROL

VIA 1

VIA 2

VIA 3

FRENTE AL NUMERO

UBICACION (desde Km. 0) y Kilómetro (con UN decimal)

VIA FERREA

RECINTO ESTACION	01	Km/Poste <input type="text"/>	CRUCE HABILITADO	03
TRAMO DE VIA	02		(con un decimal)	CRUCE NO HABILITADO

CAUSA BASAL PROBABLE

(INDIQUE CON "*" EL CAUSANTE) **VEHICULOS PARTICIPANTES**

IDENT.	PATENTE	TIPO (+)	SERV. (+)	CONS. (+)	VIA (Ⓢ)	DIREC. (+)	MANIOBRA (+)	MARCA	CODIGO	AÑO
A										
B										
C										
D										
E										
F										
G										
H										
I										
J										
K										

(Si hay más VEHICULOS, detálalos en hoja adjunta con el mismo formato de este CUADRO y marque "X")

Notas: (+) Ver CODIGO respectivo en Instructivo SIEC 2
(Ⓢ) Anotar Nro. De Vía de Ubicación Relativa por la que circula cada vehículo

ESTADO ATMOSFERICO						LUMINOSIDAD				SIE	
DESPEJADO	1	LLUVIA	3	NEBLINA	5	DIURNA	1	AMANECEER	3	LUZ ARTIFICIAL	
NUBLADO	2	LLOVIZNA	4	NIEVE	6	NOCTURNA	2	ATARDECER	4	NO EXISTE	1
										SI EXISTE	2
UNIDIRECCIONAL 1			BIDIRECCIONAL 2			BIDIRECCIONAL CON BANDEJON 3			APAGADA		
CANTIDAD DE PISTAS			CANTIDAD DE PISTAS IDA			CANTIDAD DE PISTAS REGRESO			ENCENDIDA SUFICIENTE		
(PISTAS DE IDA CORRESPONDEN AL VEHICULO CAUSANTE)											
TIPO DE CALZADA				ESTADO CALZADA				CONDICION (*) TIPO = CONCRETO/ ASFALTO/ ADOQUIN/ MIXTO			
CONCRETO (*)	1	MIXTO (*)	4	BUENO	1	SECO	1	CON BARRO	4	ESCARCHA	
ASFALTO (*)	2	RIPIO	5	REGULAR	2	HUMEDO	2	CON NIEVE	5	GRAVILLA	
ADOQUIN (*)	3	TIERRA	6	MALO	3	MOJADO	3	CON ACEITE	6	OTROS	
(*) DEMARCACION (SI TIPO DE CALZADA ES CONCRETO, ASFALTO, ADOQUIN O MIXTO)			Línea Continua		Pare en Calzada		Otras Demarcaciones				
			Línea Discontinua		Ceda el Paso en Calzada		BORRADAS				
			Línea Mixta		Paso Peatonal		SIN DEMARCACION				
(Marque TODAS las existentes)											

PERSONAS PARTICIPANTES

CALIDAD	SEXO	RESULTADO	SOLO PARA CONDUCTORES		
			CLASE DE LICENCIA	CONDICION FISICA	NACIONALIDAD
1 - CONDUCTOR 2 - PASAJERO 3 - PEATON	1. MAS. 2. FEM.	1. MUERTO 4. LEVE 2. GRAVE 5. ILESO 3. M. GRAVE	A1 B PP PERMISO PROVISORIO A2 C BC BOLETA DE CITACION A3 D SL SIN LICENCIA A4 E FV FECHA DE CONTROL VENCIDA A5 F PE PERMISO EXTRANJERO OT OTRAS	1. NORMAL 2. BAJO INFLUENCIA ALCOHOL 3. EBRIEDAD 4. BAJO INFLUENCIA DROGAS 5. FATIGA	1. CHILENA 2. ARGENTINA 3. BRASILEIRA 4. BOLIVIANA 5. PERUANA 6. OTRAS SUDAMERICANAS 9. OTRAS
(Agregue ASTERISCO (*) AL CAUSANTE)		CINT. SEGUR/CASCO (CONDUCT./PASAJ.) 0 - IGNOR 1-SI 2-NO			

##	CALID	SEXO	EDAD	RESULT.	CINT.	R.U.N.	VEHIC.	CL.LIC.	COMUNA	CODIGO	C.FIS.	NAC
01												
02												
03												
04												
05												
06												
07												
08												
09												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												

(Si hay más PARTICIPANTES, detállelos en hoja adjunta con el mismo formato de este CUADRO y marque "X")

CLASIFICACION	DETENIDO(S)	1	DENUNCIA	2	S.I.A.T.	CONCURRIO	1	NO CONCURRIO	2	SI HUBO MUERTOS	MENS:	DEL	/ /
---------------	-------------	---	----------	---	----------	-----------	---	--------------	---	-----------------	-------	-----	-----

ACLARATORIA _____

ve 8º

----- SUBCOMISARIO DE LOS SERVICIOS (Nombre, Grado y Sello) -----

----- ENCARGADO S.I.E.C. (Nombre y Grado) -----