

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

**“IMPLEMENTACIÓN DE LS-SVM CON PSO PARA EL  
PRONÓSTICO DE LA TEMPERATURA SUPERFICIAL  
DEL MAR EN LA COSTA DEL NORTE DE CHILE”**

**CHRISTIAN EDUARDO MACAYA GATICA**

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

**MARZO, 2010**

# DEDICATORIA

Dedico esta memoria a mis padres y hermano, quienes han sido mis pilares fundamentales en mi vida.

A mi profesor guía, por darme los conocimientos y consejos necesarios para ser un buen estudiante y profesional, y por su apoyo en la realización de mi memoria.

Y por último, agradezco a Dios por darme la fuerza, inteligencia y sabiduría para salir adelante en este largo y hermoso camino llamado Universidad.

*Agradecimientos*

A Dios, mis padres, mis abuelos y mi profesor guía.

## **RESUMEN.**

El Niño (ENOS), es un fenómeno climático causado por la interacción de dos sistemas de presión atmosférica a gran escala. Y como consecuencia, provoca grandes inundaciones y profundos cambios en el ecosistema. La predicción de éste fenómeno, lograría actuar de manera oportuna y preventiva, para disminuir el impacto que se produce en los diferentes sectores económicos de nuestro país.

La solución al problema, considera una variante de la técnica de las Máquinas Vectoriales, denominada LS-SVM, que incorpora la teoría de los Mínimos Cuadrados, para disminuir a dos, el número de parámetros que deberán ser optimizados. Y con la ayuda de la técnica del PSO, podremos llevar a cabo, la optimización de éstos parámetros.

Los resultados obtenidos fueron satisfactorios, logrando una predicción de un 88,75% de exactitud y un 98,21% de persistencia con el modelo.

## **ABSTRACT**

El Niño (ENSO) is a weather phenomenon caused by the interaction of two systems of large-scale atmospheric pressure. And as a result, it is causing major flooding and profound changes in the ecosystem. The prediction of this phenomenon, achieved act in a timely and preventively, to lessen the impact that occurs in different economic sectors of our country.

The solution to the problem, consider a variant of the technique of Vector Machines called LS-SVM incorporating the theory of least squares, to reduce to two the number of parameters that must be optimized. And with the help of the technique of PSO, we can carry out the optimization of these parameters.

The results were satisfactory, achieving a 88.75% prediction accuracy and 98.21% persistence with the model.

**Keywords:** Support vector machine, least squares support vector machines, kernel and Particle Swarm Optimization.

# TABLA DE CONTENIDOS.

TERMINOLOGÍA .....	VIII
ÍNDICE DE FIGURAS .....	IX
ÍNDICE DE TABLAS .....	X
<b>1. INTRODUCCIÓN. ....</b>	<b>1</b>
1.1. INTRODUCCIÓN. ....	1
1.2. OBJETIVOS. ....	3
1.2.1. <i>Objetivo General.</i> .....	3
1.2.2. <i>Objetivos Específicos.</i> .....	3
1.3. ORGANIZACIÓN DEL TEXTO. ....	3
<b>2. MÁQUINAS DE SOPORTE VECTORIAL.....</b>	<b>4</b>
2.1. INTRODUCCIÓN. ....	4
2.2. MARCO CONCEPTUAL DE LAS SVM.....	4
2.3. CLASIFICACIÓN DE LAS SVM.....	5
2.3.1. <i>Caso SVM Linealmente Separable.</i> .....	5
2.3.2. <i>Caso SVM Linealmente No Separable.</i> .....	8
2.4. REGRESIÓN PARA MÁQUINAS VECTORIALES .....	10
2.5. <i>Función de Kernel</i> .....	12
2.5.1. <i>Núcleos de Mercer</i> .....	14
2.6. MÁQUINAS VECTORIALES DE REGRESIÓN CON MÍNIMOS CUADRADOS (LS-SVM) .....	15
2.6.1. TEORÍA DE LOS MÍNIMOS CUADRADOS (LS) .....	15
2.6.2. VENTAJAS DEL MÉTODO DE LOS MÍNIMOS CUADRADOS.....	16
2.6.3. MÉTODO DE LOS MÍNIMOS CUADRADOS EN REGRESIÓN LINEAL .....	16
2.6.4. RECTAS DE REGRESIÓN POR MÍNIMOS CUADRADOS .....	17
2.7. MODELO LS-SVM .....	18
<b>3. OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS (PSO). ....</b>	<b>21</b>

3.1.	INTRODUCCIÓN .....	21
3.2.	MARCO TEÓRICO .....	22
3.3.	ALGORITMO DE PSO .....	24
3.4.	OBSERVACIONES DE PSO.....	25
3.5.	VARIANTES DEL ALGORITMO PSO.....	25
3.5.1.	PSO GAUSIANO (GPSO).....	26
3.5.2.	PSO DISIPATIVOS (DPSO).....	27
3.5.3.	PSO CON CONGREGACIÓN PASIVA (PSOPC).....	27
3.5.4.	PSO ESTIRADO (SPSO).....	28
<b>4.</b>	<b>METODOLOGÍA DE IMPLEMENTACIÓN DE LOS MODELOS SVR CON PSO .....</b>	<b>30</b>
4.1.	INTRODUCCIÓN .....	30
4.2.	DESCRIPCIÓN DE LOS DATOS DE ESTUDIO.....	30
4.3.	COMPONENTES DEL MODELO.....	31
4.3.1.	<i>Framework LS-SVM</i> .....	31
4.3.2.	<i>Parámetro de la LS-SVM</i> .....	31
4.3.3.	<i>Datos de Entrada</i> .....	32
4.3.4.	<i>Tecnología del Modelo</i> .....	33
4.3.5.	<i>Medida de Error</i> .....	33
4.4.	IMPLEMENTACIÓN DEL MODELO LS-SVM. ....	34
4.4.1.	<i>Comportamiento del Desfase</i> .....	37
4.4.2.	<i>Comportamiento del Parámetro C.</i> .....	38
4.4.3.	<i>Comportamiento de la varianza.</i> .....	39
4.4.4.	<i>Análisis de los Resultados</i> .....	40
4.5.	IMPLEMENTACIÓN DEL MODELO LS-SVM CON PSO.....	41
4.5.1.	<i>Descripción del Enjambre</i> .....	43
4.5.2.	<i>Composición de la Partícula</i> .....	43
4.5.3.	<i>Resultado del PSO – Vector GBest</i> .....	43

4.5.4.	<i>Cantidad de Iteraciones</i> .....	43
4.5.5.	COMPORTAMIENTO DE DESFASE EN MODELO LS-SVM-PSO .....	44
4.5.6.	<i>Comportamiento de individuos en modelo LS-SVM-PSO</i> .....	45
4.6.	<i>Análisis de los resultados para LS-SVM-PSO</i> .....	46
4.7.	<i>Resultado del modelo LS-SVM con GPSO</i> .....	46
4.8.	<i>Resultado del modelo LS-SVM con DPSO</i> .....	47
4.8.1.	<i>Resultado del modelo LS-SVM con PSOPC</i> .....	48
4.9.	<i>Análisis de los resultados para el modelo LS-SVM con PSO</i> .....	49
<b>5.</b>	<b>CONCLUSIONES.</b> .....	<b>50</b>
<b>6.</b>	<b>BIBLIOGRAFÍA.</b> .....	<b>52</b>
<b>7.</b>	<b>ANEXO</b> .....	<b>54</b>
7.1.	MÉTODOS OPTIMIZACIÓN DE PARTÍCULAS POR ENJAMBRE CON LS-SVM.....	54

## TERMINOLOGÍA

- SVM: Support Vector Machines
- PSO: Particle Swarm Optimization
- QP: Quadratic Programming
- SVR: Support Vector Regression
- SVR-PSO: Model Support Vector Regression with Particle Swarm Optimization
- LS-SVM: Least Square-Support Vector Machine
- MAPE: Mean Absolute Percentage Error
- RMSE: Root Mean Square Error
- R<sup>2</sup>: Coefficient of Determination



# ÍNDICE DE FIGURAS.

FIGURA 1: GRÁFICO DE LA TEMPERATURA SUPERFICIAL DEL MAR, ENTRE LOS AÑOS 1951 Y 2005 .....	2
FIGURA 2: HIPERPLANO SEPARADOR $w, b$ .....	7
FIGURA 3: MARGEN DE UNA MUESTRA AL HIPERPLANO .....	7
FIGURA 4: MARGEN MÁXIMO DEL HIPERPLANO .....	7
FIGURA 5: REPRESENTACIÓN GRÁFICA DE LA INTERACCIÓN ENTRE EL PARÁMETRO DE ERROR $\xi_i$ EN EL ERROR DE LA CLASIFICACIÓN .....	10
FIGURA 6: EFECTO DE LA TRASLACIÓN $\Phi_{x1, x2} = x1^2, 2x1x2, x2^2$ DE UN ESPACIO DE ENTRADA $X$ A UN ESPACIO DE CARACTERÍSTICAS $\mathcal{H}$ .....	13
FIGURA 7: RECTA DE REGRESIÓN.....	18
FIGURA 8: DIAGRAMA DEL MODELO DE PREDICCIÓN LS-SVM .....	36
FIGURA 9: GRÁFICO DE RESULTADOS DE LS-SVM.....	38
FIGURA 10: GRÁFICO DE RESULTADOS DE IMPLEMENTACIÓN LS-SVM .....	39
FIGURA 11: GRÁFICO DE RESULTADOS DE IMPLEMENTACIÓN DE LS-SVM .....	40
FIGURA 12: DIAGRAMA DEL MODELO DE PREDICCIÓN LS-SVM CON PSO.....	42
FIGURA 13: GRÁFICO DE DATA PRONOSTICADA UTILIZANDO EL MODELO DE LS-SVM CON PSO.....	45
FIGURA 14: GRÁFICO DE DATA PRONOSTICADA UTILIZANDO EL MODELO DE LS-SVM CON PSO.....	46
FIGURA 15: GRÁFICO DE DATA PRONOSTICADA UTILIZANDO EL MODELO DE LS-SVM CON GPSO.....	47
FIGURA 16: GRÁFICO DE DATA PRONOSTICADA UTILIZANDO EL MODELO DE LS-SVM CON DPSO .....	48
FIGURA 17: GRÁFICO DE DATA PRONOSTICADA UTILIZANDO EL MODELO DE LS-SVM CON PSOPC.....	48

# ÍNDICE DE TABLAS

<b>TABLA 1: EJEMPLOS DE FUNCIONES NÚCLEO DE MERCER.....</b>	<b>14</b>
<b>TABLA 2: RESULTADOS DE LA IMPLEMENTACIÓN LS-SVM CON VARIACIÓN DE DESFASE .....</b>	<b>37</b>
<b>TABLA 3: RESULTADOS DE LA IMPLEMENTACIÓN LS-SVM CON VARIACIÓN DE C.....</b>	<b>38</b>
<b>TABLA 4: RESULTADOS DE LA IMPLEMENTACIÓN LS-SVM CON VARIACIÓN DE VARIANZA .....</b>	<b>39</b>
<b>TABLA 5: REPRESENTACIÓN DE LA POSICIÓN DE UNA PARTÍCULA PARA EL MODELO LS-SVM CON PSO.....</b>	<b>43</b>
<b>TABLA 6: RESULTADOS DE LA IMPLEMENTACIÓN LS-SVM-PSO CON VARIACIÓN DE DESFASE .....</b>	<b>44</b>
<b>TABLA 7: RESULTADOS DE LA IMPLEMENTACIÓN LS-SVM CON VARIACIÓN DE INDIVIDUOS .....</b>	<b>45</b>
<b>TABLA 8: RESULTADOS DE LA IMPLEMENTACIÓN LS-SVM CON VARIACIÓN DE PSO.....</b>	<b>49</b>

# **1. INTRODUCCIÓN.**

## **1.1. Introducción.**

El fenómeno del niño, es un evento que ocurre cada 2 a 7 años aproximadamente y afecta a las costas de Asia, Europa, Oceanía y América. También conocido como ENOS (El Niño-Oscilación del Sur), este fenómeno se compone de dos fases. La primera corresponde a una fase de temperaturas cálidas llamada “el niño”. En cambio, la segunda, se compone de una fase fría denominada “la niña”. Esta variación de temperaturas se produce por la interacción que ocurre entre dos sistemas de presión atmosférica a gran escala. El primero es un sistema de baja presión atmosférica superficial, ubicado al oeste del Océano Pacífico (Australia). El opuesto es un sistema de alta presión atmosférica superficial, ubicada el lado este del Océano Pacífico (Sudamérica).

Desde tiempos muy antiguos los pescadores del norte del Perú observaron la existencia de una corriente marina cálida en dirección hacia el sur, que típicamente aparece cerca de la Navidad. De ahí el nombre El Niño para denominarlo, refiriéndose al acontecimiento del nacimiento del Niño Jesús cercano a esa fecha. También observaron que cada cierto tiempo esta corriente era más intensa, más cálida y se presentaba acompañada por graves alteraciones climáticas que se traducían con la ocurrencia de intensas precipitaciones en una región de características áridas. Como consecuencia, esto provoca graves inundaciones, concatenando profundos cambios en el ecosistema marino. Las aguas anormalmente cálidas son pobres en nutrientes y los peces deben migrar buscando aguas más frías. Lamentablemente no todos los organismos marinos tienen la misma capacidad de desplazamiento, originándose altos niveles de mortandad, perjudicando gravemente la actividad pesquera. Por otro lado, se produce la muerte de una gran cantidad de aves que se alimentan de estos peces, principalmente aves guaneras. Solo las aves que logran sobrevivir se desplazan hacia el sur en busca de alimento.

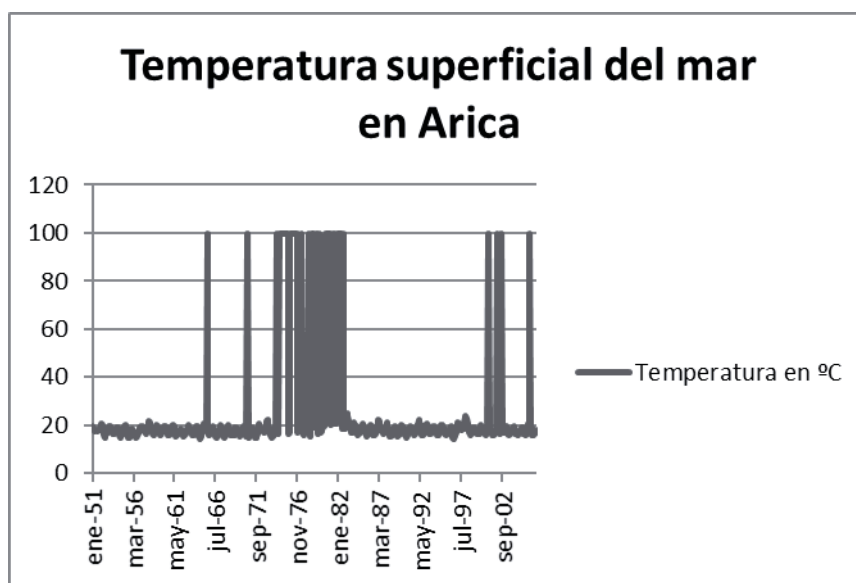
El efecto que provoca este fenómeno en Chile es un incremento en las precipitaciones entre la III y VII región, debido al aumento de la frecuencia de bandas nubosas frontales, lo cual ha producido graves consecuencias en la infraestructura vial, y en sectores económicos como el agrícola, ganadero, vivienda y minero. Es lamentable recordar la tragedia que ocurrió en Santiago, en junio 1982, en la cual se produjo una crecida del río Mapocho, producto de una de las peores temporadas de lluvias registradas en Chile. Además, en otoño e invierno 1997, se produjo una serie de inundaciones en la zona central del país, y un aluvión en la ciudad de Antofagasta. Por otro lado, como se mencionó anteriormente, el desplazamiento de peces, afecta con un gran impacto al área pesquera del norte de nuestro país, ya que se ha observado una gran disminución en el recurso de la anchoveta, y el aumento de la captura de sardinas españolas, y adicionalmente la llegada de nuevas especies no citadas en nuestro país. La predicción de este fenómeno, lograría que nuestro país se pueda preparar de mejor forma para su llegada y así disminuir el impacto

que este produce en los diferentes sectores económicos, como el agrícola, ganadero, vivienda, minero y en las infraestructuras viales.

Las principales variables afectadas por este fenómeno son la intensidad y dirección de los vientos alisios (vientos que soplan de Este a Oeste a lo largo del ecuador), la temperatura superficial del mar, el nivel del mar y la presión atmosférica.

En esta investigación se utilizó la temperatura superficial del mar, en las estaciones de Arica, e Iquique, al norte de Chile. Este conjunto de datos fue extraído manualmente, desde el portal de CENDOC, centro de datos hidrográficos y oceanográficos de Chile, el cual muestra los datos mensuales y anuales, desde las primeras mediciones que se tomaron en estas estaciones ambientales costeras. Cabe destacar que los años en que no se realizaron mediciones, se registraron los datos como 99,9°, por motivos estadísticos.

Para representar los datos, se construyó un gráfico, en el cual se puede apreciar las variaciones de la temperatura superficial del mar, en el transcurso de cada año. El gráfico que se muestra a continuación, representa a los datos registrados desde la estación ambiental costera de Arica.



**Figura 1: Gráfico de la temperatura superficial del mar, entre los años 1951 y 2005**

En este gráfico se pueden apreciar los diferentes cambios que se produjeron en la temperatura superficial del mar, como efecto de la llegada del fenómeno del Niño en las costas del norte de Chile. Como se puede observar, en los años 1997, 1992 y 1986, entre otros, las temperaturas superficiales del mar, superaron los 20° C, debido a la fase cálida que produjo este fenómeno en el norte de Chile.

## **1.2. Objetivos.**

### **1.2.1. Objetivo General.**

Desarrollar un modelo de pronóstico de la temperatura superficial del mar en la zona norte de Chile, utilizando la técnica SVM combinada con un algoritmo de optimización por enjambre de partículas.

### **1.2.2. Objetivos Específicos.**

- Diseñar la estructura de la SVM y estimar sus parámetros utilizando los algoritmos de LS (Mínimos Cuadrados) y PSO.
- Evaluar y comparar el rendimiento de los modelos propuestos.

## **1.3. Organización del Texto.**

Este documento está conformado por cinco capítulos más una sección de anexos, y cumplen con la siguiente distribución.

En el capítulo 2, se describe el estado del arte de las máquinas de soporte vectorial (SVM), mostrando tanto las SVM para clasificación como para regresión, además se realiza una detallada descripción de las máquinas de soporte vectorial de mínimos cuadrados para regresión (LS-SVM), técnica que se utilizan en los modelos se presentarán en el capítulo cuatro.

Luego en el capítulo 3, se presenta el estudio de PSO, partiendo con un marco teórico de la técnica, se muestra el algoritmo tradicional, los parámetros, y se finaliza con algunas variantes que modifican este.

Para finalizar con el estudio, en el capítulo 4, se describen los modelos implementados utilizando LS-SVM y PSO para la búsqueda del parámetro de la regresión de la máquina de soporte vectorial. En la primera sección se muestra los componentes macros que contienen los modelos de manera de dar breves explicaciones de estos. En las dos secciones siguientes se muestra la implementación del modelo LS-SVM y LS-SVM con PSO respectivamente. En la cuarta sección se presentan los resultados del modelo LS-SVM con variantes en la técnica de PSO.

Finalmente, en el capítulo 5 se presentan las conclusiones finales de la investigación, posterior a este capítulo se encuentran los anexos, donde contiene el código fuente de los distintos algoritmos utilizados en el desarrollo de los modelos.

## **2. MÁQUINAS DE SOPORTE VECTORIAL.**

### **2.1. Introducción.**

La Máquina de Vector de Soporte (SVM), es una técnica desarrollada por Vladimir Vapnik, con el objetivo de construir clasificadores para el área de las máquinas de aprendizaje. Sus orígenes comienzan con la teoría estadística del aprendizaje, también denominada como teoría de VC, la cual nos permite escoger un clasificador que minimiza una cota superior por sobre el riesgo (o error de prueba), y entrega como resultado una buena medida para obtener clasificadores que generalizan correctamente los datos no observados. Gracias a esta teoría, se puede observar una primera ventaja que proporciona la SVM, sobre otras técnicas clasificadoras, como las RNA's, que pueden generar modelos que sobre ajustan los datos del problema. Otra de las ventajas fundamentales que tienen las SVM's, es la incorporación de una función de núcleo, que les proporciona la habilidad de construir regiones de decisión no lineales de una manera discriminativa.

En los últimos años, el estudio de estas máquinas se ha incrementado gradualmente, logrando excelentes resultados en aplicaciones, que contemplan una gran variedad de temas como: el reconocimiento del habla [1], estimación de densidades probabilísticas [2], y la predicción de series de tiempo [3]. El lector podrá encontrar un tutorial de SVM, con información más detallada en [4].

Las máquinas vectoriales de soporte, surgieron como algoritmos robustos propuestos para resolver problemas de clasificación, y más tarde se extendieron a la solución de problemas de regresión [8]. La característica fundamental de las SVM, es su capacidad de buscar la solución en un espacio posiblemente diferente al espacio de entrada, lo que proporciona la capacidad de resolver ambos problemas de clasificación y regresión, incorporando datos ruidosos, posiblemente de estadística desconocida, y con presencia de valores atípicos. Estos espacios son conocidos con el nombre de espacios de características, y su incorporación conduce a la utilización de funciones de núcleo o kernel. Éstas funciones de núcleo le han entregado a las SVM's, la capacidad de extender las soluciones a problemas no lineales, preservando las ventajas de las soluciones a problemas lineales.

### **2.2. Marco Conceptual de las SVM.**

El modelo de soporte vectorial consiste en encontrar un hiperplano separador, que logre realizar una clasificación de los datos de entrada del problema en dos regiones, cada una de estas regiones corresponderá a una de las clases definidas. Los hiperplanos ubicados a mayor distancia de los datos de entrada, corresponden a un mejor rendimiento de clasificación, en cambio los hiperplanos que se ubican a una distancia más cercana de los datos de entrada, producen un menor error de clasificación. En consecuencia, un hiperplano de separación ideal, debe maximizar el margen de separación y minimizar el error de

clasificación. Sin embargo, no es posible cumplir con ambos objetivos simultáneamente en todos los casos. Para resolver este problema, se plantea una solución utilizando la técnica de minimización cuadrática convexa. Para los casos en que el número de objetos a clasificar es mayor que el número de atributos de cada objeto, el problema conlleva a una única solución óptima. El hiperplano asociado a esta solución será denominado hiperplano óptimo de separación (HOS).

En el caso anteriormente señalado, se dice que las clases son linealmente separables. Para resolver el problema en los casos en que las clases no son linealmente separables, las SVM's introdujeron la llamada función de núcleo o kernel. Luego esta técnica se extendió a resolver problemas de regresión, estudio que es contemplado en esta investigación.

## 2.3. Clasificación de las SVM

En esta sección se describen dos tipos de clasificación, la Linealmente Separable y la Linealmente no Separable, ambas son apoyadas mediante el desarrollo teórico y algorítmico de técnicas estadísticas y matemáticas que permiten su solución de una forma eficiente.

### 2.3.1. Caso SVM Linealmente Separable.

El problema de clasificación más simple consiste en decidir entre dos clases claramente diferenciables, con el fin de poder determinar una separación mediante un hiperplano. Este problema es denominado como “*clasificación binaria con conjuntos linealmente separables*”, y se define de la siguiente forma.

Sea  $S$  el conjunto de muestras de entrenamiento:

$$S = \{(x_1, y_1), \dots, (x_N, y_N)\} \subseteq (X \times Y)^N \quad (2.1)$$

donde  $N$  es el número de muestras, y  $X$  e  $Y$  denotan los espacios de entrada y salida, respectivamente. Además, los vectores  $x_i \in X \subseteq \mathbb{R}^n$  del espacio de entrada, son denominados “*ejemplos de entrenamiento*”, mientras que los valores  $y_i \in Y = \{1, -1\}$  del espacio de salida se conocen como “*etiquetas*”.

Sea  $f()$  una función real, donde  $f(): X \subseteq \mathbb{R}^n$ , la entrada  $x = [x_1, \dots, x_n]^T$  corresponde a dos clases: la *positiva*, si  $f(x) \geq 0$  y la *negativa*, si  $f(x) < 0$ . La función  $f(x)$  es una función lineal, respecto a  $x \in X$ , y corresponde a la ecuación de un hiperplano:

$$f(x) = \langle w, x \rangle + b \quad (2.2)$$

donde  $\mathbf{w} \in \mathbb{R}^n$  es el vector de los pesos, y el sesgo  $b \in \mathbb{R}$  son los parámetros que controlan la función  $f(\mathbf{x})$ , y se deben calcular a partir de los datos, además, la nomenclatura  $\langle \cdot, \cdot \rangle$  corresponde al operador del producto escalar. La regla de decisión está dada por la función

$$\text{sgn}(f(\mathbf{x})) \quad (2.3)$$

La interpretación geométrica del problema de clasificación binaria es la que considera el espacio de entrada  $\mathbf{X}$  partido en dos por un hiperplano definido por la ecuación

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = 0 \quad (2.4)$$

En la Figura 2 se muestra cómo el hiperplano que se define por una recta, divide el espacio de entrada  $\mathbb{R}^2$  en dos regiones, logrando así una separación del espacio de entrada en dos clases distintas. Sobre esta interpretación, se deben calcular  $n + 1$  parámetros, sabiendo que el vector  $\mathbf{w}$  es perpendicular al hiperplano, y la constante  $b$  mueve al hiperplano, en una dirección paralela.

Para seleccionar el hiperplano separador, se pueden usar diferentes criterios existentes. El criterio utilizado por los algoritmos de SVM para la clasificación, es el criterio de margen máximo. Éste criterio define el margen como la distancia entre el hiperplano y la muestra más cercana  $\mathbf{x}_i$ , teniendo como requisito que la ecuación del hiperplano sea:

$$\mathbf{w}^T \mathbf{x}_i + b = 1 \quad (2.5)$$

En un estudio realizado por N. Cristianini y J. Shawe-Taylor [7], se logró comprobar que la distancia de la muestra y el hiperplano está dada por:

$$d = \frac{1}{\|\mathbf{w}\|} \quad (2.6)$$

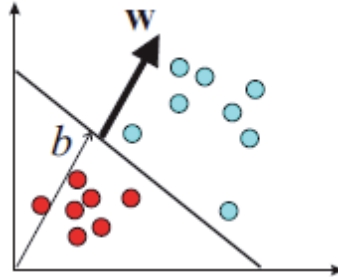
donde  $d$  es la distancia entre la muestra  $\mathbf{x}_i$  y el hiperplano, entonces el problema de maximizar el margen, es equivalente a minimizar la norma  $\|\mathbf{w}\|$ .

Las Figura 3 y 4 muestran dos hiperplanos separadores para un mismo conjunto de muestras. Donde se podrá deducir que existe un infinito número de hiperplanos separadores, para un conjunto de entrenamiento, con muestras que se puedan separar linealmente. El criterio del clasificador de margen máximo, para un número infinito de hiperplanos, determina el hiperplano con margen geométrico máximo, considerando las siguientes restricciones:

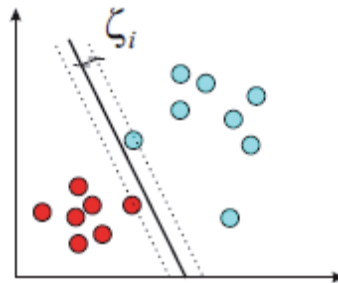


$$y_i [\langle w, x_i \rangle + b] \geq 1 \quad (2.7)$$

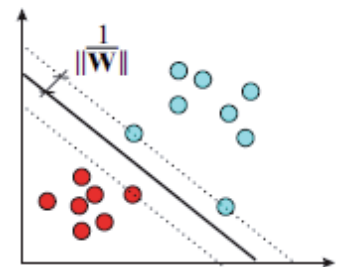
donde  $i = 1, 2, \dots, N$ .



**Figura 2: Hiperplano separador ( $w, b$ )**



**Figura 3: Margen de una muestra al hiperplano**



**Figura 4: Margen máximo del hiperplano**

El problema se puede resolver, como un problema clásico de programación cuadrática (QP), con restricciones de desigualdad, donde se busca minimizar la norma cuadrática del vector lineal del hiperplano separador, sujeto a la restricción (2.7). Sin embargo, este problema de optimización se resuelve usando la técnica de los multiplicadores de Lagrange, de la cual se obtiene la siguiente función de Lagrange:

$$\frac{1}{2} w^T w - \sum_{i=1}^N \alpha_i \{y_i [\langle w, x_i \rangle + b] - 1\} \quad (2.8)$$

donde los  $\alpha_i$  son los multiplicadores de Lagrange correspondientes a la ec. (2.7). Al resolver la ec. (2.8), se obtiene el hiperplano de margen máximo. Éste margen máximo es la base en la cual se apoyan los algoritmos SVM para generar clasificadores más complejos.

### 2.3.2. Caso SVM Linealmente No Separable.

Para tratar con datos que no son linealmente separables, el análisis previo debe ser generalizado introduciendo algunas variables no-negativas  $\xi_i \geq 0$  de tal modo que (2.7) es modificado a:

$$y_i [\langle w, x_i \rangle + b] \geq 1 - \xi_i, \quad i = 1, \dots, N \quad (2.9)$$

Los  $\xi_i \neq 0$  en (2.9) son aquellos para los cuales el punto  $x_i$  no satisface (2.8). Entonces el término  $\sum_{i=1}^N \xi_i$  puede ser tomado como algún tipo de medida del error en la clasificación.

El problema del hiperplano óptimo es entonces redefinido como la solución al problema

$$\min \left\{ \frac{1}{2} w^T \cdot w + C \sum_{i=1}^N \xi_i \right\} \quad (2.10)$$

sujeto a las siguientes restricciones:

$$y_i [\langle w, x_i \rangle + b] \geq 1 - \xi_i, \quad i = 1, \dots, N \quad (2.11)$$

$$\xi_i \geq 0, \quad i = 1, \dots, N \quad (2.12)$$

donde la constante C es definida como un parámetro de regularización. Este es el único parámetro libre de ser ajustado en la formulación de la SVM. El ajuste de éste parámetro hace un balance entre la maximización del margen y la violación a la clasificación.

La búsqueda de un hiperplano óptimo para (2.10), es un problema de programación cuadrática, que puede ser resuelto construyendo una función Lagrangiana y transformándola en un problema dual, de la siguiente forma:

$$\text{Max } W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j z_i z_j \quad (2.13)$$

con las siguientes restricciones

$$\text{Max } W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j z_i z_j \quad (2.14)$$

donde  $\alpha = (\alpha_1, \dots, \alpha_N)$  es un vector de multiplicadores de Lagrange positivos asociados con las constantes en (2.9).

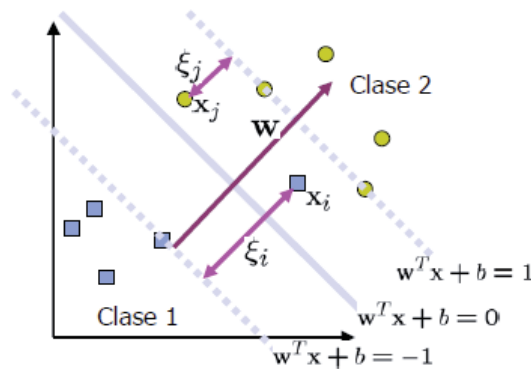
El teorema de Khun-Tucker juega un papel importante en la teoría de las SVM. De acuerdo a este teorema, la solución  $\bar{\alpha}_i$  del problema (2.11) satisface:

$$\bar{\alpha}_i (y_i (\bar{w} \cdot z_i + \bar{b}) - 1 + \bar{\xi}_i) = 0, \quad i = 1, \dots, N \quad (2.15)$$

$$(C - \bar{\alpha}_i) \bar{\xi}_i = 0, \quad i = 1, \dots, N \quad (2.16)$$

De esta igualdad se deduce que los únicos valores  $\bar{\alpha}_i \neq 0$  son aquellos que logran satisfacer la igualdad con el signo, para las constantes en (2.9). El punto  $x_i$  correspondiente con  $\bar{\alpha}_i > 0$  es llamado vector de soporte, los que pueden ser de dos tipos en un caso no separable.

En el caso  $0 < \bar{\alpha}_i < C$ , el correspondiente vector de soporte  $x_i$  satisface las igualdades  $y_i (\bar{w} \cdot z_i + \bar{b}) = 1$  y  $\bar{\xi}_i = 0$ . En el caso  $\bar{\alpha}_i = C$ , el correspondiente  $\bar{\xi}_i$  es diferente de cero y por lo tanto el vector de soporte  $x_i$  no satisface a (2.8). A los vectores que cumplen con este último caso, se les denomina "errores". El punto  $x_i$  correspondiente a  $\bar{\alpha}_i = 0$  es clasificado correctamente y está claramente alejado del margen de decisión.



**Figura 5: Representación gráfica de la interacción entre el parámetro de error  $\xi_i$  en el error de la clasificación**

Para construir el hiperplano óptimo  $\bar{w} \cdot z_i + \bar{b}$ , se utiliza:

$$\bar{w} = \sum_{i=1}^N \bar{\alpha}_i y_i z_i \quad (2.17)$$

y el escalar  $b$  puede ser determinado de las condiciones de Kuhn-Tucker (2.12) y (2.13).

La regla de decisión generalizada de (2.3) y (2.14) son tales que:

$$f(x) = \text{sign}(w \cdot z + b) = \text{sign}(\sum_{i=0}^N \alpha_i y_i z_i \cdot z + b) \quad (2.18)$$

## 2.4. Regresión para Máquinas Vectoriales

Como mencionamos anteriormente, el objetivo principal de las máquinas de vector de soporte consiste en encontrar una función lineal de la forma:

$$f(x) = \langle w, x \rangle + b \quad (2.19)$$

la cual debe ajustarse lo mejor posible a los datos de entrenamiento  $S = \{(x_1, y_1), \dots, (x_N, y_N)\}$ , donde, a diferencia del conjunto (2.1), aquí el espacio de salida  $Y \subseteq \mathbb{R}$ .

El problema de regresión corresponde al ajuste de un hiperplano al conjunto dado de entrenamiento.

Para la optimización de la SVM, se debe minimizar el siguiente término.

$$\text{Min}(\frac{1}{2} \cdot |w|^2) \quad (2.20)$$

sujeto a las siguientes condiciones:

$$y_i - \langle w \cdot x_i \rangle - b \leq \varepsilon \quad (2.21)$$

$$\langle w \cdot x_i \rangle + b - y_i \leq \varepsilon \quad (2.22)$$

Esto se logra con una función  $f(x)$ , la que se puede aproximar mediante el ingreso de todos los pares de entrada, sin embargo, es importante considerar un margen de error, incorporando unas variables de holgura ( $\xi_i, \xi_i^*$ ) cuando los datos se comportan de manera

sobreestimada o subestimada. Con esto, nuestra nueva ecuación a minimizar sería de la siguiente manera:

$$\text{Min} \left( \frac{1}{2} \cdot |w|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \right) \quad (2.23)$$

sujeto a las siguientes restricciones:

$$y_i - \langle w \cdot x_i \rangle - b \leq \varepsilon + \xi_i \quad (2.24)$$

$$\langle w \cdot x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \quad (2.25)$$

$$\xi_i, \xi_i^* \geq 0$$

donde la constante  $c > 0$ , corresponde a un parámetro que logra regularizar el error.

Representando esta ecuación a minimizar, con los multiplicadores de Lagrange ( $\alpha$ ), es equivalente a la siguiente ecuación:

$$L = \frac{1}{2} \cdot |w|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*) - \sum_{i=1}^N \alpha_i \cdot (\varepsilon + \xi_i - y_i + \langle w \cdot x_i \rangle + b) - \sum_{i=1}^N \alpha_i^* \cdot (\varepsilon + \xi_i + y_i - \langle w \cdot x_i \rangle + b) - \sum_{i=1}^N (n_i \xi_i + n_i^* \xi_i^*) \quad (2.26)$$

con las siguientes condiciones necesarias:

$$\vartheta_b L = \sum_{i=1}^N (\alpha_i^* - \alpha_i) = 0 \quad (2.27)$$

$$\vartheta_w L = w - \sum_{i=1}^N (\alpha_i^* - \alpha_i) \cdot x_i = 0 \quad (2.28)$$

$$\vartheta_{\xi_i} L = C - \alpha_i^* - n_i^* = 0 \quad (2.29)$$

$$\vartheta_{\xi_i^*} L = C - \alpha_i - n_i = 0 \quad (2.30)$$

Al sustituir las ecuaciones (2.24 – 2.27) en la ecuación (2.23), llegamos al problema de optimización en su forma dual.

$$\text{Max} \left\{ -\frac{1}{2} \sum_{i,j=0}^N (\alpha_i - \alpha_i^*) \cdot (\alpha_j - \alpha_j^*) \cdot \langle x_i \cdot x_j \rangle - \varepsilon \cdot \sum_{i=0}^N (\alpha_i + \alpha_i^*) + \sum_{i,j=0}^N y_i \cdot (\alpha_i - \alpha_i^*) \right\} \quad (2.31)$$

sujeto a las siguientes restricciones:

$$\sum_{i=1}^N (\alpha_i^* - \alpha_i) = 0 \quad (2.32)$$

$$\sum_{i=1}^N (\alpha_i^* - \alpha_i) \in [0, C] \quad (2.33)$$

El valor estimado de  $w$ , se puede obtener con la ecuación (5.25)

$$w = \sum_{i=0}^N (\alpha_i^* - \alpha_i) \cdot x_i \quad (2.34)$$

Para finalmente la ecuación de regresión lineal es la siguiente:

$$f(x) = \sum_{i=0}^N (\alpha_i^* - \alpha_i) \cdot (x_i \cdot x) + b \quad (2.35)$$

## 2.5. Función de Kernel

En la mayoría de los problemas se tiene, el caso de la clasificación, que las clases no son linealmente separables; y en el caso de la regresión, que no existen hiperplanos lineales que se ajusten de forma apropiada a las observaciones.

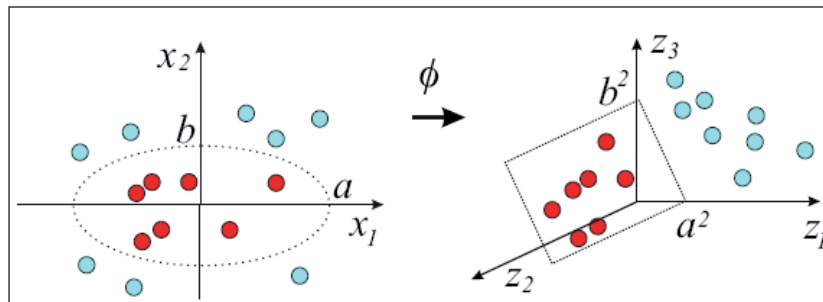
Frecuentemente, las aplicaciones reales de clasificación y regresión requieren hipótesis más complejas que las limitadas a funciones lineales. Una de estas hipótesis se basa en la idea de que los datos del espacio de entrada  $\mathbf{X}$  se puedan representar en un espacio de mayor dimensión (*espacio de características*)  $\mathcal{H}$  mediante una transformación no lineal  $\Phi: \mathbb{R}^N \rightarrow \mathcal{H}$  sobre los datos, es decir:

$$x = (x_1, \dots, x_N) \rightarrow \Phi(x) = (\Phi_1(x), \dots, \Phi_{N_H}(x)) \quad (2.36)$$

donde  $N_H$  es la dimensión del espacio de características  $\mathcal{H}$ , la que puede ser de dimensión infinita.

A modo de ejemplo se considera el problema de separar dos conjuntos de muestras linealmente no separables, tal como se muestra en la Figura 6. Como se puede observar, en el espacio de entrada el problema no es separable, debido a que no existe ningún plano en  $\mathbb{R}^2$  que pueda separar ambas clases. Si se transforman los datos del espacio de entrada utilizando la siguiente transformación no lineal:

$$\Phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) = (z_1, z_2, z_3) \quad (2.37)$$



**Figura 6: Efecto de la traslación  $\Phi(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1^2, \sqrt{2}\mathbf{x}_1\mathbf{x}_2, \mathbf{x}_2^2)$  de un espacio de entrada  $\mathbf{X}$  a un espacio de características  $\mathcal{H}$**

entonces el espacio de entrada en  $\mathbb{R}^2$  se traslada a  $\mathbb{R}^3$ . Si se supone que la frontera de separación en  $\mathbb{R}^2$  de las clases corresponde a una elipse, ésta se transformará en  $\mathbb{R}^3$  en la ecuación de un plano, como se muestra a continuación:

$$\Phi: \left(\frac{x_1}{a}\right)^2 + \left(\frac{x_2}{b}\right)^2 = 1 \rightarrow \frac{z_1}{a^2} + \frac{z_3}{b^2} = 1 \quad (2.38)$$

que corresponde a la ecuación de un plano en  $\mathbb{R}^3$ . Así, el propósito de la transformación  $\Phi(\mathbf{x})$  es *linealizar*, en el espacio de características, las estructuras no lineales. Sin embargo, los espacios de características son más complicados y de mayor dimensión que el espacio de entrada y, además, tienen la dificultad adicional de requerir el desarrollo de la operación producto punto o escalar en ese espacio. Este producto interno se calcula implícitamente con los datos del espacio de entrada, como se muestra a continuación:

$$\begin{aligned} \langle \phi(x_i), \phi(x_j) \rangle &= \langle (x_{i1}, \sqrt{2}x_{i1}x_{i2}, x_{i1}^2), (x_{j1}, \sqrt{2}x_{j1}x_{j2}, x_{j1}^2) \rangle \\ &= (x_{i1}x_{j1} + 2x_{i1}x_{i2}x_{j1}x_{j2} + x_{i1}^2x_{j1}^2) \\ &= \langle x_i, x_j \rangle^2 \\ &= K(x_i, x_j) \end{aligned} \quad (2.39)$$

donde  $K(\mathbf{x}_i, \mathbf{x}_j)$  es la función núcleo. Esta forma de cálculo del producto escalar es muy característica en las máquinas de aprendizaje, y se denomina como el truco del núcleo. Por lo tanto, la función de núcleo permitirá calcular el producto escalar o, mayormente conocido como producto punto.

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = \phi(x_i)^T \phi(x_j) \quad (2.40)$$

El uso de funciones núcleo sobre conjuntos infinitos de datos conduce a la obtención de matrices de tipo Gram como se muestra a continuación:

$$G = K(x_i, x_j) = \begin{bmatrix} K(x_1, x_1) & \dots & K(x_1, x_N) \\ \vdots & \ddots & \vdots \\ K(x_N, x_1) & \dots & K(x_N, x_N) \end{bmatrix} \quad (2.41)$$

las cuales son simétricas, y definidas positivas, es decir, sus valores son mayores a 0.

### 2.5.1. Núcleos de Mercer

Las condiciones de Mercer logran validar la función núcleo, ya que establecen que cada aplicación bivariada  $K(\mathbf{x}_i, \mathbf{x}_j)$  es continua y simétrica, por lo tanto, se puede asegurar la existencia de la aplicación  $\Phi: X \times X \rightarrow \mathcal{H}$  tal que:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = \sum_{n=1}^{\infty} \lambda_n \phi_n(\mathbf{x}_i) \phi_n(\mathbf{x}_j) \quad (2.42)$$

Las condiciones de Mercer contienen implícitamente dos propiedades necesarias que deben cumplir las funciones de núcleo:

1. La función de núcleo de Mercer es siempre simétrica.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = \langle \phi(\mathbf{x}_j), \phi(\mathbf{x}_i) \rangle = K(\mathbf{x}_j, \mathbf{x}_i) \quad (2.43)$$

2. La función de núcleo de Mercer debe cumplir con la desigualdad triangular de Cauchy-Schwartz, definida por:

$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}_j)^2 &= \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle^2 \leq \|\phi(\mathbf{x}_i)\|^2 \|\phi(\mathbf{x}_j)\|^2 \\ &= \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_i) \rangle \langle \phi(\mathbf{x}_j), \phi(\mathbf{x}_j) \rangle = K(\mathbf{x}_i, \mathbf{x}_i) K(\mathbf{x}_j, \mathbf{x}_j) \end{aligned} \quad (2.44)$$

La Tabla 1, presenta un resumen de las funciones núcleos más utilizados.

Función núcleo	Tipo
$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$	Lineal
$K(\mathbf{x}_i, \mathbf{x}_j) = (\langle \mathbf{x}_i, \mathbf{x}_j \rangle + 1)^d$	Polinómico
$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ ^2}{2\sigma^2}}$	Gausiano
$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{ \mathbf{x}_i - \mathbf{x}_j }{\sigma}}$	Laplaciano
$K(\mathbf{x}_i, \mathbf{x}_j) = \text{sinc}((\mathbf{x}_i - \mathbf{x}_j)\sigma)$	Sinc
$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \mu)$	Sigmoidal
$K(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\ \mathbf{x}_i - \mathbf{x}_j\ ^2 - c^2}$	Multicuadrática

**Tabla 1: Ejemplos de funciones núcleo de Mercer**

Como no tenemos conocimiento alguno de la función  $\Phi(\cdot)$ , basta con utilizar alguna función que logre satisfacer el teorema de Mercer, y de esta manera nuestro hiperplano no lineal quedará definido por la siguiente ecuación.

$$\text{Max } W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (2.45)$$

y por consecuencia, nuestra función de decisión clasificatoria, es:



$$f(x) = \text{sign}(w \cdot z + b) = \text{sign}(\sum_{i=0}^N \alpha_i y_i K(x_i, x_j) + b) \quad (2.46)$$

## 2.6. Maquinas vectoriales de Regresión con Mínimos Cuadrados (LS-SVM)

En esta sección se realiza un estudio del Teorema de los Mínimos Cuadrados, detallando sus ventajas y desventajas, para finalizar con el desarrollo matemático y teórico de la Maquina Vectorial de regresión con Mínimos Cuadrados (LS-SVM).

### 2.6.1. Teoría de los Mínimos Cuadrados (LS)

Es un método para determinar la relación funcional entre dos variables  $\mathbf{x}$  e  $\mathbf{y}$ , como resultado de experimentos que tienen una tendencia lineal en una secuencia de puntos.

También se conoce como teorema de Gauss-Márkov, y fue planteado por Carl Friedrich Gauss en 1795, cuando aún tenía 18 años. Este método tuvo buenos resultados en el año 1801, cuando se utilizó para determinar la trayectoria de un pequeño planeta llamado "Ceres", que fue descubierto por Giuseppe Piazzi. Luego de 8 años (1809), se publicó el método de los mínimos cuadrados, en el segundo volumen de Friedrich, titulado "*Theoria Motus Corporum Coelestium in scitionibus conicis solem ambientium*".

Esta técnica logró perfeccionar el método a mano libre o método gráfico, el cual consiste en graficar los datos y dibujar una curva a mano alzada sobre dichos datos, con el propósito de encontrar una relación entre estos. La principal dificultad de este método corresponde a su subjetividad, ya que depende de la persona que realice esta curva en el gráfico, pudiendo existir varias aproximaciones para los mismos datos.

Actualmente se han desarrollado innumerables aplicaciones basadas en la minimización cuadrática en diversos campos que tienen relación con procesamiento de datos estadísticos o experimentales.

Las principales aplicaciones se agrupan en:

- Aproximación de funciones.
- Estimación de parámetros.

La teoría de los mínimos cuadrados, es una técnica de análisis numérico categorizada dentro de la optimización matemática, en la que, dado un conjunto de pares, se intenta encontrar la función que mejor se aproxime a los datos (un "mejor ajuste"), de acuerdo con el criterio del mínimo error cuadrático, teniendo como requisito implícito para que funcione dicho método, que los errores de cada medida estén distribuidos de forma aleatoria.

## 2.6.2. Ventajas del Método de los Mínimos Cuadrados

Esta técnica logró perfeccionar el método a mano libre, el cual era poco preciso e imperfecto. Las ventajas de análisis que incorpora este modelo son las siguientes:

- Es objetivo, sólo depende de los resultados experimentales.
- Es reproducible, proporciona la misma ecuación no importa quién realice el análisis.
- Proporciona una estimación probabilística de la ecuación que representa a unos datos experimentales.

## 2.6.3. Método de los Mínimos Cuadrados en Regresión Lineal

Éste método resuelve problemas bastante frecuentes, en que se tiene un sistema de ecuaciones lineales sobre-determinado, lo que nos indica que este sistema contiene más ecuaciones que incógnitas. Habitualmente, se busca minimizar la suma de los cuadrados de los residuos.

Consideremos un sistema lineal de  $n$  ecuaciones lineales con  $m$  incógnitas.

$$\begin{pmatrix} a_{1,1} & \dots & a_{1,m} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \dots & a_{n,m} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \quad (2.47)$$

donde  $n > m$ , por ser un sistema sobre-determinado.

En notación matricial se tiene:

$$AX = b \quad (2.48)$$

por lo tanto, el vector residuo es:

$$r = AX - b \quad (2.49)$$

donde  $r$  puede ser bastante pequeño con una buena elección de  $X$ , pero en general,  $r$  es distinto de cero.

Se denomina solución mínimo cuadrática a la minimización de la norma euclídea del residuo, quedando el problema planteado de la siguiente forma:

*“Dada la matriz  $A \in \mathbb{R}^{n \times m}$  con  $n > m$  y el vector  $b \in \mathbb{R}^n$ , se quiere calcular el vector  $x \in \mathbb{R}^m$  que minimize  $\|Ax - b\|$ ”.*

## 2.6.4. Rectas de Regresión por mínimos cuadrados

Tenemos un conjunto  $(x_i, y_i)$ , donde  $i = 1, 2, \dots, N$ ;  $x_i$  corresponde a los valores de la Temperatura Superficial del Mar y  $y_i$  representa los diferentes instantes de tiempo en que se registraron estos valores. Se debe ajustar el comportamiento de estos puntos a una recta de la forma:  $y = mx + b$ , con el propósito de hallar el valor de los coeficientes  $m, b$  para resolver el problema. De esta forma, se pretende encontrar la mejor solución al siguiente sistema de ecuaciones:

$$y_1 = mx_1 + b$$

$$y_2 = mx_2 + b$$

⋮

$$y_n = mx_n + b$$

Minimizando el error medio cuadrático:

$$E(m, b) = \sum_{k=1}^n (mx_k + b - y_k)^2 \quad (2.50)$$

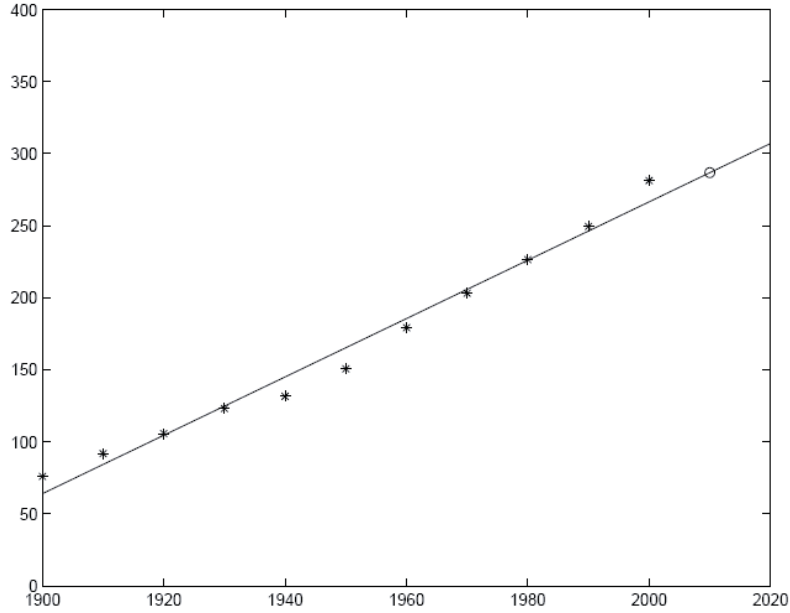
Para resolver se utiliza el Teorema de Gauss, obteniendo la forma matricial  $\mathbf{AX} = \mathbf{B}$ :

$$\begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad (2.51)$$

incorporando las ecuaciones normales de la forma  $\mathbf{A}^T \mathbf{AX} = \mathbf{A}^T \mathbf{B}$ , como se muestra a continuación:

$$\begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & n \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix} \quad (2.52)$$

Las que corresponden a las *Ecuaciones Normales de Gauss*.



**Figura 7: Recta de Regresión**

## 2.7. Modelo LS-SVM

Como mencionamos anteriormente, el mapa de SVM es llevado a un modelo de mayor dimensión, construido mediante la regresión lineal, expresada como:

$$y = f(x) = w^T \cdot \varphi(x) + b \quad (2.53)$$

donde  $\varphi(x): \mathbb{R}^n \rightarrow \mathbb{R}^{n\varphi}$ , es una función que lleva el espacio de entrada a dimensiones superiores conocidas como “espacio de características”, además  $w$  y  $b$  representan los parámetros de regresión que deben ser resueltos. Sin olvidar, que la ecuación está sujeta a las siguientes condiciones:

$$y_i - (w^N \cdot \varphi(x_i) + b) \leq \xi + \xi^* \quad (2.54)$$

$$(w^N \cdot \varphi(x_i) + b) - y_i \leq \xi + \xi^* \quad (2.55)$$

$$\xi_i, \xi_i^* \geq 0, i = 1, \dots, N$$

donde  $N$  corresponde al número de muestras,  $x_i$  es el vector de la  $i$ -ésima muestra que asigna los datos de un mayor espacio tridimensional por el núcleo de la función  $\varphi$ , los vectores  $\xi_i$  representan la forma superior del error y  $\xi_i^*$  es la formación más baja del error sujeto al tubo de sensibilidad  $[Y - (W^N \cdot \varphi(x_i) + b) \leq \varepsilon]$ .

La regresión de soporte vectorial más óptima, se obtiene minimizando la siguiente expresión:

$$\min_{w,b,\epsilon} L_p(w, \epsilon) = \frac{1}{2} \|w\|^2 + \frac{C}{2} \sum_{i=1}^N \epsilon_i \quad (2.56)$$

$$y_i = w \cdot \varphi(x_i) + b + \epsilon_i, i = 1, 2, \dots, N$$

donde el error de las variables de entrada está definido por  $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N)^T, \mathbf{e}_i \in \mathbb{R}$ , donde la constante de regularización  $C$  debe ser mayor que cero. Para resolver esta problemática, se debe utilizar los multiplicadores de Lagrange en la ecuación, de la siguiente forma:

$$L_p(w, b, \epsilon_i, \alpha) = \frac{1}{2} w^T \cdot w + \frac{C}{2} \sum_{i=1}^N \epsilon_i - \sum_{i=0}^N \alpha_i (w^T \cdot \varphi(x_i) + b + \epsilon_i - y_i) \quad (2.57)$$

donde los multiplicadores de Lagrange, se representan en  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N), \alpha_i \in \mathbb{R}$ . La solución es presentada gracias a las ecuaciones de primer orden:

$$\left\{ \begin{array}{l} \frac{\partial L_p}{\partial w} = 0 \rightarrow w = \sum_{i=0}^N \alpha_i \cdot \varphi(x_i) \\ \frac{\partial L_p}{\partial b} = 0 \rightarrow 0 = \sum_{i=1}^N \alpha_i \\ \frac{\partial L_p}{\partial \epsilon_i} = 0 \rightarrow \alpha_i = C \cdot \epsilon_i, i = 1, 2, \dots, l \\ \frac{\partial L_p}{\partial \alpha_i} = 0 \rightarrow \bar{y}_t = w^T \varphi(x_t) + b + \epsilon_i, i = 1, 2, \dots, l \end{array} \right. \quad (2.58)$$

Por lo tanto, la expresión (2.21) se deduce a la siguiente:

$$y_i = \sum_{i=1}^N \alpha_i \cdot \varphi(x_i)^T \cdot \varphi(x_i) + b + \epsilon_i, i = 1, 2, \dots, N \quad (2.59)$$

Incorporando el uso de una función kernel, que cumpla con las condiciones de Mercer, la ecuación resultante sería la siguiente:

$$y_i = \sum_{i=1}^N \alpha_i \cdot k(x_i, x_j) + b + \frac{\alpha_i}{C}, i = 1, 2, \dots, N \quad (2.60)$$

Recordando las ecuaciones de primer orden, donde  $\frac{\partial L_p}{\partial b} = 0 \rightarrow 0 = \sum_{i=1}^N \alpha_i$ , podemos concluir que nuestra ecuación resultante sería:

$$y_i = b + \frac{\alpha_i}{C}, i = 1, 2, \dots, N \quad (2.61)$$

Para determinar los valores de  $\alpha$  y  $b$ , se utiliza el teorema de los Mínimos Cuadrados, para representar el problema en su forma matricial, de la siguiente forma:

$$\begin{bmatrix} 0 & 1^{-T} \\ \bar{1} & K + \frac{1}{C} I \end{bmatrix} \cdot \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \quad (2.62)$$

Teniendo los parámetros de regresión claros,  $\alpha_i$  y  $b$ , logramos resolver la función de regresión dada por:

$$y_i = \sum_{i=1}^N \alpha_i \cdot k(x_i, x_j) + b \quad (2.63)$$

## 3. OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS (PSO).

### 3.1. Introducción.

La optimización por enjambre de partículas, en inglés Particle Swarm Optimization (PSO) es una técnica de cálculo evolutivo, en la optimización demográfica estocástica, y nació a semejanza de otras técnicas de este tipo, por el intento de imitar y mimetizar el comportamiento de algunos procesos naturales. Sus orígenes remontan de los estudios realizados por (Kennedy y Eberhart, 1995), quienes intrigados por la capacidad que tienen los bancos de peces y las bandadas de aves, para separarse, reagruparse y encontrar su alimento, tuvieron la iniciativa de simular de manera gráfica el movimiento sincronizado e impredecible de estos grupos. Incorporando algunos trabajos previos realizados en el ámbito de la biología y la sociología, se concluye que el comportamiento, la inteligencia y el movimiento de estas agrupaciones, está relacionado directamente con la capacidad de compartir información, aprovechando la experiencia acumulada de cada uno de los congéneres. Este comportamiento fue modelado utilizando la matemática, con ecuaciones simples, revelando su potencial como método de optimización. [8]

La “*Vida Artificial*” es la ciencia que realiza el estudio de cómo las técnicas biológicas nos ayudan a resolver diferentes problemas en el área de la informática. Como resultado de este estudio se encuentran, la red neuronal artificial (RNA), el cual corresponde a un modelo simplificado del cerebro humano; y el algoritmo genético (AG), que se inspira en la evolución humana; y la técnica de PSO, la cual asimila como técnica biológica, la conducta esencial de la vida de las bandadas de pájaros y bancos de peces, de manera tal que se pueda utilizar para resolver problemas informáticos. En consecuencia, la técnica de PSO debe cumplir con los siguientes principios básicos de la inteligencia de grupo, estos principios se detallan a continuación:

- **Proximidad:** La población debiera ser capaz de realizar cálculos sencillos de espacio y tiempo. En PSO, esto se puede observar en los movimientos en N dimensiones llevadas a cabo durante una serie de intervalos de tiempo que coinciden con movimientos de la población a una determinada velocidad.
- **Calidad:** Promueve la capacidad de la población para responder a factores de calidad en el espacio de soluciones. Lo cual se puede observar en PSO, con la memoria de la partícula y la historia o conocimiento social que comparten los congéneres entre sí.
- **Diversidad:** Promueve la diversidad de respuesta dentro de la población. Observándolo en PSO, por las diferentes tendencias marcadas en la memoria personal de cada partícula, y por la historia de la mejor posición visitada por todo el conjunto.
- **Estabilidad:** La población sólo cambia su comportamiento como grupo cuando se actualiza la mejor posición históricamente visitada por alguno de los miembros que

lo integran. En PSO no se cumple con este principio, ya que cumple con el principio de adaptatividad, el cual se contrapone con éste.

- **Adaptativa:** La población debe ser capaz de modificar su comportamiento y movimiento cuando hay alguna señal que así lo recomienda desde el punto de vista de ahorro computacional o de mejora en la precisión. En PSO, la población en su conjunto cambia su rumbo cuando alguna de las partículas alcanza una solución global que mejora el resultado.

La técnica PSO tiene muchas similitudes con otras técnicas de cálculo evolutivo, como los Algoritmos Genéticos (GA), y al igual que ellos, el sistema se inicializa con una población de soluciones al azar y la búsqueda de óptimos mediante la actualización de las generaciones. Sin embargo, a diferencia de GA, PSO no tiene operadores como la evolución de cruce y mutación. Adicionalmente se debe destacar la gran simplicidad de programación y el menor número de los que depende el buen funcionamiento de esta técnica, logrando ser una alternativa más óptima y de menor costo, en comparación con GA. Otra técnica que utiliza este método de enjambre, es la Optimización de Colonias de Hormigas, en inglés Ant Colony Optimization (ACO), inspirada en el comportamiento de las hormigas.

PSO, se ha aplicado a investigaciones de diferentes áreas como: optimización de funciones, formación de redes neuronales artificiales, control de sistema de aproximación, y alguna área en que se pueden aplicar Algoritmos Genéticos. Y en cada una de ellas se ha demostrado, que se obtuvo mejor resultado, de una manera más rápida y barata.

### 3.2. Marco Teórico

A modo de ejemplo, se presenta el siguiente escenario para explicar la lógica empleada por esta técnica de PSO. Este escenario consiste en un grupo de aves que de forma aleatoria buscan su alimento en un área determinada. Lamentablemente, No hay más que un trozo de comida en el sector. ¿Cuál es la mejor estrategia para encontrar la comida?, considerando que todas las aves no saben dónde está la comida, pero ellas sí saben a qué distancia de la comida se encuentran en cada iteración. Lo correcto sería, seguir al ave que está más cercana de la comida.

En PSO, cada ave corresponde a una única solución en el espacio de búsqueda del problema, y se denomina partícula. Éstas partículas o posibles soluciones, se “*echan a volar*” por el espacio del problema siguiendo las partículas actuales más óptimas, las que actúan como “*líder*” de la bandada. Cada una de estas partículas realiza un seguimiento de sus coordenadas en el espacio del problema, teniendo en cuenta la información del líder, y se almacena este mejor valor personal, en una variable denominada “*pbest*”, el cual es evaluado por una función de optimización, denominada fitness. También, paralelamente se considera el mejor valor local obtenido por las otras partículas vecinas a cada una, almacenando este mejor valor local en una variable, llamada “*lbest*”. Cuando una partícula atrae a toda la población, el mejor valor global de esta compañía, es asignado a la variable “*gbest*”.



En cada iteración se cambia la velocidad que dirige el vuelo de cada partícula de acuerdo a la ponderación de un término al azar, con distintos números aleatorios que se generan para la aceleración hacia los lugares “*pbest*” y “*gbest*”. En esto consiste la optimización del concepto de enjambre de partículas de optimización (PSO).

Una partícula se define como un vector formado por la velocidad, posición y memoria de ésta. Para comenzar, las partículas en PSO, se inicializan aleatoriamente y optimizan sus valores a medida que aumentan las iteraciones. El enjambre es representado por  $X = (x_{i1}, x_{i2}, \dots, x_{id})$  donde  $i = 1, 2, \dots, n$  corresponde a la  $i$ -ésima partícula y  $d = 1, 2, \dots, D$  corresponde a sus dimensiones.

Como se mencionó en la sección anterior, cada iteración actualiza a cada una de las partículas del enjambre en base a dos valores importantes:

- Mejor posición encontrada por una partícula, conocida como  $P_{best}$ . El  $P_{best}$  de una  $i$ -ésima partícula se representa por  $P_i = (P_{i1}, P_{i2}, \dots, P_{id})$ .
- Mejor posición que ha encontrado el enjambre, conocida como  $P_{gbest}$ .

Ambos valores se actualizan en cada iteración y se utilizan para ajustar la velocidad con la que se mueve una partícula determinada en cada una de las dimensiones. La influencia que tiene la mejor posición personal sobre la velocidad de una partícula se conoce como el factor de cognición y la influencia de la mejor partícula del enjambre se conoce como el componente social.

En el año 2001 (Kennedy, Eberhart y Shi, 2001), se incorpora el factor de inercia  $w$  que logra un balance en la búsqueda global y local de cada partícula. Incluyendo este factor, la velocidad se calcula de la siguiente manera:

$$V_{id}(t+1) = w \cdot v_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) + c_2 r_2 (p_{gd}(t) - x_{id}(t)) \quad (3.1)$$

donde  $r_1$  y  $r_2$  corresponden a valores aleatorios independientes en el intervalo  $[0,1]$ .

Además,  $c_1$  y  $c_2$  corresponden a las variables que controlan la influencia de las componentes cognitiva y social. Éstos los valores  $c_1$  y  $c_2$ , aconsejablemente ambos valores se igualan a 2, pero algunos estudios más exactos indican que debería llevar el valor a 1,49. También se debe tener en cuenta que  $c_1$  y  $c_2$  deben cumplir siguiente restricción  $c_1 + c_2 \leq 4$ .

La nueva posición de la partícula se calcula con la siguiente ecuación:

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (3.2)$$

Para prevenir que las velocidades de las partículas incrementen infinitamente, se incorpora un parámetro limitante  $V_{max}$  que delimita el rango de velocidad entre  $[-V_{max}, V_{max}]$  que puede tomar una partícula.

$$\text{Si } v_{ij} > V_{\max} \text{ o } v_{ij} < -V_{\max}$$

$$v_{ij} = \text{sign}(v_{ij}) \cdot V_{\max}$$

Descrita la matemática de la técnica PSO, se resume el proceso en los siguientes pasos:

1. Se debe definir el tamaño  $n$  del enjambre, inicializando las posiciones de las velocidades de las partículas en un espacio  $D$ -dimensional.
2. Evaluar la función fitness de cada partícula en base a algún criterio de optimización (función de costo, mínimos cuadrados, error cuadrático medio).
3. Comparar el fitness de cada partícula obtenido en el Paso 2 con el fitness de su mejor valor personal. Si el fitness actual es mejor que el histórico, se debe actualizar este valor por el nuevo obtenido.
4. Comparar el fitness actual de cada una de las partículas con el mejor fitness encontrado por el enjambre, si el actual es mejor, entonces se actualiza este mejor valor global con el valor de la posición actual.
5. Ajustar la velocidad y posición de las partículas en base a las ecuaciones de velocidad y posición descrita anteriormente, verificando que las partículas no sobrepasen la velocidad máxima definida  $V_{\max}$ .
6. Volver al paso 2 hasta que se cumplan los criterios de parada.
  - a. Número máximo de iteraciones
  - b. Número de iteraciones sin mejoras
  - c. Error mínimo de la función objetivo

### 3.3. Algoritmo de PSO

INICIO

HACER

Iniciar cada partícula.

FIN\_HACER

MIENTRAS las iteraciones máximas y el criterio de error mínimo no es alcanzado.

HACER

Calcular el valor de fitness, de cada partícula.

SI el valor de fitness es mejor que el pBest anterior.

El valor actual se establece como el nuevo pBest.

FIN\_SI

FIN\_HACER

Almacenar el mejor valor de fitness entre todas las partículas, en el  $gBest$ .

Calcular la velocidad de las partículas según la ec. (3.1)

Calcular la posición de cada partícula según la ec. (3.2)

FIN\_MIENTRAS

FINAL

Las velocidades de las partículas en cada dimensión están sujetas a la velocidad máxima " $V_{max}$ ". Si la suma de las aceleraciones da como resultado, que la velocidad en esa dimensión sea superior a  $V_{max}$ , parámetro especificado por el usuario. Entonces, la velocidad en esa dimensión se limita a  $V_{max}$ .

### 3.4. Observaciones de PSO

La definición de la función fitness cumple un papel importante, ya que debe introducir una métrica adecuada para pesar la bondad de cada partícula, si no lo logra, emergerán las carencias del PSO, las que erróneamente pueden asociarse con el algoritmo.

Si el peso de inercia  $w$  es muy grande, ayuda a la exploración global, sino, facilita la exploración local. Inicialmente  $w$  es una constante, la cual tiene fuerte impacto en el resultado de conseguir soluciones más refinadas. Generalmente  $w$  toma el valor de 1 así disminuyendo a medida que pasan las iteraciones. [10]

El tamaño de la población debe ser seleccionado de manera rigurosa, dado que los valores muy grandes pueden explorar de manera minuciosa el espacio de búsqueda pero costo computación se eleva de manera considerable debido al aumento del número de evaluaciones de la función fitness. Generalmente se utilizan poblaciones ente 10 y 50 o 100 y 200 para problemas completos.

### 3.5. Variantes del Algoritmo PSO

En la actualidad existe una serie de variantes del algoritmo PSO tradicional, las cuales incorporan nuevos parámetros, y métodos de actualización. La mayoría de estas mejoran la solución encontrada, o algún parámetro en interés. Dentro de ellas se encuentra la siguiente, "PSO con parámetros de Tiempo de Vida", lo que consigue es añadir un parámetro llamado tiempo de vida asignado a cada partícula de enjambre, esta decrecerá a medida que existan iteraciones que no se consigan partículas con fitness mejores, lo cual una vez que llega a cero, esta destruirá la partícula peor evaluada del enjambre, para así ser remplazada por una nueva partícula en la vecindad de las restantes. El gran objetivo de esto, es que el Algoritmo PSO disminuya aquellas iteraciones que son ineficientes, de manera de tener mejor convergencia y una solución óptima.

### 3.5.1. PSO Gausiano (GPSO)

El algoritmo de PSO clásico realiza su búsqueda el mejor global y el mejor local de la partícula. La forma en que se realiza la búsqueda, así como la convergencia del enjambre en la zona óptima, depende de cómo se ajustan los parámetros de aceleración y las constantes de inercia. Si estos parámetros no se ajustan correctamente, tendremos graves problemas en la búsqueda y en la convergencia del enjambre. Para corregir estas deficiencias percibidas, algunos autores han introducido las funciones de Gauss para guiar los movimientos de las partículas [11] -[13]. En este enfoque, la inercia constante ya no es necesaria y la aceleración constante se sustituye por números aleatorios con distribución gaussiana [12], [13].

Secrest y Lamont [11] propuso la fórmula de actualización siguiente:

$$|v(t)| = GRand((1 - C_1) \cdot |p_i(t-1) - p_g(t-1)| \text{ cuando } rand > C_1 \quad (3.3)$$

$$|v(t)| = GRand((C_2) \cdot |p_i(t-1) - p_g(t-1)| \text{ cuando } rand \leq C_1 \quad (3.4)$$

$$v(t) = |v(t)| \cdot Rand(\theta) \quad (3.5)$$

Donde:

$|p_i(t-1) - p_g(t-1)|$  : corresponde a la distancia entre el mejor local y el mejor global de la partícula. Cabe destacar que si ambos puntos son iguales, esta expresión corresponde a 1.

$C_1$ : equivale a una constante entre 0 y 1, que determina la “confianza” entre el mejor global y local. El  $C_1$  más largo, representa a la partícula ubicada alrededor de los mejores globales.

$C_2$ : corresponde a una constante entre 0 y 1, que establece el punto entre el mejor global y local, en una desviación estándar entre ambos.

$GRand(y)$ : equivale a un cero. Corresponde al número medio de Gauss, al azar con una desviación estándar de  $y$ .

$rand$  : contiene un número aleatorio entre 0 y 1, con una distribución uniforme.

$Rand(\theta)$ : corresponde a un vector aleatorio, con magnitud igual a 1, y su ángulo se distribuye uniformemente desde 0 a  $2\pi$ .

Teniendo en cuenta esta modificación en el algoritmo de PSO, el área alrededor de los mejores globales y locales es predominantemente buscada. Como el mejor global y el mejor local se acercan, disminuye la desviación estándar y el área que está siendo buscada converge.

Krohling [12], [13] ha propuesto un método diferente para la actualización de la velocidad en cada paso de la iteración, de la siguiente manera:

$$v_i(t) = rand_1 \cdot (p_i - x_i(t-1)) + rand_2 \cdot (p_g - x_i(t-1)) \quad (3.6)$$

Donde  $rand_1$  y  $rand_2$  corresponden a números aleatorios positivos, generados de acuerdo al valor absoluto de la distribución de probabilidad gaussiana, es decir,  $abs[N(0,1)]$ .

Teniendo en cuenta las modificaciones descritas anteriormente, en la fórmula de actualización de la velocidad, los coeficientes de los dos términos  $(p - x)$  se generan automáticamente mediante el uso de una distribución de probabilidad gaussiana. Por lo tanto, no hay necesidad de especificar algún otro parámetro. Además, Krohling afirma que la velocidad máxima  $V_{max}$  ya no es necesaria.

### 3.5.2. PSO Disipativos (DPSO)

DPSO introduce la entropía negativa para estimular el modelo de PSO, la creación de una estructura disipativa que impide el estancamiento prematuro [14], [15]. La entropía negativa introduce un caos adicional a la velocidad de las partículas, de la siguiente forma:

$$If(rand < c_v) Then v_{id} = rand \cdot V_{max,d} \quad (3.7)$$

donde  $rand$  y  $c_v$  son números aleatorios entre 0 y 1. Análogamente, el caos de la ubicación de las partículas está representado por:

$$If(rand < c_l) Then x_{id} = Rand(l_d, u_d) \quad (3.8)$$

donde  $rand$  es un número aleatorio entre 0 y 1, y  $rand(l_d, u_d)$  es otro número al azar con límites predefinidos inferior y superior [14].

El caos introduce la entropía negativa que mantiene el sistema fuera del estado de equilibrio. Entonces, la auto-organización de las estructuras disipativas, junto con las interacciones no lineales inherentes en el enjambre, conducen al desarrollo sostenible de las fluctuaciones [15].

### 3.5.3. PSO con congregación pasiva (PSOPC)

La congregación pasiva es un mecanismo que permite a los animales a agruparse en colonias, ha sido propuesto por He et al. [16] como una alternativa para evitar que el algoritmo de PSO quede atrapado en óptimos locales, además logra mejorar su precisión y velocidad de convergencia. La inclusión de la congregación pasiva modifica la fórmula de la velocidad, de la siguiente forma:

$$v_i(t) = \varphi_{ic} \cdot v_i(t-1) + \varphi_1 \cdot r_1 \cdot (p_i - x_i(t-1)) + \varphi_2 \cdot r_2 \cdot (p_g - x_i(t-1)) + \varphi_3 \cdot r_3 \cdot (X - x_i(t-1)) \quad (3.9)$$

Donde  $y_1$ ,  $y_2$  y  $y_3$  son números aleatorios entre 0 y 1,  $r_3$  es el coeficiente de la congregación pasiva, y  $\mathbf{x}$  es una partícula del enjambre, seleccionada al azar.

Aunque el trabajo presentado por He et al. En [16] no incluye especificaciones para el valor del coeficiente de la congregación pasiva, o cómo afecta al rendimiento del algoritmo. Estos dos aspectos son importantes para una investigación futura.

### 3.5.4. PSO estirado (SPSO)

El principal problema en muchas técnicas de optimización global, es el problema de la convergencia con la presencia de mínimos locales. En estas condiciones, la solución puede caer en los mínimos locales cuando la búsqueda se inicia, provocando un estancamiento. Parsopoulos y Vrahatis [17] presentan un algoritmo de PSO modificado llamado “estiramiento” (SPSO), el cual está orientado a la solución del problema de encontrar todos los mínimos globales.

En este algoritmo, la desviación de la llamada, las técnicas de estiramiento y una técnica de repulsión son incorporadas al PSO original. Las dos primeras técnicas se aplican al concepto de transformación de la función objetivo mediante la incorporación de los puntos ya que se encuentran mínimos. La última (técnica de repulsión) agrega la capacidad para garantizar que todas las partículas no se muevan hacia los mínimos que ya se encontraron [17]. Por lo tanto, el algoritmo propuesto puede evitar las soluciones que ya se encontraron, y por consecuencia, tiene más posibilidades de encontrar la solución óptima global a la función objetivo.

Las transformaciones utilizadas son las transformaciones en dos etapas. Suponiendo que una función  $\mathbf{F}$  de adecuación es elegida para el problema, la fase de primera transformación, logra transformar la función de aptitud  $\mathbf{f}(\mathbf{x})$  original en  $\mathbf{G}(\mathbf{x})$  con  $\mathbf{x}$  la representación de cualquier partícula, lo que elimina todos los mínimos locales que se encuentran sobre  $\mathbf{f}(\bar{\mathbf{x}})$ , donde  $\bar{\mathbf{x}}$  representa un mínimo local detectado.

$$G(x) = f(x) + \gamma_1 ||x - \bar{x}|| \cdot (\text{sgn}(f(x) - f(\bar{x})) + 1) \quad (3.10)$$

La segunda etapa se extiende por sobre la vecindad de  $\bar{\mathbf{x}}$ , ya que asigna valores más altos de la función de los puntos en la zona de arriba.

$$H(x) = G(x) + \gamma_2 \frac{\text{sgn}(f(x) - f(\bar{x})) + 1}{\tanh(\mu(G(x) - G(\bar{x})))} \quad (3.11)$$

En ambas ecuaciones,  $\gamma_1$ ,  $\gamma_2$  y  $\mu$  se eligen arbitrariamente como constantes positivas, y  $\text{sgn}(\mathbf{y})$  es una función triple, con valores de signo.

$$\text{sgn}(y) = \begin{cases} 1, & \text{if } y > 0 \\ 0, & \text{if } y = 0 \\ -1, & \text{if } y < 0 \end{cases} \quad (3.12)$$

Ninguna de las etapas modifica los mínimos locales situados por debajo de  $\bar{x}$ , por lo tanto la ubicación del mínimo global no se modifica [17].

## 4. METODOLOGÍA DE IMPLEMENTACIÓN DE LOS MODELOS SVR CON PSO

### 4.1. Introducción

A continuación, se describe el modelo utilizado, para el pronóstico de la Temperatura Superficial del Mar, en las costas del norte de Chile, con el objetivo de determinar las variaciones irregulares que son producto de corrientes cálidas provenientes del Ecuador, debido al fenómeno climatológico llamado “El Niño”.

El modelo LS-SVM (Least Squares Support Vector Machines), será utilizado para la regresión de soporte vectorial, siendo de real importancia para el estudio, la optimización de los parámetros  $C$  y el parámetro asociado al '*kernel gaussiano*'. Con estos parámetros la LS-SVM nos facilitará los multiplicadores de Lagrange y el parámetro  $b$ , relacionados a la función de regresión.

Debido al alto costo computacional que exigen las Máquinas Vectoriales, se utilizó en este estudio una variante de esta técnica, la LS-SVM, que incluye la teoría de los Mínimos Cuadrados simplificando en 2 los parámetros a optimizar.

Es importante mencionar que el cálculo de los multiplicadores de *Lagrange*, ya no es un desafío, debido a que existen variadas técnicas que nos permiten obtener su valor.

### 4.2. Descripción de los Datos de Estudio

Los datos de estudio, corresponden a la Temperatura Superficial del Mar, desde los años 1951, hasta 2005, extraídos de los registros almacenados en el CENDHOC (Centro de Datos Hidrográficos y Oceanográficos de Chile).

Debido a la gran cantidad de meses en que no se realizó la medición de la Temperatura de estas aguas, los datos correspondientes a estos meses particularmente se registraron con un valor de 99,9°, los que se denominan “datos sucios”. Con el propósito de no considerar estos datos en el estudio, se realizó una limpieza, con lo cual se determinó una data de estudio de 210 datos mensuales, lo que corresponde a cerca de 17 años y 6 meses en los cuales se realizó una medición correcta de la Temperatura Superficial del Mar, en la costa de Arica.



Estos 210 datos de estudio se clasificaron de la siguiente manera: 180 Datos de Entrenamiento, lo que corresponde a 15 años; y 30 Datos de Testing, lo que corresponde a cerca 2,5 años.

*Datos de Entrenamiento = 180;*

*Datos de Testing = 30;*

### 4.3. Componentes del Modelo

A continuación, se describen los componentes del modelo implementado.

#### 4.3.1. Framework LS-SVM

Corresponde al principal componente funcional del modelo, ya que nos permite obtener, de manera automática, los multiplicadores de *Lagrange* y el valor de sesgo *b*.

La técnica de LS-SVM está montada en Matlab, siendo ésta, una herramienta matemática cuya tecnología pertenece a “MATHWORKS”, y en comparación a otras técnicas como SMO, logra un resultado bastante óptimo y a muy bajo costo.

Los métodos utilizados por éste Framework son los siguientes.

- **trainlssvm:** Éste método nos entrega de manera óptima, los multiplicadores de Lagrange y el parámetro *b*, utilizados en el modelo LS-SVM.
- **simlssvm:** Éste método nos entrega el vector de predicción, utilizando la data de entrada con un desfase determinado.
- **plotslssvm:** Éste método nos ilustra el comportamiento de la curva con la data de entrenamiento.

#### 4.3.2. Parámetro de la LS-SVM

Los parámetros de entrada de la LS-SVM, corresponden a *C* y los parámetros del kernel, que pueden variar dependiendo de la función utilizada. Es de esencial importancia reconocer la obtención de estos parámetros para que puedan interactuar de mejor forma al modelo de LS-SVM.

- **Parámetro C:** se obtiene mediante la técnica de PSO, y representa el parámetro de regularización de la SVR. Este parámetro contribuye con el balance entre la maximización del margen y la optimización de la función de predicción.

- **Parámetros del Kernel:** estos parámetros también se obtienen utilizando la técnica de PSO. Aunque se pueden utilizar otras funciones de kernel, en este estudio se utilizó el kernel Gaussiano, y le corresponde como parámetro, la varianza que puede tomar valores entre 0 y 100. El aporte de este parámetro a la regresión es lograr un equilibrio entre los datos sobreestimados y los datos subestimados de la curva.

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}} \quad (4.1)$$

- **Multiplicadores de Lagrange:** los multiplicadores son obtenidos gracias al método `trainlssvm`, perteneciente a nuestro framework LS-SVM. Estos valores corresponden al multiplicador asociado a la variable de entrada. Cabe destacar que actualmente existe una gran cantidad de técnicas para obtener estos valores, por lo que su cálculo no es de materia de estudio.
- **Parámetro  $W$ :** este parámetro es calculado gracias a los multiplicadores de Lagrange y corresponde al vector de los pesos de la máquina.
- **Parámetro  $b$ :** parámetro que se obtiene como resultado de la LS-SVM, y pertenece al sesgo de la regresión. Su representación gráfica corresponde a la distancia entre el origen hasta el hiperplano.

### 4.3.3. Datos de Entrada

En la implementación de la LS-SVM se utilizó la librería `LS-SVMlab1.5`, diseñada para Matlab, la cual entrega las herramientas necesarias para implementar la máquina LS-SVM.

Los Datos de Entrenamiento, se ingresan a la máquina con un desfase de 8, y utilizando dos matrices denominadas Data de Entrada y Data de Salida, ambas utilizadas para el Entrenamiento.

Según lo anterior, la Data de Entrada corresponde a una matriz  $X$  de  $(N-8) \times 8$ , la cual se llena como se muestra a continuación. Si nuestra Data de Entrenamiento corresponde a `Entrenamiento[] = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]`, la matriz  $X$  corresponde a:

$$X[] = \begin{vmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \end{vmatrix}$$

$$\begin{vmatrix} 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\ 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \end{vmatrix}$$

La Data de Salida corresponde a la matriz Y de (N-8) x 1, la que se representa de la siguiente manera:

$$Y[] = \begin{vmatrix} 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \end{vmatrix}$$

Los Datos de Testing, se ingresan de la misma manera en que se ingresaron los Datos de Entrenamiento, y estas nuevas matrices se denominan Xt de (N-8) x 8 y Yt de (N-8) x 1, recordando que en este caso N corresponde a 30.

#### 4.3.4. Tecnología del Modelo

Este modelo es desarrollado bajo 3 tecnologías como se menciona a continuación:

- **Matlab:** Facilitando una gran cantidad de cálculos.
- **JAVA EE:** Lenguaje de programación portable, potente y orientado a objetos. Se utilizó para desarrollar la lógica de la LS-SVM y PSO.
- **MySql:** Base de Datos utilizada para almacenar los datos de la investigación.

#### 4.3.5. Medida de Error

Las medidas de error se utilizan para representar numéricamente el comportamiento de los resultados del modelo, con el objetivo de mejorar las predicciones realizadas con los datos de entrenamiento.

En la etapa de entrenamiento se utiliza el RMSE (Root Mean Square Error) como medida de error, evaluando cada una de las partículas, mientras el Algoritmo PSO continúa

iterando. Esta medida corresponde a la suma de las diferencias observadas entre los datos observados y los datos proyectados por el modelo.

Por el contrario, en la etapa de testing se utilizan dos medidas de error, la primera es el coeficiente de determinación ( $r^2$ ), que mide la dependencia entre los datos reales y los datos pronosticados, entre 0 y 1 donde 0 representa independencia y 1 corresponde a la dependencia; y la segunda medida utilizada es el MAPE (Mean Absolute Percentage Error), indicando el tamaño de los errores de pronóstico comparados con los valores reales de la serie.

Siendo  $y_i$  la data ideal,  $y'$  la data pronosticada y  $\bar{y}$  la media aritmética, las medidas de error se representan de la siguiente manera:

- Porcentaje de Error Medio Absoluto (MAPE):

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{y_i - y'}{y_i} \right| \quad (4.2)$$

- Raíz del Error Cuadrático Medio (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (y_i - y')^2}{N}} \quad (4.3)$$

- Coeficiente de Determinación ( $R^2$ ):

$$r^2 = 1 - \frac{\sum_{i=1}^N \sqrt{(y_i - y')^2}}{\sum_{i=1}^N \sqrt{(y_i - \bar{y})^2}} \quad (4.4)$$

#### 4.4. Implementación del modelo LS-SVM.

Como fue descrita en la sección 6 de este documento, la implementación de una maquina vectorial para regresión con mínimos cuadrados, se explica por medio del modelo de clasificación. La regresión pretende ocupar la misma ecuación del hiperplano optimizador de manera de convertirla en una ecuación no lineal. Esta ecuación regresión está definida por:

$$y_i = \sum_{i=1}^N \alpha_i \cdot k(x_i, x_j) + b \quad (4.5)$$

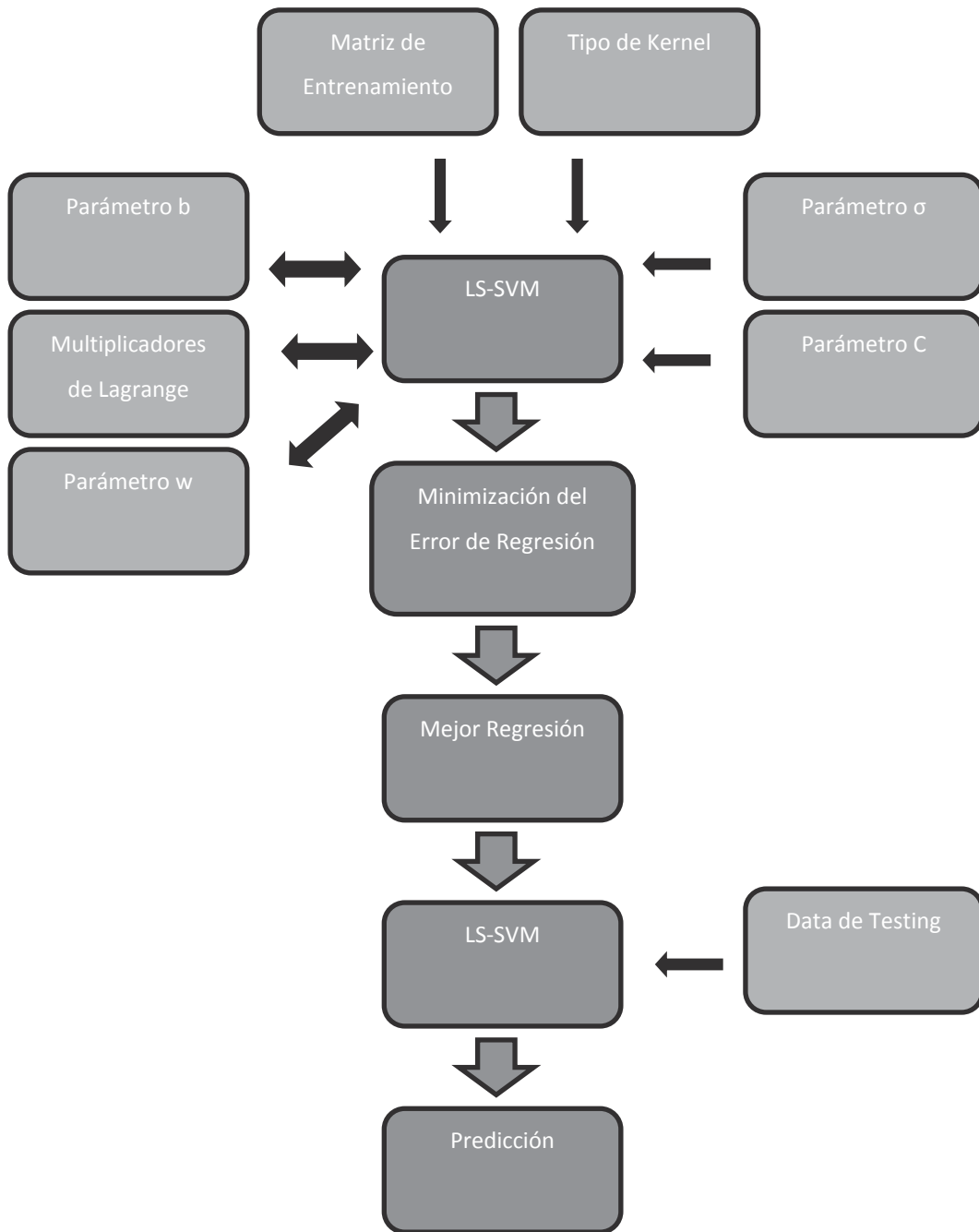
donde  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N)$ ,  $\alpha_i \in \mathbb{R}$  corresponde a los Multiplicadores de Lagrange,  $k(x_i, x_j)$  representa a la función Kernel que cumple con las condiciones de Mercer, y  $b$  que corresponde al sesgo.

Para resolver esta ecuación de regresión, se debe determinar el valor de los parámetros de regresión  $\alpha_i$  y  $\mathbf{b}$ . Esto se logra utilizando el teorema de los mínimos cuadrados, para representar el problema en su forma matricial:

$$\begin{bmatrix} 0 & \mathbf{1}^{-T} \\ \bar{\mathbf{1}} & \mathbf{K} + \frac{1}{c}\mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{b} \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix} \quad (4.6)$$

Es importante mencionar que la función de Kernel será la función Gaussiana (ecuación 4.4). Esta función contiene un parámetro  $\sigma$ . Este parámetro  $\sigma$ , junto al parámetro  $\mathbf{C}$  que se observa en la representación matricial del problema, ecuación 4.4, serán ingresados por el usuario, con el propósito de conseguir un mayor coeficiente de determinación.

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \quad (4.7)$$



**Figura 8: Diagrama del modelo de predicción LS-SVM**

En la LS-SVM se ingresan solamente las matrices  $X$ ,  $Y$  y  $X_t$ , junto a los parámetros de entrada, definidos anteriormente, teniendo como resultado la matriz  $Y_t$ , que corresponde a los datos pronosticados por el modelo.

Considerando lo siguiente:

- Datos de Entrenamiento = 180.
- Datos de Testing = 30.
- Función de Núcleo = *Kernel Gaussiano*.

Se tienen los siguientes resultados:

#### 4.4.1. Comportamiento del Desfase.

En el presente caso de pruebas, se mantuvo constante los valores de C y  $\sigma$ , logrando los siguientes resultados:

Variación de Desfase								
Iteración	desfase	C	sig2	r2	MAPE	RMSE	MSE	PI
1	3	1000	0,2	0,22963	7,505031	0,075682	0,005728	0,797407
2	4	1000	0,2	0,220038	6,24202	0,075454	0,005693	0,837974
3	5	1000	0,2	0,190877	5,549047	0,069693	0,004857	0,871338
4	6	1000	0,2	0,156951	5,498126	0,059253	0,003511	0,905327
5	7	1000	0,2	0,174691	5,079064	0,045659	0,002085	0,934644
6	8	1000	0,2	0,108662	5,070579	0,041225	0,001699	0,936073

**Tabla 2: Resultados de la implementación LS-SVM con variación de desfase**

Al observar los resultados, se reconoció que al aumentar el número de desfase se mejoró la persistencia del modelo, pero su confianza fue bastante perjudicada.

En el gráfico se puede observar que al inicio los datos presentan una gran diferencia entre lo real y lo pronosticado, sin embargo esta diferencia disminuye al observar los siguientes datos pronosticados.

El mejor resultado obtenido se representa a continuación:

Desfase = 3.

C = 1000.

$\sigma = 0,2$ .

Coefficiente de determinación ( $r^2$ ) = 0,22963022183375903



**Figura 9: Gráfico de resultados de LS-SVM**

#### 4.4.2. Comportamiento del Parámetro C.

En el presente caso de pruebas, se mantuvo constante el desfase y  $\sigma$ , logrando los siguientes resultados:

Variación del parámetro de regularización								
Iteración	desfase	C	sig2	r2	MAPE	RMSE	MSE	PI
1	8	5	0,2	0,08883	5,14458	0,041681	0,001737	0,93465
2	8	10	0,2	0,098163	5,110337	0,041467	0,001719	0,93532
3	8	50	0,2	0,106539	5,078733	0,041274	0,001704	0,93592
4	8	100	0,2	0,10765	5,074475	0,041248	0,001701	0,936
5	8	500	0,2	0,108549	5,071015	0,041227	0,0017	0,936065
6	8	1000	0,2	0,108662	5,070579	0,041225	0,001699	0,936073

**Tabla 3: Resultados de la implementación LS-SVM con variación de C**

Al observar los resultados, se reconoció que al aumentar el valor del parámetro de regularización C, se mejoró discretamente la persistencia del modelo, su confianza y disminuyó levemente las diferencias entre los valores pronosticados con los valores reales. El resultado de esta prueba, fue bastante satisfactorio, ya que se observó una mejora en los resultados obtenidos.

En el gráfico se puede observar una constante diferencia entre los datos pronosticados y los reales, diferencia que presentó un comportamiento creciente.



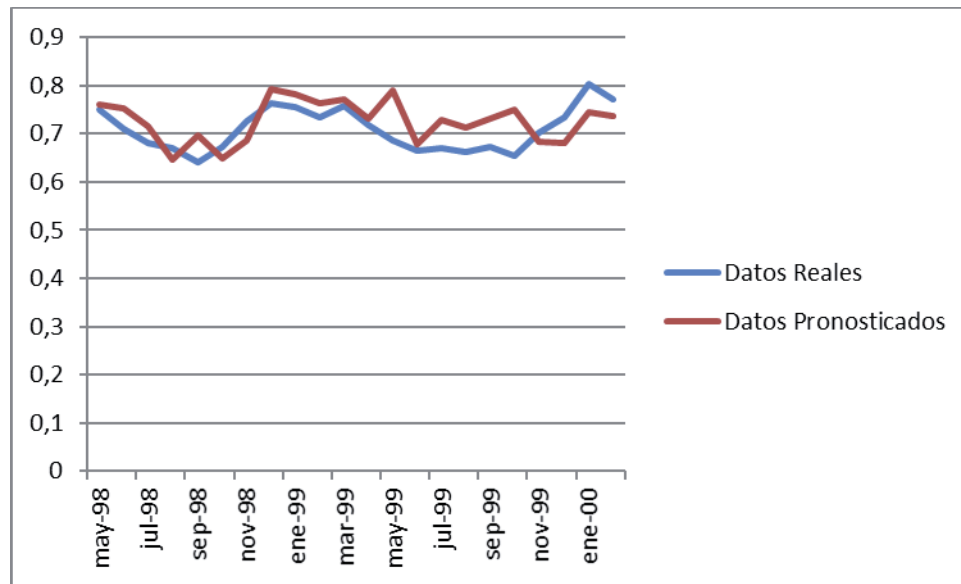
El mejor resultado obtenido se representa a continuación:

Desfase = 8

C = 1000

$\sigma = 0,2$

Coefficiente de determinación ( $r^2$ ) = 0.10866189765493528



**Figura 10: Gráfico de resultados de implementación LS-SVM**

#### 4.4.3. Comportamiento de la varianza.

En el presente caso de pruebas, se mantuvo constante el desfase y C, logrando los siguientes resultados:

Variación de la Varianza								
Iteración	desfase	C	sig2	r2	MAPE	RMSE	MSE	PI
1	8	1000	0,2	0,108662	5,070579	0,041225	0,001699	0,936073
2	8	1000	0,5	<b>0,39986</b>	<b>3,861561</b>	<b>0,033827</b>	<b>0,001144</b>	<b>0,956958</b>
3	8	1000	0,8	0,376299	3,90615	0,034484	0,001189	0,955268
4	8	1000	1	0,351175	3,957004	0,035172	0,001237	0,953466
5	8	1000	1,5	0,267914	4,323305	0,037361	0,001396	0,947494
6	8	1000	2	0,176364	4,703312	0,039628	0,00157	0,940928

**Tabla 4: Resultados de la implementación LS-SVM con variación de varianza**

Al observar los resultados, se reconoció que al aumentar el valor del parámetro del kernel  $\sigma$ , se produjo un aumento y una posterior disminución en la confianza del modelo, afectando de la misma manera a la persistencia de éste. Por el contrario las diferencias entre los datos pronosticados y los datos reales, presentaron una disminución y posterior aumento.

Este comportamiento representa una distribución de Gauss, la cual gráficamente se comporta como una campana simétrica.

En el gráfico se puede observar una constante diferencia entre los datos pronosticados y los reales, diferencia que presentó un comportamiento creciente.

El mejor resultado obtenido se representa a continuación:

Desfase = 8

$C = 1000$

$\sigma = 0,5$

Coefficiente de determinación ( $r^2$ ) = 0,39985951283369725

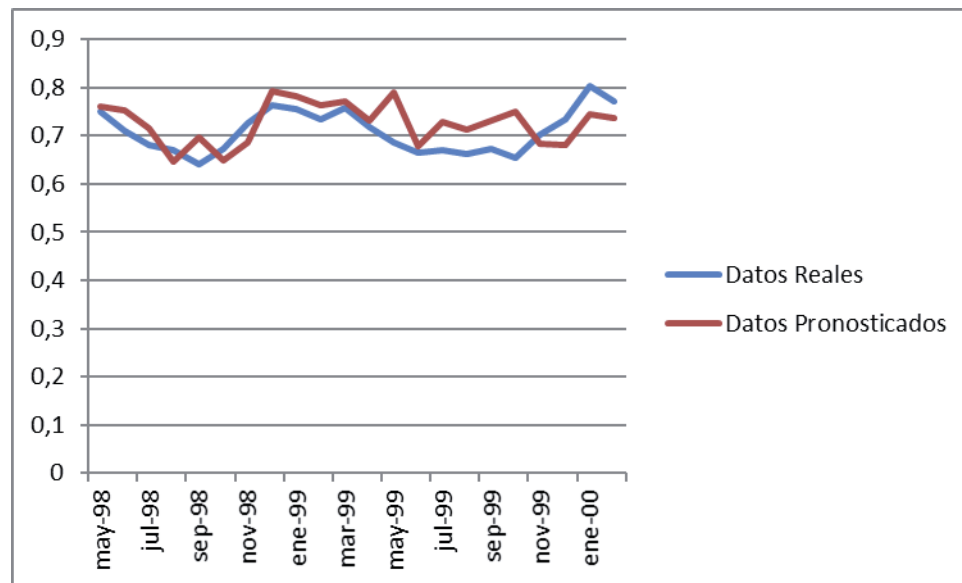


Figura 11: Gráfico de resultados de implementación de LS-SVM

#### 4.4.4. Análisis de los Resultados

Como se pudo observar en los resultados obtenidos utilizando la LS-SVM, la confianza del modelo varía bastante con diferentes valores de  $\sigma$ ,  $C$  y el desfase ingresados.

Es por esto que resulta bastante necesario encontrar los valores óptimos de estos parámetros para que el modelo LS-SVM resulte más confiable.

El aumento del desfase mejora la persistencia del modelo, pero reduce su confianza. En cambio, al ajustar los valores  $C$  y  $\sigma$ , logramos mejorar la persistencia y la confianza del modelo, mejorando significativamente la predicción y disminuyendo los errores observados al comparar la data pronosticada con la data real.

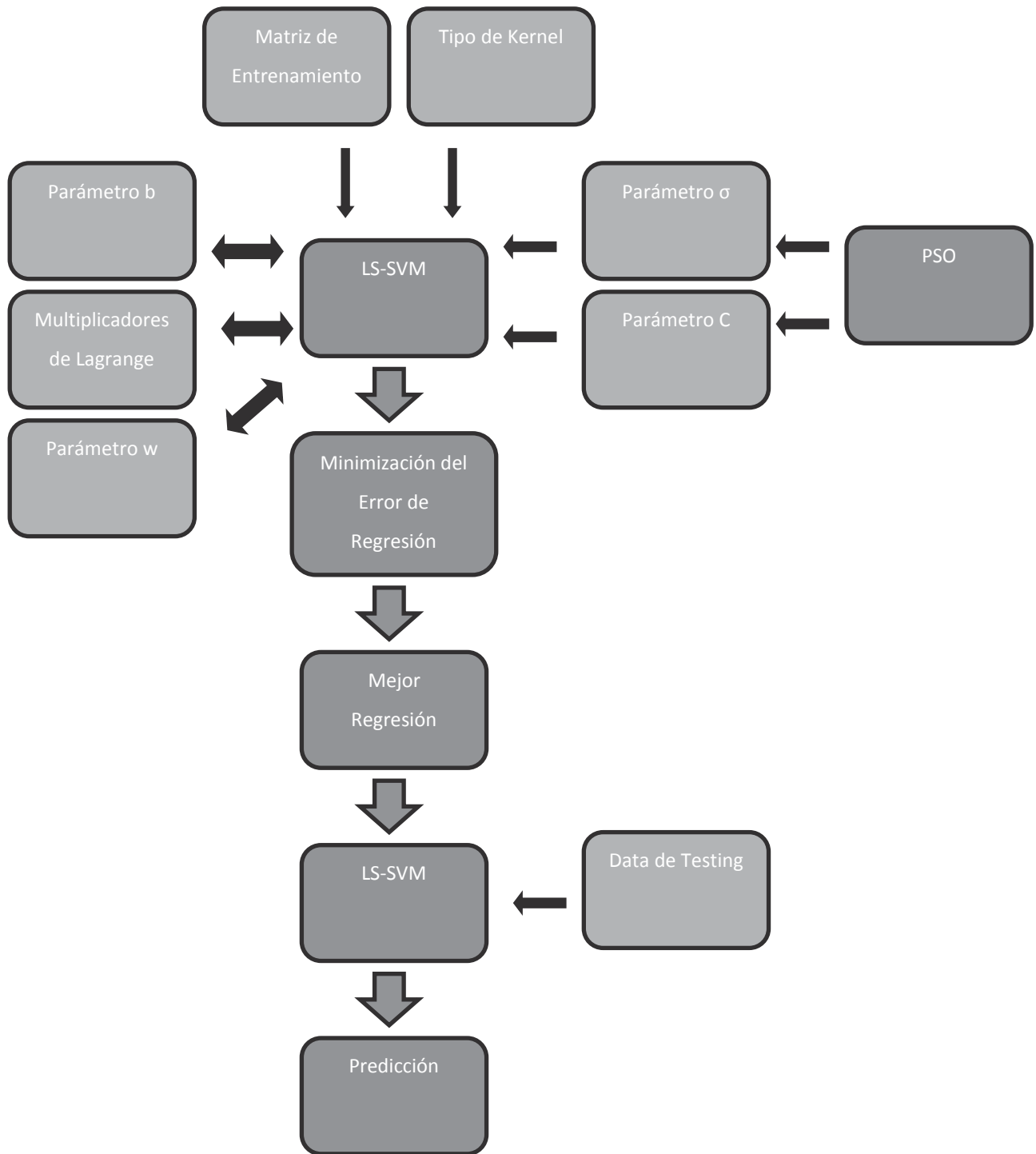
Este problema de optimización de los datos de entrada se resuelve incorporando la técnica de PSO, la cual nos permitirá encontrar la partícula más óptima, que contenga los valores de  $\sigma$  y  $C$ , necesarios para el modelo.

El mejor resultado se obtuvo con un desfase de 8,  $C = 1000$  y  $\sigma = 0,5$ , logrando una predicción de un 39,9% de exactitud, y una persistencia de un 95,6%.

#### **4.5. Implementación del modelo LS-SVM con PSO.**

La técnica de PSO fue descrita en la sección 7, y corresponde a un importante apoyo para la LS-SVM. Gracias a ella, se podrán optimizar los valores del parámetro de regularización  $C$ , y el parámetro del Kernel  $\sigma$ , logrando una mejora significativa de la confianza del modelo de predicción.

A continuación, se describirá el modelo utilizado, y adicionalmente se detallarán algunas variantes del modelo PSO, con el propósito de mejorar la precisión de nuestra predicción.



**Figura 12: Diagrama del modelo de predicción LS-SVM con PSO**

### 4.5.1. Descripción del Enjambre

Se trabaja con una cantidad de 50 individuos, cada individuo es denominado como partícula y se representa por un vector de largo dos, donde se almacena la información del parámetro de regularización  $C$  en la posición 0, y el parámetro de la función kernel es representado en la posición 1, de la siguiente manera:

$C$	$\sigma$
-----	----------

**Tabla 5: Representación de la posición de una partícula para el modelo LS-SVM con PSO**

### 4.5.2. Composición de la Partícula

Todas las partículas que pertenecen al enjambre, contienen variados parámetros, los cuales son:

- Vector Posición: Almacena los parámetros  $C$  y el parámetro correspondiente al Kernel usado.
- Vector Velocidad: Almacena las velocidades con respecto a la posición, ésta se determinada mediante las ecuaciones de velocidad y posición.
- La mejor Partícula con respecto a la función Fitness (Medidas de Error) será la mejor posición y guía del enjambre.

La posición máxima corresponde a aquella partícula que logra regular el límite máximo por medio de un parámetro.

### 4.5.3. Resultado del PSO – Vector GBest

Al finalizar todas las iteraciones del algoritmo de PSO, el vector GBest correspondiente al enjambre, contendrá los parámetros optimizados que luego deberán ser evaluados en la regresión LS-SVM para obtener el pronóstico en estudio.

### 4.5.4. Cantidad de Iteraciones

El entrenamiento a través de la técnica de Optimización por Enjambre de Partículas (PSO) necesita un criterio de manera que finalice la cantidad de iteraciones en el recorrido de las partículas, es por ello que se fijaron 100 iteraciones. Estas 100, se justifican debido a que el modelo es costoso, pero también la máquina tiene una tasa más rápida de optimización.

#### 4.5.5. Comportamiento de desfase en modelo LS-SVM-PSO

En el presente caso de pruebas, se mantuvo constante el número de iteraciones y la cantidad de individuos, logrando los siguientes resultados:

LS-SVM con PSO (Variación en Desfase)										
Itera.	desfase	n. Itera.	partículas	C	sig2	r2	MAPE	RMSE	MSE	PI
1	3	20	40	17,1058	16,11306	0,79567	3,77506	0,038977	0,001519	0,94627
2	4	20	40	67,3638	35,35953	0,84619	3,43299	0,033507	0,001123	0,96805
<b>3</b>	<b>5</b>	<b>20</b>	<b>40</b>	<b>10,3431</b>	<b>34,17454</b>	<b>0,8715</b>	<b>3,2251</b>	<b>0,027774</b>	<b>7,71E-04</b>	<b>0,97957</b>
4	6	20	40	772,087	4,37731	0,73153	3,95791	0,033437	1,12E-03	0,96985
5	7	20	40	502,875	6,873035	0,49671	4,00297	0,035656	1,27E-03	0,96014
6	8	20	40	15,6345	77,32253	0,64389	2,98694	0,026057	6,79E-04	0,97446

**Tabla 6: Resultados de la implementación LS-SVM-PSO con variación de desfase**

Al observar los resultados, se reconoció que al aumentar el desfase, se produjo un aumento y una posterior disminución en la confianza del modelo, afectando de la misma manera a la persistencia de éste. Por el contrario, las diferencias entre los datos pronosticados y los datos reales, presentaron una disminución y posterior aumento.

Este comportamiento se representa gráficamente como una campana simétrica, al igual que una distribución de Gauss.

El mejor resultado obtenido se representa a continuación:

Desfase = 5

Nº de iteraciones = 20

Cantidad de individuos = 40

Coefficiente de Determinación ( $r^2$ ): 0.8491887766949304

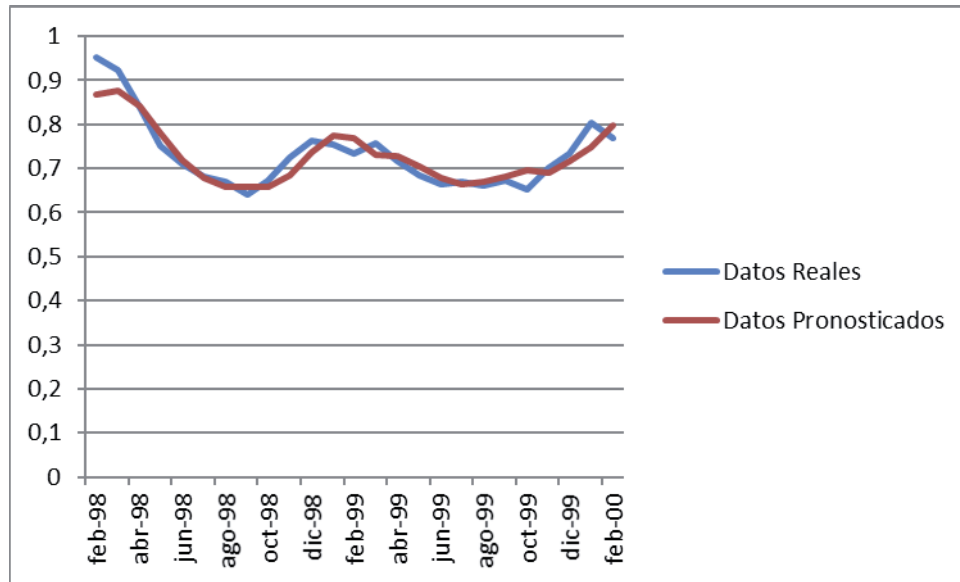


Figura 13: Gráfico de data pronosticada utilizando el modelo de LS-SVM con PSO

#### 4.5.6. Comportamiento de individuos en modelo LS-SVM-PSO

En el presente caso de pruebas, se mantuvo constante el desfase y la cantidad de iteraciones, logrando los siguientes resultados:

LS-SVM con PSO (Variación en N° de individuos)										
Itera.	desfase	n. Itera.	partículas	C	sig2	r2	MAPE	RMSE	MSE	PI
1	5	100	10	161,483	2,485762	0,69341	4,5918	0,0429	1,84E-03	0,95125
2	5	100	20	180,167	2,478068	0,68603	4,64439	0,043413	1,88E-03	0,95007
<b>3</b>	<b>5</b>	<b>100</b>	<b>40</b>	<b>53,5238</b>	<b>67,45512</b>	<b>0,88405</b>	<b>3,00895</b>	<b>0,026383</b>	<b>6,96E-04</b>	<b>0,98156</b>
4	5	100	80	1,41536	62,22975	0,85063	3,11549	0,029945	8,97E-04	0,97625
5	5	100	100	6,20223	98,19823	0,88201	2,99534	0,026613	7,08E-04	0,98124

Tabla 7: Resultados de la implementación LS-SVM con variación de individuos

Al observar los resultados, se reconoció que, al aumentar el número de individuos, se produjo un aumento en la confianza del modelo, como a su persistencia y a los errores observados entre los datos pronosticados, con los datos reales.

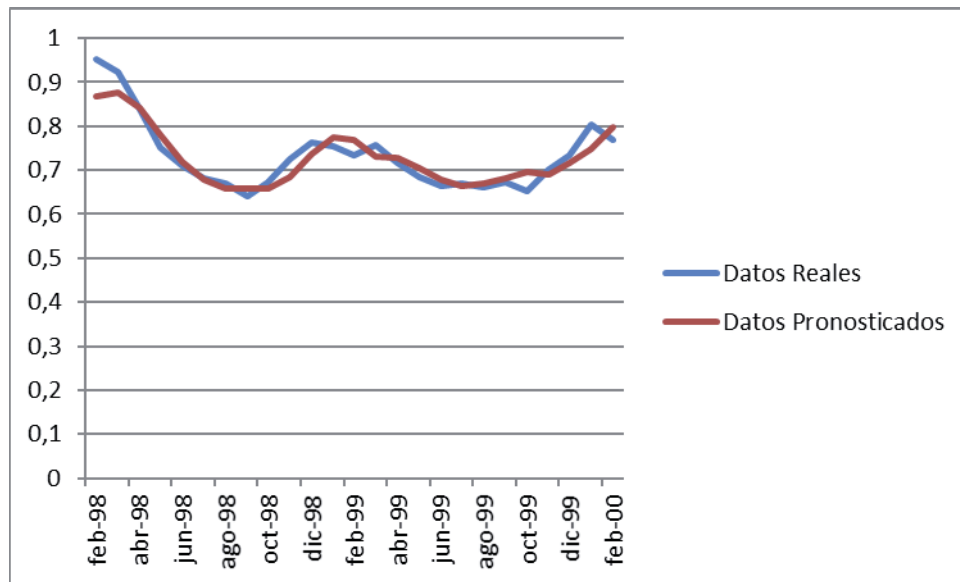
El mejor resultado obtenido se representa a continuación:

Desfase = 5

N° de iteraciones = 100

Cantidad de individuos = 40

Coefficiente de Determinación ( $r^2$ ): 0,8873432327664945



**Figura 14: Gráfico de data pronosticada utilizando el modelo de LS-SVM con PSO**

#### **4.6. Análisis de los resultados para LS-SVM-PSO.**

Como se pudo observar en los resultados obtenidos utilizando con el modelo LS-SVM-PSO, la confianza del modelo varía bastante con diferentes valores de  $\sigma$ , C y el desfase ingresados. En comparación con el modelo LS-SVM, se reconoció un aumento significativo en la confianza y persistencia del modelo, mejorando significativamente la predicción y disminuyendo los errores observados al comparar la data pronosticada con la data real.

En los gráficos se pudo observar en un comienzo una gran diferencia entre lo real con lo pronosticado, sin embargo, esta diferencia disminuyó rápidamente manteniendo un comportamiento bastante cercano a la data real.

El mejor resultado se obtuvo con un desfase de 5, 100 iteraciones y una cantidad de individuos igual a 40, logrando una predicción de un 88,78% de exactitud, y una persistencia de un 98,21%.

#### **4.7. Resultado del modelo LS-SVM con GPSO**

Con un desfase de 5, en la matriz de entrada a la LS-SVM, los resultados son:

$$C = 119.88854952528445$$



$$\sigma^2 = 70.56997201358807$$

Coefficiente de Determinación ( $r^2$ ): 0.8837675289902613

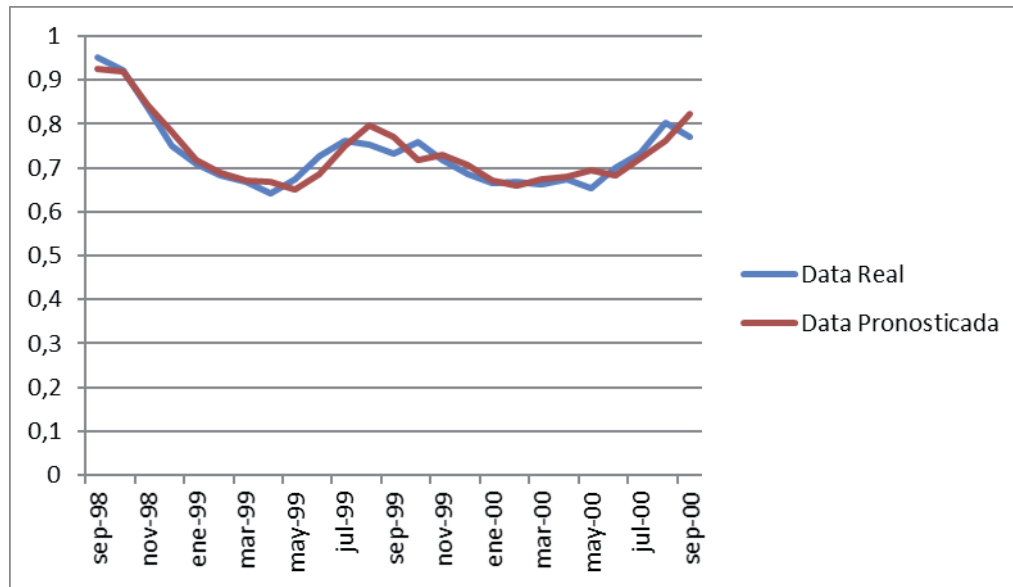


Figura 15: Gráfico de data pronosticada utilizando el modelo de LS-SVM con GPSO

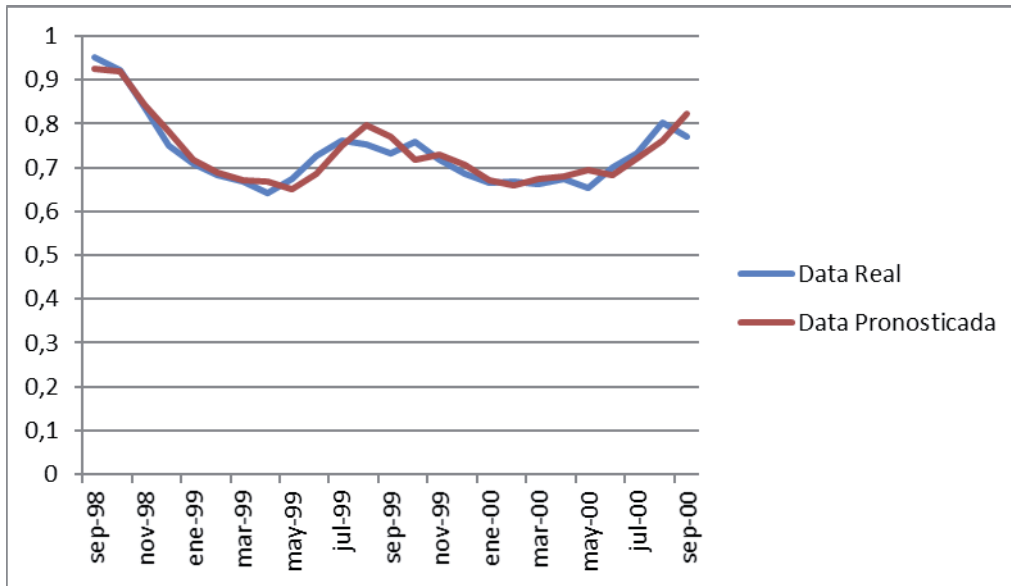
#### 4.8. Resultado del modelo LS-SVM con DPSO

Con un desfase de 5, en la matriz de entrada a la LS-SVM, los resultados son:

$$C = 59.45732796342545$$

$$\sigma^2 = 70.38257853942267$$

Coefficiente de Determinación ( $r^2$ ): 0.8844993213782861



**Figura 16: Gráfico de data pronosticada utilizando el modelo de LS-SVM con DPSO**

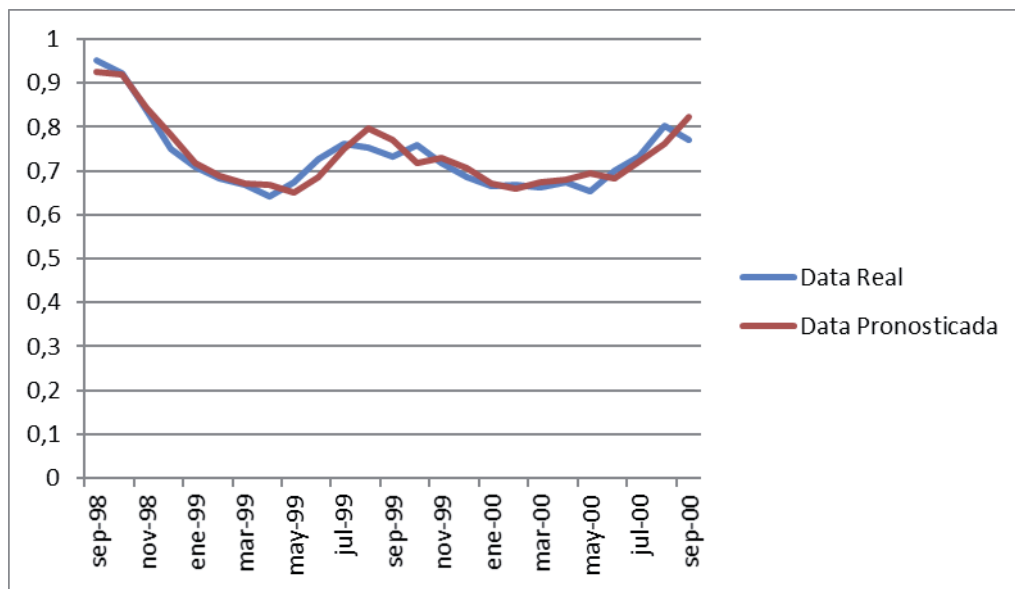
#### 4.8.1. Resultado del modelo LS-SVM con PSOPC

Con un desfase de 5, en la matriz de entrada a la LS-SVM, los resultados son:

$$C = 103.73579621342795$$

$$\sigma^2 = 70.67532456788954$$

Coefficiente de Determinación ( $r^2$ ): 0.8840580168472192



**Figura 17: Gráfico de data pronosticada utilizando el modelo de LS-SVM con PSOPC**

#### 4.9. Análisis de los resultados para el modelo LS-SVM con PSO.

Como se pudo observar con los resultados de las variantes de PSO, el coeficiente de determinación aumentó levemente, mejorando bastante la discretamente el pronóstico entregado por el modelo, esto fue gracias a que los valores de  $\sigma$  y C fueron optimizados con las variantes de PSO, para posteriormente ingresarlos en la LS-SVM.

Aunque no se observaron diferencias significativas entre las diferentes variaciones de PSO, los resultados fueron satisfactorios.

Tipo de PSO	C	$\sigma$	$r^2$
PSO	30.190234373133073	70.77517735142351	0.8845389705288078
GPSO	119.88854952528445	70.56997201358807	0.8837675289902613
DPSO	59.45732796342545	70.38257853942267	0.8844993213782861
PSOPC	103.73579621342795	70.67532456788954	0.8840580168472192

**Tabla 8: Resultados de la implementación LS-SVM con variación de PSO**

## 5. CONCLUSIONES.

En este trabajo se realizó un exhaustivo estudio del estado del arte del modelo de soporte vectorial para la regresión, y de la técnica de mínimos cuadrados, para introducir el modelo de LS-SVM. Adicionalmente, se incorporó el algoritmo de PSO, para determinar el parámetro de regularización y el relacionado a la función kernel.

Estas técnicas lograron crear un modelo de predicción que se pudo implementar correctamente entregando los primeros pronósticos, demostrando que la técnica LS-SVM no entregó resultados bastante precisos, debido el ingreso de los parámetros de una manera poco óptima. En cambio, el modelo LS-SVM con PSO, aumentó considerablemente el coeficiente de determinación ( $r^2$ ), provocando una mayor confianza y persistencia del modelo, logrando como consecuencia el aumentando de la precisión en el pronóstico entregado. De todas maneras, se concluyó que la técnica LS-SVM es perfectamente una alternativa para modelos de regresión no Lineal. Al momento de implementar las variantes de PSO (GPSO, DPSO y PSOPC), se observó un leve aumento en la precisión del modelo.

Respecto a los resultados obtenidos, en la LS-SVM se observó que al no utilizar una técnica que optimizara los valores C y  $\sigma$ , para ingresarlos en la máquina, el mejor resultado obtenido fue de un 39,9% de exactitud, con un 95,6% de persistencia del modelo. En cambio, la LS-SVM-PSO, logró optimizar estos parámetros, entregando una mejor predicción con un 88,75% de exactitud y un 98,21% de persistencia para el modelo.

Respecto al costo computacional de la aplicación, se puede concluir que las Maquinas Vectoriales Evolutivas son realmente muy costosas en comparación a otras técnicas, las estimaciones de tiempo rondaban entre los 2 y 10 minutos dependiendo del desfase, pero cabe destacar que son una herramienta realmente satisfactoria para realizar este tipo de estudios.

Este trabajo se reconoce como un real aporte para el estudio de estas técnicas, las que pueden continuar mejorando, incorporando una nueva técnica evolutiva, para construir un modelo híbrido que nos permita mejorar las predicciones de este modelo LS-SVM-PSO.

También se pudo reconocer las consecuencias que provoca el fenómeno oceanográfico, conocido como el Efecto del Niño, en Chile, siendo estas realmente devastadoras, lo cual perjudica a la economía del país y a sus habitantes, destacando sus consecuencias en el ámbito pesquero. Gracias a este estudio, se logra en un futuro cercano, abrir las puertas al interés por resguardarse de dicho fenómeno oceanográfico.

En la próxima etapa y final, se incorporará la técnica de GA, para optimizar la búsqueda de los valores de los parámetros de regresión, para luego ingresarlos a la LS-SVM, y así lograr un pronóstico más exacto, con un modelo de mayor confianza.

Finalmente se puede afirmar que los objetivos definidos en la sección 2.1, se han cumplido cabalmente y se espera que esta investigación sirva como base para futuros estudios relacionados con el pronóstico de la temperatura superficial del mar y también así, poder ampliar el campo de estudio a otras áreas que presenten datos históricos.

## 6. BIBLIOGRAFÍA.

- [1] Ganapathiraju, A.; Hamaker, J.; Picone, J. (1998) “Support vector machines for speech recognition”, Proc. ICSLP, Australia.
- [2] Weston, J.; Gammerman, A.; Stitson, M.; Vapnik, V.; Vovk, V.; Watkins. (1998) “Density estimation using support vector machines”, Technical Report CSD-TR-97-23.
- [3] Müller, K.; Smola, A.; Ratsch, G.; Schölkopf, B.; Kohlmorgan, J.; Vapnik, V. (1997) “Predicting time series with support vector machines”, Proc. of the ICANN Conference.
- [4] Burges, C. (1998) “A Tutorial on support vector machines for pattern recognition”, Data Mining and Knowledge Discovery, 2(2).
- [5] K. Vojslav, Learning and Soft Computing—Support Vector Machines, Neural Networks and Fuzzy Logic Models, The MIT Press, Massachusetts, 2001.
- [6] J. Holland, Adaptation in Natural and Artificial System, University of Michigan Press, AnnArbor, 1975.
- [7] N. Cristianini y J. Shawe-Taylor. An Introduction to Support Vector Machines. Cambridge, University Press, 2000.
- [8] V. Vapnik. The Nature of Statistical Learning Theory. Springer Verlag, New York, 1995.
- [9] Kennedy, J. y Eberhart, R., (1995) “Particle Swarm Optimization”, Proceedings of the IEEE International Conference on Neural Networks.
- [10] Kennedy, J., Eberhart, R. y Shi, Y. (2001). “Swarm Intelligence”, The Morgan Kaufmann Series in Artificial Intelligence, Morgan Kaufmann.
- [11] B. Secrest and G. Lamont, “Visualizing particle swarm optimization— Gaussian particle swarm optimization,” in Proc. IEEE Swarm Intell. Symp., Apr. 2003, pp. 198–204.
- [12] R. Krohling, “Gaussian particle swarm with jumps,” in Proc. IEEE Congr. Evol. Comput., Sep. 2005, vol. 2, pp. 1226–1231.

- [13] R. Krohling, "Gaussian swarm: A novel particle swarm optimization algorithm," in Proc. IEEE Con. Cybern. Intell. Syst., Dec. 2004, vol. 1, pp. 372–376.
- [14] P. Biskas, N. Ziogos, A. Tellidou, C. Zoumas, A. Bakirtzis, V. Petridis, and A. Tsakoumis, "Comparison of two metaheuristics with mathematical programming methods for the solution of OPF," in Proc. Int. Conf. Intell. Syst. Appl. Power Syst., Nov. 2005, pp. 510–515.
- [15] X. Xie, W. Zhang, and Z. Yang, "A dissipative particle swarm optimization," in Proc. IEEE Congr. Evol. Comput., May 2002, vol. 2, pp. 1456–1461.
- [16] S. He, J. Wen, E. Prempain, Q. Wu, J. Fitch, and S. Mann, "An improved particle swarm optimization for optimal power flow," in Proc. Int. Conf. Power Syst. Technol., Nov. 2004, pp. 1633–1637.
- [17] K. Parsopoulos and M. Vrahatis, "Recent approaches to global optimization problems through particle swarm optimization," *Natural Computing*, vol. 1, pp. 235–306, May 2002.

## 7. ANEXO.

En este anexo se muestra el código fuente asociado al proyecto de título, indicando los principales métodos utilizados para el desarrollo de la técnica de PSO.

### 7.1. Métodos Optimización de partículas por enjambre con LS-SVM

#### Inicialización de Matlab.

```
JMatLink variableMatLab = new JMatLink();
variableMatLab.engOpen();
```

#### PSO-Estructura, constructor clase Enjambre e inicialización de variables.

```
/**
 *
 * @author Christian Macaya Gatica
 */
public class Enjambre {

    private int N;
    private double w;
    private double c1;
    private double c2;
    private double Vmax;
    private double Xmax;
    private Particula[] P;
    private double[] Gbest;
    private double E; //error asociado al Gbest
    private JMatLink conMatlab;
    private double[][] entrada;
    private double[][] ideal;

    public Enjambre(JMatLink a,int N, double w, double c1, double
c2, double Vmax, double Xmax){

        this.N = N; //numero de particulas
        this.w = w; //peso de inercia
        this.c1 = c1; //factor cognitivo
        this.c2 = c2; //factor social
        this.Vmax = Vmax; //velocidad maxima
        this.Xmax = Xmax; //posicion maxima
        this.E = 1; //error inicial enjambre
```



```

        this.conMatlab = a; //inicio session de matlab
        P = new Particula[N]; //particulas del enjambre
    }

    public void inicializar(){

        for(int i=0; i<N; i++){
            P[i] = new Particula(conMatlab);
            double parametros[] = new double[2];
            double a = Math.random()*(100 - 1)+1;;
            double b = (Math.random()*(1500 - 1))+1;
            //obtencion de parámetro sig2
            parametros[0]=a;
            //obtención de parámetro C
            parametros[1]=b;

            P[i].inicializar(parametros,Vmax);
            P[i].calcularErrorParticula(entrada,ideal);
            //mejor error inicial particula
            P[i].setBestError(P[i].getError());
        }
        //primer mejor error default es P[0]
        Gbest = P[0].getPosicion();
        E = 1;
    }
}

```

## PSO-Iterar

```

public void iterar() {

    double Eit = 0;

    for(int i=0; i<N; i++){
        //Se actualiza la posición X y la velocidad V
        P[i].actualizar(w, c1, c2, Vmax, Xmax, Gbest);

        P[i].calcularErrorParticula(entrada,ideal);
        Eit = P[i].getError();

        if(Eit < P[i].getBestError()){
            P[i].setPbest(P[i].getPosicion());
            P[i].setBestError(Eit);
        }

        if(Eit < E){
            this.Gbest = P[i].getPosicion();
            this.E = Eit;
        }
    }
}

```

## PSO-Estructura, constructor e inicialización de la partícula

```

/**
 *
 * @author Christian Macaya Gatica
 */
public class Particula {
    //Vector que representa las posiciones del enjambre
    private double[] X;
    //Vector que representa las velocidades del enjambre
    private double[] V;
    private double[] Pbest;
    //Errores
    private double E;
    private double Ebest;

    JMatLink conMatlab;

    public Particula(JMatLink a){
        this.conMatlab = a;
    }

    //inicializacion de variables
    public void inicializar(double[] X, double Vmax){

        double rand;
        this.X = X;
        this.Pbest = X;
        V = new double[X.length];

        //velocidades iniciales
        for(int i=0; i<X.length; i++){
            rand = Math.random();
            V[i] = rand*Vmax + (1-rand)*(-Vmax);
        }
    }
}

```

## **PSO-Actualización de posición y velocidad de partícula**

```

//metodo para actualizar la posicion y velocidad de la particula
public void actualizar(double w, double c1, double c2, double
Vmax, double Xmax, double[] Gbest){

    for(int i=0; i<X.length; i++){
        V[i] = w*V[i] + c1*Math.random()*(Pbest[i] - X[i]) +
c2*Math.random()*(Gbest[i]-X[i]);
        if(V[i] > Vmax) V[i] = Vmax;
        if(V[i] < -Vmax) V[i] = -Vmax;
        X[i] = X[i] + V[i];
        if(X[i] > Xmax) X[i] = Xmax;
        if(X[i] < -Xmax) X[i] = -Xmax;
    }
}

```

## **PSO-Obtención de parámetro sig2 y C ubicados en la posición**

```
public int obtener_C() {
    int C = (int)X[1];
    return C;
}

public double obtener_sig2() {
    double valor;
    valor = X[0];
    return valor;
}
```