

UNIVERSIDAD DE COSTA RICA  
SISTEMA DE ESTUDIOS DE POSTGRADO

SELECCIÓN DE PARÁMETROS EN MÁQUINAS DE  
SOPORTE VECTORIAL PARA IDENTIFICACIÓN DE  
POLARIDAD

Tesis sometida a la consideración de la Comisión del Programa  
de Estudios de Posgrado en Computación e Informática para  
optar por el grado y título de Maestría Académica en  
Computación e Informática

MARCO AURELIO SANABRIA RODRÍGUEZ

Ciudad Universitaria Rodrigo Facio, Costa Rica

2016

Dedico este trabajo a mi papá y a mi mamá.

# Agradecimientos

Le agradezco a todos los que me ayudaron con este trabajo, en especial a mi comité asesor Edgar Casasola, Gabriela Marín, Gabriela Barrantes, a los compañeros del Centro Académico Alajuela, unidad desconcentrada de Ingeniería en computación en la sede Interuniversitaria de Alajuela y a los compañeros del grupo de investigación de análisis de sentimiento por sus valiosas revisiones y recomendaciones sobre el presente trabajo.

“Esta tesis fue aceptada por la Comisión del Programa de Estudios de Postgrado en Computación e Informática de la Universidad de Costa Rica, como requisito parcial para optar al grado y título de Maestría Académica en Computación e Informática.”



---

Dr. Juan José Vargas Morales  
**Representante de la Decana  
Sistema de Estudios de Posgrado**



---

Msc. Edgar Casasola Murillo  
**Director de Tesis**



---

Dra. Gabriela Marín Raventós  
**Asesora**



---

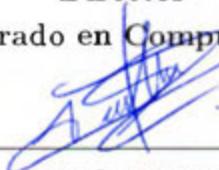
Dra Gabriela Barrantes Sliesarieva  
**Asesor**



---

Dr. Vladimir Lara Vilagrán  
**Director**

Programa de Postgrado en Computación e Informática



---

Marco Aurelio Sanabria Rodríguez  
**Candidato**

# Índice general

Dedicatoria . . . . .	i
Agradecimientos . . . . .	ii
Hoja de Aprobación . . . . .	iii
Índice General . . . . .	iv
Índice de Figuras . . . . .	v
Índice de Cuadros . . . . .	vi
Resumen . . . . .	vii
<b>1. Introducción</b>	<b>1</b>
1.1. Definición del problema . . . . .	1
1.2. Justificación . . . . .	2
1.3. Objetivos . . . . .	3
1.3.1. Objetivo General . . . . .	3
1.3.2. Objetivos Específicos . . . . .	3
1.3.3. Alcance y limitaciones . . . . .	3
<b>2. Marco Teórico</b>	<b>4</b>
2.1. Análisis de sentimientos . . . . .	4
2.1.1. Detección de polaridad . . . . .	5
2.2. Máquinas de soporte vectorial . . . . .	6
2.2.1. Clasificadores lineales . . . . .	6
2.2.2. Máquinas de soporte vectorial de margen fijo . . . . .	7
2.2.3. Máquinas de soporte vectorial de margen flexible . . . . .	8
2.2.4. Máquinas de soporte vectorial para clases múltiples . . . . .	8
2.2.5. Funciones de <i>kernel</i> . . . . .	10
2.2.6. Funciones comunes de <i>kernel</i> . . . . .	11
2.2.7. Selección de parámetros . . . . .	12
2.2.8. Búsqueda en malla . . . . .	13
2.3. Inteligencia de enjambre . . . . .	14

2.3.1.	Algoritmos de inteligencia de enjambre . . . . .	15
2.3.2.	Optimizador del lobo gris . . . . .	16
<b>3.</b>	<b>Antecedentes</b>	<b>23</b>
3.1.	Análisis de sentimientos . . . . .	23
3.2.	Selección de parámetros . . . . .	26
3.3.	Inteligencia de enjambre y Optimizador de lobo gris . . . . .	26
<b>4.</b>	<b>Metodología</b>	<b>29</b>
4.1.	Modelo de estimación de parámetros . . . . .	29
4.1.1.	Pre-procesamiento del corpus . . . . .	31
4.1.2.	Selección de parámetros . . . . .	31
4.1.3.	Generación y evaluación de los modelos . . . . .	31
4.2.	Herramientas . . . . .	32
4.3.	Experimento . . . . .	32
4.3.1.	Pre procesamiento del corpus. . . . .	33
4.3.2.	Extracción de características. . . . .	35
4.3.3.	Entrenamiento del modelo. . . . .	36
4.3.4.	Evaluar los modelos generados. . . . .	38
<b>5.</b>	<b>Resultados y conclusiones</b>	<b>42</b>
5.1.	Resultados . . . . .	42
5.2.	Discusión adicional . . . . .	45
5.3.	Conclusiones . . . . .	47
<b>A.</b>	<b>Tablas de resultados de las métricas de evaluación</b>	<b>49</b>
<b>B.</b>	<b>Algoritmo optimizador del lobo gris</b>	<b>55</b>
<b>C.</b>	<b>Tablas de resultados del proceso de selección de parámetros</b>	<b>63</b>

# Índice de figuras

1.1. Diagrama entradas y salidas del problema . . . . .	2
2.1. Proceso de cacería, tomado de [Mirjalili et al., 2014] . . . . .	18
2.2. Cambio de posición durante la ejecución de GWO, tomado de [Mirjalili et al., 2014] . . . . .	20
2.3. Búsqueda(b) y ataque(a) de la presa, tomado de [Mirjalili et al., 2014] .	21
2.4. Pseudo código del algoritmo GWO, tomado de [Mirjalili et al., 2014] . .	22
4.1. Diagrama del modelo para selección de parámetros . . . . .	30
4.2. Distribución de las instancias entre las clases, a la izquierda la distribución original y a la derecha la distribución luego del pre-procesamiento.	34
4.3. División del corpus procesado en 60 subconjuntos extraídos aleatoriamente	35
4.4. Vector de rangos de probabilidades . . . . .	36
4.5. Resultado de ejemplo de una clasificación . . . . .	38
4.6. Representación gráfica de la métrica de precisión . . . . .	39
4.7. Representación gráfica de la métrica de exhaustividad . . . . .	39
4.8. Resultado de ejemplo de una clasificación . . . . .	40
4.9. Generar y evaluar los modelos . . . . .	41
4.10. Ejemplo de resultado de evaluación de un modelo de clasificación . . .	41
5.1. Diagrama de cajas para la métrica de Precisión . . . . .	43
5.2. Diagrama de cajas para la métrica de exhaustividad . . . . .	44
5.3. Diagrama de cajas de F-Score . . . . .	45

# Índice de cuadros

2.1. Algoritmos utilizados en la selección de parámetros de máquinas de soporte vectorial . . . . .	12
3.1. Los tres mejores resultados las tareas de análisis de sentimientos a nivel global (con 3 clases), en las distintas ediciones del TASS . . . . .	25
3.2. Los tres mejores resultados las tareas de análisis de sentimientos a nivel global (con 5 clases), en las distintas ediciones del TASS . . . . .	25
4.1. Rangos de valores por defecto en la herramienta svm-grid expresados utilizando una escala logarítmica( $\log_2$ ) . . . . .	37
4.2. Configuración de los parámetros para los tratamientos correspondientes al algoritmo del lobo gris . . . . .	37
5.1. Valores de mínimo, máximo, media y desviación estándar para la métrica de Precisión . . . . .	43
5.2. Análisis estadístico de Recall . . . . .	44
5.3. Análisis estadístico de F-Score . . . . .	45
A.1. Resultados de precisión . . . . .	49
A.2. Resultados de exhaustividad . . . . .	51
A.3. Resultados de F-Score . . . . .	52
C.1. Resultados obtenidos del proceso de selección de parámetros, los valores se encuentran en escala logarítmica . . . . .	63

# Resumen

La clasificación de texto es una herramienta que permite categorizar la información generada utilizando lenguaje natural, en particular, las redes sociales como facebook <sup>1</sup> o twitter <sup>2</sup> generan gran cantidad de información en tiempo real que puede servir como insumo a una amplia variedad de procesos.

Desafortunadamente, convertir la cantidad masiva de información generada por estos medios, aún tiene mucho camino por recorrer. Debido a que esta información es de carácter subjetivo, informal y no estructurada, se recurre a diversas técnicas de inteligencia artificial y Procesamiento del lenguaje natural (NLP) con el fin de identificar y extraer opiniones, y para determinar el valor subjetivo (polaridad positiva o negativa) que carga el texto.

Específicamente, para determinar la polaridad de un opinión, se ha recurrido a modelos de clasificación de texto. En particular las máquinas de soporte vectorial (C-SVM) con función de kernel RBF, gozan popularidad entre los investigadores del área por sus buenos resultados [firmino et al., 2013, Kaya et al., 2012]. Las C-SVM deben ser entrenadas para que aprendan a clasificar textos, lo cuál involucra un proceso de selección de parámetros, el cual posee un impacto significativo en la precisión del modelo.

En investigación comparamos, utilizando las métricas de precisión, exhaustividad y valor-F, el uso de valores por defecto de la herramienta LIBSVM contra dos algoritmos de selección automatizada de parámetros:

**El algoritmo de búsqueda en malla:** de amplio uso entre los investigadores [Carrizosa et al., 2014, Devos et al., 2009, Li et al., 2010b, Li et al., 2010a, Lin and Liu, 207, Lin et al., 2008, Mu an nandi, 2007, Wei-Chih and Yu, 2009], se caracteriza por ser exhaustivo y brindar una alta precisión pero con una cantidad significativa de cálculos que implica un alto consumo de tiempo y recursos computacionales

**El algoritmo de lobo gris:** creado en 2014 por Mirjalili [Mirjalili, et al., 2014], que basa su funcionamiento en el comportamiento jerárquico de los lobos grises al cazar, teniendo entre sus principales ventajas simplicidad y rápida convergencia.

El análisis estadístico de los resultados muestra que la diferencia encontrada entre los resultados de los distintos tratamientos no es significativa.

---

<sup>1</sup><http://facebook.com>

<sup>2</sup><http://twitter.com>

# Capítulo 1

## Introducción

En el *AAAI fall symposium: commonsense knowledge* de 2010, [Cambria et al., 2010] menciona cómo el web 2.0 le ha dado a los usuarios el poder de participar activamente en la creación de contenido y ha cambiado la forma en la que estos creadores colaboran y comparten sus creaciones a través de las redes sociales, *blogs* y *wikis*, de forma que han dado vida a una inteligencia colectiva a la que muchas veces se escucha con más fuerza que los puntos de vista de los expertos en el campo.

El conocimiento generado de esta forma es muy útil, pero increíblemente difícil de procesar debido a su naturaleza no estructurada y subjetiva, razón por la cuál el análisis de sentimientos o minería de opinión, que consiste en la inferencia de estados emocionales a partir de una opinión escrita, ha cobrado fuerza en los últimos años, particularmente por su importancia en campos como el mercadeo, perfiles personales y predicción de mercados financieros.

En particular, el determinar si una opinión es negativa o positiva (detección de polaridad) se puede abordar combinando distintas áreas de conocimiento como procesamiento de lenguaje natural (NLP) y el aprendizaje de máquina.

### Definición del problema

En esta investigación nos centramos en una técnica proveniente del aprendizaje de máquina conocida como máquinas de soporte vectorial. Estas son efectivas en la clasificación de texto pero poseen una complejidad inherente en el proceso de selección de parámetros (ver figura 1.1) e hiper parámetros, pues este procedimiento, aún para un experto en el campo, involucra el uso de distintas estrategias para fijar los valores necesarios, afectando el rendimiento y efectividad de la técnica.

Para superar este problema, se propone el uso del algoritmo de selección de lobo gris (GWO, por sus siglas en inglés), el cuál es simple y flexible [Song et al., 2015]; aplicado a textos cortos en español provenientes de la red social *Twitter* <sup>1</sup>.

---

<sup>1</sup><http://www.twitter.com>

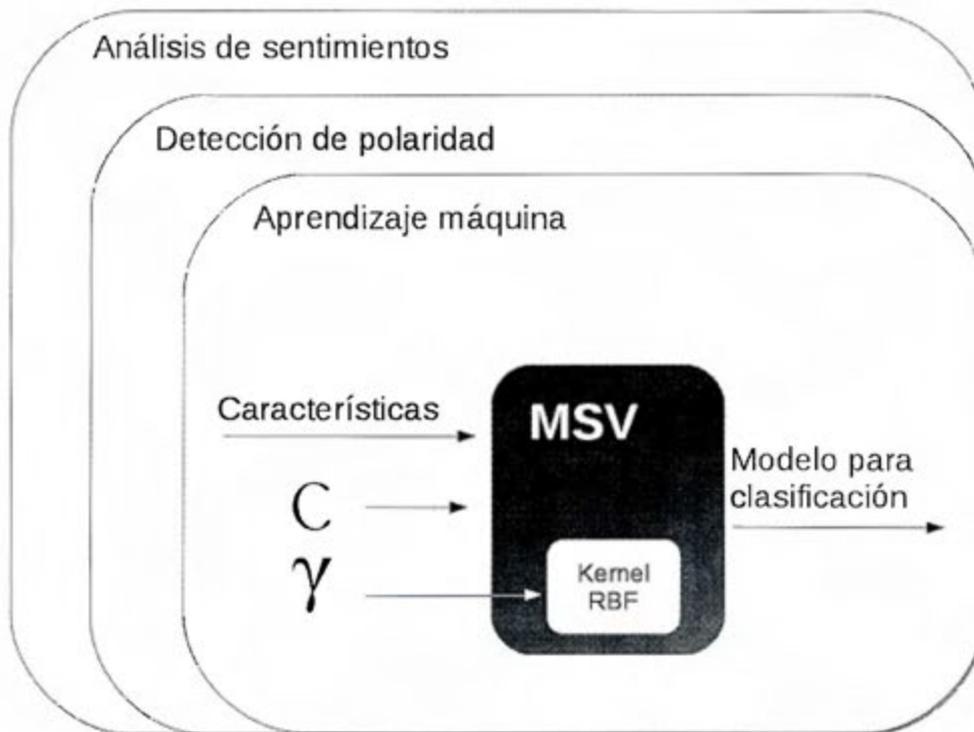


Figura 1.1: Diagrama entradas y salidas del problema

## Justificación

Se seleccionaron las máquinas de soporte vectorial ya que estas han sido utilizadas, produciendo resultados competitivos en comparación a otras técnicas utilizadas en esta área [Firmino et al., 2013, Kaya et al., 2012].

Sin embargo; de la revisión de literatura se extrae que los lingüistas computacionales puede que no estén explotando las capacidades de esta herramienta, debido los mecanismos que estos utilizan para el entrenamiento de esta herramienta no se documentan o se utilizan los parámetros que la herramienta provee como valores por defectos [Trindade et al., 2014, Aisopos et al., 2012], por lo cual se considera que ahondar en la selección de parámetros para mejorar la calidad de las soluciones en este campo puede tener un impacto significativo.

Por último, el desarrollar una metodología para optimizar la selección de parámetros de las máquinas de soporte vectorial, no solo puede ser aplicable a este campo, sino también a cualquier otra aplicación donde se necesite realizar una clasificación.

## Objetivos

### Objetivo General

Comparar el algoritmo de búsqueda en malla contra el algoritmo de lobo gris para la obtención de los valores de parámetros para máquinas de soporte vectorial *C-SVM* con función de kernel RBF sobre la precisión al identificar la polaridad de textos cortos en español.

### Objetivos Específicos

1. Proponer un modelo que utilice el método de malla y el método del lobo gris para estimar los parámetros de MSV+RBF
2. Construir el prototipo para generar los valores de los parámetros utilizando lobo gris contra malla.
3. Determinar la diferencia en términos de precisión, exhaustividad y valor-f de la salida de la máquina de soporte vectorial producto de los parámetros obtenidos por ambos algoritmos en igualdad de condiciones al identificar la polaridad de textos cortos en español.

### Alcance y limitaciones

Esta investigación involucra únicamente la aplicación de la formulación básica de las máquinas de soporte vectorial conocida como *C-SVM* junto a la función de kernel de base radial (RBF) implementada por la herramienta LibSVM.

Los resultados están limitados al ámbito de la detección de polaridad para textos cortos en español, cuya extracción de característica esta caracterizada por el uso de un diccionario de probabilidades y el uso de marcadores de énfasis en la forma de las palabras.

# Capítulo 2

## Marco Teórico

En el campo del análisis de sentimientos o minería de opinión, las técnicas provenientes del aprendizaje de máquina son un recurso valioso, particularmente las máquinas de soporte vectorial proveen al usuario con un mecanismo para clasificar textos según su polaridad (positiva o negativa). Estas aunque muy exitosas, requieren de mucha experiencia ya que su configuración no es trivial [Zong et al., 2006]; por lo que se recurre a herramientas que permitan automatizar esta configuración conocida como selección de parámetros.

Para atacar las dificultades de la selección de parámetros, se ha incursionado en el área de la inteligencia de enjambre con el algoritmo optimizador de enjambre de partículas [Lu et al., 2009, Qiu et al., 2010, Zong et al., 2006].

Este capítulo está dividido en tres secciones: Análisis de sentimientos, máquinas de soporte vectorial e inteligencia de enjambre para introducir al lector a los conceptos necesarios para entender los temas mencionados en los párrafos anteriores.

### Análisis de sentimientos

Usualmente se le asignan distintos nombres en la literatura tales como minería de opinión, análisis de sentimientos o análisis de subjetividad, estos nombres se basan en que esta es un área de investigación que trata con opiniones, sentimientos y subjetividad en textos [Pang and Lee, 2008].

Lo que otras personas piensen de un objeto, persona o producto siempre ha sido importante para nosotros; preguntamos a los amigos sobre recomendaciones de mecánicos, futuros presidentes, productos de supermercado, programas de televisión o incluso los electrodomésticos a comprar [Pang and Lee, 2008].

Este es un campo, en que el acceso generalizado al *internet* (foros, grupos de discusión y *blogs*) ha cambiado dramáticamente la forma en que la gente expresa sus puntos de vista y opiniones. De la misma forma si se desea comprar algún producto, ya no estamos limitados a preguntar a familiares y amigos al considerar opiniones. Incluso para las compañías ya no se hace tan necesario el llevar a cabo en cuenta, grupos de enfoque o consultores externos para conocer la opinión que un usuario tiene sobre el

producto que ofrece en el mercado [Pang and Lee, 2008].

El convertir todas estas opiniones en material útil es un proceso que resulta complejo [Liu, 2010]; pues para un ser humano encontrar fuentes de datos, extraer las opiniones, leerlas, procesarlas y visualizarlas convierte al análisis de sentimientos en un problema retador en las áreas de Procesamiento del lenguaje natural (PLN) y la minería de textos [Pang and Lee, 2008].

El concepto de opinión es muy amplio, por lo que la investigación se enfoca en el aspecto de la opinión que expresa subjetividad [Kim and Hovy, 2004], identificando cómo los sentimientos son expresados a través de textos y de cómo la opinión expresada puede ser negativa (desfavorable) o positiva (favorable) [Nasukawa et al., 2003].

Particularmente Pang y Lee [Pang and Lee, 2008], define una opinión como una quintupla  $(e_j, a_{jk}, so_{ijkl}, h_i, t_l)$  dónde:

- $e_j$  Entidad a la que se refiere la opinión.
- $a_{jk}$  Aspecto de la entidad a la que se refiere la opinión.
- $so_{ijkl}$  Valor subjetivo de la opinión
- $h_i$  El sujeto que emite la opinión.
- $t_l$  Fecha en que se emitió la opinión.

Visto de otra forma, el análisis de sentimientos involucra [Nasukawa et al., 2003, Thelwall et al., 2010]:

- Identificación de las opiniones
- Detección de la polaridad ( cuando esta es favorable o desfavorable) y fuerza de la expresión.
- Relación de la opinión con el sujeto.

La detección de polaridad presenta una serie de complicaciones [Kim and Hovy, 2004], por lo que es considerada una sub área de investigación que se necesita cubrir de manera particular.

## Detección de polaridad

La detección de polaridad es una sub área de investigación que se interesa en evaluar textos y determinar si estos tienen una intencionalidad negativa o positiva [Isguder-Sahin et al., 2014], la cual, cuando se procesa una gran cantidad de documentos puede llegar a ofrecer soluciones en campos como el análisis competitivo, análisis de mercadeo o incluso detección de rumores desfavorables en el contexto de manejo de riesgos [Nasukawa et al., 2003]. Particularmente desde el área de aprendizaje de máquina, se han utilizando algunos algoritmos para clasificar por categorías los textos en positivos, negativos o neutrales:

- Naïve Bayes: es un clasificador probabilístico fundamentado en el teorema de Bayes, que se caracteriza por asumir de que la presencia/ausencia de una característica particular no esta relacionada con las otras. [Firmino et al., 2013]
- Máxima entropía: predice el resultado de una variable categórica en función de las variables independientes.[Kaya et al., 2012]
- Redes neuronales: La red neuronal es un algoritmo de aprendizaje de máquina supervisado para categorización inspirado en el funcionamiento de las neuronas del cerebro humano. [Madhoushi and Hamdan, 2015]
- Máquinas de soporte vectorial: Técnica que dado un conjunto de vectores de características pertenecientes a categorías distintas distribuidas en un espacio n-dimensional, trata de encontrar un hiper-plano. [Carstens and Toni, 2013]

## Máquinas de soporte vectorial

Las máquinas de soporte vectorial son una técnica proveniente del área de aprendizaje de máquina [Chang and Lin, 2011] supervisado [Mouthami et al., 2013], que es descrito en su versión más reciente como *de margin flexible* o *soft margin SVM* descritas en [Cortes and Vapnik, 1995].

Este método de clasificación es un ejemplo de **clasificadores lineales** [Ben-Hur and Weston, 2010], que desarrollan un modelo de clasificación basado en la construcción de un hiper plano n-dimensional que separa de manera óptima el conjunto de datos de entrada en dos categorías [Mouthami et al., 2013].

## Clasificadores lineales

Los clasificadores lineales están basados en el modelo lineal [Abu-Mostafa et al., 2012] y funcionan dividiendo el conjunto de datos linealmente separables en dos categorías, asignándole una etiqueta  $y_i$  (ecuación 2.1) a un vector  $x$ .

$$\exists y_i \mid y_i \in \{-1, +1\} \quad (2.1)$$

donde los valores de  $y_i$  corresponden a una de dos categorías posibles de las que pueden ser asignados los datos [Abu-Mostafa et al., 2012].

La división exacta está dada por un hiper-plano desconocido, descrito por la función  $f(x)$  (ver ecuación 2.2) también desconocida; por lo que se trata de crear un hiper-plano [Mouthami et al., 2013] descrito por  $h(x)$  (ecuación 2.3) de tal manera que:  $h(x)$  aproxima el resultado de  $f(x)$  ( $h(x) \approx f(x)$ ) basándose en un conjunto de datos de prueba  $X$  (ecuación 2.3).

$$f(x) = y_i \quad (2.2)$$

$$h(x) = w^T x + b \quad (2.3)$$

donde  $w$  es un vector de pesos,  $b$  un valor umbral y  $x$  un elemento del conjunto de datos de entrenamiento  $x$  (ver 2.4) [Abu-Mostafa et al., 2012].

Si se toma  $b = 0$  al hiper-plano pasa por el origen.

$$\exists x \in X \mid x = (x_i, y_i) \quad (2.4)$$

La función  $h(x)$  se conoce como función *discriminante* y su signo nos indica el lado del hiper-plano donde se ubica el vector (ver 2.5).

$$y_i \approx \text{sign}(w^T x + b) \quad (2.5)$$

## Máquinas de soporte vectorial de margen fijo

Las máquinas de soporte vectorial de margen fijo, son clasificadores lineales de máximo margen, estas agregan un margen geométrico (ver 2.6) alrededor del hiper-

plano, tratando de que este tenga el mayor ancho posible.

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

Para más detalles sobre la definición del margen geométrico ver [Ben-Hur and Weston, 2010].

De esta forma, un clasificador de máximo margen es aquella función discriminante que maximice 2.6 o su equivalente, que minimice  $\|w\|^2$ , lo cual permite plantear el problema como un problema de optimización restringido [Abu-Mostafa et al., 2012]:

$$\begin{aligned} \underset{(w,b)}{\text{minimice}} \quad & \frac{1}{2} \|w\|^2 \\ \text{sujeto a} \quad & y_i(w \cdot x_i - b) \geq 1 \quad i = 1, \dots, n \end{aligned} \quad (2.7)$$

Las restricciones en el problema anterior aseguran el margen más amplio posible que logre clasificar correctamente todos los ejemplos; lo cual solo es posible si los datos son linealmente separables. Por otra parte, este planteamiento no es flexible con respecto al ruido o a casos atípicos.

## Máquinas de soporte vectorial de margen flexible

Definidas por Corina Cortes y Vladimir Vapnik en 1995 [Cortes and Vapnik, 1995], permiten corregir el problema de la tolerancia a errores, ruido y casos atípicos descritos en la sección anterior.

Para permitirle al clasificador ignorar algunos ejemplos se modifica el planteamiento de la ecuación 2.7 de la siguiente forma:

$$\begin{aligned} \underset{(w,b)}{\text{minimice}} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \\ \text{sujeto a} \quad & y_i(w \cdot x_i - b) \geq 1 - \xi \quad i = 1, \dots, n \end{aligned} \quad (2.8)$$

donde el término  $C \sum_{i=1}^n \xi_i$  se agrega para penalizar los errores de clasificación [Cortes and Vapnik, 1995].

Utilizando multiplicadores de Lagrange el planteamiento de la ecuación 2.8 se puede

escribir de la siguiente forma:

$$\begin{aligned} \underset{\alpha}{\text{maximice}} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i y_j \alpha_i \alpha_j \mathbf{x}_i^T \mathbf{x}_j \\ \text{sujeto a} \quad & \sum_{i=1}^n y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C. \end{aligned} \tag{2.9}$$

lo que lleva a la formulación del vector de pesos  $w$  en función de los vectores etiquetados de ejemplo:

$$\mathbf{w} = \sum_{i=1}^n y_i \alpha_i \mathbf{x}_i. \tag{2.10}$$

Los datos de ejemplo  $x_i$  para los cuales  $\alpha_i > 0$  son los puntos que están en el margen (o dentro de este en el caso del margen flexible) son los llamados vectores de soporte que le dan nombre a las SVM, ya que el hiper-plano se apoya solo en estos datos [Cortes and Vapnik, 1995].

El planteamiento anterior aunque es muy utilizado, únicamente cubre el caso cuando la clasificación a realizar comprende exclusivamente dos clases; una para cada lado del hiper plano. En caso de querer realizar la clasificación para múltiples clases se deben aplicar procedimientos específicos que se tratarán en la siguiente sección.

## Máquinas de soporte vectorial para clases múltiples

Inicialmente las máquinas de soporte vectorial fueron diseñadas como clasificadores binarios (para dos clases), por lo que si se requieren aplicar a problemas de clasificación que involucren más de una clase se debe recurrir a alguna estrategia que permita su funcionamiento en estos contextos:

Hsu y Lin mencionan como las tres estrategias más comunes [Hsu and Lin, 2002] para el manejo de clasificadores entre múltiples clases:

- Uno contra todos
- Uno contra uno.
- Grafos acíclicos dirigidos.

### Uno contra todos

Con esta estrategia se itera  $i$  veces sobre el modelo.

En cada iteración se aplica un clasificador binario entre la  $i$ -ésima clase y todas las demás como un conjunto.

$$h(x) = w_i^T x + b_i \quad (2.11)$$

Por cada clasificador se genera un hiper plano distinto (ver 2.11). Finalmente la clase a asignar corresponderá al resultado de la función discriminante dónde se obtuvo el mayor valor [Hsu and Lin, 2002].

### Uno contra uno

Este método construye  $\frac{k(k-1)}{2}$  clasificadores binarios, dónde cada uno se entrena utilizando dos clases distintas  $i$  y  $j$ , tal como se muestra en al siguiente ecuación [Hsu and Lin, 2002]:

$$[!h]h_{ij}(x) = w^T x + b \quad (2.12)$$

Para seleccionar a cuál clase pertenece un registro dado  $x$ , se evalúa en todos los clasificadores  $ij$  y luego se decide a cuál clase pertenece mediante un sistema de votación.

En este sistema se toman los resultados de los clasificadores y se crean pares de la forma  $(y_i, \text{cantidad de votos})$ ; cada vez que un clasificador da como resultado que el vector  $x$  corresponde a la clase  $y_i$ , se aumenta la cantidad de votos para esa clase. Al finalizar el vector  $x$ , pertenece a la clase con mas votos.

### Grafo acíclico dirigido

Al igual que la estrategia de *uno contra uno*, este algoritmo genera  $\frac{k(k-1)}{2}$  clasificadores binarios; sin embargo, al asignar la clase se utiliza un grafo acíclico no dirigido para asistir en este paso. Cada vértice del grafo representa un clasificador del que salen dos aristas (una por cada clase posible que puede ser asignada), hasta que se llega a un vértice hoja.

Es importante notar que los planteamientos discutidos en las secciones anteriores funcionan únicamente para datos linealmente separables, lo cuál no sucede con frecuencia. Una solución a este problema es la aplicación del truco de kernel, el cuál se ampliará más en la siguiente sección.

## Funciones de *kernel*

Para lograr que las SVM trabajen con datos que no son linealmente separables se puede recurrir a una transformación lineal  $\Phi(x) : \mathcal{X} \rightarrow \mathcal{Z}$  de forma que  $\Phi(x) = z$  donde  $z$  pertenece a un espacio con una mayor dimensionalidad  $\mathcal{Z}$  [Abu-Mostafa et al., 2012, Schölkopf, 2002].

El problema con transformar todos los datos a un nuevo espacio  $\mathcal{Z}$  es que el proceso de transformación tiene un costo computacional muy alto [Schölkopf, 2002].

Es por esto que se utiliza una función de *kernel* [Hofmann et al., 2008, Boser et al., 1992], la cuál ejecuta un producto punto en el espacio  $\mathcal{Z}$  sin la necesidad de hacer la transformación. Esta se puede definir cómo:

$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle \quad (2.13)$$

donde  $\langle \cdot, \cdot \rangle$  denota el producto punto [Boser et al., 1992].

Así si se reemplaza el producto punto  $\mathbf{x}_i^T \mathbf{x}_j$  en la ecuación 2.9 por la función de kernel descrita en la ecuación 2.13 obteniendo [Boser et al., 1992]:

$$\begin{aligned} \underset{\alpha}{\text{maximice}} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{sujeto a} \quad & \sum_{i=1}^n y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C. \end{aligned} \quad (2.14)$$

Toda función de *kernel* para ser válida debe cumplir con la condición de Mercer [Vapnik, 2013], la cuál establece que para cualquier función simétrica  $K(x, x')$ , esta puede ser representada cómo un producto punto en un espacio de dimensiones si y solo si [Vapnik, 2013]:

$$\int \int K(x, x') \Phi(x) \Phi(x') dx dx' > 0 \text{ para cada } \Phi(x) \neq 0 \text{ y } \int \Phi^2(x) dx < \infty \quad (2.15)$$

Burges menciona que una función de kernel que no cumpla con la condición de Mercer, corre el riesgo de encontrarse con un caso en el cual el problema de programación cuadrática no tiene solución [Burges, 1998].

A continuación se listan las funciones de kernel más comunes cuando se aplican a máquinas de soporte vectorial.

## Funciones comunes de *kernel*

Las funciones de *kernel* más utilizadas en labores de clasificación son lineal, polinomial, de funciones de base radial y funciones *sigmoid* [Patle and Chouhan, 2013]. Estas se describen a continuación:

Asumiendo  $u$  y  $v$  vectores de características pertenecientes a un espacio de dimensiones  $\mathcal{X}$  Patle define las siguientes ecuaciones para funciones de kernel [Patle and Chouhan, 2013]:

### Lineal

$$K(u, v) = u \cdot v^T \quad (2.16)$$

### Polinomial

$$K(u, v) = (1 + u \cdot v^T)^d \quad (2.17)$$

donde  $d$  es el grado de la función de *kernel*

### Funciones de base radial

$$K(u, v) = e^{-\gamma(\|u-v\|)^2} \quad (2.18)$$

### Funciones Sigmoid

$$K(u, v) = \tanh(\gamma \cdot u \cdot v^T + r) \quad (2.19)$$

Aunque las funciones de kernel resultan en máquinas de soporte vectorial capaces de clasificar datos que no son linealmente separables, introducen la dificultad de seleccionar apropiadamente sus parámetros [Li et al., 2010b], problema que conlleva amplia investigación que se tratará en la siguiente sección.

## Selección de parámetros

También conocido como el problema de la selección del modelo [Chapelle et al., 2002] es uno de los problemas que más impacta de manera di-

- \* Validación cruzada inversa [Duan et al., 2003].
- \* Dejar uno fuera [Duan et al., 2003].
- \* Xi - Alpha ( $\xi/\alpha$ ) bound [Duan et al., 2003]
- \* Generalización aproximada de validación cruzada [Duan et al., 2003]
- \* Límite aproximado de alcance [Duan et al., 2003].
- \* Límite VC [Duan et al., 2003]
- \* Optimización mínima secuencial (SMO) [Sánchez A, 2003, Yao and Li, 2009] [Davenport et al., 2010].
- \* Algoritmos genéticos [Liu et al., 2005].
- \* Optimizador de enjambre de partículas [Zong et al., 2006, Qiu et al., 2010].
- \* Descenso de gradiente [Mu and Nandi, 2007].
- \* Simulación de forjado. [Lin et al., 2008].
- \* Optimización multi-objetivo (EMO) [Watanabe and Kimura, 2008].
- \* Búsqueda en malla [Lin and Liu, 2007], [Mu and Nandi, 2007], [Mahmoud and Calvert, 2009], [Devos et al., 2009], [Wei-Chih and Yu, 2009], [Li et al., 2010b], [Li et al., 2010a].
- \* Selección automática de parámetros [Li et al., 2010a].
- \* Distancia entre dos clases [Sun et al., 2010].

Cuadro 2.1: Algoritmos utilizados en la selección de parámetros de máquinas de soporte vectorial

recta el rendimiento de las *SVM* [Li et al., 2010b] pues este depende de la selección de la función de *kernel*, los parámetros de esta función y el parámetro de regularización.

Para hacer esta selección es habitual aplicar distintas técnicas de selección de parámetros siendo búsqueda en malla [Lin and Liu, 2007] la más simple de implementar; sin embargo, consume mucho tiempo y recursos computacionales [Li et al., 2010b].

Es por esto que para resolver el problema que se menciona en los párrafos anteriores se han implementado gran variedad de algoritmos que varían en su aproximación a la solución que viene desde distintos campos tales como los algoritmos genéticos [Liu et al., 2005] y la forja [Lin et al., 2008]. Otros algoritmos utilizados para la selección de parámetros se muestran en el cuadro 2.1.

Den entre todos los algoritmos utilizados, destaca el de búsqueda en malla por ser el más común y confiable método de selección para el modelo de selección del problema [Mu and Nandi, 2007] por lo que se ampliará su definición en la siguiente sección.

## Búsqueda en malla

La búsqueda en malla o *grid search* es una técnica de selección de parámetros muy utilizada [Carrizosa et al., 2014, Devos et al., 2009, Li et al., 2010b, Li et al., 2010a, Lin and Liu, 2007, Lin et al., 2008, Mu and Nandi, 2007, Wei-Chih and Yu, 2009] en la aplicación de selección de SVM cuando son utilizadas funciones de *kernel* RBF, debido a que estas cuentan con solo dos parámetros para optimizar  $C$  (parámetro de regularización) y  $\gamma$  (parámetro de la función de *kernel*).

La técnica consiste en generar una matriz  $a_{m \times n}$  donde  $A_{i,j} = (C_i, \gamma_j)$  y determinar el Error absoluto relativo mediante *k-fold cross validation* para cada valor  $a_{i,j}$ .

Considerando que:

$$C_{k+1} = C_k + d_1$$

donde  $d_1 \in \mathbb{R}$ , definido por el investigador y  $C_k \in [a, b]$  tal que  $a$  y  $b$  son números reales definidos por el investigador.

$$\gamma_{k+1} = \gamma_k + d_2$$

donde  $d_2 \in \mathbb{R}$  definido por el investigador y  $\gamma_k \in [a, b]$  tal que  $a$  y  $b$  son números reales definidos por el investigador.

Se debe tomar en cuenta que los valores para  $d_1$  y  $d_2$  deben ser seleccionados de manera que la dimensionalidad de la matriz  $A$  se mantenga baja para evitar un incremento exagerado en el tiempo necesario para encontrar la solución, pero que mantenga un nivel de significancia alto de forma que la precisión de los resultados no se vea afectada.

De acuerdo a Lin y Liu, la búsqueda en malla tiene las siguientes ventajas y

desventajas[Lin and Liu, 2007]:

- Permite paralelizar la solución.
- Tiene un complejidad temporal de  $O(n^2)$
- Tiene una alta precisión junto a una cantidad significativa de cálculos que implica un alto consumo de tiempo y recursos computacionales.

## Inteligencia de enjambre

La definición de inteligencia de enjambre según Zhu y Tang se extrae del comportamiento en enjambre de grupos de organismos tales cómo aves, insectos, bacterias entre otros[Zhu and Tang, 2010] . De acuerdo a Ab Whahab, dos conceptos fundamentales que son necesarios para que un algoritmo se considere como parte de la categoría de inteligencia de enjambre son[Ab Wahab et al., 2015]:

- **Auto organización:** La capacidad de un sistema de desarrollar o evolucionar sus individuos de manera independiente sin ayuda externa.
- **División de tareas:** La ejecución simultánea de varias tareas simples por cada individuo.

Usualmente estos algoritmos no siguen comandos de un líder, o de un plan global determinado, sino más bien del comportamiento simple de sus agentes [Zhu and Tang, 2010].

## Algoritmos de inteligencia de enjambre

De acuerdo a la revisión de literatura realizada por Ab Wahab en 2015[Ab Wahab et al., 2015] y por Mirjalili en 2014[Mirjalili et al., 2014] se encuentran en esta categoría los siguientes algoritmos:

- Algoritmos genéticos (GA)
- Optimizador de colonia de hormigas (ACO)
- Optimizador de enjambre de partículas (PSO)
- Evolución diferencial (DE)
- Colonia artificial de abejas (ABC)
- Optimizador de enjambre de gusanos *glowwoms*(GSO)
- Algoritmo de búsqueda Cuckoo (CSA)
- Programación genética (GP)
- Estrategia de evolución (ES)
- Programación evolutiva (EP)
- Algoritmo de luciérnaga (FA)
- Algoritmo de murciélago (BA)
- Optimizador del lobo gris. (GSO)
- Algoritmo de optimización de matrimonio en abejas mieleras(MBO)
- Algoritmo de cardumen de peces artificial. (AFSA)
- Algoritmo de termina.
- Algoritmo de enjambre de avispas.
- Búsqueda de mono.
- Algoritmo de abeja conectora de polen (BCPA)
- Optimización por grupo de delfines (DPO)
- Optimizador de apareamiento de aves (BMO)
- Cardumen de krill (KH)
- Algoritmo de optimización de la mosca de la fruta (FOA)

Pese a la aplicación directa a los problemas de selección de los algoritmos mencionados, estos se han aplicado de forma limitada al problema de selección de parámetros en máquinas de soporte vectorial [Lu et al., 2009, Qiu et al., 2010, Zong et al., 2006].

Destaca como el más reciente, el optimizador de lobo gris [Mirjalili et al., 2014], el cual posee características que lo hacen muy atractivo [Song et al., 2015]:

- Simplicidad.

- Flexibilidad.
- Evasión de óptimos locales.
- Simple de implementar.
- Número bajo de parámetros de control a ajustar (2).
- Rápida convergencia.

En la siguiente sección se definirá el algoritmo con mayor amplitud.

## Optimizador del lobo gris

Optimizador del lobo gris (GWO) [Mirjalili et al., 2014] junto con el algoritmo optimizador de la hormiga león (ALO) [Mirjalili, 2015b] constituye uno de los algoritmos más recientes de la categoría de algoritmos de inteligencia de enjambre que ha recibido atención de la comunidad científica [Komaki and Kayvanfar, 2015, Song et al., 2015, Saremi et al., 2015, Mirjalili, 2015a].

Este algoritmo basa su funcionamiento en el comportamiento jerárquico de los lobos grises (*Canis lupus lupus*) al cazar [Muro et al., 2011]. La jerarquía está constituida por individuos ubicados en diferentes niveles que cumplen funciones específicas dentro de la manada. [Mech, 1999]

Los principales niveles y funciones son las siguientes:

- **Alfa  $\alpha$** : Líderes de la manada y pareja procreadora.
- **Beta  $\beta$** : Posibles remplazos de la pareja alfa y considerados como mandos intermedios.
- **Delta  $\delta$** : Mandos bajos que asisten en la caza.
- **Omega  $\omega$** : Últimos en la jerarquía, algunas veces fungen como niñeras.

El proceso de cacería cuenta principalmente con las siguientes fases [Muro et al., 2011]:

- Seguir, perseguir y acercarse a la presa.
- Perseguir, rodear, y acosar a la presa hasta que se detenga en movimiento.
- Atacar la presa.

### Descripción del algoritmo

#### 1. Distribución de la jerarquía:

La jerarquía de los lobos se representa mediante vectores de valores según la siguiente asignación:

- alfa ( $\alpha$ ): mejor solución.
- beta ( $\beta$ ): segunda mejor solución.
- delta ( $\delta$ ): tercera mejor solución.
- omega ( $\omega$ ): Agentes de búsqueda que recorren el espacio de búsqueda.

#### 2. Rodear la presa:

Para simular el comportamiento de los lobos al rodear la presa, se definen las siguientes ecuaciones [Mirjalili et al., 2014]:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (2.20)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (2.21)$$

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (2.22)$$

$$\vec{C} = 2 \cdot \vec{r}_2^t \quad (2.23)$$

Donde  $t$  indica la iteración actual,  $\vec{A}$  (calculado por 2.22),  $\vec{C}$  (calculado por 2.23) y  $\vec{X}_p$  es la posición de la presa.

Los componentes de  $\vec{a}$  son linealmente decrecientes de 2 a 0 durante el transcurso de las iteraciones y  $r_1$  y  $r_2$  son vectores con valores aleatorios en  $[0,1]$

El efecto de las ecuaciones 2.20 y 2.21 se observan en la figura 2.1 en el proceso conocido como cacería.

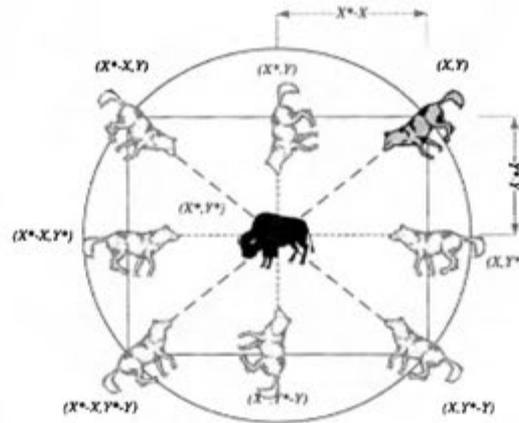


Figura 2.1: Proceso de cacería, tomado de [Mirjalili et al., 2014]

### 3. Cacería:

En el proceso de cacería de los lobos el macho alfa es el primero en aproximarse a la presa seguido por los lobos beta; sin embargo, debido a la naturaleza del problema que se trata de resolver, la posición de la presa no es conocida, por lo que se definen ecuaciones que utilizan los valores de los lobos alfa, beta y delta como aproximaciones al lugar de la presa de forma como se muestra a continuación [Mirjalili et al., 2014]:

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \quad (2.24) \quad \vec{X}_2 = \vec{X}_\beta - \vec{A}_1 \cdot (\vec{D}_\beta) \quad (2.25)$$

$$\vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \quad (2.26) \quad \vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha) \quad (2.27)$$

$$\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \quad (2.28) \quad \vec{X}_3 = \vec{X}_\delta - \vec{A}_1 \cdot (\vec{D}_\delta) \quad (2.29)$$

$$\vec{X}_{t+1} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (2.30)$$

La figura 2.2 ejemplifica el efecto de las ecuaciones en la posición de los vectores de búsqueda *lobos grises*

#### 4. Atacar la presa (explotación):

El acercamiento a la presa está modelado dentro del planteamiento matemático del problema mediante  $\vec{A}$ , el cuál decrece de de 2 a 0, por la influencia de  $\vec{a}$ .

Tomando en cuenta esto tenemos que si  $|\vec{A}| < 1$  teniendo cómo efecto que el lobo va a acercarse a la posición del lobo; en cambio si  $|\vec{A}| > 1$  el lobo se aleja de la presa, tal y cómo se muestra en la figura 2.3

5. **Buscar la presa (exploración):** El proceso empieza con la creación de una población aleatoria de lobos (omega  $\omega$ ). Luego en el curso de las distintas iteraciones, los lobos alfa, beta y delta estiman la posición posible de de la presa de

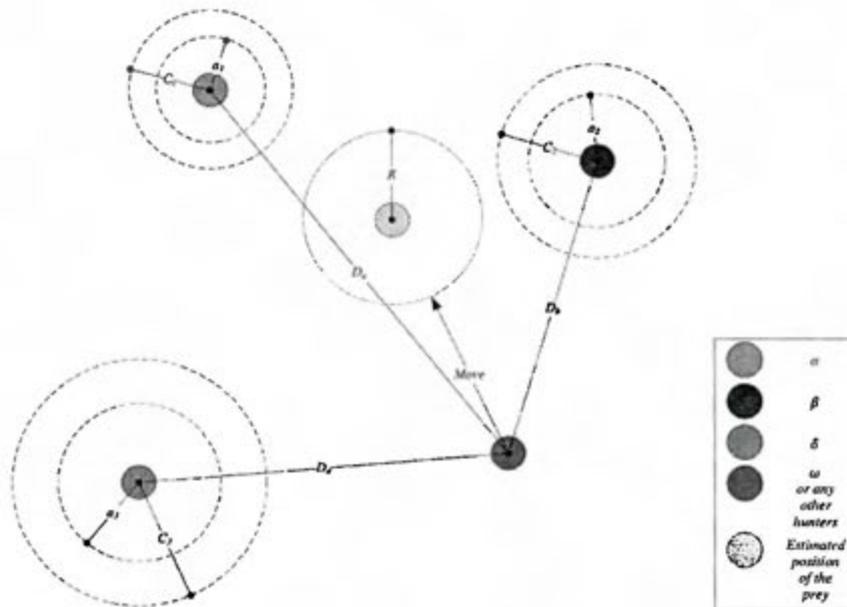


Figura 2.2: Cambio de posición durante la ejecución de GWO, tomado de [Mirjalili et al., 2014]

acuerdo a las mejores soluciones disponibles (de entre la población total de lobos). Inicialmente el proceso de exploración es influido por los valores de  $\vec{A}$ , el cual es un vector que inicia con valores reales cuyo valor absoluto es mayor a 1, ocasionando que  $|\vec{A}| > 1$  y provocando que los lobos se dispersen (como se muestra en la figura 2.3 de una presa cercana identificada para buscar otras presas que brinden mejores resultados, lo que permite que el algoritmo haga una búsqueda global y no se enfoque en óptimos locales.

También es importante notar que  $\vec{C}$  se mantiene conformado de valores aleatorios durante el transcurso de toda la ejecución del algoritmo. Permite modelar la aparición de los obstáculos que encuentran los lobos durante la cacería, con la generación de estos valores aleatorios. Estos obstáculos le impiden a los lobos acercarse directamente a la presa y con esto lograr un mecanismo adicional para evitar máximos o mínimos locales y favorecer la exploración.

El pseudo código del algoritmo se muestra en la figura 2.4 donde se muestra el uso

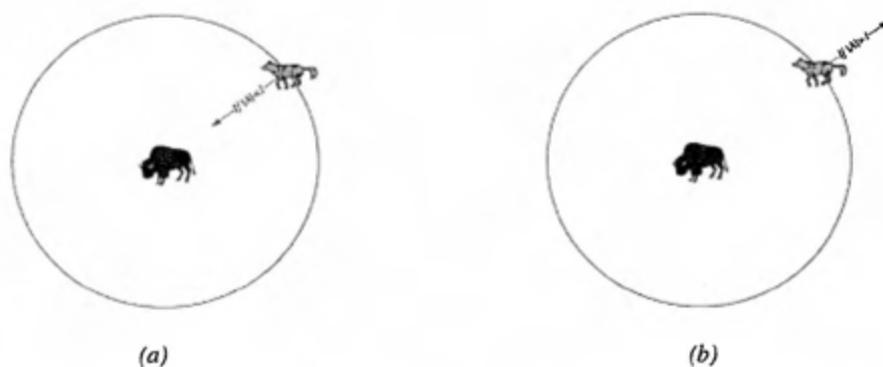


Figura 2.3: Búsqueda(b) y ataque(a) de la presa, tomado de [Mirjalili et al., 2014]

de los parámetros mencionados en las secciones anteriores. La cantidad de iteraciones permite limitar el tiempo que emplea el algoritmo en correr en caso de que no se busque el óptimo absoluto.

```

Initialize the grey wolf population  $X_i$  ( $i = 1, 2, \dots, n$ )
Initialize  $a$ ,  $A$ , and  $C$ 
Calculate the fitness of each search agent
 $X_\alpha$ =the best search agent
 $X_\beta$ =the second best search agent
 $X_\delta$ =the third best search agent
while ( $t < \text{Max number of iterations}$ )
  for each search agent
    Update the position of the current search agent
  end for
  Update  $a$ ,  $A$ , and  $C$ 
  Calculate the fitness of all search agents
  Update  $X_\alpha$ ,  $X_\beta$ , and  $X_\delta$ 
   $t=t+1$ 
end while
return  $X_\alpha$ 

```

Figura 2.4: Pseudo código del algoritmo GWO, tomado de [Mirjalili et al., 2014]

Las áreas de conocimiento mencionadas anteriormente se integrarán en esta investigación de forma que se aprovechen sus particularidades. En la siguiente sección se tratará el objetivo general, objetivos específicos así como los alcances y limitaciones de esta investigación.

## Capítulo 3

### Antecedentes

Esta investigación involucra la aplicación combinada de diversas áreas de conocimiento al análisis de sentimientos y en particular a la detección de polaridad. Por este motivo en este capítulo exploramos el trabajo realizado por los investigadores de las áreas de aprendizaje máquina, análisis de sentimientos e inteligencia de enjambre. Se da particular énfasis a los algoritmos de máquinas de soporte vectorial y optimizador de lobo gris.

#### Análisis de sentimientos

Anualmente se realiza en Alicante, España el TASS <sup>1</sup>, un taller experimental para el análisis de sentimientos enfocado en el idioma español; en el marco de la conferencia anual de La Sociedad Española para el Procesamiento del Lenguaje Natural (SEPNL)<sup>2</sup>.

A partir de su año de inicio en 2012, en cada edición los participantes envían sus experimentos que implementan para cumplir con una de las tareas asignadas en este taller.

Las tareas a desarrollar varían año con año:

- 2012 [Villena Román et al., 2013]:

1. Análisis de sentimientos.
2. Cobertura de temas tendencia (*trending topic*).

---

<sup>1</sup><http://www.daedalus.es/TASS2015/tass2015.php#about>

<sup>2</sup><http://www.sepln.org/>

- 2013 [Villena Román et al., 2014]:
  1. Análisis de sentimientos a nivel global.
  2. Clasificación de temas.
  3. Análisis de sentimientos a nivel de entidad.
  4. Identificación de tendencias políticas.
- 2014 [Villena Román et al., 2015]:
  1. Análisis de sentimientos a nivel de documento.
  2. Clasificación de temas.
  3. Detección de aspectos.
  4. Análisis de sentimientos a nivel de aspecto.
- 2015 <sup>3</sup>:
  1. Análisis de sentimientos a nivel global.
  2. Análisis de sentimientos a nivel de aspectos.

El análisis de sentimiento se puede realizar a distintos niveles: palabra, frase, oración y documento [Mouthami et al., 2013].

Con respecto al análisis de sentimientos a nivel global o de documento, que es de particular interés para esta investigación, se han registrado los resultados presentes en la tablas 3.1 y 3.2

La importancia del TASS reside en que como menciona Tromp y Pechenizkiy en 2013[Tromp and Pechenizkiy, 2013]:

*"Previous work in the area of sentiment analysis traditionally focused on benchmarking performance of sentiment classification techniques, typically*

---

<sup>3</sup><http://www.daedalus.es/TASS2015/tass2015.php>

	<b>1er resultado</b>	<b>2do resultado</b>	<b>3er resultado</b>
<b>TASS 2012</b>	71,1 %	69,0 %	69,0 %
<b>TASS 2013</b>	68,6 %	67,4 %	66,8 %
<b>TASS 2014</b>	71,0 %	71,0 %	70,0 %

Cuadro 3.1: Los tres mejores resultados las tareas de análisis de sentimientos a nivel global (con 3 clases), en las distintas ediciones del TASS

	<b>1er resultado</b>	<b>2do resultado</b>	<b>3er resultado</b>
<b>TASS 2012</b>	61,0 %	63,0 %	63,0 %
<b>TASS 2013</b>	61,0 %	60,1 %	57,6 %
<b>TASS 2014</b>	64,0 %	63,0 %	63,0 %

Cuadro 3.2: Los tres mejores resultados las tareas de análisis de sentimientos a nivel global (con 5 clases), en las distintas ediciones del TASS

*on one language only, usually English as the resources for English are best available.”*

Poca investigación se ha realizado en otros idiomas cómo el alemán [Saad and Mathiak, 2013], el español [Montejo-Ráez et al., 2013, Anta et al., 2013] o incluso multi-lenguaje[Balahur and Turchi, 2014].

En esta área de del análisis de sentimientos se distinguen las siguientes aproximaciones para la clasificación de opiniones :

- Aprendizaje de máquina (supervisado, semi-supervisado y no supervisado)[Singh et al., 2013, Mouthami et al., 2013]
- Orientación semántica [Singh et al., 2013]
- Basados en diccionarios lexicográficos (*SentiWordNet*) [Singh et al., 2013, Mouthami et al., 2013]

Particularmente en el área de aprendizaje de máquina, las máquinas de soporte vectorial se han utilizado con frecuencia dando resultados competitivos. [Mohammad et al., 2013, Zhu et al., 2013, Tromp and Pechenizkiy, 2013, Singh et al., 2013, Carstens and Toni, 2013, Bahrainian and Dengel, 2013]

El TASS se enfoca en textos cortos, extraídos de la red social *Twitter*<sup>4</sup>, lo cuál introduce retos adicionales a la detección de polaridad, pues cómo se menciona en [Kim et al., 2014] la poca cantidad de ocurrencias de las palabras dificulta la tarea.

## Selección de parámetros

Aún y cuando las máquinas de soporte vectorial han demostrado buenos resultados [Mohammad et al., 2013, Zhu et al., 2013, Tromp and Pechenizkiy, 2013, Singh et al., 2013] al ser aplicados a las tareas de clasificación de texto y análisis de sentimientos no existe una forma sistemática de escoger los parámetros [Daoud and Turabieh, 2013], lo que puede causar sub-ajuste o sobre-ajuste del modelo, afectando la precisión, el tiempo de entrenamiento [Zhao et al., 2014], la habilidad de aprender y generalizar los resultados [Wang et al., 2014].

En 2014, Wang et al lista algunos de los métodos convencionales de selección de parámetros [Wang et al., 2014]: selección empírica, búsqueda en malla (Considerada la técnica de selección por defecto [Carrizosa et al., 2014]), *gradient descent* y otros métodos basados en inteligencia de enjambre o algoritmos evolutivos.

Es importante notar que el rendimiento de los algoritmos genéticos y optimización por enjambre de partículas depende también de la selección razonable de sus respectivos parámetros [Wang et al., 2014].

Esfuerzos por resolver estos problemas incluyen el desarrollo de funciones de *kernel* sin parámetros [Daoud and Turabieh, 2013] y automatización de la selección de parámetros [Fernández Anta et al., 2012, Chen et al., 2014].

De la misma forma, constantemente se desarrollan nuevos algoritmos que disminuyan la complejidad computacional y el tiempo de ejecución del proceso de selección de parámetros. [Li et al., 2013, Wang et al., 2014, Carrizosa et al., 2014]

---

<sup>4</sup><http://twitter.com>

## Inteligencia de enjambre y Optimizador de lobo gris

El algoritmo optimizador de lobo gris fue publicado recientemente(2014)[Mirjalili et al., 2014] y desde entonces ha recibido cierta atención por ser uno de los algoritmos de inteligencia de enjambre de reciente creación [Ab Wahab et al., 2015]. Este algoritmo se ha utilizando en múltiples aplicaciones tales como el problema calendarización de flujo de ensamblaje de dos etapas [Komaki and Kayvanfar, 2015](*the two-stage assembly flow shop scheduling problem*) en el cual se utiliza el algoritmo de GWO junto al optimizador de enjambre de partículas (PSO) y simulación de templado (CSA), donde GWO tiene ligeramente menor efectividad, pero mejoras importantes de rendimiento.

En 2015, Song et al.[Song et al., 2015] utilizan GWO para estimar los parámetros en el cálculo de la dispersión de olas superficiales, con muy buenos resultados particularmente evitando mínimos locales gracias al balance de exploración y explotación, junto a una rápida convergencia y la baja cantidad de parámetros a optimizar.

Por su parte Saremi et al propone una mejora aplicando EPD (*Evolutionary Population Dynamics*) para re-colocar agentes de búsqueda deficientes en lugares cercanos a lobos con mejor rendimiento o incluso moverlos a una posición aleatoria dentro del espacio de búsqueda. Esta mejora conocida como GWO-EPD condujo a mejoras en los resultados obtenidos por GWO[Saremi et al., 2015].

En el área de aprendizaje de máquina, Mirjalili utiliza el algoritmo para el entrenamiento de *perceptrones* multi-capas (MLP)[Mirjalili, 2015a]. El autor compara los resultados de GWO contra otros algoritmos de la misma rama de inteligencia de enjambre que son ampliamente conocidos como: Optimización de enjambre de partículas (PSO), algoritmos genéticos(GA), optimización de colonia de hormigas (ACO), Estrategia evolutiva (ES) y Aprendizaje incremental basado en la población (PBIL).

El algoritmo obtuvo mejores resultados evadiendo los óptimos locales en 6 de 10

de los conjuntos de datos utilizados en el experimento, y en todos los conjuntos dio el mayor porcentaje de aciertos [Mirjalili, 2015a].

El autor hace notar que las ventajas del algoritmo son óptimas cuando la cantidad de ejemplos y características son altas, pues en caso contrario un algoritmo como *gradient descent* puede tener los resultados en una cantidad de tiempo menor.

Como se pudo notar en el las secciones anteriores, la investigación tanto en análisis de sentimientos como en herramientas para selección de parámetros sigue muy activa, aunque no se han realizado acercamientos entre los temas de optimización de proyectos y algoritmos de inteligencia de enjambre, (aunque en esta última sí se han desarrollado nuevos algoritmos).

Es por el motivo anterior que esta investigación se centrará en traer nuevamente una investigación que involucre todas las áreas, para lo cuál se plantea una metodología que permita determinar si al utilizar un algoritmo de inteligencia de enjambre se da un cambio positivo o negativo.

# Capítulo 4

## Metodología

Este capítulo describe la metodología utilizada en esta investigación, la cuál se estructuró en tres secciones distintas:

1. Creación de un modelo para estimar los parámetros de las máquinas de soporte vectorial con kernel RBF.
2. Construcción de prototipo para selección de parámetros utilizando el algoritmo optimizador del lobo gris.
3. Diseño del experimento para determinar la diferencia entre aplicar dos algoritmos de selección de parámetros de manera automatizada.

### Modelo de estimación de parámetros

El modelo de estimación de parámetros esta basado en las etapas propuestas [Chen and Zimbra, 2010] para la clasificación de textos, comprende el pre-procesamiento del corpus de datos, selección de parámetros con los algoritmos de lobo gris y búsqueda en malla, generar y evaluar los modelos de clasificación, utilizando para esto las métricas de precisión, exhaustividad y valor-F. Estas secciones se mencionan en la la figura 4.1 y que se detallarán a en las secciones subsiguientes.

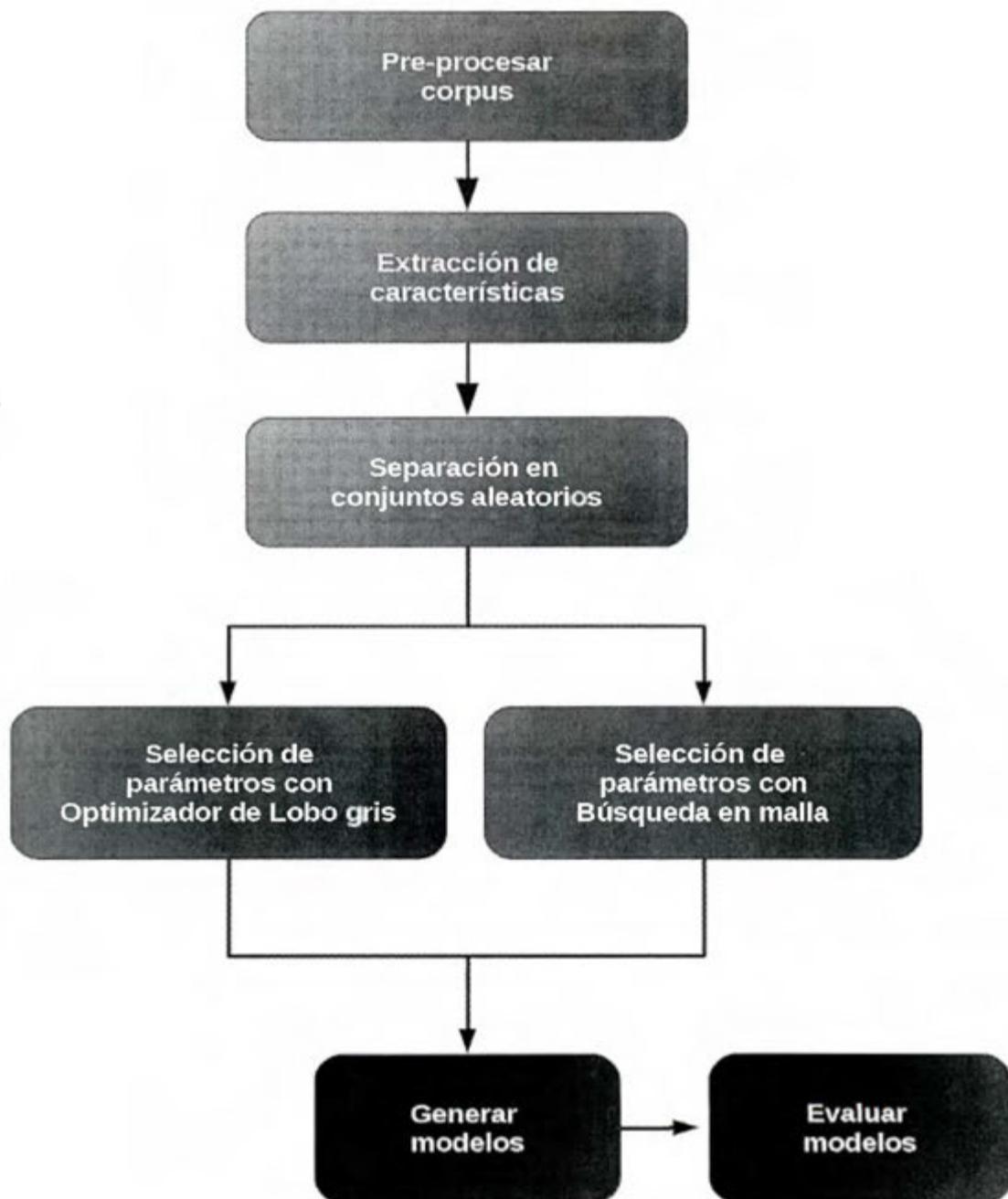


Figura 4.1: Diagrama del modelo para selección de parámetros

## Pre-procesamiento del corpus

Para que el corpus de datos pueda servir para alimentar el proceso de entrenamiento de las máquinas de soporte vectorial y crear un modelo de clasificación cada registro se debe convertir en un vector de valores numéricos. Para esto el corpus se debe simplificar para utilizar solo dos características para utilizar solo la formulación de dos clases de las máquinas de soporte vectorial. Luego se realiza el procedimiento de extracción de características, el cuál esta fuera del alcance de esta investigación, pero que sin embargo no afecta el modelo propuesto.

Para asegurar validez estadística en los resultados se preparan 60 subconjuntos del mismo tamaño del corpus de datos, pero estos estarán a su vez subdivididos en dos: un conjunto que se utilizará para generar el modelo de clasificación y el otro conjunto se utilizará para evaluar el rendimiento del modelo de clasificación. Los registros asignados a cada subconjunto se asignarán de manera aleatoria a partir de los registros disponibles en el corpus principal

## Selección de parámetros

Para el proceso de selección de parámetros se utilizan 5 distintos tratamientos: Parámetros por defecto (*default*), búsqueda en malla (*grid search*) y tres distintos tratamientos para el optimizador de lobo gris (*GWO 56*, *GWO 112*, *GWO 168*).

## Generación y evaluación de los modelos

Una vez seleccionado los parámetros para cada subconjunto (con sus 5 tratamientos) se deben generar modelos de clasificación utilizando estos valores, creando un total de 120 modelos distintos. Luego de la evaluación de los modelos, cada uno de estos le fue aplicado a los 60 subconjuntos de validación.

## Herramientas

Al ejecutar el experimento en la etapa de procesamiento del corpus, se utilizó el *framework* de desarrollo WEKA [Hall et al., 2009]. WEKA es una herramienta de software libre distribuida bajo la licencia *gnu general public license (GNU GPL)*, que permite ejecutar distintos algoritmos de *machine learning* y aplicarlos a un *dataset* particular o integrar los algoritmos en aplicaciones que utilicen java. Este software es extensible [Hall et al., 2009], y permite la manipulación de los datos mediante filtros y *plugins* desarrollados en JAVA, los cuales se utilizaron en la etapa de pre procesamiento de datos.

Por otro lado, la herramienta LIBSVM [Chang and Lin, 2011] se utilizó para completar las etapas de selección de parámetros y generación y evaluación de los modelos, con base en las instrucciones establecidas en *la guía práctica para clasificación con SVM* distribuida a través del sitio web del proyecto <sup>1</sup>.

Finalmente, se desarrolló un *script* en *Perl 6*<sup>2</sup> para implementar el algoritmo optimizador del lobo gris, el cual esta disponible a través del sitio web [github.com](https://github.com)<sup>3</sup>, junto a una serie de *scripts* también en Perl 6 para apoyar cada uno de los pasos propuestos en el modelo, también disponibles en el sitio web mencionado.<sup>4</sup>

## Experimento

El trabajo alrededor de este experimento se estructuró de acuerdo al modelo propuesto en la sección anterior y será detallado en las secciones siguientes:

<sup>1</sup><http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>

<sup>2</sup><http://perl6.org>

<sup>3</sup><https://github.com/Sufrostico/perl6-gwo>

<sup>4</sup><https://github.com/Sufrostico/perl6-svmexperiment>

## Pre procesamiento del corpus.

Como conjunto de datos a trabajar se utilizó el corpus anotado de TASS 2015 <sup>5</sup>.

El TASS <sup>6</sup> es un taller experimental de evaluación que promueve la aplicación de algoritmos y técnicas desarrollados para el área de análisis de sentimientos aplicados a textos cortos extraídos de la red social Twitter.

Este corpus es reconocido y utilizado por los investigadores del área [Moreno-Ortiz and Hernández, 2012, Anta et al., 2013, Montejo-Ráez et al., 2013, Villena Román et al., 2013, Villena Román et al., 2015] y particularmente por el grupo de Investigación en Análisis de Sentimientos, parte del CITIC<sup>7</sup>. En este corpus cada tweet esta anotado con una de seis posibles etiquetas que identifican el grado de polaridad global: positivo (P), muy positivo (P+), neutral (NEU), sin polaridad (NONE), negativos (N) y muy negativos (N+).

Inicialmente se transformó la información del formato XML original a un archivo separado con comas. A partir de este formato se cargó el conjunto de datos al formato nativo de WEKA, denominado ARFF<sup>8</sup>. Una vez en este formato se utilizaron los filtros provistos por la herramienta para ejecutar las siguientes acciones:

1. Eliminar el atributo *AGREEMENT* de todos los registros.

Filtro utilizado:

```
Filters > unsupervised > attribute
Remove -R 2
```

2. Eliminar los registros etiquetados con "*polaridad neutral*" *NEU*.

Filtro utilizado:

```
Filters > unsupervised > instance
```

<sup>5</sup><http://www.daedalus.es/TASS2015/tass2015.php#corpus>

<sup>6</sup><http://www.daedalus.es/TASS2015/tass2015.php>

<sup>7</sup><http://www.citic.ucr.ac.cr/>

<sup>8</sup><http://www.cs.waikato.ac.nz/ml/weka/arff.html>

```
RemoveWithValues -S 0.0 -C last -L 1
```

3. Eliminar los registros etiquetados como "*sin polaridad*" *NEU*.

Filtro utilizado:

```
Filters > unsupervised > instance
RemoveWithValues -S 0.0 -C last -L 1
```

4. Unir los registros etiquetados como "*muy positivos*" (*P+*) y "*positivos*" (*P*) bajo una sola etiqueta *P*.

Filtro utilizado:

```
Filters > unsupervised > attribute
MergeManyValues -C last -L P -R 3,5
```

5. Unir los registros etiquetados como "*muy negativos*" (*N+*) y "*Negativos*" (*N*) bajo una sola etiqueta *N*.

Filtro utilizado:

```
Filters > unsupervised > attribute
MergeManyValues -C last -L N -R 3,5
```

Al finalizar el pre procesamiento del corpus, este quedó con 5086 registros divididos de la siguiente forma: 2888 registros etiquetados como positivos (*P*) y 2180 registros etiquetados como negativos (*N*) (ver figura 4.2). Luego se extrajeron 60 conjuntos, compuesto cada uno por dos subconjuntos, un subconjunto de entrenamiento con un 60% de los registros totales presentes en el corpus procesado y un subconjunto de validación con un 40% de los registros totales (ver figura 4.3). El tamaño de los subconjuntos fue seleccionado siguiendo un criterio de oportunidad. Estos son lo suficientemente grandes para minimizar problemas de ajuste en el entrenamiento y para que las métricas utilizadas contemplen la capacidad de generalización del modelo generado en la etapa de evaluación.

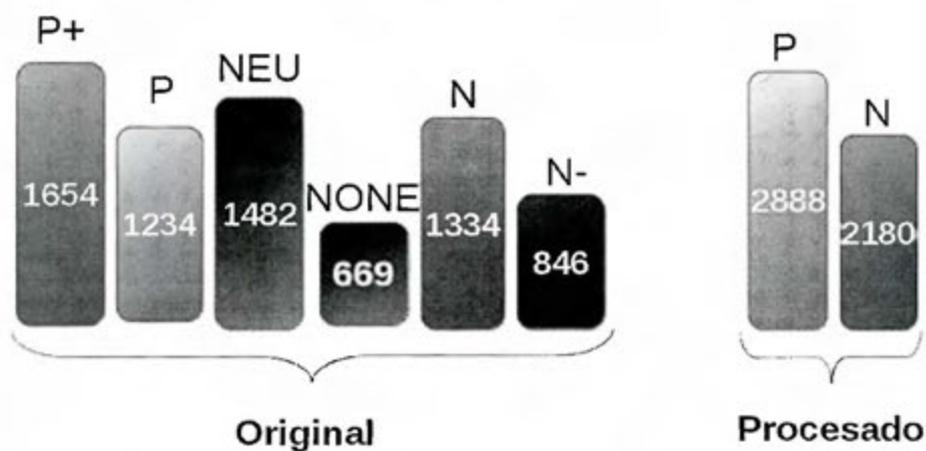


Figura 4.2: Distribución de las instancias entre las clases, a la izquierda la distribución original y a la derecha la distribución luego del pre-procesamiento.



Figura 4.3: División del corpus procesado en 60 subconjuntos extraídos aleatoriamente



Figura 4.4: Vector de rangos de probabilidades

## Extracción de características.

En esta etapa se reutilizó la extracción de características implementada por el se utilizó el procedimiento utilizado por el grupo de investigación en análisis de sentimientos, el cual es parte parte del CITIC<sup>9</sup>. Este proceso llevó a cabo mediante los siguientes pasos:

1. Se toman los textos y se separan en palabras
2. Se determina la probabilidad de que una palabra sea positiva o negativa usando el diccionario de polaridad construido como parte Trabajo Final de Investigación Aplicada (TFIA) de Mario Solís [Solís Parajeles, 2015], estudiante de la Maestría Profesional en Computación e Informática <sup>10</sup>.
3. Tomando en cuenta la polaridad asignada a cada palabra en el punto anterior se crea un vector  $\vec{A}$ , donde  $i$  corresponde a un rango de probabilidades entre -1 y 1 de ancho 0.10 cómo se muestra en figura 4.4 y  $a_i$  = cantidad de palabras en el rango
4. Se crean cuatro vectores adicionales utilizando marcadores de énfasis (mayúsculas, repetición de sílabas, repetición de caracteres y texto normalizado), los cuales son parte del trabajo del profesor Edgar Casasola.[Casasola et al., 2016]
5. Se juntan todos los vectores para crear el vector de características con una dimensionalidad total de 100 características.

## Entrenamiento del modelo.

El entrenamiento del modelo se llevó a cabo en dos etapas:

1. Selección de los parámetros por medio de búsqueda en malla y optimizador de lobo gris.

<sup>9</sup><http://www.citic.ucr.ac.cr/>

<sup>10</sup><http://pci.ucr.ac.cr/carrera/maestría-profesional-en-computación-e-informática>

Variable	Límite inferior	Límite superior	Paso
C	-5	12	2
$\gamma$	-15	3	2

Cuadro 4.1: Rangos de valores por defecto en la herramienta svm-grid expresados utilizando una escala logarítmica( $\log_2$ )

## 2. Entrenamiento de los modelos.

La selección de parámetros se ejecutó sobre los 60 subconjuntos de datos generados, a cada subconjunto se le aplicó la selección de parámetros utilizando los valores por defecto especificados por el software LIBSVM, búsqueda en malla y Lobo gris.

### Parámetros por defecto

Los parámetros por defecto a utilizar son los especificados por la herramienta LIBSVM y WEKA. Estos asignan el valor de 1 al parámetro C y de  $\frac{1}{\text{número de registros}}$  al parámetro  $\gamma$ .

En el caso del corpus utilizado en esta investigación el valor de  $\gamma$  corresponde a:

$$\frac{1}{\text{número de registros}} = \frac{1}{5086} = 0,000196618167 \quad (4.1)$$

### Búsqueda en malla

Para realizar la selección de parámetros por medio de búsqueda en malla se estructura el trabajo con base en las pautas para establecidas por [Luts et al., 2010] y utilizando la herramienta LIBSVM [Chang and Lin, 2011] con base en las instrucciones establecidas en *la guía práctica para clasificación con SVM* distribuida a través del sitio web del proyecto <sup>11</sup>

Se aplicó búsqueda en malla a los 60 subconjuntos de entrenamiento, utilizando los rangos de valores por defecto que provee la herramienta svm-grid incluida dentro de la distribución de LIBSVM los cuales corresponden a el cuadro 4.1. Para cada par de valores seleccionados por el algoritmo se evaluó su desempeño utilizando *10-fold cross validation*.

### Optimizador del lobo gris

A diferencia del algoritmo de búsqueda en malla, este algoritmo tiene como parámetros la cantidad de iteraciones a realizar y la cantidad de lobos. Para esta investigación

<sup>11</sup><http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>

	GWO 56	GWO 112	168
Cantidad de lobos	4	4	4
Cantidad de iteraciones	14	28	42

Cuadro 4.2: Configuración de los parámetros para los tratamientos correspondientes al algoritmo del lobo gris

se fijó el número de lobos en 4, lo cual corresponde al número de núcleos disponibles en la máquina. En cuanto a la cantidad de iteraciones, se fijó de modo que la cantidad total de entrenamientos mantuviera una relación de 50 % menos, igual y 50 % más que los empleados por la búsqueda en malla (ver cuadro 4.2).

Cada uno de los 60 subconjuntos de entrenamiento se utilizó para generar un modelo, para luego ser evaluado utilizando los 60 subconjuntos de validación.

La función de *fitness* para ambos algoritmos está dada por la evaluación del desempeño de los modelos de clasificación utilizando el valor de *accuracy*, el cuál consiste en la fracción de documentos clasificados correctamente por el clasificador (ver ecuación 4.2), utilizando en el entrenamiento *10-fold cross validation*.

$$\text{accuracy} = \frac{V_p + V_n}{V_p + F_p + F_n + V_n} \quad (4.2)$$

donde  $V_p$  son los verdaderos positivos,  $F_p$  falsos positivos,  $F_n$  falsos negativos y  $V_n$  verdaderos negativos.

### Generación de los modelos de clasificación

Luego de haber seleccionado los parámetros se procedió a generar los modelos de clasificación. Por cada uno de los subconjuntos de entrenamiento se generaran 5 modelos de acuerdo a los 5 tratamientos utilizados para la selección de parámetros: *Default*, *Malla*, *GWO 56*, *GWO 112*, *GWO 168*.

### Evaluar los modelos generados.

El proceso de clasificación dará como resultado instancias marcadas con una etiqueta asignada según el modelo de clasificación utilizado. Al usar los subconjuntos de validación se pueden identificar tanto las instancias clasificadas correctamente como las clasificadas incorrectamente (ver figura 4.5).

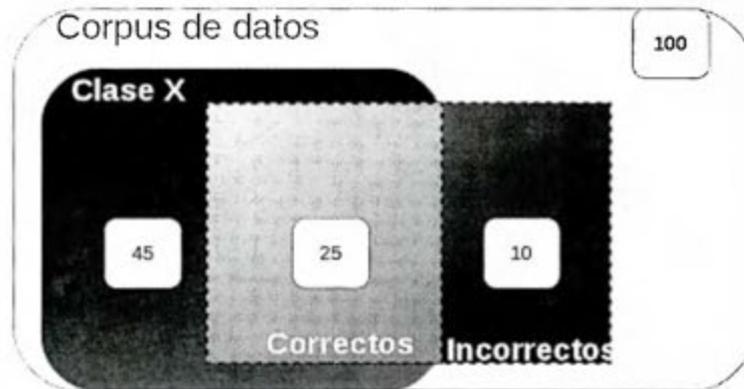


Figura 4.5: Resultado de ejemplo de una clasificación

Para evaluar y comparar los resultados se recurrirá a las métricas utilizadas en el TASS [Villena Román et al., 2013] definidas según las siguientes ecuaciones:

**Precisión:**

$$\text{Precisión} = \frac{\text{clasificados correctamente}}{\text{total de clasificaciones}} \quad (4.3)$$

Para una representación gráfica ver la figura 4.6

La representación gráfica muestra un rectángulo gris con el número "25" en el numerador, una línea horizontal de división, un rectángulo negro con el número "45" en el denominador, un signo de igualdad "=" y un recuadro con el resultado "0,625".

Figura 4.6: Representación gráfica de la métrica de precisión

**Exhaustividad:**

$$\text{Exhaustividad} = \frac{\text{documentos obtenidos}}{\text{todos los documentos}} \quad (4.4)$$

Para una representación gráfica ver la figura 4.7

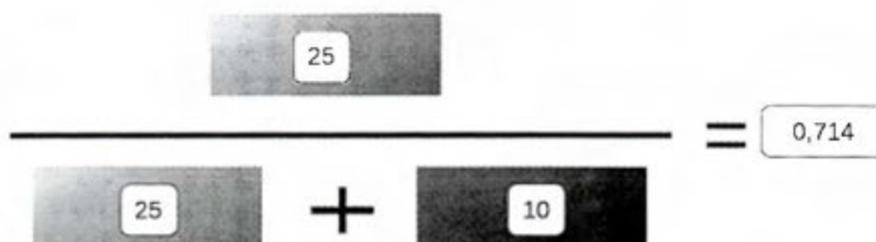


Figura 4.7: Representación gráfica de la métrica de exhaustividad

Valor-F:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (4.5)$$

Para una representación gráfica ver la figura 4.8

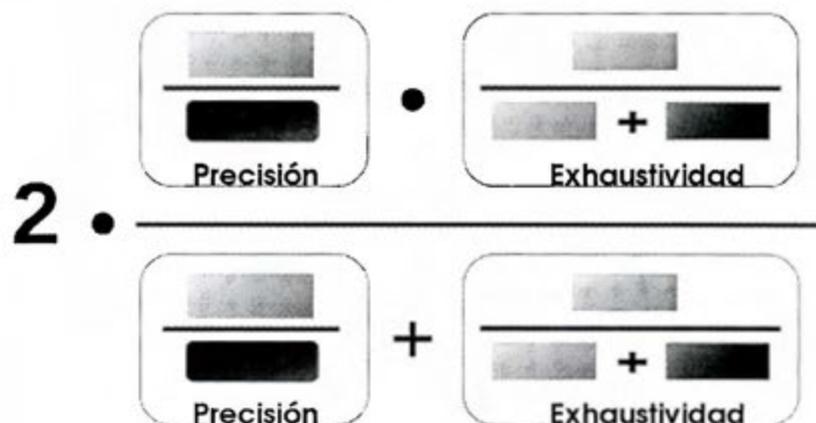


Figura 4.8: Resultado de ejemplo de una clasificación

Estas métricas son el resultado de evaluar los modelos de clasificación generados en la etapa anterior utilizando la herramienta LIBSVM con los 60 subconjuntos de validación (ver figura 4.10). En la figura 4.9 se muestra un resumen del proceso descrito en las secciones anteriores.

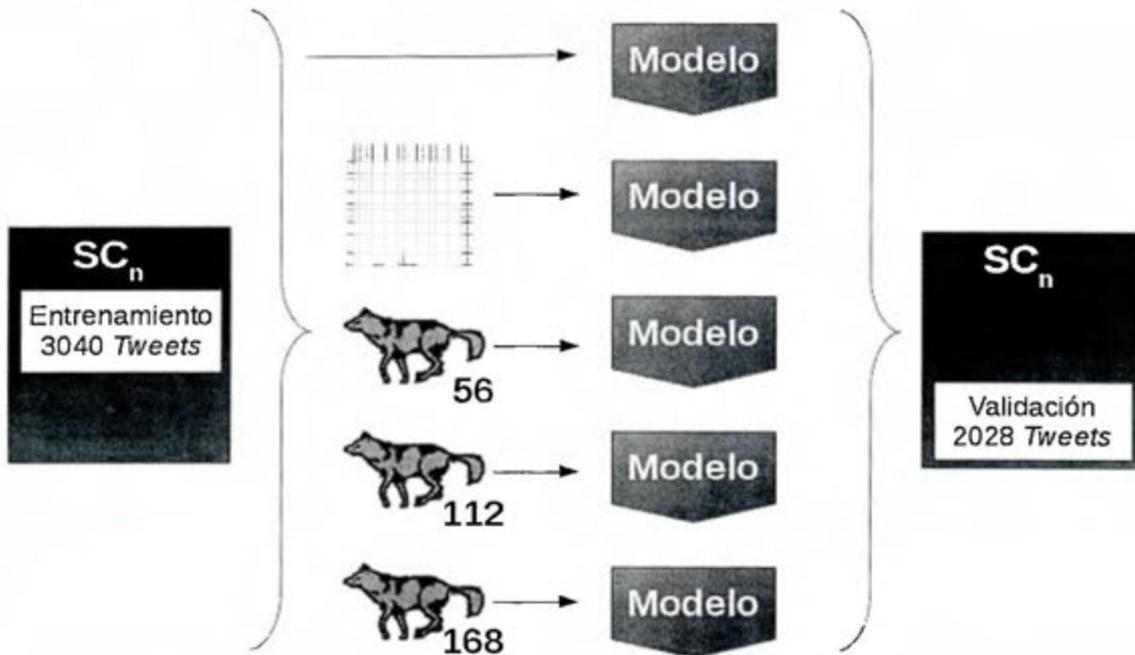


Figura 4.9: Generar y evaluar los modelos

```

21 gwo-168-9
22 /home/sufrostico/Programas/libsvm-3.20/svm-predict
23 binary_class_predict
24 Accuracy = 69.5266% (1410/2028)
25 Precision = 66.1194% (443/670)
26 Recall = 53.1175% (443/834)
27 F-score = 0.589096

```

Figura 4.10: Ejemplo de resultado de evaluación de un modelo de clasificación

# Capítulo 5

## Resultados y conclusiones

### Resultados

Los resultados de evaluar los modelos de clasificación generados a partir de la selección de parámetros (ver apéndice C, para las tablas completas de resultados) utilizando las métricas establecidas (Ver apéndice A para las tablas de resultados completas), se resumen mediante el uso de diagramas de cajas y valores tabulares. Los diagramas de cajas, también conocidos como diagramas de cajas y bigotes (box-and-whiskers diagram) se utilizan en estadística descriptiva como una forma conveniente de comparar conjuntos de datos de acuerdo a sus percentiles.

Los gráficos se generaron utilizando *GNU Octave*<sup>1</sup>. Cada gráfico cuenta con una caja por tratamiento, el extremo superior de la caja representa el valor que tomaron el 75 % de los valores de resultado, al contrario, el extremo inferior representa el valor del 25 % de los resultados. Al medio de la caja se presenta una línea que representa la media, y a cada uno de los extremos se extienden verticalmente dos rayas (bigotes) representando los valores máximo (arriba) y mínimo (abajo). Los valores considerados como atípicos se presentan como puntos independientes. Al utilizar estos gráficos se puede visualizar con facilidad el grado de dispersión y la asimetría de los datos.

---

<sup>1</sup><https://www.gnu.org/software/octave/>

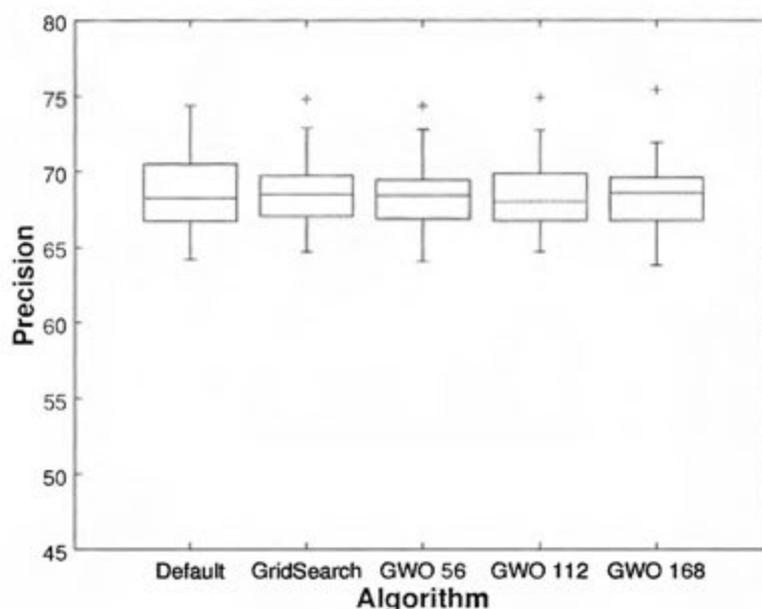


Figura 5.1: Diagrama de cajas para la métrica de Precisión

Para los valores de precisión, los resultados rondaron entre 63.80 % y 75.41 % con las medias alrededor del 68.5 %. Las desviaciones estándar obtuvieron valores pequeños, entre 2.2 y 2.4. (ver cuadro 5.1), mostrando una dispersión similar entre los 60 conjuntos de datos (ver figura 5.1)

Cuadro 5.1: Valores de mínimo, máximo, media y desviación estándar para la métrica de Precisión

	Default	Grid Search	GWO 56	GWO 112	GWO 168
<b>Mínimo</b>	64,2029	64,6753	64,0565	64,6985	63,8083
<b>Máximo</b>	74,3750	74,7720	74,3590	74,8865	75,4173
<b>Media</b>	68,4652	68,6490	68,5008	68,5049	68,4855
<b>Desviación Estándar</b>	2,2692	2,2497	2,3575	2,3947	2,2988

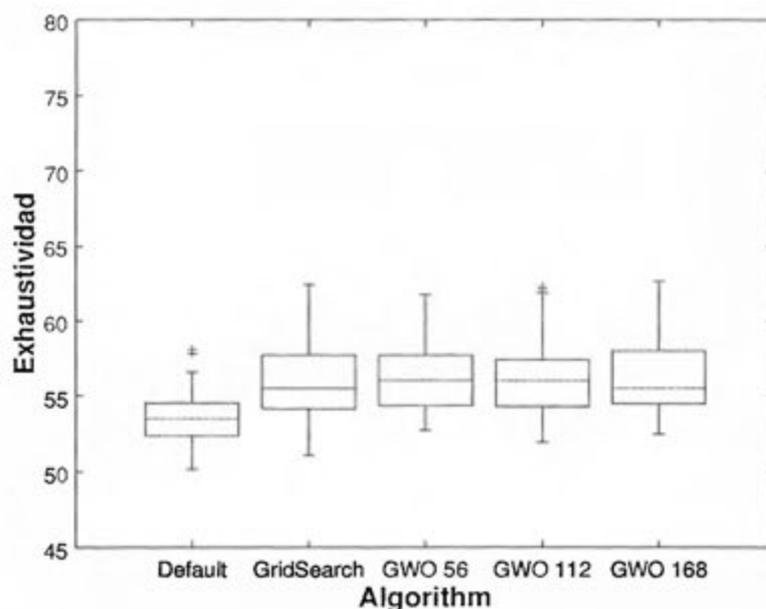


Figura 5.2: Diagrama de cajas para la métrica de exhaustividad

En el caso de la exhaustividad, los valores resultaron más bajos que en la precisión, rondando entre 51 % y 63 %. Es importante hacer notar que aunque en el diagrama de cajas (ver figura 5.2) se presenta una diferencia aparente entre el tratamiento de los parámetros por defecto (*Default*) y los otros tratamientos esta cae dentro del rango de la desviación estándar (ver cuadro 5.2).

Cuadro 5.2: Análisis estadístico de Recall

	Default	Grid Search	GWO 56	GWO 112	GWO 168
Mínimo	50,1665	51,0544	52,7253	51,9423	52,4972
Máximo	58,1032	62,4153	61,7381	62,3025	62,6411
Media	53,5394	55,9498	56,2710	56,2573	56,2569
Desviación Estándar	1,8426	2,4591	2,1447	2,4456	2,3237

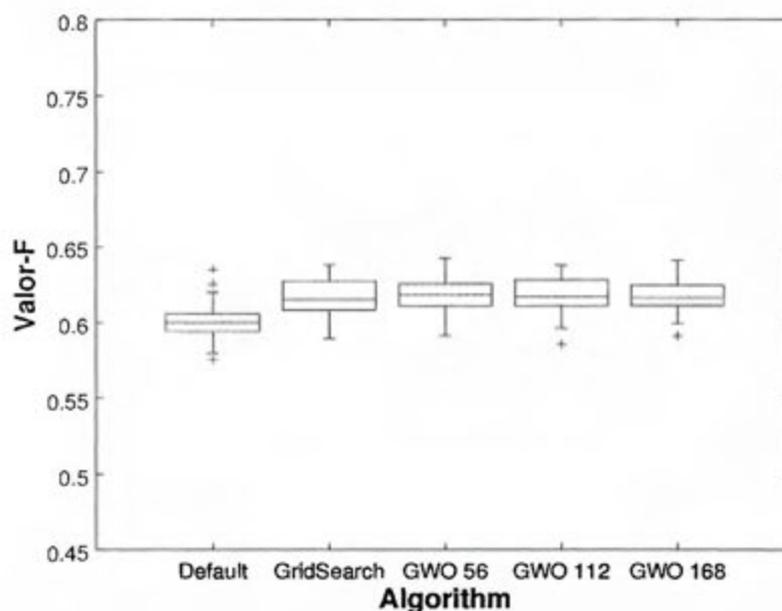


Figura 5.3: Diagrama de cajas de F-Score

La métrica del valor-F mantiene el mismo comportamiento con valores entre 0,57 y 0,64 y una desviación estándar entre 0,0108 y 0,0126 (ver cuadro 5.3).

Cuadro 5.3: Análisis estadístico de F-Score

	Default	Grid Search	GWO 56	GWO 112	GWO 168
Mínimo	0,5753	0,5894	0,5911	0,5853	0,5909
Máximo	0,6350	0,6382	0,6428	0,6380	0,6409
Media	0,6005	0,6159	0,6173	0,6171	0,6171
Desviación Estándar	0,0118	0,0126	0,0108	0,0123	0,0114

## Discusión adicional

Aún cuando esta investigación centra su interés en la detección de polaridad, aplicada al análisis de sentimientos, el modelo propuesto para seleccionar de forma automatizada de los parámetros de las máquinas de soporte vectorial que utilicen una función de kernel RBF, se puede aplicar a otras áreas de conocimiento como la medicina, ya que el proceso es independiente de corpus utilizado o de la selección de características que se aplique.

Dado que el proceso de selección de parámetros generó valores distintos para los mismos conjuntos de datos con el mismo tratamiento, esto pareciera indicar que el rendimiento del clasificador es independiente del modelo de clasificación. Esto puede obedecer a distintos motivos, entre ellos: una superposición de las categorías producto de la extracción de características utilizada en la etapa inicial. Debido a esta posible superposición los modelos generados no podrían ser capaces de clasificar correctamente todas las instancias. Para determinar si la superposición mencionada existe, se recomienda realizar un análisis de sensibilidad sobre los parámetros de las máquinas de soporte vectorial, el uso de una mayor cantidad de categorías y la utilización de diferentes corpus (de medicina, recuperación de la información u otros).

Al trabajar con más categorías se debe tomar en cuenta que la cantidad de instancias disponibles para cada categoría en el corpus de entrenamiento debe ser significativa, de no ser así se puede incurrir en errores de ajuste. Por ejemplo, en el corpus utilizado en esta investigación (TASS2015), la inclusión de la categoría para instancias con polaridad neutra (NEU) debe ser considerada con precaución pues esta solo se cuentan con 669 instancias. En casos como el anterior, la reducción de la dimensionalidad del modelo también debe considerarse.

Otras opciones para mejorar resultados es el mezclar las máquinas de soporte vectorial con otros mecanismos de clasificación de manera escalonada, utilizando cada escalón para clasificar una o más categorías dependiendo de las particularidades de esta. Este mecanismo de clasificación no se implementó durante esta investigación por estar fuera del alcance de la misma.

Por último, en esta investigación se fijaron los parámetros de la cantidad de lobos y la cantidad de iteraciones en el algoritmo optimizador del lobo gris, de forma que se pudiera comparar por medio de la cantidad de entrenamientos de C-SVM contra el algoritmo de búsqueda en malla. Sin embargo, se considera conveniente plantear un experimentos que permitan analizar si el rendimiento del algoritmo posee alguna sensibilidad a estos parámetros.

## Conclusiones

Durante la investigación se generó e implementó un modelo para estimar los parámetros de las máquinas de soporte vectorial, el cuál puede ser aplicado a otros corpus o incluso a otros algoritmos, cumpliendo de esta forma con el primer objetivo específico. También se desarrolló una implementación del algoritmo de optimizador de lobo gris para su uso con máquinas de soporte vectorial cumpliendo con el segundo objetivo específico.

Los resultados obtenidos al evaluar los modelos generados basados en los parámetros seleccionados por los distintos tratamientos muestran que los valores predeterminados de la herramienta LIBSVM para  $C$  y  $\gamma$  obtienen resultados muy similares a los otros tratamientos. Considerando que los distintos tratamientos seleccionan parámetros diferentes para el mismo sub conjunto de trabajo se considera que estos resultados son un posible efecto de la superposición de las clases P y N, teniendo como efecto que los distintos modelos generados tengan poco efecto en el resultado final de la clasificación.

Para contrarrestar esta superposición se sugiere la exploración de distintos mecanismos de extracción de características, la utilización de otros corpus de datos en áreas tales como al medicina o la recuperación de la información, con el objetivo de variar la distribución de las instancias y un profundo análisis de las particularidades de las instancias mal clasificadas. Por otro lado, los resultados de evaluar los modelos se mantienen en el rango de valores obtenidos durante los últimos tres años en el Workshop de TASS, los cuales en la tarea de detección de polaridad global (con tres clases: P, NEU y N) se mantuvieron en el rango entre el 66.8% y el 71% (ver cuadro 3.1).

Finalmente, en esta caso específico se puede observar que si se toman en cuenta el rendimiento y la calidad de los resultados, el uso de los valores por defecto de la herramienta LIBSVM, para los parámetros de las máquinas de soporte vectorial (C-SVM) son una opción viable. Los otros algoritmos utilizados requieren de muchos recursos y no proveen una mejora significativa en los resultados del clasificador.

# A P É N D I C E S

## Apéndice A

# Tablas de resultados de las métricas de evaluación

Cuadro A.1: Resultados de precisión

Default	Malla	GWO 56	GWO 110	GWO 168
67,8521	68,0990	67,8851	67,3997	69,5965
67,8521	69,0000	69,1429	69,4168	69,2745
65,9514	64,6753	68,3453	65,6812	68,0057
68,8791	69,8251	69,9856	69,2857	70,5539
67,4716	67,5427	67,8571	69,4524	69,8551
69,2082	69,4203	69,3410	69,2641	69,0100
67,1703	66,4516	67,5100	64,8418	67,8812
65,9733	66,1519	66,6223	66,9786	67,0683
66,6667	68,2678	67,0199	67,9115	68,3140
65,4646	68,7055	68,7853	67,2857	68,3688
66,9986	65,4592	65,2850	66,6207	66,5786
72,3343	70,9091	71,9764	71,1111	70,4670
66,6191	66,7598	66,7582	65,9151	66,6205
68,0660	69,0856	68,8953	68,6695	69,1977
67,6923	65,6410	64,0565	64,6985	63,8083
71,9939	71,6176	71,3256	71,9298	71,3663
66,2500	65,2812	64,2080	65,5758	67,9452
68,0441	65,2893	67,0343	65,1712	65,6028
68,6391	69,2194	66,4828	67,0270	69,4323
65,4728	68,1034	66,9986	67,0940	67,9654
70,4819	69,0934	69,1034	69,3593	68,7415
70,4819	70,6821	70,9630	70,8029	70,9489
68,2353	68,2138	68,1818	68,0328	67,8371

66,5248	68,3068	68,4438	68,0456	66,8016
70,7246	72,8592	71,9828	72,7273	71,9247
67,6901	69,0476	69,1630	66,7112	66,6667
64,2029	66,1850	65,8009	65,2677	65,9420
65,5028	65,1223	65,7030	65,2618	64,0547
70,8459	70,6475	69,7802	72,1893	70,8696
70,6250	69,6970	69,1114	69,8507	69,6613
66,0000	66,8928	67,7419	66,8022	66,3185
68,3140	68,3784	67,7763	67,6316	68,5792
66,8000	67,4003	65,6527	67,2627	65,8599
72,1440	71,3467	72,7952	69,5531	69,5833
68,2857	70,5710	70,7670	70,5710	69,2847
69,3314	69,3431	69,2641	69,9854	69,5588
68,2563	68,7135	65,0649	68,9706	68,9504
68,7688	67,0683	66,8901	66,7557	66,9291
69,0658	67,3759	68,7225	68,8141	67,1788
74,3750	74,7720	74,3590	74,8865	75,4173
70,4819	69,0934	69,1034	69,3593	68,7415
70,4819	70,6821	70,9630	70,8029	70,9489
68,2353	68,2138	68,1818	68,0328	67,8371
66,5248	68,3068	68,4438	68,0456	66,8016
70,7246	72,8592	71,9828	72,7273	71,9247
67,6901	69,0476	69,1630	66,7112	66,6667
64,2029	66,1850	65,8009	65,2677	65,9420
65,5028	65,1223	65,7030	65,2618	64,0547
70,8459	70,6475	69,7802	72,1893	70,8696
70,6250	69,6970	69,1114	69,8507	69,6613
66,0000	66,8928	67,7419	66,8022	66,3185
68,3140	68,3784	67,7763	67,6316	68,5792
66,8000	67,4003	65,6527	67,2627	65,8599
72,1440	71,3467	72,7952	69,5531	69,5833
68,2857	70,5710	70,7670	70,5710	69,2847
69,3314	69,3431	69,2641	69,9854	69,5588
68,2563	68,7135	65,0649	68,9706	68,9504
68,7688	67,0683	66,8901	66,7557	66,9291
69,0658	67,3759	68,7225	68,8141	67,1788

74,3750	74,7720	74,3590	74,8865	75,4173
---------	---------	---------	---------	---------

Cuadro A.2: Resultados de exhaustividad

Default	Malla	GWO 56	GWO 110	GWO 168
54,1430	59,4318	59,0909	59,2045	54,8864
54,1430	54,8241	54,9376	55,3916	55,2781
53,5424	57,8397	55,1684	59,3496	55,0523
52,2956	53,6394	54,3113	54,3113	54,1993
54,9133	59,4220	59,3064	55,7225	55,7225
54,0046	54,8055	55,3776	54,9199	55,0343
55,8219	58,7900	57,8767	60,8447	57,4201
51,6880	59,8370	58,3236	58,3236	58,3236
53,1215	53,2350	57,4347	55,7321	53,3485
54,3779	55,6452	56,1060	54,2627	55,5300
54,3253	58,3622	58,1315	55,7093	58,1315
56,5953	57,1590	55,0169	57,7227	57,8354
55,5952	56,9048	57,8571	59,1667	57,2619
50,5006	52,9477	52,7253	53,3927	53,7264
58,1032	61,4646	59,9040	61,8247	60,7443
53,5068	55,0905	55,9955	55,6561	55,5430
54,7646	61,3088	60,9644	62,1125	56,9460
55,7562	62,4153	61,7381	62,3025	62,6411
53,5797	54,2725	55,6582	57,2748	55,0808
53,9551	55,9622	55,6080	55,6080	55,6080
52,4076	57,8826	57,6525	57,3072	58,4580
52,4076	54,5353	53,6394	54,3113	54,4233
53,0892	55,4920	56,6362	56,9794	55,2632
54,5984	54,9476	55,2969	55,5297	57,6251
54,5251	56,0894	55,9777	56,3128	55,5307
53,1573	53,2721	54,0758	57,2905	57,1757
52,1176	53,8824	53,6471	53,0588	53,5294
53,9701	58,2278	57,5374	58,8032	59,2635
51,7660	54,1943	56,0706	53,8631	53,9735
50,1665	51,0544	54,3840	51,9423	52,4972

55,2632	58,9713	57,7751	58,9713	60,7656
53,5918	57,6967	58,0388	58,6089	57,2406
57,8522	56,5820	59,8152	56,4665	59,6998
50,8269	54,9063	53,6935	54,9063	55,2370
56,2353	56,7059	57,5294	56,7059	58,1176
53,4155	53,1915	53,7514	53,5274	52,9675
52,7650	54,1475	57,7189	54,0323	54,4931
52,1047	56,9966	56,7691	56,8828	58,0205
51,0759	53,7939	53,0011	53,2276	54,4734
52,4807	54,2448	54,3550	54,5755	54,7960
52,4076	57,8826	57,6525	57,3072	58,4580
52,4076	54,5353	53,6394	54,3113	54,4233
53,0892	55,4920	56,6362	56,9794	55,2632
54,5984	54,9476	55,2969	55,5297	57,6251
54,5251	56,0894	55,9777	56,3128	55,5307
53,1573	53,2721	54,0758	57,2905	57,1757
52,1176	53,8824	53,6471	53,0588	53,5294
53,9701	58,2278	57,5374	58,8032	59,2635
51,7660	54,1943	56,0706	53,8631	53,9735
50,1665	51,0544	54,3840	51,9423	52,4972
55,2632	58,9713	57,7751	58,9713	60,7656
53,5918	57,6967	58,0388	58,6089	57,2406
57,8522	56,5820	59,8152	56,4665	59,6998
50,8269	54,9063	53,6935	54,9063	55,2370
56,2353	56,7059	57,5294	56,7059	58,1176
53,4155	53,1915	53,7514	53,5274	52,9675
52,7650	54,1475	57,7189	54,0323	54,4931
52,1047	56,9966	56,7691	56,8828	58,0205
51,0759	53,7939	53,0011	53,2276	54,4734
52,4807	54,2448	54,3550	54,5755	54,7960

Cuadro A.3: Resultados de F-Score

Default	Malla	GWO 56	GWO 110	GWO 168
0,6023	0,6347	0,6318	0,6304	0,6137

0,6023	0,6110	0,6123	0,6162	0,6149
0,5910	0,6107	0,6105	0,6236	0,6085
0,5945	0,6067	0,6116	0,6089	0,6130
0,6055	0,6322	0,6329	0,6183	0,6199
0,6067	0,6125	0,6158	0,6126	0,6123
0,6097	0,6239	0,6232	0,6278	0,6221
0,5796	0,6284	0,6220	0,6235	0,6239
0,5913	0,5982	0,6186	0,6122	0,5991
0,5941	0,6149	0,6180	0,6008	0,6128
0,6000	0,6171	0,6150	0,6068	0,6207
0,6350	0,6330	0,6236	0,6372	0,6353
0,6061	0,6144	0,6199	0,6236	0,6159
0,5798	0,5995	0,5974	0,6008	0,6049
0,6253	0,6348	0,6191	0,6323	0,6224
0,6139	0,6228	0,6274	0,6276	0,6247
0,5996	0,6323	0,6254	0,6380	0,6196
0,6129	0,6382	0,6428	0,6370	0,6409
0,6018	0,6084	0,6059	0,6177	0,6143
0,5916	0,6144	0,6077	0,6081	0,6117
0,6012	0,6299	0,6286	0,6276	0,6318
0,6012	0,6157	0,6110	0,6147	0,6160
0,5972	0,6120	0,6188	0,6202	0,6091
0,5997	0,6090	0,6117	0,6115	0,6188
0,6158	0,6338	0,6298	0,6348	0,6267
0,5955	0,6014	0,6070	0,6164	0,6156
0,5753	0,5940	0,5911	0,5853	0,5909
0,5918	0,6148	0,6135	0,6186	0,6157
0,5982	0,6134	0,6218	0,6169	0,6128
0,5866	0,5894	0,6087	0,5958	0,5987
0,6016	0,6268	0,6236	0,6264	0,6342
0,6006	0,6259	0,6253	0,6280	0,6240
0,6201	0,6152	0,6260	0,6139	0,6263
0,5964	0,6206	0,6180	0,6137	0,6159
0,6168	0,6288	0,6347	0,6288	0,6321
0,6034	0,6020	0,6053	0,6066	0,6014
0,5952	0,6057	0,6117	0,6059	0,6088

0,5929	0,6162	0,6142	0,6143	0,6216
0,5872	0,5982	0,5985	0,6003	0,6016
0,6154	0,6288	0,6280	0,6314	0,6347
0,6012	0,6299	0,6286	0,6276	0,6318
0,6012	0,6157	0,6110	0,6147	0,6160
0,5972	0,6120	0,6188	0,6202	0,6091
0,5997	0,6090	0,6117	0,6115	0,6188
0,6158	0,6338	0,6298	0,6348	0,6267
0,5955	0,6014	0,6070	0,6164	0,6156
0,5753	0,5940	0,5911	0,5853	0,5909
0,5918	0,6148	0,6135	0,6186	0,6157
0,5982	0,6134	0,6218	0,6169	0,6128
0,5866	0,5894	0,6087	0,5958	0,5987
0,6016	0,6268	0,6236	0,6264	0,6342
0,6006	0,6259	0,6253	0,6280	0,6240
0,6201	0,6152	0,6260	0,6139	0,6263
0,5964	0,6206	0,6180	0,6137	0,6159
0,6168	0,6288	0,6347	0,6288	0,6321
0,6034	0,6020	0,6053	0,6066	0,6014
0,5952	0,6057	0,6117	0,6059	0,6088
0,5929	0,6162	0,6142	0,6143	0,6216
0,5872	0,5982	0,5985	0,6003	0,6016
0,6154	0,6288	0,6280	0,6314	0,6347

---

## Apéndice B

### Algoritmo optimizador del lobo gris

```

1  #!/usr/bin/env perl6
2  use v6;
3
4
5  =for comment
6  Inicializa la matriz de posiciones de los lobos (agentes de búsqueda)
7      $cantidad_lobos      : Cantidad de lobos a participar en la
      ↪ cazería.
8      $numero_parametros  : Cantidad de parametros por problema
9      @limites_inferiores : Límites inferiores de los parámetros
10     @limites_superiores : Límites superiores de los parámetros
11
12 sub inicializar($cantidad_lobos, $numero_parametros,
      ↪ @limites_inferiores, @limites_superiores){
13
14     # Completo las filas
15     my @matrix = [];
16
17     eager loop (my $i = 0; $i < $cantidad_lobos; $i++) {
18
19         # Completo columnas
20         eager loop (my $j = 0; $j < $numero_parametros; $j++) {
21
22             #say(@limites_inferiores[$j], " "
      ↪ , @limites_superiores[$j]);
23             # Selecciona un número random en el rango dado.
24             @matrix[$i][$j] = random(@limites_inferiores[$j] ,
      ↪ @limites_superiores[$j]);
25         }

```

```

26     }
27
28     return @matrix
29 }
30
31 =for comment
32 Verifica que los nuevos valores calculados por el algoritmo no esten
33 ↪ fuera del
34 espacio de parámetros designado para el experimento
35
36 sub retornar_espacio( $cantidad_parametros, $cantidad_lobos,
37 ↪ @posiciones, @limites_inferiores, @limites_superiores){
38
39     eager loop (my $i = 0; $i < $cantidad_lobos; $i++){
40         eager loop (my $j = 0; $j < $cantidad_parametros; $j++){
41             if @posiciones[$i][$j] <= @limites_inferiores[$j] or
42 ↪ @posiciones[$i][$j] >= @limites_superiores[$j] {
43                 @posiciones[$i][$j] = random( @limites_inferiores[$j] ,
44 ↪ @limites_superiores[$j] );
45             }
46         }
47     }
48 }
49
50 =for comment
51 Se realiza una selección de parámetros para libsvm utilizando libsvm
52
53 sub libsvm_grey_wolf_optimizer($cantidad_lobos, $cantidad_iteraciones,
54 ↪ $dataset){
55
56     my $INF = Inf;
57
58     my @alpha_position      = [0, 0];
59     my $alpha_score         = -Inf;

```

```

57     my @beta_position      = [0, 0] ;
58     my $beta_score        = -Inf;
59
60     my @delta_position     = [0, 0] ;
61     my $delta_score        = -Inf;
62
63     my $cantidad_parametros = 2;
64
65
66     #   my @limites_inferiores= [2**-5, 2**-15];
67     #   my @limites_superiores = [2**15, 2**3];
68     my @limites_inferiores = [-5, -15];
69     my @limites_superiores = [15, 3];
70     my @positions          =
↳   inicializar($cantidad_lobos, $cantidad_parametros,
↳   @limites_inferiores, @limites_superiores);
71
72     eager loop (my $iteration = 0; $iteration < $cantidad_iteraciones;
↳   $iteration++){
73         say "ITERACION    $iteration";
74
75         retornar_espacio( $cantidad_parametros, $cantidad_lobos,
↳   @positions, @limites_inferiores, @limites_superiores);
76
77         my @promesas = ();
78
79         # Crea los hilos y recolecta las promesas.
80         for 0..($cantidad_lobos-1) -> $position {
81
82             my $promesa = start conveniencia($position,
↳   @(@positions[$position]), $dataset);
83             push @promesas, $promesa;
84         }
85
86         # Espera la salida de las respuestas
87         my @resultados = await @promesas;

```

```

88
89     my $fitness = 0;
90     my $position = 0;
91
92     # Revisa los resultados de los hilos para asignar los nuevos
↪   lobos alfa,
93     # beta y delta
94     for @resultados -> $resultado {
95
96         # resultado es escalar, por lo que debo sacarlo uno a uno.
97         $position = $resultado[0];
98         $fitness = $resultado[1];
99
100        if $fitness == Inf {
101
102
103        }
104
105        # Muestra la situación actual
106        say "$position: A[$alpha_score] B[$beta_score]
↪   C[$delta_score] Fitness [$fitness] para C: @positions[$position][0]
↪   y Gamma: @positions[$position][1] ";
107
108        # Escribe a un archivo los puntos que se evaluaron.
109        # spurt "gwo.$dataset.$iteration",
↪   "@positions[$position][0],@positions[$position][1],$fitness\n",
↪   :append;
110
111        if ($fitness > $alpha_score) {
112            $alpha_score = $fitness;
113            @alpha_position = @(@positions[$position]);
114
115        }elseif ($fitness > $beta_score) {
116            $beta_score = $fitness;
117            @beta_position = @(@positions[$position]);
118

```

```

119         }elseif ($fitness > $delta_score) {
120             $delta_score = $fitness;
121             @delta_position = @(@positions[$position]);
122         }
123     }
124
125     #spurt "mejores.lobos.iteration",
↪ "@alpha_position[0],@alpha_position[1]\n", :append;
126     #spurt "mejores.lobos.iteration",
↪ "@beta_position[0],@beta_position[1]\n", :append;
127     #spurt "mejores.lobos.iteration",
↪ "@delta_position[0],@delta_position[1]\n", :append;
128
129     # Recalcula el valor de a
130     my $a = Num(2 - $iteration) * Num(2 / $cantidad_iteraciones);
131
132     # Recalcula las posiciones de los lobos con respecto a los
↪ lobos alfa,
133     # beta y delta.
134     eager loop (my $i = 0; $i < $cantidad_lobos; $i++){
135
136         eager loop (my $j = 0; $j <
↪ $cantidad_parametros; $j++){
137
138             my $d_alpha      = abs( (2*1.rand)*
↪ @alpha_position[$j] - @positions[$i][$j]);
139             my $x1           = @alpha_position[$j] -
↪ ((2.0*$a*1.rand) - 1.0) * $d_alpha;
140
141             my $d_beta      =
↪ abs((2*1.rand)*@beta_position[$j] - @positions[$i][$j]);
142             my $x2           = @beta_position[$j]
↪ - ((2.0*$a*1.rand) - 1.0) * $d_beta;
143
144             my $d_delta     = abs(2 * 1.rand *
↪ @delta_position[$j] - @positions[$i][$j]);

```

```

145             my $x3 =
↳ @delta_position[$j] - ((2.0*$a*1.rand) - 1.0) * $d_delta;
146
147             @positions[$i][$j] = ($x1 + $x2 + $x3) /
↳ 3.0;
148         }
149     }
150 }
151
152 # Retorna el mayor valor (pos del lobo gris).
153     return @alpha_position;
154 }
155
156 =for comment
157 Funcion de conveniencia para el algoritmo de gwo
158
159 sub conveniencia($position, @parametros, $dataset){
160
161     my $output = Inf;
162
163     # Comando a ejecutar
164     #my $libsum_command = "svm-train -v 2 -s 0 -t 2 -c
↳ {2**@parametros[0]} -g {2**@parametros[1]}
↳ CasasolaTass.Freq20.2C.Casasola.libsum";
165     #my $libsum_command = "svm-train -v 2 -s 0 -t 2 -c
↳ {2**@parametros[0]} -g {2**@parametros[1]} $dataset";
166     my $libsvm_command = "svm-train -v 10 -s 0 -t 2 -c
↳ {2**@parametros[0]} -g {2**@parametros[1]} {$dataset}";
167
168     #say $position, " ", @parametros[0], " ", @parametros[0];
169     #say $libsum_command;
170     #say "";
171
172     #Ejecuta el comando en un shell
173     my $proc = shell($libsvm_command, :out);
174

```

```

175     # Saco el texto
176     my $libsvm_result = $proc.out.slurp-rest;
177
178     #Saco donde está el número
179     if $libsvm_result =~ m:s/Cross Validation Accuracy \=
↪ (\d+[\.\d+]?)\%/ {
180         $output = "$0";
181     }
182
183     # Retorno el valor final
184     return @($position, Num($output.Str));
185
186 }
187
188 =for comment
189 Random dentro de parámetros
190
191 sub random($limite_inferior, $limite_superior){
192
193     my $resultado = $limite_inferior + (1.rand() / (1 /
↪ ($limite_superior - $limite_inferior)));
194
195     return $resultado ;
196 }
197
198 sub main(){
199
200     if @*ARGS.elems < 3 {
201         die "no hay suficientes argumentos";
202     }
203
204     my $wolfs = Int(@*ARGS[0]);
205     my $runs = Int(@*ARGS[1]);
206     my $dataset = @*ARGS[2];
207

```

```
208     # fcantidad_lobos / fcantidad_iteraciones / Nombre del archivo de
↪ datos
209
210     my @resultado = libsvm_grey_wolf_optimizer($wolfs, $runs, $dataset
↪ );
211
212     say "$dataset.result --> C:@resultado[0],G:@resultado[1]\n";
213     spurt "$dataset-$wolfs-$runs.result", "
↪ C:@resultado[0],G:@resultado[1]\n";
214
215 }
216
217 main();
```

## Apéndice C

### Tablas de resultados del proceso de selección de parámetros

Cuadro C.1: Resultados obtenidos del proceso de selección de parámetros, los valores se encuentran en escala logarítmica

Tratamiento	Subconjunto	Parámetros	
		C	Gamma
Malla	1	1,0000	-5,0000
Malla	2	9,0000	-11,0000
Malla	3	13,0000	-15,0000
Malla	4	3,0000	-5,0000
Malla	5	11,0000	-15,0000
Malla	6	9,0000	-9,0000
Malla	7	15,0000	-11,0000
Malla	8	11,0000	-15,0000
Malla	9	11,0000	-15,0000
Malla	10	11,0000	-15,0000
Malla	11	13,0000	-11,0000
Malla	12	5,0000	-9,0000
Malla	13	15,0000	-13,0000
Malla	14	7,0000	-11,0000
Malla	15	15,0000	-13,0000
Malla	16	13,0000	-11,0000
Malla	17	9,0000	-13,0000
Malla	18	11,0000	-13,0000
Malla	19	9,0000	-9,0000
Malla	20	13,0000	-15,0000

Malla	21	3,0000	-7,0000
Malla	22	11,0000	-11,0000
Malla	23	15,0000	-13,0000
Malla	24	9,0000	-13,0000
Malla	25	15,0000	-15,0000
Malla	26	9,0000	-13,0000
Malla	27	3,0000	-7,0000
Malla	28	11,0000	-11,0000
Malla	29	11,0000	-15,0000
Malla	30	13,0000	-15,0000
Malla	31	9,0000	-11,0000
Malla	32	11,0000	-15,0000
Malla	33	11,0000	-13,0000
Malla	34	11,0000	-15,0000
Malla	35	11,0000	-15,0000
Malla	36	15,0000	-13,0000
Malla	37	13,0000	-13,0000
Malla	38	9,0000	-11,0000
Malla	39	13,0000	-13,0000
Malla	40	11,0000	-13,0000
Malla	42	13,0000	-15,0000
Malla	42	3,0000	-9,0000
Malla	44	5,0000	-7,0000
Malla	47	11,0000	-15,0000
Malla	47	13,0000	-13,0000
Malla	48	13,0000	-15,0000
Malla	50	15,0000	-13,0000
Malla	50	13,0000	-11,0000
Malla	51	13,0000	-13,0000
Malla	52	11,0000	-11,0000
Malla	53	9,0000	-11,0000
Malla	53	13,0000	-13,0000
Malla	55	3,0000	-7,0000
Malla	56	7,0000	-9,0000
Malla	58	15,0000	-13,0000
Malla	58	13,0000	-13,0000

Malla	59	9,0000	-15,0000
Malla	59	15,0000	-15,0000
Malla	59	13,0000	-15,0000
Malla	60	15,0000	-15,0000
GWO 56	1	3,0424	-6,6356
GWO 56	2	3,0185	-9,4312
GWO 56	3	6,1240	-11,1714
GWO 56	4	3,4221	-4,8743
GWO 56	5	8,9012	-10,6035
GWO 56	6	7,4010	-7,3661
GWO 56	7	5,8535	-7,9854
GWO 56	8	-0,9180	-4,4454
GWO 56	9	14,0311	-14,8890
GWO 56	10	10,7079	-14,7617
GWO 56	11	10,8885	-13,4712
GWO 56	12	7,4134	-9,5532
GWO 56	13	8,3708	-9,4521
GWO 56	14	11,1501	-13,5881
GWO 56	15	13,6144	-13,2997
GWO 56	16	14,3128	-12,4934
GWO 56	17	13,3152	-12,0995
GWO 56	18	13,7029	-14,1185
GWO 56	19	6,3651	-6,8140
GWO 56	20	12,5850	-13,8001
GWO 56	21	2,5471	-6,7396
GWO 56	22	14,7963	-12,4849
GWO 56	23	12,3148	-14,7365
GWO 56	24	14,6888	-14,8515
GWO 56	25	2,5820	-7,5848
GWO 56	26	9,8656	-9,3380
GWO 56	27	5,5780	-7,9997
GWO 56	28	11,4457	-11,5797
GWO 56	29	8,2795	-10,7628
GWO 56	30	6,9836	-11,3739
GWO 56	31	12,6548	-12,4014
GWO 56	32	11,0951	-14,1776

GWO 56	33	11,4128	-12,0821
GWO 56	34	11,0394	-13,2155
GWO 56	35	8,6405	-12,8683
GWO 56	36	10,2776	-10,8215
GWO 56	37	8,1030	-9,5066
GWO 56	38	6,1338	-8,2287
GWO 56	39	14,1786	-14,8198
GWO 56	40	11,6239	-11,5628
GWO 56	43	9,5459	-12,5520
GWO 56	44	4,3922	-6,9659
GWO 56	45	12,3932	-13,2101
GWO 56	46	13,1216	-11,9689
GWO 56	47	7,7412	-12,0128
GWO 56	48	13,5486	-14,6510
GWO 56	49	7,6593	-10,7039
GWO 56	50	11,6291	-13,0858
GWO 56	51	14,4196	-13,8024
GWO 56	52	5,9824	-9,3175
GWO 56	53	11,0742	-13,3192
GWO 56	54	10,8200	-14,2758
GWO 56	55	11,0838	-14,9172
GWO 56	56	11,8683	-13,3476
GWO 56	57	11,5332	-12,4184
GWO 56	58	5,0036	-6,7335
GWO 56	59	10,5045	-14,6044
GWO 56	60	8,7447	-10,2477
GWO 112	1	0,7637	-4,3257
GWO 112	2	14,3706	-13,7601
GWO 112	3	12,5254	-13,4555
GWO 112	4	4,0558	-5,2978
GWO 112	5	10,5531	-14,3027
GWO 112	6	12,9919	-11,0417
GWO 112	7	13,9805	-10,9303
GWO 112	8	1,5569	-6,2110
GWO 112	9	12,6046	-14,6238
GWO 112	10	7,0176	-11,6668

GWO 112	11	2,6188	-5,4749
GWO 112	12	14,4084	-13,0161
GWO 112	13	8,8541	-12,1482
GWO 112	14	10,1389	-12,8869
GWO 112	15	2,8354	-5,4728
GWO 112	16	11,1119	-10,9325
GWO 112	17	14,4294	-13,3811
GWO 112	18	6,3991	-11,5577
GWO 112	19	7,1308	-10,2566
GWO 112	20	9,1965	-11,0804
GWO 112	21	3,2332	-7,5823
GWO 112	22	9,5693	-9,6246
GWO 112	23	7,8899	-8,9750
GWO 112	24	11,0092	-14,6243
GWO 112	25	7,6392	-11,4344
GWO 112	26	7,1055	-11,6842
GWO 112	27	5,0232	-7,6532
GWO 112	28	2,4843	-7,1417
GWO 112	29	13,3694	-14,8580
GWO 112	30	11,9892	-14,3755
GWO 112	31	12,5624	-12,0816
GWO 112	32	13,8055	-14,9490
GWO 112	33	9,4190	-12,5941
GWO 112	34	1,5868	-5,6948
GWO 112	35	11,0430	-12,5194
GWO 112	36	11,9992	-11,1652
GWO 112	37	10,7157	-12,9678
GWO 112	38	5,7683	-9,0349
GWO 112	39	10,7508	-11,9428
GWO 112	40	10,6513	-12,6617
GWO 112	41	5,5636	-10,0827
GWO 112	42	13,8126	-13,6178
GWO 112	43	13,5326	-11,4303
GWO 112	44	11,4421	-13,6422
GWO 112	45	10,3692	-13,9721
GWO 112	46	12,5996	-13,3902

GWO 112	47	10,6085	-11,4478
GWO 112	48	13,8875	-14,0621
GWO 112	49	9,2407	-9,8486
GWO 112	50	12,3584	-13,3624
GWO 112	51	11,5854	-11,9060
GWO 112	52	9,5824	-14,5294
GWO 112	53	4,4443	-6,8281
GWO 112	54	10,8674	-14,5598
GWO 112	55	6,3982	-9,5892
GWO 112	56	7,1469	-8,6775
GWO 112	57	2,3181	-8,8551
GWO 112	58	8,5875	-10,5902
GWO 112	59	8,9708	-13,7119
GWO 112	60	12,8208	-13,6423
GWO 168	1	4,9942	-9,7455
GWO 168	2	12,2584	-12,4759
GWO 168	3	13,4090	-14,7291
GWO 168	4	3,8218	-4,6549
GWO 168	5	11,3493	-13,6952
GWO 168	6	12,7486	-13,1649
GWO 168	7	12,9936	-10,4451
GWO 168	8	10,8472	-13,2509
GWO 168	9	11,0970	-14,9440
GWO 168	10	7,5944	-12,0376
GWO 168	11	8,2524	-12,6621
GWO 168	12	13,0631	-12,8546
GWO 168	13	10,6988	-12,5071
GWO 168	14	10,6726	-13,1683
GWO 168	15	12,7836	-14,9112
GWO 168	16	11,6485	-11,2542
GWO 168	17	8,1729	-12,2267
GWO 168	18	10,2643	-13,4593
GWO 168	19	3,0269	-5,9676
GWO 168	20	11,2667	-13,9363
GWO 168	21	7,5439	-9,1970
GWO 168	22	3,9435	-7,2523

GWO 168	23	11,2286	-10,3691
GWO 168	24	7,2631	-10,2159
GWO 168	25	8,1327	-12,1017
GWO 168	26	12,0229	-14,8680
GWO 168	27	2,0720	-5,7864
GWO 168	28	14,3120	-12,4820
GWO 168	29	9,6848	-13,6210
GWO 168	30	12,7603	-14,6194
GWO 168	31	4,9296	-8,0767
GWO 168	32	7,0592	-9,8151
GWO 168	33	9,8938	-14,3901
GWO 168	34	1,4703	-5,5760
GWO 168	35	8,6595	-12,6356
GWO 168	36	14,7481	-11,6496
GWO 168	37	14,9921	-13,9256
GWO 168	38	14,1523	-13,0853
GWO 168	39	9,3532	-10,6836
GWO 168	40	8,6497	-9,9606
GWO 168	41	4,6354	-8,6515
GWO 168	42	11,8609	-11,5200
GWO 168	43	12,9852	-13,7211
GWO 168	44	11,6492	-11,9967
GWO 168	45	6,9811	-8,3992
GWO 168	46	13,5857	-12,5018
GWO 168	47	9,8901	-13,7024
GWO 168	48	11,2497	-13,5820
GWO 168	49	13,2191	-14,0415
GWO 168	50	12,6121	-14,3032
GWO 168	51	7,4700	-10,2485
GWO 168	52	6,2433	-9,9260
GWO 168	53	6,9513	-9,7270
GWO 168	54	11,0677	-14,6956
GWO 168	55	14,4013	-14,9641
GWO 168	56	14,2570	-14,9786
GWO 168	57	14,0514	-14,0976
GWO 168	58	9,4115	-13,9169

GWO 168	59	8,6633	-9,7901
GWO 168	60	2,5430	-8,7700

# Bibliografía

- [Ab Wahab et al., 2015] Ab Wahab, M. N., Nefti-Meziani, S., and Atyabi, A. (2015). A comprehensive review of swarm optimization algorithms. *Plos One*, 10(5):e0122827.
- [Abu-Mostafa et al., 2012] Abu-Mostafa, Y. S., Magdon-Ismael, M., and Lin, H.-T. (2012). *Learning from data*. AMLBook.
- [Aisopos et al., 2012] Aisopos, F., Papadakis, G., Tserpes, K., and Varvarigou, T. (2012). Content vs. context for sentiment analysis: A comparative analysis over microblogs. In *Proceedings of the 23rd ACM Conference on Hypertext and Social Media*, HT '12, pages 187–196, New York, NY, USA. ACM. Published in: Proceeding HT '12 Proceedings of the 23rd ACM conference on Hypertext and social media Pages 187-196 ACM New York, NY, USA ©2012.
- [Anta et al., 2013] Anta, A. F., Chiroque, L. N., Morere, P., and Santos, A. (2013). Sentiment analysis and topic detection of spanish tweets: A comparative study of of nlp techniques. *Procesamiento del lenguaje natural*, 50:45–52.
- [Bahrainian and Dengel, 2013] Bahrainian, S.-A. and Dengel, A. (2013). Sentiment analysis and summarization of twitter data. In *Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on*, pages 227–234.
- [Balahur and Turchi, 2014] Balahur, A. and Turchi, M. (2014). Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis. *Computer Speech & Language*, 28(1):56–75.
- [Ben-Hur and Weston, 2010] Ben-Hur, A. and Weston, J. (2010). A user's guide to support vector machines. *Methods in molecular biology (Clifton, N.J.)*, 609:223–239.
- [Boser et al., 1992] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.
- [Burges, 1998] Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.
- [Cambria et al., 2010] Cambria, E., Speer, R., Havasi, C., and Hussain, A. (2010). Senticnet: A publicly available semantic resource for opinion mining. In *AAAI fall symposium: commonsense knowledge*, volume 10, page 02.
- [Carrizosa et al., 2014] Carrizosa, E., Martín-Barragán, B., and Morales, D. R. (2014). A nested heuristic for parameter tuning in support vector machines. *Computers & Operations Research*, 43(0):328 – 334.
- [Carstens and Toni, 2013] Carstens, L. and Toni, F. (2013). Enhancing sentiment extraction from text by means of arguments. In *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*, WISDOM '13, pages 4:1–4:9, New York, NY, USA. ACM. Published in: Proceeding WISDOM '13 Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining Article No. 4 ACM New York, NY, USA ©2013.
- [Casasola et al., 2016] Casasola, E., Leoni, A., and Marín, A. (2016). Evaluación de características potenciales presentes en textos cortos en español para clasificarlos por polaridad. In *I Coloquio Costarricense de Procesamiento de Lenguaje Natural*.
- [Chang and Lin, 2011] Chang, C.-C. and Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

- [Chapelle et al., 2002] Chapelle, O., Vapnik, V., Bousquet, O., and Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine learning*, 46(1-3):131-159.
- [Chen and Zimbra, 2010] Chen, H. and Zimbra, D. (2010). Ai and opinion mining. *Intelligent Systems, IEEE*, 25(3):74-80.
- [Chen et al., 2014] Chen, K.-C., Li, C.-H., Kuo, B.-C., and Wang, M.-S. (2014). Applying automatic kernel parameter selection method to the full bandwidth rbf kernel function for hyperspectral image classification. In *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*, pages 3442-3445. Ieee.
- [Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273-297.
- [Daoud and Turabieh, 2013] Daoud, E. A. and Turabieh, H. (2013). New empirical nonparametric kernels for support vector machine classification. *Applied Soft Computing*, 13(4):1759 - 1765.
- [Davenport et al., 2010] Davenport, M., Baraniuk, R. G., Scott, C. D., et al. (2010). Tuning support vector machines for minimax and neyman-pearson classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(10):1888-1898.
- [Devos et al., 2009] Devos, O., Ruckebusch, C., Durand, A., Duponchel, L., and Huvenne, J.-P. (2009). Support vector machines (svm) in near infrared (nir) spectroscopy: Focus on parameters optimization and model interpretation. *Chemometrics and Intelligent Laboratory Systems*, 96(1):27 - 33.
- [Duan et al., 2003] Duan, K., Keerthi, S. S., and Poo, A. N. (2003). Evaluation of simple performance measures for tuning svm hyperparameters. *Neurocomputing*, 51:41-59.
- [Fernández Anta et al., 2012] Fernández Anta, A., Morere, P., Chiroque, L. F., and Santos, A. (2012). Techniques for sentiment analysis and topic detection of spanish tweets: preliminary report. page 13.
- [Firmino et al., 2013] Firmino, A. A., Oliveira, M. G. D., and Paiva, A. C. D. (2013). A Comparison of SVM Versus Naive-Bayes Techniques for Sentiment Analysis in Tweets: A Case Study with the 2013 FIFA Confederations Cup. pages 123-130.
- [Hall et al., 2009] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10-18.
- [Hofmann et al., 2008] Hofmann, T., Schölkopf, B., and Smola, A. J. (2008). Kernel methods in machine learning. *The annals of statistics*, pages 1171-1220.
- [Hsu and Lin, 2002] Hsu, C. and Lin, C. (2002). A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415-425.
- [Isguder-Sahin et al., 2014] Isguder-Sahin, G. G., Zafer, H. R., and Adah, E. (2014). Polarity detection of turkish comments on technology companies. In *Asian Language Processing (IALP), 2014 International Conference on*, pages 136-139. IEEE.
- [Kaya et al., 2012] Kaya, M., Fidan, G., and Toroslu, I. (2012). Sentiment analysis of turkish political news. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on*, volume 1, pages 174-180. Ieee.
- [Kim et al., 2014] Kim, K., Chung, B.-s., Choi, Y., Lee, S., Jung, J.-Y., and Park, J. (2014). Language independent semantic kernels for short-text classification. *Expert Systems with Applications*, 41(2):735-743.
- [Kim and Hovy, 2004] Kim, S.-m. and Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1367. Association for Computational Linguistics.
- [Komaki and Kayvanfar, 2015] Komaki, G. and Kayvanfar, V. (2015). Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time. *Journal of Computational Science*, 8(0):109 - 120.

- [Li et al., 2013] Li, C.-H., Kuo, B.-C., Lin, L.-H., Wu, W., and Lan, D. (2013). Apply an automatic parameter selection method to generalized discriminant analysis with rbf kernel for hyperspectral image classification. In *Machine Learning and Cybernetics (ICMLC), 2013 International Conference on*, volume 1, pages 253–258. IEEE.
- [Li et al., 2010a] Li, C.-H., Lin, C.-T., Kuo, B.-C., and Chu, H.-S. (2010a). An automatic method for selecting the parameter of the rbf kernel function to support vector machines. In *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*, pages 836–839. Ieee.
- [Li et al., 2010b] Li, C.-H., Lin, C.-T., Kuo, B.-C., and Ho, H.-H. (2010b). An automatic method for selecting the parameter of the normalized kernel function to support vector machines. In *Technologies and Applications of Artificial Intelligence (TAAI), 2010 International Conference on*, pages 226–232.
- [Lin and Liu, 2007] Lin, S.-l. and Liu, Z. (2007). Parameter selection in svm with rbf kernel function. *Journal-Zhejiang University of Technology*, 35(2):163.
- [Lin et al., 2008] Lin, S.-W., Lee, Z.-J., Chen, S.-C., and Tseng, T.-Y. (2008). Parameter determination of support vector machine and feature selection using simulated annealing approach. *Applied Soft Computing*, 8(4):1505 – 1512. *Soft Computing for Dynamic Data Mining*.
- [Liu, 2010] Liu, B. (2010). Sentiment Analysis and Subjectivity. *Handbook of Natural Language Processing*, 5(1):1–38.
- [Liu et al., 2005] Liu, H.-J., Wang, Y.-N., and Lu, X.-F. (2005). A method to choose kernel function and its parameters for support vector machines. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, volume 7, pages 4277–4280. IEEE.
- [Lu et al., 2009] Lu, M.-Z., Chen, C. P., and Huo, J.-B. (2009). Optimization of combined kernel function for svm by particle swarm optimization. In *Machine Learning and Cybernetics, 2009 International Conference on*, volume 2, pages 1160–1166. IEEE.
- [Luts et al., 2010] Luts, J., Ojeda, F., de Plas, R. V., Moor, B. D., Huffel, S. V., and Suykens, J. A. (2010). A tutorial on support vector machine-based methods for classification problems in chemometrics. *Analytica Chimica Acta*, 665(2):129 – 145.
- [Madhoushi and Hamdan, 2015] Madhoushi, Z. and Hamdan, A. R. (2015). Sentiment Analysis Techniques in Recent Works. pages 2013–2016.
- [Mahmoud and Calvert, 2009] Mahmoud, E.-S. and Calvert, D. (2009). Auto-calibration of support vector machines for detecting disease outbreaks. In *Science and Technology for Humanity (TIC-STH), 2009 IEEE Toronto International Conference*, pages 112–117. Ieee.
- [Mech, 1999] Mech, L. D. (1999). Alpha status, dominance, and division of labor in wolf packs. *Canadian Journal of Zoology*, 77(8):1196–1203.
- [Mirjalili, 2015a] Mirjalili, S. (2015a). How effective is the grey wolf optimizer in training multi-layer perceptrons. *Applied Intelligence*, 43(1):150–161.
- [Mirjalili, 2015b] Mirjalili, S. (2015b). The Ant Lion Optimizer. *Advances in Engineering Software*, 83(0):80–98.
- [Mirjalili et al., 2014] Mirjalili, S., Mirjalili, S. M., and Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, 69:46–61.
- [Mohammad et al., 2013] Mohammad, S. M., Kiritchenko, S., and Martin, J. (2013). Identifying purpose behind electoral tweets. In *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining, WISDOM '13*, pages 1:1–1:9, New York, NY, USA. ACM. Published in: Proceeding WISDOM '13 Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining Article No. 1 ACM New York, NY, USA ©2013.

- [Montejo-Ráez et al., 2013] Montejo-Ráez, A., Díaz-Galiano, M. C., Perea-Ortega, J. M., and Ureña-López, L. A. (2013). Spanish knowledge base generation for polarity classification from masses. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 571–578. International World Wide Web Conferences Steering Committee.
- [Moreno-Ortiz and Hernández, 2012] Moreno-Ortiz, A. and Hernández, C. (2012). Lexicon-based sentiment analysis of Twitter messages in Spanish. *Procesamiento del Lenguaje ...*, pages 93–100.
- [Mouthami et al., 2013] Mouthami, K., Devi, K., and Bhaskaran, V. (2013). Sentiment analysis and classification based on textual reviews. In *Information Communication and Embedded Systems (ICICES), 2013 International Conference on*, pages 271–276.
- [Mu and Nandi, 2007] Mu, T. and Nandi, A. K. (2007). Breast cancer detection from {FNA} using {SVM} with different parameter tuning systems and som-rbf classifier. *Journal of the Franklin Institute*, 344(3-4):285 – 311. Special Issue: Medical Applications of Signal Processing, Part I.
- [Muro et al., 2011] Muro, C., Escobedo, R., Spector, L., and Coppinger, R. (2011). Wolf-pack (canis lupus) hunting strategies emerge from simple rules in computational simulations. *Behavioural processes*, 88(3):192–197.
- [Nasukawa et al., 2003] Nasukawa, T., Nasukawa, T., Yi, J., and Yi, J. (2003). Sentiment analysis: Capturing favorability using natural language processing. *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77.
- [Pang and Lee, 2008] Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.
- [Patle and Chouhan, 2013] Patle, A. and Chouhan, D. (2013). Svm kernel functions for classification. In *Advances in Technology and Engineering (ICATE), 2013 International Conference on*, pages 1–9. Ieee.
- [Qiu et al., 2010] Qiu, S., Li, Z., Zhang, L., Suu, Y., and Wang, D. (2010). Modified particle swarm optimization for multi-scale kernel function in svm. In *Information Engineering and Computer Science (ICIECS), 2010 2nd International Conference on*, pages 1–5. IEEE.
- [Saad and Mathiak, 2013] Saad, F. and Mathiak, B. (2013). Revised mutual information approach for german text sentiment classification. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 579–586. International World Wide Web Conferences Steering Committee.
- [Saremi et al., 2015] Saremi, S., Mirjalili, S., and Mirjalili, S. (2015). Evolutionary population dynamics and grey wolf optimizer. *Neural Computing and Applications*, 26(5):1257–1263.
- [Schölkopf, 2002] Schölkopf, B. (2002). Learning with kernels. *Journal of the Electrochemical Society*, 129(November):2865.
- [Singh et al., 2013] Singh, V., Piryani, R., Uddin, A., Waila, P., et al. (2013). Sentiment analysis of textual reviews; evaluating machine learning, unsupervised and sentiwordnet approaches. In *Knowledge and Smart Technology (KST), 2013 5th International Conference on*, pages 122–127. IEEE.
- [Solís Parajeles, 2015] Solís Parajeles, M. (2015). Adaptación del algoritmo pmi-ir para la creación de diccionarios de términos polarizados. Master’s thesis, Escuela de computación e Informatica, Universidad de Costa Rica.
- [Song et al., 2015] Song, X., Tang, L., Zhao, S., Zhang, X., Li, L., Huang, J., and Cai, W. (2015). Grey wolf optimizer for parameter estimation in surface waves. *Soil Dynamics and Earthquake Engineering*, 75(0):147 – 157.
- [Sun et al., 2010] Sun, J., Zheng, C., Li, X., and Zhou, Y. (2010). Analysis of the distance between two classes for tuning svm hyperparameters. *Neural Networks, IEEE Transactions on*, 21(2):305–318.

- [Sánchez A, 2003] Sánchez A, V. D. (2003). Advanced support vector machines and kernel methods. *Neurocomputing*, 55(1-2):5 – 20. Support Vector Machines.
- [Thelwall et al., 2010] Thelwall, M., Buckley, K., Paltoglou, G., and Cai, D. (2010). Sentiment Strength Detection in Short Informal Text. *Journal of the American Society for Information Science*, 61(12):2544-2558.
- [Trindade et al., 2014] Trindade, L., Wang, H., Blackburn, W., and Taylor, P. (2014). Enhanced factored sequence kernel for sentiment classification. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*, volume 2, pages 519-525.
- [Tromp and Pechenizkiy, 2013] Tromp, E. and Pechenizkiy, M. (2013). Rbem: A rule based approach to polarity detection. In *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining, WISDOM '13*, pages 8:1-8:9, New York, NY, USA. ACM. Published in: Proceeding WISDOM '13 Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining Article No. 8 ACM New York, NY, USA ©2013.
- [Vapnik, 2013] Vapnik, V. (2013). *The nature of statistical learning theory*. Springer Science & Business Media.
- [Villena Román et al., 2015] Villena Román, J., García Morena, J., Martínez Cámara, E., and Jiménez Zafra, S. M. (2015). TASS 2014 - The Challenge of Aspect-based Sentiment Analysis. *Procesamiento de Lenguaje Natural*, 54:61-68.
- [Villena Román et al., 2014] Villena Román, J., García Morera, J., Lana Serrano, S., and González Cristóbal, J. C. (2014). TASS 2013 - A second step in reputation analysis in Spanish. *Procesamiento de Lenguaje Natural*, pages 37-44.
- [Villena Román et al., 2013] Villena Román, J., Martínez-cámara, E., Lana-serrano, S., and González-cristóbal, J. C. (2013). TASS - Workshop on Sentiment Analysis at SEPLN TASS - Taller de Análisis de Sentimientos en la SEPLN. *Revistas - Procesamiento del Lenguaje Natural*, 50:37-44.
- [Wang et al., 2014] Wang, X., Huang, F., and Cheng, Y. (2014). Super-parameter selection for gaussian-kernel {SVM} based on outlier-resisting. *Measurement*, 58(0):147 – 153.
- [Watanabe and Kimura, 2008] Watanabe, S. and Kimura, Y. (2008). A methodology using emo for parameter estimation of svm kernel function. In *Soft Computing in Industrial Applications, 2008. SMCia'08. IEEE Conference on*, pages 211-216. IEEE.
- [Wei-Chih and Yu, 2009] Wei-Chih, H. and Yu, T.-Y. (2009). E-mail spam filtering using support vector machines with selection of kernel function parameters. In *Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on*, pages 764-767. IEEE.
- [Yao and Li, 2009] Yao, T. and Li, L. (2009). A kernel-based sentiment classification approach for chinese sentences. In *Computer Science and Information Engineering, 2009 WRI World Congress on*, volume 5, pages 513-518. Ieee.
- [Zhao et al., 2014] Zhao, Z., Liu, J., and Cox, J. (2014). Safe and efficient screening for sparse support vector machine. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, pages 542-551, New York, NY, USA. ACM. SESSION: Keynote talks The battle for the future of data mining Oren Etzioni Pages: 1-1.
- [Zhu and Tang, 2010] Zhu, Y. F. and Tang, X. M. (2010). Overview of swarm intelligence. In *ICCASM 2010 - 2010 International Conference on Computer Application and System Modeling, Proceedings*, volume 9, pages 400-403.
- [Zhu et al., 2013] Zhu, Z., Blanke, U., Calatroni, A., and Tröster, G. (2013). Human activity recognition using social media data. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia, MUM '13*, pages 21:1-21:10, New York, NY, USA. ACM. Published in: Proceeding MUM '13 Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia Article No. 21 ACM New York, NY, USA ©2013.

[Zong et al., 2006] Zong, Q., Liu, W., and Dou, L. (2006). Parameters selection for svr based on pso. In *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, volume 1, pages 2811–2814. Ieee.