

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

**VECTOR DE SOPORTE AUTOREGRESIVO PARA
PRONOSTICAR CAPTURAS DE ANCHOVETA
EN LA ZONA CENTRO - SUR DE CHILE**

JUAN IGNACIO OSORIO OPAZO

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

MAYO, 2012

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

**VECTOR DE SOPORTE AUTOREGRESIVO PARA
PRONOSTICAR CAPTURAS DE ANCHOVETA
EN LA ZONA CENTRO - SUR DE CHILE**

JUAN IGNACIO OSORIO OPAZO

Profesor Guía: **Nibaldo Rodríguez Agurto**

Profesor Co-referente: **Wenceslao Palma Muñoz**

Carrera: **Ingeniería Civil Informática**

MAYO, 2012

Dedicatoria

Dedicada a mis padres por su sacrificio, apoyo, confianza y amor, siendo fundamentales a la hora de alcanzar mis objetivos como persona y estudiante.

Agradecimientos

Al finalizar esta memoria, es inevitable que aparezcan en mis pensamientos personas que han sido de vital importancia en mi vida y en todo este proceso. En primer lugar, quiero agradecer a Dios por acompañarme todos los días y estar presente en cada momento. También agradezco a mis padres y familiares, los cuales siempre brindaron una palabra de aliento en los momentos más complejos, quienes a pesar de la distancia que nos separó durante estos años, estuvieron comprometidos con mi persona, dedicando tiempo y esfuerzo para que se pudiera cumplir este desafío y de manera especial y sincera, a mi polola y a cada una de las personas que me brindaron su amistad durante mi vida Universitaria, con los cuales he compartido grandes momentos.

Finalmente, quiero agradecer a mi profesor guía, Doctor Nivaldo Rodríguez, quién me brindó su apoyo y confianza para la realización del presente proyecto de investigación, siendo clave a la hora de brindar sus conocimientos para poder desarrollar mis ideas.

Resumen

El sector pesquero presenta una importante contribución en las exportaciones a nivel nacional, siendo una de las actividades más trascendentes dentro de la economía del país. Sin embargo, su relevancia se debe principalmente a la disponibilidad de los recursos pesqueros en el tiempo, siendo éste uno de los principales factores que generan incertidumbre dentro de la actividad y que a su vez, pueden entorpecer y restringir el desarrollo sustentable del sector. Con el objetivo de la conservación de estos recursos, entidades gubernamentales han establecido cuotas anuales de captura para estas especies.

Este proyecto tiene como finalidad servir de apoyo en la fijación de las cuotas anuales de anchoveta para la zona Centro-Sur de Chile, para esto, se ha desarrollado un modelo de regresión que permita pronosticar los desembarques mensuales de este recurso entre la V y la X región, utilizando Máquinas de Soporte Vectorial para Mínimos Cuadrados (LS-SVM) y la Optimización por Enjambre de Partículas (PSO).

En función de los resultados, se ha determinado que el modelo basado en LS-SVM + PSO con factor de inercia decreciente linealmente (PSO-IW) presenta el mejor desempeño, siendo más estable a la hora de realizar el pronóstico en comparación a las demás variantes implementadas, logrando su mejor resultado un 93,91 % de la varianza explicada.

Palabras Clave: Especies Pelágicas, Máquinas de Soporte Vectorial, Regresión, Kernel, Optimización por Enjambre de Partículas.

Abstract

The fishing sector has an important contribution to exports (nationally), being a very important activity for the economy of Chile. However, this importance is due to the availability of fishery resources, becoming one of the main factors causing uncertainty in this activity, in turn, can hinder and restrict sustainable development of the sector. In order to conserve these resources, government agencies have set annual catch quotas for these species around the country.

This project aims to provide support in setting annual quotas for the zone anchovy Center - South Chile, this research has developed a regression model that can predict the monthly landings of this resource between the V and X region, using Support Vector Machines for Least Squares (LS-SVM) and Particle Swarm Optimization (PSO).

According to the results, it has been determined that the model based on LS-SVM + PSO with linearly decreasing inertia factor (PSO-IW) had the best performance, becoming the most stable compared to the other variants implemented, achieving the best resulted in 93.91 % of the explained variance.

Keywords: Pelagic, Support Vector Machines, Regression, Kernel, Particle Swarm Optimization.

Indice General

Lista de Abreviaturas	viii
Indice de Figuras	ix
Indice de Tablas	xi
1. Introducción	1
2. Marco General del Proyecto	3
2.1. Objetivo General	3
2.2. Objetivos Específicos	3
2.3. Organización del Documento	3
3. Máquinas de Soporte Vectorial	4
3.1. Introducción	4
3.2. Definición Máquinas de Soporte Vectorial	5
3.3. Máquinas de Soporte Vectorial para Clasificación	6
3.3.1. Caso linealmente separable	6
3.3.2. Caso linealmente no separable	10
3.3.3. Caso no Lineal	14
3.4. Máquinas de Soporte Vectorial para Regresión	16
3.4.1. Regresión Lineal	17
3.4.1.1. Función de pérdida ϵ -insensitive	17
3.4.2. Regresión No Lineal	19
3.5. Máquinas de Soporte Vectorial de Mínimos Cuadrados para la Regresión	20
3.6. Aplicaciones	22
4. Optimización por Enjambre de Partículas	23

4.1.	Introducción	23
4.2.	Fundamentos de PSO	24
4.2.1.	Movimiento de Partículas	24
4.2.2.	Exploración y Explotación	24
4.3.	Descripción del Algoritmo PSO	25
4.4.	Topologías del Enjambre de Partículas	27
4.5.	Tipos de Algoritmos de PSO	28
4.5.1.	PSO con factor de inercia (PSO-IW)	29
4.5.2.	PSO con búsqueda local adaptativa (LSPSO)	29
4.5.3.	Optimización por Enjambre de Partículas con entorno adaptativo (LAPSO)	30
4.5.4.	Optimización por Enjambre de Partículas con factor de inercia exponencial (EPSO)	32
4.5.5.	Quantum PSO (QPSO)	32
4.5.6.	PSO con adaptación dinámica (DAPSO)	33
4.5.7.	PSO binario	34
4.5.8.	Coficiente de constricción	35
4.5.9.	Optimización por Enjambre de Partículas basada en la relación fitness distancia	35
4.6.	Aplicaciones de PSO	36
5.	Modelo LS-SVM + PSO	37
5.1.	Introducción	37
5.2.	Software y Hardware Utilizado	37
5.3.	Descripción y Tratamiento de los Datos	37
5.4.	Métricas de Evaluación	38
5.5.	Función Fitness	40
5.5.1.	Error Cuadrático Medio	40

5.6. Descripción del Modelo General (LS-SVM + PSO)	40
6. Evaluación de los Modelos de Regresión	43
6.1. Características del Entrenamiento	43
6.2. Modelo LS-SVM con PSO con factor de inercia decreciente linealmente . .	44
6.3. Modelo LS-SVM con PSO con búsqueda local adaptativa	49
6.4. Modelo LS-SVM con PSO con entorno adaptativo	53
7. Comparación de Resultados	58
8. Conclusiones	61
Referencias	63

Lista de Abreviaturas

ANN	Artificial Neuronal Network
CIT	Cuotas Individuales y Transferibles de Pesca
CNC	Computer Numerically Controlled
DAPSO	Particle Swarm Optimizer with Dynamic Adaptation
EPSO	Exponential Particle Swarm Optimization
ERM	Empirical Risk Minimisation
KKT	Condiciones Karush-Kuhn-Tucker, asegura que las soluciones del problema dual representa a la solución del problema primal.
LAPSO	Landscape adaptative PSO
LS - SVM	Least Square Support Vector Machine
MAE	Mean Absolute Error
MASE	Mean Absolute Scaled Error
MSE	Mean Square Error
PIB	Producto Interno Bruto
PSO	Particle Swarm Optimization
PSO-IW	Particle Swarm Optimization with Inertia Weight
QPSO	Quantum Particle Swarm Optimization
RMSE	Root Mean Square Error
SRM	Structural Risk Minimisation
SVM	Support Vector Machine

Indice de Figuras

3.1. Conjuntos de vectores linealmente separados mediante un hiperplano de separación óptimo.	5
3.2. Caso linealmente separable.	7
3.3. Conjuntos de vectores linealmente no separables.	10
3.4. Caso no separable linealmente con variables de holgura.	12
3.5. Mapeo de un conjunto de entradas a un espacio característico de mayor dimensión.	15
3.6. Función de pérdida $\epsilon - insensitive$	16
3.7. Relación entre el modelo de tubo y la función de pérdida de Vapnik.	18
4.1. Ejemplos swarm en la naturaleza.	23
4.2. Entorno social y gráfico en el espacio de soluciones.	28
4.3. Topologías del enjambre.	28
5.1. Proceso de optimización de los parámetros de la LS-SVM con optimización por enjambre de partículas utilizando Kernel RBF - Gaussiano.	42
6.1. Variación del fitness en el entrenamiento del mejor desempeño de la LS-SVM + PSO-IW.	47
6.2. Mejor estimación modelo LS - SVM + PSO-IW con desfase de 10 meses.	48
6.3. Modelo LS-SVM + PSO-IW con desfase de 10 meses.	48
6.4. Variación del fitness en el entrenamiento del mejor desempeño de la LS-SVM + LSPSO	52
6.5. Mejor estimación modelo LS - SVM + LSPSO con desfase de 10 meses.	52
6.6. Modelo LS-SVM + LSPSO con desfase de 10 meses.	53
6.7. Variación del fitness en el entrenamiento del mejor desempeño de la LS-SVM + LAPSO.	56
6.8. Mejor estimación modelo LS - SVM + LAPSO con desfase de 9 meses.	57
6.9. Modelo LS-SVM + LAPSO con desfase de 9 meses.	57
7.1. Comparación de Modelos según el Coeficiente de Determinación.	59

7.2. Tiempo de ejecución por cada variante.	60
---	----

Indice de Tablas

5.1. Tipos de Kernel con sus diferentes parámetros a optimizar.	40
6.1. Configuración base Algoritmo PSO.	44
6.2. Resumen variación de desfases modelo LS-SVM + PSO-IW.	44
6.3. Resumen variación de partículas modelo LS-SVM + IW.	44
6.4. Resumen variación de iteraciones modelo LS-SVM + PSO-IW.	45
6.5. Configuración base Algoritmo PSO-IW.	45
6.6. Resultados LS-SVM + PSO-IW.	46
6.7. Resumen LS-SVM + PSO-IW.	46
6.8. Resumen variación de desfases modelo LS-SVM + LSPSO.	49
6.9. Resumen variación de partículas modelo LS-SVM + LSPSO.	49
6.10. Resumen variación de iteraciones modelo LS-SVM + LSPSO.	49
6.11. Configuración base Algoritmo LSPSO.	50
6.12. Resultados LS-SVM + LSPSO.	51
6.13. Resumen LS-SVM + LSPSO.	51
6.14. Resumen variación de desfases modelo LS-SVM + LAPSO.	53
6.15. Resumen variación de partículas modelo LS-SVM + LAPSO.	53
6.16. Resumen variación de iteraciones modelo LS-SVM + LAPSO.	54
6.17. Configuración base Algoritmo LAPSO.	54
6.18. Resultados LS-SVM + LAPSO.	55
6.19. Resumen LS-SVM + LAPSO.	55
7.1. Mejores resultados por variante.	58

1 Introducción

Chile tiene acceso, junto con Perú, a uno de los cinco “oasis” pesqueros más importantes del mundo, es por ello, que cuenta con una gran dotación de recursos marítimos a lo largo de sus costas. Durante los últimos 30 años, el sector pesquero ha alcanzado una participación dinámica dentro del área exportadora del país. En la década de los sesenta, las exportaciones de este rubro representaban cerca del 6 % del total exportado, cifra que aumenta a mediados de los noventa a un 11,7 %, disminuyendo en el periodo comprendido entre los años 2005 y 2009, donde se registró una participación promedio anual de 6,8 % sobre las exportaciones totales del país, situándose como el tercer sector exportador de mayor relevancia, después de la minería y el sector forestal [Donoso, 2010]. Actualmente este sector, alcanza una participación de un 3 % sobre el Producto Interno Bruto (PIB) y un 2 % en el empleo a nivel nacional [Donoso, 2010].

La constante variación en los niveles de exportación de estos recursos, se debe principalmente a los volúmenes de desembarques totales registrados en el país, donde en 1976 se obtuvo un total de 1,4 millones de toneladas, alcanzando su mayor nivel en 1994, con un total de 7,8 millones de toneladas. Desde entonces, se ha observado una constante disminución en el desembarque de recursos pesqueros, promediando entre los años 2003 y 2009, un total de 4,6 millones de toneladas anuales. Mientras que en el año 2010, el desembarque total estimado fue de 3,9 millones de toneladas [Sectorial, 2010].

Los mayores volúmenes de desembarque de especies pelágicas se registran en la zona Centro-Sur del país, pesquería que se desarrolla entre la V y la X región, siendo una de sus especies más abundantes la anchoveta, recurso que se caracteriza por una explotación conjunta por parte de la flota industrial y artesanal. Esta especie presenta un rápido crecimiento, además posee un ciclo de vida corto y una elevada tasa de mortalidad natural, siendo su distribución espacial, biomasa y reclutamiento fuertemente influenciados por las condiciones ambientales del medio en el que habitan, y que redundan en una alta variabilidad en los desembarques. En el periodo comprendido entre los años 2006 y 2010, los desembarques de anchoveta alcanzaron un promedio anual de 375 mil toneladas, mientras que en el año 2011, las capturas para esta especie fueron de 95 mil toneladas. La importancia de la anchoveta radica en la elaboración de aceite de pescado, conservas, harina de pescado, salados y otros productos [Subpesca, 2011, ASIPES, 2012].

La explotación del recurso anchoveta, se ve afectada por una serie de factores que pueden alterar la biomasa para esta especie en la zona Centro-Sur de Chile, entre ellas se pueden destacar la temperatura superficial del mar, los eventos climáticos El Niño y La Niña, así como también la concentración de su alimento (plancton), ya que cuando éste escasea afecta a la composición física de la anchoveta y finalmente, la corriente de Humboldt, que hace más frías las aguas marinas. Junto con lo anterior, la sobreexplotación de los recursos pesqueros ha significado un impacto importante en los niveles de biomasa de las principales especies marinas del país, llegando en algunos casos, a niveles que no permiten un desarrollo sustentable de la actividad.

A consecuencia de lo expuesto anteriormente, las disminuciones en las capturas totales de anchoveta afecta directamente a la producción y exportación de la especie, por lo que obtener una cuota anual de desembarques que se ajuste a la realidad del sector pesquero es gravitante para el futuro de la actividad. Para ello, es que la Subsecretaría de Pesca implementó en el año 2003, a través de una ley transitoria, un sistema de Cuotas Individuales y Transferibles de Pesca (CIT) en las principales pesquerías pelágicas y demersales del país, como una herramienta orientada a la conservación de los recursos, así como también, a la sustentabilidad de la actividad, por lo tanto, su determinación es un elemento relevante para el sector, con el objetivo de evitar el colapso biológico y económico de las unidades pesqueras [Subpesca, 2010].

Debido a esto, el presente proyecto de título tiene como objetivo principal modelar las capturas mensuales de anchoveta, con el fin de elaborar un modelo predictivo que permita, a través de datos históricos, apoyar a las entidades gubernamentales en la fijación de las cuotas anuales de desembarques para la especie pelágica de anchoveta en la zona Centro-Sur del país y de esta manera, permitir la disponibilidad de la especie en el futuro. Para ello, se utilizarán las Máquinas de Soporte Vectorial (SVM, del inglés Support Vector Machine), las cuales corresponden a un conjunto de técnicas de aprendizaje supervisado [Vapnik, 1995, Gunn, 1998].

Las SVM en un principio fueron utilizadas para resolver diversos problemas de clasificación, pero con el pasar de los años, se ha determinado el potencial que ellas tienen en problemas de regresión, donde han obtenido excelentes resultados a través de la introducción de una función de pérdida alternativa. Siendo las Máquinas de Soporte Vectorial de Mínimos Cuadrados (LS-SVM, del inglés Least Squares Support Vector Machines) una modificación de las SVM, que tiene como objetivo mejorar el rendimiento en el análisis predictivo de datos.

Para el correcto funcionamiento de las SVM, se deben ajustar algunos parámetros internos de ellas, esto se llevará a cabo por medio de la utilización del algoritmo de Optimización por Enjambre de Partículas (PSO, del inglés Particle Swarm Optimization) y algunas de sus variantes, con el fin de abarcar el máximo espacio de soluciones y de esta manera, encontrar el modelo que mejor represente el comportamiento de la anchoveta [Sedighzadeh and Masehian, 2009, Chun-Lung et al., 2011, Tang and Zhao, 2010].

2 Marco General del Proyecto

2.1 Objetivo General

Desarrollar y evaluar un modelo de regresión en base a Máquinas de Soporte Vectorial y Optimización por Enjambre de Partículas para predecir las capturas mensuales de anchoveta en la zona Centro-Sur de Chile.

2.2 Objetivos Específicos

- Explicar la funcionalidad de las Máquinas de Soporte Vectorial y el Algoritmo de Optimización por Enjambre de Partículas.
- Diseñar la estructura y estimar los parámetros de la SVM usando los mínimos cuadrados lineales y PSO.
- Implementar y evaluar el rendimiento predictivo de los modelos propuestos.

2.3 Organización del Documento

El presente documento está estructurado de la siguiente manera:

En el capítulo 3, se presentan los conceptos básicos de las SVM, se describe formalmente las SVM para problemas de clasificación, presentando los casos a los que se puede estar enfrentado: caso linealmente separable, caso linealmente no separable y caso no lineal. Además, se describen las SVM para regresión, donde se presentan las SVR y Máquinas de Soporte Vectorial de Mínimos Cuadrados (LS - SVM, del inglés Least Square Support Vector Machine).

En el capítulo 4, se dan a conocer los aspectos generales de la Optimización por Enjambres de Partículas, donde se describen los fundamentos básicos del algoritmo, luego se explica el funcionamiento de éste. A continuación, se detallan las topologías para el enjambre, también se explican algunas variaciones existentes del algoritmo y se exhiben diversos problemas donde PSO ha sido aplicado con éxito.

El capítulo 5, explica el modelo general empleado para dar solución a la problemática en cuestión, además se darán a conocer las diversas métricas utilizadas para medir el desempeño de los modelos desarrollados.

El capítulo 6 presenta los modelos implementados en base a 3 variantes del algoritmo PSO, así como también los resultados obtenidos a partir de ellas, siendo éstos analizados y contrastados en el capítulo 7. Finalmente, el capítulo 8 presenta las conclusiones generales del trabajo realizado.

3 Máquinas de Soporte Vectorial

3.1 Introducción

En la actualidad, existe una gran variedad de problemas en diversas áreas de la ingeniería como también en las demás ciencias aplicadas (Inteligencia artificial, Multimedia, Bioinformática, entre otros), los cuales pueden asociarse a problemas de clasificación y regresión. Varios de estos problemas son muy complejos, demandando una gran cantidad de esfuerzo, tanto científico como tecnológico.

A lo largo de la historia, se pueden encontrar múltiples técnicas que pueden ser aplicadas en diferentes disciplinas y campos de investigación, con el objetivo de obtener resultados útiles para los diversos problemas que se presentan en la vida real. Entre las técnicas más utilizadas destacan las redes neuronales artificiales (ANN, del inglés Artificial Neural Network), los árboles de decisión y la programación genética, las cuales han obtenido excelentes resultados al momento de ser aplicadas. A pesar de la existencia de estas técnicas, las Máquinas de Soporte Vectorial han emergido como un paradigma que permite resolver problemas de clasificación y de regresión en dominios complejos, ruidosos y con pocos datos [Khemchandani et al., 2009].

Las SVM son una nueva técnica, parecida a una red neuronal, que han tomado gran importancia en los últimos años, siendo originalmente diseñadas para la solución de problemas no lineales de clasificación [Li et al., 2008, Rengifo and L., 2002], sin embargo, con el pasar de los años y con las investigaciones realizadas sobre ellas, se han descubierto las grandes capacidades de generalización que éstas tienen, siendo introducidas como una poderosa herramienta para resolver problemas de regresión y predicción de series temporales. Esto se debe a que su formulación encarna el principio de minimización del riesgo estructural (SRM, del inglés Structural Risk Minimisation), que ha demostrado ser superior al tradicional principio de minimización del riesgo empírico (ERM, del inglés Empirical Risk Minimisation), empleado por las convencionales redes neuronales artificiales. SRM minimiza un límite superior al riesgo esperado, en cambio ERM minimiza el error sobre los datos de entrenamiento. Por lo tanto, SVM posee una gran habilidad para generalizar, que es la meta en el aprendizaje estadístico, siendo sumamente importante para aplicaciones del mundo real con limitada cantidad de datos de entrenamiento [Gunn, 1998].

En el presente capítulo, se pretende contextualizar a las Máquinas de Soporte Vectorial. Como punto de partida, se englobará de manera conceptual las SVM, para posteriormente, describir el proceso que se realiza para la clasificación, identificando los casos que se pueden presentar. Luego, se describen las SVM para enfrentar el problema de regresión, presentando las SVR que corresponden a una extensión de las SVM, también se dan a conocer las LS-SVM que corresponden a una reformulación de las Máquinas de Soporte Vectorial de Vapnik. Finalmente, se describirán algunos problemas en los cuales se han aplicado exitosamente las SVM.

3.2 Definición Máquinas de Soporte Vectorial

Las Máquinas de Soporte Vectorial, corresponden a un conjunto de técnicas de aprendizaje supervisado, y que actualmente son utilizadas para resolver tanto problemas de clasificación como de regresión. La parte conceptual de las SVM fue desarrollada por Vapnik y sus colaboradores a principios de los años 90, y se centra en lo que se conoce como Teoría del Aprendizaje Estadístico [Vapnik, 1995].

Las SVM basan su éxito en tres ventajas fundamentales en comparación a otras técnicas: la primera consiste en que poseen una fuerte fundamentación matemática, la segunda de ellas, es que se basan en la minimización del riesgo estructural, es decir, minimizar la posibilidad de error en la clasificación de nuevos datos y por último, es que disponen de potentes herramientas y algoritmos para encontrar la solución de manera rápida y eficiente.

Formalmente, se puede explicar que las SVM se basan en encontrar un hiperplano de separación que divida el espacio de entradas en dos regiones o clases distintas, tal como se muestra en la figura 3.1. En caso que no sea posible separar los datos en el espacio de entrada con un hiperplano lineal, se deben trasladar, mediante una ecuación no lineal, los vectores de entrada a un nuevo espacio de dimensión mayor, es decir, si los datos de entrada se encuentran en \mathbb{R}^2 , son mapeados por la SVM a \mathbb{R}^3 . En este nuevo espacio se construirá una superficie de decisión lineal, buscando el hiperplano deseado [Gunn, 1998].

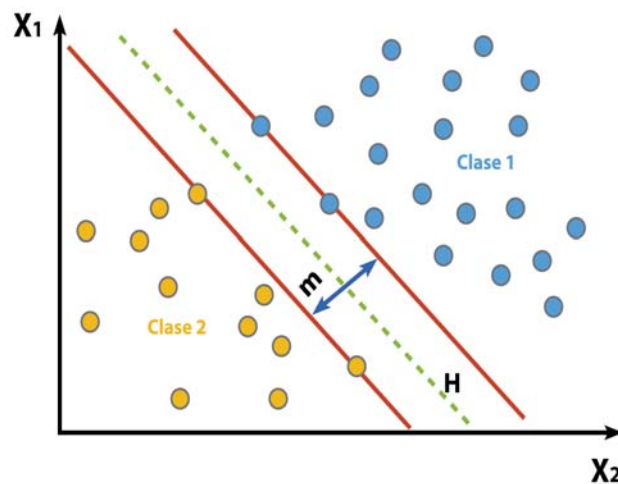


Figura 3.1 – Conjuntos de vectores linealmente separados mediante un hiperplano de separación óptimo.

El proceso de búsqueda de un hiperplano óptimo de separación, tiene como fundamento, maximizar el margen m de separación entre cualquier punto de cualquiera de las clases y el hiperplano separador [Maldonado, 2009], además debe acertar en gran medida en la asignación de los objetos a las clases que realmente pertenecen, permitiendo así un menor error en la clasificación de los datos de entrada.

Maximizar el margen m es un problema de programación cuadrática y puede ser resuelto modificando el modelo primal por intermedio de la introducción de los multiplicadores de Lagrange, quedando de esta manera un problema dual de optimización.

3.3 Máquinas de Soporte Vectorial para Clasificación

Considere que se tiene un conjunto de datos de entrada, donde a cada uno de ellos se le han definido n variables o atributos a estudiar, siendo representados por un vector característico de dimensión $n + 1$, donde las primeras n coordenadas corresponden a las variables en estudio y la última corresponde a la clase a la cual pertenece el dato.

Los vectores pertenecientes al conjunto se denotan como un par $(x, y) \in \mathfrak{R}$, donde $X = (x_1, x_2, \dots, x_{i-1}, x_i)$ e $Y = \{-1, +1\}$, siendo x_i la variable en estudio que pertenece a una determinada clase, y por su parte, Y es una etiqueta que simplemente indica a que clase pertenece el x_i asociado. La idea de las SVM, es separar el conjunto de vectores en dos clases mediante un hiperplano de separación, donde se pueden considerar, de forma general, 3 casos: cuando las clases son linealmente separables, cuando las clases son linealmente no separables y cuando no existe un hiperplano lineal que separe al conjunto de datos, caso para el cual se introducen las funciones Kernels.

3.3.1 Caso linealmente separable

Se dice que la muestra en estudio es linealmente separable cuando existe un hiperplano en \mathfrak{R}^n que deje a todos los vectores de la muestra X asociados a una clase de un lado del hiperplano y a los de la otra clase al otro lado. El objetivo es encontrar un hiperplano que pueda separar la muestra, con la condición adicional que la distancia entre el vector x_i (de cada clase) más cercano al hiperplano sea máxima.

Si representamos el hiperplano en su forma canónica (w, b) , donde w es el vector normal al hiperplano, b es un factor de desplazamiento desde el origen hasta el hiperplano y la distancia es obtenida por $\frac{|b|}{\|w\|}$, la ecuación del hiperplano óptimo quedaría de la siguiente manera [Gunn, 1998]:

$$\langle w, x \rangle + b = 0, w \in \mathfrak{R}^n, b \in \mathfrak{R} \quad (3.3.1)$$

Como se puede apreciar en la figura 3.2, existen dos hiperplanos paralelos al hiperplano separador, denotados por $H1$ y $H2$, los cuales son las fronteras de cada clase y que son generados por los puntos (vectores) que se encuentran más cercanos al hiperplano óptimo, donde estos vectores que se encuentran sobre $H1$ y $H2$ son denominados vectores de soporte. Dichos hiperplanos canónicos están definidos por las siguientes ecuaciones:

$$\langle w, x \rangle + b \geq +1; \text{ para la clase positiva} \quad (3.3.2)$$

$$\langle w, x \rangle + b \leq -1; \text{ para la clase negativa} \quad (3.3.3)$$

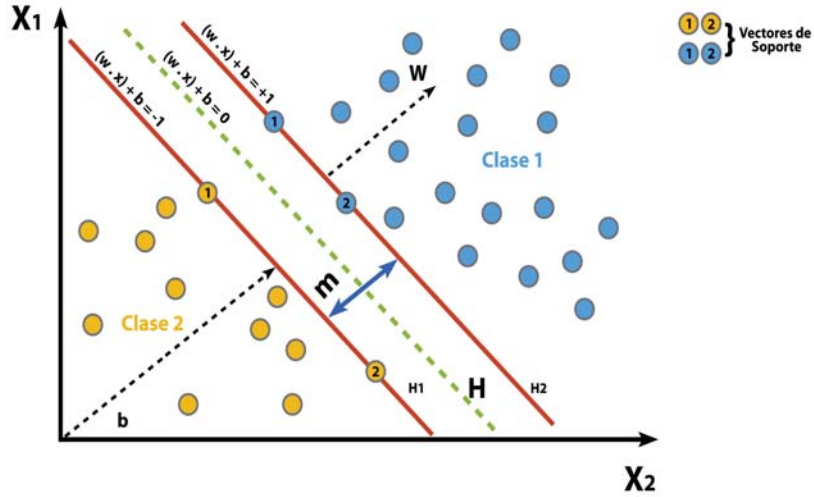


Figura 3.2 – Caso linealmente separable.

Como se ha mencionado en los párrafos anteriores, se desea maximizar la mínima distancia entre los vectores de cada clase más cercanos al hiperplano. El margen de separación del hiperplano definido por el par (w, b) , y denotado por γ_g , puede expresarse como:

$$\gamma_g = \min_+(d(w, b; x_+)) + \min_-(d(w, b; x_-)) \quad (3.3.4)$$

Entendiéndose por x_+ una observación cualquiera que esté en el hiperplano canónico de la clase positiva y x_- una observación cualquiera que esté en el hiperplano canónico de la clase negativa. Recordando la ecuación de la distancia de un punto hacia una recta, o hacia un plano, tenemos que la ecuación 3.3.4 puede ser expresada de la siguiente manera:

$$\gamma_g = \min_+\left(\frac{|\langle w, x_+ \rangle + b|}{\|w\|}\right) + \min_-\left(\frac{|\langle w, x_- \rangle + b|}{\|w\|}\right) \quad (3.3.5)$$

$$\gamma_g = \frac{|+1|}{\|w\|} + \frac{|-1|}{\|w\|} \quad (3.3.6)$$

$$\gamma_g = \frac{|2|}{\|w\|} \quad (3.3.7)$$

A partir de la ecuación 3.3.7, se tiene que la maximización del margen geométrico es equivalente a minimizar la norma w . De esta manera se obtiene el siguiente problema primal de optimización del margen máximo:

$$\min_{(w,b)} \frac{\|w\|^2}{2} \quad (3.3.8)$$

Sujeto a

$$y_i(\langle w, x_i \rangle + b) \geq 1; \text{ para } i = 1, 2, 3, \dots, n$$

Este es un problema de optimización de tipo cuadrático convexo y tiene una única solución global. Esta solución puede darse a partir del uso de la teoría de multiplicadores de Lagrange en n variables. Para lo cual se define el siguiente Lagrangiano:

$$L(w, b, \alpha) = \frac{\|w\|^2}{2} - \sum_{i=1}^m \alpha_i y_i (\langle w, x_i \rangle + b) + \sum_{i=1}^m \alpha_i \quad (3.3.9)$$

donde α_i son los multiplicadores de Lagrange. El Lagrangiano tiene que ser minimizado con respecto a w , b y maximizado con respecto a α_i , sujeto a la condición 3.3.13. Por lo tanto, se pueden plantear las siguientes condiciones de optimalidad (condiciones KKT-Karush-Kuhn-Tucker):

$$\frac{\partial L}{\partial w} = 0 \quad (3.3.10)$$

$$\frac{\partial L}{\partial b} = 0 \quad (3.3.11)$$

$$y_i(\langle w, x_i \rangle + b) - 1 \geq 0 \quad (3.3.12)$$

$$\alpha_i \geq 0 \quad (3.3.13)$$

$$\alpha_i y_i (\langle w, x_i \rangle + b) - 1 = 0 \quad (3.3.14)$$

Estas condiciones se pueden interpretar de la siguiente manera:

- Las dos primeras condiciones aseguran que la solución constituye el único punto silla de la ecuación 3.3.9.

- La tercera y cuarta condición, recogen las restricciones planteadas en el problema primal y en el problema de optimización Lagrangiano respectivamente.
- La última condición es llamada condición de complementariedad y garantiza que la solución coincide con la solución del problema primal.

Debido a las cualidades de la función objetivo (convexa) y de las restricciones (lineales) del problema de optimización primal, se puede construir el problema de optimización dual. Para ello, se deben aplicar las dos primeras condiciones de KKT, a partir de las cuales se obtendrán las siguientes ecuaciones:

$$w = \sum_{i=1}^m \alpha_i y_i x_i \quad (3.3.15)$$

$$\sum_{i=1}^m \alpha_i y_i = 0 \quad (3.3.16)$$

Al reemplazar las expresiones 3.3.15 y 3.3.16 se puede llevar el Lagrangiano a la forma:

$$L_D = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j < x_i, x_j > \quad (3.3.17)$$

Como se puede apreciar, la ecuación 3.3.17 no depende de w ni de b . Por lo tanto, se ha obtenido el siguiente problema de optimización dual asociado al clasificador de margen máximo:

$$\text{máx } L_D = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j < x_i, x_j > \quad (3.3.18)$$

Sujeto a

$$\begin{cases} \sum_{i=1}^m \alpha_i y_i = 0 \\ \alpha_i \geq 0; \text{ para } i = 1, 2, 3, \dots, n \end{cases}$$

El problema de optimización dual, a partir de las condiciones de KKT, se puede interpretar de la siguiente manera:

- Aquellos puntos para los cuales $\alpha_i \geq 0$, son llamados vectores de soportes y se encuentran sobre su correspondiente hiperplano ($H1$ ó $H2$).

- En cambio, todos los puntos de entrenamiento con $\alpha_i = 0$ no se encuentran sobre su correspondiente hiperplano.

Encontrar la solución de la ecuación 3.3.18 usualmente requerirá la aplicación de técnicas de optimización de programación cuadrática y métodos numéricos [Rengifo and L., 2002]. La resolución, se reduce a obtener los valores óptimos para los multiplicadores α_i . Una vez que se encuentran dichos valores, el hiperplano de separación óptimo estará dado por:

$$w^* = \sum_{i=1}^m \alpha_i^* y_i x_i \quad (3.3.19)$$

$$b^* = -\frac{1}{2} w^* [x_r + x_s] \quad (3.3.20)$$

donde x_r y x_s son los vectores de soporte correspondientes a cada clase. Por lo cual, la ecuación del hiperplano óptimo es la siguiente:

$$f(x) = \text{sgn}(\langle w^*, x \rangle + b) \quad (3.3.21)$$

3.3.2 Caso linealmente no separable

En muchos problemas de la vida real no es habitual trabajar con conjuntos de datos que sean linealmente separables. En estos casos, es común encontrarse con vectores de una clase dentro de la región a la cual pertenecen los miembros de la otra clase y por tal motivo, no pueden ser separados en sus clases correspondientes por medio de un hiperplano que cumpla con la restricción 3.3.12, tal como se muestra en la figura 3.3. Estos casos son conocidos como linealmente no separables.

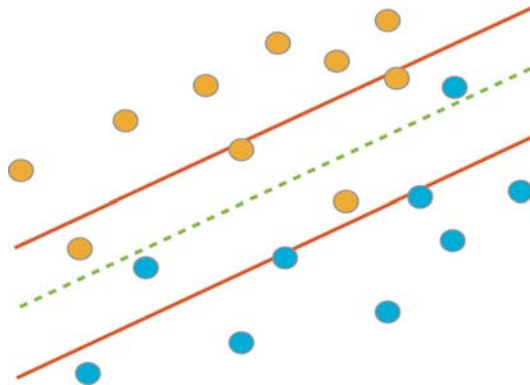


Figura 3.3 – Conjuntos de vectores linealmente no separables.

Sin embargo, no es difícil ampliar las ideas generales del caso linealmente separable al caso linealmente no separable. Esto se puede hacer incorporando valores de holgura no

negativas, cuyo propósito es permitir que algunos puntos de la muestra de aprendizaje no se encuentren en su clase correspondiente. Por lo tanto, las restricciones se pueden reformular de la siguiente manera:

$$(\langle w, x_i \rangle + b) \geq 1 - \xi_i, \text{ para } Y_i = +1 \quad (3.3.22)$$

$$(\langle w, x_i \rangle + b) \leq -1 + \xi_i, \text{ para } Y_i = -1 \quad (3.3.23)$$

$$\xi_i \geq 0, \text{ para } i = 1, 2, \dots, n \quad (3.3.24)$$

Como generalmente se cometen errores en el caso no separable, es necesario incorporar a la función objetivo un costo extra que penalice los errores, ver figura 3.4. Por lo tanto la ecuación 3.3.8 debe ser planteada como el siguiente problema de minimización [Gunn, 1998]:

$$\min_{(w,b,\xi)} \frac{\|w\|^2}{2} + C \left(\sum_{i=1}^m \xi_i \right)^k \quad (3.3.25)$$

La constante C que aparece en la función objetivo es el llamado parámetro de coste y permite controlar el *trade-off* entre el tamaño del margen y el número de elementos que no se encuentran en el lado correcto de su hiperplano de soporte. Si se considera un valor de C muy alto, se está indicando que el coste de cometer errores es muy elevado en comparación con $\|w\|^2$, y por el contrario si C es muy bajo, se está indicando que el coste de cometer errores es bajo, lo que permitirá que las variables de holgura tomen valores más elevados, asignándole un mayor peso a $\|w\|^2$. Los valores para las variables de holgura se pueden interpretar de la siguiente manera:

- Si $\xi_i = 0$ entonces el elemento está en el lado correcto de su hiperplano de soporte.
- Si $0 < \xi_i \leq 1$ entonces el elemento está en el lado incorrecto de su hiperplano de soporte, pero en el lado correcto de la superficie óptima de decisión.
- Si $\xi_i > 1$ entonces el elemento está en el lado incorrecto de la superficie óptima de decisión y en el lado incorrecto de su hiperplano de soporte.

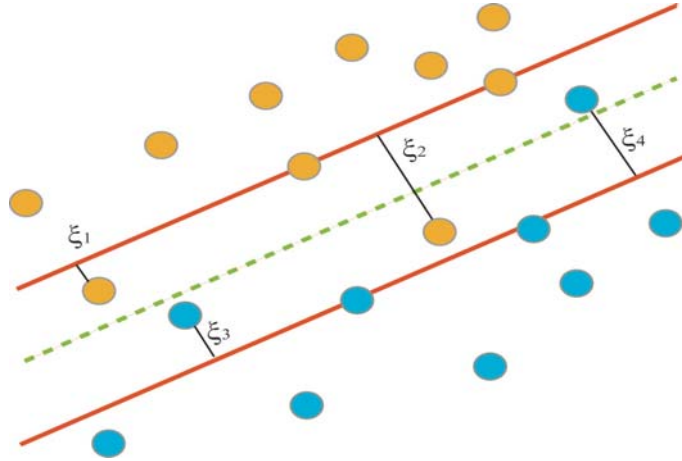


Figura 3.4 – Caso no separable linealmente con variables de holgura.

La variable k de la ecuación 3.3.25 presenta algunas complicaciones a la hora de fijar su valor, por un lado si k es grande se estaría dando mucho más peso a los errores cuantos mayores sean éstos, por otro lado, si se tiene que $k = 2$ ó $k = 1$ se tiene un problema de programación convexa cuadrático. A pesar de esto, la elección de un $k = 1$ tiene la ventaja de que ningún valor de ξ_i ni ninguno de sus correspondientes multiplicadores de Lagrange, aparecen en el problema dual. Por lo tanto, con un valor de k igual a 1 se puede plantear el siguiente problema de optimización, a partir de la ecuación 3.3.25:

$$\min_{(w,b,\epsilon)} \frac{\|w\|^2}{2} + C \left(\sum_{i=1}^m \xi_i \right) \quad (3.3.26)$$

Sujeto a

$$\begin{cases} y_i (\langle w, x_i \rangle + b) - 1 + \xi_i \geq 0; \text{ para } i= 1, 2, \dots, n \\ \xi_i \geq 0; \text{ para } i= 1, 2, \dots, n \end{cases}$$

Una vez formulado el problema de optimización primal para determinar el margen máximo, se puede plantear el problema de optimización dual. Para ello, se debe construir la siguiente función Lagrangiana:

$$L(\alpha, \beta, w, \xi, b) = \frac{\|w\|^2}{2} + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i (y_i (\langle w, x_i \rangle + b) + \xi_i - 1) - \sum_{i=1}^n \beta_i \xi_i \quad (3.3.27)$$

donde w , ξ y b son las variables primales y α , β las variables duales.

Al igual que en el caso separable, al ser la función objetivo convexa y las restricciones lineales, la función Lagrangiana presenta un único punto silla, por lo tanto, el problema

de optimización Lagrangiano debe satisfacer las condiciones KKT, donde se garantiza que se puede emplear el problema de optimización Lagrangiano para resolver el problema de optimización primal.

Al derivar la función Lagrangiana respecto a las variables primales, se obtiene lo siguiente:

$$\frac{\partial L}{\partial w} = w - \sum_{i=0}^n \alpha_i y_i x_i = 0 \Rightarrow w = \sum_{i=0}^n \alpha_i y_i x_i \quad (3.3.28)$$

$$\frac{\partial L}{\partial b} = \sum_{i=0}^n \alpha_i y_i = 0 \quad (3.3.29)$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0 \quad (3.3.30)$$

Sustituyendo los valores de w , ξ y b , la función Lagrangiana se transforma en:

$$\text{máx } L_D(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \langle x_i x_j \rangle \quad (3.3.31)$$

Sujeto a

$$\begin{cases} 0 \leq \alpha_i \leq C; \text{ para } i = 1, 2, 3, \dots, n \\ \sum_{i=1}^m \alpha_i y_i = 0; \text{ para } i = 1, 2, 3, \dots, n \end{cases}$$

De ésta ecuación, se puede deducir que el problema de optimización cuadrática es prácticamente el mismo que en el caso separable, donde se tiene como única variable α_i , cuyos valores estarán acotados entre 0 y C . Dependiendo de su valor, α_i se puede interpretar de las siguientes formas:

- Si $\alpha_i = 0$ el elemento se encuentra correctamente situado en su hiperplano de soporte y por ser igual a 0, no participa en la decisión de la superficie óptima.
- Si $0 < \alpha_i < C$ el elemento se encuentra correctamente situado respecto a su hiperplano de soporte y participa en la determinación de la superficie óptima. Estos elementos son los vectores de soporte.
- Si $\alpha_i = C$ el elemento se encontraría incorrectamente clasificado por su hiperplano de soporte.

Una vez obtenido el valor para la variable α_i , se puede realizar el calculo de w por medio de la siguiente formula.

$$w = \sum_{i=1}^m \alpha_i y_i x_i \quad (3.3.32)$$

siendo m el número de vectores de soporte y x_i los correspondientes vectores de soporte.

3.3.3 Caso no Lineal

La idea básica en diseño de SVM no lineales, es poder realizar una generalización de los métodos anteriores cuando la función de decisión no es lineal en los datos. Para esto, se realiza una transformación de los vectores de entrada x_i pertenecientes a \mathfrak{R}^n dentro de vectores $\phi(x)$ de un espacio de mayor dimensión (espacio de características) en el cual sean linealmente separables, para luego encontrar el hiperplano en ese nuevo espacio y finalmente retornar al espacio original tanto los vectores como el hiperplano.

La proyección de los vectores a un nuevo espacio se debe realizar mediante la siguiente función no lineal:

$$\phi : \mathfrak{R}^n \Rightarrow H \quad (3.3.33)$$

donde en vez de considerar el conjunto de vectores $\{x_1, x_2, \dots, x_n\}$, se deben considerar los vectores transformados $\{\phi(x_1), \phi(x_2), \dots, \phi(x_n)\}$ y si se plantea el problema de optimización original a estos vectores, se tiene que los nuevos vectores forman parte de la solución del problema solo a través del producto interno en H. Así, se puede considerar una función Kernel o núcleo, la cual tiene la facultad de entregar el resultado de un producto interno en un espacio altamente dimensional, sin la necesidad de mapear explícitamente los datos involucrados. De esta manera, la función Kernel queda representada de la siguiente forma:

$$k(x, x') = \phi(x) \cdot \phi(x') = \langle \phi(x), \phi(x') \rangle \quad (3.3.34)$$

para lo cual sólo sería necesario conocer la función Kernel apropiada para el entrenamiento y no se necesitaría tener de forma explícita ϕ .

La figura 3.5, muestra un mapeo de un espacio de entradas de dos dimensiones a un espacio de características de mayor dimensión, donde los vectores de entrada no pueden ser separados por una máquina lineal en su dimensión original.

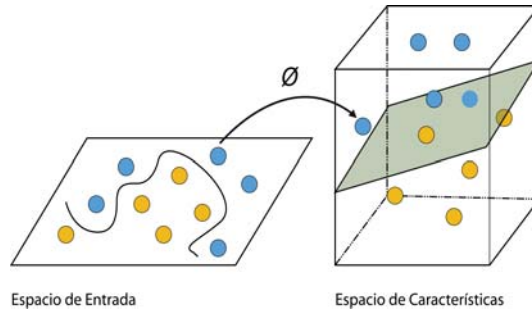


Figura 3.5 – Mapeo de un conjunto de entradas a un espacio característico de mayor dimensión.

Un Kernel adecuado debe cumplir con las siguientes condiciones para definir un espacio de características:

1. Deber ser simétrica:

$$k(x, x') = \langle \phi(x), \phi(x') \rangle = \langle \phi(x'), \phi(x) \rangle = \sum_{i=1}^m \phi_i(x)\phi_i(x') = K(x', x) \quad (3.3.35)$$

2. Cumplir la desigualdad de Cauchy - Schwarz:

$$k(x, x')^2 = \langle \phi(x), \phi(x') \rangle^2 \leq \|\phi(x)\|^2 \|\phi(x')\|^2 = \langle \phi(x), \phi(x) \rangle \langle \phi(x'), \phi(x') \rangle \quad (3.3.36)$$

3. Satisfacer las condiciones de Mercer's:

$$\int \int K(x, x')g(x)g(x')\partial(x)\partial(x') \geq 0 \quad (3.3.37)$$

Si se cumplen todas las condiciones anteriores, existe una representación legítima en el espacio de características. De esta manera, el clasificador lineal queda de la siguiente forma:

$$h(x) = \text{sgn}\left(\sum_{i=1}^m \alpha_i y_i K(x_i, x) + b\right) \quad (3.3.38)$$

Usando diferentes funciones Kernel, el algoritmo de las SVM puede construir una variedad de máquinas de aprendizaje, donde algunas pueden ser similares a arquitecturas clásicas. Dentro de las más utilizadas, se tienen las siguientes:

- Lineal:

$$K(x, y) = \langle x, y \rangle \quad (3.3.39)$$

- Polinomial:

$$K(x, y) = (\gamma \langle x, y \rangle + r)^d, \text{ donde } \gamma > 0, r, d \in \mathfrak{R}. \quad (3.3.40)$$

- Función de Base Radial:

$$K(x, y) = \exp(-\gamma |x - y|^2), \text{ donde } \gamma > 0. \quad (3.3.41)$$

3.4 Máquinas de Soporte Vectorial para Regresión

Recientemente, una versión para regresión de las SVM ha emergido como alternativa y una técnica poderosa para resolver problemas de regresión introduciendo una función de pérdida alternativa. La función de pérdida, debe ser modificada para incluir una medición de distancia. Dentro de las funciones de pérdida más conocidas, se tienen las siguientes:

- Función de pérdida cuadrática.
- Función de pérdida de Laplace.
- Función de pérdida de Huber.
- Función de pérdida $\epsilon - insensitive$.

La función de pérdida de Laplace es más sensible a valores extraños que la función de pérdida cuadrática, la que está basada en el convencional criterio de mínimo error cuadrático. Por otro lado, Huber en 1964 propone una función de pérdida robusta que posee capacidades óptimas cuando la distribución de los datos de entrenamiento es desconocida, sin embargo, no es capaz de disminuir en gran parte los vectores de soporte. Es por ello, que Vapnik propone la función de pérdida de la figura 3.6, la cual es llamada $\epsilon - insensitive$ y que corresponde a una aproximación de la función de pérdida de Huber, permitiendo obtener una menor cantidad de vectores de soporte [Vapnik, 1995].

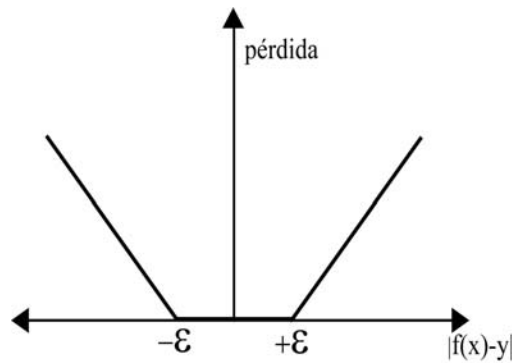


Figura 3.6 – Función de pérdida $\epsilon - insensitive$.

3.4.1 Regresión Lineal

Considerando el problema de aproximación de los datos de entrenamiento, $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$ donde $x \in \mathfrak{R}^n$ y $y \in \mathfrak{R}$, el objetivo de la regresión lineal es obtener la función lineal, definida en la ecuación 3.4.1, que mejor aproxime un conjunto de pares (x_i, y_i) .

$$f(x) = \langle w, x \rangle + b \quad (3.4.1)$$

De esta manera, la función de regresión óptima viene dada por el siguiente problema primal de estimación [Gunn, 1998]:

$$\min_{(w, \xi^*, \xi)} \frac{1}{2} \|w\|^2 + C \left(\sum_{i=1}^m (\xi_i^* + \xi_i) \right) \quad (3.4.2)$$

donde C es un valor predefinido que permite equilibrar la complejidad del modelo obtenido, de acuerdo al número de errores que éste comete. Por otro lado, las variables ξ^* y ξ representan las restricciones superiores e inferiores sobre las salidas del sistema. A continuación, se resolverá el problema primal utilizando la función de pérdida propuesta por Vapnik.

3.4.1.1. Función de pérdida ϵ -insensitive

La función de pérdida ϵ -insensitive viene dada por la siguiente ecuación:

$$|y - f(x)| = \begin{cases} 0 & \text{si } |y - f(x)| \leq \epsilon \\ |y - f(x)| - \epsilon & \text{si } |y - f(x)| > \epsilon \end{cases} \quad (3.4.3)$$

El modelo se basa en la idea que la muestra este encerrada dentro de un ϵ -tubo dimensional, donde se tienen variables de holgura permitiendo que algunos datos puedan salir del tubo. La función ϵ -insensitive acepta errores de hasta un máximo de ϵ sin penalizarlos. Sin embargo, un valor pequeño de ϵ hará que algunos puntos caigan fuera del tubo, haciendo que el error a la hora de realizar el pronóstico crezca linealmente, tal como se muestra en la figura 3.7.

Por otro lado, se tiene que si se disminuye el valor de ϵ , el tubo se va especificando hasta volverse cero, donde el punto central de la predicción será el hiperplano óptimo separador, encontrando de esta manera una función que se adapte mejor a los datos. Además, la pérdida es igual a 0 si la diferencia entre la predicción y el valor observado es menor a ϵ .

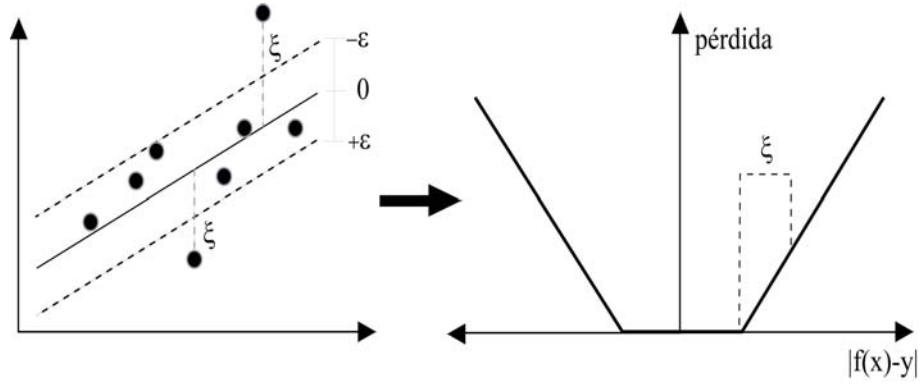


Figura 3.7 – Relación entre el modelo de tubo y la función de pérdida de Vapnik.

El problema de optimización planteado en la ecuación 3.4.2 encuentra el modelo con menor error de acuerdo a la función de pérdida 3.4.3. Para el caso de la función ϵ – *insensitive* el modelo dual a resolver es el siguiente:

$$\text{máx } \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) \langle x_i, x_j \rangle + \sum_{i=1}^m (\alpha_i^* + \alpha_i) y_i - \epsilon \sum_{i=1}^m (\alpha_i^* + \alpha_i) \quad (3.4.4)$$

Sujeto a

$$\begin{cases} 0 \leq \alpha_i, \alpha_i^* < C \\ \sum_{i=1}^m \alpha_i^* = \sum_{i=1}^m \alpha_i \end{cases}$$

Por lo tanto, al resolver el problema dual anterior se obtienen los multiplicadores de Lagrange α_i y α_i^* de las restricciones del problema y la función de regresión viene dada por la ecuación 3.4.1, donde :

$$w = \sum_{i=1}^n (\alpha_i - \alpha_i^*) x_i \quad (3.4.5)$$

$$b = -\frac{1}{2} \langle w, (x_+ + x_-) \rangle \quad (3.4.6)$$

Las condiciones KKT que satisfacen la solución son:

$$\alpha_i \cdot \alpha_i^* = 0, \quad i = 1, \dots, n \quad (3.4.7)$$

Por lo tanto, los vectores de soporte son los puntos donde exactamente uno de los multiplicadores de Lagrange es cero. Cuando $\epsilon = 0$, la función de pérdida y el problema de optimización es simplificado de la siguiente manera:

$$\min \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \beta_i \beta_j \langle x_i, x_j \rangle - \sum_{i=1}^n \beta_i y_i \quad (3.4.8)$$

Sujeto a

$$\begin{cases} -C \leq \beta_i \leq C; \text{ para } i = 1, 2, 3, \dots, n \\ \sum_{i=1}^n \beta_i = 0 \end{cases}$$

Donde la función de regresión está dada por la ecuación 3.4.1, de esta manera:

$$w = \sum_{i=1}^n \beta_i x_i \quad (3.4.9)$$

$$b = -\frac{1}{2} \langle w, (x_r + x_s) \rangle \quad (3.4.10)$$

3.4.2 Regresión No Lineal

De forma similar al problema de clasificación, un modelo no lineal es requerido usualmente para adecuar el modelo de datos. Al igual que en caso de clasificación no lineal, un mapeo no lineal puede ser usado para mapear los datos de entrada a un espacio de características de alta dimensión donde se obtiene una regresión lineal. El Kernel es nuevamente utilizado para hacer frente al curso de la dimensionalidad. La solución de SVR no lineal, utilizando una función de pérdida $\epsilon - insensite$, es la siguiente:

$$\max \sum_{i=1}^n \alpha_i^* (y_i - \epsilon) - \alpha_i (y_i + \epsilon) - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i^* - \alpha_i) (\alpha_j^* - \alpha_j) K(x_i, x_j) \quad (3.4.11)$$

Sujeto a

$$\begin{cases} 0 \leq \alpha_i, \alpha_i^* \leq 0; \text{ para } i = 1, 2, 3, \dots, n \\ \sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0 \end{cases}$$

A partir de la ecuación 3.4.11 y sus respectivas restricciones, se pueden determinar los multiplicadores de Lagrange, α_i y α_i^* , quedando la función de regresión expresada de la siguiente manera:

$$f(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(x_i, x) + b \quad (3.4.12)$$

donde

$$\langle w, x \rangle = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(x_i, x_j) \quad (3.4.13)$$

$$b = -\frac{1}{2} \sum_{i=1}^n (\alpha_i - \alpha_i^*) (K(x_i, x_r) + K(x_i, x_s)) \quad (3.4.14)$$

3.5 Máquinas de Soporte Vectorial de Mínimos Cuadrados para la Regresión

Las SVM de mínimos cuadrados corresponden a una reformulación de las Máquinas de Soporte Vectorial de Vapnik, donde la solución del problema se obtiene a partir de un conjunto de ecuaciones, siendo esto una gran ventaja en comparación con los métodos $\epsilon - insensitive$ y cuadrático de SVM, debido a que a partir de estas ecuaciones se tiene la posibilidad de trabajar con un conjunto más amplio de datos de entrenamiento [Díaz, 2007].

Considerando un modelo en el espacio de características que tiene la siguiente forma:

$$f(x) = \sum_{i=1}^m w_i \phi_i + b \quad (3.5.1)$$

donde $x \in \mathfrak{R}^n$ y $\phi : \mathfrak{R}^n \rightarrow \mathfrak{R}^m$ corresponde a un mapeo del espacio de entrada hacia un espacio de características de mayor dimensión y posiblemente de infinitas dimensiones. Dado un conjunto de entrenamiento $\{x_i, y_i\}_{i=1}^n$, con $y \in \mathfrak{R}$, se puede plantear el siguiente problema de optimización:

$$\min_{\{w, \epsilon\}} \frac{1}{2} \|w\|^2 + \frac{C}{2} \sum_{i=1}^m \epsilon_i^2 \quad (3.5.2)$$

Sujeto a

$$y_i - \langle w, \phi(x_i) \rangle = \epsilon_i$$

Esta formulación es equivalente al problema de SVM Ridge regresión, formulada en el espacio de características. Donde mediante el teorema de Lagrange, se formula el siguiente dual:

$$L(w, b, \epsilon, \alpha) = \frac{1}{2} \|w\|^2 - \frac{C}{2} \sum_{i=1}^m \epsilon_i^2 - \sum_{i=1}^m \alpha_i (\langle w, \phi(x_i) \rangle + b + \epsilon_i - y_i) \quad (3.5.3)$$

donde α_i corresponde a los multiplicadores de Lagrange, mientras que las condiciones de optimalidad vienen dadas por las derivadas parciales de todas las variables, las cuales deben ser igualadas a cero:

$$\begin{cases} \frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^m \alpha_i \phi(x_i) \\ \frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^m \alpha_i = 0 \\ \frac{\partial L}{\partial \epsilon} = 0 \Rightarrow \alpha_i = C \epsilon_i \\ \frac{\partial L}{\partial \alpha} = 0 \Rightarrow y_i = \langle w, \phi(x_i) \rangle + b + \epsilon_i \end{cases} \quad (3.5.4)$$

Una vez eliminadas las variables w y ϵ , se obtiene el siguiente sistema de ecuaciones lineales:

$$\begin{bmatrix} 0 & 1^{-T} \\ \bar{1} & K + \frac{1}{C}I \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \quad (3.5.5)$$

donde $y = [y_1, y_2, \dots, y_n]^T$ y $1 = [1, \dots, 1]^T$. Por lo tanto, a partir del sistema de ecuaciones 3.5.5, el modelo utilizado por LS-SVM para aproximación funcional, se expresa de la siguiente manera:

$$f(x) = K(x, x_i)\alpha + b \quad (3.5.6)$$

donde α , b son la soluciones al sistema de ecuaciones lineales que ha sido planteada en la ecuación 3.5.5.

Es importante mencionar, que este método no requiere la determinación del parámetro ϵ , el cual está relacionado con la función de costo $\epsilon - insensitive$ del método propuesto por Vapnik. Sin embargo, una desventaja de las LS-SVM es que a partir de ellas no se obtiene una solución dispersa tras la optimización del sistema de ecuaciones, es decir, ninguna de las soluciones α_i se desvanece en la solución óptima, por ende todos los datos de

entrenamiento son importantes para el modelo. Esto se puede resolver mediante alguna técnica de poda, eliminando aquellos puntos x_i para los que el valor de α sea pequeño.

3.6 Aplicaciones

A lo largo de su historia, las SVM han sido utilizadas de manera eficiente para resolver una gran variedad de problemas complejos, asociados a diversas áreas. Es por ello, que a medida que pasan los años, las Máquinas de Soporte Vectorial se han transformado en un punto clave para gran parte de los investigadores.

Las Máquinas de Soporte Vectorial han sido aplicadas en numerosos problemas de clasificación, por ejemplo, en [Gutiérrez, 2007] se ha utilizado SVM para dar solución al problema de clasificación de imágenes, siendo de gran importancia cuando se dispone de un gran volumen, ayudando en gran medida en la indexación y catalogación.

En sistemas autónomos y semi-autónomos es importante establecer una metodología que permita el reconocimiento de ciertos parámetros, en [Jabbour et al., 2010] se evaluó el funcionamiento de las SVM en el reconocimiento de firmas estáticas, en cambio, en [Flórez et al., 2007] se aplicó SVM en el reconocimiento de comandos por voz, siendo importante en ambientes con poca visibilidad o cuando el contacto táctil es poco práctico o imposible.

La predicción de series de tiempo es un importante problema de investigación, por ejemplo, [Kuan-Yu and Cheng-Hua, 2007] aplicaron SVM para pronosticar la demanda en el sector de turismo en China, con el objetivo de verificar las habilidades de las SVR en este campo, comparándola con la red neuronal back-propagation y con el modelo ARIMA. También, en [Pai and Hong, 2005] se ha trabajado en la industria de la electricidad, donde se ha utilizado SVM para pronosticar la demanda futura de electricidad, ayudando en gran medida en la estrategia de gestión del sistema de energía. Además, en [Díaz, 2007] se realizó un estudio del comportamiento del precio del oro, a través de las SVM, se elaboró un modelo que permite predecir los precios futuros del oro en Chile.

4 Optimización por Enjambre de Partículas

4.1 Introducción

La Optimización por Enjambres de Partículas es una metaheurística evolutiva que nace en un intento por imitar y mimetizar el comportamiento de los procesos naturales [Sedighizadeh and Masehian, 2009]. PSO es uno de los métodos más utilizados en la inteligencia computacional, fue desarrollada por James Kennedy y Russell Eberhart en 1995, está basada en poblaciones e inspirada en el comportamiento social observado en grupos de individuos, tales como, bandadas de pájaros, enjambres de insectos o cardúmenes de peces, ver figura 4.1. Kennedy y Eberhart pretendían emular la capacidad que estos grupos tienen para separarse, agruparse o encontrar su alimento [Cruz, 2005].



Figura 4.1 – Ejemplos swarm en la naturaleza.

Básicamente, la metaheurística PSO consiste en un algoritmo iterativo basado en una población de individuos denominada enjambre, en la que cada individuo, llamado partícula, se dice que sobrevuela el espacio de decisión en busca de soluciones óptimas. Cada partícula tiene asociada un **valor fitness**, una **posición** y un **vector de velocidad** que dirige su movimiento. En cada iteración, las partículas cambian de posición dentro del espacio de soluciones a una velocidad que dependerá de su **velocidad actual**, de su **memoria** y de la **información global** que comparte con el resto del enjambre.

A modo de ejemplo, suponer que una bandada de aves busca alimentarse en un determinado sector, donde solamente existe una pieza de comida. Los pájaros no saben donde ésta se encuentra, pero sí conocen su distancia a la misma, por lo que la estrategia más eficaz para hallar la comida es seguir al ave que se encuentre más cerca de ella. PSO emula este escenario para resolver problemas de optimización.

Dentro de las principales características de la Optimización por Enjambre de Partículas es que a partir de su utilización, se obtienen muy rápido los resultados, además, tiene pocos parámetros a ajustar y generalmente, trabaja sobre poblaciones pequeñas. Debido a estas cualidades, PSO es muy atractiva para diversas áreas, tanto de aplicación como de investigación, por ejemplo: entrenamientos de redes neuronales, sistemas de control difuso y en la optimización.

En el presente capítulo, se presentan los fundamentos básicos de PSO, como lo es el movimiento de las partículas y el concepto de exploración y explotación sobre los cuales se basa el algoritmo. A continuación, se presenta una descripción del algoritmo básico de PSO, para continuar con los diferentes tipos de topologías existentes para el enjambre de partículas. Posteriormente, se presentarán algunas variantes de PSO, las cuales tienen como objetivo mejorar el rendimiento del algoritmo. Finalmente, se darán a conocer algunas aplicaciones donde la Optimización por Enjambres de Partículas ha sido utilizadas exitosamente.

4.2 Fundamentos de PSO

4.2.1 Movimiento de Partículas

La Optimización por Enjambre de Partículas se basa en lo que se conoce como inteligencia de grupos, donde se deben respetar 5 principios básicos, los que se conocen como: proximidad, calidad, diversidad de respuesta, estabilidad y adaptabilidad.

De acuerdo a la **proximidad**, la población debe ser capaz de realizar cálculos sencillos de espacio y tiempo, lo cual se traduce en PSO, a movimientos en N-dimensiones llevados a cabo durante un intervalo de tiempo que coinciden con movimientos de la población a una determinada velocidad. Los factores de **calidad** en PSO, se consiguen considerando la memoria de la partícula en conjunto con la historia o conocimiento social que comparte con el resto. El tercer principio consiste en la **diversidad de respuesta**, la cual es representada por las tendencias marcadas por la memoria personal de cada partícula y por la historia de la mejor posición visitada por el enjambre.

Considerando el último principio, **la estabilidad y la adaptabilidad** son conceptos totalmente opuestos. El primero de ellos, establece que la población solo cambia su comportamiento como grupo cuando uno de sus miembros, actualiza la mejor posición visitada históricamente por su posición actual. En cambio, el principio de adaptabilidad propone que la población debe ser capaz de modificar su comportamiento y movimiento cuando exista un parámetro que así lo indique, en PSO esto se consigue cuando alguna partícula alcanza una solución global que mejora el resultado, cambiando el rumbo de la población.

4.2.2 Exploración y Explotación

Uno de los principales problemas de los algoritmos evolutivos, incluido PSO, es que en ocasiones se quedan atrapados en óptimos locales, provocando una convergencia prematura del algoritmo, por lo tanto, no se pueden explorar otras regiones del espacio de soluciones.

En PSO, se busca que las partículas del enjambre puedan recorrer gran parte del espacio de búsqueda antes de comprometerse con alguna solución en particular, esto es lo que se conoce como **diversificación**. Por otro lado, cuando las partículas encuentren una zona comprometedora deben ser capaces de mejorar la solución encontrada, lo que es conocido como **convergencia de la población**.

Dentro de los parámetros de PSO, existen dos términos a tener en consideración: **exploración** y **explotación**. El primero de ellos, hace referencia a la búsqueda de la solución del problema de manera extensa dentro del espacio, privilegiando la diversificación del enjambre. El segundo término, consiste en la búsqueda de la solución en un rango dentro del espacio de búsqueda, privilegiando la convergencia de enjambre debido a que las partículas deberán tomar la máxima cantidad de valores posibles. Por lo tanto, los parámetros del algoritmo deben ser ajustados para que en un principio funcione en modo de exploración y luego de varias iteraciones se cambie a modo de explotación.

4.3 Descripción del Algoritmo PSO

El algoritmo PSO es un proceso iterativo y estocástico que opera sobre un enjambre de partículas. Cada partícula p_i perteneciente al enjambre, tiene asociada una posición, la cual representa una posible solución del problema tratado. Generalmente, una partícula está compuesta por tres vectores:

- El vector $x_i = (x_{i1}, x_{i2}, \dots, x_{iN})$ almacena la posición actual de la partícula en el espacio de búsqueda.
- El vector $p_i = (p_{i1}, p_{i2}, \dots, p_{iN})$ tiene asociada la mejor posición obtenida por una partícula durante el proceso.
- El vector $v_i = (v_{i1}, v_{i2}, \dots, v_{iN})$ que indica la velocidad a la cual se moverá la partícula en el espacio de búsqueda.

Además, cada partícula tiene asociado el valor $fitness_x_i$ y $fitness_pBest_i$, donde el primero de ellos indica la calidad de adecuación de la solución actual, mientras que el segundo corresponde al mejor valor de adecuación encontrado por la partícula hasta el momento.

El enjambre se inicializa generando las posiciones y velocidades iniciales para cada una de las partículas de manera aleatoria en un espacio $N - dimensional$. Las velocidades deben estar dentro del intervalo $[-v_{max}, v_{max}]$, siendo v_{max} la velocidad máxima que puede alcanzar una partícula en cada movimiento, esta velocidad guía la búsqueda a través del espacio de soluciones, donde valores altos facilitan una búsqueda global y en caso contrario, facilitan una búsqueda local [Toro et al., 2006].

Una vez que se haya inicializado el enjambre, las partículas deben moverse dentro de un proceso iterativo, el cual va a finalizar cuando se alcance algún criterio de término

previamente fijado. Dentro de este proceso, se calcula el valor $fitness_{x_i}$ de cada partícula y se compara con su valor $fitness_{pBest_i}$. En caso de que su valor actual sea mejor que su $fitness_{pBest_i}$, éste último debe ser modificado.

A continuación, el valor $fitness_{pBest_i}$ de cada partícula debe ser comparado con el valor fitness de la mejor partícula de la vecindad (Global o Local). En caso de ser mejor, el valor $fitness_{gBest_i}$ debe ser actualizado por el valor de $fitness_{pBest_i}$.

Luego, el vector de velocidad asociado a cada partícula i , debe ser actualizado en cada iteración k y para ello, se debe utilizar la velocidad anterior, un componente *cognitivo* y un componente *social*. La ecuación que determina la nueva velocidad es la siguiente [Gang and Zhuping, 2011]:

$$v_{ij}^{k+1} = v_{ij}^k + \varphi_1 \cdot rand_1 \cdot (p_{ij} - x_{ij}^k) + \varphi_2 \cdot rand_2 (p_{gj} - x_{ij}^k) \quad (4.3.1)$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \quad (4.3.2)$$

donde:

- v_{ij}^k es la velocidad de la partícula i en la iteración k .
- φ_1, φ_2 son los ratios de aprendizaje que controlan los componentes cognitivo y social respectivamente.
- $rand_1, rand_2$ números aleatorios entre 0 y 1.
- x_{ij}^k corresponde a la posición actual de la partícula i en la iteración k .
- p_{ij} mejor posición encontrada por la partícula i hasta el momento.
- p_{gj} es la mejor posición obtenida entre todas las partículas durante las k iteraciones.

En la ecuación 4.3.1, el componente *cognitivo* está dado por el factor $\varphi_1 \cdot rand_1 \cdot (p_{ij} - x_{ij}^k)$ y representa la distancia entre la posición actual y la mejor posición conocida para la partícula, es decir, la decisión que tomará influenciada por su propia experiencia. El componente *social* está modelado por $\varphi_2 \cdot rand_2 (p_{gj} - x_{ij}^k)$ y representa la distancia entre la posición actual y la mejor posición del vecindario, es decir, la decisión que tomará la partícula influenciada por el resto del enjambre [Gang and Zhuping, 2011].

Los parámetros φ_1 y φ_2 pueden acelerar la convergencia del algoritmo y aliviarlos de los mínimos locales. Los valores típicos para estos parámetros son $\varphi_1 = \varphi_2 = 2$. Sin embargo, estudios realizados sobre la influencia de los ratios de aprendizaje, establecen que el parámetro cognitivo debe ser más grande que el social [Herrera et al., 2009].

El tamaño de la población es otro de los parámetros que el investigador debe determinar. Una población grande provee una búsqueda completa en el espacio de soluciones,

incrementando así el coste computacional. Según [Herrera et al., 2009] las poblaciones relativamente pequeñas son capaces de resolver una gran variedad de problemas. Generalmente, se recomienda una población de 30 partículas, sin embargo, se han logrado resolver problemas utilizando exitosamente entre 10 y 20 partículas.

A continuación se presenta el pseudocódigo del algoritmo principal de PSO.

Algoritmo 1 Pseudocódigo algoritmo básico PSO.

Inicializar Enjambre de Partículas

Mientras no se alcance la condición de parada **Hacer**

Para $i = 1$ hasta tamaño Enjambre **Hacer**

Evaluar cada partícula

Si $fitness_x_i$ es mejor que $fitness_pBest_i$ **Entonces**

$p_i \leftarrow x_i$

$fitness_pBest_i \leftarrow fitness_x_i$

Fin Si

SI $fitness_pBest_i$ es mejor que $fitness_gBest$ **Entonces**

$p_g \leftarrow p_i$

$fitness_gBest \leftarrow fitness_pBest_i$

Fin Si

Fin Para

Para $i = 1$ hasta tamaño de Enjambre **Hacer**

Para $j = 1$ hasta tamaño de Dimensión **Hacer**

$v_{ij} = v_{ij} + \varphi_1 \cdot rand_1 \cdot (p_{ij} - x_{ij}) + \varphi_2 \cdot rand_2 (p_{gj} - x_{ij})$

$x_{ij} = x_{ij} + v_{ij}$

Fin Para

Fin Para

Fin Mientras

Salida: Devuelve la mejor solución encontrada

4.4 Topologías del Enjambre de Partículas

Uno de los aspectos más importantes y de gran relevancia en PSO tiene relación con la manera en que una partícula interactúa con su vecindario. El desarrollo de una partícula, depende de la topología del enjambre así como también de la versión del algoritmo a utilizar. El entorno de interacción de una partícula con su vecindario es definido por la topología. Cada partícula tiene definido el entorno al que pertenece, éste puede ser de dos tipos:

- **Geográfico:** El entorno de cada partícula está formado por aquellos vecinos que se encuentran más cerca de su posición actual.
- **Social:** El entorno de cada partícula es definido sin importar su posición en el espacio.

Los entornos sociales son los más empleados, ya que a través de ellos, una partícula puede interactuar con más elementos del enjambre y de esta forma, puede adquirir conocimientos de la mejor partícula de la población. Una vez definido el entorno, se debe determinar su tamaño, si éste corresponde a todo el enjambre, entonces el entorno sería geográfico y social a la vez, obteniéndose de esta manera un PSO Global. La figura 4.2 muestra de manera gráfica entornos sociales y geográficos.

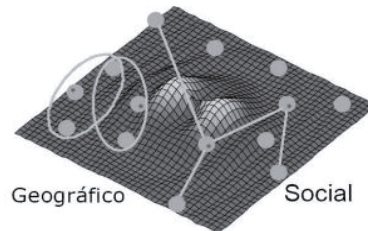


Figura 4.2 – Entorno social y gráfico en el espacio de soluciones.

Dependiendo de la topología que adquiera la población, la transmisión de la información entre individuos puede acelerarse o detenerse, lo cual está íntimamente relacionado con la velocidad de convergencia y con la capacidad del algoritmo para escapar de soluciones locales. Dentro de un entorno social se pueden configurar varios tipos de topologías, siendo el vecindario global el más utilizado, donde cada partícula es vecina de la totalidad del enjambre, favoreciendo la explotación del espacio de soluciones.

El vecindario local corresponde a otro tipo de topología que fue propuesto para resolver problemas de mayor dificultad. En éste, cada partícula es conectada con sus vecinas inmediatas en el enjambre, así, por ejemplo, la partícula p_i es vecina de la partícula p_{i-1} y de p_{i+1} . Esta topología ofrece la ventaja de establecer subcúmulos que realizan búsquedas en diversas regiones del espacio de soluciones y de esta forma favorece la exploración.

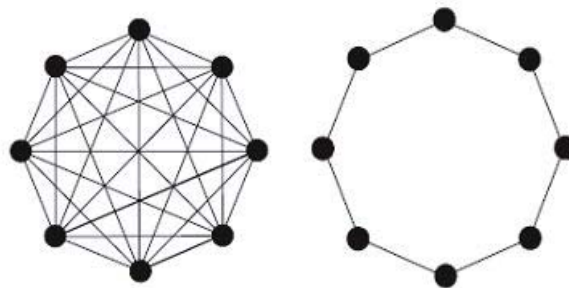


Figura 4.3 – Topologías del enjambre.

4.5 Tipos de Algoritmos de PSO

Con el pasar de los años y tras diversos estudios realizados sobre PSO, se han logrado establecer una serie de modificaciones al algoritmo tradicional. Donde los investigadores,

por medio de estas variantes, tratan de optimizar la calidad de las soluciones encontradas y la velocidad de convergencia del algoritmo, incorporando nuevos parámetros o variantes de los ya existentes. Por lo tanto, dependiendo de los factores de configuración aplicados al algoritmo básico, se pueden encontrar diferentes versiones de PSO, a continuación, se presentan algunas de las más utilizadas.

4.5.1 PSO con factor de inercia (PSO-IW)

En [Chun-Lung et al., 2011] se incorpora un factor de inercia (w) a la ecuación de velocidad del algoritmo PSO convencional, denominado PSO-IW (del inglés Particle Swarm Optimization with Inertia Weight). Este nuevo parámetro, se ha desarrollado con el objetivo de mejorar la exploración y la explotación del espacio de búsqueda. De esta manera, w es muy importante para evitar el entrampamiento en un mínimo local.

Al momento de introducir el factor de inercia, la ecuación 4.3.1 queda expresada de la siguiente manera:

$$v_{ij}^{k+1} = \omega \cdot v_{ij}^k + \varphi_1 \cdot rand_1 \cdot (p_{ij} - x_{ij}^k) + \varphi_2 \cdot rand_2 (p_{gj} - x_{ij}^k) \quad (4.5.1)$$

El factor de inercia w se va reduciendo linealmente durante las iteraciones del algoritmo y se calcula utilizando la siguiente expresión:

$$w = w_{max} - \frac{(w_{max} - w_{min})}{iter_{max}} * k \quad (4.5.2)$$

donde: w_{max} corresponde al valor máximo del factor de inercia, w_{min} es el valor mínimo que puede alcanzar el factor de inercia, $iter_{max}$ representa el número máximo de iteraciones del algoritmo y finalmente, k hace referencia a la iteración actual del algoritmo.

Es importante mencionar, que valores altos en el factor de inercia facilita una exploración global, es decir, es posible encontrar nuevas áreas de búsqueda. En cambio, mientras más pequeño sea w se tiende a facilitar la exploración local. Una selección adecuada de w puede proporcionar un equilibrio entre las capacidades de exploración global y local, por lo tanto, se requieren de menos iteraciones en promedio para encontrar el óptimo [Shi and Eberhart, 1998].

4.5.2 PSO con búsqueda local adaptativa (LPSO)

En el PSO estándar, las partículas son en gran parte influenciadas por su mejor posición alcanzada en su recorrido y por la mejor posición del enjambre. Una vez que la mejor partícula, no presenta cambios en un óptimo local, el resto de las partículas convergerán rápidamente a la posición presentada por la mejor partícula, esto se puede

lograr a través de búsquedas sobre la vecindad de la mejor partícula del enjambre en cada generación.

En [Tang and Zhao, 2010], se propone un nuevo operador de búsqueda local adaptativa, llamado también operador de mutación, el cual consiste en mutar en cada iteración la mejor partícula encontrada hasta ese momento, la cual variará respecto a las ecuaciones presentadas a continuación:

$$gBest_j(t+1) = gBest_j(t) + [\beta_j(t) - \alpha_j(t)] * rand() \quad (4.5.3)$$

$$\alpha_j(t) = \min(x_{ij}(t)); \beta_j(t) = \max(x_{ij}(t)) \quad (4.5.4)$$

$$i = 1, 2, \dots, ps, \quad j = 1, 2, \dots, n$$

donde $gBest_j$ es el vector de la partícula que presenta la mejor solución del enjambre, $\alpha_j(t)$ y $\beta_j(t)$ son los valores mínimo y máximo de la dimensión j en el espacio de búsqueda, mientras que $rand()$ es un número aleatorio entre 0 y 1, ps corresponde al tamaño de la población y t indica en número de iteraciones.

El tamaño de este operador ($\alpha_j(t) - \beta_j(t)$) puede ser considerado como el diámetro del espacio de búsqueda de la población. Al principio de la evolución, el espacio de búsqueda inicial es grande, por ende, el valor del operador es demasiado alto, siendo esto beneficioso para una búsqueda global, acelerando la convergencia del algoritmo. Sin embargo, con el aumento de las iteraciones, la población poco a poco converge a la mejor posición encontrada hasta ese momento. Por lo que el espacio de búsqueda de la población disminuye, así como también, el valor para el operador de búsqueda. Con esta situación, se beneficiará una búsqueda local en el espacio de soluciones, lo que conlleva a explotar de mejor manera la vecindad de la mejor partícula del enjambre.

4.5.3 Optimización por Enjambre de Partículas con entorno adaptativo (LAPSO)

En [Yisu et al., 2008], realizan una hipótesis basada en que la distribución de todo el enjambre puede proporcionar información adicional importante para la actualización de la nueva velocidad. Durante el proceso de búsqueda en PSO, la distribución de las partículas cambia continuamente. Los autores, definen un vector \vec{D} de distribución para describir el espacio de las partículas en un determinado momento:

$$D_j = \frac{\max_{i=1}^n(x_{ij}) - \min_{i=1}^n(x_{ij})}{\text{abs}(\max_{i=1}^n(x_{ij})) + \text{abs}(\min_{i=1}^n(x_{ij}))} \quad ; \quad j = 1, 2, 3, \dots, m \quad (4.5.5)$$

donde m es la dimensión de las partículas y n es el tamaño de la población. La dimensión de \vec{D} es m .

El valor de cada componente del vector D de distribución es entre 0 y 1. Comparando a esta variante con la actualización de velocidad dada por IW-PSO, el reajuste de la velocidad a partir del vector de distribución no afecta solamente a valor número de la velocidad, sino que también a la dirección de las partículas. De esta manera, la actualización de la velocidad viene dada por:

$$v_{ij}^{k+1} = D[v_{ij}^k + \varphi_1 \cdot rand_1 \cdot (p_{ij} - x_{ij}^k) + \varphi_2 \cdot rand_2(p_{gj} - x_{ij}^k)] \quad (4.5.6)$$

A partir de la ecuación anterior, se puede apreciar que si muchas partículas quedan atrapadas en un mínimo local, el vector de distribución será muy pequeño, casi cero, y por lo tanto la posibilidad de ajustar las velocidades se perderá. Esto llevaría a la convergencia prematura del algoritmo. Sin embargo, si la velocidad es demasiado grande puede provocar oscilaciones en la búsqueda y que sea poco probable que el PSO pueda encontrar valores mínimos.

Con el objetivo de evitar la situación mencionada, los autores modifican en vector de distribución en cada iteración del algoritmo de la siguiente manera:

$$D_j = \begin{cases} D_{max}; & Si D_j < D_{min} \\ D_{min}; & Si D_j > D_{max} \\ D_j; & en otro caso \end{cases} \quad (4.5.7)$$

En [Yisu et al., 2008], por medio de los experimentos realizados, definen los valores de D_{max} y D_{min} en 0.95 y 0.4, respectivamente.

En el LAPSO, la velocidad en la siguiente iteración se genera a partir de la distribución de las partículas presentes dentro del vector de distribución, garantizando de esta manera, una mejor propagación de las partículas según su distribución en el espacio de búsqueda. El límite inferior y límite superior para el vector de distribución se proporciona para evitar una convergencia local y una oscilación inestable.

Los autores proponen una pequeña modificación en caso de que determinadas partículas se encuentren fuera del espacio de solución. A lo largo de las iteraciones, algunas partículas abandonarán el área de búsqueda permitida. Para ello, recomiendan una redistribución de las partículas, bajo las siguientes ecuaciones:

$$\begin{aligned} \vec{X}(\vec{X} < X_{min}) &= X_{min} + rand() \\ \vec{X}(\vec{X} > X_{max}) &= X_{max} - rand() \end{aligned} \quad (4.5.8)$$

4.5.4 Optimización por Enjambre de Partículas con factor de inercia exponencial (EPSO)

En [Ghali et al., 2009], se propone una modificación al PSO convencional, donde las partículas tienden a acercarse hacia la mejor posición encontrada por el enjambre. Esta situación ayuda a descubrir soluciones óptimas rápidamente. Sin embargo, también se obtienen partículas que se estancan en óptimos locales y no convergen a un óptimo global. Una vez que se obtiene la mejor posición del enjambre, las partículas son atraídas hasta esa posición en las iteraciones siguientes, hasta que se encuentre otra mejor solución. Por lo tanto, el estancamiento del algoritmo PSO se debe a la velocidad de la mejor solución encontrada [Ghali et al., 2009]. Una mejora a esta situación, está dada por la variante presentada en la sección 4.5.1, donde el factor de inercia disminuye linealmente hasta un valor mínimo.

En [Ghali et al., 2009], los autores plantean el PSO exponencial (EPSO), el cual tiene un gran impacto tanto en la exploración global como local, con el objetivo de que el comportamiento llevado a cabo en la búsqueda sea de forma rápida e inteligente, así evitar que las partículas se estancuen en un óptimo local mediante un factor de inercia que varía exponencialmente, donde el movimiento de las partículas es más rápido y distante entre ellas. Este factor de inercia exponencial viene dado por la siguiente ecuación:

$$w = (w_{max} - w_{min}) * \exp\left(\frac{(iter_{max} - k) - 1}{iter_{max}}\right) + w_{min} \quad (4.5.9)$$

Los autores expresan, a partir de los diversos experimentos realizados, que el factor de inercia disminuya exponencialmente dentro del intervalo [0.9 - 0.4], donde $w_{max} = 0,9$, mientras que $w_{min} = 0,4$. Por otro lado, $iter_{max}$ corresponde al número máximo de iteraciones del algoritmo y k hace referencia al número actual de iteraciones.

4.5.5 Quantum PSO (QPSO)

Según el modelo clásico de PSO, el estado de las partículas está dado por el vector de posición x_i y el vector de velocidad v_i , determinando así la trayectoria de las partículas, las que se mueven siguiendo los principios básicos de la mecánica Newtoniana. Sin embargo, si consideramos la mecánica cuántica, el comportamiento dinámico de la partícula difiere ampliamente del comportamiento tradicional de PSO, donde los valores exactos de velocidades y posiciones no pueden ser determinados simultáneamente.

Por lo tanto, si las partículas pertenecientes al enjambre tienen un comportamiento cuántico, el rendimiento del algoritmo PSO estará muy lejos del PSO clásico.

En el modelo cuántico de PSO, el estado de cada partícula está representado por la función de onda $\psi(x, t)$, en vez de la posición y la velocidad. De acuerdo al significado estadístico de la función de onda, la probabilidad de que una partícula aparezca en

una determinada posición se puede obtener de la función de probabilidad de la densidad $|\psi(x, t)|^2$.

En QPSO, las partículas se mueven de acuerdo a las siguientes funciones [Pant et al., 2008]:

$$x(t + 1) = p + \beta * |mbest - x(t)| * \ln\left(\frac{1}{n}\right); \text{ Si } k \geq 0,5 \quad (4.5.10)$$

$$x(t + 1) = p - \beta * |mbest - x(t)| * \ln\left(\frac{1}{n}\right); \text{ Si } k < 0,5 \quad (4.5.11)$$

donde:

$$p = \frac{(c_1 p_{ij} + c_2 p_{gj})}{c_1 + c_2} \quad (4.5.12)$$

$$mbest = \frac{1}{M} \sum_{i=1}^M p_i = \left(\frac{1}{M} \sum_{i=1}^M p_{i1}, \frac{1}{M} \sum_{i=1}^M p_{i2}, \dots, \frac{1}{M} \sum_{i=1}^M p_{iM} \right) \quad (4.5.13)$$

El *mbest* de una población está definido como la media de la mejor posición de todas las partículas. Por otro lado, se tiene que los parámetros u , k , c_1 y c_2 son números aleatorios dentro del intervalo $[0,1]$. El parámetro β es llamado coeficiente de constricción.

4.5.6 PSO con adaptación dinámica (DAPSO)

En el PSO convencional, las partículas siguen dos valores: el valor actual de la mejor solución global y la mejor solución encontrada hasta el momento por la partícula. Es así, que las velocidades de las partículas convergen rápidamente a cero, haciendo que las partículas se encuentren atrapadas en óptimos locales. Este fenómeno se denomina “similitud” de enjambre de partículas y provoca que se disminuya el área de búsqueda de partículas. En [Yang et al., 2007] proponen que para ampliar la zona de búsqueda se puede utilizar las siguientes alternativas: por un lado aumentar el número de partículas o bien, debilitar la capacidad de las partículas para seguir al valor de la mejor solución del enjambre. Sin embargo, la primera opción implica una mayor complejidad computacional, mientras que la última, provoca una convergencia lenta del algoritmo.

Los autores incorporan a la fórmula convencional de velocidad un factor de inercia w , el cual varía en cada iteración. Sin embargo, si este factor decrece linealmente no se refleja el proceso actual de búsqueda. Es por ello, que en [Yang et al., 2007] proponen que el factor de inercia sea afectado por el estado evolutivo del algoritmo y que sea determinado por el factor de velocidad evolutivo de cada partícula y el grado de agregación del enjambre.

De esta manera, los autores establecen que el factor de velocidad evolutivo esta dado por la siguiente ecuación:

$$h_i^t = \left| \frac{\min(F(pbest_i^{t-1}), F(pbest_i^t))}{\max(F(pbest_i^{t-1}), F(pbest_i^t))} \right| \quad (4.5.14)$$

donde $F(pbest_i^t)$ es el valor fitness de $pbest_i^t$. Bajo este supuesto, se obtiene que $0 < h \leq 1$. Este parámetro toma en cuenta la historia recorrida por cada partícula y refleja la evolución de la velocidad de cada partícula, es decir, a menor valor de h , más rápida es la velocidad.

En cambio, el factor el grado de agregación viene dado por la siguiente ecuación:

$$s = \left| \frac{\min(F_{tbest}, \overline{F}_t)}{\max(F_{tbest}, \overline{F}_t)} \right| \quad (4.5.15)$$

donde \overline{F}_t corresponde a la media del valor fitness de todas las partículas del enjambre en la t-ésima iteración.

De esta manera, w puede ser determinado como una función de los parámetros h y s : $w_i^t = f(h_i^t, s)$.

El propósito de la variación de w , es dar al algoritmo una mejor capacidad de búsqueda y que pueda salir de óptimos locales. Mientras tanto, con el fin de evitar el efecto de similitud del enjambre, la capacidad de salir de óptimos locales debiese ser mejorada, es decir, cuando el grado de agregación se hace más grande, el peso de inercia debería incrementar proporcionalmente. Es por ello, que los autores proponen que el factor de inercia se obtenga a partir de la siguiente ecuación:

$$w_i^t = w_{ini} - \alpha(1 - h_i^t) + \beta s \quad (4.5.16)$$

Donde w_{ini} es el valor inicial de w y es inicializado en 1. Los valores de α y β se encuentran habitualmente dentro del rango $[0,1]$.

4.5.7 PSO binario

La versión binaria del PSO fue presentada por Kennedy y Etberhart. Ésta es útil para hacer comparaciones entre la codificación binaria en GA y PSO, así como para representar problemas que son binarios por naturaleza. Un uso típico puede ser la representación gráfica de la conexión en una red neuronal, donde un valor 1 representa una conexión entre dos nodos, mientras que un valor 0 representa la ausencia de conexión entre dos nodos de la red. Un PSO binario puede ser utilizado para representar la estructura de una red [Van Den Bergh, 2002].

4.5.8 Coeficiente de restricción

Trabajos recientes han determinado que un *factor de restricción* (K) puede ayudar a mejorar la convergencia del algoritmo PSO. El modelo con factor de restricción describe, entre otras cosas, una manera de elegir los valores de φ_1 y φ_2 para asegurar la convergencia. Al elegir estos valores correctamente, la necesidad de fijar los valores de v_i^k entre el intervalo $[-v_{max}, v_{max}]$ es obviada.

La actualización de la velocidad incorporando el factor de restricción viene dada por la siguiente ecuación:

$$v_{ij}^{k+1} = K \cdot v_{ij}^k + \varphi_1 \cdot rand_1 \cdot (p_{ij} - x_{ij}^k) + \varphi_2 \cdot rand_2 (p_{gj} - x_{ij}^k) \quad (4.5.17)$$

donde:

$$K = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad (4.5.18)$$

siendo $\varphi = \varphi_1 + \varphi_2$, $\varphi > 4$

4.5.9 Optimización por Enjambre de Partículas basada en la relación fitness distancia

El proceso de aprendizaje de PSO está basado en la capacidad de las partículas para aprender de su propia experiencia y de la experiencia de la partícula más exitosa. La modificación FDRPSO propone que las partículas son también capaces de aprender de la experiencia de otras partículas, por ejemplo, la mejor vecina, aunque ésta no corresponda a la mejor partícula global [Sedighzadeh and Masehian, 2009]. La nueva partícula a considerar, debe satisfacer dos criterios:

- La partícula elegida debe estar cerca de aquella que se está actualizando.
- La partícula elegida debe haber visitado una posición con mejor fitness.

La ecuación original de la velocidad debe ser actualizada por la siguiente:

$$v_{ij}^{k+1} = K \cdot v_{ij}^k + \varphi_1 \cdot rand_1 \cdot (p_{ij} - x_{ij}^k) + \varphi_2 \cdot rand_2 (p_{gj} - x_{ij}^k) + \varphi_3 \cdot rand_3 (n_{ij} - x_{ij}^k) \quad (4.5.19)$$

donde los valores usuales para los parámetros $\varphi_1, \varphi_2, \varphi_3$ se encuentran entre 1 y 2.

4.6 Aplicaciones de PSO

Desde sus principios, PSO ha sido utilizado en gran parte para el entrenamiento de las redes neuronales, incidiendo en gran medida en el desarrollo de la arquitectura de la red neuronal en conjunto con la determinación de los pesos de la red [Van Den Bergh, 2002]. En [Van Den Bergh, 2002] se describen varias aplicaciones de PSO para el entrenamiento de las redes neuronales, entre ellas se tiene la utilización de PSO para entrenar una red neuronal usada para simular y controlar el proceso final de la molienda, el cual consiste en la eliminación del metal en el proceso de fabricación, usando herramientas de CNC. Además, se menciona, que se han entrenado redes neuronales para poder determinar el estado de carga de un grupo de baterías de vehículos.

En [Bazi and Melgani, 2007] se trabajó con PSO para optimizar un modelo de regresión basado en SVM, donde se elaboró un nuevo enfoque de regresión semisupervisado, con el fin de estimar los parámetros biofísicos, en este caso la concentración de clorofila, que se encuentran limitados por la poca cantidad de datos de entrenamiento.

También, PSO ha sido utilizado para optimizar la mezcla de ingredientes químicos usados para facilitar el crecimiento de cepas de bacterias, dando como resultado una mejora significativa sobre las soluciones encontradas por métodos previos.

En el control de fuentes reactivas y de voltaje, se ha logrado determinar que PSO es muy efectivo para la optimización de variables continuas y discretas simultáneamente, esto se debe a la capacidad que tiene PSO de ser expandido a problemas de optimización no lineal.

Otra aplicación de PSO se encuentra en [Herrera et al., 2009], donde se presenta un estudio en el cual por primera vez se aplica PSO a la solución de problemas electromagnéticos. En el trabajo, se realiza un diseño eficiente de un filtro corrugado en guía de onda para la banda C, utilizando PSO. En [Quintero, 2010] PSO fue sometido a un ambiente con fotografías, con el objetivo de utilizarlo en el reconocimiento de rostros.

5 Modelo LS-SVM + PSO

5.1 Introducción

La obtención de los parámetros internos de una Máquina de Soporte Vectorial es una tarea compleja, es por ello, que para encontrar los valores óptimos de éstos, se hará uso de ciertas técnicas que permitirán definir la configuración que mejor se adecue al comportamiento de la especie en estudio. Para ello, se deberá hacer uso de diversas técnicas que permitan medir el desempeño de cada modelo a la hora de predecir los desembarques mensuales de anchoveta.

En el presente capítulo, se describirán cada una de las métricas de rendimiento empleadas así como también, la función fitness utilizada para el algoritmo de Optimización por Enjambre de Partículas. Finalmente, se expondrá el modelo general que será utilizado para poder implementar los diversos modelos de regresión propuestos para la investigación.

5.2 Software y Hardware Utilizado

La implementación de los modelos propuestos se llevó a cabo bajo el lenguaje de programación Matlab, utilizando su versión R2010a. Por medio de esta herramienta, se pudo hacer uso del Toolbox Ls-svmlab 1.7[De Brabanter et al., 2010], el cual provee una variedad de funciones LS-SVM para resolver problemas de clasificación, regresión y predicción de series de tiempo. Además, por medio de Matlab se desarrollaron tres variantes del algoritmo PSO, las cuales permitirán ajustar los parámetros internos de la Máquina de Soporte Vectorial de Mínimos Cuadrados.

El hardware utilizado tanto para la implementación como para la evaluación de los modelos posee las siguientes características:

- Procesador Intel Core 2 Duo.
- Velocidad del Procesador 1.5 GHz.
- Memoria RAM 2Gb.

5.3 Descripción y Tratamiento de los Datos

Los datos utilizados corresponden a los valores asociados a los desembarques totales de anchoveta registrados en la zona Centro-Sur de Chile, considerándose los volúmenes de capturas a partir de Enero de 1948 hasta Diciembre de 2008, alcanzando un total de 732 registros.

Para el modelo propuesto, los datos asociados a este recurso pelágico fueron en primera instancia normalizados con la ecuación 5.3.1, posteriormente, éstos fueron desfasados variando de 6 a 12 meses el desfase, con el fin de obtener la cantidad de meses previos que mejor represente el comportamiento de la serie de datos en estudio. A continuación, los datos de anchoveta se dividieron en dos grupos: 80 % de los datos se utilizaron durante la etapa de entrenamiento, mientras que un 20 % se emplearon en la fase de validación. Ambos grupos fueron separados en entradas y salidas, siendo las primeras de ellas suavizadas, utilizándose un promedio móvil de tres meses, esto con la finalidad de eliminar el ruido y así, lograr una clara observación de la tendencia de los datos.

$$y(t) = \frac{y(t)}{\max(y)} \quad (5.3.1)$$

5.4 Métricas de Evaluación

A la hora de crear un modelo predictivo, es de vital importancia medir su desempeño, comparando los datos pronosticados con los datos reales observados para un periodo en particular. En la actualidad, existen diversos métodos que permiten determinar los errores generados por una técnica de pronóstico. La medición del error, proporciona el nivel de confiabilidad de un modelo, con el fin de comparar el grado de exactitud que éste tiene en relación a otros y de esta manera, seleccionar aquél que sea más representativo para la serie de datos que se está analizando. La exactitud de una técnica de pronóstico, se realiza mediante la comparación de la serie original $(Y_t, Y_{t+1}, Y_{t+2}, \dots)$ con la serie de valores pronosticados $(\hat{Y}_t, \hat{Y}_{t+1}, \hat{Y}_{t+2}, \dots)$.

Los diversos métodos existentes, utilizan una notación matemática en particular, en donde se debe distinguir entre el valor observado y el estimado. De esta manera, la notación básica para evaluar los modelos predictivos es la siguiente:

Y_t = valor de una serie de tiempo en el periodo t .

\hat{Y}_t = valor pronosticado de Y_t .

$e_t = Y_t - \hat{Y}_t$ = residual o error residual de pronóstico.

Para poder determinar los parámetros más adecuados de la LS-SVM y así, obtener el modelo que presente los mejores resultados para anchoveta, se hará uso de las siguientes métricas [Hyndman and Koehler, 2006]:

- Error Cuadrático Medio (MSE): Consiste en el promedio de los errores al cuadrado entre los datos observados y los estimados por el modelo. Su valor se obtiene por medio de la siguiente ecuación:

$$MSE = \frac{1}{n} \sum_{t=1}^n (Y_t - \hat{Y}_t)^2 \quad (5.4.1)$$

- Raíz Error Cuadrático Medio (RMSE) : Corresponde a la raíz cuadrada del MSE y es una estimación de la desviación estándar del error de predicción. Se calcula a través de la siguiente ecuación:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (Y_t - \hat{Y}_t)^2} \quad (5.4.2)$$

- Error Medio Absoluto (MAE): Consiste en el promedio de los valores absolutos entre la diferencia de la predicción obtenida por el modelo y la serie de datos observada correspondiente. Su valor se obtiene por medio de la siguiente ecuación:

$$MAE = \frac{1}{n} \sum_{t=1}^n |\hat{Y}_t - Y_t| \quad (5.4.3)$$

- Coeficiente de Determinación (R^2): Permite determinar la dependencia entre los datos reales y los pronosticados. Si el coeficiente de determinación devuelve un valor 0, esto quiere decir, que existe independencia entre los datos, en cambio, si el valor es 1 se dice que los datos son dependientes. El coeficiente de determinación esta dado por la siguiente ecuación:

$$R^2 = 1 - \frac{\sum_{t=1}^n (Y_t - \hat{Y}_t)^2}{\sum_{t=1}^n (Y_t - \bar{Y})^2} \quad (5.4.4)$$

- Escala de Error Medio Absoluto (MASE): Corresponde a una de las técnicas más significativas para comparar modelos de pronóstico. Esto se debe, a que es menos sensible cuando se presentan valores cercanos a 0 en la serie de datos. El MASE, viene dado por la siguiente ecuación:

$$MASE = \frac{1}{n} \sum_{t=1}^n \left(\left| \frac{Y_t - \hat{Y}_t}{\frac{1}{n-1} \sum_{i=2}^n |Y_i - Y_{i-1}|} \right| \right) \quad (5.4.5)$$

5.5 Función Fitness

5.5.1 Error Cuadrático Medio

El error cuadrático medio, se empleará en este trabajo tanto como una métrica de evaluación del desempeño de un modelo, así como también, será utilizada como la función fitness del algoritmo PSO. El MSE corresponde a una de las técnicas más empleadas para poder determinar el modelo con mejor rendimiento para una determinada serie de datos. A partir de ella, se determinará la configuración de los parámetros internos de la LS-SVM que mejor represente el comportamiento de los desembarques mensuales de anchoveta.

Para poder estimar los valores adecuados de los parámetros, se hará uso de la ecuación 5.4.1, la que permitirá medir el error global de pronóstico a partir de la configuración que tengan estos parámetros. De esta manera, se podrá determinar aquella que tenga el menor error cuadrático medio y a su vez, obtener los valores óptimos de los parámetros a calibrar.

5.6 Descripción del Modelo General (LS-SVM + PSO)

El modelo general, consiste en una Máquina de Soporte Vectorial de Mínimos Cuadrados, donde los parámetros internos de ésta serán obtenidos mediante la metaheurística PSO, de manera que cada partícula perteneciente al enjambre represente una solución factible dentro del espacio de búsqueda y así, poder obtener aquella que permita diseñar los modelos que mejor representen la conducta de la serie de datos en estudio.

El proceso de optimización de los parámetros de la LS-SVM mediante PSO se puede observar en la figura 5.1, el cual se describe a continuación:

1. **Inicialización.** El algoritmo PSO es inicializado con una población de partículas, donde se le asignan a cada una de ellas, posiciones y velocidades de manera aleatoria.
2. **Entrenamiento del modelo.** El modelo LS-SVM es entrenado con los parámetros incluidos en una partícula. La codificación de la partícula depende del tipo de Kernel a utilizar, por lo tanto, una partícula puede estar representada por los siguientes parámetros [Wu et al., 2009]:

Tipo	Parámetro 1	Parámetro 2
Kernel Lineal	Γ	–
Kernel Polinomial	d	t
Kernel RBF	C	σ^2

Tabla 5.1 – Tipos de Kernel con sus diferentes parámetros a optimizar.

3. **Obtención valor fitness.** Se calcula el valor fitness de la partícula evaluada en el

modelo, el cual puede ser obtenido por medio de distintas funciones. Para este caso, se hará uso del error cuadrático medio, ver ecuación 5.4.1.

4. Actualización de la mejor posición de la partícula y del enjambre según valor fitness.
5. **Actualización de la velocidad y posición de las partículas.** La velocidad de cada partícula es calculada según la ecuación 4.3.1, mientras que la nueva posición de éstas, estará asociada a la ecuación 4.3.2.
6. **Finalización.** Se verifica si se cumple con el criterio de término definido, de no cumplirse se vuelve al paso 2, en caso contrario continuar.
7. **Obtención de los parámetros óptimos.** Una vez que se ha terminado con la ejecución del algoritmo PSO, se obtiene la partícula que mejor representa la solución del problema, es decir, aquella partícula que presenta menor valor fitness.
8. **Evaluación del modelo.** Luego de obtener los parámetros óptimos del modelo, éste se debe evaluar con los datos pertenecientes a la etapa de validación, para ello se debe hacer uso de las ecuaciones 5.4.1, 5.4.2, 5.4.3, 5.4.4 y 5.4.5.

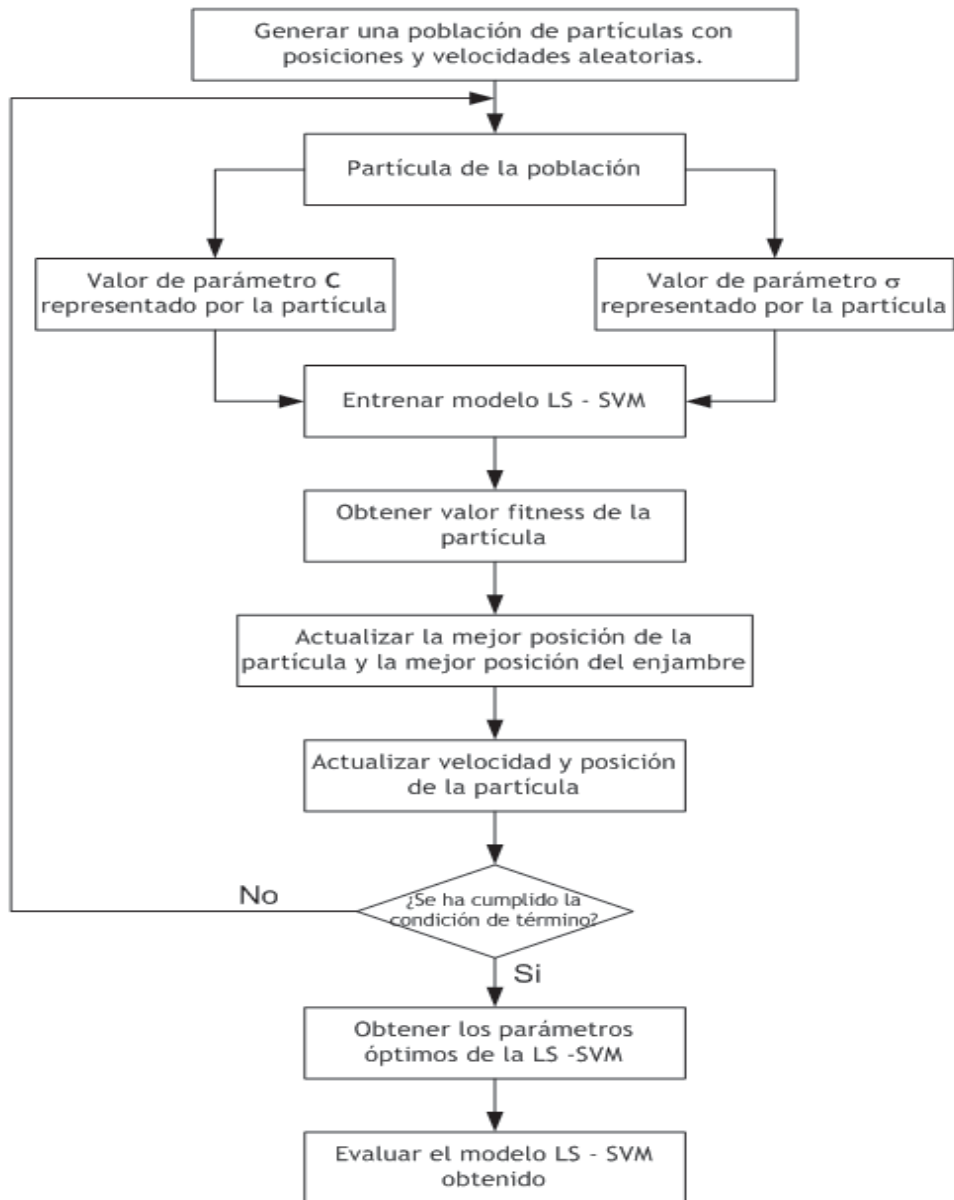


Figura 5.1 – Proceso de optimización de los parámetros de la LS-SVM con optimización por enjambre de partículas utilizando Kernel RBF – Gaussiano.

6 Evaluación de los Modelos de Regresión

6.1 Características del Entrenamiento

Como se ha mencionado en los capítulos anteriores, la configuración de los parámetros internos de la LS-SVM se llevará a cabo bajo el algoritmo PSO, siendo implementadas y evaluadas para esta investigación 3 variantes distintas de la Optimización por Enjambre de Partículas. Para la especie en estudio se ha seleccionado y utilizado el Kernel función de base radial (RBF), debido a que es éste el que presenta mejores resultados, según literatura.

Con el objetivo de determinar aquella configuración que represente de mejor manera el comportamiento de los desembarques mensuales de la especie anchoveta, se ha utilizado un 80 % de la información durante la etapa de entrenamiento, equivalente a 586 datos, mientras que para la etapa de validación del modelo se ha utilizado el 20 %, correspondiente a 146 registros.

Uno de los aspectos más importantes a la hora de realizar el entrenamiento de la máquina, consiste en poder estimar de manera adecuada la configuración óptima para cada una de las variantes del algoritmo PSO, ya sea la cantidad de partículas que conforman el enjambre así como también, el número de iteraciones del algoritmo, esto con el fin de evitar una convergencia prematura del algoritmo PSO. Además, se debe determinar la cantidad de meses previos que mejor represente el comportamiento de los desembarques mensuales de anchoveta en la zona Centro-Sur de Chile, siendo esto conocido como el número de desfases de la data de entrada.

El proceso consistió en fijar una configuración inicial de los parámetros internos de la Optimización por Enjambre de Partículas para las tres variantes en estudio. Para ello, se efectuaron las siguientes actividades:

1. La primera de ellas, consistió en alterar el número de desfases desde 6 hasta 12 meses, realizando un total de 5 ejecuciones para cada uno de éstos. Siendo fijado el número de partículas y el número de iteraciones según la tabla 6.1.
2. Una vez determinado el número de meses previos necesarios para realizar una buena estimación, la siguiente etapa consistió en determinar el tamaño del enjambre adecuado para cada variante, realizando una modificación del número de partículas desde 10 hasta 100, variando éstas de 10 en 10. Para realizar este proceso, se mantuvo constante el número de iteraciones según la tabla 6.1 así como también, el número de desfases determinado según el punto anterior.
3. Finalmente, una vez definidos los valores óptimos asociados al número de partículas y a la cantidad de desfases, se realizó una variación desde 20 hasta 80 en el total de iteraciones del algoritmo, para lo cual, el valor asociado al desfase y al tamaño del enjambre se asignaron según los resultados obtenidos previamente. Al igual que

para el paso 1 y 2, se realizaron 5 ejecuciones para cada una de las alteraciones que sufrió el número de iteraciones.

Es importante mencionar, que la selección de los valores para cada uno de los parámetros a configurar consistió en aquél que presenta el mejor promedio y el máximo coeficiente de determinación. La configuración más adecuada para cada una de las variantes será presentada al momento de explicar cada uno de los modelos propuestos.

Parámetros PSO	Valor
Tamaño del enjambre	50
Número de iteraciones	30
Rango partícula C	[0,1;1000]
Rango partícula σ^2	[1;500]
fitness	MSE
Componente cognitiva	2
Componente social	1,8
Velocidad máxima	0,7

Tabla 6.1 – Configuración base Algoritmo PSO.

6.2 Modelo LS-SVM con PSO con factor de inercia decreciente linealmente

Según el proceso descrito en la sección 6.1, se presentan las siguiente tablas que hacen mención a la obtención de los parámetros del PSO-IW:

	6	7	8	9	10	11	12
Máximo	0,9208	0,9252	0,9269	0,9312	0,9363	0,9292	0,9156
Mínimo	0,8023	0,8208	0,8403	0,8559	0,8487	0,8698	0,8615
Media	0,8498	0,8743	0,8925	0,8988	0,9006	0,8948	0,8915

Tabla 6.2 – Resumen variación de desfases modelo LS-SVM + PSO-IW.

	10	20	30	40	50	60	70	80	90	100
Máximo	0,8629	0,8735	0,9342	0,9301	0,9316	0,9308	0,9358	0,9259	0,9303	0,9315
Mínimo	0,8504	0,8521	0,8684	0,8847	0,8892	0,8591	0,9053	0,8798	0,8717	0,8982
Media	0,8564	0,8652	0,9021	0,9167	0,9191	0,8882	0,9234	0,9019	0,8923	0,9186

Tabla 6.3 – Resumen variación de partículas modelo LS-SVM + IW.

	20	30	40	50	60	70	80
Máximo	0,9347	0,9383	0,9354	0,9336	0,9387	0,9347	0,9316
Mínimo	0,9293	0,9094	0,9102	0,9283	0,9281	0,9297	0,9031
Media	0,9313	0,9256	0,9249	0,9316	0,9335	0,9325	0,9244

Tabla 6.4 – Resumen variación de iteraciones modelo LS-SVM + PSO-IW.

A partir de los resultados anteriores, se puede determinar que el desfase de 10 meses es el más adecuado para esta variante, presentando un coeficiente de determinación máximo que explica en un 93,63 % la varianza, además, registra la mejor media para el coeficiente de determinación con un valor de 90,06 %. En la tabla 6.3, el máximo coeficiente de determinación se obtiene con un enjambre de 70 partículas, valor que explica en un 93,58 % la varianza. Finalmente, el número de iteraciones para esta variante se debe configurar en 60, debido a que según la tabla 6.4 presenta un coeficiente de determinación máximo de 93,87 %. Por lo tanto, la configuración para el algoritmo PSO-IW es la siguiente:

Parámetros	Valor
Desfases	10
Tamaño del enjambre	70
Número de iteraciones	60
Rango partícula C	[0,1;1000]
Rango partícula σ^2	[1;500]
fitness	MSE
Componente cognitiva	2
Componente social	1,8
Velocidad máxima	0,7

Tabla 6.5 – Configuración base Algoritmo PSO-IW.

Al utilizar PSO-IW, la ecuación de velocidad de la Optimización por Enjambre de Partículas convencional es alterada, a ésta se le agrega un factor de inercia, w , quedando representada a través de la ecuación 4.5.1. La particularidad de este factor es que a medida que incrementan las iteraciones del algoritmo, éste decrecerá dentro de un intervalo ya definido, por medio de la ecuación 4.5.2. El intervalo escogido para este factor está comprendido entre [0,3 - 1,2].

Bajo esta configuración del Algoritmo PSO-IW, se pretende determinar los parámetros C y σ^2 con los cuales se obtenga el modelo que mejor representa el comportamiento para los desembarques mensuales de anchoveta en la zona Centro-Sur de Chile, para ello se realizó un proceso de 20 ejecuciones para el modelo LS-SVM + PSO-IW, alcanzando un tiempo cercano a los 14.702 segundos, equivalente a 4 horas con 10 minutos aproximadamente. A continuación, en las tablas 6.6 y 6.7, se presenta el detalle de los resultados obtenidos al utilizar este modelo predictivo.

Ejecución	C	σ^2	Tiempo (Seg)	MSE	RMSE	MAE	R^2	MASE
1	924,26	1,02	770,09	0,0000175	0,00418	0,00281	0,9325	0,30729
2	518,99	1,00	764,72	0,0000188	0,00433	0,00291	0,9279	0,31761
3	933,95	1,31	762,78	0,0000150	0,00400	0,00270	0,9391	0,29540
4	942,73	1,01	728,32	0,0000178	0,00421	0,00283	0,9316	0,30906
5	730,25	1,02	730,81	0,0000177	0,00421	0,00284	0,9318	0,30962
6	841,67	1,07	728,69	0,0000171	0,00413	0,00279	0,9342	0,30482
7	960,79	1,02	729,42	0,0000175	0,00419	0,00281	0,9324	0,30738
8	875,33	1,04	728,36	0,0000174	0,00417	0,00281	0,9331	0,30634
9	760,24	17,46	728,18	0,0000320	0,00566	0,00360	0,8743	0,39333
10	917,62	1,00	729,86	0,0000179	0,00423	0,00284	0,9312	0,30998
11	619,39	2,28	728,98	0,0000198	0,00445	0,00306	0,9225	0,33421
12	853,12	1,07	729,28	0,0000170	0,00413	0,00279	0,9343	0,30451
13	943,45	1,07	729,41	0,0000170	0,00412	0,00278	0,9345	0,30361
14	950,88	16,10	729,36	0,0000310	0,00557	0,00359	0,8777	0,39256
15	908,82	2,70	728,88	0,0000190	0,00436	0,00301	0,9254	0,32838
16	856,02	1,01	730,83	0,0000178	0,00422	0,00284	0,9314	0,30989
17	910,83	1,03	729,85	0,0000174	0,00417	0,00281	0,9329	0,30653
18	703,83	1,00	729,01	0,0000181	0,00425	0,00286	0,9306	0,31202
19	890,69	1,02	734,66	0,0000176	0,00419	0,00282	0,9324	0,30775
20	942,91	1,07	731,18	0,0000169	0,00411	0,00278	0,9348	0,30322

Tabla 6.6 – Resultados LS-SVM + PSO-IW.

	Media	Desv. Estándar	Máximo	Mínimo
MSE	0,000019	0,0000044	0,0000320	0,0000150
RMSE	0,00434	0,00045	0,00566	0,0040
MAE	0,00291	0,00024	0,00360	0,0027
R^2	0,9262	0,0175	0,9391	0,8743
MASE	0,31818	0,0269	0,39333	0,29540

Tabla 6.7 – Resumen LS-SVM + PSO-IW.

A través de la tabla 6.6, se puede observar que en la ejecución número 3 se obtiene el mejor desempeño para poder estimar las capturas mensuales de anchoveta, registrando las mejores medidas de error, es decir, presenta los menores MSE y RMSE, 0,0000150 y 0,00400 respectivamente, a su vez, exhibe los mínimos valores asociados al MAE y al MASE, 0,00270 y 0,2954 respectivamente. Este resultado se obtiene en un tiempo de ejecución igual a 12 minutos con 42 segundos y los parámetros óptimos obtenidos son: $C = 933,95$ y $\sigma^2 = 1,31$, con esta configuración se obtiene un coeficiente de determinación igual a 93,91% de la varianza explicada. Como se puede apreciar en la tabla 6.6, los mejores resultados se obtienen con valores pequeños para σ^2 los cuales varían entre 1 y 2,28, donde a medida que aumenta este valor se espera que el coeficiente de determinación disminuya, haciendo más inestable al modelo predictivo. Además, se puede observar que el parámetro

C en todas las ejecuciones realizadas, alcanza valores altos, siendo superior a 500 en las 20 repeticiones del modelo.

Por medio de la tabla 6.7, se puede visualizar que bajo el modelo LS-SVM + PSO-IW, se obtiene un promedio del 92,62 % del coeficiente de determinación, mientras que se alcanza un máximo de 93,91 % y un mínimo del 87,43 %, haciendo de esta combinación un modelo estable a la hora de realizar el pronóstico de los desembarques para este recurso pesquero, donde no se espera una gran variación entre cada ejecución. Además, presenta promedios bajos tanto de MSE como de RMSE, así como también, un bajo valor de la desviación estándar, lo que refleja una baja variabilidad en los resultados presentados.

Considerando la ejecución con mejor desempeño de la tabla 6.6, se presentan las gráficas obtenidas a partir de los resultados entregados por ella. La figura 6.1 expone la variación del fitness en cada iteración realizada por el algoritmo PSO-IW en la etapa de entrenamiento de la LS-SVM. Por su parte, la figura 6.2 expone la comparación entre las capturas totales mensuales observadas y estimadas, mientras que la figura 6.3 representa el gráfico de dispersión entre los valores observados y estimados de los desembarques mensuales de anchoveta.

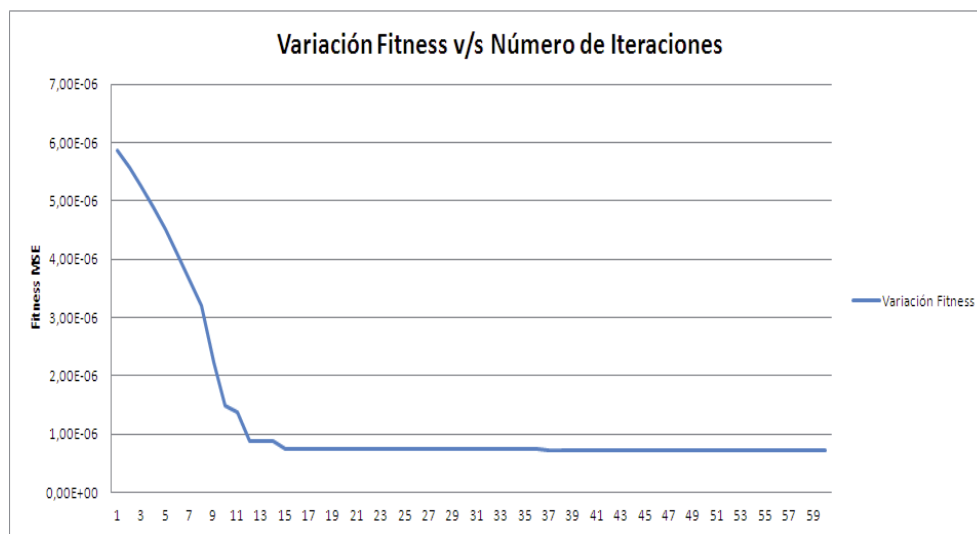


Figura 6.1 – Variación del fitness en el entrenamiento del mejor desempeño de la LS-SVM + PSO-IW.

Como se puede observar por medio de la figura 6.1, el valor inicial del fitness en la etapa de entrenamiento es igual a 0,00000586 y éste disminuye progresivamente hasta la iteración número 12, donde alcanza un valor igual a 0,000000873. A partir de esta última, el valor fitness presenta leves disminuciones hasta la iteración 49, donde alcanza un valor de 0,000000729, siendo este último el que se mantiene constante hasta el término de las iteraciones. Con esto, se puede concluir que el algoritmo PSO utilizado presenta una gran exploración del espacio de soluciones, esto se debe a que no converge rápidamente.

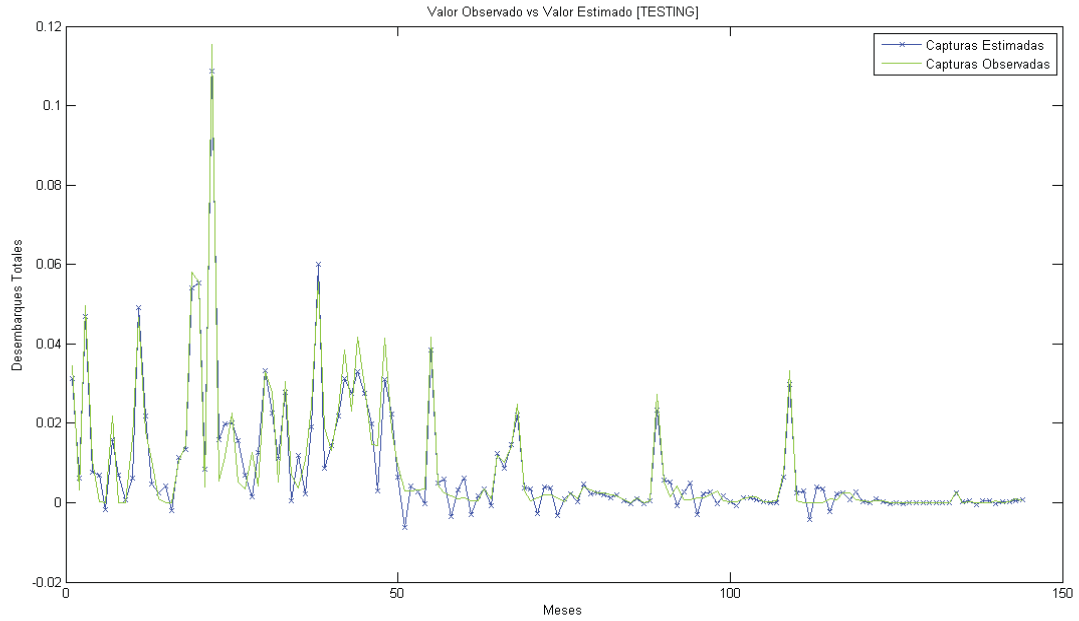


Figura 6.2 – Mejor estimación modelo LS - SVM + PSO-IW con desfase de 10 meses.

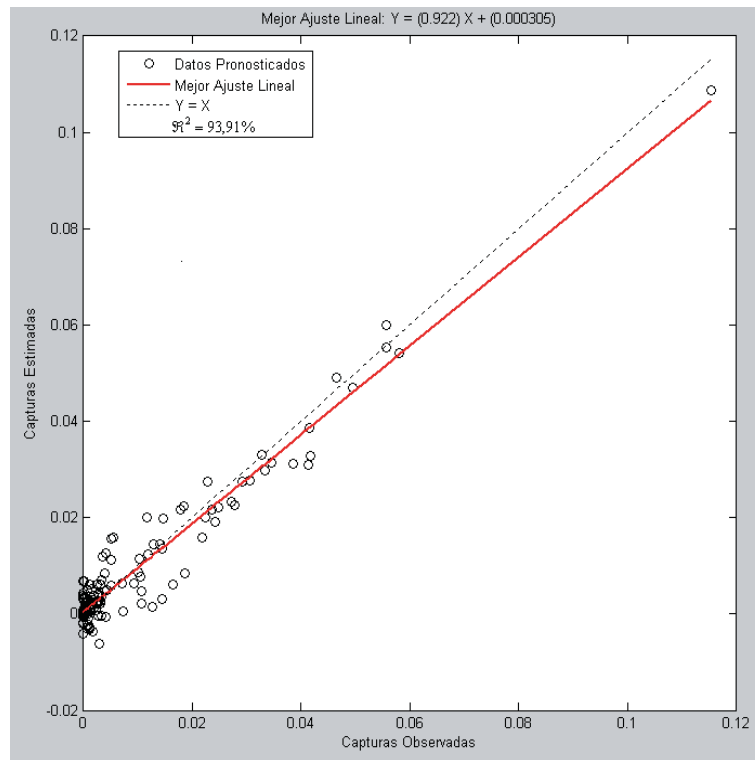


Figura 6.3 – Modelo LS-SVM + PSO-IW con desfase de 10 meses.

6.3 Modelo LS-SVM con PSO con búsqueda local adaptativa

Las siguientes tablas describen los resultados obtenidos a partir de la aplicación del proceso presentado en la sección 6.1 para la variante LSPSO.

	6	7	8	9	10	11	12
Máximo	0,9207	0,9251	0,9282	0,9334	0,9384	0,9192	0,9069
Mínimo	0,8183	0,8331	0,8421	0,8541	0,8668	0,8747	0,8671
Media	0,8387	0,8891	0,8954	0,8828	0,8974	0,8934	0,8891

Tabla 6.8 – Resumen variación de desfases modelo LS-SVM + LSPSO.

	10	20	30	40	50	60	70	80	90	100
Máximo	0,8887	0,891	0,9235	0,9301	0,9318	0,9329	0,9307	0,9319	0,9388	0,938
Mínimo	0,8541	0,8525	0,8622	0,8691	0,8712	0,8619	0,8763	0,8922	0,8805	0,8794
Media	0,8604	0,8749	0,8903	0,902	0,9021	0,9053	0,8947	0,9138	0,9249	0,9126

Tabla 6.9 – Resumen variación de partículas modelo LS-SVM + LSPSO.

	20	30	40	50	60	70	80
Máximo	0,9375	0,938	0,9172	0,9339	0,9315	0,9386	0,9382
Mínimo	0,8717	0,3519	0,8836	0,8757	0,8738	0,932	0,887
Media	0,9077	0,8194	0,9006	0,9195	0,915	0,9354	0,9249

Tabla 6.10 – Resumen variación de iteraciones modelo LS-SVM + LSPSO.

Como se puede observar por medio de las tablas 6.8, 6.9 y 6.10, la mejor configuración consiste en un desfase de 10 meses en los datos, presentando un coeficiente de determinación máximo de 93,84 % de la varianza explicada, además registra el mejor promedio de R^2 con un 89,74 %. A la hora de determinar el tamaño del enjambre, se obtiene el mejor resultado con un número de partículas igual a 90, explicando en un 93,88 % la varianza, además presenta una media para el coeficiente de determinación de 92,42 %, siendo superior a las 50 partículas utilizadas para calcular el número de desfases. Finalmente, se puede apreciar que la estimación del modelo, así como también, su estabilidad mejoran a la hora de hacer uso de 70 iteraciones, alcanzando una media de 93,54 %. En base a lo anterior, se puede plantear la configuración para la variante LSPSO, descrita en la tabla 6.12:

Parámetros	Valor
Desfases	10
Tamaño del enjambre	90
Número de iteraciones	70
Rango partícula C	[0,1;1000]
Rango partícula σ^2	[1;500]
fitness	MSE
Componente cognitiva	2
Componente social	1,8
Velocidad máxima	0,7

Tabla 6.11 – Configuración base Algoritmo LSPSO.

Además, tal como se ha mencionado en la sección 4.5.2, esta variante se basa en explotar de mayor manera el vecindario de la mejor solución encontrada por el enjambre en cada una de las iteraciones. Para ello, calcula una nueva posición a partir del gBest encontrado en la iteración actual, para lo cual hace uso de dos parámetros $\alpha_j(t)$ y $\beta_j(t)$, descritos en la sección 4.5.2.

Al igual que para el modelo anterior, se realizaron 20 ejecuciones con el objetivo de determinar los parámetros internos de la LS-SVM que mejor representan el comportamiento de los desembarques de anchoveta en la zona Centro-Sur de Chile. El entrenamiento de este modelo alcanzó una duración cercana a los 29.558 segundos, lo que equivale a 8 horas con 15 minutos aproximadamente. Por medio de las tablas 6.12 y 6.13, se exponen los resultados obtenidos a través de la utilización del LSPSO.

Como se puede observar en la tabla 6.12, el mejor desempeño alcanzado por este modelo se obtiene en la ejecución número 9. Por medio de ésta, se obtienen valores para los parámetros C y σ^2 iguales a 973,35 y 1,40 respectivamente y se logra en un tiempo de cómputo aproximado de 24 minutos con 44 segundos, duplicando el tiempo alcanzado para el modelo LS-SVM + PSO-IW.

Bajo esta configuración para los parámetros de la LS-SVM, se obtienen las mejores medidas de MSE y RMSE; 0,0000156 y 0,00394 respectivamente. Además se presentan los valores más bajos para el MAE y MASE; 0,00270 y 0,29517 respectivamente.

Siguiendo con la tabla 6.13, se puede observar que este modelo alcanza un valor máximo para el coeficiente de determinación igual a 93,95 % de la varianza explicada, siendo superior a la mejor configuración encontrada por el modelo que utiliza PSO-IW. Sin embargo, se puede advertir que la media alcanzada por este modelo explica en un 91,51 % la varianza, siendo este valor inferior al presentado por LS-SVM con factor de inercia decreciente linealmente, pero aún así se puede determinar que presenta una baja variabilidad en los resultados presentados en cada una de las ejecuciones realizadas, lo cual se aprecia en la desviación estándar alcanzada por el MSE y el RMSE, 0,0000068 y 0,00071 respectivamente.

Ejecución	C	σ^2	Tiempo (Seg)	MSE	RMSE	MAE	R^2	MASE
1	654,41	1,02	1518,93	0,0000180	0,00424	0,00286	0,9308	0,31186
2	930,42	8,84	1531,72	0,0000286	0,00535	0,00346	0,8872	0,37785
3	930,07	18,05	1525,19	0,0000315	0,00561	0,00359	0,8762	0,39191
4	965,66	39,09	1458,52	0,0000335	0,00579	0,00363	0,8685	0,39616
5	854,66	1,72	1469,05	0,0000166	0,00407	0,00282	0,9352	0,30748
6	786,86	1,00	1474,08	0,0000179	0,00423	0,00284	0,9312	0,31051
7	873,06	1,19	1480,97	0,0000161	0,00402	0,00274	0,9375	0,29880
8	980,57	1,04	1482,58	0,0000173	0,00416	0,00280	0,9333	0,30544
9	973,35	1,40	1484,03	0,0000156	0,00394	0,00270	0,9395	0,29517
10	799,90	1,08	1477,71	0,0000170	0,00413	0,00279	0,9343	0,30497
11	922,09	2,76	1474,90	0,0000191	0,00437	0,00301	0,9250	0,32912
12	749,29	24,97	1474,12	0,0000330	0,00574	0,00359	0,8708	0,39173
13	919,71	1,35	1451,70	0,0000157	0,00396	0,00271	0,9390	0,29623
14	867,38	1,19	1448,50	0,0000162	0,00402	0,00274	0,9375	0,29894
15	767,84	5,31	1461,55	0,0000257	0,00507	0,00334	0,8987	0,36435
16	909,91	14,68	1460,24	0,0000309	0,00556	0,00360	0,8782	0,39282
17	565,91	1,38	1462,37	0,0000177	0,00421	0,00290	0,9312	0,31665
18	928,82	1,44	1466,85	0,0000157	0,00396	0,00272	0,9388	0,29722
19	653,86	1,03	1471,17	0,0000179	0,00423	0,00285	0,9313	0,31117
20	657,94	10,72	1484,51	0,0000310	0,00557	0,00359	0,8779	0,39248

Tabla 6.12 – Resultados LS-SVM + LSPSO.

	Media	Desv. Estándar	Máximo	Mínimo
MSE	0,0000217	0,0000068	0,0000335	0,0000156
RMSE	0,00461	0,00071	0,00579	0,00394
MAE	0,00306	0,00037	0,00363	0,0027
R^2	0,9151	0,0275	0,9395	0,8685
MASE	0,33454	0,0405	0,39616	0,29517

Tabla 6.13 – Resumen LS-SVM + LSPSO.

Considerando la ejecución con mejor desempeño de la tabla 6.12, se presentan las gráficas obtenidas a partir de los resultados entregados por ella. La figura 6.4 expone la variación del fitness en cada iteración realizada por el algoritmo PSO en la etapa de entrenamiento de la LS-SVM. Por su parte, la figura 6.5 expone la comparación entre las capturas totales mensuales observadas y estimadas, mientras que la figura 6.6 representa el gráfico de dispersión entre los valores observados y estimados de los desembarques mensuales de anchoveta.

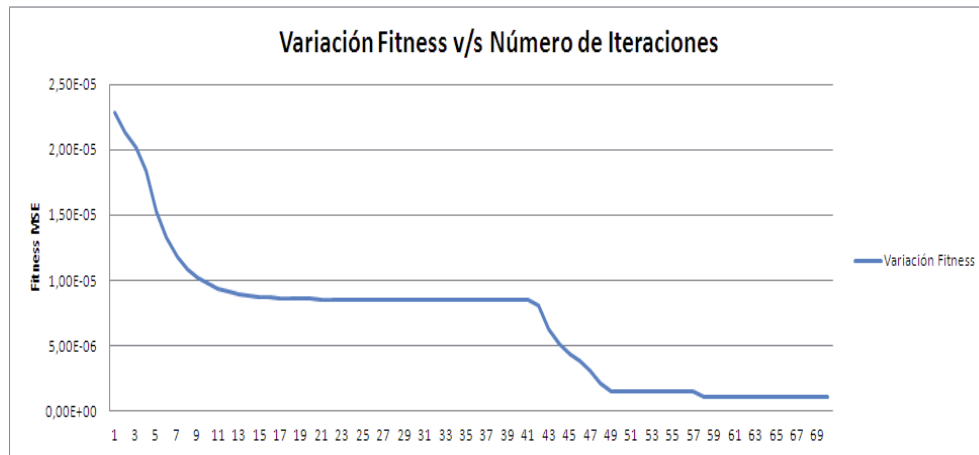


Figura 6.4 – Variación del fitness en el entrenamiento del mejor desempeño de la LS-SVM + LSPSO .

Como se exhibe en la figura 6.4, el valor inicial del fitness en la etapa de entrenamiento es igual a 0,0000229, el cual comienza a disminuir gradualmente a medida que aumentan las iteraciones hasta la número 13, donde alcanza un valor igual a 0,000000897. A partir de aquí, se puede observar una leve baja del fitness hasta la iteración 43, con un valor de 0,00000626. En este punto, el algoritmo vuelve a realizar una exploración del espacio de soluciones, encontrando nuevas partículas que se adecuan de buena manera al problema, lo que se ve reflejado en una nueva disminución del valor fitness hasta la iteración 58, alcanzando un valor cercano a 0,00000104, momento en el que el algoritmo converge a la solución óptima.

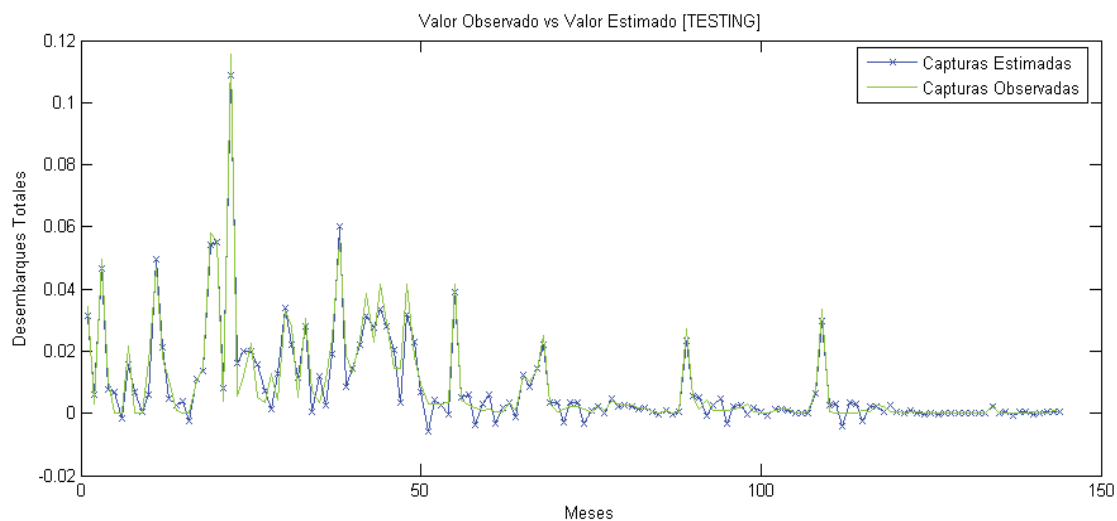


Figura 6.5 – Mejor estimación modelo LS - SVM + LSPSO con desfase de 10 meses.

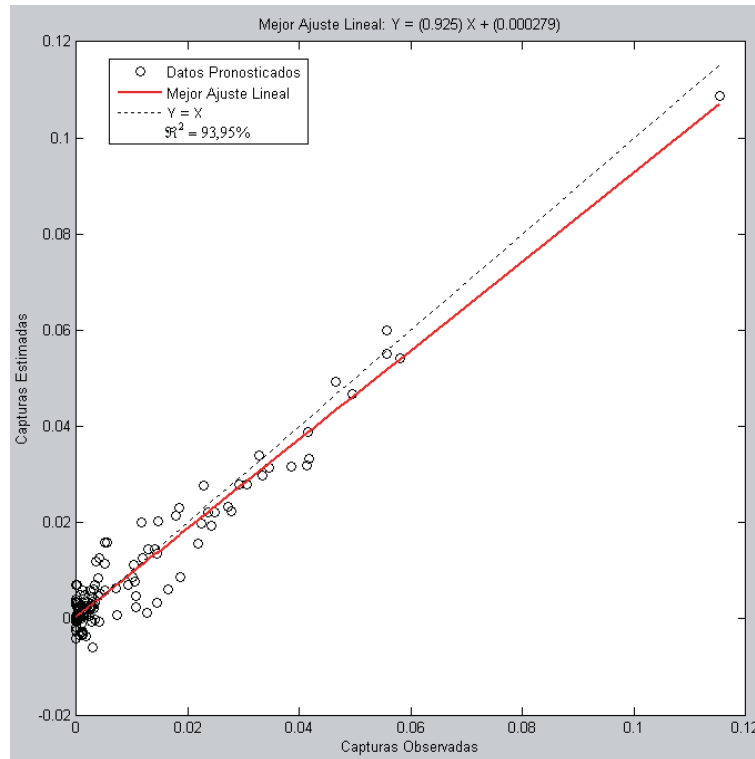


Figura 6.6 – Modelo LS-SVM + LPSO con desfase de 10 meses.

6.4 Modelo LS-SVM con PSO con entorno adaptativo

A continuación, se presentan las tablas que resumen los resultados obtenidos para estimar los parámetros más adecuados de para la variante LAPSO.

	6	7	8	9	10	11	12
Máximo	0,8936	0,8919	0,9283	0,9386	0,9035	0,9209	0,9144
Mínimo	0,8137	0,8176	0,8395	0,8669	0,8786	0,8848	0,8762
Media	0,8354	0,8457	0,9	0,9057	0,8938	0,9044	0,8959

Tabla 6.14 – Resumen variación de desfases modelo LS-SVM + LAPSO.

	10	20	30	40	50	60	70	80	90	100
Máximo	0,8544	0,8597	0,8724	0,9378	0,8982	0,9325	0,9391	0,9399	0,9253	0,9307
Mínimo	0,8471	0,8325	0,852	0,8501	0,8585	0,8642	0,866	0,9377	0,8651	0,8693
Media	0,8517	0,8472	0,8624	0,8962	0,8765	0,9054	0,892	0,9386	0,8916	0,9071

Tabla 6.15 – Resumen variación de partículas modelo LS-SVM + LAPSO.

	20	30	40	50	60	70	80
Máximo	0,9021	0,9338	0,9385	0,9361	0,9387	0,8791	0,8719
Mínimo	0,8746	0,8627	0,8684	0,8663	0,9024	0,8227	0,8685
Media	0,8842	0,8881	0,8923	0,9124	0,9234	0,8705	0,8699

Tabla 6.16 – Resumen variación de iteraciones modelo LS-SVM + LAPSO.

A través de la tabla 6.14 se puede establecer que el número de desfases más adecuado debe ser de 9 meses, debido a que presenta el coeficiente de determinación máximo así como también, presenta el mejor promedio de R^2 , con un 90,57% de la varianza explicada. Por otro lado, a partir de la tabla 6.15 se puede determinar que el enjambre debe ser inicializado en 80 partículas, donde se registra un valor medio de 93,86% en el coeficiente de determinación. Finalmente, la tabla 6.16 presenta los resultados para determinar el número de iteraciones, las que se pueden fijar en 60, debido a que presenta el mejor promedio del coeficiente de determinación, con un valor que explica el 92,34% de la varianza. De esta manera, se puede presentar la tabla 6.17, que describe la inicialización de los parámetros para esta variante.

Parámetros	Valor
Desfases	9
Tamaño del enjambre	80
Número de iteraciones	60
Rango partícula C	[0,1;1000]
Rango partícula σ^2	[1;500]
fitness	MSE
Componente cognitiva	2
Componente social	1,8
Velocidad máxima	0,7

Tabla 6.17 – Configuración base Algoritmo LAPSO.

Para este modelo, se deben configurar además dos parámetros, los cuales hacen referencia a los límites asignados para la magnitud del vector de distribución descrito en la sección 4.5.3, siendo éstos inicializados de la siguiente manera:

- $D_{max} = 0,95$
- $D_{min} = 0,4$

Por medio de las tablas 6.18 y 6.19, se presentan los resultados obtenidos a partir de un total de 20 ejecuciones realizadas para el modelo LS-SVM + LAPSO, con el objetivo de determinar aquella configuración de C y σ^2 que mejor represente las capturas del recurso anchoveta en la zona en estudio. El tiempo de cómputo total para este modelo fue de alrededor de 14.717 segundos, equivalente a 4 horas 10 minutos aproximadamente.

Ejecución	C	σ^2	Tiempo (Seg)	MSE	RMSE	MAE	R^2	MASE
1	943,68	1,08	805,38	0,0000155	0,00393	0,00268	0,9397	0,29272
2	948,07	1,03	825,20	0,0000157	0,00396	0,00269	0,9390	0,29372
3	743,99	4,88	828,31	0,0000279	0,00528	0,00349	0,8905	0,38096
4	901,61	1,11	433,66	0,0000155	0,00393	0,00268	0,9397	0,29292
5	622,45	4,06	836,86	0,0000272	0,00521	0,00348	0,8934	0,38046
6	677,30	8,56	834,39	0,0000330	0,00575	0,00363	0,8698	0,39593
7	821,43	4,11	822,14	0,0000259	0,00509	0,00343	0,8983	0,37416
8	936,29	11,58	833,00	0,0000335	0,00579	0,00365	0,8681	0,39868
9	708,23	1,05	831,83	0,0000348	0,00590	0,00478	0,9356	0,52255
10	904,83	1,13	905,63	0,0000154	0,00393	0,00268	0,9399	0,29279
11	738,12	1,01	862,58	0,0000161	0,00401	0,00273	0,9376	0,29808
12	938,38	9,23	874,49	0,0000321	0,00567	0,00359	0,8733	0,39182
13	997,36	9,14	821,28	0,0000318	0,00564	0,00358	0,8746	0,39044
14	948,65	1,12	822,15	0,0000154	0,00392	0,00267	0,9400	0,29208
15	919,26	3,79	820,50	0,0000247	0,00497	0,00337	0,9033	0,36767
16	818,31	1,05	425,88	0,0000158	0,00397	0,00270	0,9387	0,29528
17	539,77	2,43	837,21	0,0000230	0,00480	0,00329	0,9101	0,35931
18	807,60	2,34	844,20	0,0000203	0,00451	0,00310	0,9204	0,33904
19	935,86	14,28	882,10	0,0000343	0,00586	0,00368	0,8648	0,40207
20	916,63	1,06	859,09	0,0000156	0,00395	0,00269	0,9393	0,29347

Tabla 6.18 – Resultados LS-SVM + LAPSO.

	Media	Desv. Estándar	Máximo	Mínimo
MSE	0,0000237	0,0000077	0,0000348	0,0000154
RMSE	0,00480	0,000796	0,00590	0,00392
MAE	0,00323	0,00054	0,00478	0,00267
R^2	0,9108	0,0294	0,9400	0,8648
MASE	0,3527	0,06	0,5225	0,29208

Tabla 6.19 – Resumen LS-SVM + LAPSO.

Como se puede determinar a través de la tabla 6.18, la configuración que mejor estima los desembarques de anchoveta, se obtiene en la ejecución número 14. Donde los valores de los parámetros C y σ^2 corresponden a 948,65 y 1,12 respectivamente.

Para el mejor desempeño alcanzado, se registra un coeficiente de determinación que explica en un 94,00% la varianza, equivalente al mejor resultado obtenido al comparar los 3 modelos desarrollados. Además, se puede apreciar que los valores de MSE y RMSE son iguales a 0,0000154 y 0,00392 respectivamente. Como se puede observar, se sigue manteniendo la tendencia presentada en los dos modelos anteriores, esto es, que los mejores rendimientos obtenidos presentan valores para el parámetro σ^2 cercanos a 1, por lo cual, mientras más alto es este valor el desempeño del modelo se ve afectado. A su vez, nuevamente se registran valores altos para el parámetro C .

A través de la tabla 6.19, se puede observar que este modelo presenta una media para el coeficiente de determinación que explica en un 91,08 % la varianza, mientras que el valor mínimo de R^2 es igual a un 86,48 %.

A continuación, se presenta la gráfica que describe la variación que presentó el fitness MSE registrado durante la etapa de entrenamiento para el mejor desempeño presentado por el modelo LS-SVM + LAPSO.

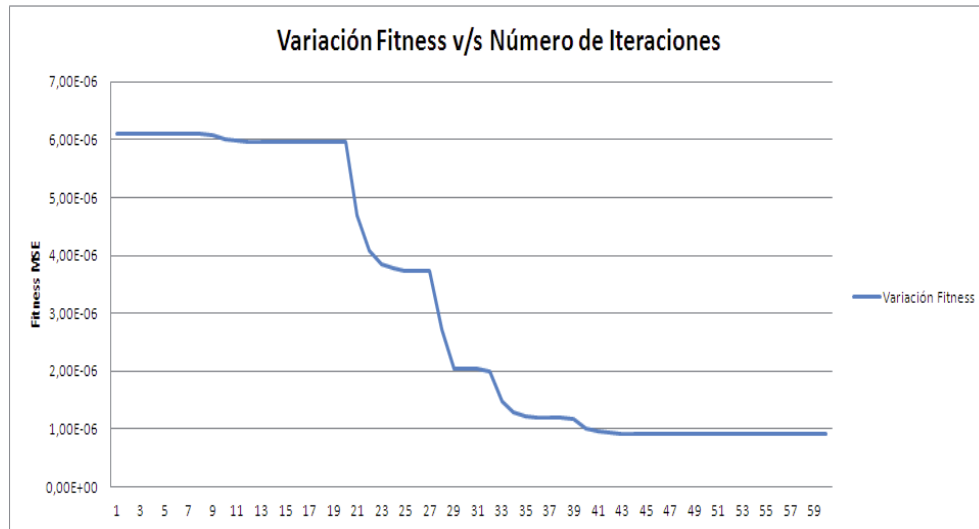


Figura 6.7 – Variación del fitness en el entrenamiento del mejor desempeño de la LS-SVM + LAPSO.

Como se aprecia por medio de la figura 6.7, la presente variante del algoritmo PSO exhibe una gran exploración y explotación del espacio de soluciones, donde se puede percibir claramente que el valor del fitness MSE decrece considerablemente a medida que aumentan las iteraciones. Como punto de partida el grado de adecuación de la mejor partícula tiene un valor cercano a 0,0000061 y que disminuye gradualmente hasta la iteración 41 alcanzando un valor de 0,00000955 a partir de la cual tiene una leve disminución hasta la iteración 48, donde la partícula con mejor desempeño obtiene un valor fitness de 0,00000916, manteniéndose constante hasta el término de las iteraciones.

La figura 6.8 describe las capturas totales mensuales observadas y estimadas por el modelo, a su vez la figura 6.9, representa la correlación que existe entre la data pronosticada y la observada para la anchoveta.

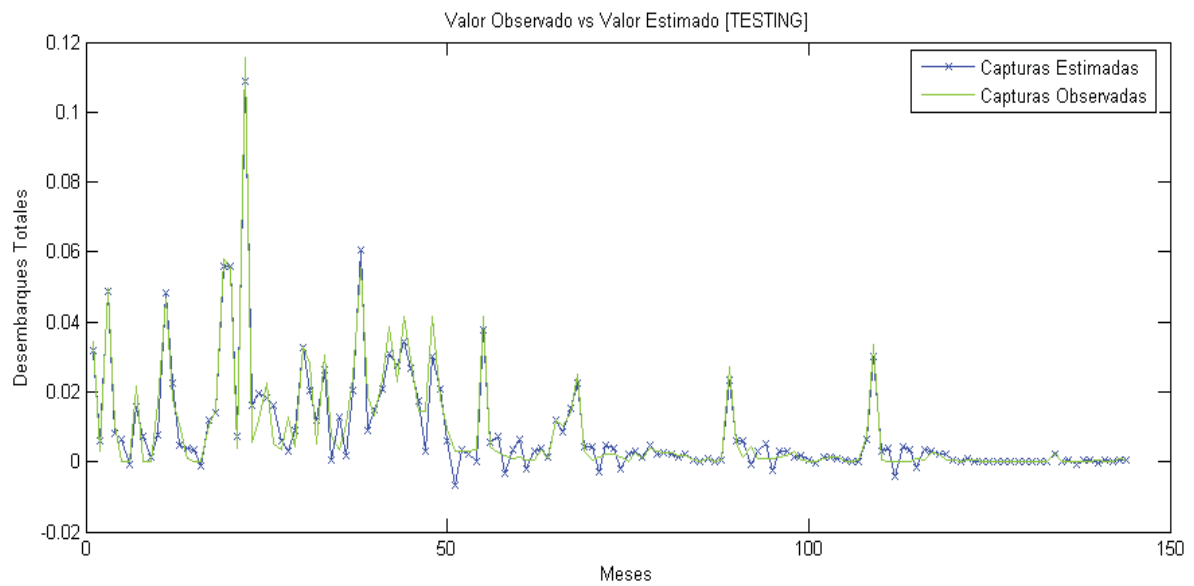


Figura 6.8 – Mejor estimación modelo LS - SVM + LAPSO con desfase de 9 meses.

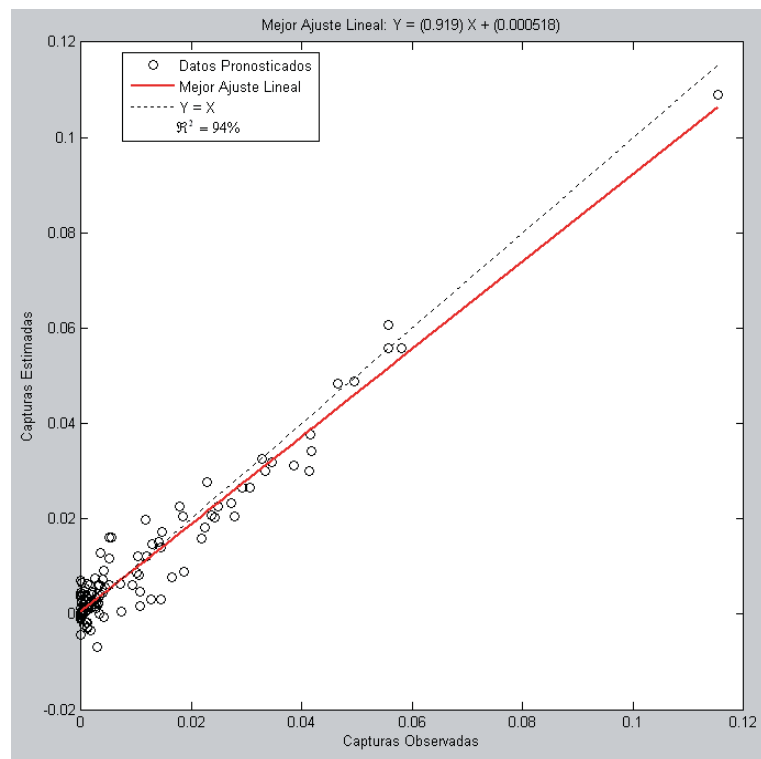


Figura 6.9 – Modelo LS-SVM + LAPSO con desfase de 9 meses.

7 Comparación de Resultados

Este capítulo, tiene como objetivo contrastar los desempeños alcanzados por los distintos modelos presentados en la sección 6. Al comparar los resultados, se puede observar similitudes al utilizar las distintas variantes del algoritmo PSO. En la tabla 7.1, se presentan los mejores resultados obtenidos por cada una de las variantes empleadas, por medio de la cual, se puede observar que los tres modelos desarrollados obtienen desempeños semejantes, donde la variante LAPSO presenta un coeficiente de determinación igual a 94 %, mientras que LSPSO logra explicar un 93,95 % de la varianza. A su vez, el modelo que emplea PSO-IW alcanza un R^2 del 93,91 %.

Variante PSO	Iteraciones	Partículas	MSE	RMSE	MAE	R^2	MASE
PSO-IW	60	70	0,0000150	0,00400	0,00270	0,9391	0,29540
LSPSO	70	90	0,0000156	0,00394	0,00270	0,9395	0,29517
LAPSO	60	80	0,0000154	0,00392	0,00267	0,9400	0,29208

Tabla 7.1 – Mejores resultados por variante.

Debido a que no existe una gran variación entre los modelos, se debe realizar una comparación que permita determinar la selección de uno de ellos. Para esto, se presenta la figura 7.1 donde se puede visualizar la estabilidad de los modelos desarrollados, siendo el que utiliza la variante PSO-IW el que presenta una mayor regularidad en cada una de las ejecuciones, lo cual se puede determinar debido a que registra un valor medio del coeficiente de determinación igual a un 92,62 %, superando a LSPSO y a LAPSO, quienes registran un 91,51 % y un 91,61 % de la varianza explicada respectivamente. Además, PSO-IW consigue el mayor R^2 mínimo, el cual corresponde a un 87,43 % de la varianza explicada, superando nuevamente a LSPSO y LAPSO, donde el primero de ellos alcanza un coeficiente de determinación igual a un 86,85 %, mientras que el segundo, alcanza un valor mínimo del 86,48 % de la varianza explicada.

Junto con lo anterior, se puede apreciar la desviación estándar que se presenta al realizar las 20 ejecuciones por cada modelo. La utilización de la variante PSO con factor de inercia decreciente linealmente registra la menor desviación estándar, igual a 0,0175, en comparación a LSPSO y LAPSO, donde la primera de ellas alcanza un valor igual a 0,0275, mientras que la segunda, presenta una desviación estándar igual a 0,0294. Por medio de estos valores, se puede determinar que el modelo LS-SVM + PSO-IW presenta una menor dispersión en relación a su media, haciendo que este modelo sea más estable a la hora de realizar el pronóstico.

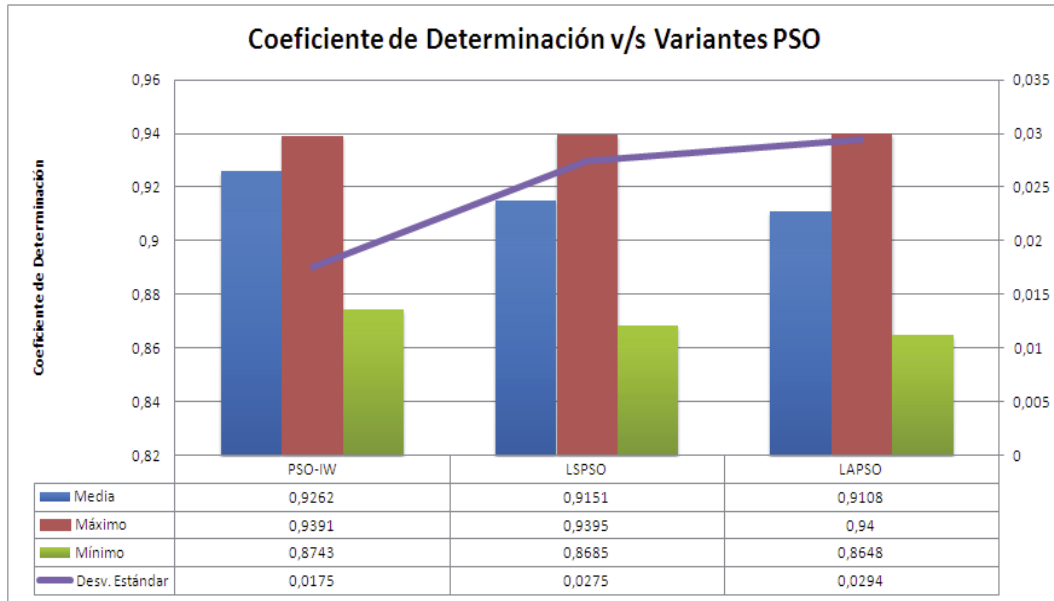


Figura 7.1 – Comparación de Modelos según el Coeficiente de Determinación.

Junto con lo expuesto anteriormente, se observa un contraste en el tiempo de ejecución alcanzado por los mejores desempeños de cada modelo (figura 7.2). La variante PSO-IW exhibe el menor costo de procesamiento, con un tiempo de 12 minutos con 42 segundos, seguido por la variante LAPSO que alcanzó su configuración óptima en 13 minutos con 42 segundos, sin embargo, el tiempo de ejecución sube considerablemente al utilizar el algoritmo LSPSO, quién alcanzó un tiempo de 24 minutos y 44 segundos. La variación que se presenta en el tiempo de cómputo de cada modelo se puede explicar en base a la cantidad de iteraciones y partículas que conforman la topología de cada una de ellas. Como se puede apreciar en la tabla 7.1, la variante PSO-IW registra la menor combinación de iteraciones y partículas, 60 y 70 respectivamente, siendo más eficiente en relación a las variantes LSPSO y LAPSO, donde la primera de ellas fue configurada con un total de 70 iteraciones y 90 partículas, mientras que para la segunda se utilizaron 60 iteraciones y 80 partículas, con lo cual se obtiene una mayor ganancia en términos de procesamiento.

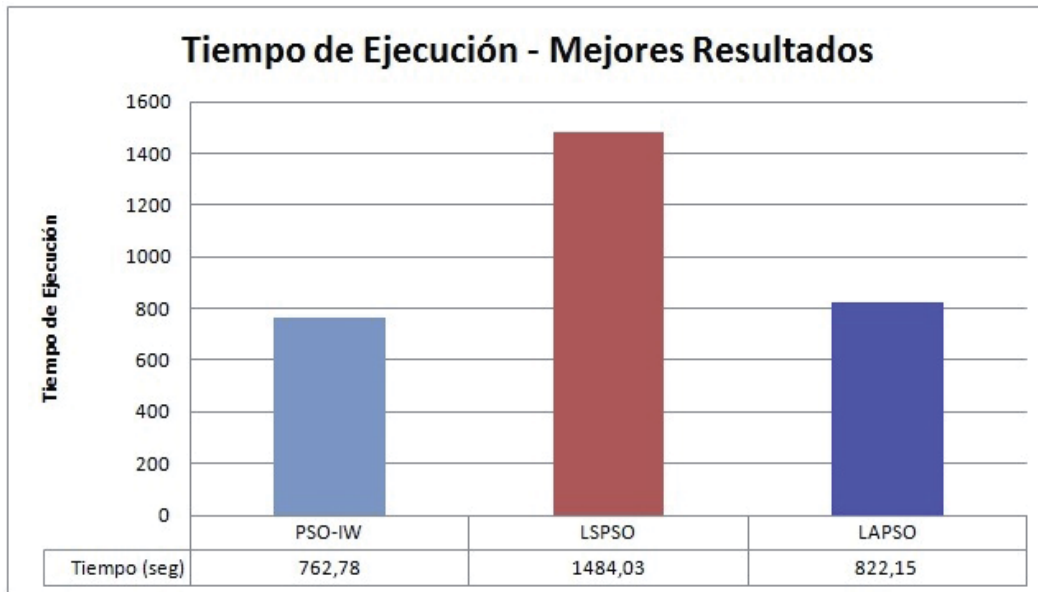


Figura 7.2 – Tiempo de ejecución por cada variante.

Al observar las figuras 6.1, 6.4 y 6.7, se puede determinar que las variantes utilizadas en la presente investigación, registran una gran exploración, así como también, una alta explotación del espacio de soluciones, debido a que no tienen una convergencia prematura hacia la solución óptima y se encuentran constantemente encontrando partículas que se adecuan de mejor manera a la solución del problema.

Por medio de las tablas 6.6, 6.12 y 6.18, se logra apreciar la importancia de restringir adecuadamente el valor asociado al Kernel RBF, ya que valores cercanos a 1 presentan desempeños altos, mientras que si este valor crece demasiado, afecta considerablemente a la estimación del modelo, esta característica estuvo presente en las 3 variantes utilizadas. Si bien el parámetro σ^2 fue restringido en el intervalo [1-500], éste puede seguir ajustándose de una mejor manera con el objetivo de encontrar modelos muchos más estables. La cota inferior fue asignada de esta manera ya que valores por debajo de la unidad, presentaban un modelo sobreajustado a los datos utilizados en la etapa de entrenamiento y por ende, presentaban un desempeño bajo a la hora de realizar la estimación para los desembarques mensuales de anchoveta en la zona Centro-Sur de Chile.

La figura 6.2, presenta la mejor estimación para los desembarques mensuales de anchoveta, donde se espera que la data pronosticada siga la misma trayectoria que la data observada. A partir de ella, se puede observar que los resultados entregados por el modelo LS-SVM + PSO-IW, en gran parte del tiempo, se comporta de manera similar a las capturas reales para la especie, sin embargo, existen algunos puntos en los cuales no se realiza un buen pronóstico, presentado un desfase en el tiempo con respecto a los volúmenes de desembarques observados en la zona en estudio.

8 Conclusiones

En el presente informe, se ha expuesto el estado del arte de las técnicas utilizadas para la elaboración de un modelo autoregresivo que permita estimar los volúmenes de captura mensuales para la especie de anchoveta en la zona Centro-Sur de Chile, para el cual se utilizó LS-SVM con aprendizaje basado en el algoritmo de Optimización por Enjambre de Partículas y sus variantes.

A través del estudio de las SVM, se logró apreciar las grandes potencialidades que éstas poseen para resolver diversos problemas, tanto de clasificación como de regresión. Las Máquinas de Soporte Vectorial difieren de técnicas que han sido históricamente utilizadas para este tipo de problemas, al estar formuladas bajo el principio de minimización del riesgo estructural, en base a lo cual tienen una gran capacidad de generalización. Además, por medio de la investigación se pudo apreciar que las SVM pueden ser aplicadas de gran manera en conjunto de datos no lineales, para ello, se hace necesaria la incorporación de una función Kernel, siendo el parámetro asociado a éste, uno de los más importantes a calibrar, se pudo determinar que la correcta configuración de este parámetro es fundamental para evitar obtener modelos sobreajustados a los datos de entrenamiento.

Para esto último, se utilizó una técnica evolutiva denominada Optimización por Enjambre de Partículas, la que además, fue empleada para configurar de manera adecuada el parámetro asociado al costo de penalización del pronóstico. A través del estudio de PSO, se puede determinar que existen innumerables variantes de esta técnica, las cuales aportan con una serie de características que permiten la correcta estimación de los parámetros que se desean configurar, para ello, se hace necesario que las partículas que conforman la población de individuos, puedan explorar y explotar el espacio de búsqueda, con el objetivo de abarcar la mayor cantidad de posibles soluciones.

Por medio de la investigación realizada, se logró diseñar tres modelos predictivos en base a LS-SVM y la Optimización por Enjambre de Partículas. Para poder estimar correctamente los parámetros internos de la LS-SVM, se utilizó el fitness del Error Cuadrático Medio para cada una de las variantes empleadas, siendo estas el PSO-IW, LSPSO y LAPSO.

Mediante los resultados entregados por cada modelo desarrollado y previo análisis de ellos, se puede concluir que las 3 variantes utilizadas presentan una gran similitud a la hora de estimar los desembarques mensuales de anchoveta, donde el modelo LS-SVM + LAPSO registró un coeficiente de determinación que explica en un 94,00% la varianza, presentando un desfase de 9 meses en la data de entrada. A su vez, la LS-SVM + LSPSO logró explicar en un 93,95% la varianza con un desfase de 10 meses, mientras que el modelo basado en LS-SVM con factor de inercia decreciente linealmente, presenta un coeficiente de determinación igual a 93,91% de la varianza explicada, aplicando un desfase de 10 meses en los datos.

Junto con lo anterior, se pudo observar que el modelo LS-SVM + PSO-IW presenta una mayor estabilidad en las 20 ejecuciones realizadas, lo que se puede ver reflejado en

el valor medio de su coeficiente de determinación, el cual logra explicar un 92,62 % de la varianza. A su vez, el modelo basado en LSPSO presenta un R^2 igual a 91,51 %. Por último, al emplear la variante LAPSO se registra una media para el coeficiente de determinación igual a un 91,08 %.

Uno de los puntos a destacar, consiste en el tiempo de ejecución empleado por cada modelo a la hora de encontrar la configuración que mejor se adecue al problema en estudio. En relación a esto, se logró percibir que el modelo basado en PSO con factor de inercia decreciente linealmente presenta el menor tiempo de procesamiento, donde su mejor desempeño alcanzó los 12 minutos con 42 segundos. Por su parte, el modelo LS-SVM + LSPSO, presenta un incremento considerable a la hora de estimar las capturas mensuales de anchoveta, alcanzando aproximadamente los 24 minutos, lo que se debe principalmente a que en cada iteración del algoritmo, debe calcular una nueva posición para la mejor partícula del enjambre, para la cual se debe realizar una nueva estimación de las capturas mensuales de anchoveta y obtener su nuevo valor fitness. Finalmente, la variante LAPSO registra un tiempo cercano a los 13 minutos con 42 segundos para su mejor estimación.

De esta manera, tanto el valor medio del coeficiente de determinación como el tiempo de cómputo empleado por las mejores estimaciones, cumplen un papel fundamental a la hora de seleccionar aquél modelo con mejor desempeño, es así, que debido a que no se presenta una gran variación entre los resultados más óptimos de cada una de las variantes, es que se destaca al modelo LS-SVM + PSO-IW como aquél que exhibe la mejor estimación para los desembarques de anchoveta en la zona Centro-Sur de Chile. Registrando para esta especie, un 93,91 % de la varianza explicada.

Finalmente, se puede concluir que las Máquinas de Soporte Vectorial pueden ser aplicadas útilmente al problema en estudio, logrando una alta capacidad de aprendizaje sobre los registros históricos de desembarques de anchoveta en la zona Centro-Sur de Chile, a pesar de la gran variabilidad que presentan las capturas para esta especie.

Referencias

- [ASIPES, 2012] ASIPES (2012). *Asociación de Industriales Pesqueros A.G.*
<http://www.asipes.cl>.
- [Bazi and Melgani, 2007] Bazi, Y. and Melgani, F. (2007). Semisupervised pso-svm regression for biophysical parameter estimation. *IEEE T. Geoscience and Remote Sensing*, 45(6):1887–1895.
- [Chun-Lung et al., 2011] Chun-Lung, C., Rong-Mow, J., Tsung-Ying, L., and Cheng-Hsiung, C. (2011). A novel particle swarm optimization algorithm solution of economic dispatch with valve point loading. *Journal of Marine Science and Technology*, 19(1):43–51.
- [Cruz, 2005] Cruz, C. (2005). *Estrategias Coordinadas Paralelas Basadas en Soft-Computing para la Solución de Problemas de Optimización*. PhD thesis, Granada, España.
- [Díaz, 2007] Díaz, G. (2007). *Pronóstico del Precio del Oro Mediante Least Square Support Vector Machines (LS - SVM)*. PhD thesis, Universidad de las Américas.
- [De Brabanter et al., 2010] De Brabanter, K., Karsmakers, P., Ojeda, F., Alzate, C., De Brabanter, J., Pelckmans, K., De Moor, B., Vandewalle, J., and Suykens, J. A. K. (2010). *Ls-svmlab toolbox user’s guide : version 1.7*. Technical Report 10-146, Uppsala University, Division of Systems and Control.
- [Donoso, 2010] Donoso, W. (2010). *Informe Sectorial - Sector Pesca y Salmonicultura*.
- [Flórez et al., 2007] Flórez, J., Morales, G., and Barrera, R. (2007). Voice commands recognizing using support vector machines and spectral bands. *Scientia et Technica*, 37:79–84.
- [Gang and Zhuping, 2011] Gang, R. and Zhuping, Z. (2011). Traffic safety forecasting method by particle swarm optimization and support vector machine. *Expert Syst. Appl.*, 38:10420–10424.
- [Ghali et al., 2009] Ghali, N. I., El-Dessouki, N., Mervat, A. N., and Bakrawi, L. (2009). exponential particle swarm optimization approach for improving data clustering. *International Journal of Electrical, Computer, and Systems Engineering*, 3:208–212.
- [Gunn, 1998] Gunn, S. R. (1998). Support vector machines for classification and regression. Technical report, Faculty of Engineering, Science and Mathematics School of Electronics and Computer Science.
- [Gutiérrez, 2007] Gutiérrez, J. (2007). *Clasificación de Imágenes Usando Máquinas de Soporte Vectorial*.

- [Herrera et al., 2009] Herrera, O. E., Mantilla, I. A., Balbastre, J., and Díaz-Morcillo, A. (2009). Particle Swarm Optimization aplicado al Diseño de Filtros Corrugados. In *XXIV Simposium Nacional de la Unión Científica Internacional de Radio Sección Española*, Cantabria.
- [Hyndman and Koehler, 2006] Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, pages 679–688.
- [Jabbour et al., 2010] Jabbour, G., Márquez, R., Ruiz, L., and Maldonado, L. (2010). Reconocimiento de firmas off-line mediante máquinas de vectores de soporte. *Ciencia e Ingeniería*, 31(1):43–52.
- [Khemchandani et al., 2009] Khemchandani, R., Jayadeva, and Chandra, S. (2009). Regularized least squares fuzzy support vector regression for financial time series forecasting. *Expert Syst. Appl.*, 36:132–138.
- [Kuan-Yu and Cheng-Hua, 2007] Kuan-Yu, C. and Cheng-Hua, W. (2007). Support vector regression with genetic algorithms in forecasting tourism demand. *Tourism Management*, 28(1):215–226.
- [Li et al., 2008] Li, B., Hu, J., and Hirasawa, K. (2008). Financial time series prediction using a support vector regression network. In *IJCNN*, pages 621–627.
- [Maldonado, 2009] Maldonado, S. (2009). *Utilización de Support Vector Machines No Lineal y Selección de Atributos para Credit Scoring*.
- [Pai and Hong, 2005] Pai, P. and Hong, W. (2005). Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms. *Electric Power Systems Research*, 74(3):417–425.
- [Pant et al., 2008] Pant, M., Thangaraj, R., and Abraham, A. (2008). A new quantum behaved particle swarm optimization. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, GECCO '08, pages 87–94, New York, NY, USA. ACM.
- [Quintero, 2010] Quintero, M. (2010). *Reconocimiento de Rostros con Visión por Computadora y Optimización Badas en Cúmulos de Partículas*.
- [Rengifo and L., 2002] Rengifo, P. and L., J. (2002). *Máquinas de Soporte Vectorial*.
- [Sectorial, 2010] Sectorial, D. A. (2010). *Informe Sectorial de Pesca y Acuicultura*.
- [Sedighizadeh and Masehian, 2009] Sedighizadeh, D. and Masehian, E. (2009). Particle swarm optimization methods, taxonomy and applications. *International Journal of Computer Theory and Engineering*, 1(5):486–502.
- [Shi and Eberhart, 1998] Shi, Y. and Eberhart, R. C. (1998). Parameter selection in particle swarm optimization. In *Proceedings of the 7th International Conference on Evolutionary Programming VII*, EP '98, pages 591–600, London, UK. Springer-Verlag.

- [Subpesca, 2010] Subpesca (2010). *Cuota Global Anual de Capturas para Unidades de Pesquería de Anchoveta y Sardina Común V a X Regiones, Año 2011*.
- [Subpesca, 2011] Subpesca (2011). *Cuota Global Anual de Capturas para Unidades de Pesquería de Anchoveta y Sardina Común V a X Regiones, Año 2012*.
- [Tang and Zhao, 2010] Tang, J. and Zhao, X. (2010). A hybrid particle swarm optimization with adaptive local search. *JNW*, 5(4):411–418.
- [Toro et al., 2006] Toro, E., Restrepo, Y., and Granada, M. (2006). Adaptación de la técnica de particle swarm al problema de secuenciamiento de tareas. *Scientia Et Technica*, 12(32):307–312.
- [Van Den Bergh, 2002] Van Den Bergh, F. (2002). *An analysis of particle swarm optimizers*. PhD thesis, Universidad de Granda, Pretoria, South Africa. AAI0804353.
- [Vapnik, 1995] Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA.
- [Wu et al., 2009] Wu, C.-H., Tzeng, G.-H., and Lin, R.-H. (2009). A novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression. *Expert Syst. Appl.*, 36:4725–4735.
- [Yang et al., 2007] Yang, X., Yuan, J., Yuan, J., and Mao, H. (2007). A modified particle swarm optimizer with dynamic adaptation. *Applied Mathematics and Computation*, 189(2):1205–1213.
- [Yisu et al., 2008] Yisu, J., Knowles, J., Hongmei, L., Yizeng, L., and Kell, D. B. (2008). The landscape adaptive particle swarm optimizer. *Appl. Soft Comput.*, 8:295–304.