

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

**PRONÓSTICO DE MAREA ROJA PARA LA
REGIÓN DE MAGALLANES UTILIZANDO
MÁQUINAS DE SOPORTE VECTORIAL CON
OPTIMIZACIÓN POR ENJAMBRE DE
PARTÍCULAS**

DIEGO ANDRÉS VERA BÓRQUEZ

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

MARZO 2012

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

**PRONÓSTICO DE MAREA ROJA PARA LA
REGIÓN DE MAGALLANES UTILIZANDO
MÁQUINAS DE SOPORTE VECTORIAL CON
OPTIMIZACIÓN POR ENJAMBRE DE
PARTÍCULAS**

DIEGO ANDRÉS VERA BÓRQUEZ

Profesor Guía: **Nibaldo Rodríguez Agurto**

Carrera: **Ingeniería Civil Informática**

MARZO 2012

Dedicatoria

Dedicada a mis padres, que me nutrieron con fuerzas e incondicional cariño a lo largo de todo este proceso. También a mis abuelos, que en paz descansen.

Agradecimientos

En mi vida universitaria conocí a personas muy especiales e hice muy buenos amigos, con quienes disfrute de innumerables anécdotas y gratos momentos. A ellos, les doy las gracias por haber formado parte de esta travesía.

Agradeceré una y mil veces a mis padres, por cada palabra de aliento, gesto de cariño y por estar siempre junto a mí, a pesar de aquella geografía que tanto nos distancia. También a mi hermano, a quien permanentemente sentí como un apoyo, con nuestras espaciadas pero oportunas conversaciones. Ellos fueron pilares fundamentales en la construcción de este camino.

Agradezco a mi profesor guía, por estar siempre dispuesto a resolver mis dudas y a sugerirme distintas vías para el desarrollo del presente proyecto. Finalmente, le doy las gracias a la Sra. Georgina Lembeye, de la Subsecretaría de Pesca, quien me facilitó la fuente de datos, elemento base, para el desarrollo de esta investigación.

Índice

Lista de Abreviaturas	v
Lista de Figuras	vi
Lista de Tablas	vii
Resumen	viii
Abstract	viii
1 Introducción	1
2 Marco General del Proyecto.....	4
2.1 Objetivo General.....	4
2.2 Objetivos Específicos	4
2.3 Plan de Trabajo.....	4
3 Máquinas de Soporte Vectorial	6
3.1 Máquinas de Soporte Vectorial para Clasificación	6
3.1.1 Caso Linealmente Separable o de Margen Máximo	6
3.1.2 Caso Linealmente no Separable o de Margen Flexible.....	11
3.1.3 Funciones Kernel.....	14
3.1.4 Caso no Lineal.....	15
3.2 Máquinas de Soporte Vectorial para Regresión	17
3.3 Máquinas de Soporte Vectorial de Mínimos Cuadrados	22
3.4 Ventajas y Desventajas de las SVM.....	24
4 Optimización por Enjambres de Partículas	25
4.1 Principios	25
4.2 Formalización de PSO.....	26
4.2.1 Algoritmo PSO.....	27
4.2.2 Parámetros	28
4.3 Variantes de PSO.....	29
4.3.1 Vecindad Local	29
4.3.2 Adaptive PSO (APSO).....	30
4.3.3 Fitness Distance Ratio Based Particle Swarm Optimization (FDRPSO).....	31
4.3.4 Particle Swarm With Exponential Distribution (PSO-E).....	32
4.3.5 Adaptive Inertia Weight PSO (AIWPSO).....	33
5 Elementos del Estudio	35
5.1 Estructura y Contexto de los Datos	35

5.2	Variables.....	35
5.3	Estaciones.....	37
6	Medidas de Evaluación.....	39
6.1.1	Ejemplo Contextual.....	42
7	Modelo LS-SVM PSO.....	44
7.1	Descripción del Modelo.....	44
7.1.1	Pre-proceso de Datos.....	44
7.1.2	Inicialización del Enjambre PSO.....	45
7.1.3	Entrenamiento.....	46
7.1.4	Actualización de Partículas.....	46
7.1.5	Validación del Modelo.....	46
8	Implementación y Resultados del Modelo.....	47
8.1	Datos Utilizados.....	47
8.2	Codificación.....	48
8.3	Normalización y Desfases.....	48
8.4	Mezcla de Registros.....	49
8.5	Resultados Modelo LS-SVM PSO.....	49
8.5.1	Metodología de Pruebas.....	50
8.5.2	Resultados PSO Con Factor de Inercia.....	51
8.5.3	Resultados APSO.....	52
8.5.4	Resultados FDRPSO.....	53
8.5.5	Resultados AIWPSO.....	54
9	Análisis de Resultados.....	56
9.1	Análisis Por Modelo.....	56
9.1.1	Modelo PSO – FI.....	56
9.1.2	Modelo APSO.....	57
9.1.3	Modelo FDRPSO.....	58
9.1.4	Modelo AIWPSO.....	60
9.2	Análisis General.....	61
10	Conclusiones.....	66
	Referencias.....	67
	Anexos.....	71
	A: Estaciones de Monitoreo.....	72

Lista de Abreviaturas

- AIWPSO : Adaptive Inertia Weight PSO.
- APSO : Adaptative Particle Swarm Optimization.
- FAC : Floración de Alexandrium Catenella.
- FAN : Floración de Algas Nocivas.
- FDRPSO : Fitness Distance Ratio Based Particle Swarm Optimization.
- LS-SVM : Least Squares Support Vector Machines.
- MSV : Máquinas de Soporte Vectorial.
- PSO : Particle Swarm Optimization.
- PSO FI : Particle Swarm Optimization con Factor de Inercia
- SVM : Support Vector Machine.
- VAM : Veneno Amnésico de los Mariscos.
- VDM : Veneno Diarreico de los Mariscos.
- VPM : Veneno Paralizante de los Mariscos.

Lista de Figuras

Figura 3.1 Hiperplano separadores.	7
Figura 3.2 Hiperplanos canónicos.	8
Figura 3.3 Margen, hiperplanos de soporte y vectores de soporte.	8
Figura 3.4 Caso linealmente no separable.	12
Figura 3.5 Proyección de puntos al espacio de características.	15
Figura 3.6 Casos en análisis de las SVM.	15
Figura 3.7 Esquema general Máquinas de Soporte Vectorial.	17
Figura 3.8 Máxima desviación	18
Figura 3.9 Función de pérdida para un caso lineal [17].	19
Figura 4.1 Actualización de una partícula.	27
Figura 4.2 Algoritmo PSO [25].	28
Figura 4.3 Topologías, asociadas a gbest y lbest [25].	30
Figura 4.4 Pseudocódigo APSO [32].	31
Figura 4.5 Distribución exponencial [46].	33
Figura 5.1 Distribución geográfica de las estaciones de muestreo en la región de Magallanes. En rojo de destaca el núcleo de toxicidad “Última Esperanza” [1].	38
Figura 6.1 Curva ROC.	41
Figura 7.1 Modelo LS-SVM PSO.	44
Figura 9.1 Medidas de evaluación. PSO-FI.	57
Figura 9.2 Medidas de evaluación. APSO.	58
Figura 9.3 Medidas de evaluación. FDRPSO.	59
Figura 9.4 Medidas de evaluación. AIWPSO.	61
Figura 9.5 Medidas de evaluación. Mejores Resultados.	62
Figura 9.6 Curvas ROC PSO-FI, APSO, AIWPSO y FDRPSO.	63
Figura 9.7 Curva ROC PSO-FI.	63
Figura 9.8 Curva ROC APSO.	64
Figura 9.9 Curva ROC FDRPSO.	64
Figura 9.10 Curva ROC AIWPSO.	65

Lista de Tablas

Tabla 2.1 Plan de trabajo.....	5
Tabla 5.1 Descripción de variables.....	36
Tabla 6.1 Matriz de confusión.....	39
Tabla 6.2 Grupos, matriz de confusión.	39
Tabla 6.3 Matriz de confusión. Ejemplo.	42
Tabla 6.4 Resultados matriz de confusión.....	42
Tabla 6.5 Resultados medidas de evaluación.	43
Tabla 7.1 Ejemplo desfases	45
Tabla 8.1 Ejemplo codificación registros.....	48
Tabla 8.2 Parámetros PSO.....	50
Tabla 8.3 Resultado Modelo LS-SVM PSO Con Factor de Inercia. Etapa testing.	51
Tabla 8.4 Resultado Modelo LS-SVM APSO.....	53
Tabla 8.5 Resultado Modelo LS-SVM FDRPSO.....	54
Tabla 8.6 Resultado Modelo LS-SVM AIWPSO.....	55
Tabla 9.1 Comparación porcentual de resultados por validación. Modelo PSO - FI.....	56
Tabla 9.2 Configuración entrenamiento PSO -FI.....	56
Tabla 9.3 Comparación porcentual de resultados por validación. Modelo APSO.	57
Tabla 9.4 Configuración entrenamiento APSO.....	58
Tabla 9.5 Comparación porcentual de resultados por validación. Modelo FDRPSO.	59
Tabla 9.6 Configuración entrenamiento FDRPSO.....	59
Tabla 9.7 Comparación porcentual de resultados por validación. Modelo AIWPSO.....	60
Tabla 9.8 Configuración entrenamiento AIWPSO.....	60
Tabla 9.9 Diferencia porcentual APSO v/s PSO FI, FDRPSO y AIWPSO. Mejor Modelo.	61
Tabla 9.10 Áreas bajo la curva ROC.....	62
Tabla 9.11 Diferencias de puntos de área bajo la curva ROC. APSO v/s PSO FI, FDRPSO, AIWPSO.....	62
Tabla A.1 Estaciones de Monitoreo. Sector Última Esperanza. Región de Magallanes.....	72

Resumen

En Chile, los fenómenos de Floraciones de Algas Nocivas (FAN) se concentran principalmente en las regiones de Los Lagos, Aysén y Magallanes, dichos eventos pueden contaminar mariscos con toxinas y a su vez generar un problema de salud pública y/o afectar negativamente la producción del sector pesquero, al comercio y al turismo. Por lo tanto, es necesario mantener monitoreadas las zonas afectadas y construir modelos para predecir el comportamiento de dichos fenómenos.

En este proyecto se desarrolla un modelo para pronosticar la presencia de *Alexandrium Catenella* (microalga nociva, fuente de marea roja) en la región de Magallanes, utilizando Máquinas de Soporte Vectorial de Mínimos Cuadrados (LS-SVM) y algunas variantes de Optimización por Enjambre de Partículas (PSO).

Los resultados de las pruebas se analizan ocupando las métricas: precisión y sensibilidad. El mejor resultado se obtiene con el modelo LS-SVM APSO (Optimización por Enjambre de Partículas Adaptativo), logrando una precisión de 85% y una sensibilidad de 80%.

Palabras claves: *Alexandrium Catenella, Clasificación, Floración de Algas Nocivas, Máquinas de Soporte Vectorial, Máquinas de Soporte Vectorial de Mínimos Cuadrados, Optimización por Enjambre de Partículas, Pronóstico.*

Abstract

In Chile, the phenomena of Harmful Algal Blooms (HABs) are a problem mainly concentrated in Los Lagos, Aysén and Magallanes regions. These events can contaminate seafood with toxins and become a public health problem, affecting negatively the fisheries sector, commerce and tourism. Under the above context, it is necessary to keep the affected areas monitored and try to predict the behavior of these phenomena.

This project has developed a forecasting model of *Alexandrium Catenella* (microalgae harmful red tide source or HAB) for the Magallanes region, using Support Vector Machine Least Squares (LS-SVM) and some variants by Swarm Optimization Particle (PSO).

The results of this study are discussed occupying two metrics: precision and sensitivity. The best result was achieved with the LS-SVM model APSO (Particle Swarm Optimization by Adaptive), with which it has obtained an accuracy of 85% and a sensitivity of 80%.

Key words: *Alexandrium Catenella, Classification, Harmful Algal Blooms, Support Vector Machines, Support Vector Machine Least Squares, Particles Swarm Optimization, Forecast.*

1 Introducción

Al fenómeno conocido como marea roja se le denomina técnicamente “Floración de Algas Nocivas” (FAN), lo que corresponde a una proliferación, en ambientes acuáticos, de algas microscópicas, que pueden, por ejemplo, contaminar mariscos con toxinas, causar muertes masivas de peces, disminuir los niveles de oxígeno en el mar, producto de su descomposición [1], lo que consecuentemente puede causar la muerte por anoxia de otros organismos. Las condiciones que favorecen la proliferación de las algas nocivas están relacionadas con la interacción de un conjunto de aspectos biológicos, químicos y físicos que aún no están del todo comprendidos. Las consecuencias y los mecanismos de impacto de las FAN varían dependiendo de las especies involucradas. Según [7] existen cuatro categorías de efectos nocivos: riesgos para la salud humana, la pérdida de recursos pesqueros naturales o cultivados, el deterioro de las actividades turísticas y recreativas, e impactos al ecosistema marino que afecta a los recursos no comerciales.

En general, las microalgas nocivas son catalogadas como productoras de toxinas y de altas biomásas. Entre las principales toxinas producidas por las FAN se encuentra el Veneno Paralizante de los Mariscos (VPM), el Veneno Diarreico de los Mariscos (VDM) y el Veneno Amnésico de los Mariscos (VAM). Estas toxinas pueden ser transmitidas a los seres humanos por medio del consumo, principalmente, de moluscos bivalvos contaminados, causando graves problemas de salud, incluso, la muerte. Entre las principales especies de mariscos susceptibles de contaminarse (transvectores) se encuentran las cholgas, choritos, choro-zapato, ostras, almejas, navajuelas y ostiones. También se han descubierto otros organismos marinos transvectores no bivalvos, como los locos, caracoles, picorocos, piures y erizos [12].

En Chile los fenómenos de FAN [1] [2] [3] no han estado ausentes, siendo las regiones sur austral del país (Los Lagos, Aysén y Magallanes) las más afectadas. La región de los Lagos ha tenido floraciones asociadas al veneno diarreico desde 1970, al veneno amnésico desde 2000 y desde 2002 al veneno paralizante. La región de Aysén se ha visto perturbada por floraciones asociadas a los venenos diarreico y paralizante desde mediados de los noventa. Magallanes se ha caracterizado por contaminaciones producidas por el veneno paralizante desde 1972. Este último tipo de veneno se distribuye dentro una amplia extensión geográfica, que abarca desde la región de Los Lagos hasta la región de Magallanes. Las intoxicaciones fatales, en Chile, están asociadas únicamente con VPM y suman un total de 34 fallecidos de los cuales 22 se produjeron en Magallanes, 11 en Aysén y 1 en Los Lagos.

Actualmente, se exportan cerca de 30 recursos susceptibles de contener toxinas marinas. Las exportaciones de estos mariscos potencialmente portadores representan 100 millones de dólares anuales, esto sumado a los aproximadamente 30 millones de dólares por consumo interno, sugiere el volumen de ventas amenazadas por las FAN.

Las potenciales consecuencias de la aparición de una FAN, son las siguientes [3]:

Efectos inmediatos: disminución de la oferta de mariscos y pérdida de ventas, debido a restricciones a la cosecha o extracción. Este efecto es inmediato a la detección de una FAN y resulta de las resoluciones que restringen la captura o cosecha, procesamiento y comercialización.

Efectos a mediano plazo: como consecuencia de lo anterior, y dependiendo de lo prolongado de la restricción impuesta y de la ubicación del área, se puede producir a mediano plazo una disminución de la mano de obra.

Efectos a largo plazo: si la situación se convierte en crónica, se puede producir un desincentivo a las inversiones en el sector productor de mariscos y, en el peor de los casos, cambios en los hábitos de los consumidores, lo que significa una disminución de la demanda en el largo plazo.

El organismo identificado como fuente primaria del VPM es la microalga *Alexandrium Catenella*, la cual es un dinoflagelado planctónico de amplia distribución mundial y que habita zonas costeras de aguas templadas frías. Produce saxitoxina (VPM) y derivados que son neurotoxinas hidrosolubles y termoestables [4] [11]. Debido al riesgo y al efecto negativo que representa este microorganismo para la actividad pesquera y la acuicultura, la Subsecretaría de Pesca propuso a la microalga *Alexandrium Catenella* como plaga hidrobiológica [3], lo que implica generar programas de vigilancia, detección, control y/o erradicación¹. Actualmente en Chile existen programas de monitoreo, los cuales permiten tener un control preventivo de las zonas expuestas a las FAN, estos programas son necesarios ya que no existen antídotos para las intoxicaciones con VPM, VDM o VAM, tampoco existen tecnologías que permitan controlar las floraciones en el ambiente o eliminar las toxinas de los mariscos. Además, no se han desarrollado modelos que permitan predecir el comportamiento de dichos fenómenos en la zona sur austral de Chile.

A nivel mundial existen algunos programas como [8] y [9] que tienen como objetivo fomentar y promover la investigación cooperativa dirigida a mejorar la predicción de eventos FANs. En [5] y [13] se exponen dos modelos que pretenden predecir floraciones de algas nocivas muy particulares en zonas muy específicas, con lo cual se puede visualizar la inmensidad del problema y su gran extensión geográfica.

En el desarrollo del presente proyecto se utiliza como fuente de datos, el conjunto de muestras tomadas desde el programa “*Manejo y monitoreo de las mareas rojas en las regiones de Los Lagos, Aysén y Magallanes*”² y mediante la utilización de Máquinas de Soporte Vectorial (SVM, Support Vector Machines) se busca predecir el comportamiento de *Alexandrium Catenella* en la región de Magallanes.

Las Máquinas de Soporte Vectorial son un conjunto de algoritmos de aprendizaje supervisado que persiguen la identificación de patrones dentro de un grupo de datos. Las SVM tienen dos versiones: una para resolver problemas de clasificación y otra para resolver problemas de regresión. Existen algunas variaciones a la formulación original de las SVM, como las Máquinas de Soporte Vectorial de Mínimos Cuadrados (LS-SVM, Least Squares Support Vector Machines), la cual modifica algunos aspectos de las SVM con el fin de mejorar su rendimiento. Las SVM tienen algunos parámetros de ajuste, los cuales se pueden optimizar. Como técnica de optimización se utilizará una metaheurística evolutiva basada en

¹ D.S. N° 345-05 Reglamento sobre Plagas Hidrobiológicas (www.sernapesca.cl).

² Programa ejecutado por el Instituto de Fomento Pesquero.

el comportamiento social de cardúmenes de peces o bandadas de pájaros, denominada Optimización por Enjambre de Partículas (PSO, Particle Swarm Optimization).

En el capítulo 2 se plantean los objetivos y el plan de trabajo correspondientes al proyecto. Luego, en el capítulo 3 se desarrolla la técnica de Máquinas de Soporte Vectorial, tanto para clasificación como para regresión y se explica la variante Máquinas de Soporte Vectorial de Mínimos Cuadrados. Posteriormente, en el capítulo 4 se manifiesta la técnica de Optimización por Enjambre de Partículas junto a algunas variantes. En el capítulo 5 se exponen los elementos del estudio. En el capítulo 6 se presentan las medidas de evaluación utilizadas para evaluar el rendimiento de las distintas configuraciones del modelo. En el capítulo 7 se detalla la estructura del modelo LS-SVM PSO. Luego, en el capítulo 8 se explica la implementación del modelo y se exponen los resultados obtenidos. En el capítulo 9 se analizan los resultados. Finalmente, en el capítulo 10 se presentan las conclusiones.

2 Marco General del Proyecto

2.1 Objetivo General

Desarrollar un modelo de predicción de *Alexandrium Catenella*, para la región de Magallanes, utilizando Máquinas de Soporte Vectorial combinado con algoritmos de Optimización por Enjambre de Partículas.

2.2 Objetivos Específicos

- Explicar el funcionamiento de las Máquinas de Soporte Vectorial (SVM) y Optimización por Enjambre de Partículas (PSO).
- Diseñar la estructura del predictor y estimar sus parámetros utilizando PSO y mínimos cuadrados.
- Evaluar y contrastar el rendimiento de los predictores propuestos.

2.3 Plan de Trabajo

En la tabla 2.1 se describe el plan de trabajo desarrollado durante el proyecto, el cual se divide en dos etapas: la primera consiste en la interiorización del problema junto con la elaboración de un marco teórico, que sirve como base fundamental para la construcción del modelo de pronóstico; también se diseña el modelo y se realizan pruebas preliminares mediante las cuales se logra un primer acercamiento al uso empírico de las técnicas expuestas en el marco teórico. La segunda etapa comienza con la implementación, evaluación y el contraste de las configuraciones del modelo de pronóstico, utilizando cuatro variantes de PSO. Posteriormente, se realizan pruebas para calibrar el modelo con dichas variantes. Finalmente, se realiza el análisis de los resultados y se formulan las conclusiones.

Tabla 2.1 Plan de trabajo.

Actividades	Descripción
<i>Formulación del Tema</i>	<ul style="list-style-type: none"> • Introducción e identificación del contexto del problema. • Recolección de material para la investigación. • Establecimiento de objetivos.
<i>Marco Teórico</i>	<ul style="list-style-type: none"> • Explicar el funcionamiento de las Máquinas de Soporte Vectorial (SVM), Optimización por Enjambre de Partículas (PSO) y algunas variantes. •
<i>Diseño del modelo</i>	<ul style="list-style-type: none"> • Diseñar la estructura del modelo.
<i>Implementación para el cluster “Última Esperanza”</i>	<ul style="list-style-type: none"> • Implementar el modelo de pronóstico para el cluster de 9 estaciones correspondientes al núcleo de toxicidad “Última Esperanza” utilizando PSO-FI, APSO, FDRPSO y AIWPSO. • Evaluar y contrastar las configuraciones propuestas. • Analizar los resultados.
<i>Conclusiones</i>	<ul style="list-style-type: none"> • Conclusiones finales del trabajo realizado.

3 Máquinas de Soporte Vectorial

Las Máquinas de Soporte Vectorial (SVM, Support Vector Machine) [15] [18] [19] [20] son un conjunto de algoritmos de aprendizaje supervisado que buscan encontrar patrones dentro de un grupo de datos generalmente no lineales. Las SVM son el resultado de combinar un clasificador de margen máximo con una función kernel. Los fundamentos teóricos en los que se basan datan de los años setenta y ochenta con los trabajos de Vapnik y otros autores sobre la teoría del aprendizaje estadístico, en la década de los noventa Cortes y Vapnik anuncian la primera formulación de las SVM en su versión actual y posteriormente en la misma década Vapnik y Smola extienden la técnica para su uso como herramienta para regresión.

Algunos trabajos expuestos en la literatura dan luces de la gran cantidad de aplicaciones y diversidad de ámbitos en los que se pueden ocupar las SVM, por ejemplo: en [35] se utilizan las Máquinas de Soporte Vectorial para pronosticar las condiciones de borde (valores de presión, temperatura y flujo en los extremos de las tuberías) necesarias en la simulación del estado hidráulico de la Red Nacional de Gasoductos de Venezuela y mejorar el modelo propuesto por una empresa de servicios. En [36] las SVM se emplean para predecir los costos de producción de la industria de la maquinaria de Taiwan. Los valores de producción de la industria de la maquinaria de Taiwan han estado aumentando continuamente en los últimos años. Por lo tanto, la predicción de los valores de producción es una cuestión esencial para dicha industria. Otro ejemplo se presenta en [37] donde se desarrolló un modelo de predicción para la economía rural usando Máquinas de Soporte Vectorial. El modelo de pronóstico se basa en datos de 14 años sobre la producción de algodón en la provincia de Shanxi.

En 3.1 se explica la técnica para clasificación de las Máquinas de Soporte Vectorial. Luego, en 3.2 se expone la versión para regresión. Por último, en 3.3, se describe una variante de las SVM denominada Máquinas de Soporte Vectorial de Mínimos Cuadrados.

3.1 Máquinas de Soporte Vectorial para Clasificación

Las SVM para clasificación corresponden a la primera formulación de las SVM en donde se busca, por medio de un margen separador, clasificar elementos según una función objetivo. A continuación se expone la técnica explicando de forma evolutiva su concepción.

3.1.1 Caso Linealmente Separable o de Margen Máximo

Dada una muestra de aprendizaje separable $D = \{(x_1, y_1), \dots, (x_i, y_i)\} \subseteq \mathbb{R}^d \times \{1, -1\}$ con $i \in \{1 \dots n\}$, donde x_i es un vector de características e y_i es la clase que representa a dicho vector. Se busca un hiperplano separador $wx + b = 0$, con parámetros $w \in \mathbb{R}^d$ y $b \in \mathbb{R}$. Además, se necesita una función de decisión $h(x_i)$, de modo que dado un vector x_i evaluado en h , este pueda ser asignado a una clase.

$$f(x) = \mathbf{w}x + b$$

$$h(x_i) = y_i = \text{signo}(f(x_i))$$

$$\begin{aligned} \text{si } \mathbf{w}x_i + b > 0, & \quad \text{entonces } y_i = +1 \\ \text{si } \mathbf{w}x_i + b < 0, & \quad \text{entonces } y_i = -1 \end{aligned}$$

Si las clases son linealmente separables se pueden encontrar infinitos hiperplanos separadores. El objetivo de este planteamiento es encontrar un *hiperplano separador óptimo* que maximice la distancia entre las clases representadas por los conjuntos $\{(\mathbf{x}_i, 1)\}$ y $\{(\mathbf{x}_i, -1)\}$. Ver figura 3.1.

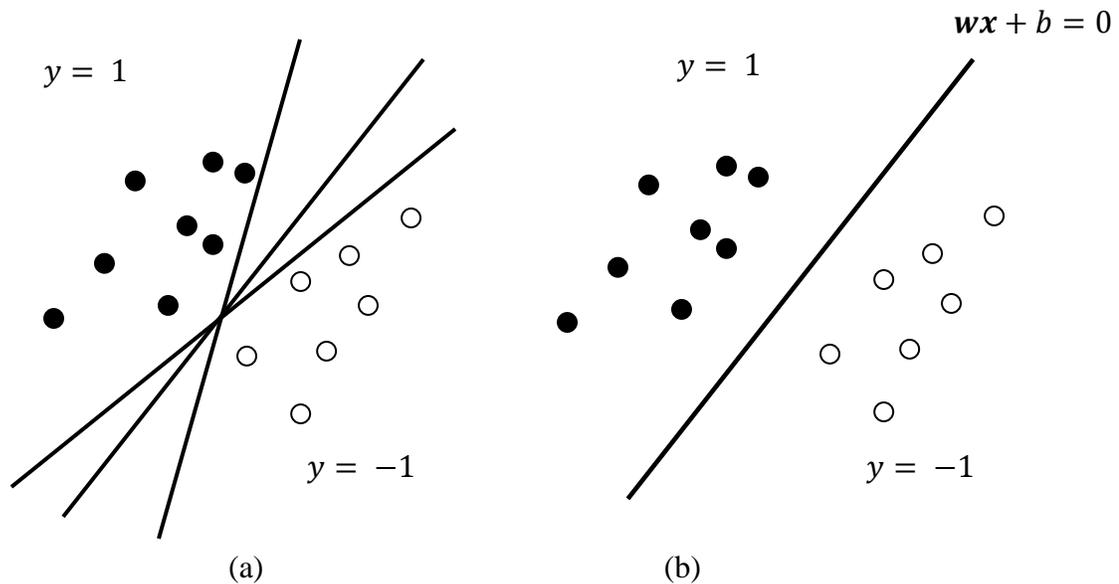


Figura 3.1 Hiperplano separadores.
(a) Infinitos hiperplanos separadores. (b) Hiperplano separador óptimo.

A partir del hiperplano definido por \mathbf{w} y b se precisan dos hiperplanos paralelos (*hiperplanos canónicos*) a este, de tal forma que en los puntos más cercanos al hiperplano se cumpla que $\min_{x_i \in D} |\mathbf{w}x_i + b| = 1$, esto se logra re-escalando adecuadamente \mathbf{w} y b , de tal forma que:

$$\begin{aligned} \mathbf{w}x + b &\geq 1 & \text{para } y_i = 1 \\ \mathbf{w}x + b &\leq -1 & \text{para } y_i = -1 \end{aligned}$$

Estas dos expresiones se pueden combinar en la siguiente desigualdad (ver figura 3.2):

$$y_i(\mathbf{w}x + b) \geq 1 \tag{3.1.1.1}$$

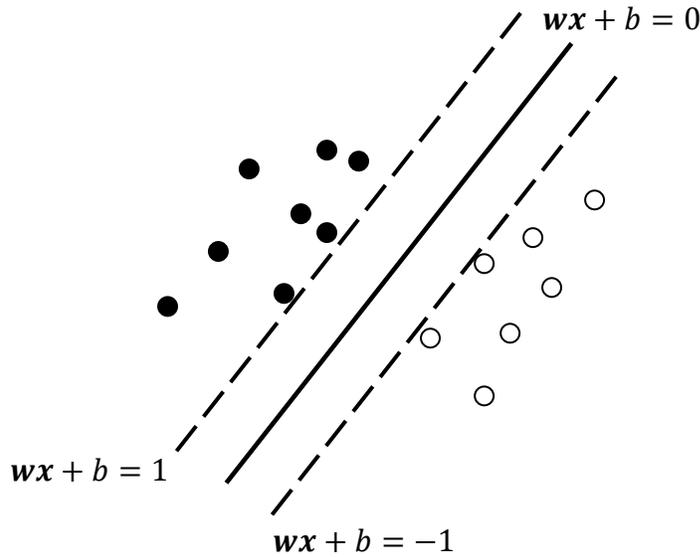


Figura 3.2 Hiperplanos canónicos.

Los hiperplanos canónico reciben el nombre de *hiperplanos de soporte*, mientras que los vectores x_i contenidos en los *hiperplanos de soporte* se denominan *vectores de soporte*. El *margen* corresponde a la distancia existente entre los dos *hiperplanos de soporte* (ver figura 3.3).

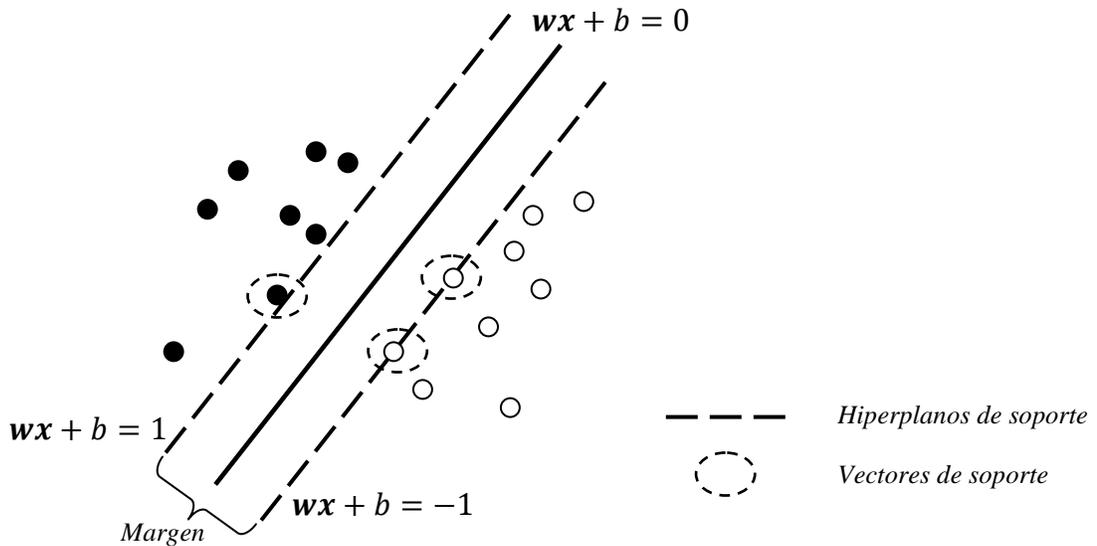


Figura 3.3 Margen, hiperplanos de soporte y vectores de soporte.

A continuación se formula el problema de optimización consistente en encontrar *el hiperplano separador* con margen máximo.

La distancia de un punto x_i (*vector de soporte*) al *hiperplano separador óptimo* $wx + b = 0$ estada dada por:

$$dist(x_i; w, b) = \frac{|wx_i + b|}{\|w\|} \tag{3.1.1.2}$$

El *margen* se calcula sumando las mínimas distancias existentes entre los vectores de soporte y el *hiperplano separador óptimo*. Lo que se describe de la siguiente manera:

$$\begin{aligned}
 M(\mathbf{w}, b) &= \min_{x_i, y_i = -1} \text{dist}(\mathbf{x}_i; \mathbf{w}, b) + \min_{x_i, y_i = 1} \text{dist}(\mathbf{x}_i; \mathbf{w}, b) \\
 M(\mathbf{w}, b) &= \min_{x_i, y_i = -1} \frac{|\mathbf{w}\mathbf{x}_i + b|}{\|\mathbf{w}\|} + \min_{x_i, y_i = 1} \frac{|\mathbf{w}\mathbf{x}_i + b|}{\|\mathbf{w}\|} \\
 M(\mathbf{w}, b) &= \frac{1}{\|\mathbf{w}\|} \left(\min_{x_i, y_i = -1} |\mathbf{w}\mathbf{x}_i + b| + \min_{x_i, y_i = 1} |\mathbf{w}\mathbf{x}_i + b| \right) \\
 M(\mathbf{w}, b) &= \frac{2}{\|\mathbf{w}\|} \tag{3.1.1.3}
 \end{aligned}$$

Para optimizar el margen se puede maximizar 3.1.1.3 o de modo equivalente minimizar $\frac{\|\mathbf{w}\|^2}{2}$.

Finalmente, el problema de optimización a resolver queda expresado de la siguiente manera:

$$\begin{aligned}
 \min_{\mathbf{w}, b} \frac{1}{2} \sum_{i=1}^n \mathbf{w}_i^2 \\
 \text{sujeto a } y_i(\mathbf{w}\mathbf{x}_i + b) \geq 1, \quad \forall i \in \{1 \dots n\} \tag{3.1.1.4}
 \end{aligned}$$

Para presentar la *formulación dual* del problema de optimización 3.1.1.4 se debe convertir el problema, a un *problema de optimización lagrangiano*.

La función de Lagrange correspondiente al problema 3.1.1.4 está dada por:

$$\begin{aligned}
 \mathcal{L}(\alpha, \mathbf{w}, b) &= \frac{1}{2} \sum_{i=1}^n \mathbf{w}_i^2 - \sum_{i=1}^n \alpha_i (y_i(\mathbf{w}\mathbf{x}_i - b) - 1) \\
 \text{donde } \alpha_i &\geq 0, \quad \forall i \in \{1 \dots n\} \tag{3.1.1.5}
 \end{aligned}$$

Los valores α_i corresponden a los *multiplicadores de Lagrange*.

El *problema de optimización lagrangiano* se puede definir como:

$$\begin{aligned}
 \max_{\alpha} \min_{\mathbf{w}, b} \mathcal{L}(\alpha, \mathbf{w}, b) \\
 \text{sujeto a } \alpha_i \geq 0, \quad \forall i \in \{1 \dots n\} \tag{3.1.1.6}
 \end{aligned}$$

Las soluciones de un *problema de optimización lagrangiano* son los puntos sillas de la función, pero como la función objetivo 3.1.1.4 es convexa y las restricciones son lineales $\mathcal{L}(\alpha, \mathbf{w}, b)$ tiene un único punto silla.

La solución al *problema de optimización lagrangiano* es una solución al problema de optimización 3.1.1.4 si y solo si $\mathcal{L}(\alpha, \mathbf{w}, b)$ cumple las *condiciones de Karush-Kuhn-Tucker* (KKT):

$$\frac{\partial \mathcal{L}(\alpha, \mathbf{w}, b)}{\partial \mathbf{w}} = 0 \quad (3.1.1.7)$$

$$\frac{\partial \mathcal{L}(\alpha, \mathbf{w}, b)}{\partial b} = 0 \quad (3.1.1.8)$$

$$\alpha_i (y_i (\mathbf{w}_i \mathbf{x}_i - b_i) - 1) = 0, \quad \forall i \in \{1 \dots n\} \quad (3.1.1.9)$$

$$y_i (\mathbf{w}_i \mathbf{x}_i - b_i) - 1 \geq 0, \quad \forall i \in \{1 \dots n\} \quad (3.1.1.10)$$

$$\alpha_i \geq 0, \quad \forall i 1 \dots n \quad (3.1.1.11)$$

Las condiciones 3.1.1.7 y 3.1.1.8 aseguran que la solución al problema de optimización 3.1.1.6 constituye el único punto silla de $\mathcal{L}(\alpha, \mathbf{w}, b)$. La condición 3.1.1.9 garantiza que la solución del *problema de optimización lagrangiano* coincide con la solución del problema de optimización 3.1.1.4. Finalmente, las condiciones 3.1.1.10 y 3.1.1.11 corresponden a las restricciones de los problemas de optimización 3.1.1.4 y 3.1.1.6 respectivamente.

De la condición 3.1.1.7 se obtiene:

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \quad (3.1.1.11)$$

De la condición 3.1.1.8 se consigue:

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad (3.1.1.12)$$

b se puede obtener utilizando un vector de soporte \mathbf{x}_s el cual es evaluado junto a \mathbf{w} en la ecuación referente los *hiperplanos de soporte* definida como $y_s (\mathbf{w} \mathbf{x}_s + b) - 1 = 0$, resultando la siguiente expresión:

$$\mathbf{b} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \mathbf{x}_s - y_s \quad (3.1.1.13)$$

Utilizando 3.1.1.11 y 3.1.1.12 en 3.1.1.5 se logra:

$$\mathcal{L}(\alpha) = \sum_1^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j \quad (3.1.1.14)$$

Finalmente, la *formulación dual del problema de optimización* para el **caso linealmente separable** queda expresada de la siguiente manera:

$$\max_{\alpha} \sum_1^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i x_j$$

sujeto a:

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

$$\alpha_i \geq 0, \quad \forall i \ 1 \dots n \quad (3.1.1.15)$$

Analizando las condiciones 3.1.1.9 y 3.1.1.11 se pueden obtener las siguientes conclusiones:

- Si $\alpha_i > 0$, significa que aquellos vectores x_i que tienen un *multiplicador de Lagrange* positivo están sobre su correspondiente *hiperplano de soporte*, y en consecuencia son *vectores de soporte*.
- Si $\alpha_i = 0$, significa que aquellos vectores x_i que tienen un *multiplicador de Lagrange* nulo no se encuentran sobre su *hiperplano de soporte*.

3.1.2 Caso Linealmente no Separable o de Margen Flexible

El caso linealmente no separable o de margen flexible se basa en la idea de que el clasificador puede cometer errores. Esto permite que el modelo sea más general y de esa manera se disminuye el riesgo de que se sobreajuste. Este planteamiento corresponde a una generalización del caso linealmente separable.

Para construir el *hiperplano óptimo* se introducen variables de holgura $\xi_i \geq 0$, lo cual permite que algunos vectores de la muestra de aprendizaje se encuentren mal clasificados, y una penalización $C > 0$, denominado *parámetro de coste*, que regula el margen para los puntos correctamente clasificados.

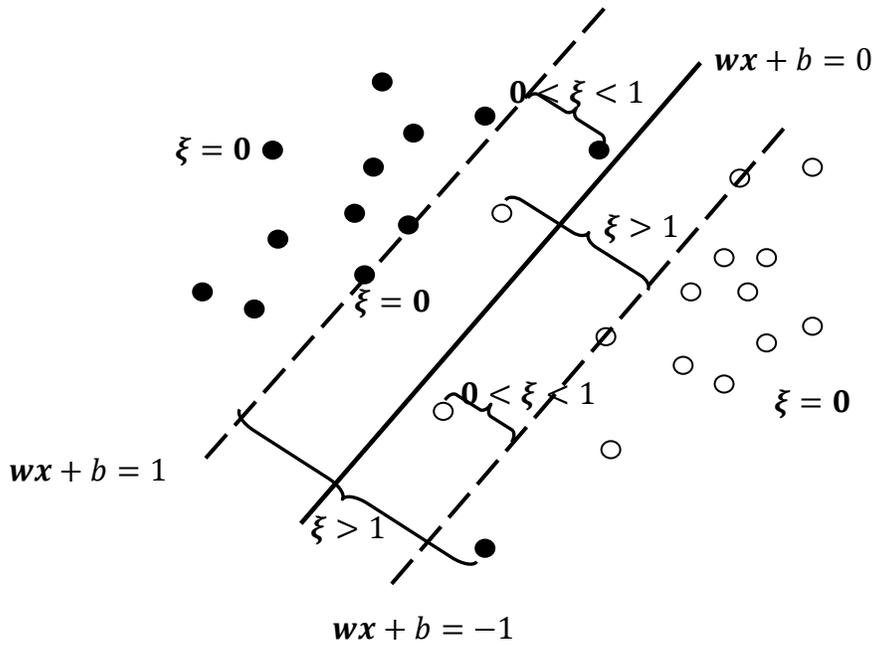


Figura 3.4 Caso linealmente no separable.

El nuevo problema de optimización se puede expresar extendiendo la primera formulación del caso linealmente separable.

Sea $D = \{(x_1, y_1), \dots, (x_i, y_i)\} \subseteq \mathbb{R}^d \times \{1, -1\}$ una muestra de aprendizaje con $i = 1 \dots n$, entonces el problema de optimización queda planteado de la siguiente manera:

$$\min_{w, b, \xi} \frac{1}{2} \sum_{i=1}^n w_i^2 + C \sum_{i=1}^n \xi_i$$

sujeto a:

$$y_i(w x_i + b) \geq 1 - \xi_i, \quad \forall i \in 1 \dots n$$

$$\xi_i \geq 0, \quad \forall i \in 1 \dots n \tag{3.1.2.1}$$

Interpretación del parámetro C:

- Si C es muy grande, significa que se le está asignando un costo muy alto a los errores cometidos.
- Un valor pequeño de C , significa que el costo de cometer errores es pequeño.

Interpretación de la variable ξ (ver figura 3.4):

- Si $\xi_i = 0$ implica que el vector x_i está en el lado correcto de su *hiperplano de soporte*.
- Si $0 < \xi_i \leq 1$, implica que el vector x_i está en el lado incorrecto de su *hiperplano de soporte*, pero en el lado correcto del *hiperplano separador*.
- Si $\xi_i > 1$, implica que el vector x_i está en el lado incorrecto de la *superficie óptima de decisión* y naturalmente en el lado incorrecto de su *hiperplano de soporte*.

Dado el problema de optimización 3.1.2.1 se puede construir la *función lagrangiana*:

$$\mathcal{L}(\alpha, \mu, \mathbf{w}, b, \xi) = \frac{1}{2} \mathbf{w} \mathbf{w} + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i (y_i (\mathbf{w} \mathbf{x}_i + b) + \xi_i - 1) - \sum_{i=1}^n \mu_i \xi_i \quad (3.1.2.2)$$

\mathbf{w}, b, ξ corresponden a las variables del problema de optimización 3.1.2.1 y α, μ son los *multiplicadores de Lagrange* o variables duales. La *formulación dual* del problema de optimización 3.1.2.1 queda expresada de la siguiente manera:

$$\begin{aligned} & \max_{\alpha, \mu} \min_{\mathbf{w}, b, \xi} \mathcal{L}(\alpha, \mu, \mathbf{w}, b, \xi) \\ & \text{sujeto a } \alpha_i, \mu_i \geq 0, \quad \forall i \in \{1 \dots n\} \end{aligned} \quad (3.1.2.3)$$

La solución del problema de *optimización Lagrangiano*, al igual que en la formulación dual del caso linealmente separable, debe satisfacer las condiciones KKT, con las cuales se puede obtener las necesidades de optimalidad del problema y asegurar que dada una solución óptima del problema 3.1.2.3, esta será también una solución óptima para el problema 3.1.2.1.

Las necesidades de optimalidad del problema se obtienen igualando a cero las derivadas parciales $\frac{\partial \mathcal{L}}{\partial \mathbf{w}}$, $\frac{\partial \mathcal{L}}{\partial b}$ y $\frac{\partial \mathcal{L}}{\partial \xi}$, desde las cuales se rescata:

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \quad (3.1.2.4)$$

$$\sum_{i=1}^n \alpha_i y_i x_i = 0 \quad (3.1.2.5)$$

$$C - \alpha_i - \mu_i = 0 \quad \forall i \in \{1 \dots n\} \quad (3.1.2.6)$$

Dada las expresiones 3.1.2.4, 3.1.2.5 y 3.1.2.6 y sustituidas en 3.1.2.2 se puede determinar que la *formulación dual del problema de optimización* 3.1.1.16 es:

$$\max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i x_j$$

sujeto a:

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

$$0 \leq \alpha_i \leq C, \quad \forall i \in \{1 \dots n\} \quad (3.1.2.7)$$

b se puede obtener utilizando un vector de soporte \mathbf{x}_s el cual es evaluado junto a \mathbf{w} en la ecuación referente los *hiperplanos de soporte* definida como $y_s (\mathbf{w} \mathbf{x}_s + b) - 1 = 0$, resultando la siguiente expresión:

$$\mathbf{b} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \mathbf{x}_s - y_s \quad (3.1.2.8)$$

Basado en las siguientes condiciones KKT para el problema de optimización lagangriano:

$$\alpha_i, \mu_i \geq 0, \quad \forall i \in \{1 \dots n\}$$

$$C - \alpha_i - \mu_i = 0 \quad \forall i \in \{1 \dots n\}$$

$$\mu_i \xi_i = 0 \quad \forall i \in \{1 \dots n\}$$

se puede interpretar el parámetro α de la siguiente manera:

- Si $\alpha_i = 0$, el vector \mathbf{x}_i se encuentra situado en su correspondiente *hiperplano de soporte* y no participa en la determinación de la superficie optima.
- Si $0 < \alpha_i < C$, el vector \mathbf{x}_i se encuentra situado en su correspondiente *hiperplano de soporte* y corresponde a un *vector de soporte*.
- Si $\alpha_i = C$, el vector \mathbf{x}_i se encuentra incorrectamente clasificado.

3.1.3 Funciones Kernel

Las funciones Kernel tienen como objetivo modificar la representación del conjunto de datos original de manera que sea sencillo identificar relaciones de tipo lineal. Estas funciones son esenciales en la formalización de las Máquinas de Soporte Vectorial.

Sea M un conjunto de datos de la forma $M = \{x_1, \dots, x_i\}$ con $x_i \in X, \forall i \in \{1, \dots, i\}$ y $X \subset \mathbb{R}^n$ y sea \wp una función de proyección que lleva a los elementos de M a un *espacio de características* \mathcal{U} (ver figura 3.5):

$$\wp: M \rightarrow \mathcal{U} \quad (3.1.3.1)$$

El espacio de características \mathcal{U} debe ser un espacio de Hilbert, o sea un espacio de dimensión infinita o isomorfo a \mathbb{R}^N en $N > n$.

La transformación de los elementos del conjunto de datos M está dada por:

$$\wp(M) = \{\wp(x_1), \dots, \wp(x_i)\} \quad (3.1.3.2)$$

El problema existente se basa en la necesidad de evaluar los elementos del conjunto M en el espacio de características \mathcal{U} para luego calcular los productos escalares $\langle \wp(x_n), \wp(x_m) \rangle$.

Para resolver dicho problema se debe determinar una función $K: X \times X \rightarrow \mathbb{R}$ que efectúe simultáneamente la operación $\wp: M \rightarrow \mathcal{U}$ y calcule los productos escalares de las imágenes $\langle \wp(x_n), \wp(x_m) \rangle$, o sea $K(x_n, x_m) = \langle \wp(x_n), \wp(x_m) \rangle$, a K se le denomina *función Kernel*.

Entre las funciones Kernel mas utilizadas están:

$$\text{Kernel Lineal:} \quad K(x, x_k) = x_k^T x \quad (3.1.3.3)$$

$$\text{Kernel Polinomial:} \quad K(x, x_k) = (x_k^T x + 1)^d \quad (3.1.3.4)$$

$$\text{Kernel Sigmoidal:} \quad K(x, x_k) = \tan[sx_k^T x + \theta] \quad (3.1.3.5)$$

$$\text{Kernel Función de Base Radial:} \quad K(x, x_k) = \exp\left\{\frac{-\|x - x_k\|^2}{\sigma^2}\right\} \quad (3.1.3.6)$$

Dependiendo del Kernel que se desee utilizar se deben ajustar sus respectivos parámetros: d para el Kernel Polinomial, s y θ para el Kernel Sigmoidal y σ para el Kernel Función de Base Radial; el Kernel Lineal no utiliza parámetros de ajuste.

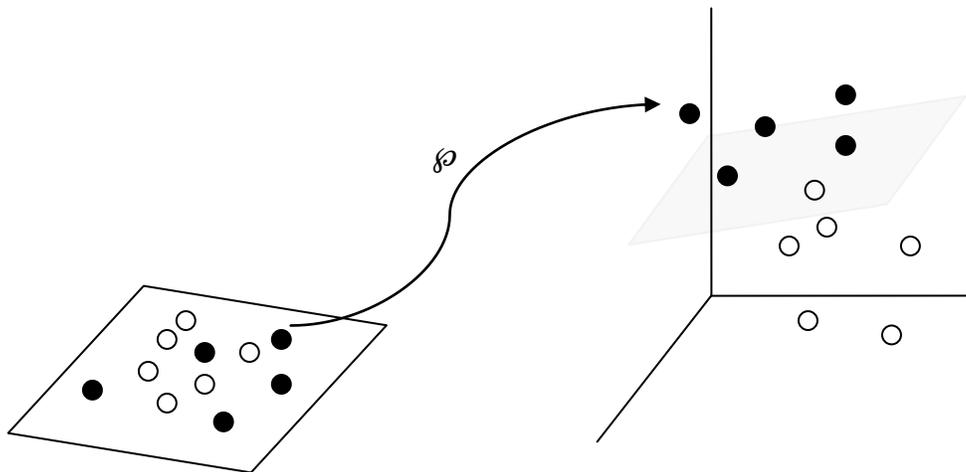


Figura 3.5 Proyección de puntos al espacio de características.

3.1.4 Caso no Lineal

A la formulación 3.1.2.7 del caso linealmente no separable se le puede incorporar una función Kernel para proyectar los datos del conjunto de entrenamiento a un espacio de mayor dimensión donde se puedan encontrar patrones lineales, que son imposibles de descubrir en su espacio original. La figura 3.6 grafica los tres tipos de casos en estudio.

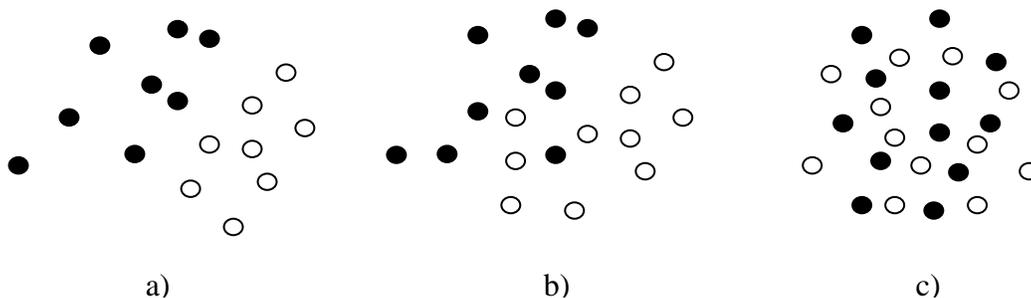


Figura 3.6 Casos en análisis de las SVM.

a) Caso linealmente separable. b) Caso linealmente no separable. c) Caso no lineal.

Para introducir la función Kernel se puede sustituir el producto $x_i x_j$ de 3.1.2.7 por la función $K(x_i, x_j)$ entonces el problema de optimización para el caso no lineal queda representado por:

$$\max_{\alpha} \sum_1^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

sujeto a:

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

$$0 \leq \alpha_i \leq C, \quad \forall i \in \{1 \dots n\}$$

$$\alpha_i \geq 0 \quad \forall i \in \{1 \dots n\}$$

El valor de \mathbf{w} (3.1.2.4) puede escribirse de la siguiente forma:

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \phi(\mathbf{x}_i)$$

El valor de b (3.1.1.13) se expresa como:

$$\mathbf{b} = \sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_s) - y_s$$

Finalmente, la función de decisión se puede reescribir como:

$$h(x) = \text{signo} \left(\sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \right), \quad \text{donde } i \dots n \text{ son vectores de soporte}$$

En la figura 3.7 se muestra un esquema general de las Máquinas de Soporte Vectorial.

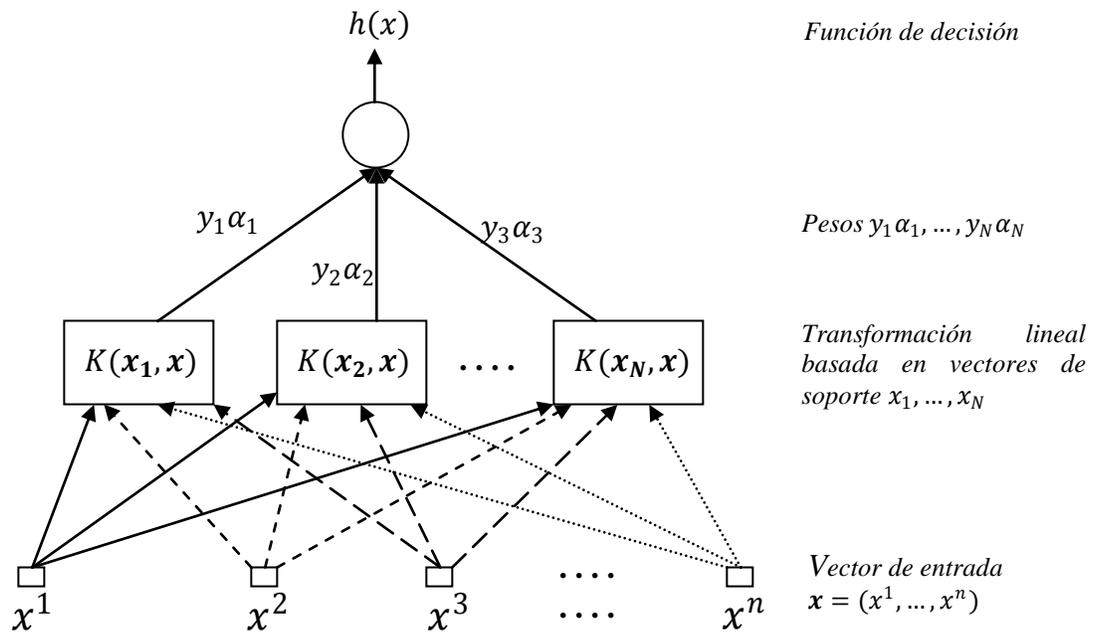


Figura 3.7 Esquema general Máquinas de Soporte Vectorial.

3.2 Máquinas de Soporte Vectorial para Regresión

Las Máquinas de Soporte Vectorial para regresión, mantienen las principales características de las SVM para clasificación; formular un problema de optimización que permita maximizar un margen y luego generar una formulación dual del problema que admita incrustar una función Kernel.

Dada una muestra de aprendizaje $D = \{(x_1, y_1), \dots, (x_i, y_i)\} \subseteq \mathbb{R}^d \times \mathbb{R}$ con $i \in \{1 \dots n\}$, donde \mathbf{x}_i es un vector correspondiente al conjunto de variables predictivas e y_i es la variable dependiente que se busca predecir. El objetivo es encontrar una función $f(x) = \mathbf{w}\mathbf{x} + b$ con parámetros $\mathbf{w} \in \mathbb{R}^d$ y $b \in \mathbb{R}$ que produzca un pronóstico $f(x_i)$ el cual no debe desviarse más allá de una cantidad ϵ respecto a su verdadero valor observado y_i , en otras palabras ϵ es la máxima desviación permitida para los datos de la muestra. Ver Figura 3.8.

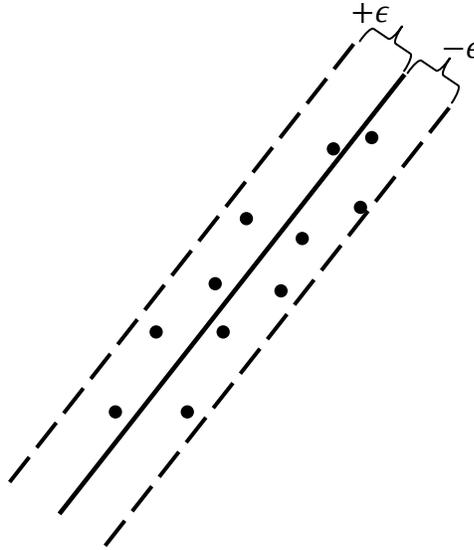


Figura 3.8 Máxima desviación

Como la desviación entre y_i y $f(x_i)$ debe ser a lo más ϵ , se obtienen las siguientes restricciones:

$$\begin{aligned} y_i - (\mathbf{w}\mathbf{x}_i + b) &\leq \epsilon, & \forall i \in D \\ (\mathbf{w}\mathbf{x}_i + b) - y_i &\leq \epsilon, & \forall i \in D \end{aligned}$$

El problema de optimización convexa está dado por:

$$\min_{\mathbf{w}, b} \frac{1}{2} \sum_{i=1}^n \mathbf{w}_i^2$$

sujeto a

$$y_i - (\mathbf{w}\mathbf{x}_i + b) \leq \epsilon, \quad \forall i \in D$$

$$(\mathbf{w}\mathbf{x}_i + b) - y_i \leq \epsilon, \quad \forall i \in D$$

(3.2.1)

En la formulación 3.2.1 se asume una función f que aproxima todos los pares (x_i, y_i) con ϵ precisión. Sin embargo, puede que este no siempre se dé. De manera análoga al caso *linealmente no separable - margen flexible* dado para clasificación se pueden introducir variables de holgura ξ_i, ξ_i^* , que flexibilizaran las restricciones de 3.2.1 permitiendo errores y una variable de costo C que representa el equilibrio entre la pendiente de la función de predicción y las desviaciones mayores a ϵ . El problema de optimización queda expresado de la siguiente manera:

$$\min_{w,b,\xi,\xi^*} \frac{1}{2} \sum_{i=1}^d \mathbf{w}_i^2 + C \sum_{i=1}^n \xi_i + \xi_i^*$$

sujeto a:

$$y_i - (\mathbf{w}x_i + b) \leq \epsilon + \xi_i, \quad \forall i \in D$$

$$(\mathbf{w}x_i + b) - y_i \leq \epsilon + \xi_i^*, \quad \forall i \in D$$

$$\xi_i, \xi_i^* \geq 0, \quad \forall i \in D \quad (3.2.2)$$

La desviación de los valores se puede evaluar utilizando una función de pérdida llamada $\epsilon - intensive$ ($|\xi|_\epsilon$). En la figura 3.9 se aprecia el tubo formado por ϵ , el cual se denomina tubo $\epsilon - intensive$. La función $\epsilon - intensive$ se describe como:

$$|\xi|_\epsilon = \begin{cases} 0, & \text{si } |\xi| \leq \epsilon \\ |\xi| - \epsilon, & \text{si } |\xi| > \epsilon \end{cases} \quad (3.2.3)$$

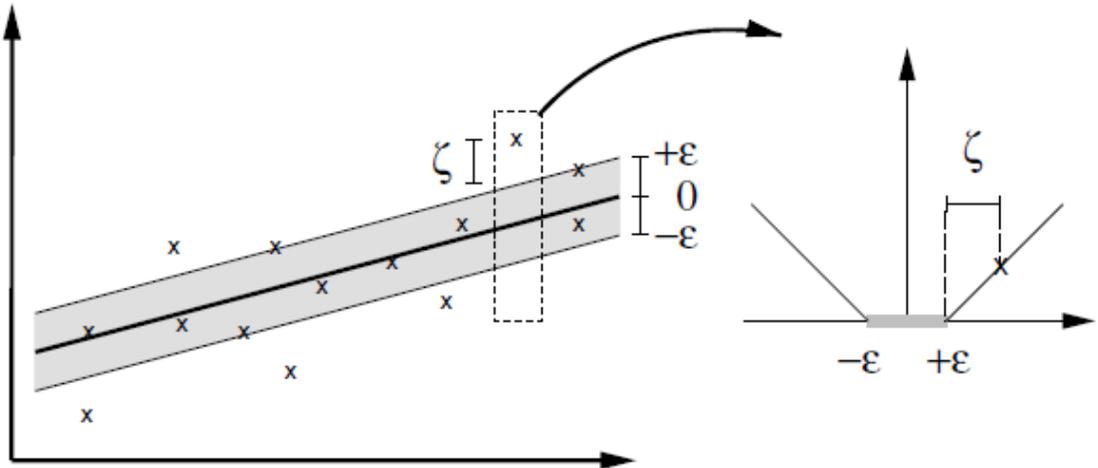


Figura 3.9 Función de pérdida para un caso lineal [17].

La *formulación dual* de problema 3.2.2, corresponde a construir una función de Lagrange basada en la función objetivo y sus respectivas restricciones. La función Lagrangiana queda de la siguiente forma:

$$\begin{aligned} \mathcal{L}(\alpha, \mu, \mathbf{w}, b, \xi, \xi^*) &= \frac{1}{2} \sum_{i=1}^d \mathbf{w}_i^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) - \sum_{i=1}^n (\mu_i \xi_i + \mu_i^* \xi_i^*) \\ &\quad - \sum_{i=1}^n \alpha_i (\epsilon + \xi_i - y_i + \langle \mathbf{w}, x_i \rangle + b) \\ &\quad - \sum_{i=1}^n \alpha_i^* (\epsilon + \xi_i^* - y_i + \langle \mathbf{w}, x_i \rangle - b) \end{aligned} \quad (3.2.4)$$

$\alpha, \mu, \alpha^*, \mu^* \geq 0$, son los multiplicadores de Lagrange.

Las condiciones de optimalidad de la función \mathcal{L} están dadas por:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^n (\alpha_i^* - \alpha_i) \mathbf{x}_i = 0 \quad (3.2.5)$$

$$\frac{\partial \mathcal{L}}{\partial b} = \sum_{i=1}^n (\alpha_i^* - \alpha_i) = 0 \quad (3.2.6)$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \quad (3.2.7)$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i^*} = C - \alpha_i^* - \mu^* = 0 \quad (3.2.8)$$

De 3.2.5 se puede obtener que:

$$\mathbf{w} = \sum_{i=1}^n (\alpha_i^* + \alpha_i) \mathbf{x}_i \quad (3.2.9)$$

$$\sum_{i=1}^n \alpha_i^* = \sum_{i=1}^n \alpha_i \quad (3.2.10)$$

$$0 \leq \alpha_i, \alpha_i^* \leq C \quad (3.2.11)$$

Reemplazando 3.2.9 y utilizando las restricciones 3.2.11 en 3.2.4 se puede obtener la formulación dual del problema de optimización convexa en las variables α_i, α_i^* .

$$\begin{aligned} \max_{\alpha_i, \alpha_i^*} & -\frac{1}{2} \sum_{i,j=1}^n (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ & - \epsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*) \end{aligned}$$

sujeto a

$$\begin{aligned} \sum_{i=1}^n (\alpha_i^* - \alpha_i) &= 0 \\ 0 &\leq \alpha_i, \alpha_i^* \leq C \end{aligned} \quad (3.2.12)$$

Al cumplirse las siguientes condiciones KKT se puede asegurar que la solución del problema dual corresponde a una solución para el problema 3.2.2.

$$\alpha_i (y_i - \epsilon - \xi_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b) = 0 \quad (3.2.13)$$

$$\alpha_i^*(y_i - \epsilon - \xi_i^* - \langle \mathbf{w}, x_i \rangle - b) = 0 \quad (3.2.14)$$

$$\xi_i(C - \alpha_i) = 0 \quad (3.2.15)$$

$$\xi_i^*(C - \alpha_i) = 0 \quad (3.2.16)$$

$$0 \leq \alpha_i, \alpha_i^* \leq C \quad (3.2.17)$$

Al realizar un análisis de b en función de las condiciones KKT, se obtienen los siguientes intervalos en donde b puede ser una solución optima.

$$\begin{aligned} \max\{-\epsilon + y_i - \langle \mathbf{w}, x_i \rangle \mid \alpha_i < C \text{ o } \alpha_i^* > 0\} &\leq b \\ &\leq \min\{-\epsilon + y_i - \langle \mathbf{w}, x_i \rangle \mid \alpha_i > 0 \text{ o } \alpha_i^* < C\} \end{aligned} \quad (3.2.18)$$

*Interpretación de α_i y α_i^**

- Si $\alpha_i = C$ o $\alpha_i^* = C$, significa que el elemento x_i se encuentra fuera del tubo $\epsilon - insensitive$.
- Si $\alpha_i = 0$ y $\alpha_i^* = 0$, significa que el elemento x_i se encuentra dentro del tubo $\epsilon - insensitive$.
- Si $0 < \alpha_i < C$ o $0 < \alpha_i^* < C$, significa que el elemento x_i se encuentra en la frontera del tubo $\epsilon - insensitive$.

También se debe tener en cuenta que con $\epsilon > 0$ se debe cumplir que $\alpha_i * \alpha_i^* = 0$, debido a que un elemento no puede ser un vector de soporte para ambos lados del tubo al mismo tiempo.

Finalmente, se puede extender la formulación 3.2.12 (*formulación dual*) para generar una representación no lineal basada en funciones Kernel, obteniendo el siguiente problema de optimización:

$$\begin{aligned} \max_{\alpha_i, \alpha_i^*} & -\frac{1}{2} \sum_{i,j=1}^n (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(x_i, x_j) \\ & - \epsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*) \end{aligned}$$

sujeto a

$$\sum_{i=1}^n (\alpha_i^* - \alpha_i) = 0$$

$$0 \leq \alpha_i, \alpha_i^* \leq C \quad (3.2.19)$$

La ecuación 3.2.9 puede ser reescrita de la siguiente forma:

$$\mathbf{w} = \sum_{i=1}^n (\alpha_i^* + \alpha_i) \wp(\mathbf{x}_i) \quad (3.2.20)$$

Donde \wp es la función de proyección descrita en 3.1.3.1.

La expresión 3.2.18 que explica a la variable b se puede reescribir de la siguiente manera:

$$\begin{aligned} & \max \left\{ -\epsilon + y_i - \sum_{i,j=1}^n (\alpha_i^* + \alpha_i) K(\mathbf{x}_i, \mathbf{x}_j) \mid \alpha_i < C \text{ o } \alpha_i^* > 0 \right\} \leq b \\ & \leq \min \left\{ -\epsilon + y_i - \sum_{i,j=1}^n (\alpha_i^* + \alpha_i) K(\mathbf{x}_i, \mathbf{x}_j) \mid \alpha_i > 0 \text{ o } \alpha_i^* < 0 \right\} \end{aligned} \quad (3.2.21)$$

Por último, se representa la función de predicción como se aprecia en 3.2.22:

$$f(\mathbf{x}) = \sum_{i=1}^n (\alpha_i^* + \alpha_i) K(\mathbf{x}_i, \mathbf{x}) + b \quad (3.2.22)$$

3.3 Máquinas de Soporte Vectorial de Mínimos Cuadrados

Las Máquinas de Soporte Vectorial de Mínimos Cuadrados (LS-SVM, Least Squares – Support Vector Machine) [14][21] corresponden a una modificación de las SVM desarrolladas por Vapnik. Las LS-SVM buscan simplificar la formulación original sin perder sus ventajas. Su propuesta consiste en solucionar el problema de optimización resolviendo un sistema de ecuaciones lineales, lo cual resulta más simple de desarrollar que resolver el problema por medio de programación cuadrática. El planteamiento original de las SVM se ve modificada en dos aspectos: una función de pérdida cuadrática es utilizada para medir el error y las inecuaciones asociadas a las restricciones se modifican por ecuaciones.

El problema de optimización LS-SVM se formula de la siguiente forma:

$$\begin{aligned} & \min_{w,b,e} J(w, e) = \frac{1}{2} w^T w + C \frac{1}{2} \sum_{i=1}^n e_i^2 \\ & \text{sujeto a:} \\ & y_i [w^T \phi(x_i) + b] = 1 - e_i, \quad i = 1, \dots, n \end{aligned} \quad (3.3.1)$$

El valor 1 a la derecha de la restricción es considerado como un valor objetivo y no como un umbral. Sobre este valor objetivo se plantea una variable e_i que representa la tolerancia permitida para cometer errores de clasificación.

El clasificador en su formulación primal toma la siguiente forma:

$$y(x) = \text{sign}[w^T \phi(x) + b] \quad (3.3.2)$$

Donde $x \in \mathbb{R}^n$ e $y \in \mathbb{R}$; $\phi(\cdot): \mathbb{R}^n \rightarrow \mathbb{R}^h$ es la función de proyección a un espacio de características de dimensión alta y potencialmente infinita.

Cuando w toma una dimensión infinita, el problema 3.3.1 no se puede resolver con la formulación primal. Por lo tanto, se debe formular el problema dual para construir un clasificador LS-SVM no lineal.

El Lagrangiano para el problema 3.3.1 es:

$$\mathcal{L}(w, b, e; \alpha) = J(w, e) - \sum_{i=1}^n \alpha_i (y_i [w^T \phi(x_i) + b] - 1 + e_i) \quad (3.3.3)$$

Donde α_i son los multiplicadores de Lagrange.

Calculando las derivadas parciales del Lagrangiano con respecto a cada una de sus variables e igualando a cero se obtienen las condiciones de optimalidad:

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \rightarrow w = \sum_{i=1}^n \alpha_i y_i \phi(x_i) \quad (3.3.4)$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \rightarrow \sum_{i=1}^n \alpha_i y_i = 0 \quad (3.3.5)$$

$$\frac{\partial \mathcal{L}}{\partial e_i} = 0 \rightarrow \alpha_i = c e_i, \quad i = 1, \dots, n \quad (3.3.6)$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \rightarrow y_i [w^T \phi(x_i) + b] - 1 + e_i = 0, \quad i = 1, \dots, n \quad (3.3.7)$$

Luego de eliminar las variables w y e se consigue el siguiente sistema KKT, resuelto en b, α :

$$\begin{pmatrix} 0 & y^T \\ y & \Omega + \frac{I}{c} \end{pmatrix} \begin{pmatrix} b \\ \alpha \end{pmatrix} = \begin{pmatrix} 0 \\ \vec{1} \end{pmatrix} \quad (3.3.8)$$

donde:

$$Z^T = [\phi(x_1)^T y_1; \dots; \phi(x_n)^T y_n]$$

$$y = [y_1; \dots; y_n]$$

$$\vec{1} = [1; \dots; 1]$$

$$\alpha = [\alpha_1; \dots; \alpha_n]$$

La función Kernel es aplicada de la siguiente forma:

$$\Omega = Z^T Z$$

$$\Omega_{il} = y_k y_l \phi(x_i)^T \phi(x_l) = y_k y_l K(x_k, x_l), \quad k, l = 1, \dots, n.$$

Finalmente, el clasificador en su formulación dual corresponde a:

$$y(x) = \text{sign} \left[\sum_{i=1}^n \alpha_i y_i K(x, x_i) + b \right] \quad (3.3.9)$$

Donde α_i, b son las soluciones al sistema de ecuaciones lineales 3.3.8

La principal desventaja es que ningún multiplicador de Lagrange toma el valor de 0, por lo tanto el modelo depende de todos los datos de entrenamiento.

3.4 Ventajas y Desventajas de las SVM

A continuación se listan las principales ventajas de las Máquinas de Soporte Vectorial:

1. *Permiten manejar el trade-off* entre el sobre-ajuste y la sobre-generalización del modelo. Esto es posible modificando el parámetro C . Un modelo sobre-ajustado se adapta muy bien al conjunto de datos de entrenamiento, pero pierde la capacidad de clasificar o pronosticar correctamente nuevos eventos. Por otro lado la sobre-generalización permite que se cometan mayores errores. Lo ideal es encontrar un valor C que permita una comunión entre dichas características.
2. *Para construir el hiperplano separador solo se necesitan los vectores de soporte*, lo que posibilita la eliminación del resto de los vectores del conjunto de entrenamiento.
3. *Al incorporar funciones Kernel al caso linealmente separable*, se puede utilizar la técnica tanto para conjuntos de datos lineales como no lineales.

Entre las principales desventajas de la técnica están:

1. *Existe un alto costo computacional* al momento de resolver el problema de optimización, utilizando técnicas de programación cuadrática, debido al elevado número de restricciones.
2. *Es necesario ajustar el parámetro C del trade-off y los parámetros asociados a la función Kernel*, con el fin de conseguir la configuración que entregue los mejores resultados. Por lo tanto, es necesario utilizar una técnica de optimización, para encontrar los valores más adecuados. También es preciso realizar pruebas empíricas para escoger el Kernel que mejor se adapte al conjunto de datos de entrenamiento.

4 Optimización por Enjambres de Partículas

Optimización por Enjambre de Partículas (PSO, Particle Swarm Optimization) [23] [24] [25] [26] [27] [28] es una metaheurística evolutiva basada en poblaciones, propuesta por James Kennedy and Russell Eberhart en 1995 [22], la cual se inspira en el comportamiento social del vuelo de las bandadas de aves o el movimiento de los bancos de peces y en la teoría de enjambres (swarming theory). PSO, establece un procedimiento de búsqueda basado en una población o enjambre en la que los individuos, llamados *partículas* “vuelan” dentro de un espacio de búsqueda y durante el vuelo cada partícula evoluciona ajustando su posición de acuerdo a su propia experiencia (*memoria autobiográfica*) y de acuerdo a la experiencia compartida por el enjambre que retiene la mejor posición obtenida hasta el momento (*conocimiento social*), para en definitiva, encontrar un resultado lo más óptimo posible para el objetivo propuesto.

En la literatura se pueden encontrar muchas aplicaciones de PSO, por ejemplo, en [38] se utiliza la optimización por enjambre de partículas para estimar la función de generación de calor de un aparato de placa caliente con guarda (APCG). Este aparato sirve para la determinación de la conductividad térmica de materiales aislantes. Otro trabajo [39] presenta la experiencia en la aplicación de la técnica Optimización por Enjambre de Partículas en el diseño de filtros corrugados en guía de onda. Los resultados obtenidos indican que esta técnica permite realizar diseños que se adaptan a una máscara de filtrado predeterminada con una alta convergencia hacia la solución óptima.

La distribución del capítulo se observa de la siguiente manera, en 4.1 se describen los principios de PSO, en 4.2 se formaliza el concepto y se explica el algoritmo. Finalmente, en 4.3 se presentan algunas variantes de PSO.

4.1 Principios

PSO se adhiere a los cinco principios básicos de la “inteligencia de enjambres” (swarm intelligence) propuestos por Millonas [29]. El primero corresponde a **proximidad**: la población debiera ser capaz de realizar cálculos sencillos de espacio y tiempo. PSO realiza una serie de cálculos para modificar la posición de las partículas. El segundo principio es **calidad**: la población debe tener la capacidad para responder a factores de calidad en el ambiente. En PSO los factores de calidad están relacionados con la memoria autobiográfica de las partículas y con el conocimiento social del enjambre. El tercer principio corresponde a la **diversidad de respuesta**: el grupo no debería asignar todos sus recursos a lo largo de líneas demasiado estrechas. Se deben distribuir los recursos como un seguro contra el cambio repentino debido a las fluctuaciones ambientales. En PSO se logra obtener una diversidad de recursos gracias a la existencia de una memoria autobiográfica para cada partícula y del conocimiento social del enjambre. El cuarto principio es la **estabilidad**: el grupo no debe cambiar su comportamiento de un modo a otro, en cada fluctuación del medio ambiente, si es que no vale la pena el gasto de energía. En PSO el grupo cambia su comportamiento solo cuando cambia conocimiento social. Finalmente, el quinto principio corresponde a **adaptabilidad**: el grupo debe ser capaz de cambiar si es que existe una recompensa por la cual valga la pena la inversión de energía. En PSO el grupo no cambia su comportamiento mientras no cambie el conocimiento social.

4.2 Formalización de PSO

En PSO, un candidato a solución es presentado como una *partícula*. Existe un conjunto de partículas que están en permanente movimiento (cada una representa una posible solución) dentro de un área de búsqueda, intentando encontrar un área prometedora en donde puedan conseguir un óptimo global. Cada individuo o partícula en el enjambre está representado por un vector en el espacio de búsqueda multidimensional. Dicho vector tiene asociado otro vector que determina el siguiente movimiento de la partícula denominado *vector de velocidad*. El algoritmo PSO también determina cómo actualizar la velocidad de una partícula. Cada partícula actualiza su velocidad basándose en su velocidad actual, en su mejor posición encontrada hasta el momento, y en la mejor posición global explorada por el enjambre. PSO contempla un proceso iterativo, el cual se detiene luego de un número fijo de veces o hasta que se alcance un error mínimo deseado basado en algún índice de desempeño.

Formalmente se puede describir el algoritmo de la siguiente manera. Dado un espacio de decisión n-dimensional cada partícula i del enjambre conoce su *posición actual* $X_i = [x_{i1}, x_{i2}, \dots, x_{in}]$, la *velocidad* $V_i = [v_{i1}, v_{i2}, \dots, v_{in}]$ con la cual ha llegado a dicha posición y su mejor posición $pBest_i = [p_{i1}, p_{i2}, \dots, p_{in}]$, denominada *mejor personal*. Además, todas las partículas conocen la mejor posición $gBest_i = [g_{i1}, g_{i2}, \dots, g_{in}]$ encontrada dentro del enjambre, llamada *mejor global*.

En cada iteración t del algoritmo, la componente $j = 1 \dots n$ de la velocidad correspondiente a la partícula i del enjambre se actualiza según:

$$v_{ij}^{t+1} = \omega v_{ij}^t + C_1 r_1 (pBest_{ij}^t - x_{ij}^t) + C_2 r_2 (gBest_{ij}^t - x_{ij}^t) \quad (4.2.1.1)$$

El componente *cognitivo* esta descrito por $C_1 r_1 (pBest_{ij}^t - x_{ij}^t)$ y representa la distancia entre la posición actual y la mejor conocida por esa partícula. El componente *social* esta modelado por $C_2 r_2 (gBest_{ij}^t - x_{ij}^t)$ y representa la distancia entre la posición actual y la mejor posición lograda hasta el momento.

C_1 y C_2 son conocidos como parámetros de aceleración, donde C_1 es el *parámetro cognitivo* que indica la influencia máxima de la mejor experiencia individual de la partícula en su nueva velocidad. C_2 es el *parámetro social* que indica la influencia máxima de la información social en la nueva velocidad de la partícula. r_1 y r_2 corresponden a valores aleatorios independientes en el intervalo $[0,1]$ y diferentes en cada iteración, mediante los cuales se determina la influencia real de las informaciones individual y social en la nueva velocidad para la partícula. ω , es el *factor de inercia* el cual regula el impacto de las velocidades en la nueva partícula. Mientras mayor sea el factor de inercia se facilita la búsqueda global (exploración), de lo contrario de facilita la búsqueda local (explotación). Este último parámetro no forma parte de la formulación inicial de PSO, al incluirlo se puede entender que se está realizando una variación del algoritmo original, la cual es denominada PSO *con factor de inercia*.

Por otra parte, la posición de la partícula i del enjambre se actualiza para cada componente $j = 1 \dots n$ según:

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (4.2.1.2)$$

En la Figura 5.1 se visualizan los componentes que influyen en el movimiento de una partícula.

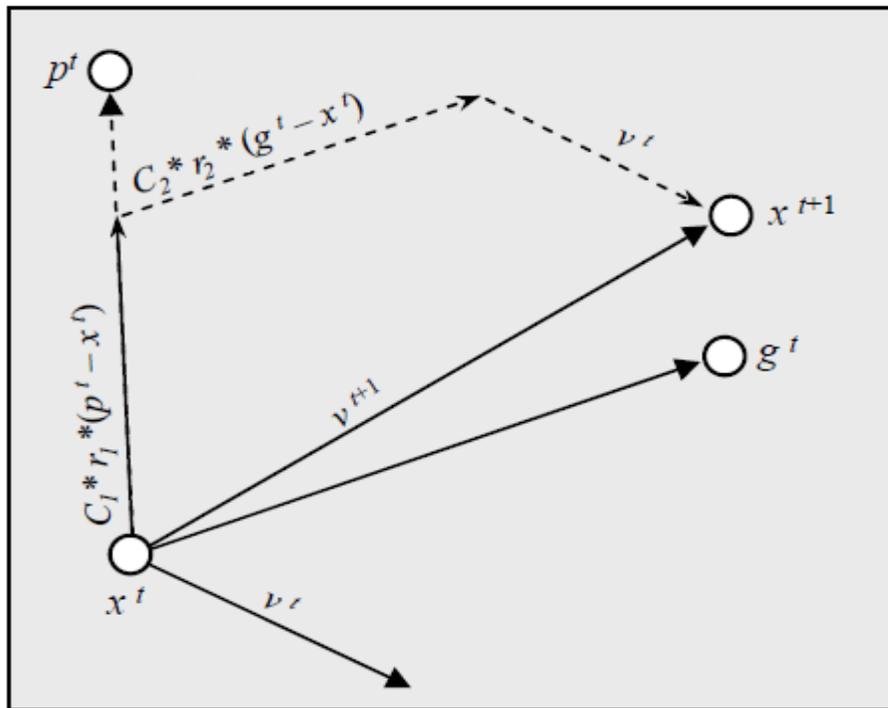


Figura 4.1 Actualización de una partícula.

Dado los valores que toman los parámetros se pueden identificar diferentes modelos del algoritmo:

- *Modelo Completo*: $C_1, C_2 > 0$. El componente cognitivo y el social participan en el movimiento de la partícula.
- *Modelo puramente Cognitivo*: $C_1 > 0$ y $C_2 = 0$. Solo el componente cognitivo interviene en el movimiento de la partícula.
- *Modelo puramente Social*: $C_1 = 0$ y $C_2 > 0$. Solo el componente social interviene en el movimiento de la partícula.

4.2.1 Algoritmo PSO

En la figura 4.2 se muestra el pseudocódigo del algoritmo PSO, en el cual se puede apreciar que por cada iteración se evalúan las posiciones de cada partícula en una función fitness, la cual tiene algún criterio de optimización, si el resultado de dicha evaluación es mejor que el resultado de la evaluación de su mejor posición histórica (*mejor personal o pBest*), la posición actual pasa a ocupar el puesto de mejor posición histórica. Luego se compara el fitness de la mejor posición histórica de cada partícula, con el fitness de la mejor posición histórica del enjambre (*mejor global o gBest*). Si el fitness de la *mejor personal* supera al fitness de la *mejor global*, se actualiza el valor de la *mejor global* tomando el valor de la *mejor personal*. Posteriormente, se actualiza la velocidad y posición de cada partícula y

se vuelve a iterar. Se debe establecer algún criterio de término para detener el algoritmo, como por ejemplo, un número predeterminado de iteraciones o hasta que se obtenga algún fitness en particular.

Algoritmo PSO

```

S ← InicializarCumulo()
while no se alcance la condición de parada do
  for i = 1 to size(S) do
    evaluar cada partícula  $x_i$  del cúmulo S
    if fitness( $x_i$ ) es mejor que fitness(pBesti) then
      pBesti ←  $x_i$ ; fitness(pBesti) ← fitness( $x_i$ )
    end if
    if fitness(pBesti) es mejor que fitness(gBest) then
      gBest ← pBesti; fitness(gBest) ← fitness(pBesti)
    end if
  end for
  for i = 1 to size(S) do
     $v_i$  ←  $\omega \cdot v_i + \varphi_1 \cdot rand_1 \cdot (pBest_i - x_i) + \varphi_2 \cdot rand_2 \cdot (gBest - x_i)$ 
     $x_i$  ←  $x_i + v_i$ 
  end for
end while
Salida: la mejor solución encontrada

```

Figura 4.2 Algoritmo PSO [25].

4.2.2 Parámetros

En el planeamiento original de PSO algunos parámetros permanecen implícitos, por ejemplo C_1 , C_2 y ω no están realmente enunciados, sino que tiene un valor implícito de 1. Modificando dichos parámetros se pueden realizar ajustes sutiles en el sistema que permitirán controlar comportamientos tan importantes como la convergencia y la explosión, esto último corresponde a un comportamiento oscilatorio y creciente de la posición de las partículas provocando la ineficiencia de PSO. Por ende, el ajuste de los parámetros del sistema permite optimizar el rendimiento del algoritmo.

Parámetros C_1 y C_2

Una buena elección de los parámetros C_1 y C_2 puede producir una rápida convergencia del algoritmo y evitar mínimos locales. Para el planteamiento original de PSO se propuso como valores por defecto $C_1 = C_2 = 2$. Los resultados de algunos experimentos indicaron que $C_1 = C_2 = 0,5$ puede arrojar mejores resultados. En la literatura se plantea que puede resultar positivo elegir un parámetro cognitivo mayor que el parámetro social pero con la restricción $C_1 + C_2 \leq 4$ [30].

Velocidad

En un momento dado las velocidades de las partículas pueden aumentar o disminuir acercándose a infinito, esto se debe al comportamiento estocástico que presentan las

velocidades dentro del sistema. Para evitar aquello se introduce un rango $[-V_{max}, V_{max}]$ que limita el crecimiento o el decrecimiento excesivo:

$$\begin{aligned} \text{if } v_{in} > V_{max} \quad \text{then } v_{in} &= V_{max} \\ \text{else if } v_{in} < -V_{max} \quad \text{then } v_{in} &= -V_{max} \end{aligned}$$

Factor de inercia

En [31] dan a conocer que Eberhart y Shi usualmente implementan el *factor de inercia*, ω , de manera decreciente con respecto al tiempo, generalmente disminuyendo desde 0,9 a 0,4 con $C_1 + C_2 = 4$. Esta disminución sostenida en el tiempo del factor de inercia induce un cambio desde un modo de búsqueda exploratoria a un modo de búsqueda de explotación. La disminución del factor de inercia puede realizarse mediante la siguiente ecuación:

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{iter_{max}} \cdot iter,$$

donde ω_{max} sería 0,9, ω_{min} sería 0,4, $iter$, la iteración actual y $iter_{max}$ la máxima iteración previamente definida.

Exploración y Explotación

En PSO es importante mantener un equilibrio entre diversificación e intensificación. El término *exploración o diversificación* se refiere a la evaluación de soluciones en regiones distantes del espacio de búsqueda. El termino *explotación o intensificación* se refiere a la evaluación de soluciones en regiones acotadas y pequeñas con respecto al espacio de búsqueda. En [31] se dan a conocer algunos estudios en donde altos valores para el *factor de inercia*, el parámetro de cognición y V_{max} favorecen el modo de exploración del algoritmo PSO, mientras que valores bajos favorecen el modo de explotación.

4.3 Variantes de PSO

Desde la creación de PSO en 1995 han aparecido una gran cantidad de modificaciones las cuales en general buscan obtener mejores resultados a una mayor velocidad de convergencia. A continuación se explican cinco de estas modificaciones.

4.3.1 Vecindad Local

El modelo original de PSO responde a una estructura en la cual todos los individuos de un enjambre son influenciados de igual manera por un individuo líder que es quien ha encontrado una mejor ubicación respecto a un objetivo en un momento determinado. Este comportamiento social se puede asociar a una *topología de vecindad global* [34] [25], en la cual todos los individuos están interrelacionados. En PSO la información global es compartida por medio de la variable denominada *gbest*. Una alternativa a PSO (*modelo óptimo global*) es aquella que se puede representar por medio de una *topología de vecindad local* en donde ya no existe una influencia global ni interrelaciones entre todos los individuos (ver Figura 5.3). Las relaciones están acotadas a una vecindad más pequeña en donde cada individuo solo comparte con una parte del enjambre, basado en dicho planteamiento se constituye una variante de PSO,

a la cual se le puede denominar *modelo de óptimo local* que considera como variable de comunicación *lbest* (*local best*). La ecuación (4.2.1.1) es actualizada reemplazando *gbest* por *lbest* :

$$v_{ij}^{t+1} = \omega v_{ij}^t + C_1 r_1 (pbest_{ij}^t - x_{ij}^t) + C_2 r_2 (lbest_{ij}^t - x_{ij}^t),$$

Las partículas en el *modelo de óptimo local* que se encuentren en un subconjunto del enjambre y no tienen relación directa con partículas de otros vecindarios.

En [34] se expone que el *modelo óptimo global* converge más rápido pero tiende a quedarse estancado en óptimos locales, mientras que el *modelo óptimo local* converge más lentamente pero puede superar el problema de los óptimos locales, pues los individuos exploran distintas regiones.

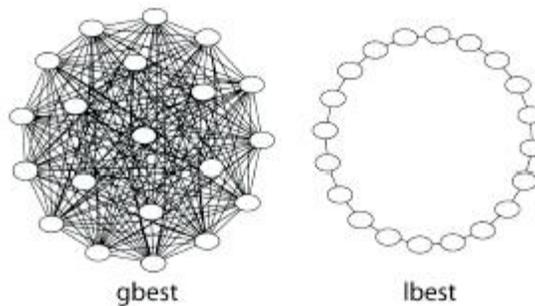


Figura 4.3 Topologías, asociadas a *gbest* y *lbest* [25].

4.3.2 Adaptive PSO (APSO)

En el proceso de evolución, el enjambre podría someterse a un proceso no deseado de pérdida de la diversidad. Esto quiere decir que algunas partículas podrían comenzar a perder su capacidad de búsqueda global y local en las sucesivas generaciones. La pérdida de la capacidad de búsqueda global para una partícula significa que su “vuelo” estará reducido a un espacio muy pequeño, lo que ocurre cuando la ubicación del *pbest* está muy cerca del *gbest* y su velocidad es cercana a 0 (para todas las dimensiones), de acuerdo a las ecuaciones 4.2.1.1. y 4.2.1.2; la pérdida de capacidad de búsqueda local significa que el “vuelo” de las partículas no genera efectos perceptibles dada su evaluación en la función fitness. Si se cumplen dichas condiciones significa que el proceso de evolución se estanca. Si *gbest* está localizado en un óptimo local, entonces el enjambre converge prematuramente, puesto que todas las partículas llegan a estar inactivas [32].

Para estimular al enjambre y continuar con su evolución sostenida, las partículas inactivas deben ser sustituidas por una nueva con el fin de mantener las relaciones no-lineales de retroalimentación mediante el mantenimiento de la diversidad social del enjambre. APSO realiza dicha tarea sustituyendo, del algoritmo PSO, el paso en el cual se actualiza la velocidad y posición de partícula de acuerdo a las ecuaciones 4.2.1.1 y 4.2.1.2, por una seguidilla de pasos que permiten sustituir una partícula que se encuentra inactiva. En la figura 4.4 se muestra el pseudocódigo APSO para la sustitución o reinicio de la partícula.

```

int[] similarCount = new int[m]; // Un estado inicial
FOR (i=0; i<m; i++) {           // por cada partícula
    IF (i!=g && |ΔFi| < ε )
        THEN similarCount[i] ++; // incremento
    ELSE similarCount[i] = 0;    // reinicia
    IF (similarCount[i] > Tc) // contador
        THEN replace (i-ésima partícula);
    ELSE execute ( ejecuta paso * en PSO );
}

```

Figura 4.4 Pseudocódigo APSO [32].

A continuación se describen las funciones y parámetros del Pseudocódigo APSO:

F_i	: Fitness de la i -ésima partícula
F_{gbest}	: Fitness de $gbest$
$\Delta F_i = f(F_i, F_{gbest})$: Función de error.
ε	: Constante predefinida acorde a la precisión requerida.
T_c	: Contador
$replace()$: Función empleada para reemplazar la i -ésima partícula donde x_i y v_i son reinicializadas o cambiadas y su $pbest$ se iguala a x_i .
$similarCount[i]$: Es un arreglo para almacenar la cuenta que satisface la condición $\Delta F_i < \varepsilon$.

4.3.3 Fitness Distance Ratio Based Particle Swarm Optimization (FDRPSO)

FDRPSO [33] propone que las partículas también puedan ser capaces de aprender de la experiencia de una partícula vecina, sin importar que esta no corresponda a la *mejor global*. Cada partícula incluirá en la actualización de su velocidad a su *mejor vecino (nbest)*, la cual será elegida basándose en dos criterios:

- Debe estar cerca de la partícula que se está actualizando.
- Debe haber visitado una posición con mejor fitness.

Una de los modos más simples de satisfacer estos criterios es maximizando el cociente entre la diferencia de fitness de ambas partículas y la distancia que las separa. En otras palabras, la dimensión d -ésima de la i -ésima partícula es actualizada usando la partícula $nbest$.

$$\frac{Fitness(P_j) - Fitness(X_i)}{|P_{jd} - X_{id}|}$$

El algoritmo FDRPSO modifica la ecuación original de velocidad 4.2.1.1:

$$v_{ij}^{t+1} = \omega v_{ij}^t + C_1 r_1 (pbest_{ij}^t - x_{ij}^t) + C_2 r_2 (gbest_{ij}^t - x_{ij}^t) + C_3 r_3 (nbest_{ij}^t - x_{ij}^t),$$

4.3.4 Particle Swarm With Exponential Distribution (PSO-E)

PSO-E [46], plantea el uso de la distribución de probabilidad exponencial con función de densidad:

$$f(x) = \frac{1}{2b} e^{\left(-\frac{|x-a|}{b}\right)}, -\infty \leq x < \infty, \text{ con } a, b > 0$$

Se hace evidente que se puede controlar la varianza cambiando los parámetros a y b . En la figura 4.5 se muestra la distribución de la función de probabilidad exponencial con el parámetro de locación $a = 0$ y el parámetro de escala $b = 0.5$ y $b = 1.0$. El objetivo propuesto por [46] es investigar como dicha distribución influye en el rendimiento de PSO. La generación de números aleatorios usando la distribución exponencial se describe de la siguiente manera:

```
% Generación de números aleatorios usando
distribución exponencial

u1 = rand; número aleatorio uniforme [0;1]
u2 = rand; número aleatorio uniforme [0;1]

if u1 > 0.5
    x = a + b * log(u2);
else
    x = a + b * log(u2);
end
y = abs(x);
```

Los números aleatorios generados a partir de $abs(E(a, b))$, utilizados para los coeficientes estocásticos de PSO, pueden proporcionar un buen compromiso entre la probabilidad de tener un gran número de pequeñas amplitudes en torno a los puntos actuales (ajuste fino) y una pequeña probabilidad de tener grandes amplitudes, lo que puede permitir que las partículas se alejen del punto actual y escapar de los mínimos locales.

La velocidad es actualizada de acuerdo a:

$$= abs(E(a, b))(p_i - x_i) + abs(E(a, b))(p_g - x_i)$$

La distribución de probabilidad exponencial se utiliza para la generación de los coeficientes de ponderación en el algoritmo estándar de PSO.

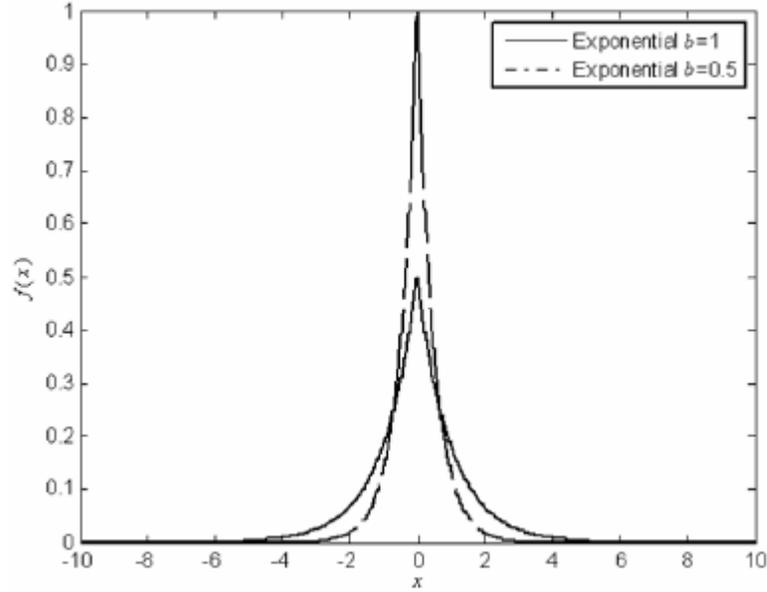


Figura 4.5 Distribución exponencial [46].

4.3.5 Adaptive Inertia Weight PSO (AIWPSO)

Los métodos tradicionales utilizados para actualizar los factores de inercia o pesos de inercia se basan en el fitness o en el número de iteraciones. En [47] se propone un método en donde se utiliza la tasa de éxito del enjambre para determinar el peso de inercia.

En primer lugar, es necesario determinar la situación del enjambre en cada iteración, para aquello se utiliza la tasa de éxito del enjambre. Un alto porcentaje de éxito indica que las partículas han convergido a un punto que dista mucho de ser el punto óptimo y el enjambre entero se está moviendo lentamente hacia el óptimo. Del mismo modo, un bajo porcentaje de éxito demuestra que las partículas están oscilando en torno al óptimo, sin mucha mejoría.

El éxito de las partículas i en la iteración t en un problema de minimización se define como:

$$S(i, t) = \begin{cases} 1, & \text{fit}(pbest_i^t) < \text{fit}(pbest_i^{t-1}) \\ 0, & \text{fit}(pbest_i^t) = \text{fit}(pbest_i^{t-1}) \end{cases}$$

Donde $pbest_i^t$ es la mejor posición encontrada por la partícula i hasta la iteración t y $fit()$ es la función a ser optimizada. Usando los valores de éxito de las partículas, el porcentaje de éxito del enjambre es calculado de la siguiente forma:

$$P_s(i, t) = \frac{\sum_{i=1}^n S(i, t)}{n},$$

Donde n es el número de partículas y $P_s \in [0,1]$ es el porcentaje de las partículas que han tenido una mejoría en su fitness en la última iteración. Mientras más alto sea el valor de P_s acelerará el movimiento de las partículas hacia el punto óptimo, mientras que valores más pequeños de P_s reduce la velocidad de las partículas para converger en óptimo que hayan

localizado. Por lo tanto, es razonable escoger un valor de w proporcional a P_s . De esta manera, el factor de inercia puede ser cualquier función creciente de P_s definida como:

$$w(t) = f(P_s(t))$$

En [47], se utiliza una función lineal para mapear los valores de P_s a los factores de inercia, como se ve a continuación:

$$w(t) = (w_{max} - w_{min})P_s(t) + w_{min}$$

El rango de factores de inercia ($[w_{min}, w_{max}]$) es seleccionado para estar entre 0 y 1. Aunque P_s está en el rango $[0,1]$, w pueden estar en cualquier rango aceptable.

5 Elementos del Estudio

En este capítulo se explica la estructura de los datos; el contexto en el que son obtenidos; se describen las variables que participan en la construcción y las medidas de evaluación del modelo.

5.1 Estructura y Contexto de los Datos

Los datos utilizados para construir y evaluar el modelo se han obtenidos desde el estudio: “*Manejo y Monitoreo de mareas rojas en las regiones de Los Lagos, Aysén y Magallanes*”, el cual ha sido diseñado para monitorear las variaciones espacio temporales del Veneno Paralizante de los Mariscos (VPM), el Veneno Diarreico de los Mariscos (VDM) y el Veneno Amnésico de los Mariscos (VAM) en especies transvectoras centinelas; también monitorea especies de microalgas nocivas o potencialmente nocivas; variables ambientales de tipo hidrográficas y meteorológicas; fitoplancton cualitativo y cuantitativo. El objetivo general del estudio es disponer de un sistema de muestreo, detección y cuantificación periódico del VPM, VDM, VAM y de las especies nocivas fuentes primarias de las toxinas (*Alexandrium catenella*, *Dinophysis acuta* y *Pseudo-nitzschia cf. australis*).

El área de análisis para el diseño e implementación del modelo se restringe a la zona denominada “Última Esperanza” ubicada en la región de Magallanes. Dicha región cuenta con 48 estaciones de monitoreo de las cuales 9 se ubican en Última Esperanza. La base de datos utilizada en la elaboración del modelo contiene registros desde mayo de 2006 hasta diciembre de 2010, los cuales son obtenidos con periodicidad mensual. Existen lagunas de tiempo en las que no hay registros, debido a que no se realizaron muestreos por razones administrativas.

Luego de realizar un proceso de limpieza en la base datos, eliminando registros incompletos, se obtuvo un total de 314 registros útiles. Los registros se identifican por un código de estación y la fecha del muestreo (ver anexo 1).

5.2 Variables

Las variables contenidas en los registros de la base de datos se dividieron en tres grupos: variables meteorológicas, oceanográficas y otras. En total son 20 variables de las cuales 19 corresponden a las variables independientes del modelo y una a la variable dependiente (Densidad de *Alexandrium Catenella*) o variable a partir de la cual se definen las clases del clasificador. Las 20 variables que conforman los registros son las siguientes:

- Clorofila a entre 0 y 10 metros de profundidad ($mg\ m^{-3}$).
- Clorofila a entre 10 y 20 metros de profundidad ($mg\ m^{-3}$).
- Temperatura Ambiente ($^{\circ}C$).
- Nubosidad (*octavos de cielo*).
- Presión barométrica (*mba*).
- Velocidad promedio del viento ($m\ s^{-1}$)
- Temperatura oceanográfica ($^{\circ}C$) a 0 metros.
- Temperatura oceanográfica ($^{\circ}C$) a 5 metros.

- Temperatura oceanográfica ($^{\circ}C$) a 10 metros.
- Temperatura oceanográfica ($^{\circ}C$) a 20 metros.
- Salinidad (*psu*) a 0 metros.
- Salinidad (*psu*) a 5 metros.
- Salinidad (*psu*) a 10 metros.
- Salinidad (*psu*) a 20 metros.
- Sigma-t (kg/m^3) a 0 metros.
- Sigma-t (kg/m^3) a 5 metros.
- Sigma-t (kg/m^3) a 10 metros.
- Sigma-t (kg/m^3) a 20 metros.
- Profundidad de disco Secchi (*m*).
- Densidad de Alexandrium Catenella (*cel/L*).

En la tabla 5.1 se agrupan y se describen los distintos tipos de variables.

Tabla 5.1 Descripción de variables.

VARIABLES OCEANOGRÁFICAS	DESCRIPCIÓN
Profundidad de disco Secchi (<i>m</i>):	Círculo de 20 cm de diámetro, dividido en cuadrantes pintados alternadamente de negro y blanco, atado a una cuerda graduada. El disco se sumerge del lado sombreado de la embarcación hasta que deja de verse, se registra la profundidad y vuelve a subirse hasta que nuevamente se haga visible. Luego se promedian estas dos medidas (desaparición con disco descendiendo y aparición en ascenso) para obtener una medida dependiente de la transparencia del agua. Se clasifica como sistemas de aguas claras a los que presentan lecturas de Secchi mayores o iguales a un metro [40].
Salinidad (<i>psu</i>):	La definición más sencilla corresponde a la cantidad total de materia sólida (en gramos) disuelta en un kilogramo de agua de mar [45].
Sigma-t (kg/m^3):	Mide la densidad del agua de mar. La densidad del agua de mar, a presión atmosférica, por lo general varía entre 1025 y 1028 kilogramos por metro cúbico. Dado el pequeño margen de variación de la densidad en el océano, los oceanógrafos restan en 1000 el valor original ($Sigma - t = densidad - 1000 kg/m^3$) [41].
Temperatura oceanográfica ($^{\circ}C$):	Medida de la cantidad de calor en el agua [41].
VARIABLES METEOROLÓGICAS	DESCRIPCIÓN
Nubosidad (<i>octavos de cielo</i>):	Parte de cielo cubierta por nubes. Se mide en octas o décimas de superficie de la esfera celeste cubierta por nubosidad. 1 octa corresponde a la octava parte del cielo

	cubierto por nubes [42].
Presión (<i>mba</i>):	Es la fuerza por unidad de superficie ejercida por el peso de la atmósfera [43].
Temperatura Ambiente ($^{\circ}C$):	Magnitud proporcional a la energía cinética media de las moléculas de aire [42].
Velocidad promedio del viento ($m s^{-1}$):	Distancia recorrida por el viento en la unidad de tiempo [52].
Otras variables	Descripción
Clorofila a ($mg m^{-3}$):	La clorofila es la sustancia que permite a las plantas absorber luz solar y convertirla en energía utilizable [44]. La concentración de clorofila- <i>a</i> se utiliza para estimar en forma indirecta la biomasa de las comunidades fitoplanctónicas, debido a que es el principal pigmento fotosintético presente en las algas [45].
Densidad de Alexandrium Catenella (<i>cel/L</i>):	Corresponde al número de células de Alexandrium Catenella en un litro de agua de mar.

5.3 Estaciones

Como ya se explicó, la investigación esta acotada a la región de Magallanes en la cual se mantienen estaciones de monitoreo de marea roja. En la figura 5.1 se puede observar la ubicación espacial de las estaciones y se destaca con un círculo la ubicación del núcleo de toxicidad “Última Esperanza”. Los núcleos de toxicidad corresponden a sectores que habitualmente registran niveles importantes de toxinas. La lista de estaciones se puede ver en el anexo 1.

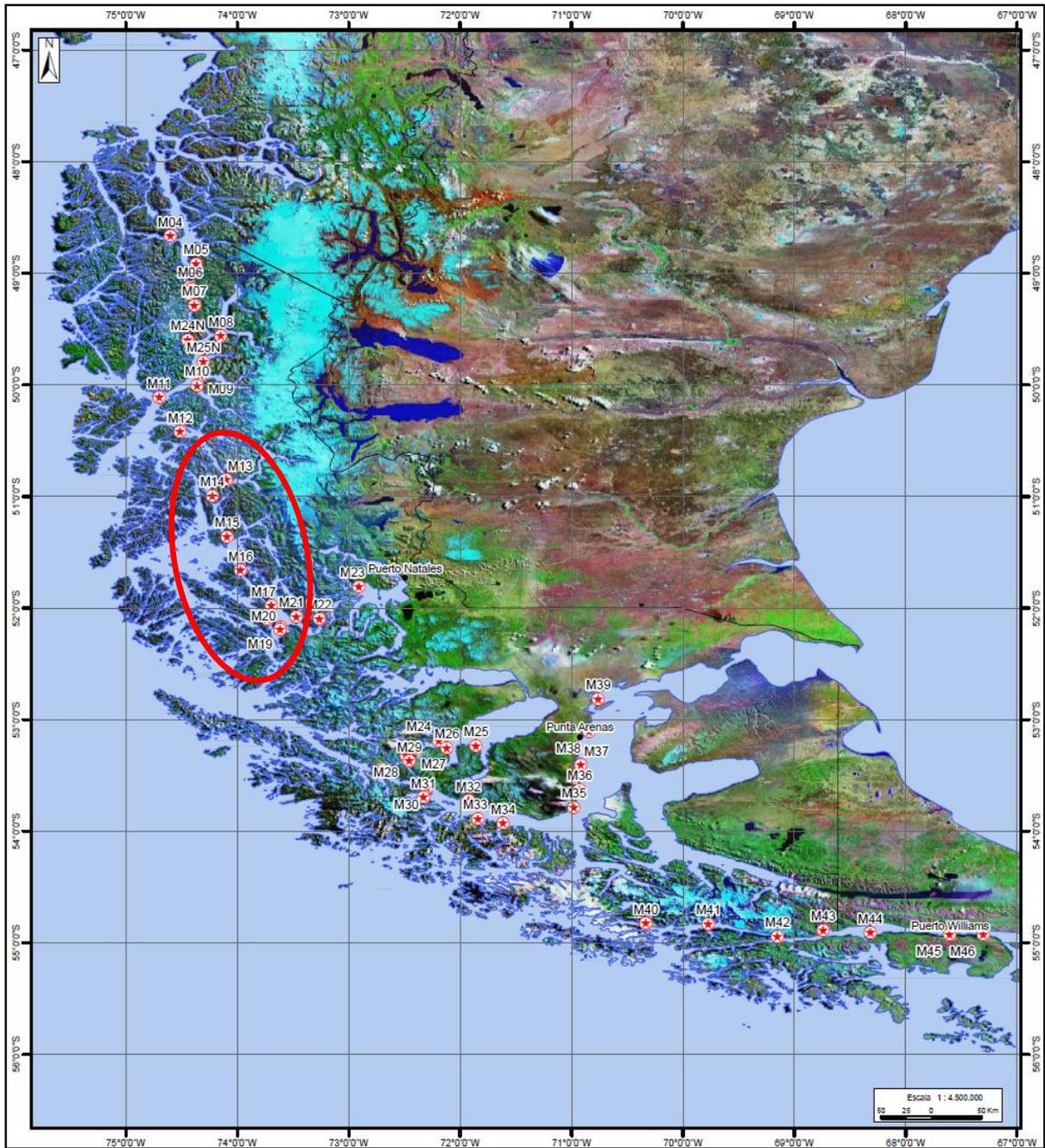


Figura 5.1 Distribución geográfica de las estaciones de muestreo en la región de Magallanes. En rojo de destaca el núcleo de toxicidad “Última Esperanza” [1].

6 Medidas de Evaluación

El modelo pretende determinar la existencia o inexistencia de marea roja en un lugar y periodo determinado. Los resultados se pueden clasificar en cuatro grupos según una tabla 2x2 denominada matriz de confusión (ver tabla 6.1). Esta matriz cruza los resultados de las pruebas (filas) con el estado real de los eventos (columnas). El resultado de las pruebas puede ser correcto (verdaderos positivos y verdaderos negativos) o incorrectos (falso positivos y falsos negativos).

Tabla 6.1 Matriz de confusión

Resultados obtenidos	Verdadero Diagnóstico	
	Con Marea Roja	Sin Marea Roja
Positivo	Verdaderos Positivos (VP)	Falsos Positivos (FP)
Negativo	Falsos Negativos (FN)	Verdaderos Negativos (VN)

A continuación se interpretan los cuatros grupos que conforman la matriz de confusión, en el contexto del problema:

Tabla 6.2 Grupos, matriz de confusión.

<i>Verdaderos Positivos (VP).</i>	Número de eventos detectados correctamente “con marea roja”.
<i>Verdaderos Negativos (VN):</i>	Número de eventos detectados correctamente “sin marea roja”.
<i>Falsos Positivos (FP):</i>	Número de eventos detectados “con marea roja” de forma errónea. Estos eventos estaban “sin marea roja”.
<i>Falsos Negativos (FN):</i>	Número de eventos detectados “sin marea roja” de forma errónea. Estos eventos estaban “con marea roja”.

A partir de la matriz de confusión se pueden obtener algunas medidas que permiten evaluar la **validez** y la **seguridad** del modelo. Se define *validez* como el grado con el que un test mide realmente lo que se supone debe medir, son condiciones de validez la *exactitud*, *sensibilidad* y *especificidad*. La *seguridad* se refiere a la capacidad predictiva del clasificador para obtener un valor positivo o negativo (con marea roja o sin marea roja), son medidas de seguridad el *valor predictivo positivo* y el *valor predictivo negativo*.

En caso de ser posible, según la cantidad de registros en la base de datos, se debe medir el grado de **reproductividad** del modelo, el cual corresponde a la capacidad del clasificador para ofrecer los mismos resultados cuando se repite su aplicación en circunstancias similares.

Las condiciones de validez proporcionan información acerca de la probabilidad de obtener un resultado concreto (positivo o negativo) en función de la verdadera condición del evento.

Exactitud

Corresponde a la proporción de eventos bien clasificados, ya sea con o sin marea roja, dentro del total de eventos clasificados. Esta medida se describe de la siguiente forma:

$$Exactitud = \frac{VP + VN}{VP + VN + FN + FP}$$

Sensibilidad

Corresponde a la proporción de eventos clasificados con marea roja, que fueron detectados como tal, de entre todos los eventos que realmente poseen marea roja. Esta medida se describe de la siguiente forma:

$$Sensibilidad = \frac{VP}{VP + FN}$$

Especificidad

Corresponde a la proporción de eventos sin marea roja, que fueron detectados como tal, por la prueba. Esta medida se describe de la siguiente forma:

$$Especificidad = \frac{VN}{VN + FP}$$

Las condiciones de seguridad se basa en la inexistencia de información a priori sobre el estado de los eventos y plantea: ¿cuál es la probabilidad de que el evento tenga o no marea roja?.

Precisión o Valor Predictivo Positivo

Corresponde a la proporción de eventos detectados correctamente con marea roja por la prueba, de entre todos los eventos clasificados (correcta o incorrectamente) como “con marea roja”. Esta medida se describe de la siguiente forma:

$$VPP = \frac{VP}{VP + FP}$$

Valor Predictivo Negativo

Corresponde a la proporción de eventos detectados correctamente sin marea roja por la prueba, de entre todos los eventos clasificados (correcta o incorrectamente) como “sin marea roja”. Esta medida se describe de la siguiente forma:

$$VPN = \frac{VN}{VN + FN}$$

En clasificación binaria también se utiliza como medida de evaluación la curva ROC (*Receiver Operating Characteristic*). ROC es un gráfico de dos dimensiones, donde su eje y representa la proporción de verdaderos positivos (VP), mientras que su eje x representa la proporción de falsos positivos (FP). La proporción de VP es equivalente a sensibilidad y la proporción de FP es equivalente a 1-especificidad.

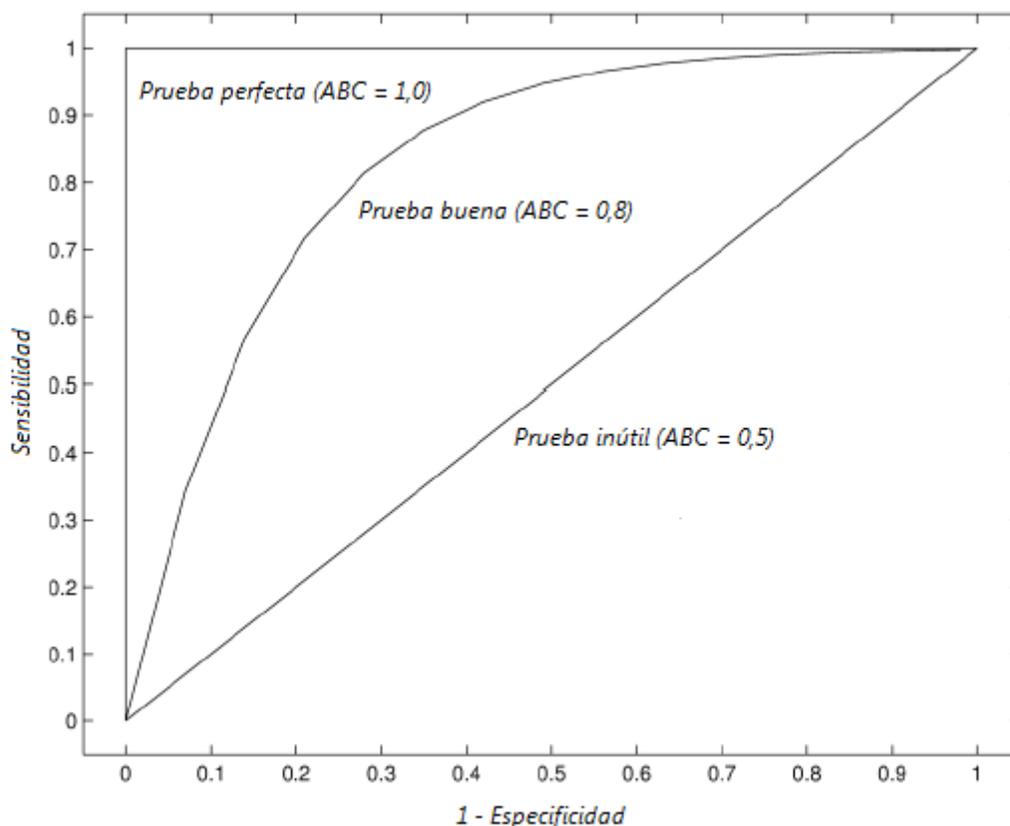


Figura 6.1 Curva ROC.

Una prueba perfecta, sin solapamientos de resultados en las poblaciones, significa que hay una región en la que cualquier punto de corte tienen sensibilidad y especificidad iguales a 1, lo que se interpreta como que todos los eventos fueron clasificados correctamente, esto se representa en el punto (0,1) de la curva ROC. Si la prueba es inútil, la proporción de sensibilidad coincide con la proporción de falsos positivos. Finalmente, la curva consistiría en una diagonal de (0,0) a (1,1). Por lo general, las pruebas generan curvas intermedias.

Se puede observar (ver figura 6.1) que la mayor exactitud en la clasificación de una prueba se traduce en un desplazamiento "hacia arriba y a la izquierda" de la curva ROC. Esto sugiere que el área bajo la curva ROC se puede emplear como un índice de la exactitud global de la prueba. En donde la exactitud máxima correspondería a un valor del área bajo la curva de 1 y la mínima de 0.5 (si fuera menor de 0.5 debería invertirse el criterio de positividad de la prueba).

6.1.1 Ejemplo Contextual

A continuación se describe un ejemplo contextual para clarificar el uso de la tabla de confusión y de las medidas de evaluación que se desprenden de esta.

Supóngase una base de datos con 300 registros, los cuales son clasificados según la existencia o inexistencia de marea roja. A partir de esta clasificación se desprende la siguiente matriz de confusión:

Tabla 6.3 Matriz de confusión. Ejemplo.

Resultados obtenidos	Diagnóstico	
	Con Marea Roja	Sin Marea Roja
Positivo	90	48
Negativo	22	140

Los resultados de la matriz de confusión se pueden interpretar de la siguiente manera:

Tabla 6.4 Resultados matriz de confusión.

<i>Verdaderos Positivos (VP):</i>	Se clasificaron 90 eventos correctamente “con marea roja”.
<i>Verdaderos Negativos (VN):</i>	Se clasificaron 140 eventos correctamente “sin marea roja”.
<i>Falsos Positivos (FP):</i>	Hay 48 eventos clasificados “con marea roja” equivocadamente, ya que debieron ser clasificados “sin marea roja”.
<i>Falsos Negativos (FN):</i>	Hay 22 eventos clasificados “sin marea roja” equivocadamente, ya que debieron ser clasificados como “con marea roja”.

A partir de la matriz de confusión se pueden obtener los siguientes resultados de las medidas de evaluación:

Tabla 6.5 Resultados medidas de evaluación.

Medida	Resultado	Descripción
<i>Exactitud</i>	0,76	Un 76% de los eventos fueron clasificados correctamente.
<i>Sensibilidad</i>	0,80	Entre todos los eventos que realmente tenían marea roja, un 80% fueron clasificados como tal.
<i>Especificidad</i>	0,74	Entre todos los eventos que realmente no tenían marea roja, un 74% fueron clasificados como tal.
<i>Valor Predictivo Positivo</i>	0,65	Del total de eventos clasificados con marea roja, un 65% está correctamente clasificado.
<i>Valor Predictivo Negativo</i>	0,86	Del total de eventos clasificados sin marea roja, un 86% está correctamente clasificado.

7 Modelo LS-SVM PSO

En este capítulo se explica el modelo propuesto para pronosticar la existencia o inexistencia de marea roja en la región de Magallanes. Este modelo considera, en primer lugar, una etapa de entrenamiento para dotar de memoria a la LS-SVM y posteriormente una etapa de validación.

7.1 Descripción del Modelo

Para obtener un buen pronosticador se debe conseguir un aprendizaje adecuado sobre la conducta espacio temporal del fenómeno de floración de *Alexandrium Catenella* (marea roja), en la región de Magallanes. Para lograr dicho cometido se entrena la LS-SVM con los datos obtenidos desde las estaciones de monitoreo, e iterativamente se optimizan ciertos parámetros involucrados en este proceso, utilizando la metaheurística PSO. De esta manera, se está construyendo un clasificador binario con el cual se podrá pronosticar la presencia o ausencia de marea roja.

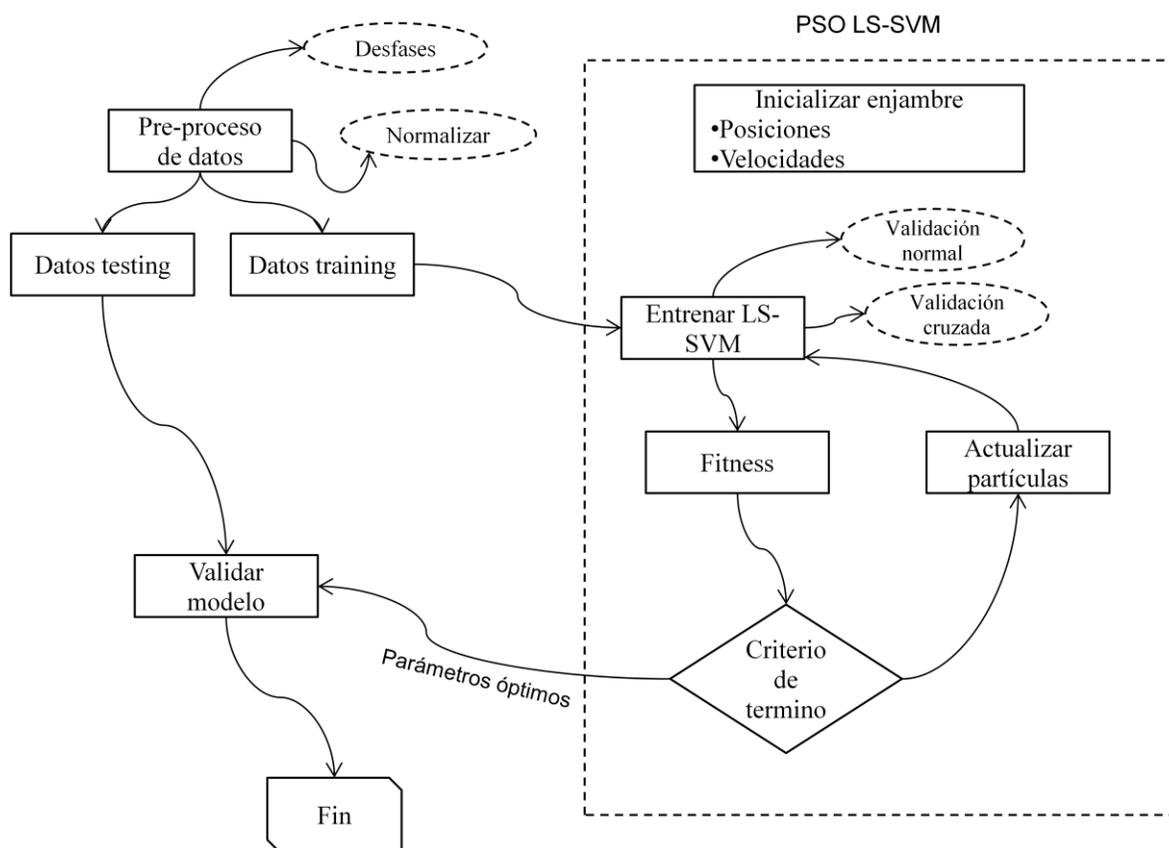


Figura 7.1 Modelo LS-SVM PSO.

7.1.1 Pre-proceso de Datos

Previo al pre-proceso propio del modelo, existen ciertas tareas de limpieza y cohesión de los registros de la base de datos, que facilitan su manipulación. Ejemplos de estas tareas son:

la eliminación de registros incompletos, el formateo de los tipos de datos para uniformar todos los registros y la discriminación de variables cualitativas debido a la imposibilidad de procesarlas por el modelo.

Una vez realizado el pre-proceso externo al modelo, los datos son normalizados y desfasados. Normalizar ayuda a asegurar que la influencia o importancia de cada atributo sea la misma.

Desfasar los registros permite que la LS-SVM obtenga la habilidad de pronóstico, ya que de dicha manera se asocia una variable dependiente obtenida en un tiempo t , con variables independientes obtenidas en periodos anteriores.

Ejemplo:

Se realizan dos desfases (suponiendo periodos mensuales) para un conjunto de registros con una variable dependiente Y y cuatro variables independientes $X1, X2, X3$ y $X4$:

Tabla 7.1 Ejemplo desfases

Periodo	Sin Desfases					Desfase 1				Desfase 2				Y	
	Y	X1	X2	X3	X4	X1	X2	X3	X4	X1	X2	X3	X4		
1	3	4	34	8	9										3
2	5	12	23	10	14	4	34	8	9						5
3	4	14	12	11	3	12	23	10	14	4	34	8	9	4	
4	5	6	6	3	6	14	12	11	3	12	23	10	14	5	
5	4	3	5	6	23	6	6	3	6	14	12	11	3	7	
6	12	9	7	5	3	3	5	6	23	6	6	3	6	12	

Los registros encasillados son los que se utilizan para entrenar o testear la LS-SVM. El registro |12 |23 |10 |14 |4 |34 |8 |9|, formado por las datos de los periodos 1 y 2, está asociado con la variable dependiente Y del periodo 3, o sea con el valor 4. Luego, el registro |14 |12 |11 |3 |12 |23 |10 |14| está formado por los datos de los periodos 2 y 3, y asociado con la variable dependiente del periodo 4, o sea con el valor 5. Esta secuencia se repite hasta que no se pueda seguir desfasando. La cantidad de desfases adecuada se determina de manera empírica.

Con los datos normalizados y los registros desfasados se divide la base de datos en dos grupos; datos para entrenamiento (training) y datos para test (testing).

7.1.2 Inicialización del Enjambre PSO

La posición y la velocidad de las partículas del enjambre PSO se inicializan con valores aleatorios. Cada dimensión de la partícula contiene los valores que se pretenden optimizar, para el modelo LS-SVM PSO dichos valores corresponden al parámetro C que controla el trade-off de la LS-SVM y a los parámetros del Kernel, estos últimos varían según la función

Kernel utilizada. PSO retorna la partícula con la combinación de parámetros que mejores resultados le otorgan al modelo. Además, PSO y sus variantes necesitan inicializar otros parámetros constantes asociados a la fórmula de velocidad de las partículas. El número óptimo de partículas se determina de manera empírica.

7.1.3 Entrenamiento

En la etapa de entrenamiento, se realiza un número determinado de iteraciones en donde se prueba la LS-SVM y se calcula el Fitness de cada partícula. El Fitness es una medida que permite discriminar la calidad de las partículas. Por lo tanto, se debe escoger aquella partícula con el mejor Fitness, o sea aquella partícula que minimice el error de pronóstico.

Dos técnicas por medio de las cuales se puede calcular el Fitness son:

7.1.3.1 Validación Típica

Se evalúa cada partícula en la LS-SVM utilizando los datos de training y por cada partícula se obtiene un Fitness. El Fitness se calcula a partir de una de las medidas de evaluación presentadas en el capítulo 6. Ejemplo de estas son: 1-Exactitud, que se interpreta como el ratio de registros mal clasificados; 1-Precisión, ratio de eventos clasificados erróneamente con marea roja.

7.1.3.2 Validación Cruzada

Se dividen los datos de training en L grupos, y se utilizan $L-1$ grupos para entrenar la LS-SVM (con la i -ésima partícula) y 1 para validar los resultados del entrenamiento. Se realizan L iteraciones y en cada iteración se utiliza un grupo distinto para evaluar el modelo. Finalmente, se calcula la media de los Fitness obtenidos a partir de las L pruebas y se asocia como el Fitness de la i -ésima partícula. Luego, se evalúan de igual forma todas las partículas.

7.1.4 Actualización de Partículas

Por cada iteración se actualizan las posiciones y velocidades de las partículas. Las partículas se actualizan en búsqueda de una mejor posición dentro del espacio de búsqueda. Dependiendo del tipo de PSO, se introducen algunas modificaciones en la forma y en los criterios utilizados para actualizar las partículas, por ejemplo APSO reinicia con valores aleatorios la posición y la velocidad de una partícula si ve que se está estancando en un óptimo local.

7.1.5 Validación del Modelo

Finalmente, se valida el modelo utilizando el conjunto de datos de testing y los parámetros óptimos obtenidos desde el entrenamiento. Las medidas de evaluación utilizadas son Exactitud, Sensibilidad, Especificidad, Precisión, Valor Predictivo Negativo y la Curva ROC.

8 Implementación y Resultados del Modelo

En este capítulo se detalla la implementación del modelo propuesto y se presentan los resultados obtenidos a partir de las diferentes configuraciones.

8.1 Datos Utilizados

Para entrenar y evaluar el modelo, se utilizaron los registros obtenidos desde las 9 estaciones que conforman el núcleo de toxicidad “Última Esperanza”. Esto se debe a que en dicha zona se ha detectado una homogeneidad temporal y de magnitud de densidad de *Alexandrium Catenella*. Lo que hace suponer mejores características para el aprendizaje de la LS-SVM.

Para el periodo 2006-2010 existen en la base de datos un total de 314 registros obtenidos desde las estaciones del núcleo, lo que equivale a aproximadamente 35 meses de muestreos.

Las variables independientes que conforman los registros son:

- Clorofila a entre 0 y 10 metros de profundidad ($mg\ m^{-3}$).
- Clorofila a entre 10 y 20 metros de profundidad ($mg\ m^{-3}$).
- Temperatura Ambiente ($^{\circ}C$).
- Nubosidad (*octavos de cielo*).
- Presión barométrica (*mba*).
- Velocidad promedio del viento ($m\ s^{-1}$).
- Temperatura oceanográfica ($^{\circ}C$) a 0 metros.
- Temperatura oceanográfica ($^{\circ}C$) a 5 metros.
- Temperatura oceanográfica ($^{\circ}C$) a 10 metros.
- Temperatura oceanográfica ($^{\circ}C$) a 20 metros.
- Salinidad (*psu*) a 0 metros.
- Salinidad (*psu*) a 5 metros.
- Salinidad (*psu*) a 10 metros.
- Salinidad (*psu*) a 20 metros.
- Sigma-t (kg/m^3) a 0 metros.
- Sigma-t (kg/m^3) a 5 metros.
- Sigma-t (kg/m^3) a 10 metros.
- Sigma-t (kg/m^3) a 20 metros.
- Profundidad de disco Secchi (*m*).

La variable independiente a partir de la cual se generan las clases del modelo para clasificar y pronosticar es:

- Densidad de *Alexandrium Catenella* (*cel/L*).

8.2 Codificación

Cuando la densidad de *Alexandrium Catenella* (DAC) es superior a 0, significa que se ha detectado una floración de la microalga en una estación determinada. Por lo tanto, los registros serán clasificados con 1 si la DAC es superior a 0 y con -1 si la densidad de *Alexandrium Catenella* es igual a 0.

$$1, \forall DAC > 0$$

$$-1, \forall DAC = 0$$

Finalmente, la variable dependiente que se utiliza para entrenar y evaluar la LS-SVM corresponde a la codificación realizada a partir de los resultados de las muestras de DAC. Ejemplo:

Tabla 8.1 Ejemplo codificación registros

Clorofila 0-10	Clorofila 10-20	...	Temperatura	...	Salinidad	DAC	Clases
0,35	0,32	...	13	...	24,5	300	1
0,32	0,33	...	15	...	22,3	0	-1
0,30	0,31	...	12	...	23,0	1000	1
0,31	0,33	...	11	...	25,3	0	-1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

8.3 Normalización y Desfases

El conjunto de datos utilizados en el estudio ha sido normalizado dividiendo cada dato de cada variable por su registro con mayor valor. Como ejemplo se pueden utilizar los 4 primeros registros de la variable *Clorofila 0-10* de la tabla 8.1 y se puede observar que el mayor valor de sus registros corresponde a 0,35. Entonces al dividir los registros de dicha variable por 0,35 se obtienen los siguientes valores normalizados:

<i>Clorofila 0-10</i>
1
0,914
0,857
0,885

El mismo procedimiento se realiza por cada una de las variables independientes.

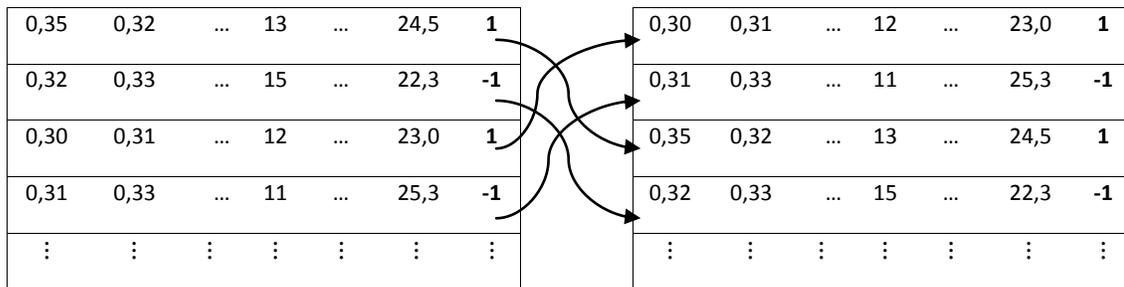
Los desfases son realizados de la manera que se explica en el apartado 7.1.1, con la salvedad de que ahora la variable Y corresponde a la columna Clases. Para las distintas configuraciones del modelo se realizaron pruebas con **desfases desde 0 (sin desfases) hasta 5**. Al configurar el modelo sin desfases solo se entrena la capacidad de clasificación del mismo, en cambio al configurarlo con desfases se le agrega la capacidad de pronóstico.

8.4 Mezcla de Registros

Existen muchos periodos seguidos en los cuales no se han detectado floraciones de *Alexandrium Catenella*. Esta situación puede derivar en que los registros de entrenamiento que ingresan a la LS-SVM sean demasiados homogéneos y por ende se produzca un aprendizaje sesgado que derive en un mal pronosticador.

Para disminuir dicho sesgo, se realiza un proceso de mezcla aleatoria de los registros desfases. Así se pueden generar grupos de datos de training y testing más heterogéneos. A continuación se muestra un ejemplo de dicho proceso, utilizando los registros de la tabla 8.1.

Ejemplo:



8.5 Resultados Modelo LS-SVM PSO

En este apartado se dan a conocer los resultados obtenidos a partir de diferentes configuraciones del modelo LS-SVM, las cuales se distinguen principalmente por el tipo de PSO utilizado. Se implementó el modelo con cuatro variantes de PSO:

- PSO-FI (*PSO con Factor de Inercia Lineal*)
- APSO (*Adaptive Particle Swarm Optimization*)
- FDRPSO (*Fitness Distance Ratio Based Particle Swarm Optimization*)
- AIWPSO (*Adaptive Inertia Weight*)

La función Kernel utilizada en todas las configuraciones corresponde al *Kernel Función de Base Radial*, se elige de entre las funciones Kernels expuestas en el apartado 3.1.3, debido a que según pruebas empíricas preliminares realizadas con los datos de marea roja, es el que mejores resultados entregó.

Los parámetros a optimizar corresponden a C y σ^2 . Donde el primero controla el trade-off entre la maximización del margen y la minimización del error de entrenamiento. Mientras que el segundo corresponde a un parámetro del *Kernel Función de Base Radial*.

Las partículas de las cuatro variantes de PSO están conformadas por el vector bidimensional $[C, \sigma^2]$. El rango de parámetros utilizado para realizar los ensayos en la etapa de entrenamiento se observan en la tabla 8.2:

Tabla 8.2 Parámetros PSO

Parámetros del Modelo	
<i>Parámetros</i>	<i>Rango</i>
Nº Partículas	20 – 30
Número de iteraciones:	60 – 200
w:	[0.4 ; 0.9]
Componente cognitiva:	2
Componente social:	2
Velocidad máxima:	1.6 – 999.9
Posición máxima:	1600 – 5000
Posición mínima:	0.001
Nº desfases:	0 – 5
% entrenamiento:	50 – 70
Validaciones:	Normal – Cruzada

Al realizar algunas pruebas preliminares se detectó una tendencia del pronosticador a predecir correctamente la mayoría de los eventos asociados a la clase -1 (sin marea roja), pero muy pocos de los asociados a la clase 1 (con marea roja). Esto se debe a que existe un sesgo en el aprendizaje debido a que la mayoría de los eventos existentes en la base de datos están asociados a la clase -1. Es por aquello, que se decidió utilizar como medida de evaluación (fitness) **1 – Precisión**, mediante la cual se obtiene el ratio de eventos clasificados erróneamente como pertenecientes a la clase 1, de entre todos los eventos clasificados como 1.

8.5.1 Metodología de Pruebas

Por cada una de las variantes de PSO se realizaron ensayos utilizando 50, 60 y 70 % del total de los registros en la etapa de entrenamiento. Por cada uno de estos conjuntos de registros se realizan 0, 1, 2, 3, 4 y 5 desfases. Además, se utilizaron diferentes configuraciones de parámetros (número de partículas, tamaño del enjambre, velocidad máxima de una partícula, etc...). De los distintos ensayos se escogieron los mejores resultados obtenidos para los diferentes desfases, utilizando tanto validación normal como validación cruzada. Posteriormente, se validan dichos resultados con el 50, 40 o 30% restante de la data y se

comparan los resultados a partir de las variaciones porcentuales de sus medidas de evaluación, enfatizando en la precisión y en la sensibilidad.

8.5.2 Resultados PSO Con Factor de Inercia

PSO con factor de inercia utiliza el parámetro ω , para regular el impacto de las velocidades en la nueva partícula. ω se comporta de forma decreciente según avanzan las iteraciones, generalmente disminuyendo desde 0,9 a 0,4 con $C_1 + C_2 = 4$. Esta disminución sostenida en el tiempo induce un cambio en el algoritmo desde un modo de búsqueda exploratoria a un modo de búsqueda de explotación. ω se calcula según la siguiente formula

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{iter_{max}} \cdot iter$$

Con validación normal, se logró el mejor resultado entrenando el modelo con 20 partículas, 60 iteraciones, 4 desfases y con el 70% del total de los registros. Los parámetros óptimos obtenidos fueron $C = 6,2082$ y $\sigma^2 = 0,5612$.

En la tabla 8.3 (validación normal), se pueden observar los resultados de las medidas de evaluación obtenidos en la etapa de testing. La *Precisión* explica que el 80% de los eventos con marea roja fueron clasificados correctamente, de entre el total de eventos clasificados de esa forma. Con la *Exactitud* se indica que el 84% de los eventos fueron clasificados correctamente (con o sin marea roja). La *Sensibilidad* explica que el 70% de los eventos con marea roja fueron clasificados correctamente. Luego, la *Especificidad* declara que el 91% de los eventos sin marea roja fueron clasificados correctamente. Finalmente, un 86% de los eventos sin marea roja (VPN) fueron clasificados correctamente, de entre el total de eventos clasificados de esa forma.

Tabla 8.3 Resultado Modelo LS-SVM PSO Con Factor de Inercia. Etapa testing.

RESULTADOS						
	Validación Normal			Validación cruzada		
% de Entrenamiento	50	60	70	50	60	70
Desfases	5	5	4	5	4	5
Precisión	0,74	0,68	0,80	0,79	0,80	0,85
Exactitud	0,84	0,80	0,84	0,81	0,85	0,82
Sensibilidad	0,66	0,71	0,70	0,60	0,68	0,74
Especificidad	0,91	0,85	0,91	0,91	0,92	0,88
VPN	0,88	0,86	0,86	0,81	0,87	0,80

Con validación cruzada, se logró el mejor resultado entrenando el modelo con 20 partículas, 60 iteraciones, 5 desfases y con el 70% del total de los registros. Además, este subconjunto de registros se dividió en 8 grupos, para aplicar la validación. Los parámetros óptimos obtenidos fueron $C = 8,6748$ y $\sigma^2 = 0,5082$.

En la tabla 8.3 (validación cruzada) se pueden observar los resultados de las medidas de evaluación obtenidos en la etapa de testing. La *Precisión* explica que el 85% de los eventos

con marea roja fueron clasificados correctamente, de entre el total de eventos clasificados de esa forma. Con la *Exactitud* se indica que el 82% de los eventos fueron clasificados correctamente (con o sin marea roja). La *Sensibilidad* explica que el 74% de los eventos con marea roja fueron clasificados correctamente. Luego, la *Especificidad* declara que el 88% de los eventos sin marea roja fueron clasificados correctamente. Finalmente, un 80% de los eventos sin marea roja (VPN) fueron clasificados correctamente, de entre el total de eventos clasificados de esa forma.

8.5.3 Resultados APSO

En el proceso de evolución, el enjambre podría someterse a un proceso no deseado de pérdida de la diversidad. Esto quiere decir que algunas partículas podrían comenzar a perder su capacidad de búsqueda global y local en las sucesivas generaciones. Para evitar aquello APSO realiza un proceso de reinicio de las partículas cuando detecta que están perdiendo su capacidad de búsqueda global y local, esto ocurre cuando la ubicación del *pbest* está muy cerca del *gbest* y su velocidad es cercana a 0.

La condición para reiniciar la partícula corresponde a determinar la diferencia del Fitness de la partícula *i* con la de la mejor global, si dicha variación es menor a ε se opta por reiniciar la partícula.

$$|\Delta Fitness| < \varepsilon$$

Las pruebas se realizaron utilizando un ε de 0,02.

Con *validación normal*, se logró el mejor resultado entrenando el modelo con 20 partículas, 60 iteraciones, 5 desfases y con el 70% del total de los registros. Los parámetros óptimos obtenidos fueron $C = 7,2632$ y $\sigma^2 = 0,7506$.

En la tabla 8.4 (validación normal), se pueden observar los resultados de las medidas de evaluación obtenidos en la etapa de testing. La *Precisión* explica que el 85% de los eventos con marea roja fueron clasificados correctamente, de entre el total de eventos clasificados de esa forma. Con la *Exactitud* se indica que el 89% de los eventos fueron clasificados correctamente (con o sin marea roja). La *Sensibilidad* explica que el 80% de los eventos con marea roja fueron clasificados correctamente. Luego, la *Especificidad* declara que el 93% de los eventos sin marea roja fueron clasificados correctamente. Finalmente, un 91% de los eventos sin marea roja (VPN) fueron clasificados correctamente, de entre el total de eventos clasificados de esa forma.

Tabla 8.4 Resultado Modelo LS-SVM APSO.

RESULTADOS						
	Validación Normal			Validación cruzada		
% de Entrenamiento	50	60	70	50	60	70
Desfases	5	5	5	5	5	5
Precisión	0,78	0,77	0,85	0,76	0,73	0,86
Exactitud	0,85	0,86	0,89	0,79	0,83	0,83
Sensibilidad	0,74	0,70	0,80	0,58	0,70	0,70
Especificidad	0,90	0,92	0,93	0,90	0,88	0,92
VPN	0,88	0,89	0,91	0,80	0,87	0,82

Con validación cruzada, se logró el mejor resultado entrenando el modelo con 30 partículas, 200 iteraciones, 5 desfases y con el 70% del total de los registros. Además, este subconjunto de registros se dividió en 8 grupos, para aplicar la validación. Los parámetros óptimos obtenidos fueron $C = 2064,1684$ y $\sigma^2 = 4,9293$.

En la tabla 8.4 (validación cruzada) se pueden observar los resultados de las medidas de evaluación obtenidos en la etapa de testing. La *Precisión* explica que el 86% de los eventos con marea roja fueron clasificados correctamente, de entre el total de eventos clasificados de esa forma. Con la *Exactitud* se indica que el 83% de los eventos fueron clasificados correctamente (con o sin marea roja). La *Sensibilidad* explica que el 70% de los eventos con marea roja fueron clasificados correctamente. Luego, la *Especificidad* declara que el 92% de los eventos sin marea roja fueron clasificados correctamente. Finalmente, un 82% de los eventos sin marea roja (VPN) fueron clasificados correctamente, de entre el total de eventos clasificados de esa forma.

8.5.4 Resultados FDRPSO

FDRPSO propone que las partículas también puedan ser capaces de aprender de la experiencia de una partícula vecina. Para escoger a la partícula vecina adecuada se utilizan los siguientes criterios: debe estar cerca de la partícula que se está actualizando y debe haber visitado una posición con mejor fitness. FDRPSO implementa un tercer *Componente* (C_3), el cual es utilizado con el valor 2.

Con validación normal, se logró el mejor resultado entrenando el modelo con 20 partículas, 60 iteraciones, 3 desfases y con el 60% del total de los registros. Los parámetros óptimos obtenidos fueron $C = 7,0922$ y $\sigma^2 = 0,3121$.

En la tabla 8.5 (validación normal), se pueden observar los resultados de las medidas de evaluación obtenidos en la etapa de testing. La *Precisión* explica que el 78% de los eventos con marea roja fueron clasificados correctamente, de entre el total de eventos clasificados de esa forma. Con la *Exactitud* se indica que el 85% de los eventos fueron clasificados correctamente (con o sin marea roja). La *Sensibilidad* explica que el 75% de los eventos con marea roja fueron clasificados correctamente. Luego, la *Especificidad* declara que el 92% de los eventos sin marea roja fueron clasificados correctamente. Finalmente, un 90% de los

eventos sin marea roja (VPN) fueron clasificados correctamente, de entre el total de eventos clasificados de esa forma.

Tabla 8.5 Resultado Modelo LS-SVM FDRPSO.

RESULTADOS						
	Validación Normal			Validación cruzada		
% de Entrenamiento	50	60	70	50	60	70
Desfases	4	3	4	4	5	5
Precisión	0,72	0,78	0,84	0,75	0,78	0,81
Exactitud	0,83	0,85	0,86	0,80	0,83	0,86
Sensibilidad	0,72	0,75	0,57	0,56	0,64	0,78
Especificidad	0,88	0,92	0,96	0,91	0,91	0,90
VPN	0,88	0,90	0,86	0,81	0,84	0,88

Con validación cruzada, se logró el mejor resultado entrenando el modelo con 30 partículas, 200 iteraciones, 5 desfases y con el 70% del total de los registros. Además, este subconjunto de registros se dividió en 8 grupos, para aplicar la validación. Los parámetros óptimos obtenidos fueron $C = 77,6267$ y $\sigma^2 = 2,6476$.

En la tabla 8.5 (validación cruzada) se pueden observar los resultados de las medidas de evaluación obtenidos en la etapa de testing. La *Precisión* explica que el 81% de los eventos con marea roja fueron clasificados correctamente, de entre el total de eventos clasificados de esa forma. Con la *Exactitud* se indica que el 86% de los eventos fueron clasificados correctamente (con o sin marea roja). La *Sensibilidad* explica que el 78% de los eventos con marea roja fueron clasificados correctamente. Luego, la *Especificidad* declara que el 90% de los eventos sin marea roja fueron clasificados correctamente. Finalmente, un 88% de los eventos sin marea roja (VPN) fueron clasificados correctamente, de entre el total de eventos clasificados de esa forma.

8.5.5 Resultados AIWPSO

Los métodos tradicionales utilizados para actualizar los factores de inercia o pesos de inercia se basan en el fitness o en el número de iteraciones. AIWPSO propone un método en donde se utiliza la tasa de existo del enjambre para determinar el peso de inercia.

Con validación normal, se logró el mejor resultado entrenando el modelo con 20 partículas, 60 iteraciones, 5 desfases y con el 60% del total de los registros. Los parámetros óptimos obtenidos fueron $C = 9,7543$ y $\sigma^2 = 0,8365$.

En la tabla 8.6 (validación normal), se pueden observar los resultados de las medidas de evaluación obtenidos en la etapa de testing. La *Precisión* explica que el 84% de los eventos con marea roja fueron clasificados correctamente, de entre el total de eventos clasificados de esa forma. Con la *Exactitud* se indica que el 82% de los eventos fueron clasificados correctamente (con o sin marea roja). La *Sensibilidad* explica que el 63% de los eventos con marea roja fueron clasificados correctamente. Luego, la *Especificidad* declara que el 92% de

los eventos sin marea roja fueron clasificados correctamente. Finalmente, un 81% de los eventos sin marea roja (VPN) fueron clasificados correctamente, de entre el total de eventos clasificados de esa forma.

Tabla 8.6 Resultado Modelo LS-SVM AIWPSO.

RESULTADOS						
	Validación Normal			Validación cruzada		
% de Entrenamiento	50	60	70	50	60	70
Desfases	4	5	5	5	3	5
Precisión	0,70	0,84	0,75	0,73	0,76	0,81
Exactitud	0,82	0,82	0,85	0,76	0,76	0,86
Sensibilidad	0,72	0,63	0,66	0,55	0,57	0,78
Especificidad	0,87	0,92	0,91	0,88	0,89	0,90
VPN	0,88	0,81	0,88	0,78	0,77	0,88

Con *validación cruzada*, se logró el mejor resultado entrenando el modelo con 30 partículas, 200 iteraciones, 5 desfases y con el 70% del total de los registros. Además, este subconjunto de registros se dividió en 8 grupos, para aplicar la validación. Los parámetros óptimos obtenidos fueron $C = 49,6978$ y $\sigma^2 = 1,9832$.

En la tabla 8.6 (validación cruzada) se pueden observar los resultados de las medidas de evaluación obtenidos en la etapa de testing. La *Precisión* explica que el 81% de los eventos con marea roja fueron clasificados correctamente, de entre el total de eventos clasificados de esa forma. Con la *Exactitud* se indica que el 86% de los eventos fueron clasificados correctamente (con o sin marea roja). La *Sensibilidad* explica que el 78% de los eventos con marea roja fueron clasificados correctamente. Luego, la *Especificidad* declara que el 90% de los eventos sin marea roja fueron clasificados correctamente. Finalmente, un 88% de los eventos sin marea roja (VPN) fueron clasificados correctamente, de entre el total de eventos clasificados de esa forma.

9 Análisis de Resultados

A continuación se comparan los resultados obtenidos a partir de las pruebas realizadas con las distintas configuraciones del modelo. Se consideran las medidas de evaluación definidas en el capítulo 6 para seleccionar los mejores modelos, haciendo hincapié en la precisión y en la sensibilidad, debido a que estas medidas se relacionan con la capacidad del pronosticador de predecir correctamente aquellos eventos en los cuales existe marea roja.

En total se utilizaron 314 registros, los cuales fueron obtenidos a partir de las muestras rescatadas desde las 9 estaciones que monitorean el núcleo de toxicidad “Última Esperanza”, ubicado en la región de Magallanes.

9.1 Análisis Por Modelo

En este apartado se comparan los resultados obtenidos, a partir de las dos validaciones utilizadas. Esto se realiza por cada una de las variantes del modelo general.

9.1.1 Modelo PSO – FI

Al comparar los resultados obtenidos utilizando validación normal y validación cruzada, se puede observar que con validación cruzada se obtuvo un mejor resultado, explicado por una diferencia de un 5,88% en precisión y de 5,40% en sensibilidad por sobre los resultados generados a partir del uso de validación normal (ver tabla 9.1).

Tabla 9.1 Comparación porcentual de resultados por validación. Modelo PSO - FI

	PSO FI (Val. normal)	
	Diferencia % (Precisión)	Diferencia % (Sensibilidad)
PSO FI (Val. cruzada)	5,88	5,40

La configuración utilizada para construir el modelo, utilizando validación cruzada, se observa en la tabla 9.2.

Tabla 9.2 Configuración entrenamiento PSO -FI

TRAINING			
<i>Parámetros de entrada</i>		<i>Parámetros de salida</i>	
Nº Partículas:	20	C :	8,6748
Número de iteraciones:	60	σ^2 :	0,5082
Nº desfases:	5		
% entrenamiento:	70		

A partir de esta configuración se obtuvo: 85% de precisión, 82% de exactitud, 74% de sensibilidad, 88% de especificidad y 80% de VPN. Al interpretar la precisión y la sensibilidad en función del problema se puede entender que el 85% de los eventos con marea roja fueron clasificados correctamente, sobre el total de eventos clasificados (correcta o incorrectamente) por la prueba de dicha forma. Además, un 74% de los eventos fueron clasificados

correctamente con marea roja considerando todos los eventos que realmente poseen marea roja.

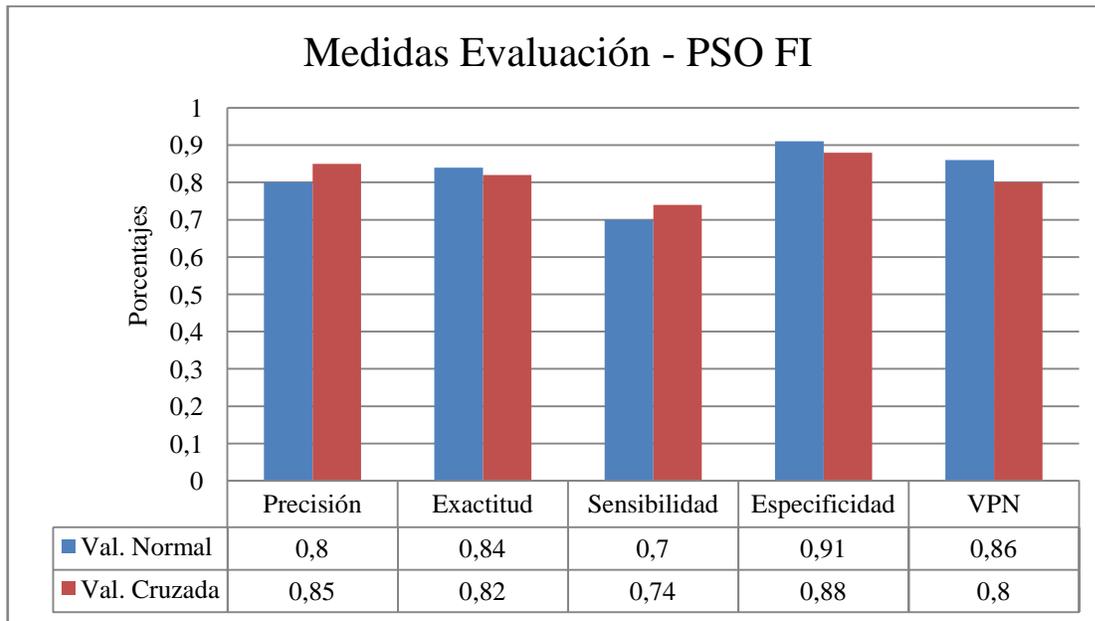


Figura 9.1 Medidas de evaluación. PSO-FI.

9.1.2 Modelo APSO

Al analizar los resultados obtenidos utilizando validación normal y cruzada, se puede observar que con validación normal se logró un mejor resultado. Ya que, si bien, se obtuvo una precisión inferior en un 1,16%, todas sus medidas de evaluación superaron el 80% y la sensibilidad se sobrepuso en un 12,5% a su similar en validación cruzada (ver tabla 9.3).

Tabla 9.3 Comparación porcentual de resultados por validación. Modelo APSO.

	APSO (Val. cruzada)	
	Diferencia % (Precisión)	Diferencia % (Sensibilidad)
APSO (Val. normal)	-1,16	12,5

La configuración utilizada para construir el modelo, utilizando validación normal, se observa en la tabla 9.4.

Tabla 9.4 Configuración entrenamiento APSO.

TRAINING			
Parámetros de entrada		Parámetros de salida	
Nº Partículas:	20	C :	7,2632
Número de iteraciones:	60	σ^2 :	0,7506
Nº desfases:	5		
% entrenamiento:	70		
Error (ε):	0,02		

A partir de esta configuración se obtuvo: 85% de precisión, 89% de exactitud, 80% de sensibilidad, 93% de especificidad y 91% de VPN. Al interpretar la precisión y la sensibilidad en función del problema se puede entender que el 85% de los eventos con marea roja fueron clasificados correctamente, sobre el total de eventos clasificados (correcta o incorrectamente) por la prueba de dicha forma. Además, un 80% de los eventos fueron clasificados correctamente con marea roja considerando todos los eventos que realmente poseen marea roja.

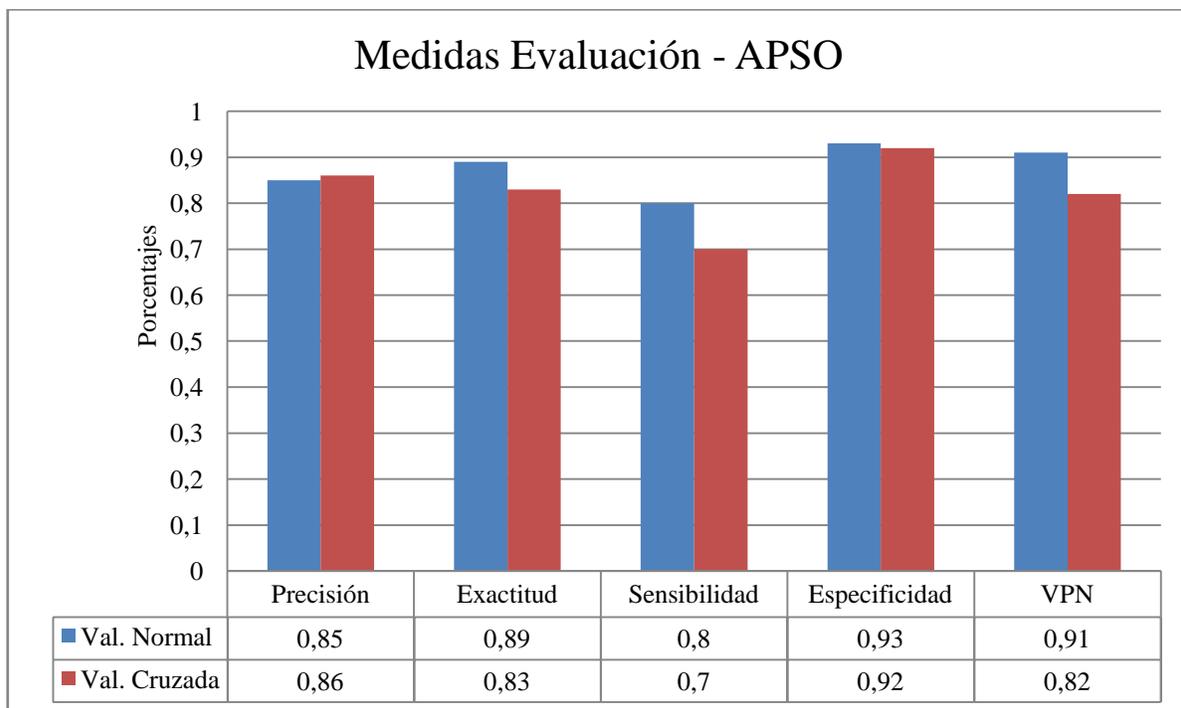


Figura 9.2 Medidas de evaluación. APSO.

9.1.3 Modelo FDRPSO

Al comparar los resultados obtenidos al utilizar validación normal y validación cruzada se observa que con validación cruzada se obtuvo un mejor resultado, explicado por una diferencia de un 3,70% en precisión y de 3,84% en sensibilidad (ver tabla 9.5).

Tabla 9.5 Comparación porcentual de resultados por validación. Modelo FDRPSO.

FDRPSO (Val. cruzada)	FDRPSO (Val. normal)	
	Diferencia % (Precisión)	Diferencia % (Sensibilidad)
	3,70	3,84

La configuración utilizada para construir el modelo, utilizando validación cruzada, se observa en la tabla 9.6.

Tabla 9.6 Configuración entrenamiento FDRPSO.

TRAINING			
Parámetros de entrada		Parámetros de salida	
Nº Partículas:	30	C :	77,6267
Número de iteraciones:	200	σ^2 :	2,6476
Nº desfases:	5		
% entrenamiento:	70		
Componente (C_3):	2		

A partir de esta configuración, se obtuvo en la etapa de testing: 81% de precisión, 86% de exactitud, 78% de sensibilidad, 90% de especificidad y 88% de VPN. Al interpretar la precisión y la sensibilidad en función del problema se puede entender que el 81% de los eventos con marea roja fueron clasificados correctamente, sobre el total de eventos clasificados (correcta o incorrectamente) por la prueba de dicha forma. Además, un 78% de los eventos fueron clasificados correctamente con marea roja considerando todos los eventos que realmente poseen marea roja.

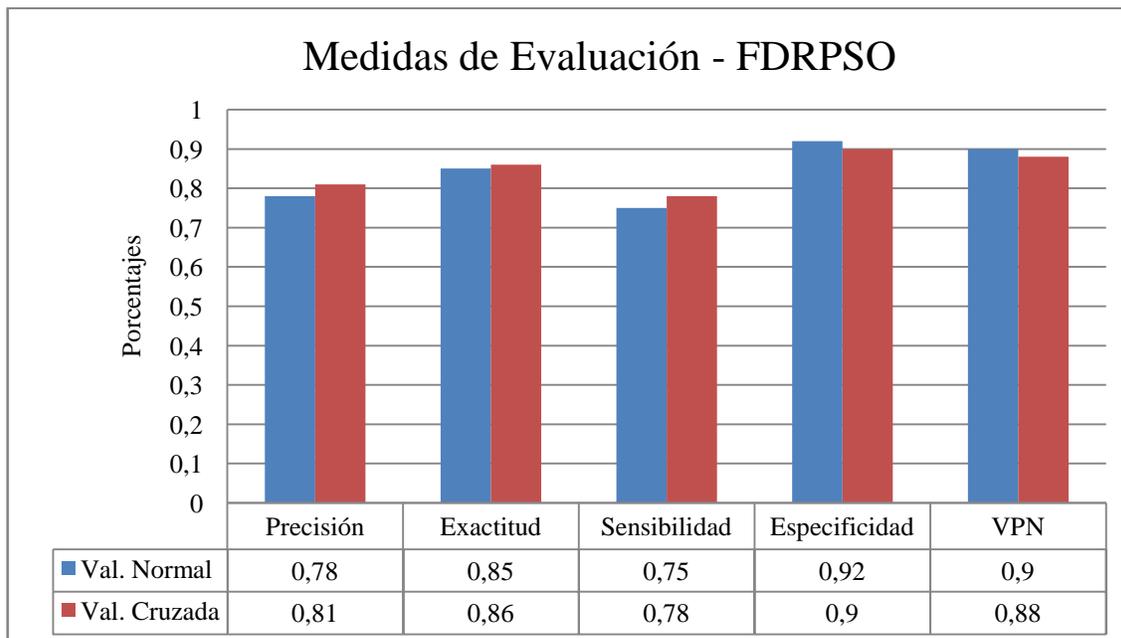


Figura 9.3 Medidas de evaluación. FDRPSO.

9.1.4 Modelo AIWPSO

Al analizar los resultados obtenidos utilizando validación normal y cruzada, se observa que al promediar las diferencias porcentuales obtenidas a partir de la precisión y de la sensibilidad se logró un mejor rendimiento utilizando validación cruzada. Alcanzando una diferencia porcentual de -3,70% en precisión y de 19,23% en sensibilidad (ver tabla 9.7).

Tabla 9.7 Comparación porcentual de resultados por validación. Modelo AIWPSO.

	AIWPSO (Val. normal)	
	Diferencia % (Precisión)	Diferencia % (Sensibilidad)
AIWPSO (Val. cruzada)	-3,70	19,23

La configuración utilizada para construir el modelo, utilizando validación cruzada, se observa en la tabla 9.8.

Tabla 9.8 Configuración entrenamiento AIWPSO.

TRAINING			
<i>Parámetros de entrada</i>		<i>Parámetros de salida</i>	
Nº Partículas:	30	C :	49,6978
Número de iteraciones:	200	σ^2 :	1,9832
Nº desfases:	5		
% entrenamiento:	70		

A partir de esta configuración, se obtuvo en la etapa de testing: 81% de precisión, 86% de exactitud, 78% de sensibilidad, 90% de especificidad y 88% de VPN. Al interpretar la precisión y la sensibilidad en función del problema se puede entender que el 81% de los eventos con marea roja fueron clasificados correctamente, sobre el total de eventos clasificados (correcta o incorrectamente) por la prueba de dicha forma. Además, un 78% de los eventos fueron clasificados correctamente con marea roja considerando todos los eventos que realmente poseen marea roja.

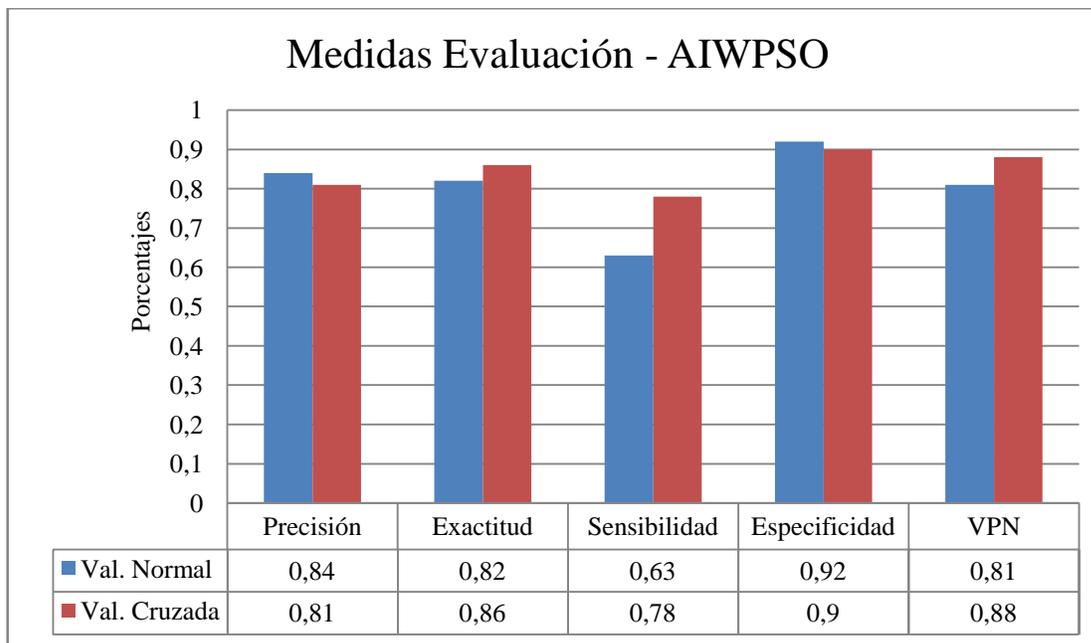


Figura 9.4 Medidas de evaluación. AIWPSO.

9.2 Análisis General

Al comparar los resultados de los modelos antes expuestos, se observa que utilizando APSO se obtuvieron métricas de buena calidad, dado que todas superaron el 80%. Además, se visualiza un predominio de APSO por sobre PSO FI, FDRPSO y AIWPSO, al conseguir diferencias porcentuales positivas (ver tabla 9.9) en sus medidas de evaluación. En la tabla 9.9 se muestran dichas diferencias. En la figura 9.5 se expone una grafica en donde se contrastan los resultados de las medidas de evaluación obtenidas por los distintos modelos.

Tabla 9.9 Diferencia porcentual APSO v/s PSO FI, FDRPSO y AIWPSO. Mejor Modelo.

	PSO FI				
	Precisión	Exactitud	Sensibilidad	Especificidad	VPN
APSO	0	7,86	7,50	5,37	12,08
	FDRPSO				
	Precisión	Exactitud	Sensibilidad	Especificidad	VPN
APSO	4,70	3,37	2,50	3,22	3,29
	AIWPSO				
	Precisión	Exactitud	Sensibilidad	Especificidad	VPN
APSO	4,70	3,37	2,50	3,22	3,29

Al analizar el área bajo la curva de las gráficas ROC, de los modelos desarrollados, se ratifica a APSO como el modelo con mejor resultado en comparación a PSO FI, FDRPSO y AIWPSO, obteniendo 0,0727, 0,0138 y 0,0138 puntos más respectivamente. En la tabla 9.10 se observan las áreas bajo la curva ROC de cada uno de los modelos, en la tabla 9.11 se aprecian las diferencias puntuales de APSO con el resto de los modelos. Finalmente, en la figura 9.6 se comparan las curvas ROC de los cuatro modelos y se aprecia que la curva

asociada a APSO se acerca más a la esquina superior izquierda, lo que cualitativamente supone una superioridad en la exactitud de su clasificación. En las figuras 9.7, 9.8, 9.9 y 9.10 se exponen las curvas ROC correspondientes a cada uno de los modelos y se indica el área bajo la curva obtenida por cada una de ellos.

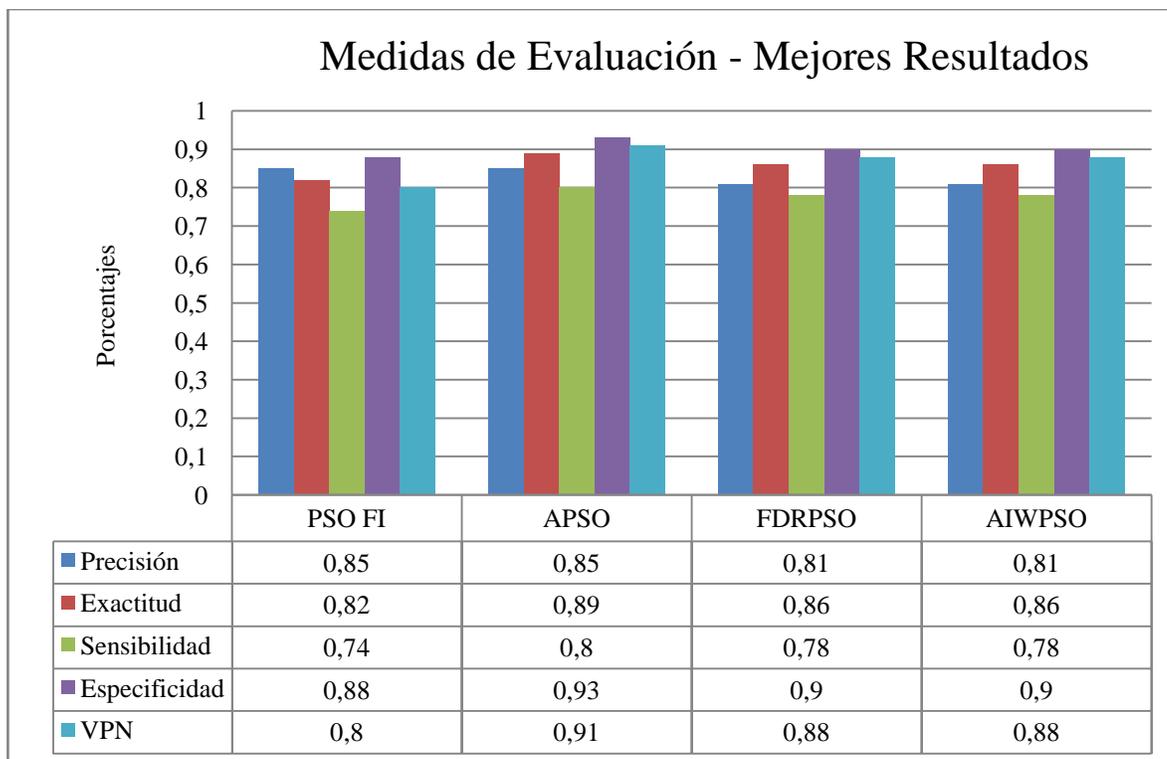


Figura 9.5 Medidas de evaluación. Mejores Resultados.

Tabla 9.10 Áreas bajo la curva ROC.

	APSO	PSO FI	FDRPSO	AIWPSO
Área bajo la curva	0,9319	0,8592	0,9181	0,9181

Tabla 9.11 Diferencias de puntos de área bajo la curva ROC. APSO v/s PSO FI, FDRPSO, AIWPSO.

Diferencia % APSO – PSO FI:	0,0727
Diferencia % APSO – FDRPSO:	0,0138
Diferencia % APSO – AIWPSO:	0,0138

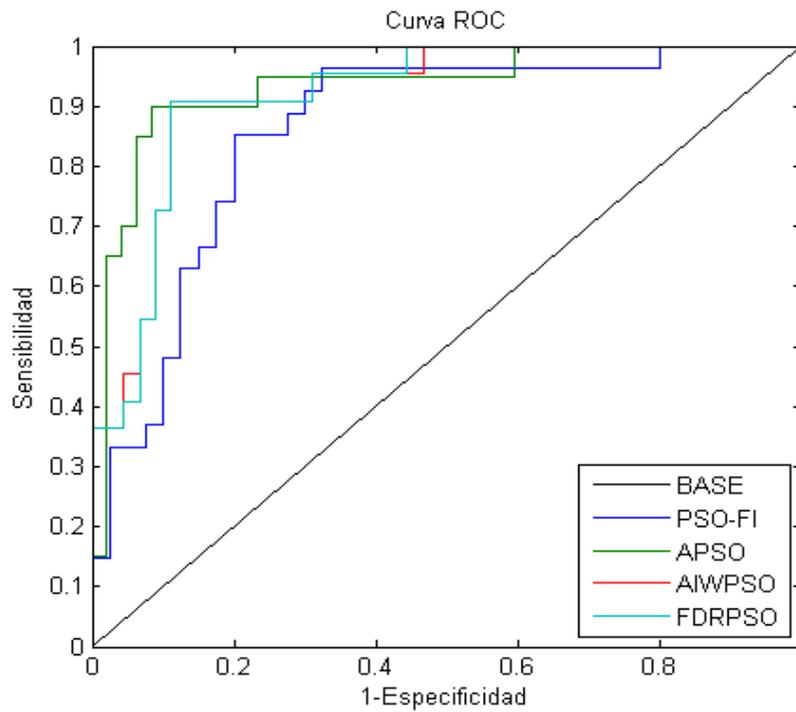


Figura 9.6 Curvas ROC PSO-FI, APSO, AIWPSO y FDRPSO.

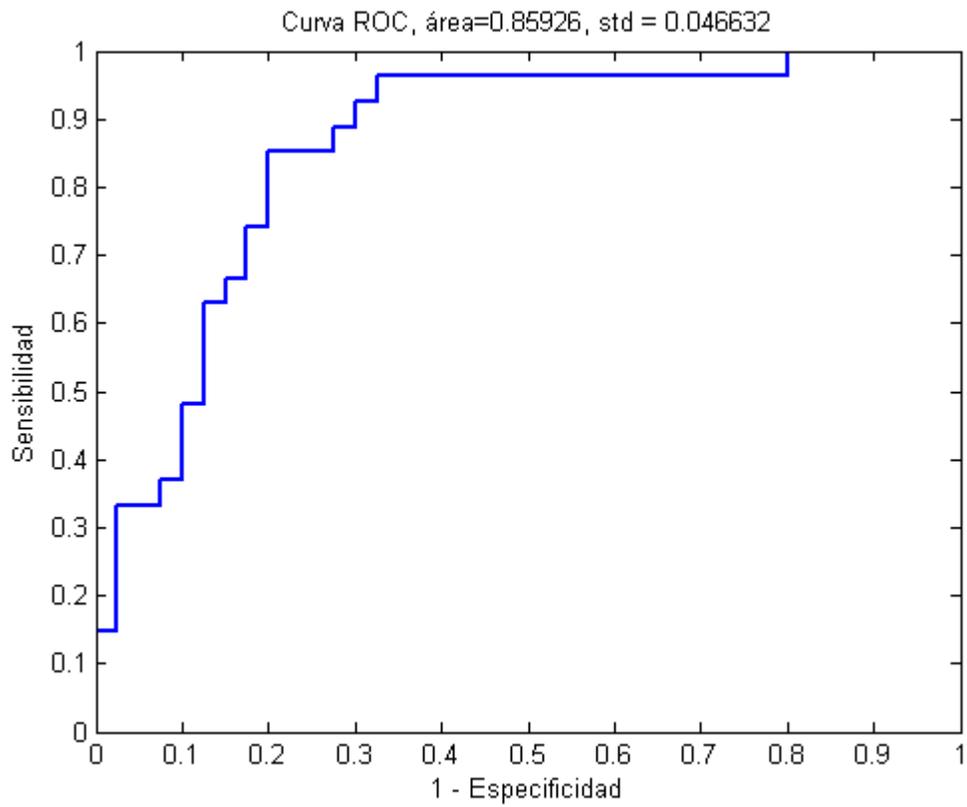


Figura 9.7 Curva ROC PSO-FI.

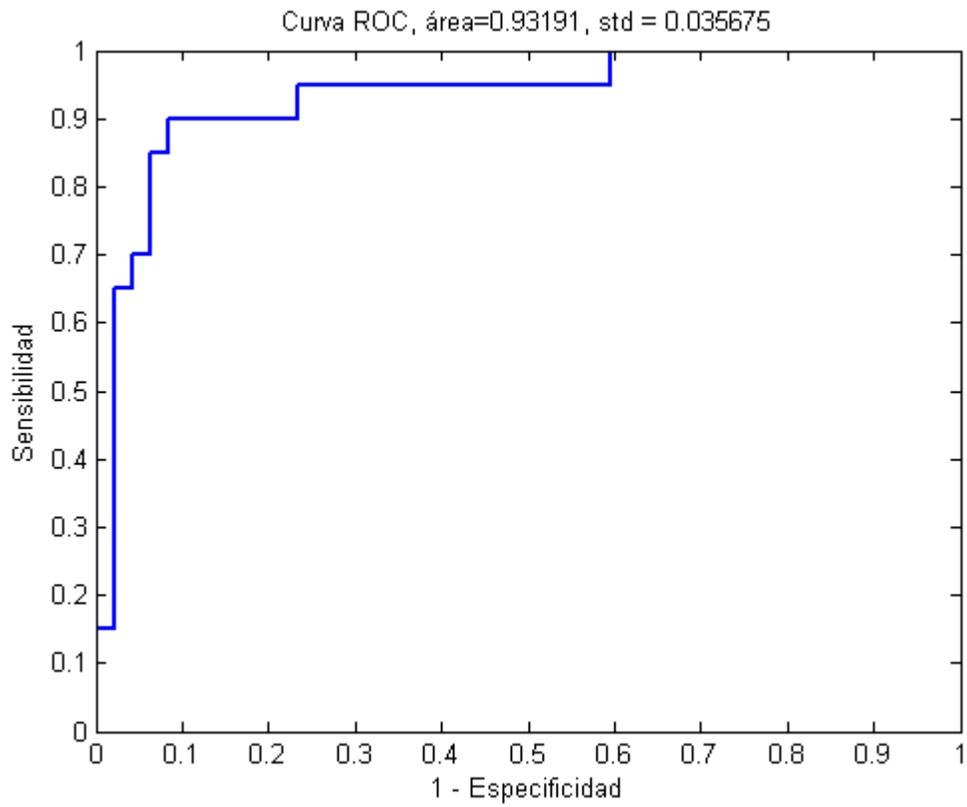


Figura 9.8 Curva ROC APSO.

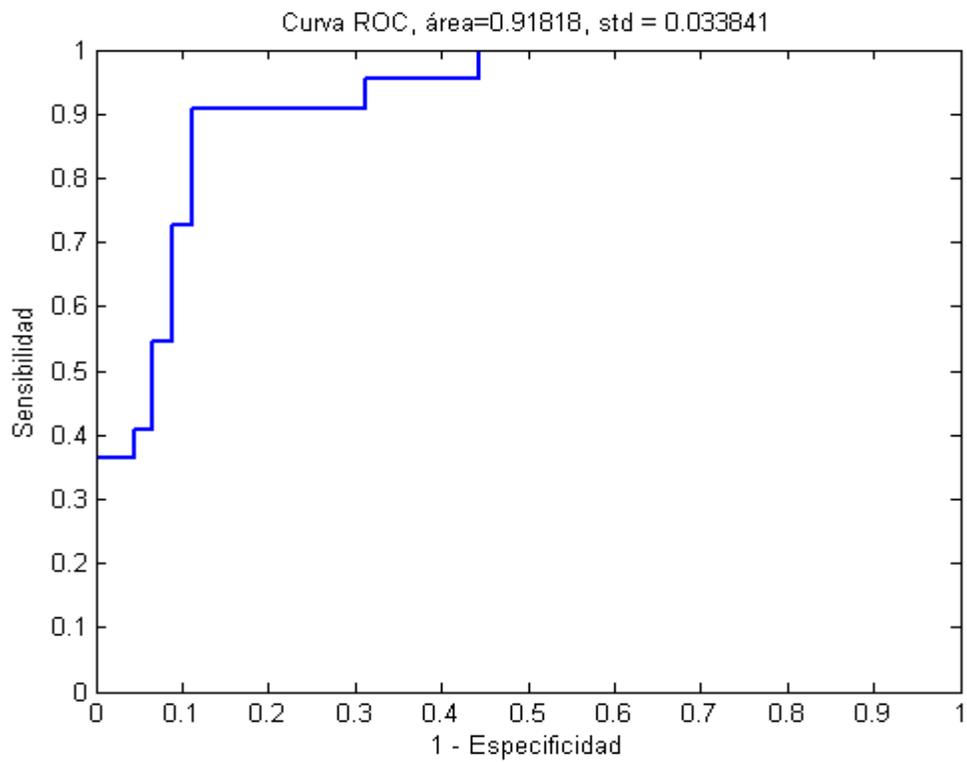


Figura 9.9 Curva ROC FDRPSO.

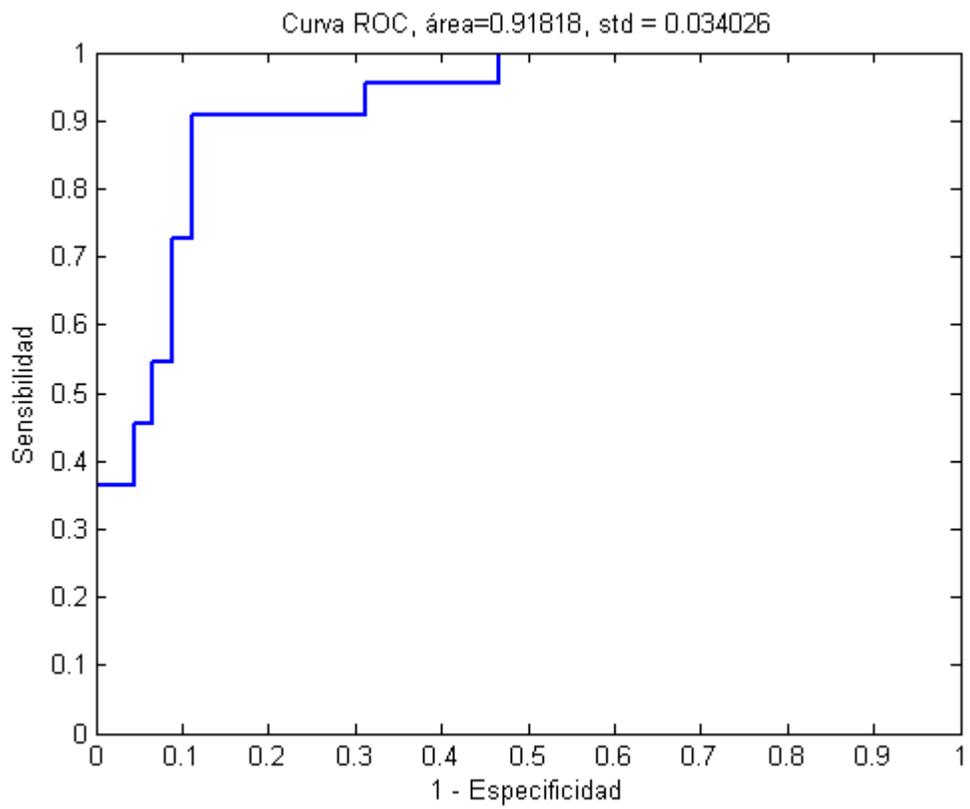


Figura 9.10 Curva ROC AIWPSO.

10 Conclusiones

Se presentaron los conceptos asociados a los fenómenos de marea roja que contextualizan el problema y se desarrolla un marco conceptual, en el cual se explican los fundamentos de las técnicas que sirvieron como base para el diseño e implementación del modelo de pronóstico.

Utilizando LS-SVM y PSO se diseña la estructura del modelo, la cual cuenta principalmente de dos etapas: entrenamiento y evaluación. Para estimar los parámetros de la LS-SVM se utilizaron cuatro variantes de PSO (PSO FI, APSO, FDRPSO y AIWPSO). Como fitness o función de costo se utiliza $1 - \text{Precisión}$. Además, se realizaron dos tipos de validaciones para evaluar los fitness: validación normal y cruzada.

Los principales resultados se obtuvieron utilizando un 70% del total de registros para entrenamiento y el 30% restante para evaluación. Los números de desfases utilizados para realizar las pruebas variaron dentro del intervalo [1 ; 5], destacando los resultados obtenidos al utilizar 5 desfases.

A partir del análisis de los distintos modelos propuestos se concluye que LS-SVM con APSO, utilizando validación cruzada, es el que mejores resultados logró, según sus medidas de evaluación. También se destaca que los resultados de todas sus métricas estuvieron sobre un 80%. Los valores obtenidos por el modelo LS-SVM APSO corresponden a los siguientes: 85% de precisión, 89% en exactitud, 80% de sensibilidad, 93 de especificidad y un 91% de valor predictivo negativo. Al interpretar la precisión y la sensibilidad en el contexto del problema se puede determinar que el 85% de los eventos con marea roja fueron clasificados correctamente, sobre el total de eventos clasificados (correcta o incorrectamente) por la prueba de dicha forma. Además, un 80% de los eventos fueron clasificados correctamente con marea roja considerando todos los eventos que realmente poseen marea roja.

Se verifica que las Máquinas de Soporte Vectorial fueron capaces de aprender de los registros históricos de marea roja pertenecientes a la región de Magallanes, específicamente al sector “Última Esperanza”, esto a pesar de que existe un sesgo hacia el número de registros con ausencia de floraciones de *Alexandrium Catenella*, o sea sin marea roja. Es por aquello, que para discriminar la calidad de los modelos se utiliza principalmente la precisión y la sensibilidad.

Para medir el grado de reproductividad del modelo, es necesario que se mantengan los muestreos, en el tiempo, de las zonas afectadas por marea roja, específicamente desde las estaciones pertenecientes a Última Esperanza. Esto deriva en la posibilidad de desarrollar nuevos proyectos relacionados con el presente, como puede ser la calibración o reconstrucción del modelo utilizando los datos correspondientes al 2011. Además, es posible utilizar los datos del resto de las estaciones para construir un modelo más general para la región de Magallanes. También sería de gran interés conocer cuáles de las 19 variables empleadas como independientes en la construcción del modelo, son las que más influyen en el pronóstico de *Alexandrium Catenella*.

Referencias

- [1] Guzmán Mendez, Leonardo, et al. Informe Final SUBPESCA, Manejo y monitoreo de las mareas rojas en las regiones de Los Lagos, Aysén y Magallanes, Chile, 2007.
- [2] Guzmán Mendez, Leonardo, et al, Informe Final SUBPESCA, Manejo y monitoreo de las mareas rojas en las regiones de Los Lagos, Aysén y Magallanes, periodo 2007-2008, Chile, 2009.
- [3] Guzmán Mendez, Leonardo, et al, Informe Final Corregido Tomo 1 (Texto Informe), Manejo y monitoreo de las mareas rojas en las regiones de Los Lagos, Aysén y Magallanes 2008-2009, SUBPESCA, Chile, 2010.
- [4] Suarez Isla, Benjamín y Guzmán Méndez Leonardo, Floraciones de Algas Nocivas, Mareas Rojas y Toxinas Marinas, Chile, 1998.
- [5] Fdez-Riverola, Florentino y Corchado, Juan, FSfRT Forecasting System for Red Tides. Applied Intelligence 21, pp 251–264, Kluwer Academic Publishers. 2004.
- [6] Berthold, Michael, et al., Guide to Intelligent Data Analysis, How to Intelligently Make Sense of Real Data, Springer, 2010.
- [7] Gentien, Patrick, et al., Global Ecology and Oceanography of Harmful Algal Blooms, Science Pla, SCOR and IOC, Baltimore and Paris, 86 pp, 2001.
- [8] Gentien, Patrick, et al., Global Ecology and Oceanography of Harmful Algal Blooms, Implementation Plan, SCOR and IOC, Baltimore and Paris, 36 pp, 2003.
- [9] Cembella, A., et al., Global Ecology and oceanography of harmful algal blooms, GEOHAB Core Research Project: HABs in Fiords and Coastal Embayments, IOC and SCOR, Paris, France and Newark, Delaware USA, 57 pp, 2010.
- [10] Suárez Isla, et al., Impacto económico de las floraciones de microalgas nocivas en Chile y datos recientes sobre la ocurrencia de Veneno Amnésico de los Mariscos, Floraciones de algas nocivas en el Cono Sur Americano, 2002.
- [11] Lembeye, Georgina, Florecimiento algal nocivo causado por Alexandrium Catenella, Propuesta de área FAN (D. S. MINECON N° 345/2005), Informe técnico D. AC. N° 2168 / 2008, 2008.
- [12] Agencia Chilena para la inocuidad alimentaria, Componente Marea Roja o Floraciones Algas Nocivas, 2010.
- [13] Corchado, J., Torres, J., and Sacai, M., Forecasting Red Tides Caused by Pseudo-nitzschia spp. Using an Artificial Intelligence Model, K.A. Steidinger, J.H. Landsberg, C.R. Tomas, and G.A. Vargo [eds.] Harmful Algae 2002. Florida Fish and Wildlife Conservation Commission, Florida Institute of Oceanography, and Intergovernmental Oceanographic Commission of UNESCO, 2004.

- [14] Herrera, Luis, Tesis doctoral, Sistemas inteligentes para aproximación y predicción utilizando arquitecturas avanzadas, Universidad de Granada, 2007.
- [15] Schölkopf, B., Support Vector Learning. R. Oldenbourg Verlag, Munchen, Doktorarbeit, Technische Universität Berlin, 1997.
- [16] Gibaj, J., Aprendizaje Estadístico con Funciones Kernel, Trabajo Fin de Master, Máster en Matemáticas Avanzadas, Universidad Nacional de Educación a Distancia, 2010.
- [17] Smola, A., Schölkopf, B., A Tutorial on Support Vector Regression, NeuroCOLT2 Technical Report Series, NC2-TR-1998-030, 1998.
- [18] Gordillo, J., Optimization Based Methods for Classification and Regression Problems with Imprecise Data, Tesis Doctoral, Universidad de Sevilla, 2008.
- [19] Vapnik, V., The Natural of Statistical Learning Theor, Springer, 1995.
- [20] Cristianini, N., Shawe-Taylor, J., An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, published by the Press Syndicate Of The University Of Cambridge, 2000.
- [21] Suykens, J., et al., Least Squares Support Vector Machines, 2002.
- [22] Kennedy, J., Eberhart, R., Particle Swarm Optimization, Proc. IEEE Int. Conf. on N.N., pp. 1942-1948, 1995.
- [23] Eberhart, R., Kennedy, J., A New Optimizer Using Particle Swarm Theory, Sixth International Symposium on Micro Machine and Human Science, IEEE, 1995.
- [24] Sedighizadeh, D., Masehian, E., Particle Swarm Optimization Methods, Taxonomy and Applications, International Junal of Computer Theory and Engineering, Vol.1, N°. 5, 2009.
- [25] Melián, B., Moreno, J. Metaheurísticas: una visión global, Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. N°. 19, pp 7-28, 2003.
- [26] García, J., Algoritmos Basados en Cúmulos de Partículas Para la Resolución de Problemas Complejos, 2006.
- [27] Chicano. J., Metaheurísticas e Ingeniería de Software. Tesis Doctoral. Universidad de Malaga, 2007.
- [28] Cruz, C., Estrategias Coordinadas Paralelas Basadas en Soft-Computing para la Solución de Problemas de Optimización. Memoria para optar al grado de Doctor, Universidad de Granada, 2005.
- [29] Millonas, M., Swarms, Phase Transitions, and Collective Intelligence. In C. G. Langton, Ed. Artificial Life III. Addison Wesley, Reading, 1994.
- [30] Beielstein, T., et al., Tunning PSO Parameters Through Sensitivity, 2002.

- [31] Kennedy, J., The Particle Swarm: Social Adaptation of Knowledge. IEEE International Conference on Evolutionary Computation, 1997.
- [32] Xie, Zhang, Yang, Adaptive Particle Swarm Optimization on Individual Level, Int. Conf. On Signal Processing (ICSP), pp: 1215-1218, 2002.
- [33] Peram, T., et al., Fitness-distance-ratio Particle Swarm Optimization, In Proc. IEEE Swarm Intelligence Symposium, pp 174-181, 2003.
- [34] Abraham, A., et al., Swarm Intelligence: Foundations, Perspectives and Applications, Studies in Computational Intelligence (SCI), Springer-Verlag, 2006.
- [35] Molina, J., Modelo y simulación para pronosticar parámetros operacionales de la red nacional de gasoductos de Venezuela usando Máquinas de Soporte Vectorial, Universidad de Los Andes Mérida, Venezuela, 2009.
- [36] Ping-Feng, P., Chih-Sheng, L., Using support vector machines to forecast the production values of the machinery industry in Taiwan, The International Journal of Advanced Manufacturing Technology , Volume 27, Numbers 1-2, 205-210, 2005.
- [37] HUO Bin, BAI Miao-qing, Forecast for the Rural Economy by Using Support Vector Machine, Journal of Shanxi University(Natural Science Edition), 2009.
- [38] Cortés O., et al., Aplicación de la Optimización por Enjambre de Partículas a un Problema Inverso de Conducción de Calor, Programación Matemática y Software Vol.2. Num. 1. 2010.
- [39] Díaz, A., et al., Diseño de Filtros Corrugados mediante la Técnica de Enjambre de Partículas, Journal Telecoforum, 2008.
- [40] Goyenola, G., Guía Para la Utilización de las Valijas Viajeras. Red de Monitoreo Ambiental Participativo de Sistemas Acuáticos (RED MAPSA), Versión 1.0, 2007.
- [41] Evans, J., Magnum, L., Buoy System Handbook. Disponible vía web en http://gyre.umeoce.maine.edu/data/gomoos/buoy/doc/buoy_system_doc/buoy_system/book1.html. Revisada por última vez el 18 de marzo de 2012.
- [42] Ramírez Espinoza, M., Universidad Católica de Temuco, Facultad de Ingeniería., Departamento de Cs. Matemáticas y Físicas, Grupo Física de la Atmósfera, Estación Meteorológica. Disponible vía web en http://www.uctemuco.cl/meteorologia/est_met.html. Revisada por última vez el 18 de marzo de 2012.
- [43] Dirección General de Aeronáutica Civil de Chile, Meteorología Aeronáutica Para Pilotos y Personal de Operaciones, Escuela Técnica Aeronáutica, 1988.
- [44] Reardon, J., Troxler, S., ¿Por qué la clorofila es saludable?, North Carolina Department of Agriculture and Consumer Services, Food and Drug Protection Division.

- [45] Rivera, C., et al., Comparación de la Estimación de la Clorofila-a Mediante los Métodos Espectrofotométrico y Fluorométrico, Acta Biológica Colombiana, Vol. 10 N° 2, 2005.
- [46] Krohling, R., dos Santos Coelho, L., PSO-E: Particle Swarm with Exponencial Distribution, 0-78039487-9, 2006.
- [47] Nickabadi, A., et al., A novel particle swarm optimization algorithm with adaptive inertia weight. Applied Soft Computing 11 (2011) 3658-3670, 2011.

Anexos

A: Estaciones de Monitoreo

En la tabla A.1 se identifican las estaciones de monitoreo, desde donde se extraen las muestras utilizadas para realizar la investigación en cuestión. La tabla detalla el código de identificación de cada estación, el área en que se ubican dentro de la región de Magallanes, el nombre de la estación, el tipo de marisco transvector analizado y sus respectivas coordenadas.

Tabla A.1 Estaciones de Monitoreo. Sector Última Esperanza. Región de Magallanes

ID Estación	Área	Estación	Transvector	Latitud (S)	Longitud (W)
M13	Norte	Estero Peel	Cholga	-50°51'05"	-74°05'55"
M14	Norte	Puerto Buneo	Cholga	-50°59'32"	-74°13'23"
M15	Norte	Isla Vancouver	Cholga	-51°21'33"	-74°05'25"
M16	Norte	Isla Piazzzi	Cholga	-51°39'43"	-73°58'14"
M17	Norte	Caleta Williams	Cholga	-51°58'41"	-73°41'30"
M18	Norte	Bahía Ensenada	Cholga	-52°05'57"	-73°43'51"
M19	Norte	Bahía Isthmus	Cholga	-52°10'07"	-73°36'31"
M20	Norte	Isla Larga	Cholga	-52°11'39"	-73°36'55"
M21	Norte	Puerto Fontaine	Cholga	-52°04'28"	-73°28'09"