



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Dirección General de Estudios de Posgrado
Facultad de Ingeniería de Sistemas e Informática
Unidad de Posgrado

**Un algoritmo multiplicador proximal para
clasificación binaria en máquinas de vectores soporte**

TESIS

Para optar el Grado Académico de Doctor en Ingeniería de
Sistemas e Informática

AUTOR

Miguel Angel CANO LENGUA

ASESOR

Dr. Erik Alex PAPA QUIROZ

Lima, Perú

2023



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Cano, M. (2023). *Un algoritmo multiplicador proximal para clasificación binaria en máquinas de vectores soporte*. [Tesis de doctorado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Unidad de Posgrado]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Miguel Angel Cano Lengua
Tipo de documento de identidad	DNI
Número de documento de identidad	10324686
URL de ORCID	https://orcid.org/0000-0002-1809-5082
Datos de asesor	
Nombres y apellidos	Erik Alex Papa Quiroz
Tipo de documento de identidad	DNI
Número de documento de identidad	10451642
URL de ORCID	https://orcid.org/0000-0002-8678-6918
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	David Santos Mauricio Sánchez
Tipo de documento	DNI
Número de documento de identidad	06445495
Miembro del jurado 1	
Nombres y apellidos	José Alfredo Herrera Quispe
Tipo de documento	DNI
Número de documento de identidad	40362859
Miembro del jurado 2	
Nombres y apellidos	Cayo Víctor León Fernández
Tipo de documento	DNI
Número de documento de identidad	07001405
Datos de investigación	
Línea de investigación	No aplica

Grupo de investigación	Optimización Matemática y Computacional (OPTIMACO)
Agencia de financiamiento	Sin Financiamiento
Ubicación geográfica de la investigación	Edificio: Universidad Nacional Mayor de San Marcos País: Perú Departamento: Lima Provincia: Lima Latitud: -12.05819215 Longitud: -777.0189181894387
Año o rango de años en que se realizó la investigación	2018 - 2022
URL de disciplinas OCDE	Ciencias de la Computación https://purl.org/pe-repo/ocde/ford#1.02.01 Matemática Aplicada https://purl.org/pe-repo/ocde/ford#1.01.02



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
Universidad del Perú. Decana de América
Facultad de Ingeniería de Sistemas e Informática
Vicedecanato de Investigación y Posgrado
Unidad de Posgrado

ACTA DE SUSTENTACIÓN VIRTUAL DE TESIS PARA OPTAR EL GRADO ACADÉMICO DE DOCTOR EN INGENIERÍA DE SISTEMAS E INFORMÁTICA

A los veintidós (22) días del mes de febrero de 2023, siendo las 5:00 pm., se reunieron en la sala virtual <https://meet.google.com/jtz-vuji-ion> el Jurado de Tesis conformado por los siguientes docentes:

Dr. David Santos Mauricio Sánchez (Presidente)
Dr. José Alfredo Herrera Quispe (Miembro)
Dr. Cayo Víctor León Fernández (Miembro)
Dr. Erik Alex Papa Quiroz (Miembro asesor)

Se inició la Sustentación invitando al candidato a Doctor **MIGUEL ANGEL CANO LENGUA**, para que realice la exposición oral y virtual de la tesis para optar el Grado Académico de Doctor en Ingeniería de Sistemas e Informática, siendo la Tesis intitulada:

“UN ALGORITMO MULTIPLICADOR PROXIMAL PARA CLASIFICACIÓN BINARIA EN MÁQUINAS DE VECTORES SOPORTE”

Concluida la exposición, los miembros del Jurado de Tesis procedieron a formular sus preguntas que fueron absueltas por el graduando; acto seguido se procedió a la evaluación correspondiente, habiendo obtenido la siguiente calificación:

..... Diecisiete (17) - Muy Bueno.....

Por tanto, el presidente del Jurado, de acuerdo con el Reglamento General de Estudios de Posgrado, otorga al Magíster **MIGUEL ANGEL CANO LENGUA** el Grado Académico de Doctor en Ingeniería de Sistemas e Informática.

Siendo las. 18:30 horas, el presidente del Jurado de Tesis, da por concluido el acto académico de Sustentación de Tesis.


Dr. David Santos Mauricio Sánchez
(Presidente)


Dr. José Alfredo Herrera Quispe
(Miembro)


Dr. Cayo Víctor León Fernández
(Miembro)


Dr. Erik Alex Papa Quiroz
(Miembro Asesor)



Lima, 18 de Noviembre del 2022

INFORME N° 000012-2022-UPG-VDIP-FISI/UNMSM

INFORME DE EVALUACIÓN DE ORIGINALIDAD

1. Autoridad Académica que emite el Informe de Originalidad:	Director de la Unidad de Posgrado
2. Apellidos y Nombres de la autoridad académica:	Dr. Cayo Víctor León Fernández
3. Operador del programa informático de similitudes:	Dr. Cayo Víctor León Fernández
4. Documento evaluado:	Tesis para Posgrado Título: "Un Algoritmo Multiplicador Proximal para Clasificación Binaria en Máquinas de Vectores Soporte"
5. Autor del documento:	Miguel Ángel Cano Lengua
6. Fecha de recepción de documento	15/11/2022
7. Fecha de aplicación del programa detector de similitudes:	16/11/2022
8. Software utilizado:	Turnitin
9. Configuración del programa detector de similitudes:	i. Excluye textos entrecomillados: SI ii. Excluye biografías: SI iii. Excluye cadenas menores a 40 palabras: SI iv. Otro criterio (especificar): NO
10. Porcentaje de similitudes según programa detector de similitudes	Ocho por ciento (8%)
11. Fuentes originales de las similitudes encontradas	Se adjuntan en cinco (05) fojas al presente informe
12. Observaciones:	Ninguna
13. Calificación de originalidad i. Documento cumple criterios de originalidad, sin observaciones. ii. Documento cumple criterio de originalidad, con observaciones. iii. Documento no cumple criterios de originalidad.	Documento cumple criterio de originalidad, sin observaciones.
14. Fecha del Informe:	17/11/2022

DR. CAYO VICTOR LEON FERNANDEZ
DIRECTOR DE LA UNIDAD DE POSGRADO(e)



AGRADECIMIENTO

Mi agradecimiento especial a mi asesor y amigo el Dr. Erik Alex Papa Quiroz, por su asesoramiento y paciencia en el desarrollo de la tesis. Al Dr. Julio Cesar López Luis por su participación en la elaboración del algoritmo, a los docentes de la Facultad de ingeniería de Sistemas e Informática de la Universidad Nacional Mayor de San Marcos por sus conocimientos brindados.

A mi madre María Luisa, a mis hermanos, Johnny, Fernando, Jorge y mi hermana María, a mis cuñadas, cuñado y a todos mis sobrinos, por su apoyo incondicional.

DEDICATORIA

- A mi esposa Marleny
- A mis engreídos Ale y Aaron

ÍNDICE GENERAL

AGRADECIMIENTO	IV
DEDICATORIA	V
ÍNDICE DE TABLAS	X
ÍNDICE DE FIGURAS	XI
RESUMEN	XIII
ABSTRACT	XV
CAPÍTULO I: INTRODUCCIÓN	17
1.1 Situación Problemática	17
1.2 Formulación del Problema	20
1.3 Justificación de la Investigación	22
1.3.1 Justificación Teórica	22
1.3.2 Justificación Práctica	22
1.4 Objetivos	22
1.4.1 Objetivo General	22
1.4.2 Objetivos Específicos	23
1.5 Hipótesis	23
1.5.1. Hipótesis Generales	23
1.5.2. Hipótesis Específicas	23
1.6 Propuesta	24
1.7 Organización de la Tesis	25
1.8 Variables	26
1.8.1. Identificación de Variables	26
CAPÍTULO II: MARCO TEÓRICO	27
2.1 Antecedente del Problema	27
2.1.1 Métodos Punto Proximales	29
2.1.2 Algoritmo de Clasificación Máquinas de Vectores de Soporte	31
2.2 Bases Teóricas	34
2.2.1 Algoritmo	34
2.2.2 Método del Punto Proximal	35
2.2.3 Optimización Simétrica del cono convexo (SCP) con estructura separable	35
2.2.4 Modelos de Máquina de vectores de soporte utilizando cono simétrico de segundo orden con estructura separable	37

2.2.5 Modelo de máquinas vectoriales de soporte gemelas (TWSVM)	38
2.2.6 Optimización matemática	38
2.3 Marco Conceptual	41
CAPÍTULO III: ESTADO DEL ARTE	42
3.1. Introducción	42
3.2. Metodología de Investigación	43
3.2.1 Planificación de la Revisión	44
3.2.2 Desarrollo de la Revisión	45
3.2.3 Resultados de la Revisión	46
3.3. Análisis del Estudio	49
3.3.1 Áreas de Investigación	50
3.3.2 Aplicaciones relacionadas con la clasificación	51
3.3.3 Métodos de optimización o algoritmos utilizados en SVM	52
3.3.4 Métodos para construir hiperplanos proximales en SVM	55
3.4. Análisis de Resultados	59
A. Áreas de investigación (Q1)	59
B. Aplicaciones relacionadas con la clasificación (Q2)	60
C. Algoritmos de optimización utilizados en SVM (Q3)	60
D. Métodos relacionados con hiperplanos proximales en SVM para clasificación (Q4)	61
3.5. Conclusiones	62
CAPÍTULO IV: UN MÉTODO DE MULTIPLICADOR PROXIMAL PARA SEPARABLES CONVEXOS OPTIMIZACIÓN DE CONO SIMÉTRICO	63
4.1. Introducción	63
4.2. Preliminares	65
4.3. Distancias proximales sobre conos simétricos	67
4.4. Algoritmo inexacto con distancia proximal generalizada	71
4.4.1. Algoritmo del multiplicador proximal	72
4.4.2. Análisis de convergencia del algoritmo SC-PMA	75
4.4.3. Complejidad del Algoritmo	82
CAPÍTULO V: METODOLOGÍA DE LA INVESTIGACIÓN	85
5.1. Tipo de Investigación	85
5.2. Técnicas descriptivas para la contratación o demostración de la hipótesis	85
5.3. Metodología del desarrollo de Tesis	85
5.4. Desarrollo del diseño del algoritmo SC - PMA	87
5.5. Resultados	88
CAPÍTULO VI: IMPLEMENTACIÓN DEL ALGORITMO	89
6.1. Entrada y salida del SC - PMA	91

6.2. Generación Lineal.....	93
6.3. Subprograma Principal.....	94
6.4. Ejemplo 1	95
6.5. Ejemplo 2	97
6.6. Conclusiones.....	100
CAPÍTULO VII. DISEÑO.....	102
7.1. Sistema	102
7.2. Herramienta de Desarrollo	103
7.2.1 NetBeans	103
7.2.2 Lenguaje de Programación Java	103
7.2.3 Figma	103
7.2.4 SQL SERVER	103
7.2.5 MVC	103
7.2.6 Modelos.....	104
7.2.7 Vista.....	104
7.2.8 Controladores	104
7.2.9 Framework.....	104
7.3. Principios de Scrum.....	105
7.4. Roles	106
7.4.1. Product Owner	106
7.4.2. Equipo de Desarrollo	106
7.4.3. Scrum Máster.....	107
7.5. Artefactos de Scrum.....	107
7.5.1. Lista de Producto (Product Backlog)	107
7.5.2. Lista de Pendientes del Sprint (Sprint Backlog)	108
7.5.3. Incremento.....	108
7.6. Ciclo de vida de la Metodología SRUM	109
7.6.1. Inicio del Proyecto	109
7.6.2. Planificación	109
7.6.3. Implementación	110
7.6.4. Review y Retrospectiva.....	110
7.6.5. Lanzamiento	110
7.7. Caso de Estudio	111
7.7.1. Inicio del Proyecto	111
7.7.2. Usuarios del proyecto.....	112
7.7.3. Planning Poker.....	113
7.7.4. Diagrama de Gantt	114

7.8. Resultados	114
7.8.1. Desarrollo de los prototipos de las Historias de Usuario	114
7.8.1.1. Presentación de los Sprint	114
7.8.1.1.1. Primer Sprint	114
7.8.1.1.2. Segundo Sprint	115
7.8.2. Desarrollo del modelo de la base de datos	117
7.9. Arquitectura del Software	118
CAPÍTULO VIII. RESULTADOS, CONCLUSIONES Y TRABAJOS FUTUROS	120
8.1. Resultados	120
8.1.1. Experimento Numérico	120
8.1.2. Análisis Comparativo entre Algoritmo Proximales.	121
8.2. Conclusiones.....	124
8.3. Discusiones.....	125
8.4. Trabajos Futuros	126
REFERENCIAS	127
ANEXOS	139
ANEXOS A. Implementaciones	139
ANEXO A.1. Función Principal.....	139
ANEXO A.2. Generador Lineal	141
ANEXO A.3. CVX_SVM (DATA).....	143
ANEXO A.4. SVM_LIN_SC_PMA.....	144
ANEXO B. Algoritmo Proximales	149
ANEXO B.1. (PMAPD).....	149
Algoritmo Multiplicador Proximal con Distancias Proximales (PMAPD)	149
ANEXO B.2. (RIPADM).....	150
Algoritmo del Método de Direcciones Alternantes con Regularización Proximal Interior inexacto	150
ANEXO C. Aplicación del Problema (P).....	151
ANEXO D. Producción Científica.....	152
ANEXOS D.1. Artículos Publicados	152
ANEXOS D.2. Artículos Aceptados.....	158

ÍNDICE DE TABLAS

Tabla 1	Criterios de Inclusión	45
Tabla 2	Criterios de Exclusión	45
Tabla 3	Seleccionar artículos	46
Tabla 4	Distribución por Publicación y Fuente.....	48
Tabla 5	Preguntas relacionadas con Áreas de Investigación	50
Tabla 6	Preguntas relacionadas con las aplicaciones a la clasificación	51
Tabla 7	Preguntas de Investigación relacionadas con los Métodos o Algoritmos de Optimización utilizados en las SVM.....	52
Tabla 8	Métodos relacionados con hiperplanos proximales en SVM para clasificación	55
Tabla 9	Actividades desarrolladas para la implementación del algoritmo	89
Tabla 10	Resultados computacionales para el Ejemplo 1.....	97
Tabla 11	Resultados computacionales para el Ejemplo 2.....	99
Tabla 12	Historia de Usuario 1	112
Tabla 13	Historia de Usuario 2	112
Tabla 14	Historia de Usuario 3	113
Tabla 15	Resultados Computacionales	121

ÍNDICE DE FIGURAS

Figura 1 Esquema de la propuesta de Investigación	24
Figura 2 Áreas de Investigación.....	32
Figura 3 Proceso de Revisión Sistemática de la literatura.....	46
Figura 4 Tendencias de publicación de máquinas de vectores de soporte aplicadas a la clasificación en SCOPUS.	47
Figura 5 Estudios de evolución en máquinas de vectores de soporte.....	48
Figura 6 Marco general propuesto para la revisión literaria.	50
Figura 7 Áreas de Investigación.....	59
Figura 8 Los métodos con hiperplanos proximales estudiados por máquinas de vectores de soporte (SVM).	61
Figura 9 Diagrama de Flujo del Algoritmo SC - PMA	74
Figura 10 Diagrama de Procesamiento SC - PMA.....	91
Figura 11 Datos separables lineales.	96
Figura 12 Convergencia de la función objetivo y las normas $\mathbf{z}^k - \mathbf{z}^{k-1}$ y $\mathbf{v}^k - \mathbf{v}^{k-1}$	96
Figura 13 Conjunto de datos separados por un plano.....	97
Figura 14 Datos separables lineales.	98
Figura 15 Convergencia de la función objetivo y las normas $\mathbf{z}^k - \mathbf{z}^{k-1}$ y $\mathbf{v}^k - \mathbf{v}^{k-1}$	99
Figura 16 Conjunto de datos separados por un plano.....	99
Figura 17 Ciclo de vida del Marco Metodológico de SRUM.....	105
Figura 18 Diagrama de equipo Scrum.....	107

Figura 19 Artefactos de Scrum	109
Figura 20 Logueo Clasificación	115
Figura 21 Generador Lineal	116
Figura 22 Gráfico de los Resultados procesados	116
Figura 23 Resultados de la ecuación del plano	117
Figura 24 Modelo de Base de Datos del Sistema.....	118
Figura 25 Arquitectura del Software	119
Figura 26 Diagrama del Hardware	119
Figura 27 Esquema Final del aporte de Investigación.....	123

RESUMEN

Las técnicas de clasificación de datos tienen una gran variedad de aplicaciones en los distintos campos debido al avance de la ciencia afortunadamente existen diversos clasificadores como redes neuronales, arboles de decisión, máquinas de vectores de soporte entre otros.

Estas técnicas se pueden aplicar a las áreas como la medicina, para clasificar y prevenir el cáncer de piel generado por el melanoma, se establece una muestra y con la técnica de clasificación se determina que zona ha sido afectada, otra aplicación importante es relacionado con el medio ambiente, y es el referido al reciclaje para clasificación de desechos, también se tiene importantes aportes para el reconocimiento facial entre otras aplicaciones.

Las técnicas de máquina de vectores de soporte para clasificar datos resuelven un modelo de optimización cuya función objetivo es cuadrática por ende diferenciable. Sin embargo si la función fuese no diferenciable no se podría usar esta técnica, es por ello que se requiere una técnica que trabaje con funciones diferenciables y no diferenciables.

De acuerdo con lo relevante que son las técnicas de clasificación, se desarrolló el Algoritmo Multiplicador Proximal de Cono Simétrico para Optimización Separable aplicado a máquina de vectores de soporte o SVM por sus siglas en inglés (Support Vector Machine) para Clasificación Binaria, este algoritmo multiplicador trabaja con subdiferenciables, es decir, pueden encontrar la función óptima, si la función es diferenciable o no.

Este algoritmo tiene como fundamento teórico los espacios vectoriales de dimensión finita, además de un producto interior, un Álgebra de Jordan Euclidiana, distancias proximales, cono simétrico de segundo orden, entre otros conceptos teóricos.

El algoritmo para clasificación de datos binario se desarrolló probando la buena

definición, el análisis de la convergencia, el análisis de la complejidad de este.

Este algoritmo surge como una nueva técnica el cual puede ser utilizado para la clasificación de datos, se probó su convergencia global bajo determinados supuestos, se realizó la implementación del algoritmo en el software Matlab versión 2017 (R2017a), en una computadora 8 th Gen Intel (R) Core (TM) i5-8250U CPU, 1.60 GHz 1.80 GHz, 4.00 GB, Windows 1064 bits.

Posteriormente se realizó el diseño utilizando el marco metodológico Scrum. Para ello se desarrolló 3 sprint de acuerdo con los requerimientos de las historias de usuarios.

Se realizó un análisis comparativo con otros algoritmos proximales, así como también experimentos computacionales con el uso del algoritmo implementado.

Finalmente mencionaremos que el presente trabajo es inédito y surge de la necesidad de tener técnicas apropiadas para clasificar datos.

Keywords: Multiplicador Proximal, Cono Simétrico, Optimización Separable, máquina de vectores de soporte, Marco referencial Scrum

ABSTRACT

Data classification techniques have a wide variety of applications in different fields due to the advancement of science. Fortunately, there are various classifiers such as neural networks, decision trees, support vector machines, among others.

These techniques can be applied to areas such as medicine, to classify and prevent skin cancer generated by melanoma, a sample is established and with the classification technique it is determined which area has been affected, another important application is related to the environment, and it refers to recycling for waste classification, it also has important contributions for facial recognition among other applications.

Support vector machine techniques to classify data solve an optimization model whose objective function is quadratic and therefore differentiable.

However, if the function were non-differentiable, this technique could not be used, which is why a technique is required that works with differentiable and non-differentiable functions.

According to the relevance of the classification techniques, the Symmetrical Cone Proximal Multiplier Algorithm for Separable Optimization applied to the support vector machine or SVM for its acronym in English (Support Vector Machine) for Binary Classification was developed, this multiplier algorithm works with subdifferentiable, that is, they can find the optimal function, if the function is differentiable or not.

This algorithm is theoretically based on finite-dimensional vector spaces, as well as an inner product, a Euclidean Jordan Algebra, proximal distances, a second-order symmetric cone, among other theoretical concepts.

The binary data classification algorithm was developed testing the good definition, the convergence analysis, the analysis of its complexity.

This algorithm emerges as a new technique which can be used for data classification,

its global convergence was tested under certain assumptions, the algorithm was implemented in the MATLAB software version 2017 (R2017a), on an 8th computer. Gen Intel (R) Core (TM) i5-8250U CPU, 1.60 GHz 1.80 GHz, 4.00 GB, windows 1064 bit.

Subsequently, the design was carried out using the Scrum methodological framework. For this, 3 sprints were developed according to the requirements of the user stories.

A comparative analysis with other proximal algorithms was carried out, as well as computational experiments with the use of the implemented algorithm.

Finally, we will mention that this work is unprecedented and arises from the need to have appropriate techniques to classify data.

Keywords: Proximal Multiplier, Symmetric Cone, Separable Optimization, support vector machine, Scrum Framework

CAPÍTULO I: INTRODUCCIÓN

1.1 Situación Problemática

La línea de investigación de Inteligencia Artificial (IA) es un área de la Computación Científica que está evolucionando significativamente, la sociedad utiliza la tecnología que produce para interactuar, consumir, compartir, etc. De ahí la importancia de tener algoritmos eficientes.

La IA tiene aplicaciones en el campo de la salud, para la detección biosensorial de la tuberculosis, el paquete del dispositivo conocido como nariz digital, que es un enorme elemento sensible sintético construido con la galectina. Los indicadores de hibridación en la unión se pueden capturar y procesar con la ayuda del sistema adquiriendo conocimiento del sensor y la salida se muestra con el uso de estos tipos de algoritmo ver Tamilselvi, S., et al (2021) El uso de la misma para la detección del cáncer de próstata en resonancia magnética (Aphinives, C. y Aphinives, P., 2021) se aplica en la predicción y el diagnóstico temprano de la sepsis utilizando datos no estructurados en la asistencia sanitaria (Goh, K.H. et al., 2021), preparación clínica del diagnóstico basado en imágenes digitales en modelos de pruebas de estrés (Young, A.T. et al., 2021), en el campo de la ingeniería se tiene; Modelos de lodos activados para el pronóstico de lluvias en la agricultura. Una perspectiva sobre el avance del modelado de procesos (Sin, G. y Al, R., 2021) otra aplicación es en el plan de desarrollo de las empresas (Lin, S., 2021).

Otras aplicaciones como un estudio de segmentación automática de la hiperintensidad de la materia blanca para la detección de la enfermedad de Alzheimer; pruebas éticas de piratería y penetración para protegernos de los piratas informáticos; análisis de colas de

prioridad difusa utilizando números difusos heptagonales; clasificación de perturbaciones de la calidad de la energía basada en el filtro de Kalman y el sistema de inferencia neural difusa adaptativa, Ver International Conference on Robotics and Artificial Intelligence (2020).

Debido a ello se requieren algoritmos con buena velocidad de convergencia, existen varios algoritmos que se pueden utilizar como redes neuronales regresión logística, máquinas de vectores de soporte, etc. Uno de los algoritmos importantes para aplicaciones de regresión y clasificación es las máquinas de vectores de soporte.

Las Maquinas de Vectores de Soporte (MVS) es un algoritmo de aprendizaje automático cuyo objetivo principal es hallar un hiperplano en un espacio N - dimensional (N - el número de características) que clasifican la nube de datos.

Mejorar la distancia del margen brinda refuerzo para que los puntos de datos futuros se puedan clasificar con más garantía, es decir nos permiten hallar de modo óptimo la clasificación entre varias clases. Esta separación óptima se efectúa mejorando el margen de separación entre las clases. Aquellos vectores que soportan el borde de esta separación son los llamados vectores de soporte. Si las clases no son linealmente separables, se puede utilizar el truco de Kernel, el cual añade una dimensión nueva donde sí lo sean.

Los algoritmos de máquinas de vectores de soporte tienen sus orígenes en los trabajos de Vapnik, (1995), los primeros estudios era una nueva máquina de aprendizaje para problemas de clasificación de dos grupos.

La red de vectores de soporte combina 3 ideas: la técnica de solución de hiperplanos óptimos (que permiten una expansión del vector de solución en los vectores de soporte), la idea de convolución del producto escalar (que extienda las superficies de la solución de lineal a no lineal), y la noción de márgenes blandos (para permitir errores en el conjunto de entrenamiento). El algoritmo fue probado y comparado con el rendimiento de otros

algoritmos clásicos y obtuvo buenos resultados, posteriormente se realizó aplicaciones a clasificación de imágenes 3D (Blantz, V. et al., 1996).

Los algoritmos de máquinas de vectores de soporte tienen varias aplicaciones en diferentes áreas, como aplicaciones a las ciudades inteligentes, el rápido crecimiento de la población urbana es una serie preocupación para la administración de la ciudad. Esto conduce a varios desafíos al manejar problemas de escasez de recursos y gestionarlos (Rai, P., et al., 2021) otra aplicación importante es en el campo de la salud, el coronavirus COVID-19 en la actualidad es la enfermedad contagiosa más común que prevalece en todo el mundo. El principal desafío de esta enfermedad es el diagnóstico primario para prevenir infecciones secundarias y su propagación de una persona a otra (Sharifrazi, D. et al., 2021), reconocimiento de voz (Aouani, H. y Ben Ayed, Y., 2021), reconocimiento facial (Ahonen, T., Hadid, A., y Pietikainen., 2004), métodos de reconocimiento de rostros humanos basados en el análisis de componentes principales (PCA), wavelet y máquina de vectores de soporte (SVM): un estudio comparativo (Gheni, E.A. y Algelal, Z.M., 2020), algoritmo de clasificación de la máquina de vector de soporte de regresión y su aplicación para mejorar la capacidad de transferencia de calor anticongelante en el sistema de bomba de calor de fuente de tierra (Yan, M., Wang, J.), máquina de vectores de soporte generalizado de múltiples vistas a través de la extracción de la relación inherente entre las vistas con aplicaciones para reconocer el rostro y el humo del fuego (Cheng, Y. et al., 2020), procesamiento de señales (Li, K. et al., 2015) en neurociencia (Park, E. et al., 2017), en Internet de las cosas (IOT) para la protección de seguridad de la privacidad de la ubicación basado en una estructura de árbol y caja anónima (Jingfu Li., 2020), entre otros.

1.2 Formulación del Problema

El desarrollo de la tecnología y exigencias de las aplicaciones en todas las áreas trae consigo el estudio de nuevos algoritmos que sirvan para solucionar los problemas en el campo de la informática, internet de las cosas, computación científica, etc. De ahí la importancia de contar con algoritmos que tengan buena velocidad de convergencia y sean eficientes.

Los métodos punto proximal son muy importantes para la optimización pero que aún no se ha demostrado en los problemas de clasificación, se tiene algunos algoritmos como redes neuronales, máquinas de vectores de soporte, etc.

Presentamos la formulación del modelo:

Dado un conjunto de puntos $x^1, x^2, \dots, x^m \in \mathbb{R}^n$ con sus respectivas etiquetas $y^1, y^2, \dots, y^m \in \{-1, +1\}$, formamos la m-upla $(x^1, y^1), (x^2, y^2), \dots, (x^m, y^m)$. El objetivo de la SVM es determinar un hiperplano óptimo.

$$H(\mathbf{w}, \alpha) = \{\mathbf{x} \in \mathbb{R}^n: \mathbf{w}^T \mathbf{x} + \alpha = 0\}$$

Donde $\mathbf{w} \in \mathbb{R}^n$ y $\alpha \in \mathbb{R}$, que separa los puntos dados.

La formulación del modelo de clasificación binaria que se va a resolver es el siguiente:

$$\begin{aligned} \min_{\mathbf{w}, b, \xi} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m \xi_i \\ \text{s. t. } & y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, i = 1, \dots, m \\ & \xi_i \geq 0, \quad i = 1, \dots, m, \end{aligned}$$

$$\min_{\mathbf{z}, \mathbf{v}} \{f(\mathbf{z}) + g(\mathbf{v}): A\mathbf{z} + B\mathbf{v} = \mathbf{b}, \mathbf{v} \geq 0\},$$

donde las variables son:

$$\mathbf{z} = (\mathbf{w}, \alpha) \in \mathbb{R}^{n+1} \text{ y } \mathbf{v} = (\xi, \mathbf{u}) \in \mathbb{R}^{2m} \text{ con}$$

$$\begin{aligned}\boldsymbol{\xi} &= (\xi_1, \dots, \xi_m) \in \mathbb{R}^m \text{ with } \xi_i \geq 0 \\ \mathbf{u} &= (u_1, \dots, u_m) \in \mathbb{R}^m \text{ with } u_i \geq 0\end{aligned}$$

Las funciones separables son

$$f(\mathbf{z}) = \frac{1}{2} \|\mathbf{w}\|^2 y g(\mathbf{v}) = C \mathbf{e}^T \boldsymbol{\xi},$$

La matriz A y B son las siguientes

$$A = Y\hat{X} \in \mathbb{R}^{m \times (n+1)}, \quad B = (I - I) \in \mathbb{B}^{m \times 2m}$$

dónde

$$Y = \text{Diag}(\mathbf{y}) = \text{Diag}(y^1, y^2, \dots, y^m)$$

y

$$\hat{X} = \begin{pmatrix} x^{1T} & 1 \\ \vdots & \vdots \\ x^m & 1 \end{pmatrix} \in \mathbb{R}^{m \times (n+1)},$$

Finalmente, el vector b viene dado por $\mathbf{b} = \mathbf{e} = (\mathbf{1}, \mathbf{1}, \dots, \mathbf{1}) \in \mathbb{R}^m$.

En el campo de clasificación las máquinas de vectores de soporte tienen múltiples aplicaciones, se puede modelar como un problema de clasificación vinculados a máquinas de vectores de soporte mediante este algoritmo proximal.

Este algoritmo puede resolver modelos de problemas de clasificación, el cual tiene buena velocidad de convergencia.

Presentaremos un algoritmo de punto proximal para problemas de optimización sobre conos simétricos de segundo orden y su aplicación para modelos de clasificación en máquinas de vectores de soporte.

¿De qué manera se podrá introducir un algoritmo multiplicador proximal para problemas de optimización convexa separable sobre conos simétricos y aplicarlo a modelos de clasificación en máquinas de vectores de soporte?

1.3 Justificación de la Investigación

Se ha considerado justificaciones de tipo: teórico, práctico, social y tecnológico debido a las bondades y aplicaciones que se pueden obtener con el uso de este algoritmo, las mismas que están detalladas a continuación:

1.3.1 *Justificación Teórica*

Se desarrolló el Algoritmo Multiplicador Proximal de Cono Simétrico (SCPMA) Debido a que no se tiene algoritmos en cono simétricos de segundo orden, el cual utiliza distancias proximales y puede aplicar a modelos de clasificación binaria, en máquinas de vectores de soporte.

Se realizó la demostración matemática de la convergencia del algoritmo.

1.3.2 *Justificación Práctica*

Se tendrá un nuevo algoritmo para resolver problemas de clasificación binarias para máquinas de vectores de soporte.

El cual puede ser aplicado a inteligencia artificial o Internet de las cosas y problemas específicos como reconocimiento facial, clasificación de imágenes, detección de células cancerígenas entre otras.

el cual requieren algoritmos con buena velocidad de convergencia para los diversos problemas que existen en las distintas áreas del saber.

1.4 Objetivos

1.4.1 *Objetivo General*

Desarrollar un algoritmo multiplicador proximal para resolver problemas de optimización convexa separable sobre conos simétricos y aplicarlo a problemas de clasificación binaria

para máquinas de vectores de soporte.

1.4.2 Objetivos Específicos

- OE1: Definir un algoritmo multiplicador proximal para resolver problemas de optimización convexa separable sobre conos simétricos, verificar la buena definición y analizar su convergencia.
- OE2: Aplicar el algoritmo a la solución de modelos de máquinas de vectores de soporte para clasificación.
- OE3: Implementar y validar el algoritmo para clasificación sometándolo a experimentos computacionales y comparando su performance.
- OE4: Aplicar el algoritmo a un problema específico de clasificación de datos.

1.5 Hipótesis

1.5.1. Hipótesis Generales

En relación con el proyecto de Tesis de investigación se ha definido las siguientes hipótesis:

La introducción de un algoritmo multiplicador proximal para resolver problemas de optimización convexa separable sobre conos simétricos y aplicarlos a problemas de clasificación binaria resuelve modelos de clasificación de máquinas de vectores de soporte.

1.5.2. Hipótesis Específicas

Con la finalidad de contribuir a la hipótesis general definido en el punto anterior como H0 a continuación se detallan las 4 hipótesis denominadas específicas:

- La definición del nuevo algoritmo de clasificación, la verificación de la buena

definición y el análisis de su convergencia brindará un algoritmo que se podrá usar en clasificación.

- La aplicación del algoritmo permitirá resolver modelos de máquinas de vectores de soporte relacionadas a clasificación.
- La implementación y validación del algoritmo para clasificación sometiéndolo a experimentos computacionales permitirá comparar su performance.
- Puede el algoritmo aplicarse a un problema específico de clasificación de datos.

1.6 Propuesta

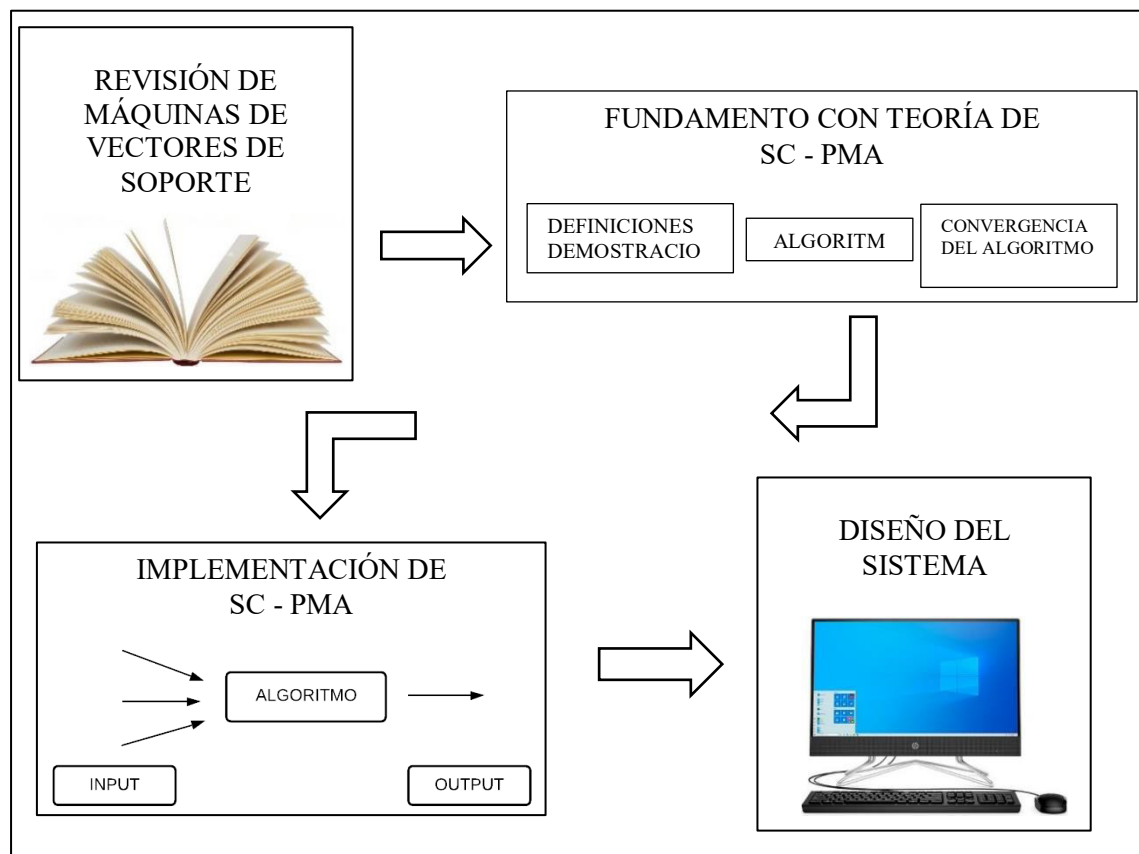


Figura 1 *Esquema de la propuesta de Investigación*

1.7 Organización de la Tesis

En la presente Tesis se organizó en siete capítulos, los mismos que serán descritos a continuación.

En el Capítulo II presentamos el marco teórico de la Tesis, los conceptos necesarios que se utilizarán en el desarrollo del trabajo, seguidamente en el Capítulo III se verá el estado del arte sobre algoritmos de máquinas de vectores de soporte, las aplicaciones en los distintos campos y los diferentes tipos de algoritmos de clasificación SVM por sus siglas en inglés Support Vector Machines.

En el Capítulo IV se presenta la formalización teórica del algoritmo Multiplicador Proximal de Cono Simétrico, se expondrán los conceptos previos, demostrarán teoremas y lemas. Además, se pondrá bajo algunos supuestos la convergencia del algoritmo.

En el Capítulo V se presenta la descripción del tipo de investigación, en el Capítulo VI la implementación del algoritmo propuesto el mismo que se implementa en el software Matlab, con sus subprogramas, como el generador lineal, el programa principal y el cvx para la comparación de los resultados.

En el Capítulo VII presentamos el diseño del sistema desarrollado con el marco referencial Scrum e implementado en java, además de su base de datos, arquitectura del programa, arquitectura de usuario, etc.

1.8 Variables

1.8.1. Identificación de Variables

Las variables para esta Tesis esta categorizados en:

- Variables Dependientes: Clasificación binaria en máquinas de vectores de soporte.
- Variables Independientes: Algoritmo multiplicador proximal.

CAPÍTULO II: MARCO TEÓRICO

2.1 Antecedente del Problema

Con el propósito de encontrar información relevante que aporte o que sirva como guía para el desarrollo de la propuesta de la tesis, se realizó el estado de arte mediante una revisión sistemática de los algoritmos que resuelvan problemas de clasificación y los métodos de punto proximal con sus aplicaciones.

En los inicios para la clasificación de datos se utilizaba La red neuronal artificial, el cual es la imitación de una neurona humana en un sistema. Esto se hace para desarrollar un sistema que pueda realizar las actividades del cerebro humano. Hay muchos algoritmos que han sido utilizados en Redes Neuronales artificiales.

Un ejemplo de dicho algoritmo es el perceptrón multicapa. El Perceptrón multicapa entrenado con la red de retro propagación mediante la actualización del peso sináptico y la prueba mediante red de avance. El objetivo óptimo de este sistema de aprendizaje es tener un error mínimo que es relativo a tener salidas que es igual o más cercano al objetivo. Posteriormente surge la máquina de vectores de soporte es una nueva técnica supervisada de aprendizaje, que han demostrado ser especialmente atractivas para análisis biológico debido a su capacidad para manejar el ruido y grandes espacios de entrada (Ding & Dubchak, 2001).A diferencia del algoritmo de perceptrón que encontrar cualquier separador lineal con la ayuda de algún criterio.

Las máquinas de vectores de soporte en particular definen el criterio buscando una superficie de decisión que esté lo más lejos de cualquier punto de datos (Manning, Raghavan, & Schütze, 2008).El margen del clasificador está determinado por la distancia desde la superficie de decisión hasta el punto de patrón más cercano. Esto significa que la

función de decisión del patrón generalmente se especifica mediante el subconjunto del patrón o pequeña porción del patrón que especificar la posición del clasificador. La pequeña porción del patrón o un subconjunto del patrón se refiere como soporte vector. La idea es buscar un hiperplano optimo clasificador.

Fue introducido por (Cortes & Vapnik, 1995) ,este modelo inicial para problemas de clasificación presenta un espacio de características, se construye una superficie de decisión lineal, además este modelo presenta en su desarrollo transformaciones de entradas polinómicas.

Utiliza inicialmente un hiperplano clasificador en el cuadrante no negativo, aunque anteriormente algunos autores que presentaron modelos de clasificación, estos no consideraban el error en su modelo.

En la actualidad, el artículo (Cortes & Vapnik, 1995) ha sido citado por diferentes autores por ejemplo en Scopus podemos verificar que tiene 35 806 citas., los cuales tienen aplicaciones a clasificación de datos, clasificación de imágenes, reconocimiento facial, entre otros.

En esta Tesis, hemos ampliado el método de descomposición proximal dado en (Chen & Teboulle, 1994) (Auslender, A.; Teboulle, M., 2001) (Kyono & Fukushima, 2000) (Sarmiento, O.; Papa Quiroz, E.; Oliveira, P., 2014) para resolver problemas de programación de conos simétricos convexos.

La extensión se basa en proximal distancia dada en (Auslender, A.; Teboulle, M., 2006) .Una clase de esta distancia puede ser generada por una función ϕ que pertenece Al conjunto Σ o $\hat{\Sigma}$. Se han presentado algunos ejemplos de esta distancia. Bajo ciertas suposiciones y la distancia proximal, hemos establecido que cada punto límite es un punto silla del Lagrangiana asociado del problema. Además, se ha obtenido la convergencia global.

Recientemente tenemos los trabajos desarrollados por (Ruiz Garrido, 2016) para problemas con estructura separables denominado; Método de Direcciones Alternantes con Regularización Proximal Interior (RIPADM), este algoritmo utiliza un factor de relajación

ρ , el cual está comprendido en cierto intervalo $\rho \in (0, \frac{\sqrt{5}+1}{2})$ y añade a la regla de actualización del multiplicador de Lagrange para la aceleración de convergencia del algoritmo.

Finalmente, Sarmiento et al, ver (Sarmiento, O.; Papa Quiroz, E.A.; Oliveira, P. R., 2015) desarrolló una extensión del Algoritmo Multiplicador Proximal con distancias proximales (PCPM) para resolver con estructura separable, en el espacio vectorial euclidiano, utilizando distancias proximales regularizadas. Bajo determinadas condiciones se verifica para una amplia clase de distancias proximales inducidas.

Además, este algoritmo converge linealmente a una solución óptima del problema.

2.1.1 Métodos Punto Proximales

Los métodos proximales son muy utilizados en ingeniería, teoría económica, etc. Según Verma, M. y Shukla, K.K (2020) estos métodos son populares para manejar la estructura de escasez en los conjuntos de datos debido a sus bajos costos de iteración y una convergencia más rápida.

En su trabajo considera el marco de la suma de dos funciones convexas, una de las cuales es una función suave con un gradiente de Lipschitz, mientras que la otra puede ser una función no suave, presenta el análisis de convergencia para los sistemas fijos basados en extra-grados. Además, ampliando las áreas de aplicación de este algoritmo, lo aplicamos para resolver (i) la regresión logística problema con penalizaciones complejas basadas en ℓ_1 , a saber, marcos de lazo de grupo superpuestos y de lazo fusionado, y (ii) un aprendizaje estructuralmente regularizado propuesto recientemente problema para la selección de representación donde la función objetivo consiste en un error de reconstrucción y regularizadores estructurados como combinación de regularizador de escasez de grupo, regularizador de diversidad y regularizador de sensibilidad de localidad. Con la ayuda de extensos experimentos en varios conjuntos de datos del mundo real disponible públicamente, demostró la eficacia de los métodos de extra-grado basados en la inercia para resolver la selección de representación y lazo extendido problemas del aprendizaje

automático. Otro trabajo del algoritmo proximal lo encontramos en Sahu, D.R., et al (2021) propone una nueva modificación proximal punto algoritmo basado en el proceso de iteración de tipo S para aproximar un elemento común del conjunto de soluciones de problemas de minimización convexa y el conjunto de puntos fijos de asignaciones casi asintóticamente cuasi no expansivas en el marco de espacios CAT (0) y probar la Δ -convergencia de la propuesta algoritmo para resolver problemas comunes de minimización y arreglos comunes del problema. Otro de los recientes trabajos que realiza una aplicación a un modelo de máquina de vectores de soporte es la investigación de Quiroz, E.A.P. et al (2020) el cual está dedicado al estudio de un algoritmo de descomposición proximal para resolver la optimización de conos simétricos convexos con estructuras separables. El algoritmo considerado se basa en un método de descomposición y distancias proximales. Bajo supuestos adecuados, probamos que cada punto límite de las secuencias primal-dual generadas por el algoritmo resuelve el problema. Finalmente, se establece la convergencia global.

El trabajo realizado por Cano Lengua, M.A., Quiroz, E.A.P. (2020) está relacionado a resolver un problema de la ubicación, el cual es de gran interés para establecer distintos requerimientos relacionados al sector público o privado. El modelo de este problema suele reducirse a un problema de optimización matemática. En este artículo damos a conocer un método proximal que resuelve problemas de ubicación donde la función objetivo es cuasi-convexa y no diferenciable. Se demuestra que las iteraciones dadas por el método están bien definidas y bajo algunos supuestos sobre la función objetivo, es posible verificar la prueba de la convergencia del método.

Maldonado, S. et al., (2020), en este estudio presenta un sistema experto para analizar la carga de trabajo mental de interactuar con un teléfono móvil frente a tareas cotidianas habituales. Se recopilaron señales psicofisiológicas de varios dispositivos, cada uno de los cuales se caracterizó por un costo y una molestia diferente. Para tratar con datos de señal a nivel de usuario, se propone un enfoque de selección de características basadas en máquinas de vectores de soporte. Dada la limitada información disponible a nivel de persona, nuestro objetivo era construir modelos sólidos mediante la combinación de

información a nivel de población entre usuarios (como control de heterogeneidad). Se propone un único problema de optimización que combina cuatro objetivos: modelo, maximización de márgenes, selección de características y control de heterogeneidad. Se estimaron los costos de uso de los dispositivos, conduciendo a una herramienta de decisión que permitió a los diseñadores de experimentos evaluar el beneficio marginal de usar un dispositivo dado en términos de rendimiento y su costo.

2.1.2 Algoritmo de Clasificación Máquinas de Vectores de Soporte

Estos algoritmos se utilizan métodos proximales en el sentido de la proximidad al hiperplano de separación. Se realizó una búsqueda de estos tipos de algoritmos basados en recientes investigaciones, en trabajo de Cano Lengua, M.A., Papa Quiroz, E.A. (2020), en este artículo tiene como finalidad identificar el estado del arte actual de las últimas investigaciones relacionadas con las máquinas de vectores de soporte a través de un sistema de revisión de la literatura según la metodología propuesta por Kitchenham y Charter, con el fin de dar respuesta a las siguientes preguntas de investigación: P1: ¿En qué áreas de investigación se utilizan? P2: ¿Cuáles son las principales aplicaciones relacionadas con la clasificación? P3: ¿Qué métodos o algoritmos de optimización se utilizan en las SVM?, según este artículo las aplicaciones de SVM están dadas en distintas disciplinas según la Figura 2

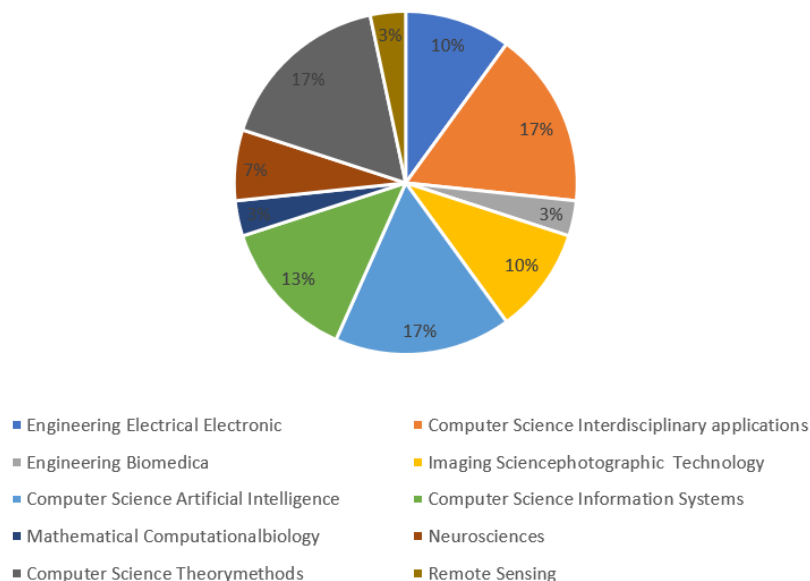


Figura 2 Áreas de Investigación

En la Figura 2 se muestra el porcentaje de distribución según las áreas de investigación, principalmente vemos que el 17% está relacionado con Ciencias de la Computación y el 10% Ingeniería Eléctrica Electrónica e Informática en Sistemas de Información.

Esta Tesis nos permite saber que SVM tiene un número considerable de aplicaciones en las diferentes áreas. Cabe mencionar que también existen aplicaciones en regresión, como los métodos de pronósticos, que pueden ser objeto de otras investigaciones.

En el campo de la salud se tiene este trabajo de Patil, A.K. et al (2019) que está vinculado la clasificación automática del cáncer del pulmón, el cual sigue siendo un desafío para los investigadores, debido a los ruidosos datos de expresión génica, los datos de alta dimensión y el pequeño tamaño de la muestra. Para abordar estos problemas, se desarrollan un algoritmo de selección de genes mejorado y un clasificador multiclase. En esta investigación, los genes relacionados con el cáncer de pulmón (GEO ID: GSE10245, GSE19804, GSE7670, GSE10072 y GSE6044) se recopilaron del conjunto de datos de Gene Expression Omnibus (GEO). Después de adquirir los genes relacionados con el cáncer de pulmón, la selección de genes se llevó a cabo utilizando un algoritmo de relieve mejorado para seleccionar los genes óptimos. En el algoritmo de selección de genes de relieve mejorado, se utilizaron la medida de distancia de movimiento de tierras y el

optimizador de luciérnagas en lugar de la medida de distancia de Manhattan para identificar las instancias de falla más cercana y de impacto más cercano lo que reduce significativamente el problema de la “maldición de la dimensionalidad”. Estos genes óptimos se proporcionaron como entrada para el clasificador Multiclass Support Vector Machine (MSVM) para clasificar las subclases de cáncer de pulmón. La selección experimental mostro que el sistema propuesto mejoró la precisión de clasificación hasta un 3-10% en relación con los sistemas existentes a la luz de la precisión, tasa de falsos positivos (FPR), tasa de error y tasa de verdaderos positivos (TPR).

El siguiente trabajo de Zheng, X. et al (2021) CTSVM (Twin Support Vector Machine With Correntropy) está diseñado para realizar las siguientes optimizaciones: minimización de la función C-loss, la regularización de 2 normas de los coeficientes del modelo y la distancia entre las muestras positivas (negativas) y el hiperplano de clase positiva (clase negativa), lo que garantiza que las muestras negativas (positivas) están lejos del hiperplano positivo (negativo). CTSVM se convierte en dos problemas de optimización semicuadráticos que se pueden resolver mediante un método iterativo alternativo. Este enfoque permite que el tiempo de cálculo de CTSVM siga siendo comparable al de los métodos relacionados, especialmente al resolver problemas lineales. Los resultados experimentales se procesaron estadísticamente para comparar el rendimiento de múltiples algoritmos en múltiples conjuntos de datos. Estos resultados confirmaron que CTSVM supera a clasificadores similares en términos de su solidez a valores atípicos y precisión de clasificación para tareas binarias.

Cheng, Y., et al (2020) se trata de un nuevo método denominado Máquina vectorial de soporte generalizado multivisorio mediante la minería de la relación inherente entre vistas (MRMvGSVM Multiviewer Generalized Support Vector Machine by Inherent Relationship Mining between Views, MRMvGSVM). En este algoritmo se ha llevado a cabo un estudio adicional sobre el aprendizaje de múltiples vistas, conociendo así las conexiones entre las vistas de muestra desde una nueva dirección y estableciendo un marco de múltiples vistas, socavando las relaciones internas entre diferentes puntos de vistas. Otro trabajo de las máquinas vectores de soporte relacionado a clasificación es el

de Hou, Q. et al (2018), este trabajo con algoritmos no lineales a clasificación denominados projection nonparallel support vector machine (PNPSVM), que es completamente diferente de los clasificadores no paralelos existentes. Construir dos problemas de optimización para obtener dos direcciones de proyección que logran la máxima separabilidad de la clase minimizando la distancia dentro de la clase y maximizando a la distancia entre clases simultáneamente, y luego las ubicaciones específicas de los hiperplanos proximales correspondientes están determinadas por un centro geométrico adecuado basado en el a priori subyacente.

La enfermedad del cáncer ha creado una grave amenaza para la vida de las personas durante más de una década. A medida que aumenta el tiempo, los problemas y casos relacionados tienden a crecer a mayor escala con respecto a los diferentes estilos de vida que llevan las personas. Uno de estos tipos de cáncer, que juega un papel dominante en la muerte de muchas mujeres, es el cáncer de cuello uterino. Es el trabajo de Suresh, K. (2021), ayuda a clasificar la presencia de cáncer de cuello uterino entre los sujetos. Los modelos se construyen en función de los diferentes tipos de factores de entrada, como el estilo de vida, los hábitos, la historia clínica de los pacientes y la práctica sexual del paciente, este trabajo analiza las características importantes en el conjunto de datos responsable de la causa del cáncer de cuello uterino mediante la identificación de la importancia de la variable utilizando un enfoque de bosque aleatorio.

2.2 Bases Teóricas

Para un mejor estudio desarrollaremos los conceptos más relevantes del tema de Tesis

2.2.1 Algoritmo

Un algoritmo es un procedimiento de un conjunto ordenado de pasos, el cual se sigue para obtener un resultado. Los algoritmos generalmente se desarrollan independientemente de los lenguajes de programación.

- Características de un Algoritmo:
- Entrada: parámetros a ingresar
- Salida: resultados procesados
- No ambiguo: Realiza la acción requerida.
- Finito: Cantidad de pasos finitos.
- Correcto: Realiza el cálculo con eficiencia
- Efectivo: Realiza el cálculo con rapidez
- Definido: Si se sigue dos veces, se obtiene el mismo resultado. (Esteva G., 2016)

2.2.2 Método del Punto Proximal

El Método de Punto Proximal (MPP), fue incorporado por Martinet (1970), y posteriormente utilizado por Rockafellar (1976) cuyo objetivo era ubicar los ceros de operadores monótonos maximales, principalmente podía resolver problemas irrestrictos como;

$$\min\{f(x): x \in \mathbb{R}^n\},$$

Donde $f: \mathbb{R}^n \rightarrow \mathbb{R}$ es una función convexa. El algoritmo de punto proximal forma una sucesión $\{x^k\}$, a través del siguiente proceso iterativo: con un punto de inicio arbitrario $x^0 \in \mathbb{R}^n$, calcular

$$x^{k+1} = \operatorname{argmin} \left\{ f(x) + \frac{\lambda_k}{2} \|x - x^k\|^2, x \in \mathbb{R}^n \right\},$$

Donde $\lambda_k > 0$ es el parámetro de regularización, $\|\cdot\|$ denota la norma Euclidiana, y la notación argmin^1 significa que x^k es el único punto mínimo global de $f(x) + \frac{\lambda_k}{2} \|x - x^k\|^2, x \in \mathbb{R}^n$. Si $\{\lambda_k\}$ satisface $\sum_{k=1}^{+\infty} \frac{1}{\lambda_k} < +\infty$ se demuestra que la sucesión $\{f(x^k)\}$ converge al ínfimo de f ; y si además el conjunto de soluciones óptimas es no vacío, entonces $\{x^k\}$ converge a una solución óptima del problema, ver Guler (1992).

2.2.3 Optimización Simétrica del cono convexo (SCP) con estructura separable

Daremos dos espacios vectoriales dimensionales finitos en el campo real equipados con el producto interior y respectivamente. A continuación, considere el siguiente algebra de

Euclidán Jordiana y , donde y denota el producto jordaniano en y , respectivamente.

Sea $\mathbb{V}_1$ y $\mathbb{V}_2$ dos espacios finitos vectoriales sobre $\mathbb{R}$ con el producto interno $\langle \cdot, \cdot \rangle_{\mathbb{V}_1}$ y $\langle \cdot, \cdot \rangle_{\mathbb{V}_2}$ respectivamente. Consideremos el Álgebra Euclideano de Jordan $\mathbb{V}_1 = (\mathbb{V}_1, \circ_1, \langle \cdot, \cdot \rangle_{\mathbb{V}_1})$ y $\mathbb{V}_2 = (\mathbb{V}_2, \circ_2, \langle \cdot, \cdot \rangle_{\mathbb{V}_2})$, donde $\circ_1$ y $\circ_2$ denota el producto interno $\mathbb{V}_1$ y $\mathbb{V}_2$ respectivamente.

Consideremos la estructura de cono simétrico de segundo orden:

$$\min\{f(x) + g(z): Ax + Bz = b, x \in k_1, z \in k_2\}, \quad (2.1)$$

Donde $f: \mathbb{V}_1 \rightarrow \mathbb{R} \cup \{+\infty\}$ y $g: \mathbb{V}_2 \rightarrow \mathbb{R} \cup \{+\infty\}$ son funciones propias cerradas y convexas, $A: \mathbb{V}_1 \rightarrow \mathbb{R}^m$ y $B: \mathbb{V}_2 \rightarrow \mathbb{R}^m$ son dos mapeos lineales, $b \in \mathbb{R}^m$, y $k_1 := \{x \circ_1 x: x \in \mathbb{V}_1\}$ y $k_2 := \{z \circ_2 z: z \in \mathbb{V}_2\}$ y la denotación de los conjuntos de elementos cuadrados (conos simétricos) en $\mathbb{V}_1$ y $\mathbb{V}_2$, respectivamente.

Este problema proporciona un marco unificado simple para varios problemas de optimización existente, como la programación no lineal, cono de segundo orden y semidefinida, y por lo tanto tiene amplias aplicaciones en ingeniería, economía, teoría de juegos, ciencias de la gestión y otros campos; véase (Alizadeh, F. y Goldfarb.; 2012), (Alvarez, F. et al, 2012), (Li, A. and Tuncel, L., 2006), (Vandenberghe, L. y Boyd, S., 1994). En particular, una clase de problemas de (2.1) han llamado recientemente mucha atención debido a sus aplicaciones emergentes de interés práctico, como por ejemplo las máquinas vectoriales de apoyo o la escasa selección inversa de covarianza.

Recientemente, SCP ha traído la atención para algún investigador que se centra en el desarrollo de métodos de punto interior, ya que la clase de conos simétricos contiene octante positivo en el cono de segundo orden, y en el cono de matrices simétricas semidefinidas positivas en $\mathbb{R}^n$.

2.2.4 Modelos de Máquina de vectores de soporte utilizando como simétrico de segundo orden con estructura separable

Las máquinas de vectores de soporte en un buen algoritmo de clasificación el algoritmo que diseñaremos se podrá modelar como un problema de optimización que resuelve máquinas de vectores de soporte en ese sentido utilizará la teoría de optimización.

Proporcionamos la aplicación a máquinas de vectores de soporte (SVM) para clasificación binaria. Plantearemos el problema, dado un conjunto de instancias (x_i, y_i) , donde $x_i \in \mathbb{R}^n, i = 1, \dots, m$ y $y_i \in \{-1, +1\}$, el SVM determina un hiperplano óptimo $H(w, b) = \{x \in \mathbb{R}^n : w^T x + b = 0\}$, que separa los cascos convexos reducidos de ambos patrones de entrenamiento. Este hiperplano óptimo se puede encontrar resolviendo el siguiente problema de programación cuadrática:

$$\begin{aligned} \min_{w, b, \xi} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \\ \text{s. t. } & y_i (w^T x_i + b) \geq 1 - \xi_i, i = 1, \dots, m \\ & \xi_i \geq 0, \quad i = 1, \dots, m, \end{aligned} \quad (2.2)$$

Donde $\xi = (\xi_1, \dots, \xi_m) \in \mathbb{R}^m$ y $C > 0$ es un parámetro de penalidad.

Vamos a denotar $z = (w, b) \in \mathbb{R}^{n+1}$, de $Diag(y)$ es una matriz diagonal de vector $y = (y_1, \dots, y_m)$, sea $\mathbf{1}$ un vector de unos en $\mathbb{R}^m$, y por

$$\hat{X} = \begin{pmatrix} x_1^T & 1 \\ \vdots & \vdots \\ x_m^T & 1 \end{pmatrix} \in \mathbb{R}^{m \times n+1},$$

A continuación, la primera restricción de desigualdad se puede escribir como mediante la introducción de las variables y , el problema (2.2) se puede reescribir como $Diag(y)\hat{X}z + \xi - \mathbf{1} \geq 0$. Introducimos las variables $u = Diag(y)\hat{X}z + \xi - \mathbf{1}$ y $v = (\xi, u) \in \mathbb{R}^{2m}$, el problema (2.2) se puede reescribir como:

$$\min_{z, v} \{f(z) + g(v) : Az + Bv = b, v \geq 0\}, \quad (2.3)$$

Donde $f(z) = \frac{1}{2} \|w\|^2, g(v) = C \mathbf{1}^T \xi, A = Diag(y)\hat{X} \in \mathbb{R}^{m \times n+1}, B = (I - I) \in \mathbb{R}^{m \times 2m}$, y $b = \mathbf{1}$. Este modelo tiene la forma de (2.1).

2.2.5 Modelo de máquinas vectoriales de soporte gemelas (TWSVM)

El modelo de máquinas de vectores soporte gemelas (TWSVM), presenta una norma mixta el cual resuelve el problema siguiente:

$$\min_{w_1, t_1} \|D_1 w_1 + e_1 t_1\|_\infty + \frac{c_1}{2} (\|w_1\|^2 + t_1^2) \quad (2.4)$$

$$st. -(D_2 w_1 + e_2 t_1) \geq e_2, y$$

$$\min_{w_2, t_2} \|D_2 w_2 + e_2 t_2\|_\infty + \frac{c_2}{2} (\|w_2\|^2 + t_2^2) \quad (2.5)$$

$$st. -(D_1 w_2 + e_1 t_2) \geq e_2$$

Donde $\|\cdot\|_\infty$ denota la norma infinita en el espacio Euclidiano $D_1 \in \mathbb{R}^{m_1 \times n}$ y $D_2 \in \mathbb{R}^{m_2 \times n}$ son matrices que contienen el conjunto de datos de entrenamiento de clase positiva y negativa, respectivamente c_1 y c_2 son parámetros positivos $e_1 \in \mathbb{R}^{m_1}$ y $e_2 \in \mathbb{R}^{m_2}$ son vectores cuyas componentes son unos.

Solucionar el problema (2.3) consideremos $z = [w_1^T, t_1^T] \in \mathbb{R}^{n+1}$, el problema (2.4) puede ser reescrito de la forma:

$$\min_z \left\{ \|[D_1 e_1]z\|_\infty + \frac{c_1}{2} \|z\|^2 - [D_2 e_2]z \geq e_2 \right\}, \quad (2.6)$$

Introduciendo una variable de holgura $x \in \mathbb{R}^{m_2}$ a las restricciones de desigualdad del problema anterior, podemos reformularlo en forma de (2.1) con $f(x) = 0$,

$g(z) = \|[D_1 e_1]z\|_\infty + \frac{c_1}{2} \|z\|^2$, $A = I$, $B = [D_2 e_2]z \in \mathbb{R}^{m_2 \times n+1}$ y $b = -e_2$. Es decir, el problema (2.6) es equivalente a:

$$\min_z \left\{ \|[D_1 e_1]z\|_\infty + \frac{c_1}{2} \|z\|^2 / x + Bz = -e_2, x \geq 0 \right\}, \quad (2.7)$$

2.2.6 Optimización matemática

Optimizar significa buscar la mejor opción de desarrollar una tarea, es decir, hallar el máximo o mínimo de una cierta función, definida en algún dominio. La optimización es un proceso para ubicar la mejor solución de un problema donde “lo mejor” esta con antelación bajo determinados criterios preestablecido.

La programación matemática tiene aplicación en diferentes áreas, el cual se encarga de la teoría, métodos y aplicaciones computacionales para resolver problemas relacionados a optimización con determinadas condiciones.

En el campo de la matemática aplicada, la aplicación de la optimización matemática es relevante sobre todos en las áreas de ingeniería y economía.

El modelo general de optimización esta dado por:

$$\begin{aligned} & \text{Opt. } f(x) \\ & \text{s. a:} \\ & g_i(x) \leq 0; \forall i = 1, \dots, m \\ & h_j(x) = 0; \forall j = 1, \dots, p \end{aligned} \tag{2.8}$$

Donde $f: \mathbb{R}^n \rightarrow \mathbb{R}$: es una función dada, $\text{Opt. } f(x)$ significa minimizar o maximizar la función f , además $g: \mathbb{R}^n \rightarrow \mathbb{R}$ y $h_j: \mathbb{R}^n \rightarrow \mathbb{R}$ son funciones dadas.

Una clase particular y bien amplia del modelo (2.8) es la optimización cuasi-convexa (problemas donde la función objetivo f es cuasi-convexa, esto es,

$$f(\lambda x + (1 - \lambda)y) \leq \max\{f(x), f(y)\}, \forall \lambda \in [0,1], \forall x, y \in \mathbb{R}^n$$

y las funciones que definen las restricciones g_i y h_j son cuasi-convexas).

Este problema fue estudiado por Arrow y Enthoven (1961), motivados por las aplicaciones de las preferencias y utilidades en la teoría del consumidor y posteriormente por diversos investigadores como Kannai (1997).

A continuación, presentamos un ejemplo de Clasificación mediante máquinas de vectores de soporte.

La formulación del problema está dada por:

$$\min_w \frac{1}{2} w^T w + C \sum_{t=1}^l \xi(w; x_t, y_t)$$

Sujeta a ciertas restricciones y donde $\xi(w; x_t, y_t)$ es la función de pérdida.

Aplicando la teoría lagrangiana se obtenemos su dual

$$s. t \quad 0 \leq \alpha_i \leq U \quad \forall i$$

$$\min_{\alpha} f(\alpha) = \frac{1}{2} \alpha^T \hat{Q} \alpha - \mathbf{e}^T \alpha$$

Este modelo se resuelve mediante el algoritmo Dual Coordinado de Descenso (DCD) el cual resuelve el problema de SVM.

Podemos observar que el dual se puede reescribir como una función cuadrática de d :

$$f(\alpha^{k,i} + de_i) = \frac{1}{2} \hat{Q}_{ii} d^2 + \nabla_i f(\alpha^{k,i}) d + cte$$

Donde $\nabla_i f$ es la componente i del gradiente total de f , ∇f .

Para encontrar la solución a este problema vemos que la función debe ser diferenciable, es decir por teoría se sabe que la gradiente de la función coincide con la derivada el punto óptimo, por lo tanto se obtendría la solución óptima, sin embargo si la función no fuera diferenciable no se podría resolver el problema, ya que se tendría que utilizar sus diferenciables.

El SC - PMA trabaja con un subdiferencial convexo el cual podemos visualizar en el Capítulo IV, es ahí una limitación que tiene las técnicas de SVM con respecto al algoritmo de solución. (Gala García , 2013)

2.3 Marco Conceptual

Se tiene algoritmos de clasificación de máquinas de vectores de soporte que utilizan métodos proximales con relación a la distancia tiene aplicaciones a las distintas áreas del saber entre estos algoritmos tenemos:

MvGEPSVMs. Multiview Generalized eigenvalue proximal support vector machines es un algoritmo de clasificación binaria simple y eficaz.

RNPSVM Regularized robust nonparallel proximal support vector machine for noisy classification, este algoritmo se utiliza en modelo de clasificación ruidosa.

IMvGEOSVM Improved Multiview Generalized Eigenvalue Proximal Support Vector Machine, este algoritmo tiene aplicaciones a vistas múltiples.

WLTSVM Weighted Lagrangian Twin Support Vector Machine, este algoritmo propone un Lagrangiano ponderado eficiente, para la clasificación de datos desequilibrados basada en el uso de diferentes puntos de entrenamiento para construir los dos hiperplanos proximales. Entre otros algoritmos que trabajan con métodos proximales.

CAPÍTULO III: ESTADO DEL ARTE

3.1. Introducción

Este artículo presenta una revisión sistemática sobre la máquina de vectores de soporte (SVM) en el campo de la clasificación, en particular, nos interesan los algoritmos o métodos más utilizados para la clasificación. Este trabajo completa el trabajo inicial (Cano & Papa, 2020) donde presentamos una revisión preliminar basada en una base de datos reducida y respondimos tres preguntas de investigación. En este trabajo, consideramos todas las bases de datos más importantes, actualizamos la investigación hasta setiembre de 2021, presentamos información detallada relacionada con cada pregunta y agregamos una nueva pregunta relacionada con los hiperplanos proximales debido a la importancia de estos métodos en las aplicaciones. Observe también que aunque SVM se ha desarrollado como un método de clasificación binaria (Cortes & Vapnik, 1995), sus aplicaciones se han extendido a múltiples problemas de clasificación y regresión, ver (Yang, Qu, Cai, Zhang, & Zhao, 2020) (Hamza & Ayed, 2020) (Almahdi, Yassen, & Dakhil, 2021) (Silva & Machado, 2021) (Revathi, Anitha, Jude, & Hemanth, 2021) (Lidula, Perera, & Rajapakse, 2009) (Le, Parvathy, & Gupta).

En este artículo, utilizamos la metodología de Kitchenham y Charte (Kitchenham & Charters, 2007) y presentamos una revisión ampliada de la literatura sobre SVM para la clasificación. Los aportes de este artículo serán mostrar diversas aplicaciones de la SVM en diversas áreas de investigación, aplicaciones recientes de la SVM en clasificación y, entre otras cosas, los algoritmos o métodos de optimización más utilizados para resolver modelos de SVM; lo cual será importante para construir nuevos algoritmos eficientes en esta línea de investigación.

La organización del trabajo es la siguiente: En la Sección 3.2, presentamos la metodología de investigación desarrollada en este trabajo. En la Sección 3.3, analizamos la información obtenida. En la Sección 3.4 presentamos los resultados de cada pregunta de investigación y en la Sección 3.5 presentamos nuestras conclusiones. Finalmente, damos las referencias de la obra.

3.2. Metodología de Investigación

La revisión sistemática que utilizaremos fue propuesta por Kitchenham y Charter (Kitchenham & Charters, 2007). Esta metodología posibilita la identificación, evaluación e interpretación de toda la investigación encontrada respecto a cuestiones especiales, áreas temáticas o fenómenos de interés. Varios autores han utilizado esta metodología, ver (Vargas & Mauricio, 2018) (Wong, Mauricio, & Rodriguez, 2017) y referencias allí. Por otro lado, existen otras metodologías, por ejemplo (Kostoff, 2008) y (Porter & Newman, 2011), que no encajan en el campo de la ingeniería.

La metodología comprende las siguientes fases:

- Planificación de la revisión: En esta parte se justifica la revisión, se formulan las preguntas de investigación y se diseña el protocolo de búsqueda.
- Realización de la revisión: Continúa con la búsqueda de documentos, selección de documentos según los criterios de inclusión y exclusión, evaluación de calidad, finalmente extracción de datos.
- Informe de la revisión: Se presenta las estadísticas, el análisis de los documentos seleccionados, las conclusiones y discusiones.

3.2.1 Planificación de la Revisión

Estamos interesados en responder las siguientes preguntas:

- P1: ¿En qué áreas de investigación se utilizan las SVM?
- P2: ¿Cuáles son las últimas aplicaciones relacionadas?
- P3: ¿Qué métodos o algoritmos de optimización se utilizan en SVM?
- P4: ¿Cuáles son los métodos relacionados con los hiperplanos proximales en SVM para la clasificación?

La búsqueda se realizó en las siguientes bases de datos: Scopus, IEE Xplore Digital Library, ACM Digital Library, Web of Science, Springer Link, Science Direct y Taylor & Francis. La revisión cubre el periodo de 1992 a 2021.

Se utilizó la siguiente búsqueda de flujo:

- TITTLE-ABS-KEY (“areas where SVMs are used”) o TITTLE-ABS-KEY (“Classification Related Applications”) o TITTLE-ABS-KEY (“Optimization algorithms used in SVMs”), o TITTLE-ABS-KEY (“Which methods thought in proximal hyperplanes towards the data set have been studied in SVM”)
- TITLE-ABS-KEY ((algorithms OR proximal)AND(svm OR “support AND vector AND machine”)

Se consideró Journal y conference.

Los mismos que se han aplicado sobre el título, resumen y palabras clave. Luego, se aplicaron los criterios de selección y exclusión mostrados en las Tabla 1 y Tabla 2

3.2.2 Desarrollo de la Revisión

Tabla 1
Criterios de Inclusión

Criterio	Justificación
Enfoque de la investigación	Documentos con las máquinas vectoriales de soporte de texto como palabra clave.
Información cuantitativa	Los artículos deben estar publicados en revistas indexadas.
Factor de importancia	SVM vinculado a la clasificación y aplicado a la Inteligencia Artificial
Idioma	Inglés

Tabla 2
Criterios de Exclusión

Criterio	Justificación
Tipo de publicación	Libros, disertaciones, capítulos de libros entre 1990-2000 (No actualizado)
Unidad de análisis	Artículos que no consideren SVM como su foco de interés
Factor de Importancia	Investigaciones que no presentan claramente una metodología de desarrollo

La figura 1 muestra las fuentes de las que se obtuvo la investigación. Observe que en la Figura 3 otros incluyen las fuentes Science Direct y Taylor & Francis.

Para la obtención de los 73 trabajos finales utilizamos criterios de inclusión y exclusión y filtramos los trabajos según la justificación.

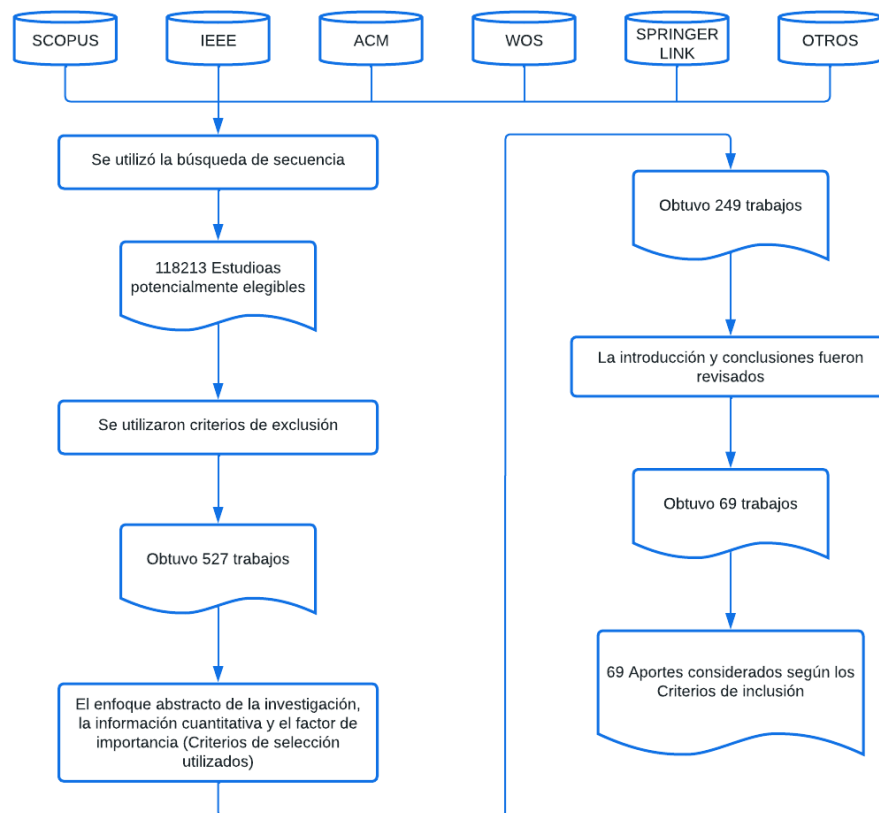


Figura 3 Proceso de Revisión Sistemática de la literatura

3.2.3 Resultados de la Revisión

Se encontraron 118213 artículos relacionados con la clasificación de SVM, los cuales son aplicados a las diferentes áreas de investigación, ver Figura 3. Enumeramos el número de artículos fundados en las bases de datos.

Tabla 3
Seleccionar artículos

Fuente	Estudio potencialmente elegibles	Estudios seleccionados
Scopus	11 435	29
IEE Xplore	8 401	15
ACM	43 021	7
Web of Science	6 185	6
Springer Link	47 914	6
Otros	1 257	6
Total	118 213	69

Además, utilizando la búsqueda de flujo en SCOPUS (SCOPUS, 2021), encontramos las tendencias de las publicaciones presentadas en la Figura 4. También encontramos que el primer artículo publicado indexado en SCOPUS sobre SVM aplicado a la clasificación fue (Bonneville, Meunier, & Bengio, 1998)

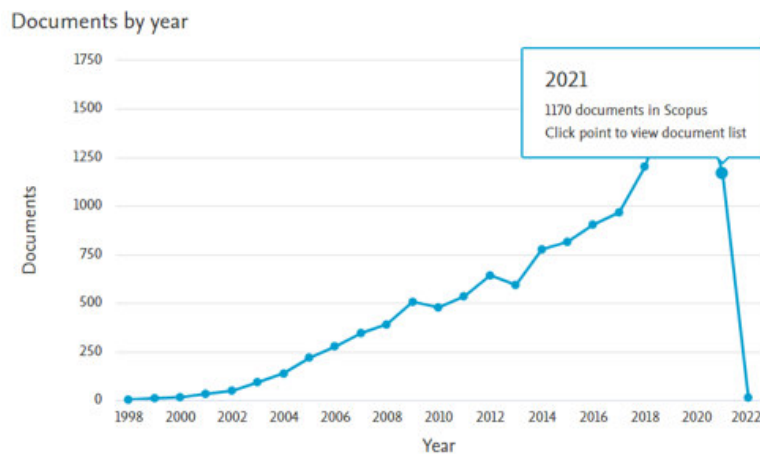


Figura 4 Tendencias de publicación de máquinas de vectores de soporte aplicadas a la clasificación en SCOPUS.

En la Figura 5 mostramos el número de estudios relacionados con la SVM relacionados con la clasificación comprendidos entre los años 2004-2021.

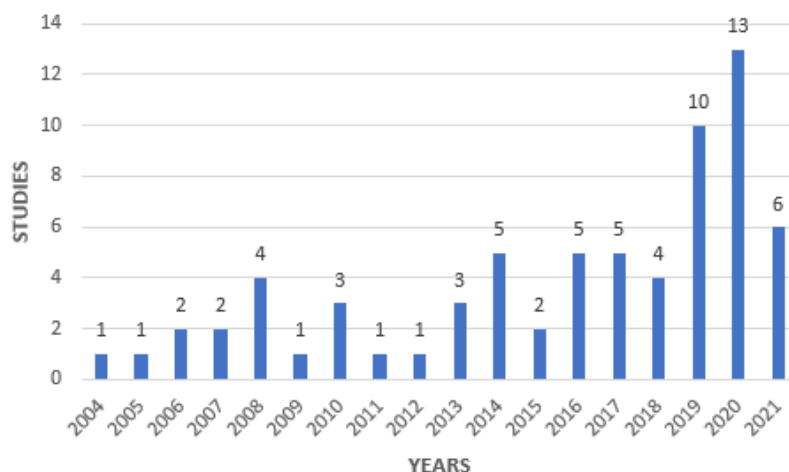


Figura 5 Estudios de evolución en máquinas de vectores de soporte.

En la Tabla 4 se muestran los estudios encontrados en la revisión sistemática de la literatura, según la selección del marco propuesto y las bases de datos utilizadas para esta investigación.

Tabla 4
Distribución por Publicación y Fuente

Fuente	Publicaciones
Scopus	(Singh, Gupta, & Gupta, 2010), (Gotsopoulos, y otros, 2018), (Guo, y otros, 2020), (Wang, Yan, Wang, & Wang, 2017), (Wu, Yu, & Chen, 2017), (Liao, Huang, & Huang, 2019), (Xu, y otros, 2019), (Wu & Pham, 2020), (Ergen & Kozat, 2020), (Razaque, Ben Haj Frej, Almiani, Alotaibi, & Alotaibi, 2021), (Moosaei, Ketabchi, Razzaghi, & Tanveer, 2021), (Alomari, y otros, 2021), (Kumar & Gopal, 2008), (Maldonado, S.; López, J., 2014), (Li, Yang, Wu, & Wu, 2021), (Liang, Zhang, Xiong, Chen, & Lan, 2016), (Panchal, Gopi, & Pradeepa, 2013), (Niu, 2011), (Yang & Chen, 2006), (Yang Z., 2013), (Zhou, Zhu, & Fan, 2012), (Bairagi, 2019), (Li, y otros, 2017), (Zhu, Zhang, & Qiao, 2020), (Cheng, y otros, 2020), (Sun, Xie, & Dong, 2019), (Shao, Chen, Zhang, Wang, & Deng, 2014), (Hua & Li, 2014), (Cheng, y otros, 2020)
IEE Xplore	(Gayathri, y otros, 2020), (Wang, Zhang, Mei, Lu, & Chen, 2021), (Ahsan, Stoyanov, Bailey, & Albarbar, 2020), (Longjam & Kisku, 2018-January), (Baygin, Baygin, & Karakose, 2019), (Almobarak, Almohammadi, Aboalnaser, & Syed, October-2019), (Kivinen, Smola, & Williamson, 2004), (Dufrenois & Noyer, 2015), (Zhu, Huang, Zhang, & Ban,

	2015), (Saravanan & Ramachandran, 2009), (Soman, Loganathan, Vijaya, Ajay , & Shivsubramani, 2007), (Yang & Chen, 2006), (Drosou & Koukouvinos, 2017), (Chen, y otros, 2019), (Hou, Liu, Zhen, & Jing, 2018)
ACM	(Papa, López, & Cano, 2020), (Hsieh, Chang, Lin, Keerthi, & Sundararajan, 2008), (Keerthi & DeCoste, 2005), (Teo, Smola, Vishwanathan, & Le, 2007), (Franc & Sonnenburg, 2008), (Mangasarian, 2006), (Ye, Liu , & Li, 2008)
Web of Science	[(Shi, Yu, & Wang, 2018), (Maldonado, López, Jimenez-Molina, & Lira, 2020), (Tikka S. , y otros, 2020), (Masoudi-Sobhanzadeh, Motieghader, Omid, & Masoudi-Nejad, 2021), (Tan, Zhen, Deng , & Zhang, 2014), (Chen & Wu, 2016)
Springer Link	(Kalidindi, Gottumukkala, & Davuluri, 2020), (Carrasco, López, & Maldonado, 2019), (López, J.; Maldonado, S.; Carrasco, S., 2011), (Carrasco, M.; López, J.; Maldonado, S., 2016), (Malik & Chack, 2019), (Wang, Phillips, Liu, & Du, 2017)
Otros	(Azarmdel, Jahanbakhshi, Mohtasebi, & Muñoz, 2020), (Sandeep & Sagar, 2019), (Alvarez, López, & Ramírez , 2010), (Couellan & Trafalis, 2013), (Ni & Zhai, 2016), (Li, y otros, 2017)

3.3. Análisis del Estudio

De 73 investigaciones seleccionadas, responderemos las preguntas de investigación (Q1, -Q2, Q3 y Q4) de acuerdo con el siguiente marco:



Figura 6 Marco general propuesto para la revisión literaria.

Las áreas anteriores fueron seleccionadas porque respondieron a las preguntas de la subinvestigación. Cabe mencionar que existen otras áreas pero se ajustan a las subpreguntas formuladas bajo la metodología de investigación.

Las aplicaciones consideradas en esta investigación fueron las que mostraron mayor relevancia de ellas según las áreas elegidas. Luego, encontramos el porcentaje por áreas del total de artículos seleccionados.

3.3.1 Áreas de Investigación

La siguiente tabla está relacionada con la primera pregunta de investigación. La Tabla 5 muestra las subpreguntas de Q1.

Tabla 5
Preguntas relacionadas con Áreas de Investigación

Contribuciones	Preguntas de investigación
Algoritmos relacionados con la inteligencia artificial, (Azarmdel, Jahanbakhshi, Mohtasebi, & Muñoz, 2020).	Q1.1 ¿Qué aportes se encuentran en la Inteligencia Artificial en Ciencias de la Computación?
Técnicas de clasificación de plantas según su forma (Singh, Gupta, & Gupta, 2010)	Q1.2. ¿Qué aportes se encuentran en la Ingeniería biomédica?
Aplicaciones al enfoque de clasificación útil en estudios de imágenes de resonancia magnética funcional, (Gotsopoulos, y otros, 2018)	Q1.3. ¿Qué aportes se encuentran en la ingeniería de las neurociencias?
El algoritmo SVM se utiliza para clasificar la célula tumoral, (Gayathri, y otros, 2020)	Q1.4. ¿Qué aportes tienen las aplicaciones interdisciplinarias de la informática?

Modelo híbrido para mejorar el rendimiento del Sistema de gestión de la energía, (Guo, y otros, 2020)	Q1.5. ¿Qué aportes se encuentran en Sistemas de Información en Ciencias de la Computación?
Selección de la variable de entrada para modelos basados en datos para la medición del caudal (Wang, Yan, Wang, & Wang, 2017)	Q1.6. ¿Qué aportes se encuentran en Matemática Computacional Biológica?
Estudio sobre la amplitud de inversión de las ondas solitarias internas aplicadas al mar poco profundo en el laboratorio, (Wang, Zhang, Mei, Lu, & Chen, 2021)	Q1.7. ¿Qué aportes se encuentran en la teledetección?
Cualificación de piezas electrónicas que puede permitir “pruebas inteligentes” utilizando técnicas de modelado basadas en datos, (Ahsan, Stoyanov, Bailey, & Albarbar, 2020)	Q1.8. ¿Qué aportes se encuentran en la ingeniería electrónica eléctrica?
Aplicaciones de clasificación de imágenes en el reconocimiento de caracteres, basadas en el análisis de componentes principales (PCA), (Wu, Yu, & Chen, 2017)	Q1.9. ¿Qué contribuciones se encuentran en los métodos de teoría de la informática?
Presentar un Sistema de reconocimiento de caracteres Brahmi utilizando un clasificador de máquina lineal de vectores de soporte, (Sandeep & Sagar, 2019)	Q1.10. ¿Qué aportes se encuentran en Imágenes Ciencia Tecnología Fotográfica?
Para clasificar satélites se aplican cuatro algoritmos de aprendizaje automático, es decir, el k-vecino más cercano (kNN), las máquinas vectoriales de soporte (SVM), las redes neuronales convolucionales (CNN) y la red Capsula (CapsNET) (Liao, Huang, & Huang, 2019)	Q1.11. ¿Qué aportes se encuentran en las aplicaciones interdisciplinarias de la informática?
La combinación de máquinas de vectores de apoyo y bosques aleatorios para la clasificación de la cobertura de la tierra rural basada en objetos mediante el uso de imágenes de alta resolución espacial, (Xu, y otros, 2019)	Q1.12. ¿Qué aportes se encuentran en la Teledetección?
Propone un método de alerta de tifones basado en el análisis de imágenes de teledetección utilizando máquinas de vectores de soporte mejoradas (SVM), (Shi, Yu, & Wang, 2018)	Q1.13. ¿Qué aportes se encuentran en la Teledetección?

3.3.2 Aplicaciones relacionadas con la clasificación

La siguiente tabla está relacionada con la primera pregunta de investigación. La Tabla 6 muestra las subpreguntas de la Q2.

Tabla 6
Preguntas relacionadas con las aplicaciones a la clasificación

Aplicaciones	Preguntas de Investigación
Se presenta un Sistema experto para analizar la carga de trabajo mental de interactuar con un teléfono móvil mientras se enfrentan las tareas diarias comunes, (Maldonado, López, Jimenez-Molina, & Lira, 2020)	Q2.1. ¿Qué aplicaciones están relacionadas con la clasificación en los sistemas de información informática?
Propone un enfoque de aprendizaje supervisado para verificar a las personas utilizando sus	Q2.2. ¿Qué aplicaciones están relacionadas con la clasificación en los sistemas de información

firmas manuscritas fuera de línea, (Longjam & Kisku, 2018-January)	informática?
Presentó un novedoso algoritmo de optimización: la genética híbrida algoritmo y algoritmo de búsqueda gravitacional (GA-GSA), (Kalidindi, Gottumukkala, & Davuluri, 2020)	Q2.3. ¿Qué aplicaciones están relacionadas con la clasificación en los sistemas de información informática?
Propuso una optimización mínima secuencial de dos fases para reducir en gran medida el costo de capacitación en big data, (Wu & Pham, 2020)	Q2.4. ¿Qué aplicaciones están relacionadas con la clasificación en los sistemas de información informática?
Estudio de la detección de anomalías en una red distribuida de nodos e introdujo un algoritmo novedoso basado en máquinas de vectores de soporte, (Ergen & Kozat, 2020)	Q2.5. ¿Qué aplicaciones están relacionadas con la clasificación de la ingeniería electrónica eléctrica?
Los datos del sensor obtenidos del movimiento del robot SCITOS G5 a lo largo de la carretera se clasifican, (Baygin, Baygin, & Karakose, 2019)	Q2.6. ¿Qué aplicaciones están relacionadas con la clasificación de la Informática de la Inteligencia Artificial?
La aplicación del aprendizaje automático supervisado (sML) de inteligencia artificial aplicado al diagnóstico de esquizofrenia, (Tikka S. , y otros, 2020)	Q2.7. ¿Qué aplicaciones están relacionadas con la clasificación de la Informática de la Inteligencia Artificial?
Clasificación de las características de las imágenes estéticas, que son: contraste, ruido, desenfoque/enfoque, composición de la imagen, etc. Utilizando Support Vector machine y K-vecino más cercano (kNN), (Almoharak, Almohammadi, Aboalnaser, & Syed, October-2019)	Q2.8. ¿Qué aplicaciones están relacionadas con la clasificación en Imágenes Ciencia Tecnología Fotográfica?
La clasificación del uso del suelo basada en la función de base radial mejorada habilitada por SVM y SVM-Linear para detección de imágenes, (Razaque, Ben Haj Frej, Almiani, Alotaibi, & Alotaibi, 2021)	Q2.9. ¿Qué aplicaciones están relacionadas con la clasificación en Teledetección?

3.3.3 Métodos de optimización o algoritmos utilizados en SVM

La siguiente tabla está relacionada con la tercera pregunta de investigación. La Tabla 7 muestra las subpreguntas de Q3.

Tabla 7

Preguntas de Investigación relacionadas con los Métodos o Algoritmos de Optimización utilizados en las SVM

Métodos de algoritmos de optimización	Preguntas de investigación
La regresión de vectores de soporte epsilon no paralelos (ϵ -NPSVR) que generaliza la regresión de vectores de soporte ϵ -twin the ϵ -TSVR, este método de clasificación binaria se amplía para predecir salidas numéricas, (Carrasco, López, & Maldonado, 2019)	Q3.1. ¿Qué métodos de optimización o algoritmos se utilizan en las aplicaciones interdisciplinarias de la informática?
Rango de enfoque máximo para máquinas vectoriales de soporte multiclase basadas en	Q3.2. ¿Qué métodos o algoritmos de optimización se utilizan en Ingeniería Biomédica?

programación de cono de segundo orden, el método propuesto consiste en un modelo de optimización único para construir todas las funciones de clasificación, (López, J.; Maldonado, S.; Carrasco, S., 2011)	
El uso de técnicas de optimización robustas para construir dos hiperplanos no paralelos con un rendimiento superior y atractivas propiedades geométricas, (Carrasco, M.; López, J.; Maldonado, S., 2016)	Q3.3. ¿Qué métodos o algoritmos de optimización se utilizan en la ingeniería de las neurociencias?
Un algoritmo proximal inexacto con distancia generalizada para resolver problemas de optimización convexa simétrica con estructuras separables, (Papa, López, & Cano, 2020)	Q3.4. ¿Qué métodos o algoritmos de optimización se utilizan en los sistemas de información informática?
Un método de descenso de coordenadas dual para SVM lineal con funciones de pérdida L1 y L2, (Hsieh, Chang, Lin, Keerthi, & Sundararajan, 2008)	Q3.5. ¿Qué métodos o algoritmos de optimización se utilizan en los sistemas de información informática?
SVM lineales resueltas con función de pérdida L2 que es adecuada para tareas de minería de datos a gran escala, como la clasificación de texto. Esto se hace modificando el método finito de Newton de Mangasarian de varias formas, (Keerthi & DeCoste, 2005)	Q3.6. ¿Qué métodos o algoritmos de optimización se utilizan en los sistemas de información informática?
Teniendo en cuenta el descenso de gradiente estocástico clásico dentro de un espacio de características y el uso de algunos trucos sencillo, muestra la convergencia de la hipótesis al minimizador de la función de riesgo regularizado, (Kivinen, Smola, & Williamson, 2004)	Q3.7. ¿Qué métodos de optimización o algoritmos se utilizan en los métodos de teoría de la informática?
Haga una variante eficiente del método de Bundle y proporcione tasas de convergencia, permita la localidad de datos y pueda manejar regularizaciones como penalizaciones L1 y L2, (Teo, Smola, Vishwanathan, & Le, 2007)	Q3.8. ¿Qué métodos o algoritmos de optimización se utilizan en Inteligencia Artificial en Ciencias de la Computación?
Un método de selección de características de envoltorio-envoltorio de dos pasos basado en el algoritmo genético (GA) y el algoritmo de optimización de los concursos competitivos mundiales (WCC). Esta propuesta se aplicó a los trece conjuntos de datos biológicos con diferentes propiedades, (Masoudi-Sobhanzadeh, Motieghader, Omidi, & Masoudi-Nejad, 2021)	Q3.9. ¿Qué métodos o algoritmos de optimización se utilizan en Ingeniería Biomédica?
Un nuevo algoritmo de entrenamiento de máquina de vector de soporte lineal (SVM) llamado algoritmo de plano de corte optimizado para SVM (OCAS). Su esfuerzo computacional escala linealmente con el tamaño de la muestra, (Franc & Sonnenburg, 2008)	Q3.10. ¿Qué métodos o algoritmos de optimización se utilizan en Inteligencia Artificial en Ciencias de la Computación?
Un algoritmo de tipo proximal interior inexacto para resolver programas de conos convexos de Segundo orden. Este tipo de problema consiste en minimizar una función convexa (posiblemente no suave) sobre la intersección de un espacio lineal afín con el producto cartesiano de conos de segundo orden, (Alvarez, López, & Ramírez, 2010)	Q3.11. ¿Qué métodos o algoritmos de optimización se utilizan en Inteligencia Artificial en Ciencias de la Computación?
Un nuevo algoritmo para entrenar SVM de manera eficiente. El problema de entrenamiento se expresa como un problema general de optimización no	Q3.12. ¿Qué métodos o algoritmos de optimización se utilizan en los sistemas de información informática?

lineal con propiedades especiales de descomposición. La técnica de gradiente incremental se utiliza y aplica a métodos de puntos interiores, (Couellan & Trafalis, 2013)	
Una reformulación de minimización convexa diferenciable no restringida de un programa lineal más general que se aplica a problemas de clasificación y aproximación de la norma. La formulación se resuelve mediante el algoritmo de Newton generalizado, (Mangasarian, 2006)	Q3.13. ¿Qué métodos de optimización o algoritmo se utilizan en los métodos de teoría de la informática?
Un algoritmo de suavizado sin matrices para resolver máquinas de vectores de soporte a gran escala. Es un algoritmo híbrido que combina las ventajas de los métodos iterativos de primer orden para resolver sistemas lineales de ecuaciones preconditionadas y la información de segundo orden basada en la técnica de suavizado, (Ni & Zhai, 2016)	Q3.14. ¿Qué métodos de optimización o algoritmos se utilizan en los métodos de teoría de la informática?
Dos enfoques de máquinas vectoriales de soporte gemelos (TWSVM). El primer enfoque es reformular la formulación TWSVM introduciendo normas L_1 y L_∞ en las funciones objetivo y convirtiéndolas en problemas de programación lineal. El segundo es resolver el TWSVM primordial y convertirlo en un problema de minimización sin restricciones, que se resuelven mediante el método de Newton generalizado, (Moosaei, Ketabchi, Razzaghi, & Tanveer, 2021)	Q3.15. ¿Qué métodos o algoritmos de optimización se utilizan en las aplicaciones interdisciplinarias de la informática?
La propuesta de un nuevo enfoque de envoltura de filtro híbrido que utiliza una relevancia máxima de redundancia mínima sólida (rMRMR) como un enfoque de filtro para elegir los genes mejor clasificados. Optimizador de lobo gris modificado (MGWO) se utiliza como un enfoque de envoltura para buscar más pequeños conjunto de genes, (Alomari, y otros, 2021)	Q3.16. ¿Qué métodos o algoritmo de optimización se utilizan en las aplicaciones interdisciplinarias de la informática?
Uso de técnicas de suavizado para suavizar la SVM gemela para la clasificación binaria. Intenta resolver los problemas primarios de programación cuadrática de SVM gemelas convirtiéndolos en problemas de minimización suaves reformulaciones. Estas suaves reformulaciones se resuelven utilizando el conocido algoritmo de Newton-Armijo, (Kumar & Gopal, 2008)	Q3.17. ¿Qué métodos de optimización o algoritmos se utilizan en los métodos de teoría de la informática?
Se presentan dos formulaciones novedosas de programación de cono de Segundo orden (SOCP) que determinan un predictor lineal utilizando máquinas de vectores de soporte. Inspirado en la formulación de SVM de margen suave, el primer enfoque (-SCOP-SVM) propone una relajación de las restricciones cónicas a través de una variable de holgura, penalizándola en la función objetivo. La segunda formulación (r-SOCP-SVM) se basa en el principio de formulación LP-SVM, (Maldonado, S.; López, J., 2014)	Q3.18. ¿Qué métodos de optimización o algoritmos se utilizan en los métodos de teoría de la informática?

3.3.4 Métodos para construir hiperplanos proximales en SVM

La siguiente tabla está relacionada con la cuarta pregunta de investigación. En la se muestran las técnicas o aplicaciones que se realizaron según la revisión y los algoritmos o métodos que se utilizaron, para este caso solo se respondió una pregunta, P4: ¿Cuáles son los métodos relacionados con los hiperplanos proximales en SVM para la clasificación

Tabla 8
Métodos relacionados con hiperplanos proximales en SVM para clasificación

Técnica o aplicación	Métodos
Genera un par de hiperplanos no paralelos mediante la introducción de la norma ℓ_0 en PSVM, luego la formulación resultante se resuelve mediante un algoritmo novedoso de diferencia de funciones convexas, (Li, Yang, Wu, & Wu, 2021)	Algoritmo de máquina de vector de soporte proximal (PSVM).
Busque un plano cerca de la población objetivo o dominante y lo más lejos posible de los valores atípicos, (Dufrenois & Noyer, 2015)	Máquinas de vector de soporte proximal de valor propio generalizado (GEPSVM).
Aplicación de un algoritmo de clasificación PSVM a los datos sísmicos, que lo ilustran para diferenciar la piedra caliza de la lutita en un juego de gas de Barnett Shale, (Zhao, Jayaram, Marfurt, & Zhou, 2014)	Máquina de vectores de soporte proximal (PSVM).
Incorpora información de geometría local dentro de cada clase de introducción a GEPSVM mediante la técnica de regularización, (Liang, Zhang, Xiong, Chen, & Lan, 2016)	Manifold Regularized Proximal support vector machine via generalized eigenvalue (GEPSVM).
Presenta un marco inteligente para clasificar las fallas en las líneas de transmisión, (Malik & Chack, 2019)	La combinación de la descomposición en modo empírico de Ensemble (EEMD) y la máquina de vectores de soporte proximal (PSVM).
Propone un enfoque novedoso para aumentar la precisión de clasificación del sistema de identificación de categoría de té. La propuesta extrae la información de color y la información de textura. Luego, se aprovechó el análisis de componentes principales para reducir las características. Las características reducidas se utilizaron como entrada para GEPSVM, (Wang, Phillips, Liu, & Du, 2017)	Máquina de vectores de soporte proximal de valor propio generalizado (GEPSVM).
Propone una novedosa metodología de aprendizaje semisupervisada que puede realizar no solo la clasificación sino también la selección de características de forma automática. Para ello, se introduce la regularización de Laplace sobre la información de los ejemplos sin etiquetar y una norma ajustable para controlar la escasez y la selección de características, (Tan, Zhen, Deng, & Zhang, 2014)	Máquina de vectores de soporte proximal p-norm laplaciano (Lap-PPSVM).
Los problemas de descomposición de valores propios generalizados son reemplazados por problemas de descomposición de valores propios estándar, lo que conduce a problemas de optimización más simples en el contexto de la	Máquina de vector de soporte proximal de valor propio generalizado mejorada (IGEPSVM)

clasificación binaria y de clases múltiples. Introdujo un parámetro adicional, que puede ajustar el rendimiento del modelo y mejorar la precisión de clasificación en GEPSVM, (Chen & Wu, 2016)	
Describe un nuevo PSVM incremental para incorporar nueva información de clase de objetivo no disponible en el sistema de clasificación de objetivos submarinos. Esta propuesta es capaz de actualizar las ponderaciones incorporando nuevos datos de la clase vista anteriormente o información de la nueva clase previamente definida, (Panchal, Gopi , & Pradeepa, 2013)	Máquina de vectores de soporte de clases múltiples (MPSVM)
Un nuevo modelo de clasificación binaria que introduce la función de pérdida ε –insensible en lugar de la función de pérdida cuadrática en PSVM. El parámetro ε controla la escasez y se puede elegir de forma flexible, mejorando la PSVM simple de muchas formas. También proporcionaron una opción útil de $\varepsilon - y\varepsilon +$ para diferentes clases en el algoritmo ε –PSVM ponderado. En comparaciones computacionales entre ε –PSVM y otros métodos. Incluidos SVC y PSVM, en varios conjuntos de datos, mostró una mayor precisión en la clasificación. Esta propuesta se aplica al reconocimiento de vehículos, (Zhu, Huang, Zhang , & Ban, 2015)	ε –máquina de vectores de soporte proximal (ε –PSVM)
Describe un algoritmo de máquina de vectores de soporte proximal para problemas de multclasificación mediante el esquema de uno contra todos. Proporcionaron una implementación paralela del algoritmo para manejar conjuntos de datos a gran escala y acelerar el proceso de entrenamiento. Lograron extender el algoritmo y la técnica de posprocesamiento para el caso del kernel no lineal, (Niu, 2011)	Máquina de vectores de soporte proximal de clases múltiples (MPSVM)
Presenta una técnica novedosa que generaliza el Clasificador de valor propio generalizado regularizado (ReGEC) a problemas de clasificación multiclase. Este método se basa en consideraciones estadísticas y geométricas, lo que proporciona una base sólida para la extensión propuesta. La contribución tiene las siguientes características: i) ser fácil de implementar, ii) fácil de paralelizar, y iii) proporcionar una precisión de clasificación media en veinte conjuntos de datos mejor que otros métodos de clasificación de última generación, (Guarracino, Irpino, & Verde, 2010)	Máquinas vectoriales de soporte proximal de valor propio generalizado (MGEPSVM).
Se trata de la aplicación de una máquina vectorial de soporte proximal multiclase rápido de un solo disparo para el diagnóstico de fallas de una caja de engranajes que consta de veinticuatro clases. Los vectores de características estadísticas de los coeficientes de ondas de Morlet se clasifican utilizando el algoritmo J48, (Saravanan & Ramachandran, 2009)	Algoritmo J48 y máquina vectorial de soporte proximal multiclase (MPSVM).
Presenta un trabajo relacionado con la calificación crediticia. Propone el uso de técnicas de aprendizaje automático para predecir calificaciones corporativas y demuestra empíricamente que los algoritmos de aprendizaje automático multiclase superan a los modelos econométricos tradicionales en calificación exacta, de 1 nivel o de 2 niveles.	Máquina vectorial de soporte proximal multiclase (MPSVM).

Muestra que con la heteroscedasticidad adecuada y los ajustes multiclase, el PSVM multiclase computacionalmente económico se puede utilizar para crear sistemas automatizados de calificación crediticia corporativa válidos en el mercado actual, (Ye, Liu , & Li, 2008)	
Muestra que las máquinas vectoriales de soporte proximal multiclase simplifican el cálculo y arrojan algo de luz sobre la geometría de la formulación multiclase dada por Szedmak y Shawe-Taylor. La nueva formulación requiere sólo un solucionador de ecuaciones lineales, (Soman, Loganathan, Vijaya, Ajay , & Shivsubramani, 2007)	Máquinas vectoriales de soporte proximal multiclase (MPSVM).
Propone un método novedoso, la SVM proximal basada en hiperplanos prototípicos de clasificación múltiple (MHPSVM), que puede obtener directamente hiperplanos prototípicos múltiples para la clasificación de clases múltiples. Los resultados experimentales sobre conjuntos de datos de referencia muestran que el rendimiento de clasificación de MHPSVM puede ser significativamente superior al de GEPSVM, (Yang & Chen, 2006)	Hiperplanos de clasificación múltiple (MHPSVM).
Propone dos clasificadores proximales de hiperplanos no paralelos de Laplacia (LapNPPCs) para problemas de clasificación semi-supervisados y completamente supervisados, respectivamente, agregando múltiples términos de regularización. Para problemas de clasificación semi-supervisada, SLapNPPC puede aprovechar la información de los datos sin etiquetar debido al tiempo de regularización múltiple adicional. Sobre la base de estas formulaciones, se pueden obtener algunas otras formulaciones laplacianas, (Yang Z. , 2013)	Clasificadores proximales de hiperplano no paralelo laplaciano (LapNPPCs)
Basado en las ideas del clasificador de hiperplano lineal con información métrica a priori (LIAM) y la máquina de vector de soporte en U, propone una nueva máquina de vector de soporte proximal semi-supervisada mejorada, denotada por NPLIAM, NPLIAM da como resultado un problema cuadrático sin restricciones para el cual se soluciona resolviendo un sistema simple de ecuaciones lineales, (Zhou, Zhu, & Fan, 2012) Propone una versión mejorada de la máquina vectorial de soporte proximal (PSVM) para clasificación binaria. En esta propuesta, el vector de peso se modifica en función del algoritmo de mínimos cuadrados medios (LSM), que reduce el error de clasificación de entrenamiento, (Bairagi, 2019)	Nueva máquina de vectores de soporte proximal semi-supervisada (PSVM). Máquina de vectores de soporte proximal mejorada (PSVM).
Se lleva a cabo un estudio extenso de cuatro conjuntos de datos médicos utilizando el modelo de máquina de vector de soporte proximal (PSVM). Además, para hacer frente a la naturaleza desequilibrada de los conjuntos de datos médicos, aplicamos la máquina de vectores de soporte de dos costos y el PSVM modificado. Ambos enfoques incorporan diferentes ponderaciones, uno para cada clase de ejemplos, (Drosou & Koukouvins, 2017)	Máquina de vectores de soporte proximal (PSVM).
Propone un algoritmo de agrupamiento en cluster difuso c-means offline (FCM) y el enfoque de reconocimiento en línea de la máquina de vector de	Máquina de vector de soporte proximal ponderado aproximado (WPSVM).

soporte proximal aproximado (WPSVM) para reducir el tamaño de la característica y mejorar la velocidad de clasificación de las características eléctricas en la nave espacial, (Li, y otros, 2017)	
Mejora el método más rápido PSVM (Proximal SVM) y propone un nuevo tipo de algoritmo PSVM ponderado por clase. Este tipo de algoritmo no solo tiene una velocidad de clasificación rápido, sino que también mejora la precisión de la clasificación, (Zhu, Zhang, & Qiao, 2020)	Máquina de vectores de soporte proximal ponderado (WPSVM).
Propone un método GEPSVM multivista mejorado (IMvGEPSVM) que agrega una regularización multivista que puede conectar diferentes vistas de la misma clase y simultáneamente considera la maximización de las muestras de diferentes clases en vistas heterogéneas para promover discriminaciones sea más eficaz, (Cheng, y otros, 2020)	Máquina de vectores de soporte proximal de valor propio generalizado de vista múltiple mejorada (IMvGEPSVM).
Propone GEPSVMs de múltiples vistas (MvGSVMs) que combinan dos vistas mediante la introducción de un término de coregularización de múltiples vistas para maximizar el consenso sobre distintas vistas y transformar hábilmente un problema de optimización complicado en un problema de valor propio generalizado simple. También propone GEPSVMs mejorados multivista (MvIGSVMs), (Sun, Xie, & Dong, 2019)	Máquinas vectoriales de soporte proximal de valor propio generalizado multivista (MvGEPSVMs).
Propone una máquina de vector de soporte robusto proximal no paralelo regularizada con lazo de traza novedosa (RNPSVM) para clasificación ruidosa. El riesgo empírico es implementado por la norma L1, y para capturar la escasez y la correlación subyacente de los datos, se introduce un lazo de seguimiento para regularizar la RNPSVM. Luego, se diseña un algoritmo iterativo para resolver el problema de optimización resultante, (Chen, y otros, 2019)	Máquina robusta de vectores de soporte proximal no paralelo (RNPSVM).
Propone un clasificador no paralelo novedoso denominado máquina de vectores de soporte no paralelo de proyección (PNPSVM). El nuevo clasificador necesita dos pasos para obtener los hiperplanos proximales óptimos. El primer paso el obtener dos direcciones de proyección y el segundo es determinar las ubicaciones específicas de los hiperplanos proximales óptimos basándose en una muestra central apropiada, (Hou, Liu, Zhen, & Jing, 2018)	Máquina de vectores de soporte de proyección no paralela (PNPSVM).
Propone una máquina de vector de soporte de gemelos de Lagrange ponderada eficiente (WLTSVM) para la clasificación de datos desequilibrados basada en el uso de diferentes puntos de entrenamiento para construir los dos hiperplanos proximales, (Shao, Chen, Zhang, Wang, & Deng, 2014)	Máquina de vectores de soporte gemelo lagrangiana ponderado (WLTSVM).
Propone una novedosa máquina de vectores de soporte gemelo (TWSVM) con regularización múltiple al introducir la idea básica de la localidad que conserva la matriz de dispersión dentro de la clase en TWSVM, (Hua & Li, 2014)	Máquinas de vectores de soporte Novel Twin con regularización múltiple (MTSVM).
Propone un nuevo método llamado Máquina de vectores de soporte generalizado de múltiples vistas a través de la extracción de la relación inherente entre vistas (MRMvGSVM). Al mismo tiempo, utiliza la matriz de relaciones de restricción de la	Máquinas de vectores de soporte generalizado de múltiples vistas mediante la extracción de la relación inherente entre vistas (MRMvGSVM).

norma L2, 1 como un término de regularización de múltiples vistas para seleccionar los datos de muestra más relevantes de diferentes vistas, (Cheng, y otros, 2020)	
---	--

3.4. Análisis de Resultados

Los porcentajes presentados en este apartado se obtienen en referencia a las preguntas Q1, Q2 y Q3 de los estudios analizados sin redundancia

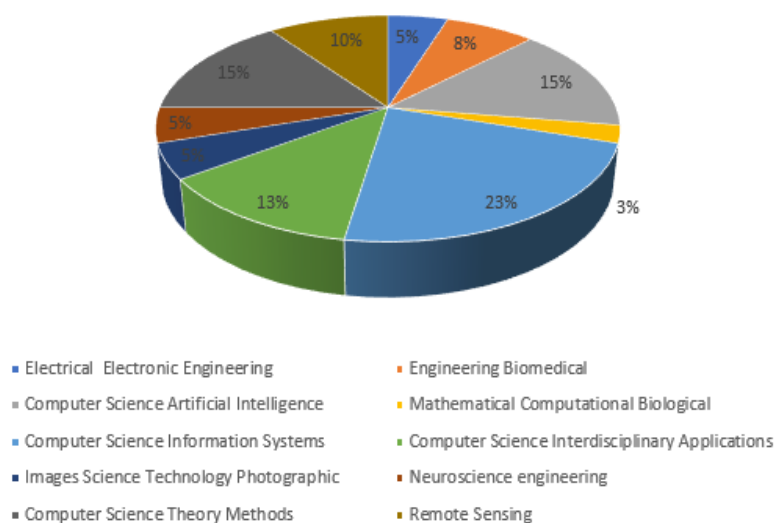


Figura 7 Áreas de Investigación.

A. Áreas de investigación (Q1)

Vemos las áreas donde se utilizan las máquinas de vectores de soporte, según investigado, se obtuvo la siguiente información. Observe en la Figura 7, el porcentaje de distribución según las áreas de investigación, principalmente vemos que 23% está relacionado con Sistemas de Información en Ciencias de la Computación, 15% Inteligencia Artificial en Ciencias de la Computación, 15% Métodos Teóricos de las Ciencias de la Computación, 13% Aplicaciones Interdisciplinarias de las Ciencias de la Computación, 10% Teledetección, 8%

Ingeniería Biomédica, 5% Ingeniería Eléctrica Electrónica, 5% Imágenes Ciencia Tecnología Fotográfica, 5% Ingeniería en Neurociencia finalmente 3% Matemática Computacional Biológica.

B. Aplicaciones relacionadas con la clasificación (Q2)

En este apartado identificamos que las aplicaciones de las máquinas de vectores de soporte relacionadas con la clasificación son: sistemas expertos, verificación de firma manuscrita, selección de banda optima de imágenes hiperespectrales, reducción del costo de prueba en big data, detección de anomalías en una red distribuida de nodos, clasificar datos de sensores obtenidos del movimiento del robot SCITOS G5, clasificación de esquizofrenia basada en inteligencia artificial, clasificación de las características de imágenes estéticas, clasificación de uso del suelo, entre otros.

C. Algoritmos de optimización utilizados en SVM (Q3)

Los algoritmos o métodos de optimización utilizados en SVM dependen del área de aplicación, entre los que se encuentran los siguientes:

Regresión de vector de soporte épsilon no paralelo ($\epsilon - NPSVR$), margen máximo para máquinas de vector de soporte de clase múltiple basadas en programación de cono de segundo orden, SVM de hiperplano no paralelo (NH-SVM), algoritmo SC-PMA (método de multiplicador proximal de cono simétrico). Un método de descenso de coordenadas duales para SVM lineal, el algoritmo de descenso de coordenadas con la selección aleatoria de una instancia a la vez, el algoritmo de Newton finito modificado (MFN), el algoritmo de plano de corte optimizado para SVM (OCAS), un algoritmo de Newton de suavizado, el filtro híbrido. El enfoque envolvente mediante el uso de una robusta Mínima Redundancia Máxima Relevancia (rMRMR), entre otros.

Mencionaremos que en el caso de los algoritmos de (SVM), tiene un problema de optimización por resolver dentro de su estructura, este modelo de optimización su característica principal es que su función objetivo a optimizar es cuadrática, por lo

tanto, es diferenciable y se encontrar la solución óptima.

La deficiencia estos algoritmos es si se tiene que resolver un modelo de clasificación y la función es no diferenciable ya no se podría resolver con os algoritmos de SVM.

Debido a que la función no es diferenciable, se tendría que trabajar con Subdiferenciales, en el caso del algoritmo SC-PMA se puede resolver con funciones que pueden o no ser diferenciales, es decir es más general.

D. Métodos relacionados con hiperplanos proximales en SVM para clasificación (Q4)

El porcentaje de métodos próximos encontrados en este estudio utilizando máquinas de vectores de soporte (SVM) se presenta en la Figura 8.

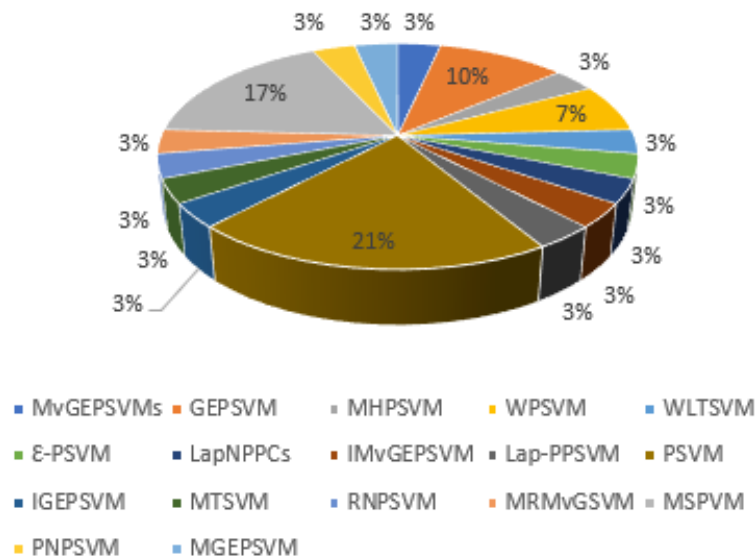


Figura 8 Los métodos con hiperplanos proximales estudiados por máquinas de vectores de soporte (SVM).

Para esta investigación, los porcentajes de los hiperplanos proximales utilizados por las máquinas de vectores de soporte relacionados con la clasificación fueron: 21% Máquinas de vectores de soporte proximal (PSVM), seguido de (MPSVM) con 17%, (GEPSVM) con 10% y Soporte proximal ponderado Vector Machine (WPSVM) con 7% respectivamente.

3.5. Conclusiones

En este trabajo, aplicamos la metodología propuesta por Kitchenham y Charter para estudiar el estado del arte en SVM aplicado a la clasificación. Encontramos aplicaciones en diferentes áreas, tales como Ingeniería Eléctrica Electrónica, Ingeniería Biomédica, Informática Artificial, Matemática Computacional Biológica, Sistemas de Información Informática, Aplicaciones Interdisciplinarias de la Informática, entre otras.

Por otro lado, también encontramos diferentes algoritmos de optimización que utiliza SVM para la clasificación tales como: Algoritmo SC-PMA (Método Simétrico Cone – Multiplicador Proximal), un método de descenso de coordenadas dual para SVM lineal, el algoritmo de descenso de coordenadas con selección aleatoria de una instancia a la vez, el algoritmo de Newton finito modificado (MFN), el algoritmo de plano de corte optimizado para SVM (OCAS), el algoritmo de tipo proximal con métrica variable (PAVM), entre otros.

Asimismo, encontramos los métodos más utilizados en (SVM) para clasificación: el (PSVM) con un 21%, (MPSVM) con un 17% y (GEPSVM) con un 10% respectivamente.

En una revisión sistemática previa, los autores de (Cano & Papa, 2020) revisaron 108 306 artículos en diferentes bases de datos como Scopus, WOS, Science Direct, IEEE Xplore, ACM, entre otras. En este documento, abarcamos los artículos revisados a 118 213.

Finalmente, mencionaremos que algunos de los artículos considerados para esta investigación también pueden estar en más de una base de datos de búsqueda. En este sentido, recomendamos que futuras investigaciones agreguen un criterio para excluir esta posibilidad.

CAPÍTULO IV: UN MÉTODO DE MULTIPLICADOR PROXIMAL PARA SEPARABLES CONVEXOS OPTIMIZACIÓN DE CONO SIMÉTRICO

4.1. Introducción

Sea $\mathbb{V}_1$ y $\mathbb{V}_2$ dos espacios vectoriales de dimensión finita en el campo real $\mathfrak{R}$ dotado del producto interior $\langle \cdot, \cdot \rangle_{\mathbb{V}_1}$ y $\langle \cdot, \cdot \rangle_{\mathbb{V}_2}$ respectivamente. Entonces considere el siguiente Jordán euclideano álgebra $\mathbb{V}_1 = (\mathbb{V}_1, \circ_1, \langle \cdot, \cdot \rangle_{\mathbb{V}_1})$ y $\mathbb{V}_2 = (\mathbb{V}_2, \circ_2, \langle \cdot, \cdot \rangle_{\mathbb{V}_2})$, donde $\circ_1$ y $\circ_2$ denote el producto de Jordan en $\mathbb{V}_1$ y $\mathbb{V}_2$, respectivamente. En este trabajo, consideramos el siguiente *cono simétrico convexo problema de optimización* (SCP) con estructura separable:

$$\min\{f(x) + g(z): \mathbb{A}x + \mathbb{B}z = \mathbb{b}, x \in K_1, z \in K_2\}, \quad (4.1)$$

Donde $f: \mathbb{V}_1 \rightarrow \mathfrak{R} \cup \{+\infty\}$ y $g: \mathbb{V}_2 \rightarrow \mathfrak{R} \cup \{+\infty\}$ es cerrado función convexa, propia $\mathbb{A}: \mathbb{V}_1 \rightarrow \mathfrak{R}^m$ y $\mathbb{B}: \mathbb{V}_2 \rightarrow \mathfrak{R}^m$ son dos mapas lineales, $\mathbb{b} \in \mathfrak{R}^m$, y $K_1 := \{x \circ_1 x: x \in \mathbb{V}_1\}$ y $K_2 := \{z \circ_2 z: z \in \mathbb{V}_2\}$ denotan los conjuntos de elementos cuadrados (conos simétricos) en $\mathbb{V}_1$ y $\mathbb{V}_2$, respectivamente.

Este problema proporciona un marco unificado simple para varios problemas de optimización existentes, como la programación no lineal, de cono de segundo orden y semidefinida, y por lo tanto tiene una amplia aplicación en ingeniería, economía, teoría de juegos, ciencia administrativa y otros campos; véase y las referencias allí contenidas. En particular, cierta clase de problemas de (2) ha llamado mucho la atención recientemente debido a sus aplicaciones emergentes de interés práctico, siendo estas las máquinas de vectores de soporte y la selección de covarianza inversa dispersa.

La idea del multiplicador proximal predictos correctos (PCPM) para resolver (4.1) sin

restricciones cónicas es la siguiente:

$$\begin{aligned}
 p^{k+1} &= y^k + \lambda_k (Ax^k + Bz^k - b), \\
 x^{k+1} &= \arg \min_{x \in \mathfrak{R}^m} \left\{ f(x) + \langle p^{k+1}, Ax \rangle + \frac{1}{2\lambda_k} \|x - x^k\|^2 \right\}, \\
 z^{k+1} &= \arg \min_{z \in \mathfrak{R}^m} \left\{ g(z) + \langle p^{k+1}, Bz \rangle + \frac{1}{2\lambda_k} \|z - z^k\|^2 \right\}, \\
 y^{k+1} &= y^k + \lambda_k (Ax^k + Bz^k - b)
 \end{aligned}$$

Donde $\{\lambda_k\}_{k \in \mathbb{N}}$ es una secuencia de parámetros positivos.

Recientemente, Sarmiento *et.al* (Sarmiento, Papa Quiroz, & Oliveira, 2017) desarrollan una extensión del método (PCPM) para resolver (4.1), cuando el álgebra de Jordan euclidiana es el conocido espacio vectorial euclidiana, utilizando distancias proximales regularizadas (Auslender & Teboulle, 2006). Con el fin de considerar una amplia gama de aplicaciones, como por ejemplo problemas de optimización de conos de segundo orden y semidefinidos separables (Ver también la Sección 4), que no pueden ser cubiertos por el trabajo (Sarmiento, Papa Quiroz, & Oliveira, 2017), el propósito de esta tesis

es ampliar la (PCPM) método para resolver los problemas. En concreto, primero definimos una distancia proximal con respecto al interior del cono simétrico K . Esta distancia puede ser generada por una clase apropiada de funciones conexas estrictas continuamente diferenciables en $\mathfrak{R}_{++}$. En segundo lugar proponemos una aproximación de algoritmo de descomposición usando la distancia proximal para resolver (4.1). Entonces, bajo algunas suposiciones moderadas, establecemos la global convergencia del algoritmo propuesto.

Este capítulo está organizado de la siguiente forma: La Sección 4.2 recuerda algunas nociones básicas y revisa algunos resultados esenciales sobre el álgebra de Jordan Euclidiano. En la Sección 4.3, presentamos la clase de distancias proximales definidas en el cono simétrico. Finalmente, en la Sección 4.3, describimos el algoritmo con distancia proximal para resolver (4.1) y establecemos su convergencia global.

4.2. Preliminares

En esta subsección, describimos brevemente algunos conceptos, propiedades y resultados de las álgebras de Euclides de Jordan necesarios en este artículo.

La teoría se puede encontrar en (Faraut & Korányi, 1994)

Un *álgebra euclidiana de Jordan* es un triple $(\mathbb{V}, \circ, \langle \cdot, \cdot \rangle_V)$, donde $(\mathbb{V}, \langle \cdot, \cdot \rangle_V)$ es un espacio -de dimensión finita sobre el campo real equipado con un producto interno $\langle \cdot, \cdot \rangle_V$, y el producto de Jordan $(x, y) \mapsto x \circ y: V \times V \rightarrow V$, que es un mapeo bilineal satisfaciendo las siguientes tres condiciones:

$$x \circ y = y \circ x, \text{ para todo } x, y \in \mathbb{V},$$

$$x \circ (x^2 \circ y) = x^2 \circ (x \circ y), \text{ para todo } x, y \in \mathbb{V} \text{ donde } x^2 = x \circ x,$$

$$\langle x \circ y, z \rangle_V = \langle y, x \circ z \rangle_V, \text{ para todo } x, y, z \in \mathbb{V};$$

Y existe un elemento unitario (único) para todo $e \in \mathbb{V}$, tal que $x \circ e = x$ para todo $x \in \mathbb{V}$.

De ahora en adelante, simplemente afirmamos que $\mathbb{V}$ es un álgebra de Jordan Euclideana, y $x \circ y$ es llamado el *Producto Jordan* de x e y . Decimos que el elemento $x \in \mathbb{V}$ es *invertible*, si hay $w \in \mathbb{V}$ tal que $x \circ w = e$. El conjunto $K = \{x^2: x \in \mathbb{V}\}$ es llamado el cono de cuadrados del Álgebra de Jordan Euclideana, que es un cono simétrico

Teorema 2.1 (Teorema de descomposición espectral): *Para todo $x \in V$ existe un marco de Jordan $\{e_1(x), \dots, e_r(x)\}$ y números reales $\lambda_1(x), \dots, \lambda_r(x)$, dispuesto en orden decreciente, tal que $x = \sum_{i=1}^r \lambda_i(x) e_i(x)$.*

Los números $\lambda_i(x)$ (contando multiplicidades) son únicamente determinados por x , y se denominan *valores propios* de x . La *traza* de x , denotada como $tr(x)$, está definida como $tr(x) := \sum_{i=1}^r \lambda_i(x)$; mientras que la *determinante* está definida por $\det(x) := \prod_{i=1}^r \lambda_i(x)$.

Es fácil demostrar que para todo $x \in K$ (*resp. int(K)*) si y si $\lambda_i(x) \geq 0$ (*resp. $\lambda_i(x) > 0$*), para todo $i = 1, \dots, r$. Además, un elemento $x \in V$ es *invertible*, si $\det(x) \neq 0$.

La forma bilineal $\langle x, y \rangle := tr(x \circ y)$ es un producto interno en $\mathbb{V}$ y se llama el *producto interno canónico* (o *traza*) en $\mathbb{V}$. Además, podemos definir la norma inducida por el producto interior $\langle x, y \rangle$ sobre $\mathbb{V}$, llamado *norma de Frobenius*, por

$$\|x\|_F := \sqrt{\langle x, x \rangle} = \text{tr}(x^2) = \left(\sum_{i=1}^r \lambda_i^2(x) \right)^{\frac{1}{2}}, \forall x \in V.$$

Entonces, en este producto interior, la norma de cualquier primitivo idempotente es uno.

Por eso, $\|e\|_F := \sqrt{r}$.

Además, esta norma satisface a la desigualdad de Cauchy-Schwarz $|\langle x, y \rangle| \leq$

$\|x\|_F \|y\|_F$. Finalmente, recordamos los siguientes resultados técnicos.

Lema 2.2 (Desigualdad de Von Neumann): *Para cualquier $x, y \in V$, uno tiene*

$$\langle x, y \rangle = \text{tr}(x \circ y) \leq \sum_{i=1}^r \lambda_i(x) \lambda_i(y).$$

La igualdad se cumple si y si x tiene una descomposición, es decir, si existe un

marco de Jordan $\{e_1, \dots, e_r\}$ tal que $x = \sum_{i=1}^r \lambda_i(x) e_i$ e $y = \sum_{i=1}^r \lambda_i(y) e_i$.

Lema 2.3: *Sea $x \in V$. Entonces, $x \in k$ si y si $\langle x, y \rangle \geq 0$ vale para todo $y \in K$. Es más, $x \in \text{int}(K)$ si y si $\langle x, y \rangle > 0$ para todo $y \in K \setminus \{0\}$.*

Sea $\phi: \mathfrak{R} \rightarrow \mathfrak{R}$ sea una función escalar. Se puede definir una función con valores vectoriales $\phi^{SC}: \mathbb{V} \rightarrow \mathbb{V}$, llamado *operador de Lowner*, por $\phi^{SC}(x) := \sum_{i=1}^r \phi(\lambda_i(x)) e_i(x)$, si $\lambda_i(x) \in \text{dom}(\phi)$,

Donde $x \in V$ tiene la siguiente descomposición espectral $x = \sum_{i=1}^r \lambda_i(x) e_i(x)$, con $\lambda_1(x) \geq \lambda_2(x) \geq \dots \geq \lambda_r(x)$. Entonces podemos considerar su correspondiente *definido espectralmente* función por $\Phi: \mathbb{V} \rightarrow \mathfrak{R} \cup \{+\infty\}$ de

$$\Phi(x) := \text{tr}(\phi^{SC}(x)) = \sum_{i=1}^r \phi(\lambda_i(x)),$$

Si $\lambda_i(x) \in \text{dom}(\phi)$, y $+\infty$ de lo contrario. Tenga en cuenta que si $\phi: \mathfrak{R} \rightarrow \mathfrak{R}$ es una función continuamente diferenciable en un subconjunto de $\text{dom}(\phi)$, podemos definir la función vectorial $(\phi')^{SC}: \mathbb{V} \rightarrow \mathbb{V}$ por $(\phi')^{SC}(x) := \sum_{i=1}^r \phi'(\lambda_i(x)) e_i(x)$, si $\lambda_i(x) \in \text{dom}(\phi')$.

Los siguientes resultados bien conocidos nos dan algunas propiedades conocidas de las funciones ϕ y ϕ^{SC} .

Lema 2.4: *Si la función ϕ es convexa (respectivamente convexa) y continua en su*

dominio, entonces su función definida espectralmente ϕ es convexa (respectivamente convexa) y continua en su dominio.

Lema 2.5: Sea ϕ ser una función continuamente diferenciable en un subconjunto de $\text{dom}(\phi)$ y $x \in V$ con su descomposición espectral $x = \sum_{i=1}^r \lambda_i(x) e_i(x)$. Entonces

- (a) ϕ es continuamente diferenciable en $\text{dom}((\phi')^{SC})$ y $\nabla \phi(x) = \sum_{i=1}^r \phi'(\lambda_i(x)) e_i(x) = (\phi')^{SC}(x)$, $\forall x \in \text{dom}((\phi')^{SC})$,
- (b) ϕ^{SC} es continuamente diferenciable en $\text{dom}((\phi')^{SC})$ y $\nabla \phi^{SC}(x)h = \sum_{i=1}^r \phi'(\lambda_i(x)) \langle e_i(x), h \rangle e_i(x) + 4 \sum_{1 \leq i < j \leq r} [\lambda_i(x), \lambda_j(x)]_{\phi} e_i(x) \circ (e_j(x) \circ h)$, $\forall h \in V$, donde $[\lambda_i(x), \lambda_j(x)]_{\phi} = \frac{\phi(\lambda_i(x)) - \phi(\lambda_j(x))}{\lambda_i(x) - \lambda_j(x)}$, $\forall i, j = 1, \dots, r$ y $i \neq j$. En particular $\nabla \phi^{SC}(x)x = \sum_{i=1}^r \phi'(\lambda_i(x)) \lambda_j(x) e_i(x) = (\phi')^{SC}(x) \circ x$, $\forall x \in \text{dom}((\phi')^{SC})$.

4.3. Distancias proximales sobre conos simétricos

Sea $\mathbb{V}$ sea un espacio vectorial de dimensión finita. Una función de valor extendido $H: \mathbb{V} \times \mathbb{V} \rightarrow \mathbb{R} \cup \{+\infty\}$ se llama una distancia proximal con respecto a $\text{int}(K)$, interior de K , si cumple las siguientes propiedades:

- (P1) $\text{dom}(H(\cdot, \cdot)) = \mathcal{C}_1 \times \mathcal{C}_2$
con $\text{int}(K) \times \text{int}(K) \subseteq \mathcal{C}_1 \times \mathcal{C}_2 \subseteq K \times K$.
- (P2) $H(u, v) \geq 0$, para todo $u, v \in \mathbb{V}$, y $H(u, v) = 0$ para todo $v \in \text{int}(K)$.
- (P3) Para cada $v \in \text{int}(K)$, $H(\cdot, v)$ es continua y estrictamente convexa en $\mathcal{C}_1$, y es continuamente diferenciable en $\text{int}(K)$ con $\text{dom}(\nabla_1 H(\cdot, v)) = \text{int}(K)$, donde $\nabla_1 H(\cdot, v)$ denota el gradiente de $H(\cdot, v)$ con respecto a la primera variable.
- (P4) Para cada $\gamma \in \mathbb{R}$ el conjunto de nivel inferior $\{u \in \mathcal{C}_1: H(u, v) \leq \gamma\}$ esta

acotado para cualquier $v \in \mathcal{C}_2$.

La definición anterior fue considerada en (Pan & Chen, 2010) para una distancia proximal con respecto al interior del cono de segundo orden $\mathcal{L}_{++}^n := \text{int}(\mathcal{L}_+^n)$, donde

$$\mathcal{L}_+^n := \{x = (x_1, \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1} : \|\bar{x}\| \leq x_1\},$$

Y tiene una pequeña diferencia con la definición introducida por Auslender y Teboulle (ver (Lobo, Vandenberghe, Boyd, & Lebre, 1998), Definición 2.1), ya que aquí $H(\cdot, y)$ se requiere que sea estrictamente convexo sobre $\mathcal{C}_1$ para cualquier fijo $y \in \text{int}(K)$.

Denotemos $\mathcal{D}(\text{int}(K))$ por la familia de funciones H satisfaciendo la Definición de (P2) y (P3) se sigue que $H(\cdot, v)$ alcanza su valor mínimo global en v , lo que demuestre que $\nabla_1 H(v, v) = 0$. Desafortunadamente, la clase anterior no es suficiente para establecer la convergencia global del algoritmo propuesto, por lo tanto, usaremos algunas condiciones adicionales en la distancia proximal $H \in \mathcal{D}(\text{int}(K))$:

(B1) Para cualquier $u, v \in \text{int}(K)$ y $w \in \mathcal{C}_1$,

$$\langle \nabla_u H(u, v), w - u \rangle \leq H(w, u) - \gamma H(u, v)$$

para algunos $\gamma \in (0, 1]$.

(B1') Para algunos $u, v \in \text{int}(K)$ y $w \in \mathcal{C}_2$

$$\langle \nabla_u H(u, v), w - u \rangle \leq H(v, w) - H(u, w) - \gamma H(u, v)$$

para algunos $\gamma \in (0, 1]$.

(B2) Para cada $u \in \mathcal{C}_1$, $H(u, \cdot)$ esta acotado por el nivel $\mathcal{C}_2$.

(B3) Para cualquier $\{v^k\}_{k \in \mathbb{N}} \subseteq \text{int}(K) : v^k \rightarrow v^*$, y $c \in \mathcal{C}_1$, tenemos esto $H(c, v^k) \rightarrow H(c, v^*)$.

(B3') Para cualquier $\{v^k\}_{k \in \mathbb{N}} \subseteq \text{int}(K) : v^k \rightarrow v^*$, y $c \in \mathcal{C}_2$, tenemos esto $H(v^k, c) \rightarrow H(v^*, c)$.

(B4) Para cualquier $\{v^k\}_{k \in \mathbb{N}} \subseteq \text{int}(K)$ convergiendo a $v^* \in K$, tenemos esto $H(v^*, v^k) \rightarrow 0$.

Proposición 3.1. Dado $\phi \in \Sigma$, sea H la función definida por

$$H(x, y) := \begin{cases} \Phi(x) - \Phi(y) - \langle \nabla \Phi(y), x - y \rangle, & x \in \text{dom}(\Phi), y \in \text{int}(K) \\ +\infty, & \text{de lo contrario} \end{cases}$$

Entonces se cumplen los siguientes resultados:

(a) $H(\cdot, \cdot)$ es continua en $\text{dom}(\Phi) \times \text{int}(K)$ y, para cualquier $y \in \text{int}(K)$, la

función $H(\cdot, y)$ es estrictamente convexa en $\text{dom}(\Phi)$.

(b) Para cualquier $y \in \text{int}(K)$, $H(\cdot, y)$ es continuamente diferenciable en $\text{int}(K)$ con

$$\nabla_x H(x, y) = \nabla \Phi(x) - \nabla \Phi(y)$$

(c) $H(x, y) \geq 0$ para cualquier $x \in \text{dom}(\Phi)$ y $y \in \text{int}(K)$, y $H(x, y) = 0$ si y si $x = y$

(d) $H(x, y) \geq \sum_{i=1}^r d\phi(\lambda_i(x), \lambda_i(y)) \geq 0$, para cualquier $x \in \text{dom}(\Phi)$ y $y \in \text{int}(K)$.

(e) Para todo $\gamma \geq 0$, el nivel establece $L_H(y, \gamma) = \{x \in \text{dom}(\Phi): H(x, y) \leq \gamma\}$ y $L_H(x, \gamma) = \{y \in \text{int}(K): H(x, y) \leq \gamma\}$ están acotados, para cualquier $y \in \text{int}(K)$ y $x \in \text{dom}(\Phi)$, respectivamente.

(f) Para cualquier $x, y \in \text{int}(K)$ y $z \in \text{dom}(\Phi)$, se cumple la siguiente identidad de tres puntos:

$$\langle \nabla_x H(x, y), z - x \rangle = H(z, y) - H(z, x) - H(x, y).$$

(g) Si $\{y_k\} \subset \text{int}(K)$ es tal que $\lim_{k \rightarrow +\infty} y_k = \bar{y} \in \text{bd}(K)$, entonces

$$\lim_{k \rightarrow +\infty} \langle \nabla \Phi(y_k), x - y_k \rangle = -\infty, \quad \forall x \in \text{int}(K)$$

En el documento (López & Papa Quiroz, 2017) se presentaron ejemplos de distintos proximales para conos simétricos.

Daremos los siguientes ejemplos de aplicación de distancias proximales.

Ejemplo 3.1. (Distancia proximal similar a la entropía)

Sean $\phi(t) = t \ln(t)$, si $t \geq 0$ (con la convención $0 \ln(0) = 0$) y $\phi(t) = +\infty$, si $t < 0$. Es fácil comprobar que $\phi \in \Sigma$ con $\text{dom}(\phi) = \mathbb{R}_+$. Su función asociada definida espectralmente es $\Phi(x) = \text{tr}(x \circ \ln(x))$, si $x \in K$, $\nabla \Phi(x) = \ln(x) + e$, para $x \in \text{int}(K)$. Entonces, la función $H: \mathbb{V} \times \mathbb{V} \rightarrow (-\infty, +\infty]$ está dada por

$$H(x, y) = \begin{cases} \text{tr}(x \circ \ln(x) - x \circ \ln(y) + y - x), & \forall x \in K, y \in \text{int}(K) \\ +\infty, & \text{lo contrario} \end{cases}$$

De la proposición 3.2. se sigue que H es una distancia proximal y que satisface

(B1) y (B2) con $\mathcal{C}_1 = K$ y $\mathcal{C}_2 = \text{int}(K)$.

(a) Note que cuando $\mathbb{V} = \mathbb{R}^n$, $K = \mathbb{R}_+^n$, la función H tiene la forma

$$H(x, y) = \begin{cases} \sum_{i=1}^n \left(x_i \ln \left(\frac{x_i}{y_i} \right) + y_i - x_i \right) & , \forall x \in \mathbb{R}_+^n, y \in \mathbb{R}_{++}^n, \\ +\infty & , \text{de lo contrario} \end{cases}$$

Que es llamada distancia de entropía relativa de Kullback – Leibler (Liese & Vajda, 1987) (Teboulle, 1997) .Esta distancia satisface la condición (B4), luego satisface (B3).

(b) Para $\mathbb{V} = \mathbb{R}^n, K = \mathcal{L}_+^n$, la función H tiene la forma

$$H(x, y) = \begin{cases} \text{tr}(x \circ \ln(x) - x \circ \ln(y) + y - x) & , \forall x \in \mathcal{L}_+^n, y \in \mathcal{L}_{++}^n \\ +\infty & , \text{de lo contrario} \end{cases}$$

A diferencia del caso anterior, esta función no cumple la propiedad (B4). Este hecho se ilustra en (Pan, S.; Chen, J.S., 2010)

(c) Para $\mathbb{V} = \mathcal{S}^n, K = \mathcal{S}_+^n$, la función H tiene la forma

$$H(X, Y) = \begin{cases} \text{tr}(X \ln(X - Y) + Y - Y) & , \forall X \in \mathcal{S}_+^n, Y \in \mathcal{S}_{++}^n \\ +\infty & , \text{de lo contrario} \end{cases}$$

Como se observa en un contraejemplo dado por Doljansky y Teboulle (Doljansky & Teboulle, 1998) , la propiedad (B4) no se cumple para esta función.

Ejemplo 3.2.

Para $p \in (0,1)$, consideremos la familia de funciones $\phi(t) = \frac{(pt-t^p)}{(1-p)}$ si $t \geq 0$, y $\phi(t) = +\infty$ si, $t < 0$ [29]. No es difícil verificar que $\phi \in \Sigma$ con $\text{dom}(\phi) = \mathbb{R}_+$.

Entonces, su función asociada definida espectralmente es $\Phi(x) = \frac{1}{1-p} \text{tr}(px - x^p)$ para $x \in K$. Por el Lemma 2.5 se sigue que $\nabla \Phi(x) = \frac{p}{1-p}(e - x^{p-1})$, para $x \in \text{int}(K)$. Así, tenemos que

$$H(x, y) = \begin{cases} \frac{1}{1-p} \text{tr}((1-p)y^p + (py^{p-1} - x^{p-1}) \circ x) & , \forall x \in K, y \in \text{int}(K) \\ +\infty & , \text{de lo contrario} \end{cases}$$

De la proposición 3.2 obtenemos que H es una distancia proximal que satisface (B1) – (B2) con $\mathcal{C}_1 = K$ y $\mathcal{C}_2 = \text{int}(K)$

Tenga en cuenta que cuando $\mathbb{V} = \mathbb{R}^n$ y $K = \mathbb{R}_+^n$, la distancia proximal H toma la forma

$$H(x, y) = \begin{cases} \sum_{i=1}^n (y_i^{p-1} (y_i + qx_i) - (1+q)x_i^p), & \forall x \in \mathbb{R}_+^n, \quad y \in \mathbb{R}_{++}^n \\ \infty & , \quad \text{de lo contrario} \end{cases}$$

Donde $q = \frac{p}{1-p}$, y satisface la condición (B4) (Auslender, A.; Teboulle, M., 2006)

Ejemplo 3.3. (Distancia Proximal de la barrera de troncos)

Sea $\phi(t) = -\ln(t)$, para $t > 0$. Es fácil verificar que $\phi \in \Sigma$ con $\text{dom}(\phi) = \mathbb{R}_{++}$, y que su función asociada definida espectralmente es $\Phi(x) = -\text{tr}(\ln(x))$, para $x \in \text{int}(K)$. Por el Lema 2.5 se sigue que $\nabla \Phi(x) = -x^{-1}$, para $x \in \text{int}(K)$. Entonces uno tiene

$$H(x, y) = \begin{cases} \text{tr}(\ln(y) - \ln(x) + y^{-1} \circ x) - r & , \forall x, y \in \text{int}(K) \\ +\infty & , \text{de lo contrario} \end{cases}$$

Esta función extiende la distancia proximal de la barrera logarítmica dada en [2,30,26] nuestro contexto. Además, usando la Proposición 3.2 se tiene que H es una distancia proximal que satisface (B1) – (B2) con $\mathcal{C}_1 = \mathcal{C}_2 = \text{int}(K)$

4.4. Algoritmo inexacto con distancia proximal generalizada

Proporcionamos un ejemplo del problema (4.1) en Máquinas de Vectores de Soporte (SVM) para clasificación binaria. Para otro ejemplo, consulte nuestro apéndice a continuación.

Dado un conjunto de instancias con sus respectivas variables (x_i, y_i) , donde $x_i \in \mathbb{R}^n$, $i = 1, \dots, m$ y $y_i \in \{-1, +1\}$, la SVM determina un hiperplano +óptimo, a saber

$$H(w, b) = \{x \in \mathbb{R}^n: w^T x + b = 0\}$$

Que separa los cascos convexos reducidos de ambos patrones de entrenamiento. Este hiperplano óptimo se puede encontrar resolviendo el siguiente problema de programación cuadrática:

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \quad (4.2)$$

$$\text{s. t. } y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \dots, m,$$

Donde $\xi = (\xi_1, \dots, \xi_m) \in \mathfrak{R}^m$ y $C > 0$ es un parámetro de penalización. Denotemos para $z = (w, b) \in \mathfrak{R}^{n+1}$ y $Diag(y)$ una diagonal mixta, cuya diagonal principal es el vector $y = (y_1, \dots, y_m)$, de $\mathbf{1}$ un vector de $\mathfrak{R}^m$, y de

$$\hat{X} = \begin{pmatrix} x_1^T & 1 \\ \vdots & \vdots \\ x_m^T & 1 \end{pmatrix} \in \mathfrak{R}^{m \times n+1}$$

Entonces, la primera restricción de desigualdad se puede escribir como $Diag(y)\hat{X}z + \xi - \mathbf{1} \geq 0$. Introduciendo las variables $u = Diag(y)\hat{X}z + \xi - \mathbf{1}$ y $v = (\xi, u)$, el problema (4.2) puede ser reescrito como

$$\min_{z,v} \{f(z) + g(v) : \mathbb{A}z + \mathbb{B}v = \mathbb{b}, v \geq \mathbb{0}\}, \quad (4.3)$$

Donde $f(z) = \frac{1}{2} \|w\|^2$, $g(v) = C\mathbf{1}^T \xi$, $\mathbb{A} = Diag(y)\hat{X} \in \mathfrak{R}^{m \times n+1}$, $\mathbb{B} = (I - I) \in \mathfrak{R}^{m \times 2m}$ y $\mathbb{b} = \mathbf{1}$. Por lo tanto, este modelo tiene la forma del problema (4.2).

4.4.1. Algoritmo del multiplicador proximal

Para cada $i = 1, 2$, dejar $H_i: \mathbb{V}_i \times \mathbb{V}_i \rightarrow \mathfrak{R} \cup \{+\infty\}$ ser una distancia proximal, es decir, $H_i \in \mathcal{D}(int(K_i))$ y $\theta_i > 0$ un parámetro fijo.

Definimos

$$H_{\theta_1}(x_1, x_2) := H_1(x_1, x_2) + \frac{\theta_1}{2} \|x_1 - x_2\|^2, \quad (4.4)$$

$$H_{\theta_2}(z_1, z_2) := H_2(z_1, z_2) + \frac{\theta_2}{2} \|z_1 - z_2\|^2 \quad (4.5)$$

Es fácil demostrar que para cada $i = 1, 2$, H_{θ_i} , es también una distancia proximal con respecto a $int(K_i)$, es decir, $H_{\theta_i} \in \mathcal{D}(int(K_i))$.

Algoritmo SC-PMA

Dejar $H_i \in \mathcal{D}(\text{int}(K_i))$, $\theta_i > 0$, $i = 1, 2$, $\text{tol} > 0$ y $\{\varepsilon_k\}, \{\zeta_k\}, \{\lambda_k\}$ son secuencias positivas escalares.

Paso 0: Comience con algún punto inicial $w^0 = (x^0, z^0, y^0) \in \text{int}(K_1) \times \text{int}(K_2) \times \mathfrak{R}^m$. Colocar $k = 0$.

Paso 1: Calcular

$$p^{k+1} = y^k + \lambda_k(\mathbb{A}x^k + \mathbb{B}z^k - b), \quad (4.6)$$

Paso 2: Encontrar $(x^{k+1}, z^{k+1}) \in \text{int}(K_1) \times \text{int}(K_2)$ y $(g_1^{k+1}, g_2^{k+1}) \in \mathbb{V}_1 \times \mathbb{V}_2$, tal que

$$\begin{aligned} g_1^{k+1} &\in \partial_{\varepsilon_k} f(x^{k+1}) \\ g_1^{k+1} + \mathbb{A}^* p^{k+1} + \frac{1}{\lambda_k} \nabla_x H_{\theta_1}(x^{k+1}, x^k) &= 0, \end{aligned} \quad (4.7)$$

$$\begin{aligned} g_2^{k+1} &\in \partial_{\zeta_k} g(z^{k+1}) \\ g_2^{k+1} + \mathbb{B}^* p^{k+1} + \frac{1}{\lambda_k} \nabla_z H_{\theta_2}(z^{k+1}, z^k) &= 0, \end{aligned} \quad (4.8)$$

Paso 3: Calcular

$$y^{k+1} = y^k + \lambda_k(\mathbb{A}x^{k+1} + \mathbb{B}z^{k+1} - b). \quad (4.9)$$

Paso 4: dejar $w^{k+1} = (x^{k+1}, z^{k+1}, y^{k+1})$. Si $\|w^{k+1} - w^k\| \leq \text{tol}$, se detiene, de lo contrario, reemplace k por $k + 1$ y vaya al Paso 1.

Presentamos a continuación el Diagrama de Flujo del Algoritmo.

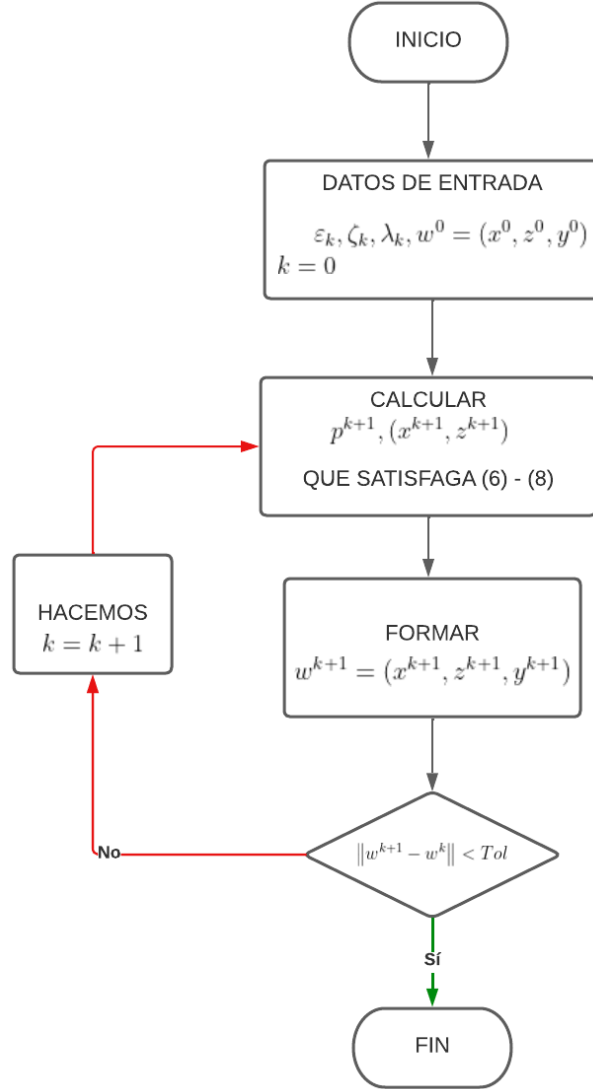


Figura 9 Diagrama de Flujo del Algoritmo SC - PMA

El siguiente Lema es un resultado clásico en análisis convexo adaptado a nuestro trabajo.

Lema 4.1: Dejar $F: \mathbb{V} \rightarrow \mathbb{R} \cup \{+\infty\}$ sea una función convexa propia cerrada, $H \in \mathcal{D}(\text{int}(K))$ satisface (B1) o (B1') y $\lambda_k > 0$. Si $\{(u^k, g_1^k)\}_{k \in \mathbb{N}}$ y $\{(v^k, g_2^k)\}_{k \in \mathbb{N}}$ son dos secuencias que satisfacen

$$\begin{aligned}
 g_1^{k+1} &\in \partial_{\varepsilon_k} F(x^{k+1}) \\
 g_1^{k+1} + \lambda_k^{-1} \nabla_x H(u^{k+1}, u^k) &= 0, \\
 \tilde{g}_2^{k+1} &\in \partial_{\zeta_k} F(v^{k+1})
 \end{aligned}$$

$$\tilde{g}_2^{k+1} + \lambda_k^{-1}(v^{k+1} - v^k) = 0,$$

Entonces para cualquier $k \geq 0$, obtenemos

$$\lambda_k(F(u^{k+1}) - F(u)) \leq H(u, u^k) + H(u, u^{k+1}) + \lambda_k \varepsilon_k, \forall u \in \mathcal{C}_1, \text{ si (B1) se}$$

cumple;

$$\lambda_k(F(u^{k+1}) - F(u)) \leq H(u^k, u) - H(u^{k+1}, u) + \lambda_k \varepsilon_k, \forall u \in \mathcal{C}_2, \text{ si (B1')} \text{ se}$$

cumple;

Y finalmente,

$$\begin{aligned} 2\lambda_k(F(v^{k+1}) - F(v)) \\ \leq \|v^k - v\|^2 - \|v^{k+1} - v\|^2 - \|v^{k+1} - v^k\|^2 + 2\lambda_k \zeta_k, \forall v \\ \in \mathfrak{R}^m. \end{aligned}$$

4.4.2. Análisis de convergencia del algoritmo SC-PMA

En esta sección, bajo supuestos apropiados, probaremos que las secuencias generadas por el algoritmo SC-PMA están bien definidas y estableceremos la convergencia global del algoritmo propuesto.

De ahora en adelante, supongamos que las siguientes suposiciones son ciertas:

- (A1) El problema (P) admite al menos una solución óptima (x^*, z^*) .
- (A2) $\exists x \in \text{int}(k_1) \cap \text{ri}(\text{dom}(f)), z \in \text{int}(k_2) \cap \text{ri}(\text{dom}(g))$ tal que $\mathbb{A}x + \mathbb{B}z = \mathbb{b}$.
- (A3) Las secuencias $\{\zeta_k\}_{k \in \mathbb{N}}$ y $\{\varepsilon_k\}_{k \in \mathbb{N}}$ son no negativas, y $\sum_{k=0}^{\infty} (\zeta_k + \varepsilon_k) < \infty$.

Observación 4.2. El supuesto (A2) asegura la existencia de un multiplicador de Lagrange dual óptimo y^* . Por lo tanto, bajo los supuestos (A1) – (A2), (x^*, z^*, y^*) es un punto de silla de L , es decir,

$$\begin{aligned} L(x^*, z^*, y^*) \leq L(x^*, z^*, y^*) \leq L(x, z, y^*), \forall x \in k_1 \cap \text{dom}(f), z \\ \in k_2 \cap \text{dom}(g), y \in \mathfrak{R}^m, \end{aligned}$$

Donde

$$L(x, z, y) = f(x) + g(z) + \langle y, \mathbb{A}x + \mathbb{B}z - \mathbb{b} \rangle, x \in k_1, z \in k_2, y \in \mathfrak{R}^m.$$

Denota la función Lagrangiana para el problema (1).

La suposición (A3) es una condición natural cuando trabajamos con iteraciones inexactas.

El siguiente resultado muestra que el algoritmo SC – PMA está bien definido.

Proposición 4.1. Bajo los supuestos (A1), (A2) $H_i \in \mathcal{D}(\text{int}(k_i)), i = 1, 2$, y $(x^k, z^k, y^k) \in \text{int}(k_1) \times \text{int}(k_2) \times \mathfrak{R}^m$, existe un único punto $(x^{k+1}, z^{k+1}) \in \text{int}(k_1) \times \text{int}(k_2)$ satisfaciendo (4.7) – (4.8).

Prueba. Colocar

$$F_k(x) = f(x) + \langle \mathbb{A}^* p^{k+1}, x \rangle + \frac{1}{\lambda_k} H_{\theta_1}(x, x^k) + \delta_{k_1}(x), \text{ luego de supuesto (A2)}$$

tenemos que

$$\text{dom}(F_k) = \text{dom}(f) \cap \text{dom}\left(H_{\theta_1}(\cdot, x^k)\right) \cap k_1 \neq \emptyset. \text{ Definir } S_k = \arg \min_{x \in k_1} \{F_k(x)\},$$

Probaremos que el conjunto S_k no es empatía. De hecho, debido a que

$$f(\cdot), \langle \mathbb{A}^* p^{k+1}, \cdot \rangle \text{ y } \left(\frac{1}{\lambda_k}\right) H_{\theta_1}(\cdot, x^k) \text{ son convexos propios cerrados.}$$

Funciones, entonces F_k es propiamente cerrado y convexo también, por lo tanto, es suficiente para mostrar que denoten $(F_k)_\infty(d) > 0, \forall d \neq 0$, donde $(F_k)_\infty$ denota la función de recesión de F_k .

Dejar $d \neq 0$. Entonces, tenemos que

$$(F_k)_\infty(d) = f_\infty(d) + (\langle \mathbb{A}^* p^{k+1}, \cdot \rangle)_\infty(d) + \frac{1}{\lambda_k} (H_{\theta_1})_\infty(d, x^k) + (\delta_{k_1})_\infty(d).$$

Es fácil probar que $f_\infty(d) > 0, (\delta_{k_1})_\infty(d) \geq 0, (\langle \mathbb{A}^* p^{k+1}, \cdot \rangle)_\infty(d) =$

$$\langle \mathbb{A}^* p^{k+1}, \cdot \rangle, \text{ entonces haremos eso } (H_{\theta_1})_\infty(d, x^k) = +\infty.$$

Como $H_{\theta_1}(\cdot, z)$ es propiamente cerrado y convexo, uno tiene $(H_{\theta_1})_\infty(d, x^k) =$

$$(H_{\theta_1})_\infty(d, x^k) + \frac{\theta_1}{2} (\|\cdot - x^k\|^2)_\infty(d).$$

Debido a que H_1 es coercitivo $(H_1)_\infty(d, x^k) > 0$ y como $(\|\cdot - x^k\|^2)_\infty(d) = +\infty$ obtenemos que $(H_{\theta_1})_\infty(d, x^k) = +\infty$ y luego también obtenemos

$$(F_k)_\infty(d) = +\infty > 0, \forall d \neq 0$$

Por lo tanto, existe $\bar{x} := x^{k+1} \in \mathbb{V}_1$ tal que

$$0 \in \partial \left(f(\cdot) + \langle \mathbb{A}^* p^{k+1}, \cdot \rangle + \frac{1}{\lambda_k} (H_{\theta_1})_{\infty}(\cdot, x^k) + \delta_{k_1}(\cdot) \right) (x^{k+1}).$$

Asimismo, la estricta convexidad de F_k garantiza la unicidad de x^{k+1} .

Ahora por la suposición A2), tenemos que

$$0 \in \partial f(x^{k+1}) + \mathbb{A}^* p^{k+1} + \left(\frac{1}{\lambda_k} \right) \partial_x H_{\theta_1}(\cdot, x^k)(x^{k+1}) + \partial \delta_{k_1}(x^{k+1}).$$

Por definición tenemos que $\text{dom}(\partial_x H_{\theta_1}(\cdot, y)) = \text{int}(k_1)$, entonces $x^{k+1} \in \text{int}(k_1)$, por eso $\partial \delta_{k_1}(x^{k+1}) = \mathcal{N}_{k_1}(x^{k+1}) = \{0\}$.

Así, existe $g_1^{k+1} \in \partial f(x^{k+1})$ tal que (7) sostiene con $\varepsilon_k = 0$. Dado que $\partial f(x^{k+1}) \subseteq \partial_{\varepsilon_k} f(x^{k+1})$, (7) se cumple para $\varepsilon_k > 0$.

La existencia y unicidad de z^{k+1} y (8) se muestran en una forma similar.

El siguiente resultado establece estimaciones para las secuencias generada por el algoritmo (SC - PMA).

Proposición 4.2. Sea $\{x^k, z^k, y^k, p^k\}$ la secuencia generada del algoritmo (SC - PMA), (x^*, z^*) sea una solución óptima de (1) y y^* sea un multiplicador de Lagrange correspondiente. Suponga que $H_i \in \mathcal{D}(\text{int}(k_i))$, $i = 1, 2$, y (B1) o (B1') se cumple. Entonces, lo siguiente las desigualdades se cumplen para todo $k \geq 0$

$$\begin{aligned} & H_{\theta_1}(x^*, x^{k+1}) + H_{\theta_2}(z^*, z^{k+1}) \leq H_{\theta_1}(x^*, x^k) + H_{\theta_2}(z^*, z^k) - \\ & \gamma_1 H_{\theta_1}(x^{k+1}, x^k) - \gamma_2 H_{\theta_2}(z^{k+1}, z^k) - \lambda_k \langle p^{k+1} - y^*, \mathbb{A}x^{k+1} + \mathbb{B}z^{k+1} - b \rangle + \\ & \lambda_k (\zeta_k, \varepsilon_k), \text{ si (B1) se cumple; (4.10)} \\ & H_{\theta_1}(x^{k+1}, x^*) + H_{\theta_2}(z^{k+1}, z^*) \leq H_{\theta_1}(x^k, x^*) + H_{\theta_2}(z^k, z^*) - \\ & \gamma_1 H_{\theta_1}(x^{k+1}, x^k) - \gamma_2 H_{\theta_2}(z^{k+1}, z^k) - \lambda_k \langle p^{k+1} - y^*, \mathbb{A}x^{k+1} + \mathbb{B}z^{k+1} - b \rangle + \\ & \lambda_k (\zeta_k + \varepsilon_k), \text{ si (B1') se cumple.} \end{aligned} \quad (4.11)$$

Prueba. Suponga que (B1) se cumple. Usando el Lema 4.1. Parte (i) con $F(\cdot) := f(\cdot) + \langle p^{k+1}, \mathbb{A} \cdot \rangle$ y con $F(\cdot) := g(\cdot) + \langle p^{k+1}, \mathbb{B} \cdot \rangle$ tenemos que, para todo $x \in \mathcal{C}_1$:

$$\begin{aligned} & \lambda_k (f(x^{k+1}) + \langle p^{k+1}, \mathbb{A}x^{k+1} \rangle - f(x) - \langle p^{k+1}, \mathbb{A}x \rangle) \\ & \leq H_{\theta_1}(x, x^k) - H_{\theta_1}(x, x^{k+1}) - \gamma_1 H_{\theta_1}(x^{k+1}, x^k) + \lambda_k \varepsilon_k, \end{aligned}$$

Y para todo $z \in \mathcal{C}_1$:

$$\begin{aligned} & \lambda_k(g(z^{k+1}) + \langle p^{k+1}, \mathbb{B}z^{k+1} \rangle - g(z) - \langle p^{k+1}, \mathbb{B}z \rangle) \\ & \leq H_{\theta_2}(z, z^k) - H_{\theta_2}(z, z^{k+1}) - \gamma_2 H_{\theta_2}(z^{k+1}, z^k) + \lambda_k \zeta_k. \end{aligned}$$

Sumando las dos desigualdades anteriores, obtenemos

$$\begin{aligned} & \lambda_k(L(x^{k+1}, z^{k+1}, p^{k+1}) - L(x, z, o^{k+1})) \\ & \leq H_{\theta_1}(x, x^k) - H_{\theta_1}(x, x^{k+1}) - \gamma_1 H_{\theta_1}(x^{k+1}, x^k) \\ & \quad + H_{\theta_2}(z, z^k) - H_{\theta_2}(z, z^{k+1}) - \gamma_2 H_{\theta_2}(z^{k+1}, z^k) \\ & \quad + \lambda_k(\zeta_k + \varepsilon_k) \end{aligned} \tag{4.12}$$

Por otra parte, desde (x^*, z^*, y^*) es el punto de silla de L que obtenemos

$$\lambda_k(L(x^*, z^*, p^{k+1}) - L(x^{k+1}, z^{k+1}, y^*)) \leq 0.$$

Usando (4.12) con $x = x^*, z = z^*$, y sumando con la desigualdad anterior obtenemos (4.10) después de reordenar términos. La desigualdad (4.11) se sigue usando los argumentos anteriores y la propiedad (B1').

Proposición 4.3. Sea $\{x^k, z^k, y^k, p^k\}_{k \in \mathbb{N}}$ la secuencia generada por el algoritmo (SC - PMA), (x^k, z^k) sea una solución óptima de (4.1) y y^* sea un multiplicador de Lagrange correspondiente. Entonces, las siguientes desigualdades se cumplen para todo $k \geq 0$.

$$\begin{aligned} & \lambda_k \langle \mathbb{A}x^{k+1} + \mathbb{B}z^{k+1} - b, y^* - y^{k+1} \rangle \\ & \leq \frac{1}{2} (\|y^k - y^*\|^2 - \|y^{k+1} - y^*\|^2 - \|y^{k+1} - y^k\|^2), \end{aligned} \tag{4.13}$$

$$\begin{aligned} & \lambda_k \langle \mathbb{A}x^{k+1} + \mathbb{B}z^{k+1} - b, y^* - p^{k+1} \rangle \\ & \leq \frac{1}{2} (\|y^k - y^{k+1}\|^2 - \|p^{k+1} - y^{k+1}\|^2 - \|p^{k+1} - y^k\|^2) \end{aligned} \tag{4.14}$$

Prueba. Las desigualdades (4.13) – (4.14) se derivan directamente de [3, Proposición 2] o [8, Lema 4.1].

Para cualquier vector $w_1 = (x_1, z_1, y_1) \in \mathcal{C}_1 \times \mathcal{C}_1 \times \mathfrak{R}^m$ y $w_2 = (x_2, z_2, y_2) \in \mathcal{C}_2 \times \mathcal{C}_2 \times \mathfrak{R}^m$ definimos $\hat{H}_\theta(w_1, w_2) = H_{\theta_1}(x_1, x_2) + H_{\theta_2}(z_1, z_2) + \frac{1}{2} \|y_1 - y_2\|^2$. (4.15)

Lema 4.2. Sea $\{x^k, z^k, y^k, p^k\}_{k \in \mathbb{N}}$ la secuencia generada por el algoritmo (SC - PMA) y sea $w^* = (x^*, z^*, y^*)$ con (x^*, z^*) una solución óptima de (4.1) y y^* su correspondiente multiplicados de Lagrange. Asumir que $H_i \in \mathcal{D}(\text{int}(k_i))$, para $i = 1, 2$. Si (B1) se cumple, entonces

$$\begin{aligned} \widehat{H}_\theta(w^*, w^{k+1}) &\leq \widehat{H}_\theta(w^*, w^k) - \frac{1}{2}(\theta_1 - 4\lambda_k^2 \|\mathbb{A}\|^2) \|x^{k+1} - x^k\|^2 \\ &\quad - \frac{1}{2}(\theta_2 - 4\lambda_k^2 \|\mathbb{B}\|^2) \|z^{k+1} - z^k\|^2 \\ &\quad - \frac{1}{2} \|p^{k+1} - y^{k+1}\|^2 - \frac{1}{2} \|p^{k+1} - y^k\|^2 \\ &\quad + \lambda_k(\zeta_k + \varepsilon_k); \end{aligned} \tag{4.16}$$

Y si (B1') se cumple, entonces

$$\begin{aligned} \widehat{H}_\theta(w^{k+1}, w^*) &\leq \widehat{H}_\theta(w^k, w^*) - \frac{1}{2}(\theta_1 - 4\lambda_k^2 \|\mathbb{A}\|^2) \|x^{k+1} - x^k\|^2 \\ &\quad - \frac{1}{2}(\theta_2 - 4\lambda_k^2 \|\mathbb{B}\|^2) \|z^{k+1} - z^k\|^2 \\ &\quad - \frac{1}{2} \|p^{k+1} - y^{k+1}\|^2 - \frac{1}{2} \|p^{k+1} - y^k\|^2 \\ &\quad + \lambda_k(\zeta_k + \varepsilon_k); \end{aligned} \tag{4.17}$$

Prueba. Suponer que (B1) sostiene sumando las desigualdades (4.13) – (4.14), obtenemos

$$\begin{aligned} &\frac{1}{2} \|y^{k+1} - y^*\|^2 \\ &\leq \frac{1}{2} \|y^k - y^*\|^2 - \frac{1}{2} (\|p^{k+1} - y^{k+1}\|^2 + \|p^{k+1} - y^k\|^2) \\ &\quad - \lambda_k \langle \mathbb{A}x^{k+1} + \mathbb{B}z^{k+1} - b, y^{k+1} - p^{k+1} \rangle \\ &\quad - \lambda_k \langle \mathbb{A}x^{k+1} + \mathbb{B}z^k - b, y^* - y^{k+1} \rangle \end{aligned}$$

Dejar $w^{k+1} = (x^{k+1}, z^{k+1}, y^{k+1})$ y $w^k = (x^k, z^k, y^k)$.

Luego, sumando (4.10) con la desigualdad anterior obtenemos

$$\begin{aligned}
\hat{H}_\theta(w^*, w^{k+1}) &\leq \hat{H}_\theta(w^*, w^k) - \frac{\gamma_1 \theta_1}{2} \|x^{k+1} - x^k\|^2 \\
&\quad - \frac{\gamma_2 \theta_2}{2} \|z^{k+1} - z^k\|^2 \\
&\quad - \frac{1}{2} (\|p^{k+1} - y^{k+1}\|^2 - \|p^{k+1} - y^k\|^2) + \rho_k \\
&\quad + \lambda_k (\zeta_k + \varepsilon_k);
\end{aligned} \tag{4.18}$$

Donde $\rho_k = \lambda_k \langle y^{k+1} - p^{k+1}, \mathbb{A}(x^{k+1} - x^k) + \mathbb{B}(z^{k+1} - z^k) \rangle$

Ahora, usando (4.6) y (4.9) se sigue que

$$\begin{aligned}
\rho_k &= \lambda_k^2 \|\mathbb{A}(x^{k+1} - x^k) + \mathbb{B}(z^{k+1} - z^k)\|^2 \\
&\leq 2\lambda_k^2 (\|\mathbb{A}\|^2 \|x^{k+1} - x^k\|^2 + \|\mathbb{B}\|^2 \|z^{k+1} - z^k\|^2).
\end{aligned}$$

Por lo tanto, al usar esta desigualdad en (4.18) se sigue el resultado. La desigualdad (4.17) se sigue usando los mismos argumentos anteriores y la propiedad (B1').

Teorema 4.1. Sea $\{x^k, z^k, y^k, p^k\}_{k \in \mathbb{N}}$ la secuencia generada por el algoritmo (SC - PMA) y sea $w^* = (x^*, z^*, y^*)$ con (x^*, z^*) una solución óptima de (1) y y^* su correspondiente multiplicador de Lagrange. Asumir que $H_i \in \mathcal{D}(\text{int}(k_i))$, para $i = 1, 2$, y (B1) – (B3) o (B1'), (B3') se cumple. Si $\{\lambda_k\}$ satisface

$$\begin{aligned}
\lambda_k &\geq \eta, \quad \lambda_k \|\mathbb{A}\| \leq \frac{1}{2} (\gamma_1 \theta_1 - \vartheta)^{\frac{1}{2}}, \\
\lambda_k \|\mathbb{B}\| &\leq \frac{1}{2} (\theta_2 - \vartheta)^{\frac{1}{2}}, \forall k \geq 0,
\end{aligned} \tag{4.19}$$

Para algunos $\eta > 0$ y $0 < \vartheta < \min\{\theta_1, \theta_2\}$, entonces lo siguiente se cumple:

- (i) La secuencia $w^k = (x^k, z^k, y^k)$ esta acotado y cada punto limite de w^k es un punto de silla del Lagrangiano.
- (ii) Además, si (B4) – (B5) o (B4') – (B5') se cumplen, entonces la secuencia $\{(x^k, z^k, y^k)\}$ globalmente converge a una solución del problema (4.3).

Prueba. (i) Suponga que (B1) – (B2) se cumplen. Ya que λ_k satisface (4.19), de (4.16) tenemos que

$$\begin{aligned}
\hat{H}_\theta(w^*, w^{k+1}) &\leq \hat{H}_\theta(w^*, w^k) - \frac{\vartheta}{2}(\|x^{k+1} - x^k\|^2 - \|z^{k+1} - z^k\|^2) \\
&\quad - \frac{1}{2}(\|p^{k+1} - y^{k+1}\|^2 - \|p^{k+1} - y^k\|^2) \\
&\quad + \lambda_k(\zeta_k + \varepsilon_k)
\end{aligned} \tag{4.20}$$

Esta implica que $\{w^k\}_{k \in \mathbb{N}} \subseteq \{w \in \text{int}(k) \times \mathfrak{R}^p \times \mathfrak{R}^m : \hat{H}_\theta(w^*, w) \leq \bar{\alpha}\}$, con

$$\bar{\alpha} = \hat{H}_\theta(w^*, w^k) + \sum_{k=0}^{\infty} \lambda_k(\zeta_k + \varepsilon_k)$$

Por suposición (B2), y el hecho de que

$$\sum_{k=0}^{\infty} \lambda_k(\zeta_k + \varepsilon_k) < \infty$$

(cf. Suposición (A3) y (4.19)), se sigue que la secuencia $\{w^k\}_{k \in \mathbb{N}}$ está acotada.

Además, (4.20) junto con $\hat{H}_\theta(w^*, w^{k+1}) \geq 0$ implica que existe $l(w^k) \geq 0$ tal que

$$\lim_{k \rightarrow \infty} \hat{H}_\theta(w^*, w^k) = l(w^*) \tag{4.21}$$

Por lo tanto, tomando los límites a ambos lados de (4.20), obtenemos

$$\begin{aligned}
\|x^{k+1} - x^k\| &\rightarrow 0, \|z^{k+1} - z^k\| \rightarrow 0, \|p^{k+1} - y^{k+1}\| \\
&\rightarrow 0, \|p^{k+1} - y^k\| \rightarrow 0.
\end{aligned} \tag{4.22}$$

Por otra parte, dado que la $(w^k)_{k \in \mathbb{N}}$ está acotado, existe una subsecuencia $\{w^{kj} = (x^{kj}, z^{kj}, y^{kj})\}_{k \in \mathbb{N}}$ y un punto límite $w^\infty = (x^\infty, z^\infty, y^\infty)$ tal que $w^{kj} \rightarrow w^\infty$. Ahora procederemos a mostrar que w^∞ es un punto silla de L . Primero, ya que $\lambda_k \geq \eta$, pasando el límite el (4.12) sobre la subsucesión, y usando (B3) y (22), obtenemos

$$L(x^\infty, z^\infty, y^\infty) \leq L(x, z, y^\infty), \forall x \in \mathcal{C}_1, \forall z \in \mathcal{C}_2 \tag{4.23}$$

Segundo, aplicando el Lema 4.1, Parte (iii) (en su forma exacta) con $F(\cdot) := -L(x^{k+1}, z^{k+1}, \cdot)$, tenemos

$$\begin{aligned}
\lambda_k \left(L(x^{k+1}, z^{k+1}, y) - L(x^{k+1}, z^{k+1}, y^{k+1}) \right) &\leq \frac{1}{2}(\|y^k - y\|^2 \|y^{k+1} - y\|^2), \forall y \\
&\in \mathfrak{R}^m
\end{aligned}$$

Tomando el límite de la subsecuencia en la desigualdad anterior y usando (4.22), tenemos

$$L(x^\infty, z^\infty, y) \leq L(x^\infty, z^\infty, y^\infty), \forall y \in \mathfrak{R}^m \quad (4.24)$$

De la forma (4.24) se sigue que $Ax^\infty + Bz^\infty = b$. Finalmente, dado que $\{x^k\}_{k \in \mathbb{N}} \subset \text{int}(k_1)$ y $\{z^k\}_{k \in \mathbb{N}} \subset \text{int}(k_2)$, pasando el límite uno tiene que $(x^\infty, z^\infty) \in k_1 \times k_2$. Por lo tanto, de (4.23) – (4.24) el resultado sigue. En el caso de que (B1'), (B3) se cumpla, la prueba es similar a arriba y usando (P4) de la Definición 3.1 en lugar de (B2).

(ii) Suponga que (B4) – (B5) se cumpla. Sea w^∞ el límite de una subsecuencia $\{w^{kj}\}_{j \in \mathbb{N}}$ de $\{w^k\}_{k \in \mathbb{N}}$, de eso, $w^{kj} \rightarrow w^\infty$. Entonces, por (B4) tenemos

$$\lim_{j \rightarrow \infty} \hat{H}_\theta(w^\infty, w^{kj}) = 0 \quad (4.25)$$

Dado que w^∞ es un punto silla de L (por la Parte (i)), se sigue de (4.21) y (4.25) que $l(w^\infty) = 0$. Por lo tanto, usando (B5) obtenemos que la secuencia $\{w^k\}_{k \in \mathbb{N}}$ converge a w^∞ . Ahora, si (B4') – (B5') aguanta, resultado que sigue usando los mismos argumentos que antes.

Otro ejemplo aplicado al problema (P) podemos ver en el ANEXO C. Aplicación del Problema (P)

4.4.3. Complejidad del Algoritmo

El algoritmo presentado en esta tesis es la siguiente:

Algoritmo SC-PMA

Dejar $H_i \in \mathcal{D}(\text{int}(K_i))$, $\theta_i > 0$, $i = 1, 2$, $tol > 0$ y $\{\varepsilon_k\}, \{\zeta_k\}, \{\lambda_k\}$ son secuencias positivas escalares.

Paso 0: Comience con algún punto inicial $w^0 = (x^0, z^0, y^0) \in \text{int}(K_1) \times \text{int}(K_2) \times \mathfrak{R}^m$. Colocar $k = 0$.

Paso 1: Calcular

$$p^{k+1} = y^k + \lambda_k(\mathbb{A}x^k + \mathbb{B}z^k - b), \quad (4.26)$$

Paso 2: Encontrar $(x^{k+1}, z^{k+1}) \in \text{int}(K_1) \times \text{int}(K_2)$ y $(g_1^{k+1}, g_2^{k+1}) \in \mathbb{V}_1 \times \mathbb{V}_2$, tal que

$$\begin{aligned} g_1^{k+1} &\in \partial_{\varepsilon_k} f(x^{k+1}) \\ g_1^{k+1} + \mathbb{A}^* p^{k+1} + \frac{1}{\lambda_k} \nabla_x H_{\theta_1}(x^{k+1}, x^k) &= 0, \end{aligned} \quad (4.27)$$

$$\begin{aligned} g_2^{k+1} &\in \partial_{\zeta_k} g(z^{k+1}) \\ g_2^{k+1} + \mathbb{B}^* p^{k+1} + \frac{1}{\lambda_k} \nabla_z H_{\theta_2}(z^{k+1}, z^k) &= 0, \end{aligned} \quad (4.28)$$

Paso 3: Calcular

$$y^{k+1} = y^k + \lambda_k(\mathbb{A}x^{k+1} + \mathbb{B}z^{k+1} - b). \quad (4.29)$$

Paso 4: dejar $w^{k+1} = (x^{k+1}, z^{k+1}, y^{k+1})$. Si $\|w^{k+1} - w^k\| \leq \text{tol}$, se detiene, de lo contrario, reemplace k por $k + 1$ y vaya al Paso 1.

Analizaremos la complejidad en cada paso:

- En el paso cero, dado que es el ingreso de datos no repercute en la complejidad.
- En el paso uno, consideremos las matrices de orden $A \in R^{m \times n}$, $B \in R^{m \times p}$, $X \in R^{n \times 1}$, $Z \in R^{p \times 1}$

Vamos a considerar las operaciones aritméticas básicas como la adición, diferencia, división y multiplicación como una unidad de tiempo $O(1)$.

Las operaciones que se realizan son:

$$\begin{aligned} &m\{(2n - 1) + (2p - 1) + 1\} + m + m \\ &= m\{(2n + 2p) - 1 - 1 + 1 + 1 + 1\} \\ &= m\{2(n + p) + 1\} \\ &O(m\{2(n + p) + 1\}) \end{aligned}$$

Es la complejidad computacional.

- En el paso dos, en la práctica se resuelve dos problemas de optimización

$$x^{k+1} = \arg \min_{x \in \mathbb{R}^m} \left\{ f(x) + \langle p^{k+1}, Ax \rangle + \frac{1}{2\lambda_k} H_{\theta_1}(x, x^k) \right\},$$

$$z^{k+1} = \arg \min_{x \in \mathbb{R}^m} \left\{ g(z) + \langle p^{k+1}, Bx \rangle + \frac{1}{2\lambda_k} H_{\theta_2}(z, z^k) \right\},$$

Resolver estos problemas dependerá del algoritmo empleado, por ejemplo si la distancias $H_{\theta_1}(x, x^k)$, $H_{\theta_2}(z, z^k)$ son autocorcondantes entonces se puede utilizar métodos de puntos interiores cuya complejidad computacional es polinomial.

Sea $O(l)$ la complejidad computacional.

En el paso tres, similar a las operaciones realizadas en el paso uno, obtenemos la complejidad computacional.

$$O(m\{2(n + p) + 1\})$$

- El paso cuatro está relacionadas con la complejidad total del algoritmo, es decir se quiere saber cuántas iteraciones son necesarias para encontrar una ε solución. Es bien conocido que el método Proximal satisface la siguiente desigualdad, ver (Patrascu, Andrei; Irofti, Paul, 2022) (Bento, Ferreira, & Melo, 2016)

$$(k + 1) \left((f + g)(x^k, z^k) - (f + g)^* \right) < C$$

$$(f + g)(x^k, z^k) - (f + g)^* < \frac{C}{k + 1} < \varepsilon$$

$$\frac{C}{\varepsilon} < k + 1$$

$$O\left(\frac{C}{\varepsilon}\right)$$

La complejidad este paso será; $O\left(\frac{C}{\varepsilon}\right)$

Por lo tanto, la complejidad total del algoritmo será:

$$\frac{C}{\varepsilon} (2xO(m\{2(n + p) + 1\}) + 2xO(l))$$

CAPÍTULO V: METODOLOGÍA DE LA INVESTIGACIÓN

5.1. Tipo de Investigación

El tipo de investigación es básica con una gran motivación en las aplicaciones, en especial para la clasificación de datos. Es de carácter básico dado que se estudió la convergencia del algoritmo SC – PMA con la ayuda del álgebra de Jordan, la teoría de cono simétrico de segundo orden, entre otros tópicos del campo de la matemática.

5.2. Técnicas descriptivas para la contratación o demostración de la hipótesis

Para el estudio del algoritmo SC – PMA se utilizó el fundamento del Álgebra de Jordan, y para su convergencia se realizó definiciones, demostraciones de teoremas, proposiciones y lemas, los cuales ayudaron a establecer la teoría necesaria para desarrollar la convergencia.

Se hizo una revisión literaria de fuentes bibliográficas relacionados a este tipo de algoritmo, artículos journal, conference, libros, etc. que ayudaron a establecer la teoría.

5.3. Metodología del desarrollo de Tesis

Se realizaron las siguientes actividades para el desarrollo de la investigación.

5.3.1. Búsqueda de la investigación científica.

Se realizó una revisión sistemática literaria, para ello se utilizó la metodología Kitchenham y Charter, debido a que se ajustaba a la investigación.

Dentro de su metodología se tienen las siguientes fases:

- Planificación de la revisión.
- Realización de la revisión.
- Informe de la revisión.

El cual se puede ver el desarrollo de la revisión literaria a detalle en el Capítulo III.

Además, se desarrolló un artículo de revisión literaria, el cual está indexado en SCOPUS, cuyo título del artículo es “**A Systematic Literature Review on Support Vector Machines Applied to Classification**”, ver *ANEXOS D.1. Artículos Publicados*.

5.3.2. Estudio del Algoritmo SC – PMA

En esta sección se desarrolló el fundamento teórico del algoritmo, es decir las definiciones fundamentales y las demostraciones de los teoremas, colorarios y lemas.

Así mismo se desarrolló la convergencia del algoritmo, se definió y dio ejemplos de distancias proximales los mismos que se utilizaran en la implementación y demostración del algoritmo.

El desarrollo a detalle se puede visualizar en el Capítulo IV.

De la publicación de un artículo teórico el cual está indexado en SCOPUS, cuyo título del artículo es “**A Proximal Multiplier Method for Convex Separable**

Symmetric Cone Optimization”, ver *ANEXOS D.1. Artículos Publicados*.

5.3.3. Implementación del Algoritmo

Para el desarrollo de la implementación del algoritmo se realizaron las siguientes actividades:

- Evaluación y selección del software para la implementación de los algoritmos.
- Estudio de los diferentes algoritmos de optimización para SVM.
- Implementación del algoritmo del cono simétrico.
- Implementación del algoritmo de la máquina de vectores de soporte.
- Evaluación de los resultados.

Entre otras actividades. También se desarrollaron ejemplos con su respectivo análisis representados a través de gráficos y tablas. A mayor detalle se puede visualizar en el Capítulo VI.

Se realizó un artículo el cual está o, cuyo título del artículo es “**A Symmetric Cone Proximal Multiplier Algorithm**”, ver *ANEXOS D.2. Artículos Aceptados***Error! Reference source not found..**

5.4. Desarrollo del diseño del algoritmo SC - PMA

Para realizar el desarrollo del diseño del algoritmo se utilizó el marco metodológico SCRUM, con la ayuda del software Figma, se efectuaron las siguientes actividades:

- Historia de usuario.
- Modelo de base de datos del sistema.
- La arquitectura del sistema, modelo vista controlador (MVC).

Entre otras actividades se pueden visualizar en el Capítulo VII.

Se desarrolló un artículo el cual está aceptado para su publicación en una revista indexada en SCOPUS, cuyo título es “**Design of a Optimization Algorithm for Binary Classification**”, ver *ANEXOS D.2. Artículos Aceptados*.

5.5. Resultados

En esta sección se evidenciaron los resultados obtenidos por el algoritmo, los cuales se validaron mediante el software Matlab.

También se hizo un análisis comparativo con otros tipos de algoritmo proximal.

Para mayor detalle ver el Capítulo VIII.

CAPÍTULO VI: IMPLEMENTACIÓN DEL ALGORITMO

En esta sección presentamos la implementación del SC – PMA propuesto, el cual es aplicado para encontrar hiperplanos lineales en clasificación binaria en SVM.

Se realizaron las siguientes actividades clasificadas por semanas:

Tabla 9
Actividades desarrolladas para la implementación del algoritmo

Semana 1	• Revisión de la propuesta del algoritmo. • Evaluación y selección del software para la implementación de los algoritmos. • Presentación del esquema de implementación
Semana 2	• Estudio y evaluación de la implementación de las máquinas de soporte vectorial. • Estudio de los diferentes algoritmos de optimización para SVM. • Estudio de la propuesta del algoritmo de Cono Simétrico. • Introducir el algoritmo propuesto en el proceso de aprendizaje del SVM
Semana 3	• Implementación del algoritmo de optimización de Cono Simétrico. • Primera evaluación de los resultados. • Segunda evaluación de los resultados. • Presentación de los resultados preliminares del algoritmo de optimización.

Semana 4	• Implementación del algoritmo de Máquinas de Soporte Vectorial (software elegido). • Primera evaluación de los resultados. • Implementación del generados lineal.
Semana 5	• Implementación del programa principal. • Presentaciones preliminares del SC-PMA • Primera evaluación de los resultados. • Segunda evaluación de los resultados con sus parámetros y gráficos.

Presentamos a continuación los fundamentos teóricos necesarios para la implementación del algoritmo. Dado un conjunto de puntos $x^1, x^2, \dots, x^m \in \mathbb{R}^n$ con sus respectivas etiquetas $y^1, y^2, \dots, y^m \in \{-1, +1\}$, formamos la m-upla $(x^1, y^1), (x^2, y^2), \dots, (x^m, y^m)$. El objetivo de la SVM es determinar un hiperplano óptimo.

$$H(\mathbf{w}, \alpha) = \{\mathbf{x} \in \mathbb{R}^n: \mathbf{w}^T \mathbf{x} + \alpha = 0\}$$

Donde $\mathbf{w} \in \mathbb{R}^n$ y $\alpha \in \mathbb{R}$, que separa los puntos dados.

Como observamos en el Capítulo, esa clase de problemas se puede expresar como nuestro modelo (4.1), es decir,

$$\min_{\mathbf{z}, \mathbf{v}} \{f(\mathbf{z}) + g(\mathbf{v}): A\mathbf{z} + B\mathbf{v} = \mathbf{b}, \mathbf{v} \geq 0\},$$

donde las variables son:

$$\begin{aligned} \mathbf{z} = (\mathbf{w}, \alpha) &\in \mathbb{R}^{n+1} \text{ y } \mathbf{v} = (\boldsymbol{\xi}, \mathbf{u}) \in \mathbb{R}^{2m} \text{ con} \\ \boldsymbol{\xi} &= (\xi_1, \dots, \xi_m) \in \mathbb{R}^m \text{ with } \xi_i \geq 0 \\ \mathbf{u} &= (u_1, \dots, u_m) \in \mathbb{R}^m \text{ with } u_i \geq 0 \end{aligned}$$

Las funciones separables son

$$f(\mathbf{z}) = \frac{1}{2} \|\mathbf{w}\|^2 \text{ y } g(\mathbf{v}) = C \mathbf{e}^T \boldsymbol{\xi},$$

La matriz A y B son las siguientes

$$A = Y\hat{X} \in \mathbb{R}^{m \times (n+1)}, \quad B = (I - I) \in \mathbb{B}^{m \times 2m}$$

dónde

$$Y = \text{Diag}(\mathbf{y}) = \text{Diag}(y^1, y^2, \dots, y^m)$$

y

$$\hat{X} = \begin{pmatrix} x^{1T} & 1 \\ \vdots & \vdots \\ x^m & 1 \end{pmatrix} \in \mathbb{R}^{m \times (n+1)},$$

Finalmente, el vector b viene dado por $\mathbf{b} = \mathbf{e} = (\mathbf{1}, \mathbf{1}, \dots, \mathbf{1}) \in \mathbb{R}^m$.

Utilizamos el software Matlab (R2017a), una computadora CPU Intel (R) Core (TM) i5-8250U de octava generación, 1.60 GHz, 1.80 GHz, 4.00 GB, Windows 1064 bits.

Daremos tres ejemplos de búsqueda de hiperplanos óptimos usando el SC-PMA con la cantidad de datos de $m=10$, $m=50$ y $m=100$, respectivamente, obtenidos por una implementación usando la función Rand de Matlab.

6.1. Entrada y salida del SC - PMA

La siguiente figura 1 nos muestra el diagrama de flujo de los parámetros necesarios para ingresar a la implementación del algoritmo SC-PMA y los resultados obtenidos luego del procesamiento.

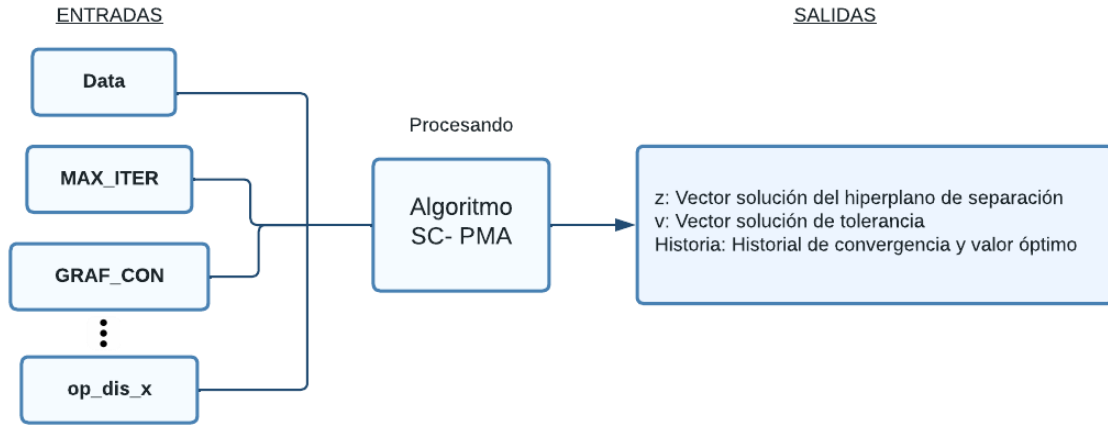


Figura 10 Diagrama de Procesamiento SC - PMA.

Los parámetros por ingresar son:

Datos: Conjunto de datos separables linealmente

- MAX_ITER: Número máximo de iteraciones (el número de interacciones máximas dadas al algoritmo)
- GRAF_CON: (1) Para graficar y (0) Para no graficar (con (1) muestra la gráfica del plano en otro caso (0) no muestra el hiperplano de separación)
- ITER_CON: (1) Para mostrar convergencia y (0) Para no mostrar (con (1) muestra la gráfica de la convergencia del punto óptimo también de “z” y “v” otro caso (0) no presenta la separación hiperplano)
- Rho: Parámetro de aceleración.
- Lambda: Tamaño de paso en cada iteración.
- Punto inicial $\omega^0 = (z^0, v^0, y^0) \in \mathbb{R}^{n+1} \times \mathbb{R}^{2m} \times \mathbb{R}^m$ un punto de partida arbitrario.
- Op_dis_f: Distancia proximal para f .
- Op_dis_g: Distancia proximal para g .
- Op_dis_x: Tipo de distancia proximal.

En el SC-PMA son necesarias dos distancias proximales. La primera distancia relacionada con la variable $\mathbf{z} = (\mathbf{w}, \alpha) \in \mathbb{R}^{n+1}$ (asociada a op_disf):

$$H(z, y) = \sqrt{\sum_{i=1}^{n+1} (z_i - y_i)^2}$$

La otra distancia está relacionada con la variable $\mathbf{v} = (\boldsymbol{\xi}, \mathbf{u}) \in \mathbb{R}^{2m}$ (asociada a `op_dis_g`). Como esta variable tiene condiciones de no negatividad, es decir, $\xi_i \geq 0$ y $u_i \geq 0$, para $i = 1, 2, \dots, m$; podemos elegir en la ejecución cualquiera de las siguientes distancias:

1) *Elección de la distancia Kullback – Leibler Bregman:*

$$H(v, y) = \sum_{i=1}^{2m} \left(v_i \log \left(\frac{v_i}{y_i} \right) + y_i - v_i \right).$$

2) *Distancia proximal de Itakura Saito:*

$$H(v, y) = \sum_{i=1}^{2m} \left(\frac{v_i}{y_i} - \log \left(\frac{v_i}{y_i} \right) \right) - 1$$

3) *Distancia proximal homogénea de segundo orden:*

$$H(z, y) = \sum_{i=1}^{2m} \frac{y}{2} (z_i - y_i)^2 + \sigma(y_i^2 \log \frac{y_i}{z_i} + z_i y_i - y_i^2)$$

Debemos observar que estas distancias se utilizan para medir la separación de los puntos con respecto al hiperplano de separación. Además, como la implementación del algoritmo depende de la distancia proximal empleada, estas distancias también se utilizan para comparar cuál de ellas converge mejor.

Las salidas después del proceso del SC-PMA son las siguientes:

- $\mathbf{z}$: Vector solución del hiperplano de separación
- $\mathbf{v}$: Vector solución de tolerancias
- Historial: historial de convergencia y valor óptimo

La historia es muy importante para obtener las tablas 10 y 11, donde vamos a mostrar

- El número de iteraciones
- El número de iteraciones internas para resolver cada subproblema (4.7) y (4.8) denotado por $N(\mathbf{z}_{\{k\}}), N(\mathbf{v}_{\{k\}})$, respectivamente.
- La diferencia entre puntos consecutivos de cada variable:

$$\|\mathbf{z}_{\{k\}} - \mathbf{z}_{\{k-1\}}\|, \|\mathbf{v}_{\{k\}} - \mathbf{v}_{\{k-1\}}\| \text{ y } \|\mathbf{w}_{\{k\}} - \mathbf{w}_{\{k-1\}}\|$$

- El valor de la función objetivo en cada iteración.

La implementación del algoritmo SC-PMA ver en *ANEXO A.4. SVM_LIN_SC_PMA*

6.2. Generación Lineal

Para generar los datos para la prueba del algoritmo, se implementó un generador lineal que encuentra aleatoriamente los puntos en el plano. Usamos el comando **rand** del Matlab software y el número de puntos se elige con los requisitos que se describen a continuación.

Las entradas para generar los datos son las siguientes:

- **n_ var:** Número de variables (Dependiendo de la dimensión, puede ser bidimensional o tridimensional).
En nuestro caso, usamos el valor $n_var = 3$, porque los puntos pertenecen a esta dimensión.
- **n_ obs:** Número de observaciones (Cantidad de datos a clasificar con el algoritmo).
El valor de n_obs cambia según la cantidad de datos utilizados. En nuestra implementación usamos los valores de 10, 50 y 100.
- **Mínimo:** Valor mínimo de los cuadrados (Límite inferior del intervalo)
- **Máximo:** Valor máximo de los cuadrados (Límite más alto del intervalo)
- **GRAF:** (1) para gráfico (Presenta el conjunto de datos en un espacio tridimensional) y (0) para ningún gráfico (No presenta el conjunto de datos en un espacio tridimensional).

El proceso de salida del SC-PMA es:

- **Datos:** Datos generados que son linealmente separables.

Generaremos puntos $\mathbb{R}^3$, para que el valor de n en el modelo (3.3) sea 3, es decir, $n = 3$. la implementación de esta generación lineal se muestra en el apéndice.

La implementación ver en *ANEXO A.2. Generador Lineal*

6.3. Subprograma Principal

Este programa integra el generador lineal con el SC-PMA. Después de haber generado los datos a través del generador lineal, el subprograma principal de la implementación llama al SC -PMA y procede a encontrar el plano de separación, el número de iteraciones y el tiempo de ejecución.

Así, tenemos como resultados el plano de separación, el valor óptimo, así como los puntos solución "z" y "v". Como

$$\mathbf{z} = (\mathbf{w}, \alpha) \in \mathbb{R}^{n+1},$$

entonces, obtendremos el hiperplano óptimo definido por

$$H(\mathbf{w}, \alpha) = \{\mathbf{x} \in \mathbb{R}^n: \mathbf{w}^T \mathbf{x} + \alpha = 0\}.$$

Para ejecutar el SC-PMA es necesario resolver en cada iteración los subproblemas (3.9) y (3.10). En este artículo usamos el comando **fmincon**.

Presentamos ejemplos numéricos de fecha creados por un generador lineal aleatorio. Para esta presentación consideramos la distancia proximal homogénea de segundo orden:

$$H_\rho(u, v) = \sum_{i=1}^n \rho \left(v_i^2 \ln \frac{v_i}{u_i} + u_i v_i - v_i^2 \right) + \frac{\rho}{2} \sum_{i=1}^n (u_i - v_i)^2$$

y diferentes valores para el parámetro proximal: $\lambda = 0.09$ para el primer ejemplo, $\lambda = 0.05$. Para el segundo ejemplo y $\lambda = 0.01$ para el tercer ejemplo. Observamos que a medida que λ decrece las iteraciones del algoritmo son menores.

La implementación ver *ANEXO A.3. CVX_SVM (DATA)*, *ANEXO A.1. Función Principal*

6.4. Ejemplo 1

Para este ejemplo, consideraremos el siguiente conjunto de datos que es linealmente separable como se muestra en la Figura 11 También consideramos

$$\lambda_k = 0.09, \rho = 1, tol = 10^{-3}$$

y el número máximo de iteraciones 1500.

En la siguiente figura Figura 12, vemos la convergencia de la función objetivo $f + g$, la variable z y v . Cuando escribimos convergencia de la función objetivo nos referimos a la diferencia en valor absoluto entre la función objetivo $f + g$ obtenida por el SC-PMA y la función $f + g$ objetivo obtenido por la función CVX de Matlab.

La Figura 13 muestra el conjunto de 10 observaciones que se distribuyen en el espacio tridimensional y son linealmente separables.

tiempo transcurrido es 2344.122166 de segundos, la convergencia ocurrió en la iteración 967. La Figura 13 muestra el conjunto de datos separados por un plano, cuyos coeficientes de la ecuación son:

$$-0.2913; -0.1007; -0.4574; 0.0251;$$

y su respectivo hiperplano es:

$$-0.2913X - 0.1007Y - 0.4574Z + 0.0251 = 0.$$

La Tabla 10 muestra los resultados computacionales.

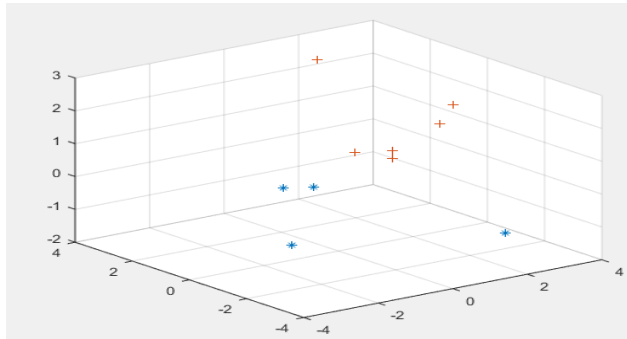


Figura 11 Datos separables lineales.

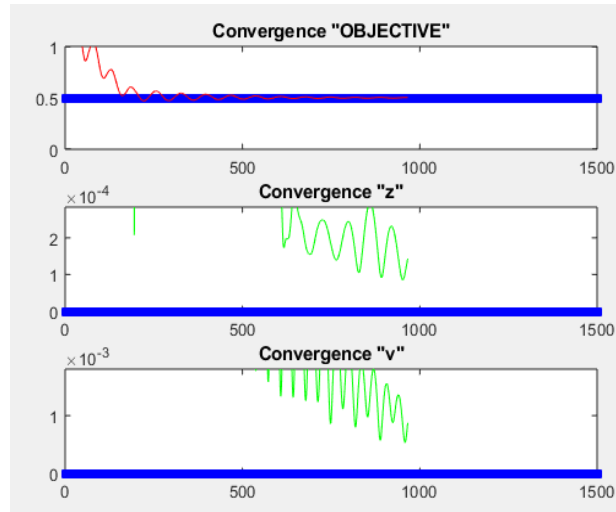


Figura 12 Convergencia de la función objetivo y las normas $\|z_k - z_{k-1}\|$ y $\|v_k - v_{k-1}\|$

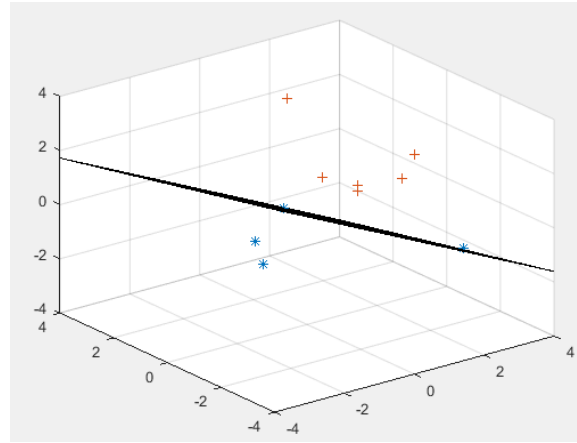


Figura 13 Conjunto de datos separados por un plano.

La Tabla 10 muestra los resultados computacionales obtenidos luego del procesamiento de los datos, así vemos la convergencia al valor óptimo, el número de iteraciones, el módulo de la diferencia de puntos de z_k con respecto al anterior z_{k-1} de igual manera para v_k con respecto al anterior v_{k-1} y así también de w_k con respecto a la anterior w_{k-1} .

Tabla 10
Resultados computacionales para el Ejemplo 1.

Iteración	$N(z_{\{k\}})$	$N(v_{\{k\}})$	$\ z_{\{k\}} - z_{\{k-1\}}\ $	$\ v_{\{k\}} - v_{\{k-1\}}\ $	$\ w_{\{k\}} - w_{\{k-1\}}\ $	Función objetiva
1	4	14	0.78167	0.32408	1.43995	3.18318
2	9	22	0.48174	0.23141	1.25799	2.56337

3	8	19	0.87878	0.25250	0.87878	1.51274
4	8	24	0.93322	0.25200	0.93322	1.00437
5	8	20	0.72358	0.20552	0.72358	1.05220
⋮	⋮	⋮	⋮	⋮	⋮	⋮
966	5	33	0.00014	0.00087	0.00104	0.50203
967	5	32	0.00015	0.00093	0.00098	0.50188

6.5. Ejemplo 2

Para este ejemplo, consideraremos $\lambda = 0.05$, $\rho = 1$, $tol = 10^{-3}$ también $Max.Iterations = 2000$. La Figura 14 muestra el conjunto de 50 observaciones que se distribuyen en el espacio tridimensional y son linealmente separables. En la Figura 15 vemos converger a la función objetivo y las variables la variable “z” y “v” a un punto óptimo.

Se puede observar que la convergencia del valor óptimo es rápida al igual que z pero es más lenta con respecto al parámetro v. Al igual que en el ejemplo anterior, la convergencia de la función objetivo se refiere a la diferencia en valor absoluto entre la función objetivo $f + g$ obtenida por el SC-PMA y la función $f + g$ objetivo obtenido por la función CVX de Matlab.

El tiempo transcurrido es 4930.763008 segundos y la Figura 16 muestra el conjunto de datos separados por un plano, cuyos coeficientes de la ecuación son: -0.7259 ; -0.1075 ; -0.4545 ; 1.1248 ; su respectivo plano es:

$$-0.7259X - 0.1075Y - 0.4545Z + 1.1248 = 0$$

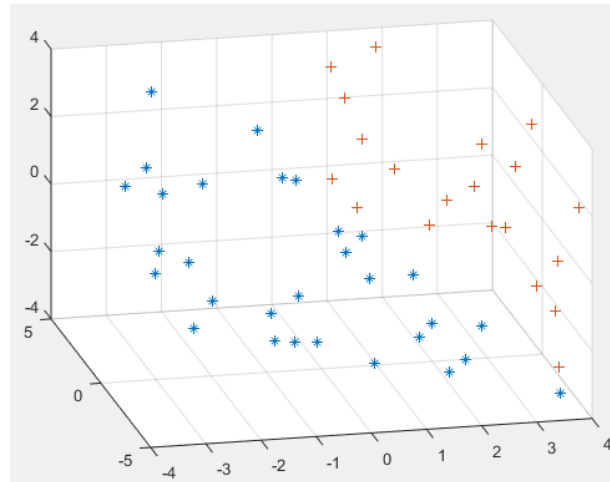


Figura 14 Datos separables lineales.

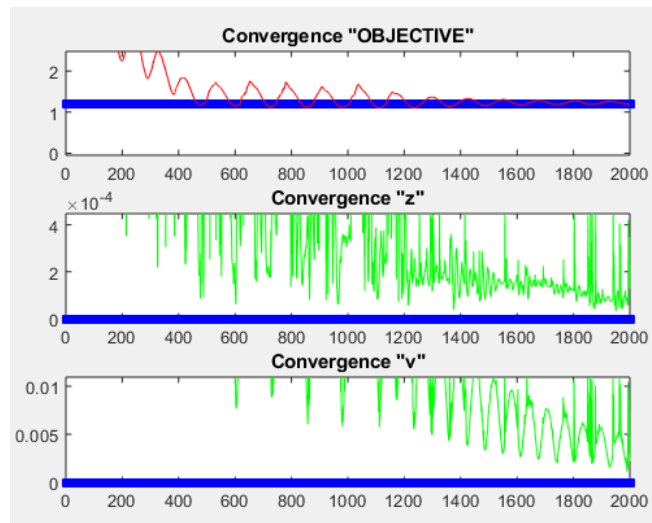


Figura 15 Convergencia de la función objetivo y las normas $\|z_k - z_{k-1}\|$ y $\|v_k - v_{k-1}\|$

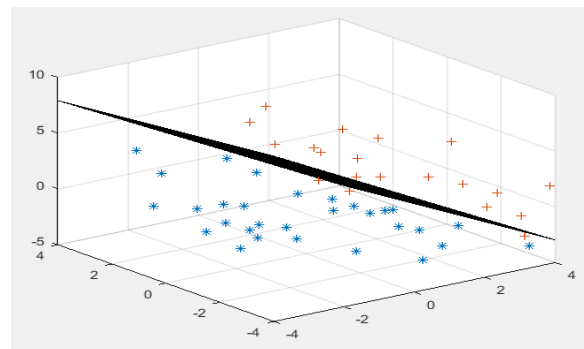


Figura 16 Conjunto de datos separados por un plano.

La Tabla 11 muestra los resultados computacionales para un dato de 50 observaciones,

Tabla 11
Resultados computacionales para el Ejemplo 2

Iteración	$N(\mathbf{z}_{\{k\}})$	$N(\mathbf{v}_{\{k\}})$	$\ \mathbf{z}_{\{k\}} - \mathbf{z}_{\{k-1\}}\ $	$\ \mathbf{v}_{\{k\}} - \mathbf{v}_{\{k-1\}}\ $	$\ \mathbf{w}_{\{k\}} - \mathbf{w}_{\{k-1\}}\ $	Función objetiva
1	12	29	2.73490	0.44568	3.48678	13.55714
2	10	22	1.78088	0.33687	2.36787	8.60897
3	9	29	2.31439	0.33125	2.31439	4.92598
4	10	29	1.37369	0.25792	1.37369	4.92598
5	8	28	0.34034	0.20416	1.01441	4.96926
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1999	6	29	0.00014	0.00265	0.00430	1.19902
2000	4	29	0.00013	0.00220	0.00423	1.19951

6.6. Conclusiones

En el presente artículo, presentamos un algoritmo multiplicador proximal de cono simétrico (SC-PMA) para resolver problemas de optimización separables. Se demuestra que el punto de convergencia de las variables primarias duales es un punto de silla del Lagrangiana asociado al problema y por lo tanto resolvemos el problema de optimización. Luego, aplicamos SC-PMA para encontrar hiperplanos lineales en clasificación binaria en una máquina de vectores de soporte. Esta es la primera vez que se implementa el SC-PMA para resolver esta clase de problemas aplicados a la inteligencia artificial y los resultados obtenidos motivan más investigaciones.

Este artículo es la continuación de trabajos anteriores publicados y desarrollados por los autores. En los artículos (Cano & Papa, 2020) , (Nieto, Quiroz, & Cano Lengua, 2021) desarrollamos una revisión sistemática sobre máquinas de vectores de soporte aplicadas a clasificación y regresión, respectivamente. En el artículo (López, J.; Papa Quiroz, E.A., 2017) desarrollamos una metodología para construir distancias proximales sobre conos simétricos y en el artículo anterior (Papa, López, & Cano, 2020) obtuvimos resultados

matemáticos preliminares del SC-PMA. En este artículo consolidamos los resultados de convergencia del SC-PMA y lo aplicamos para encontrar hiperplanos lineales a la clasificación binaria en SVM, complementando así el resultado del artículo.

Para mejorar los resultados computacionales es necesario resolver los problemas (3.9) y (3.10) de manera eficiente. Para esos casos, pensamos que la familia Bunde de métodos (Bonnans, Gilbert, Lemarechal, & Sagastizábal, 2006), cap. 9) es quizás la herramienta computacional más práctica para la optimización no suave. Se puede considerar como una investigación futura para mejorar la eficiencia del algoritmo.

Otra investigación futura puede ser comparar el SC-PMA con los diferentes algoritmos de clasificación como redes neuronales, clasificadores bayesianos y máquinas de vectores de soporte, entre otros.

Además, se realizó el diseño del sistema del algoritmo SC – PMA el mismo que fue dado usando el marco referencial SCRUM el cual se puede ver en el ANEXO B([linkear](#))

CAPÍTULO VII. DISEÑO

Para el desarrollo del diseño del sistema se ha decidió implementar la metodología ágil Scrum, del cual se tiene varios trabajo desarrollados con esta metodología por ejemplo para la implementación de un sistema de una tienda virtual ver (Ramirez, Diestra, & Cano Lengua, 2021) , otro trabajo importante con esta metodología es el desarrollados para un software de aplicaciones móviles para el monitoreo de pacientes Pos-Covid-19 (Palomino Gonzales, Cano Lengua, & Andrade - Arenas, 2022). Debido a que se adapta bastante bien a nuestro marco de trabajo y que la composición de nuestro equipo es reducida incorporamos a dos miembros para que formen parte del equipo, el miso que se centra en la multifuncionalidad de todos los miembros es así como, se le asignan distintas funciones y los miembros son toditas, por ello se ha utilizado la metodología ágil Scrum para el desarrollo de esta investigación.

Requerimos de algunos conceptos teóricos para el diseño del sistema:

7.1. Sistema

La automatización informática, también conocida como automatización de infraestructura, es el uso de un sistema de software para crear instrucciones y operaciones repetibles que reemplazan o reducen la interacción humana con el sistema. También engloba una serie de elementos que concreta un objetivo en una acción que interviene la persona.

7.2. Herramienta de Desarrollo

7.2.1 NetBeans

Es una herramienta de trabajo de acceso libre, que está enfocado al desarrollo de aplicaciones y permiten mostrar una estructurada componente en sus módulos. NetBeans es un entorno de desarrollo que permite realizar con facilidad y rapidez aplicaciones de java, así como también Html5 y contiene herramientas para desarrolladores de PHP, C++.

7.2.2 Lenguaje de Programación Java

Esta herramienta es de lenguaje sencilla y está orientado a objetos, que facilita la interacción en diversas áreas, que consta de un cliente servidor y de interfaces graficas.

7.2.3 Figma

Es una plataforma de diseño construida a escala, el cual es utilizada para, diseños, prototipos, interfaces, entre otras aplicaciones, Figma es potente software colaborativo basado en web disponible para macos y Windows.

Mediante este software se diseñarán los mockups.

7.2.4 SQL SERVER

Es un sistema de gestión de bases de datos relacional, brinda servicio a una amplia gama de aplicaciones de software relacionadas, a este servicio se puede autenticar varios tipos de usuarios.

Este sistema está formado por archivos, tablas, etc.

7.2.5 MVC

Se conoce como (modelo, vista, controlador) por su conectividad y es una arquitectura de software, se enfoca en separar la lógica del negocio y su entorno de visualización.

Según (Aguilar, 2019). El MVC web es un patrón donde se utiliza para desarrollar entornos web y usa un modelo de datos tradicional como SQL y su código de control se puede realizar en HTML/JavaScript, tiene una similitud con MVC teniendo en cuenta que sus componentes con más estrictos.

7.2.6 Modelos

Es un formulario que tiene una base de requerimientos del sistema, si este se realiza cambios entonces afectara a la vista y de ser necesario al controlador.

7.2.7 Vista

Es una interfaz o formulario visible que muestra la información que están dentro del base de datos.

7.2.8 Controladores

Funciona según la lógica para controlar al funcionamiento de la vista en la intervención del usuario en la aplicación.

7.2.9 Framework

Es una colección de herramientas que facilita el desarrollo web, tiene un conjunto de

módulos que ayuda al desarrollo en un tiempo más rápido, al tener una biblioteca y funciones ya desarrolladas.

A continuación, podemos ver en la siguiente Figura 17 un ciclo de vida de los proyectos realizados con esta metodología ágil.

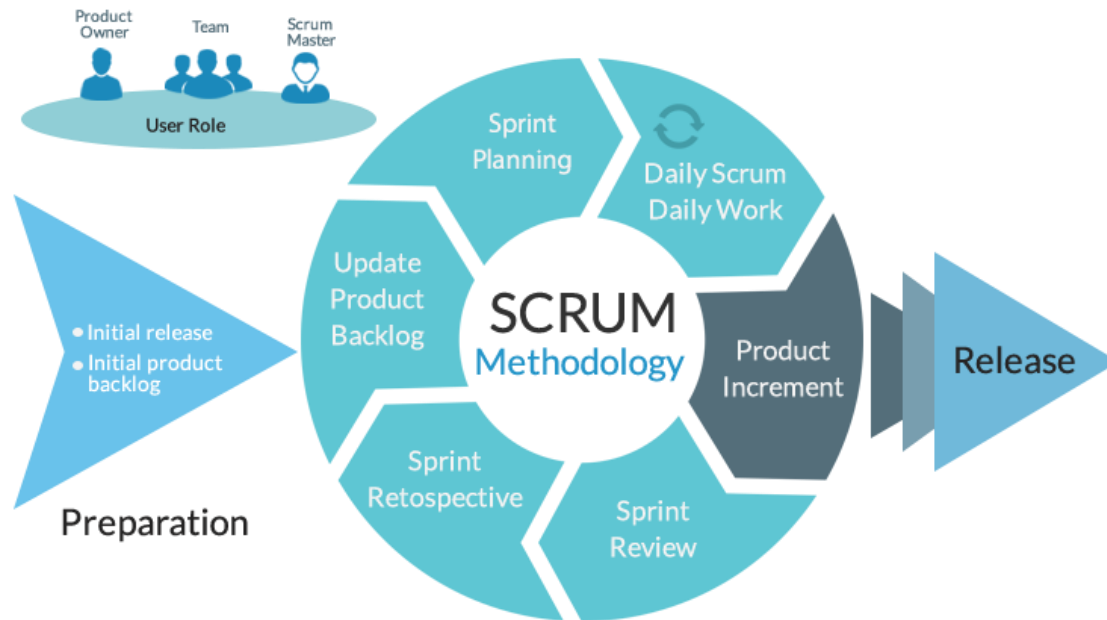


Figura 17 Ciclo de vida del Marco Metodológico de SRUM

A diferencia de un desarrollo de trabajo de una manera más tradicional el cual se basa en un orden y jerarquía, el equipo de trabajo en un proyecto Scrum debe ser debe ser apto de autoorganizarse y los miembros capaces de asumir cualquier actividad necesaria para completar y desarrollar el proyecto. Para ello se detallarán las 5 fases definidas en el Scrum, según Salpathy, (2016), las cuales son: Inicio; Planificación y Estimación; Implementación, Revisión y Retrospectiva; y Lanzamiento.

7.3. Principios de Scrum

Scrum son consideraciones fundamentales que se deben tener al ejecutar el marco metodológico y se emplean en todos los proyectos que definan esta metodología como

base para el desarrollo.

Los 6 principios Scrum son los siguientes:

- Inspección de series rutinarias
- Organización propia
- Contribución
- Primar el valor
- Caja de Tiempo
- Progreso repetitivo.

7.4. Roles

Los miembros del equipo de la presente investigación serán designados a cumplir los roles según sus habilidades y capacidades, asimismo los autores tienen en cuenta el trabajo en equipo como una de las ventajas que presenta Scrum, en donde se considera el compromiso de cada miembro para ayudarse a resolver problemas que puedan presentarse en el desarrollo de sus actividades.

El aporte de mejoras continuo en el proyecto puede tratar superar cualquier barrera que impida al equipo lograr sus objetivos de sprint.

7.4.1. Product Owner

Es el profesional que se encarga de gestionar el producto y encargado de priorizar el trabajo a desarrollar. Pero además es encargado de recolectar los requerimientos del cliente, que son escuchados en el producto backlog. Asimismo, se designó para el rol de Product Owner a Miguel Cano, debido a que tiene las capacidades adecuadas para realizar la toma de requerimientos y se encargará de comunicar la visión del producto al equipo de desarrollo.

7.4.2. Equipo de Desarrollo

Según (Vlietland & Van Vliet, 2015) señala que el grupo desarrollador es el encargado de implementar la funcionalidad basada en el producto backlog.

Es por ello que se designó a dos especialistas que se desempeñarán en el rol de Development team Marcos y Carlos Alvarado, puesto que tienen la experiencia requerida para el realizar el diseño de sistemas.

7.4.3. Scrum Máster

De acuerdo con (Cho, 2008) menciona que el scrum máster se asegura que el valor y las normas de Scrum se ejecuten de manera precisa, facilitando las actividades relacionadas con el equipo de desarrollo. Respecto a la designación de los roles del proyecto, definimos como Scrum Master a Erik Papa ya que cuenta con la experiencia necesaria en el manejo equipos de trabajo.

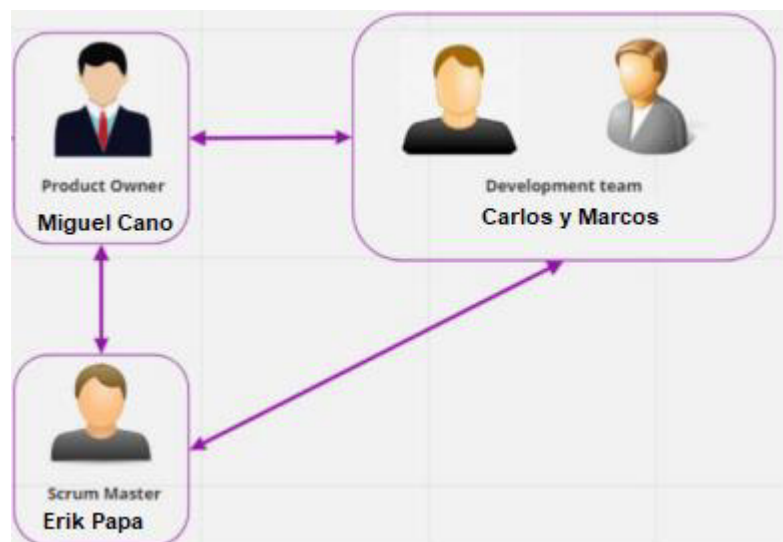


Figura 18 Diagrama de equipo Scrum

7.5. Artefactos de Scrum

Los artefactos Scrum, son información o valor de trabajo, y se muestran en una variedad de tipos, ayudan a proporcionar transparencia y permiten la supervisión y la

personalización. En la definición de artefactos en Scrum, son específicamente creados para aumentar la transparencia en la información crítica, para garantizar que todos obtengan la misma comprensión del artefacto.

7.5.1. Lista de Producto (Product Backlog)

La Lista de Producto, ordena la totalidad de cambios, correcciones, problemas y mejoras encontrados, que modifican las versiones siguientes del producto. Los artículos de inventario tienen descripción, orden, estimación y valor como atributos. Los elementos de la cartera de productos a menudo incluyen descripciones de prueba que demostrarán la integridad de los elementos cuando estén "completos".

7.5.2. Lista de Pendientes del Sprint (Sprint Backlog)

Sprint Backlog es una pila de productos pendientes, seleccionados para el Sprint en conjunto con un planeamiento, y así impulsar el crecimiento del producto cumpliendo las metas del Sprint. El Sprint Backlog nos permite proyectar distintas estrategias y funcionalidades para la siguiente fase de desarrollo del Spring.

7.5.3. Incremento

El incremento, es la suma general de la lista de productos terminados en el sprint y la suma total de todos los Sprint anteriores. La fase final del sprint, el nuevo crecimiento tiene que ser "completado", para después se pueda usar y ejecutar, la definición de "finalización" del equipo Scrum. Al final del sprint, hay trabajo incremental y comprobable que respalda el trabajo empírico. El incremento es una parte hacia la finalización de la meta. Es importante que el delta esté disponible, es responsabilidad del propietario del producto, decidir liberar el delta.

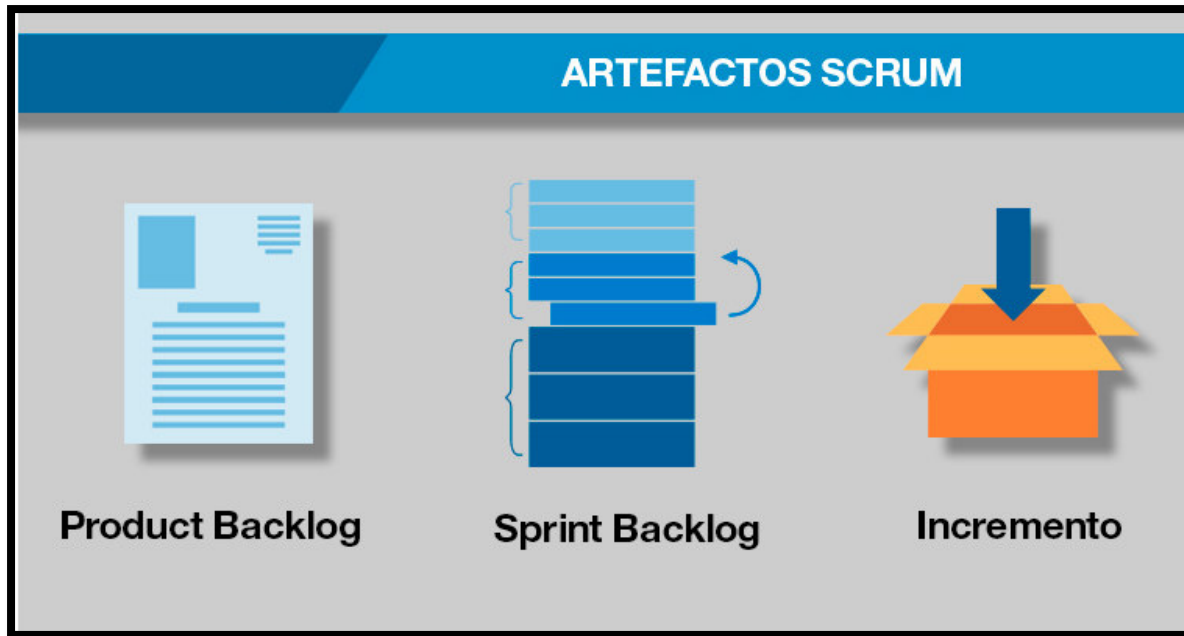


Figura 19 Artefactos de Scrum

7.6. Ciclo de vida de la Metodología SRUM

7.6.1. Inicio del Proyecto

Aquí se definirán los encargados del proyecto, así como también las necesidades básicas para los Sprint en general. Nos enfocaremos en detallar los objetivos primarios de nuestro proyecto, así como también los objetivos específicos.

7.6.2. Planificación

En este apartado se llevará a establecer ciertas metas de nuestro proyecto, así como entender la necesidad de nuestro cliente, traducida a requerimientos, los cuales se convertirán en tareas definidas para el Scrum Team y serán ordenadas en un Sprint Backlog.

- **Sprint Planning.** Es en la fase del Spring Planning en la cual se crea el Sprint Backlog con todo su contenido. Se dividen las tareas en Sprints para que estas sean más fáciles de abordar, sean pequeñas y se pueda tener un mejor manejo sobre

ellas. Esto facilita la obtención de resultados previos para así poder localizar errores o desviaciones de los requisitos por lo que permite trabajar con menor incertidumbre y tener los objetivos más claros.

7.6.3. Implementación

Durante esta fase se realizará la ejecución de las tareas generadas durante la planificación, además se detallarán las actividades a llevar a cabo para generar los productos en el proyecto. Esta fase tiene dos procesos importantes, el mantenimiento y actualización de pendientes y el daily scrum que describiremos a continuación.

- **Daily Scrum**

El Daily Scrum es una reunión que se realiza de manera diaria entre el equipo de trabajo, así como también el Scrum Máster con el fin de mantener el ritmo de trabajo y fomentar la comunicación entre los miembros del equipo.

7.6.4. Review y Retrospectiva

En la etapa de revisión y retrospectiva es en la cual todo el equipo de trabajo se junta para hacer una reunión y repasar todo lo trabajado hasta el momento, con el fin de tener una mejor organización lo que permita a futuro trabajar de manera más eficiente. Mediante el Review se analiza el resultado del trabajo realizado y la retrospectiva nos permite mejorar la manera con la cual se ha venido desarrollando el proyecto. El momento propicio para realizar estos 2 procesos es siempre al terminar y al comenzar una iteración para lograr una mejora constante en el flujo de trabajo.

7.6.5. Lanzamiento

En la etapa del desarrollo del proyecto de Scrum, se realizará la entrega al cliente del producto o servicio final generado, para posteriormente documentarlo.

7.7. Caso de Estudio

7.7.1. Inicio del Proyecto

Este proceso forma parte de una fórmula para capturar requerimientos y tiene como finalidad explicar de manera general una función que cumplirá el software, además de los criterios de aceptación de cada una de ellas. En cada historia se definirá de manera breve, las funcionalidades establecidas por los interesados en el proyecto y propietarios del producto, las cuales estarán definidas por los requerimientos de estos y serán de fácil entendimiento para todos los implicados. El proceso para definir estas historias tiene como involucrados por lo general al propietario del producto, el experto en Scrum y el equipo de Scrum, con el fin de facilitar el desglose y la transformación de productos priorizados, asegurar la calidad en el producto entregado incluso antes de llegar a la etapa de planificación de procesos para crear el producto en cuestión.

Esta herramienta nos permite enfocarnos en el valor nato de cada parte del entregable y en como estos generan algún efecto en otros componentes del proyecto, en lugar de pasar directamente a la fase de desarrollo de detalles del producto, según (Salazar & Abad Londoño, 2018)

La utilidad de esta herramienta se ve reflejada en situaciones en donde los requerimientos del producto son cambiantes, cosa que sucede en la mayoría de los proyectos. Esto ayuda de manera que el equipo Scrum ayude a resolver las dudas que los interesados puedan tener a lo largo de esta etapa.

Con ello podemos concluir que el uso de esta herramienta nos aporta múltiples ventajas al momento de definir los requisitos del producto, ya que fomenta la colaboración constante entre miembros del equipo de desarrollo o Scrum Team, así como también le permite al cliente tener una perspectiva más amplia y real de los procesos usados para desarrollar el proyecto y poder definir de manera más precisa los requisitos que no estén claros o sean volátiles.

De esta manera simplificaremos el proyecto a pequeños entregables parciales de fácil estimación, estos entregables tendrán una duración aproximada de unas pocas semanas y serán de (Salazar & Abad Londoño, 2018)fácil mantenimiento, según (Menzinsky, y

otros, 2018).

7.7.2. Usuarios del proyecto

De acuerdo con el objetivo del proyecto, los investigadores establecen las historias de usuarios que interactuarán con el diseño a elaborar, específicamente aquellos que ocupen uno de los cargos de la siguiente tabla, la cual se mostrará en orden de relevancia.

Tabla 12
Historia de Usuario 1

Historias de Usuario	
Número: 1	Usuario: Product Owner
Nombre: Logeo por credenciales	
Prioridad: Alta	Riesgo: Medio
Puntos estimados: 2	Iteración: 1
Responsable: Miguel Cano Lengua	
Descripción: Los usuarios que deseen usar el aplicativo se deben poder logear en este mediante una interfaz de validación de credenciales ingresando las designadas previamente por el administrador.	
Validación: El usuario va a poder acceder al sistema mediante un logeo ingresando las credenciales asignadas por el administrador, las cuales serán validadas mediante un servicio de consulta y autenticación con la base de datos	

Tabla 13
Historia de Usuario 2

Historias de Usuario	
Número: 2	Usuario: Product Owner
Nombre: Generación Lineal de Datos	
Prioridad: Alta	Riesgo: Bajo
Puntos estimados: 5	Iteración: 2
Responsable: Miguel Cano Lengua	

Descripción: Deseo que los usuarios puedan generar datos linealmente separables a través de una interfaz que acepte variables definidas por el usuario, estos datos serán solamente visibles a través de un plano tridimensional
Validación: En esta pantalla el usuario va a poder ingresar datos del mínimo y máximo valor de cuadrante, cantidad de observaciones y tomar la decisión de generar o no el gráfico en R3.

Tabla 14
Historia de Usuario 3

Historias de Usuario	
Número: 3	Usuario: Product Owner
Nombre: Generación de gráfico en R3 y procesado de datos	
Prioridad: Alta	Riesgo: Alto
Puntos estimados: 13	Iteración: 2
Responsable: Erik Papa Quiroz	
Descripción: En una interfaz el usuario debe poder procesar los datos y obtener un hiperplano que divida el espacio en R3 y de esta manera clasificar los datos, con ello facilitar la toma de decisiones.	
Validación: El usuario podrá ver los datos previamente procesados y delimitados en un hiperplano, los cuales van a ser almacenados temporalmente de manera local.	

7.7.3. Planning Poker

Para estimar la prioridad de las historias de usuario se ha utilizado el Planning Poker que es una técnica de estimación en SCRUM. Esto consiste en que cada miembro del equipo tiene una baraja de cartas modificada que se corresponde con la secuencia de Fibonacci. Cuando una tarea está lista para la estimación, cada miembro elige individualmente una tarjeta que mejor se adapte a su estimación para ese elemento. Luego, todos revelan sus

cartas simultáneamente y discuten sus estimaciones hasta llegar a un consenso.

La secuencia de Fibonacci es una serie de números. Crecen exponencialmente porque cada número es la suma de los dos números anteriores: 0, 1, 2, 3, 5, 8, 13, 21, etc.

La secuencia fue inventada en la Edad Media por el matemático italiano Leonardo Bonacci, también conocido como Fibonacci. Verheyen (2019) p.118.

7.7.4. Diagrama de Gantt

El Diagrama de Gantt es una herramienta que sirve para tener definido el alcance del proyecto, en donde se plasman las tareas y actividades necesarias que forman parte de este, así como sus dependencias. En sí se puede considerar que el diagrama de Gantt es el cronograma del proyecto expresado de manera en que se puedan estimar las tareas con tiempos, así como los responsables de cada tarea, los pesos y el porcentaje del progreso de estas [22].

7.8. Resultados

7.8.1. Desarrollo de los prototipos de las Historias de Usuario

7.8.1.1. Presentación de los Sprint

En esta etapa del desarrollo del proyecto, se mostrarán los prototipos del sistema web, ordenados de acuerdo con los sprints a los que pertenecen. Cada imagen especifica a que historia de usuario pertenece y los criterios que han seguido para la creación del diseño.

7.8.1.1.1. Primer Sprint

La historia de usuario que abarca el primer sprint está dado por la historia de usuario 1.

H1. Como administrador quiero que exista la opción de loguearse en el sistema por medio de las credenciales dadas.

Para la historia H1 se adjunta la imagen del prototipo Login, en donde se muestra un interfaz mediante el cual el administrador se loguear y accede a las características del sistema brindado las credenciales definidas previamente, esto se muestra en la Figura 20.



Figura 20 Logueo Clasificación

7.8.1.1.2. Segundo Sprint

Incluye dos historias de usuario, historia de usuario H2 e historia de usuario H3.

H2. Deseo que los usuarios puedan generar datos linealmente separables a través de una interfaz que acepte variables definidas por el usuario, estos datos serán solamente visibles a través de un plano tridimensional, de acuerdo con las condiciones dadas.

H3. En una interfaz el usuario debe poder procesar los datos y obtener un hiperplano que divida los datos evaluados en el espacio tridimensional R^3 y de esta manera visualizar los datos, con ello facilitar la toma de decisiones.

Estos resultados de acuerdo con las historias de usuario H2, H3 s pueden visualizar en la Figura 21 y Figura 22.

GENERADOR LINEAL

GeneradorLineal

Ingrese mínimo valor de los cuadrantes : Generar grafico :

Cuadrante min. ☐ Si ☐ No

Ingrese máximo valor de los cuadrantes : Ingrese número observaciones :

Cuadrante max. Cant.

Generar Data

3.3067	3.6952	2.4005	-1.0000
-2.7810	-3.9629	-0.5487	1.0000
2.6065	2.1993	3.2852	-1.0000
0.3067	2.5384	-2.5452	-1.0000
3.9691	2.9496	-1.8896	-1.0000
-3.3746	-3.3245	-2.8357	1.0000
-0.4586	-0.8017	-2.9115	1.0000
-3.1468	-1.9210	2.9543	1.0000

Figura 21 Generador Lineal

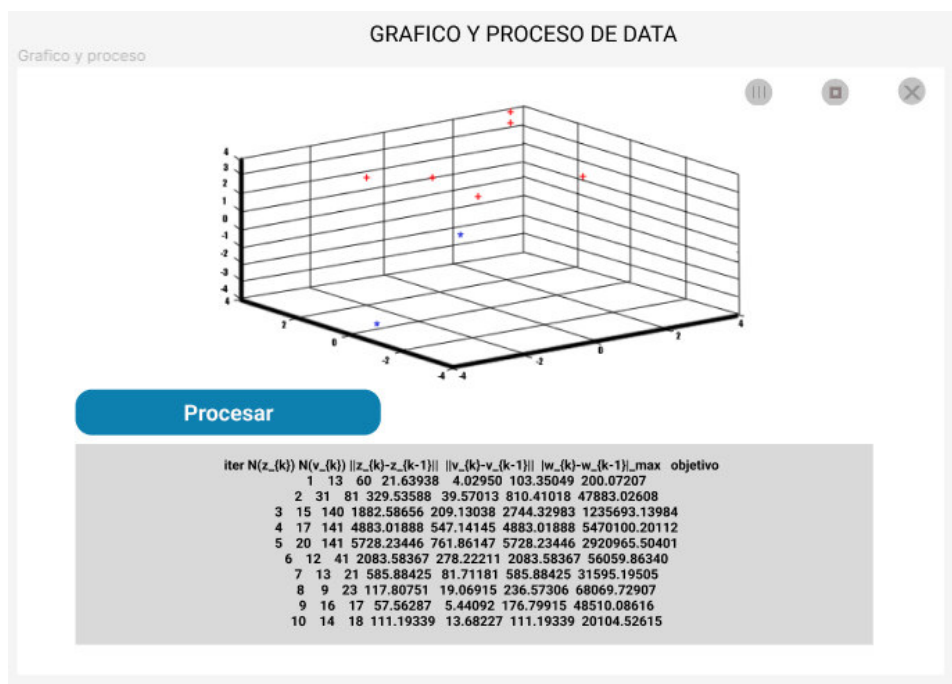


Figura 22 Gráfico de los Resultados procesados

En la siguiente figura podemos observar parte del proceso del Sprint 2 en el cual se obtiene la ecuación del plano y la gráfica que clasifica los datos.

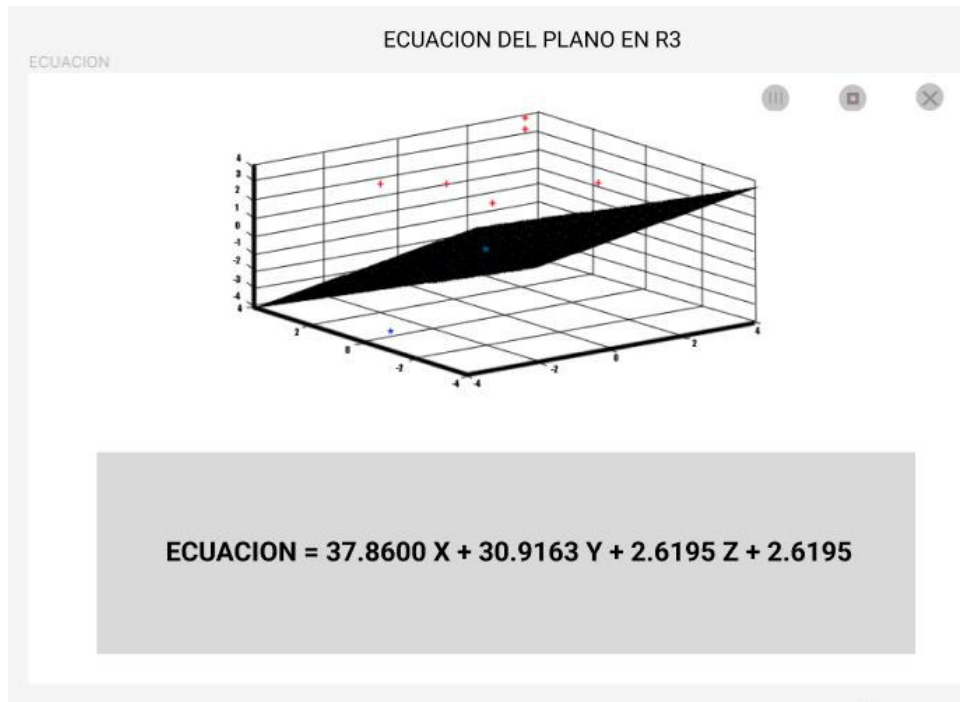


Figura 23 Resultados de la ecuación del plano

7.8.2. Desarrollo del modelo de la base de datos

Según (Silberschatz, Korth, & Sudarshan, 2002), este modelo de datos nos permitirá visualizar la estructura lógica de la base de datos, así como también todas sus interacciones y excepciones, además guardar la información y después visualizarlos. Como bien se sabe, gran parte de los modelos es posible representarlo mediante un diagrama acoplado a este. En la Figura 24, presentaremos nuestro modelo de base de datos obtenido posteriormente al recojo de requerimientos, luego de la reunión con nuestros Stakeholders.

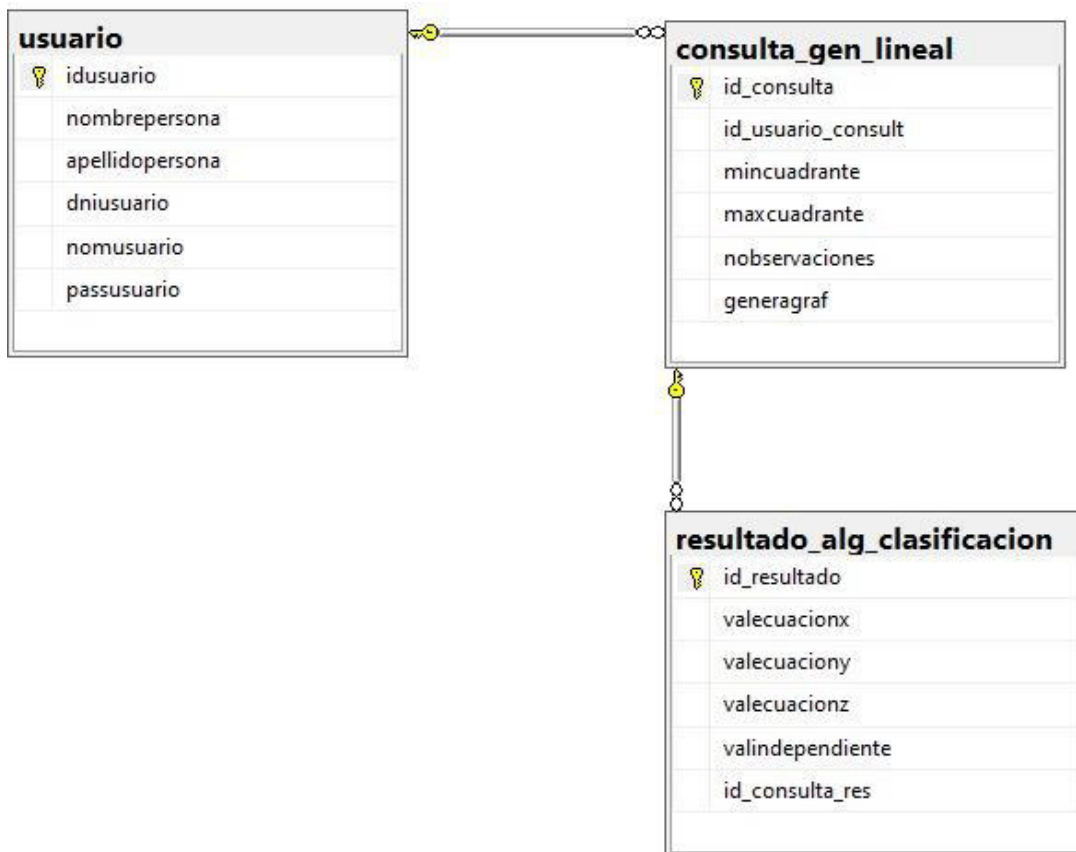


Figura 24 Modelo de Base de Datos del Sistema

7.9. Arquitectura del Software

La arquitectura del software es parte lógica funcional del software que ayudara a descifrar la estructura de un proceso del sistema. Para su complemento utilizaremos el patrón modelo, vista y controlador, este patrón trabaja en capaz donde el modelo tiene la función de a ver entender la parte lógica del negocio, la vista es todo que se presenta en la parte visual de un usuario para que pueda interactuar y realizar consultas o actividades de mandos definidos como botones o casillas. El controlador es quien tiene la comunicación con la capa modelo para interactuar en solicitudes de parte del usuario. Para una buena comunicación en este proyecto no enfocaremos en el uso de la herramienta SQL, que nos brinda una eficiente comunicación y desarrollo de base de datos.

La arquitectura MVC, nos facilitará el desarrollo del sistema web.

En la Figura 25 podemos ver la arquitectura del software y en la Figura 26 el diagrama del Hardware.

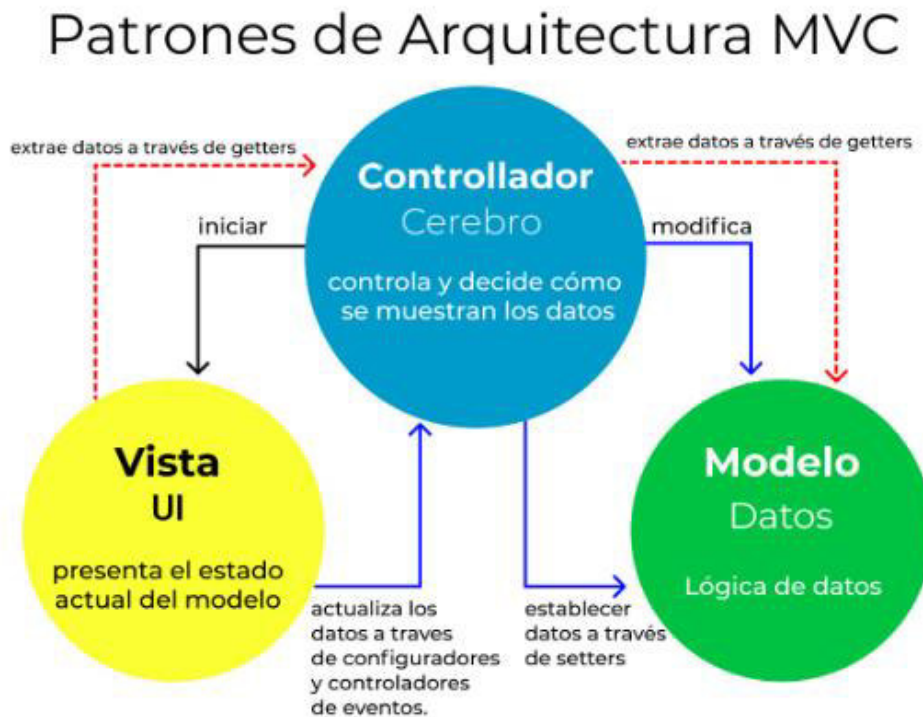


Figura 25 Arquitectura del Software

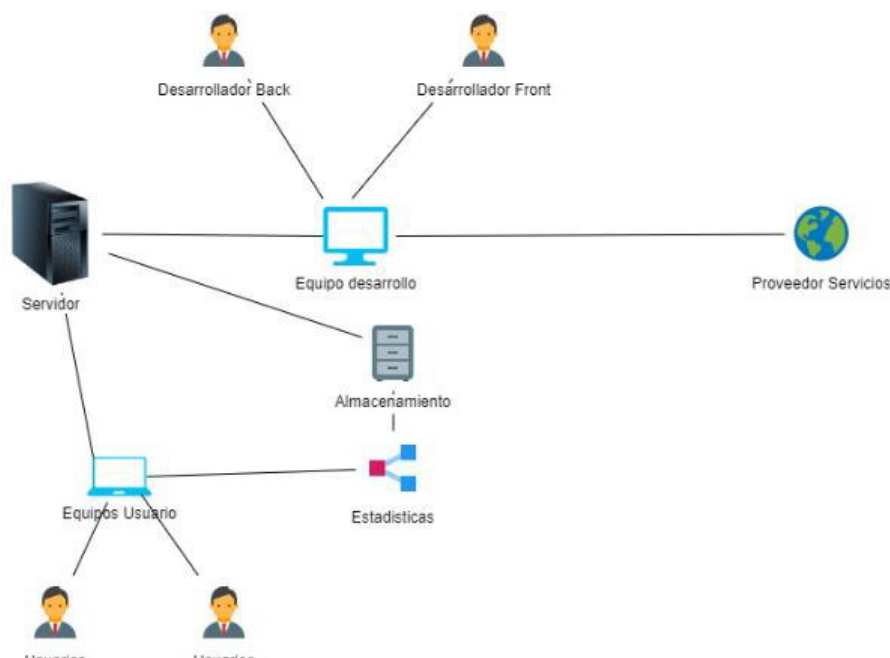


Figura 26 Diagrama del Hardware

CAPÍTULO VIII. RESULTADOS, CONCLUSIONES Y TRABAJOS FUTUROS

8.1. Resultados

8.1.1. Experimento Numérico

Veremos los resultados de un experimento numérico, para ello consideraremos los siguientes parámetros:

- $\lambda = 0.05, \rho = 1, tol = 10^{-3}$
- Máximo número de Iteraciones = 2000.
- Muestra _ 50 observaciones

Se puede observar de acuerdo con los resultados que la convergencia del valor óptimo es rápida al igual que z pero es más lenta con respecto al parámetro v . La convergencia de la función objetivo se refiere a la diferencia en valor absoluto entre la función objetivo $f + g$ obtenida por el SC-PMA y la función $f + g$ objetivo obtenido por la función CVX de Matlab.

- El tiempo transcurrido es 4930.763008 segundos
- Coeficientes de la ecuación son: 0.7259; -0.1075; -0.4545; 1.1248;
- Plano obtenido: $-0.7259X - 0.1075Y - 0.4545Z + 1.1248 = 0$
- Historial: historial de convergencia y valor óptimo

La historia es muy importante para obtener la Tabla 15 donde se muestra

- El número de iteraciones
- El número de iteraciones internas para resolver cada subproblema denotado por $N(z_{\{k\}}), N(v_{\{k\}})$, respectivamente.

- La diferencia entre puntos consecutivos de cada variable:

$$\|z_{\{k\}} - z_{\{k-1\}}\|, \|v_{\{k\}} - v_{\{k-1\}}\| \text{ y } \|w_{\{k\}} - w_{\{k-1\}}\|$$

- El valor de la función objetivo en cada iteración.

Tabla 15
Resultados Computacionales

Iteración	$N(z_{\{k\}})$	$N(v_{\{k\}})$	$\ z_{\{k\}} - z_{\{k-1\}}\ $	$\ v_{\{k\}} - v_{\{k-1\}}\ $	$\ w_{\{k\}} - w_{\{k-1\}}\ $	Función objetiva
1	12	29	2.73490	0.44568	3.48678	13.55714
2	10	22	1.78088	0.33687	2.36787	8.60897
3	9	29	2.31439	0.33125	2.31439	4.92598
4	10	29	1.37369	0.25792	1.37369	4.92598
5	8	28	0.34034	0.20416	1.01441	4.96926
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1999	6	29	0.00014	0.00265	0.00430	1.19902
2000	4	29	0.00013	0.00220	0.00423	1.19951

8.1.2. Análisis Comparativo entre Algoritmo Proximales.

Realizaremos una comparación del Algoritmo con distancias proximales:

- **Algoritmo Multiplicador Proximal con distancias proximales (PMAPD)**

Este algoritmo PMAPD fue desarrollado para un modelo de optimización con estructura separable, ver (Sarmiento, O.; Papa Quiroz, E.A.; Oliveira, P. R., 2015)

En este algoritmo propuesto puede resolver problemas de minimización convexa, además converge linealmente a una solución óptima del problema.

Bajo determinadas condiciones se verifica para una amplia clase de distancias proximales inducidas.

Con respecto al trabajo desarrollado en el algoritmo PMAPD, en esta tesis se extendió los resultados obtenidos para el caso de cono simétrico, es decir el trabajo desarrollado es más general.

Este algoritmo sus funciones f y g están definidas sobre $\mathbb{R}^n$, ver el algoritmo en (ANEXO B.1. (PMAPD))

- **Método de Direcciones Alternantes con Regularización Proximal Interior (RIPADM)**

Vemos en esta investigación que trabaja con distancias proximales, cuyo modelo de problema de optimización posee una estructura separable, para el desarrollo, utiliza un factor de relajación ρ , el cual está comprendido en cierto intervalo $\rho \in (0, \frac{\sqrt{5}+1}{2})$ y añade a la regla de actualización del multiplicador de Lagrange. Este algoritmo posee convergencia global.

Se encontró aplicaciones al modelo TWSVM

En este algoritmo las funciones f y g están definidas sobre el espacio $\mathbb{R}^n$ a diferencia del trabajo desarrollado por la tesis que trabaja sobre conos simétricos. Esta investigación fue desarrollada por (Ruiz Garrido, 2016) ver el algoritmo en el **ANEXO B.2. (RIPADM)**

- **Algoritmo Multiplicado Proximal Cono Simétrico (SC-PMA)**

Este algoritmo desarrollado en la presente Tesis, debido a las características de convexidad y de acuerdo con la hipótesis (A1-A2) posee una solución óptima y de acuerdo a la estricta convexidad se demostró que la convergencia es global.

Además de verificar algunas distancias proximales inducidas como; la distancia proximal homogénea, distancia de Kulback-Leiber bregman, entre otras.

Cabe mencionar que los algoritmos proximales poseen complejidad

$$\frac{C}{\varepsilon} (2xO(m\{2(n+p)+1\}) + 2xO(l))$$

8.1.2. Esquema final del Aporte de la Tesis

Después de la elaboración del algoritmo, probando los teoremas, lemas y viendo la convergencia del algoritmo se verifica que este algoritmo Multiplicador Proximal de Cono Simétrico para Optimización Separable, Aplicación a SVM para Clasificación Binaria.

En la Figura 27 mostramos el Esquema Final del aporte de Investigación, es decir el proceso desde la obtención de los datos, la clasificación de estos mediante el algoritmo y presentamos a través del diseño del sistema.

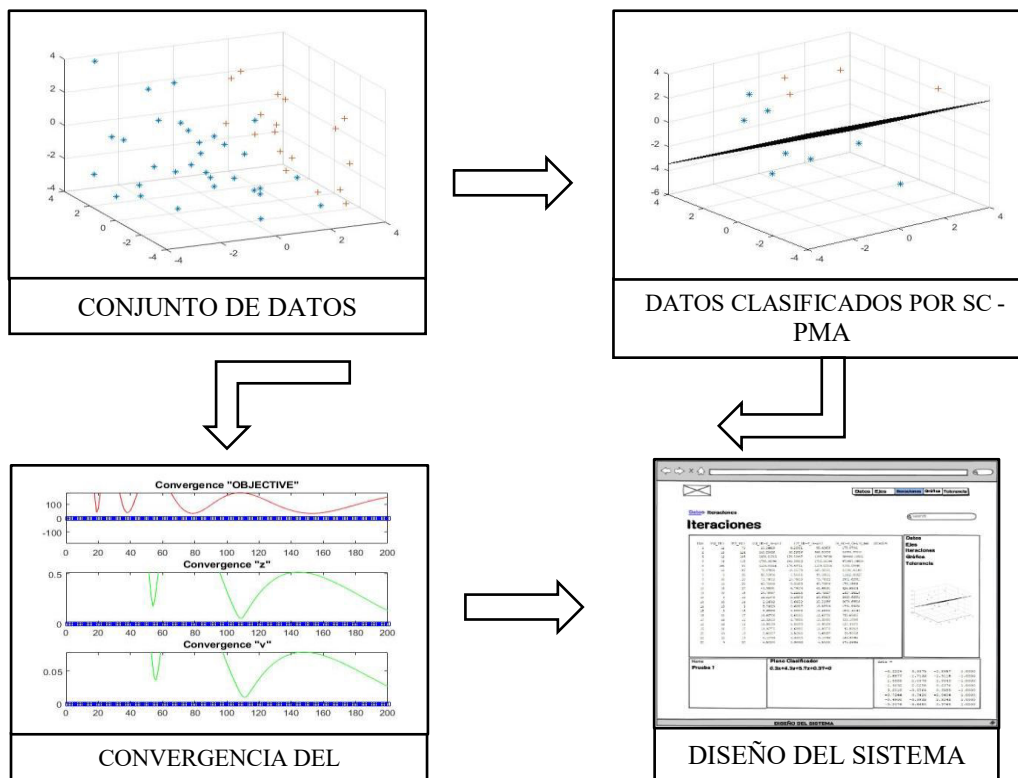


Figura 27 Esquema Final del aporte de Investigación.

Se tiene una nueva alternativa de clasificar datos con un algoritmo con fundamento teórico en Multiplicador Proximal en Cono Simétrico, donde una aplicación está dado para Máquina de Vectores de Soporte relacionado a clasificación de datos.

Después de realizar el fundamento teórico de su convergencia y posteriormente su implementación en el software MATLAB (R2017a) en una computadora de 8va Generación Intel® Core™ i5-8250U CPU, 1.60 GHz 1.80GHz, 4.00 GB, Windows 1064 bits.

Se realizó el diseño con el marco referencial SCRUM, cuya metodología se adecúa al diseño de software, se realizó su base de datos, su arquitectura del sistema y la presentación de las pantallas de diseño.

8.2. Conclusiones

El algoritmo estudiado en esta Tesis es un tipo de algoritmo determinístico, a diferencia de otros algoritmos que son estadísticos o aleatorios en donde los resultados son probabilísticos, es decir se aproxima a la solución o en un determinado número de pruebas los resultados no siempre es el óptimo.

En ese sentido no se puede hacer una comparación con este tipo de algoritmo SC – PMA en donde se analiza la convergencia, la buena definición, la complejidad y de acuerdo con las características de convexidad y estricta convexidad. La solución obtenida del problema es única y óptima, es por ello que este algoritmo converge a una solución global.

La métrica para medir este tipo de algoritmo está dada por:

- La convergencia.
- La velocidad de convergencia.
- La complejidad del algoritmo.

Se tienen algunas conclusiones adicionales que responden a nuestros objetivos.

- Se logró definir un algoritmo multiplicador proximal para resolver problemas de optimización convexa separable sobre conos simétricos, se verificó su convergencia.
- Se Realizó una aplicación del algoritmo a la solución de modelos de máquinas de vectores de soporte para clasificación de datos.
- Se implementó y validó el algoritmo para clasificación de datos sometiendo a experimentos computacionales y comparando su performance.
- Se diseñó el sistema del Algoritmo Multiplicador Proximal de Cono Simétrico para Optimización Separable: Aplicación a SVM para Clasificación Binaria.
- Por lo todo lo expuesto se concluye que el aporte de esta investigación es un algoritmo de cono simétrico para clasificar datos, un ejemplo de aplicación es a la máquina de vectores de soporte relacionados a clasificación.

8.3. Discusiones

En esta tesis se logró ampliar el método de descomposición proximal dado por (Chen & Teboulle, 1994) (Bonnans, Gilbert, Lemarechal, & Sagastiz'abal, 2006) para problemas de programación de conos simétricos convexos, se logró realizar una aplicación a un modelo de clasificación de máquinas de vectores de soporte (SVM).

La extensión realizada en esta Tesis se basa en las distancias proximales dado por (Auslender, A.; Teboulle, M., 2006) bajo determinadas hipótesis moderadas sobre distancias proximales se estableció que cada punto limite es un punto silla de Lagrangiana asociado al problema, además se obtuvo la convergencia global.

En la mayoría de los casos es imposible resolver los problemas (7) – (8). La familia de Método de Bundle es (Bonnans, Gilbert, Lemarechal, & Sagastiz'abal, 2006), es quizás la herramienta computacional más práctica para la optimización no diferenciable.

Con respecto al Algoritmo Multiplicador Proximal con Distancias Proximales (PMAPD) el cual tiene convergencia lineal, el método de esta Tesis se demostró que tiene una convergencia global, la cual es una solución del problema (1).

Con respecto al Método de Direcciones Alternantes con Regularización Proximal Interior (RIPADM) una característica fuerte es que posee un factor de relajación ρ que ayuda a la convergencia de este.

En nuestro trabajo no se considera un factor de relajación, sin embargo, encontramos la convergencia global de acuerdo a la buena definición.

8.4. Trabajos Futuros

Para trabajos futuros se puede considerar trabajar sobre una base de datos no linealmente separables, el cual se puede clasificar con determinados supuestos, otro tema importante sería mejorar la velocidad de convergencia del algoritmo y trabajar con otras distancias.

REFERENCIAS

- & Arbós, A. (2005). *Automatización de procesos mediante la guía GEMMA*. Barcelona: Edit. Edicions UPC.
- Aguilar. (2019). el patrón MVC en programación y por qué es útil. *campusmvp*.
- Ahonen, T., Hadid, A., & Pietikainen, M. (2004). Face recognition with local binary patterns. *European Conference on Computer Vision*, 469-481.
- Ahsan, M., Stoyanov, S., Bailey, C., & Albarbar, A. (2020). Developing Computational Intelligence for Smart Qualification Testing of Electronic Products. *IEEE Access*, 8, 16922-16933.
- Alizadeh, F., & Goldfarb, D. (2012). Second-order cone programming. *Mathematical Programming*, 95, 3-51.
- Almahdi, A. J., Yassen, A. J., & Dakhil, A. F. (2021). EEG signals analysis for epileptic seizure detection using DWT method with SVM and KNN classifiers. *Iraqi Journal of Science*, 54-62.
- Almobarak, A., Almohammadi, H., Aboalnaser, S., & Syed, L. (October-2019). Classification of aesthetic photographic images using SVM and KNN classifiers. *Proceedings. International Conference on Developments in eSystems Engineering, DeSE*, 170-175. doi:10.1109 / DeSE.2019.00040
- Alomari, O., Makhadmeh, S., Al-Betar, M., Alyasseri, Z., Doush, I., Abasi, A., . . . Zitar, R. (2021). Gene selection for microarray data classification based on Gray Wolf Optimizer enhanced with TRIZ-inspired operators. *Knowledge-Based Systems*, 223, 107034.
- Alvarez, F., López, J., & Ramírez, H. (2010). Interior proximal algorithm with variable metric for second-order cone programming: Applications to structural optimization and support vector machines. *Optimization Methods and Software*, 25(6), 859-881.
- Alvarez, F.; López, J.; Ramírez, C.H. (2012). Interior proximal algorithm with variable metric for second-order cone programming: applications to structural optimization and support vector machines. *Optimization Methods Software*, 25(6), 859-881.
- Aouani, H., & Ben Ayed, Y. (2021). Deep support vector machines for speech emotion recognition. *Advances in Intelligent Systems and Computing 1181 AISC*, 406-415.

- Aphinives, C., & Aphinives, P. (2021). Artificial intelligence development for detecting prostate cancer in MRI Egyptian Journal of Radiology and Nuclear Medicine. 52(1), 87.
- Arrow, K., & A.C. (1961). Enthoven, Quasi-concave Programming. *Econometria*, V, 779-800.
- Auslender, A., & Teboulle, M. (2006). Optimiz. *SIAM J*, 16(29).
- Auslender, A.; Teboulle, M. (2006). Interior gradient and proximal methods for convex and conic optimization. *SIAM Journal on Oprimization*, 16(3), 697-725.
- Azarmdel, H., Jahanbakhshi, A., Mohtasebi, S., & Muñoz, A. (2020). Evaluation of image processing technique as an expert system in mulberry fruit grading based on ripeness level using artificial neural networks (ANNs) and support vector machine (SVM). *Postharvest Biology and Technology*, 166, 111-201. doi:10.1016/j.postharvbio.2020.111201
- Bairagi, D. (2019). Proximal Support Vector Machine Classifier based on LMS Algorithm. *2019 International Conference on Innovative Trends and Advances in Engineering and Technology (ICITAET)*, 116-123. doi:10.1109/ICITAET47105.2019.9170231
- Baygin, N., Baygin, M., & Karakose, M. (2019). A SVM-PSO Classifier for Robot Motion in Environment with Obstacles. *2019 International Conference on Artificial Intelligence and Data Processing Symposium, IDAP 2019*. doi:10.1109/IDAP.2019.8875921
- Bento, G., Ferreira, O., & Melo, J. (2016). Iteration - Complexity of Gradient, Subgradient and Proximal Point Method on Riemannian Manifolds. *Journal of Optimization Theory and Applications* .
- Blanz, V., Schölkopf, B., Bühlhoff, H., Burges, C., Vapnik, V., & Vetter, T. (1996). Comparison of view-based object recognition algorithms using realistic 3D models. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1112, 251-256.
- Bonnans, J., Gilbert, J., Lemarechal, C., & Sagastiz'abal, C. (2006). Numerical Optimization: Theoretical and Practical Aspects. *Springer*.
- Bonneville, M., Meunier, J., & Bengio, Y. (1998). Support vector machines for improving the classification of brain PET images. *Proceedings of SPIE - The International Society for Optical Engineering*, 3338, 264-273. doi:10.1117/12.310900.
- Cano Lengua, M., & Quiroz, E. (2020). A Proximal Method to Solve Quasiconvex Non-differentiable Location Problems. *ACM International Conference Proceeding Series*, 98-104.

- Cano, M. A., & Papa, E. A. (2020). A Systematic Literature Review on Support Vector Machines Applied to Classification,. *Proceedings of the 2020 IEEE Engineering International Research Conference, EIRCON*. doi:10.1109/EIRCON51178.2020.9254028
- Carrasco, M., López, J., & Maldonado, S. (2019). Epsilon-nonparallel support vector regression. *Applied Intelligence*, 49(12), 4223-4236.
- Carrasco, M.; López, J.; Maldonado, S. (2016). A second-order cone programming formulation for nonparallel hyperplane support vector machine. *Expert Systems with Applications*, 54, 95-104.
- Chen , S., & Wu, X. (2016). Eigenvalue proximal support vector machine algorithm based on eigenvalue decomposition. *Journal of Electronics and Information Technology*, 38(3), 557-564.
- Chen, G., & Teboulle, M. (1994). A proximal-based decomposition method for convex minimization problems. 64(1-3), 81-101.
- Chen, W., Yang, K., Shao, Y., Chen, Y., Zhang, J., & Yao, J. (2019). A Trace Lasso Regularized Robust Nonparallel Proximal Support Vector Machine for Noisy Classification. *IEEE Access*, 7, 47171-47184.
- Cheng, Y., Fu, L., Luo, P., Ye, Q., Liu, F., & Zhu, W. (2020). Multi-view generalized support vector machine via mining the inherent relationship between views with applications to face and fire smoke recognition. *Knowledge-Based Systems*, 210, 106488.
- Cheng, Y., Yin, H., Ye, Q., Huang, P., Fu, L., Yang, Z., & Tian, Y. (2020). Improved multi-view GEPSVM via Inter-View Difference Maximization and Intra-view Agreement Minimization. *Neural Networks*, 125, 313-329.
- Cho, J. (2008). Issues and Challenges of agile software development with SCRUM. *Issues in Information Systems*, 9(2), 188-195.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Mach. Learn*, 20, 273-297.
- Couellan , N., & Trafalis, T. (2013). On-line SVM learning via an incremental primal-dual technique. *Optimization Methods and Software*, 28(2), 256-275.
- Doljansky, M., & Teboulle, M. (1998). An interior proximal algorithm and the exponential multiplier method for semidefinite programming. *SIAM Journal on Optimization*, 9(1), 1-13.
- Drosou , K., & Koukouvinos, C. (2017). Proximal support vector machine techniques on medical prediction outcome. *Journal of Applied Statistics*, 44(3), 533-553.

- Dufrenois, F., & Noyer, J. (2015). Generalized eigenvalue proximal support vector machines for outlier description. *Proceedings of the International Joint Conference on Neural Networks*. doi:10.1109/IJCNN.2015.7280343
- Ergen, T., & Kozat, S. (2020). A novel distributed anomaly detection algorithm based on support vector machines. *Digital Signal Processing: A Review Journal*, 99, 102657.
- Esteva, G. (2016). *Logica de Algoritmos- Un enfoque Practico- Programación en Lenguaje C*.
- Faraut, J., & Korányi, A. (1994). *Analysis on Symmetric Cones*. New.
- Franc, V., & Sonnenburg, S. (2008). Optimized cutting plane algorithm for support vector machines. *Proceedings of the 25th International Conference on Machine Learning*, 320-327. doi:10.1145/1390156.1390197
- Gala García, Y. (2013). *Algoritmos SVM para problemas sobre big data*. Universidad Autónoma de Madrid.
- Gayathri, S., Wise, D., C.J.W, Janani, V., Eleaswari, M., & Hema, S. (2020). Analyzing, Detecting and Automatic Classification of Different Stages of Brain Tumor Using Region Segmentation and Support Vector Machine. *Proceedings of the International Conference on Electronics and Sustainable Communication Systems, ICESC 2020*(9156057), 404-408. doi:10.1109/ICESC48915.2020.9156057
- Gheni, E., & Algelal, Z. (2020). Human face recognition methods based on principle component analysis (PCA), wavelet and support vector machine (SVM): a comparative study. *Indonesian Journal of Electrical Engineering and Computer Science*, 991-999.
- Goh, K., Wang, L., Yeow, A., Poh, H., Li, K., Yeow, J., & Tan, G. (2021). Artificial intelligence in sepsis early prediction and diagnosis using unstructured data in healthcare *Nature Communications*. 12(1), 711.
- Gotsopoulos, A., Saarimäki, H., Glerean, E., Jääskeläinen, L., Sams, M., Nummenmaa, L., & Lampinen, J. (2018). Reproducibility of importance extraction methods in neural network based fMRI classification. *NeuroImage*, 181, 44-54.
- Guarracino, M., Iripino, A., & Verde, R. (2010). Multiclass generalized eigenvalue proximal support vector machines. *CISIS 2010 - The 4th International Conference on Complex, Intelligent and Software Intensive Systems*(5447398), 25-32. doi:10.1109/CISIS.2010.162
- Guler, O. (1992). New proximal point proximal algorithms for convex minimization. *SIAM Journal on Optimization*(2), 649-664.

- Guo, N., Gui, W., Chen, W., Tian, X., Qui, W., Tian, Z., & Zhang, X. (2020). Using improved support vector regression to predict the transmitted energy consumption data by distributed wireless sensor network. *Eurasip Journal on Wireless Communications and Networking*(120). doi:10.1186/s13638-020-01729-x
- Hamza, S., & Ayed, Y. B. (2020). SVM for human identification using the ECG signal. *Procedia Computer Science*, 176, 430-439.
- Hou, Q., Liu, L., Zhen, L., & Jing, L. (2018). A novel projection nonparallel support vector machine for pattern classification. *Engineering Applications of Artificial Intelligence*, 75, 64-75.
- Hsieh, C., Chang, K., Lin, C., Keerthi, S., & Sundararajan, S. (2008). A dual coordinate descent method for large-scale linear SVM. *Proceedings of the 25th International Conference on Machine Learning*, 408-415. doi:10.1145/1390156.1390208.
- Hua , X., & Li, X. (2014). A novel twin support vector machines with manifold regularization. *Applied Mechanics and Materials*, 644-650, 1640-1643.
- International Conference on Robotics and Artificial Intelligence. (2021). *RoAI 2020 Journal of Physics: Conference Series*, 1831(1), 367.
- Jingfu , L. (2020). IOT security analysis of BDT-SVM multiclassification algorithm. *International Journal of Computers and Applications*.
- Kalidindi, K., Gottumukkala, P., & Davuluri, R. (2020). Derivative-based band clustering and multi-agent PSO optimization for optimal band selection of hyper-spectral images. *Journal of Supercomputing*, 76(8), 5873-5898.
- Kannai, Y. (1977). Concavifiability and Constructions of Concave Utility Functions. *Journal of Mathematical Economics*, V(4), 1-56.
- Keerthi , S., & DeCoste, D. (2005). A modified finite Newton method for fast solution of large scale linear SVMs. *Journal of Machine Learning Research*, 6, 341-361.
- Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Obtenido de https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf.
- Kivinen, J., Smola, A., & Williamson, R. (2004). Online learning with kernels. *IEEE Transactions on Signal Processing*, 52, 2165-2175.
- Kostoff, R. (2008). Literature-related discovery (LRD): introduction and background. *Technol. Forecast. Soc.*, 75(2), 186-202.
- Kumar , M., & Gopal, M. (2008). Application of smoothing technique on twin support vector machines. *Pattern Recognition Letters*, 29(13), 1842-1848.

- Le, D., Parvathy, V., & Gupta, D. (s.f.). IoT enabled depthwise separable convolution neural network with deep support vector machine for COVID-19 diagnosis and classification. *Int. J. Mach. Learn. & Cyber.* doi:10.1007/s13042-020-01248-7
- Li, A., & Tuncel, L. (2006). Some applications of symmetric cone programming in financial mathematics. *Transactions in Operational Research*(17), 1-19.
- Li, G., Yang, L., Wu, Z., & Wu, C. (2021). DC programming for sparse proximal support vector machines. *Information Sciences*, 547, 187-201.
- Li, K., Liu, Y., Wang, Q., Wu, Y., Song, S., Sun, Y., . . . Du, S. (2015). A spacecraft electrical characteristics multi-label classification method based on off-line FCM clustering and on-line WPSVM. *PLoS ONE*, 10(11).
- Li, K., Wu, Y., Song, S., Sun, Y., Wang, J., & Li, Y. (2017). A novel method for spacecraft electrical fault detection based on FCM clustering and WPSVM classification with PCA feature extraction. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 231(1), 98-108.
- Liang, J., Zhang, F., Xiong, X., Chen, X., & Lan, G. (2016). Manifold Regularized Proximal Support Vector Machine via Generalized Eigenvalue. *International Journal of Computational Intelligence Systems*, 9(6), 1041-1054.
- Liao, X., Huang, X., & Huang, W. (2019). ML-LUM: A system for land use mapping by machine learning algorithms. *Journal of Computer Languages*, 54, 100908.
- Lidula, N., Perera, N., & Rajapakse, A. (2009). Investigation of a fast islanding detection methodology using transient signals. *IEEE Power and Energy Society General Meeting*, (5275780). doi:10.1109/PES.2009.5275780.
- Liese, F., & Vajda, I. (1987). Convex statistical distance. *Teubner*.
- Lin, S. (2021). Application of Computer Artificial Intelligence in Circuit Logic Simulation *Journal of Physics: Conference Series*. 1856(1), 012071.
- Lobo, M., Vandenberghe, L., Boyd, S., & Lebre, H. (1998). Linear Algebra Appl. 284(37).
- Longjam, T., & Kisku, D. (2018-January). A supervised manipuri offline signature verification system with global and local features. *7th International Symposium on Embedded Computing and System Design, ISED 2017*, 1-6. doi:10.1109/ISED.2017.8303951
- López, J.; Papa Quiroz, E.A. (2017). Construction of proximal distances over symmetric cones. *Optimization*, 66(8), 1301-21.
- López, J., & Papa Quiroz, E. (2017). Optimization. 66(8), 21.
- López, J.; Maldonado, S.; Carrasco, S. (2011). A novel multi-class SVM model using second-order cone constraints. *Applied Intelligence*, 44(2), 457-469.

- Maldonado, S., López, J., Jimenez-Molina, A., & Lira, H. (2020). Simultaneous feature selection and heterogeneity control for SVM classification: An application to mental workload assessment. *Expert Systems with Applications*, 143, 112988.
- Maldonado, S.; López, J. (2014). Alternative second-order cone programming formulations for support vector classification. *Information Sciences*, 268, 328-341.
- Malik, H., & Chack, D. (2019). A novel intelligent transmission line fault diagnosis model based on EEMD and multiclass PSVM. *Advances in Intelligent Systems and Computing*, 698, 85-92.
- Mangasarian, O. (2006). Exact 1-norm support vector machines via unconstrained convex differentiate minimization. *Journal of Machine Learning Research*, 7, 1517-1530.
- Masoudi-Sobhanzadeh, Y., Motieghader, H., Omid, Y., & Masoudi-Nejad, A. (2021). A machine learning method based on the genetic and world competitive contests algorithms for selecting genes or features in biological applications. *Sci. Rep*, 11, 3349.
- Mendoza González, G. (2015). Herramienta de Desarrollo Netbeans. *Consultorjava*, 1-5.
- Menzinsky, A., López, G., Palacio, J., Sobrino, M., Álvarez, R., & Rivas, V. (2018). *Historia de Usuario. Ingeniería de Requisitos Ágil. SCRUM MANAGER*.
- Moosaei, H., Ketabchi, S., Razzaghi, M., & Tanveer, M. (2021). Generalized Twin Support Vector Machines. *Neural Processing Letters*, 53(2), 1545-1564.
- Ni, T., & Zhai, J. (2016). A matrix-free smoothing algorithm for large-scale support vector machines. *Information Sciences*, 358-359, 29-43.
- Nieto, D., Quiroz, E., & Cano Lengua, M. (2021). A systematic literature review on support vector machines applied to regression. *Proceedings of the 2021 IEEE Sciences and Humanities International Research Conference*.
- Niu, L. (2011). Parallel algorithm for training multiclass proximal Support Vector Machines. *Applied Mathematics and Computation*, 217(12), 5328-5337.
- Palomino Gonzales, J. C., Cano Lengua, M. A., & Andrade - Arenas, L. (2022). Multiplatform Mobile: Post - COVID - 19 Patient Management in Lima, Peru. *Seventh Sense Research Group*, 70(10). doi:10.14445/22315381/IJETT-V70I10P241
- Pan, S., & Chen, J. (2010). Nonlinear Anal Theory Appl. 73(18).
- Pan, S.; Chen, J.S. (2010). Interior proximal methods and central paths for convex second-order cone programming. *Nonlinear Analysis: Theory, Methods & Applications*, 73(9), 3083-3100.
- Panchal, P., Gopi, S., & Pradeepa, R. (2013). Incremental PSVM for underwater target classification with incorporation of new classes. *2013 Fourth International*

Conference on Computing, Communications and Networking Technologies (ICCCNT), 1-6. doi:10.1109/ICCCNT.2013.6726498

- Papa, E., López, J., & Cano, M. (2020). A Proximal Multiplier Method for Convex Separable Symmetric Cone Optimization. *ACM International Conference on Multimedia Systems and Signal, ICMSSP 2020*, 2, 92-97. doi:doi.org/10.1145/3404716.3404734
- Park, E., Chang, H., & Nam, H. (2017). Use of machine learning classifiers and sensor data to detect neurological deficit in stroke patients. *Journal of Medical Internet Research*, 19(4).
- Patil, A., Patil, S., & Prabhakar, M. (2019). Optimal classification of lung cancer related genes using enhanced reliefF algorithm and multiclass support vector machine. *International Journal of Innovative Technology and Exploring Engineering*, 8(10), 771-778.
- Patrascu, Andrei; Irofti, Paul;. (2022). Complexity of Inexact Proximal Point Algorithm for minimizing convex functions with Holderian Growth.
- Porter, A., & Newman, N. (2011). Mining external R&D. *Search Technology, Inc.*, 6025 The Corners Pkwy, Norcross, GA 30092, USA. *Technovation* , 171-176.
- Rai, P., Kumar, S., & Verma, D. (2021). Prediction of effort required to design software for smart city applications. *Journal of Physics: Conference Series*, 1714(1), 012033.
- Ramirez, A., Diestra, B., & Cano Lengua, M. (2021). Implementation of a virtual store to exponentiate the flow of product sales in a private company in the city of Lima. *Proceedings of the 2021 IEEE Engineering International Research Conference, EIRCON*. doi:10.1109/EIRCON52903.2021.9613146
- Razaque, A., Ben Haj Frej, M., Almiani, M., Alotaibi, M., & Alotaibi, B. (2021). Improved support vector machine enabled radial basis function and linear variants for remote sensing image classification. 21(13), 4431.
- Revathi, J., Anitha, J., Jude, J., & Hemanth, D. (2021). An intelligent medical decision support system for diagnosis of heart abnormalities in ECG signals. *Intelligent Decision Technologie*, 15(1), 19-31.
- Rockafellar, R. (1976). Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*(14), 877-898.
- Ruiz Garrido, N. S. (2016). *Contributions to the convergence theory and computational implementation of interior optimization methods for convex problems*. Chile: Universidad de Chile.

- Sahu, D., Kumar, A., & Kang, S. (2021). Proximal point algorithms based on Siterative technique for nearly asymptotically quasi-nonexpansive mappings and applications. *Numerical Algorithms*, 86(4), 1561-1590.
- Salazar, L., & Abad Londoño, J. H. (2018). *Historias de usuario: Una visión pragmática*.
- Sandeep, K., & Sagar, B. (2019). Brahmi character recognition based on SVM (support vector machine) classifier using image gradient features. *Journal of Discrete Mathematical Sciences and Cryptography*, 22(8), 1365-1381.
- Saravanan , N., & Ramachandran, K. (2009). A case study on classification of features by fast single-shot multiclass PSVM using Morlet wavelet for fault diagnosis of spur bevel gear box. *Expert Systems with Applications*, 36(8), 10854-10862.
- Sarmiento, O., Papa Quiroz, E., & Oliveira, P. (2017). Optimization. 66(20).
- Sarmiento, O.; Papa Quiroz, E.A.; Oliveira, P. R.; (2015). Converge rate of proximal multiplier algorithm for separable convex minimization. *Optimization*, 66(2), 251-270.
- SCOPUS. (2021). List of articles published on Support vector machines applied to classification. Retrieved September 19, 2021. Obtenido de <https://acortar.link/sT26TX>.
- Shao, Y., Chen, W., Zhang, J., Wang, Z., & Deng, N. (2014). An efficient weighted Lagrangian twin support vector machine for imbalanced data classification. *Pattern Recognition*, 47(9), 3158-3167.
- Sharifrazi, D., Alizadehsani, R., Roshanzamir, M., Joloudari, J., Shoeibi, A., Jafari, M., . . . Acharya, U. (2021). Fusion of convolution neural network, support vector machine and Sobel filter for accurate detection of COVID-19 patients using X-ray images Biomedical Signal Processing and Control. 68, 102622.
- Shi, H., Yu, Y., & Wang, Y. (2018). Early Warning Method for Sea Typhoons using Remote-Sensing Imagery Based on Improved Support Vector Machines (SVMs). *Journal of Coastal Research*, 82, 180-185.
- Silberschatz, A., Korth, H., & Sudarshan, S. (2002). *Fundamentos de Bases de Datos*. España: McGRAW - HILL.
- Silva, R., & Machado, A. (2021). Automatic measurement of pressure ulcers using Support Vector Machines and GrabCut. *Comput. Methods Programs Biomed*, 200, 105867.
- Sin, G., & Al, R. (2021). Activated sludge models at the crossroad of artificial intelligence—A perspective on advancing process modeling npj Clean Water. 4(1), 16.

- Singh, K., Gupta, I., & Gupta, S. (2010). Retrieval and classification of leaf shape by Support Vector Machine using binary Decision Tree, probabilistic neural network and generic Fourier Moment technique: A comparative study. *Proc. of the IADIS Int. Conf. - Computer Graphics, Visualization, Computer Vision and Image Processing, CGVCVIP 2010, Visual Commun., VC 2010, Web3DW 2010, Part of the MCCSIS 2010*, 412-417.
- Soman, K., Loganathan, R., Vijaya, M., Ajay , V., & Shivsubramani, K. (2007). Fast single-shot multiclass proximal support vector machines and perceptrons. *2007 International Conference on Computing: Theory and Applications (ICCTA '07)*, 294-298. doi:10.1109/ICCTA.2007.60
- Sun, S., Xie, X., & Dong, C. (2019). Multiview Learning With Generalized Eigenvalue Proximal Support Vector Machines. *IEEE Transactions on Cybernetics*, 49(2), 688-697.
- Suresh, K. (2021). Classification Study and Prediction of Cervical Cancer. *Advances in Intelligent Systems and Computing*, 1133, 329-347.
- Tamilselvi, S., Saravana Kumar, N., Lavanya, S., Bindhu, J., & Kaviyavarshini, N. (2021). Artificial intelligence for a bio-sensored detection of tuberculosis Network Modeling Analysis in Health Informatics and Bioinformatics. *10*(25).
- Tan, J., Zhen, L., Deng , N., & Zhang, Z. (2014). Laplacian p-norm proximal support vector machine for semi-supervised classification. *Neurocomputing*, 144, 151-158.
- Teboulle, M. (1997). Convergence of proximal - like algorithms. *SIAM Journal on Optimization*, 7(4), 1069 - 1083.
- Teo, C., Smola, A., Vishwanathan, S., & Le, Q. (2007). A scalable modular convex solver for regularized risk minimization. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 727-736. doi:10.1145/1281192.1281270
- Tikka, S., Singh, B., Nizamie, S., Garg, S., Mandal, S., Thakur, K., & Singh, L. (2020). Artificial intelligence-based classification of schizophrenia: A high density electroencephalographic and support vector machine study. *Indian Journal of Psychiatry*, 62(3), 273-282. doi:10.4103/psychiatry.IndianJPsychiatry_91_20
- Tikka, S., Singh, B., Nizamie, S., Garg, S., Mandal, S., Thakur, K., & Singh, L. (2020). Artificial intelligence-based classification of schizophrenia: A high density electroencephalographic and support vector machine study. *Indian Journal of Psychiatry*, 62(3), 273-282. doi:10.4103/psychiatry.IndianJPsychiatry_91_20
- Vandenbergh, L., & Boyd, S. (1994). Semidefinite programming. *SIAM Review*(38), 49-95.

- Vargas, P., & Mauricio, D. (2018). A review of literature about models and factors of productivity in the software factory. *International Journal of Information Technologies and Systems Approach*, 11(1), 48-71.
- Verma, M., & Shukla, K. (2020). Convergence analysis of accelerated proximal extra gradient method with applications Neurocomputing. 388, 288-300.
- Vlietland, J., & Van Vliet, H. (2015). Towards a governance framework for chains of Scrum teams. *Information and Software Technology*, 57(1), 52-65.
- Wang, J., Zhang, M., Mei, Y., Lu, K., & Chen, X. (2021). Study on Inversion Amplitude of Internal Solitary Waves Applied to Shallow Sea in the Laboratory. *IEEE Geoscience and Remote Sensing Letters*, 18(4), 577-581.
- Wang, L., Yan, Y., Wang, X., & Wang, T. (2017). Input variable selection for data-driven models of Coriolis flowmeters for two-phase flow measurement. *Measurement Science and Technology*, 8(3), 035305. doi:10.1088/1361-6501/aa57d6.
- Wang, S., Phillips, P., Liu, A., & Du, S. (2017). Tea category identification using computer vision and generalized eigenvalue proximal SVM. *Fundamenta Informaticae*, 151(1-4), 325-339.
- Wong, L., Mauricio, D., & Rodriguez, G. (2017). A systematic literature review about software requirements elicitation. *Journal of Engineering Science and Technology*, 12(2), 296-317.
- Wu, B., Yu, H., & Chen, X. (2017). New applications of image classification in character recognition. *Proceedings of the 2017 12th International Conference on Intelligent Systems and Knowledge Engineering, ISKE 2017, 2018-January*. doi:10.1109/ISKE.2017.8258827
- Wu, S., & Pham, V. (2020). Lost-min voting strategies for speeding up multi-SVMs. *Proceedings of 2019 the 9th International Workshop on Computer Science and Engineering*.
- Xu, S., Zhao, Q., Yin, K., Zhang, F., Liu, D., & Yang, G. (2019). Combining random forest and support vector machines for object-based rural-land-cover classification using high spatial resolution imagery. *Journal of Applied Remote Sensing*, 13(1), 014521.
- Yan, M., & Wang, J. (2013). Classification algorithm of regression support vector machine and its application to enhance antifreeze heat transfer capability in ground source heat pump system. *Journal of Theoretical and Applied Information Technology*, 50(1), 182-188.
- Yang, X., & Chen, S. (2006). Proximal support vector machine based on prototypal multiclassification hyperplanes. *Computer Research and Development*, 43(10), 1700-1705.

- Yang, H., Qu, C., Cai, J., Zhang, S., & Zhao, X. (2020). SVM-Lattice: A recognition and evaluation frame for double-peaked profiles. *IEEE access*, 08, 80978-80996.
- Yang, Z. (2013). Nonparallel hyperplanes proximal classifiers based on manifold regularization for labeled and unlabeled examples. *International Journal of Pattern Recognition and Artificial Intelligence*, 27(5), 1350015.
- Ye, Y., Liu, S., & Li, J. (2008). A multiclass machine learning approach to credit rating prediction. *2008 International Symposium on Information Processing*, 57-61. doi:10.1109/ISIP.2008.37
- Young, A., Fernandez, K., Pfau, J., Reddy, R., Cao, N., von Franque, M., . . . Wei, M. (2021). Stress testing reveals gaps in clinic readiness of image-based diagnostic artificial intelligence models *npj Digital Medicine*. 4(1), 10.
- Zhao, T., Jayaram, V., Marfurt, K., & Zhou, H. (2014). Lithofacies classification in Barnett Shale using proximal support vector machines. *Society of Exploration Geophysicists International Exposition and 84th Annual Meeting SEG 2014*, 1714-1719. doi:10.1190/segam2014-1210.1
- Zheng, X., Zhang, L., & Yan, L. (2019). CTSVM: A robust twin support vector machine with correntropy-induced loss function for binary classification problems. *Information Sciences*, 559, 22-45.
- Zhou, A., Zhu, Z., & Fan, H. (2012). new semi-supervised PSVM classifier. *Applied Mathematics and Computation*, 219(8), 4006-4012.
- Zhu, G., Huang, D., Zhang, P., & Ban, W. (2015). ϵ -Proximal support vector machine for binary classification and its application in vehicle recognition. *Neurocomputing*, 161, 260-266.
- Zhu, X., Zhang, Q., & Qiao, S. (2020). A New Class Weighted Algorithm for PSVM. *Journal of Physics: Conference Series*, 1624, 042006.

ANEXOS

ANEXOS A. Implementaciones

ANEXO A.1. Función Principal

```
function main

%-----
% ParÃ ;metros a controlar para el generador de datos
%-----clc-----
%
%   clc;
%   n_var = 3;
%   n_obs = 10;
%   minimo = -4;
%   maximo = 4;
%   GRAF = 1;
%   data = gen_lineal_datos(n_var, n_obs,minimo,maximo,GRAF);

% Elegir la data a cargar, el minimo y el maximo con el que creaste
la
% data
load('data_333.mat');
minimo = -4;
maximo = 4;
%-----
% ParÃ;metros a controlar para el algoritmo proximal
%-----

[z, v,history] = svm_lin_SC_PMA(data,1,0.09,1,4,200,10^-3,1,0);

disp(z);

%-----
% ParÃ;metros a controlar para el grÃfico
%-----
[nrow, ncol] = size(data);
data_neg = data(data(:,ncol)==1,:);
data_posi = data(data(:,ncol)==-1,:);

if ncol > 2 && ncol < 5
```

```

figure (3);
if ncol == 3
    plot(data_neg(:,1),data_neg(:,2),'*')
    hold on;
    plot(data_posi(:,1),data_posi(:,2),'+');
    hold on;
    x = minimo:0.1:maximo;
    y = (-z(3)/z(2))-(z(1)/z(2))*x;
    plot(x,y,'b')
    grid on
    xlim([minimo,maximo])
    ylim([minimo,maximo])

else
    scatter3(data_neg(:,1),data_neg(:,2),data_neg(:,3),'');
    hold on;
    scatter3(data_posi(:,1),data_posi(:,2),data_posi(:,3),'');
    hold on;
    [x,y] = meshgrid(minimo:0.1:maximo,minimo:0.1:maximo);
    z = -(z(4)/z(3))-(z(1)/z(3))*x-(z(2)/z(3))*y;
    surf(x,y,z,'FaceAlpha',0.5);
end

else
    disp("NO SE PUEDE VISUALIZAR LOS DATOS EN DIMENSIÃN MAYOR A
3");
end

end

```

ANEXO A.2. Generador Lineal

```
function data = gen_lineal_datos(n_var,n_obs,minimo,maximo,GRAF)
#####
% Genera datos que son separables linealmente en "n" dim.
#####
%-----
% INPUT
%-----
% n_var   : Numero de variables
% n_obs   : Numero de observaciones
% minimo  : M nimo valor de los cuadrantes
% maximo  : M ximo valos de los cuadrantes
% GRAF    : (1) para graficar y (0) para no graficar
%-----
% OUTPUT
%-----
% data    : Data generada que es separable linealmente
%-----

clc;

data = minimo + (maximo-minimo)*rand(n_obs,n_var);
[nrow,ncol] = size(data);
coef = round(-10+(100+10)*rand(1,ncol+1));
data = [data zeros(nrow,1)];

for i = 1:nrow
    if sum(coef(1:ncol).*(data(i,1:ncol))) > coef(ncol+1)
        data(i,ncol+1) = -1;
    else
        data(i,ncol+1) = 1;
    end
end

if GRAF == 1
    if ncol > 1 && ncol < 4
        figure(1);
        data_neg = data(data(:,ncol+1)==1,:);
        data_posi = data(data(:,ncol+1)==-1,:);
        if ncol == 2
            plot(data_neg(:,1),data_neg(:,2),'*')
            hold on;
            plot(data_posi(:,1),data_posi(:,2),'+')
            hold on;
            x = minimo:0.1:maximo;
            y = (coef(3)/coef(2))-(coef(1)/coef(2))*x;
            plot(x,y,'b')
            grid on
            xlim([minimo,maximo])
            ylim([minimo,maximo])
        else

```

```

scatter3(data_neg(:,1),data_neg(:,2),data_neg(:,3),"*");
        hold on;

scatter3(data_posi(:,1),data_posi(:,2),data_posi(:,3),"+");
        hold on;
%
% [x,y] =
meshgrid(minimo:0.1:maximo,minimo:0.1:maximo);
%
% z = (coef(4)/coef(3))-(coef(1)/coef(3))*x-
(coef(2)/coef(3))*y;
%
% scatter3(x,y,z);
% surf(x,y,z,'FaceAlpha',0.5);
        end

    else
        disp("NO SE PUEDE VISUALIZAR LOS DATOS EN DIMENSIÃ"N MAYOR
A 3");
    end
end
end
end

```

ANEXO A.3. CVX_SVM (DATA)

```
function [z,v,cvx_optval,cvx_status,cvx_toc] = CVX_svm(data)

    %clc;

    [m,n] = size(data);
    Y = data(:,n);
    Diag_Y = diag(Y);
    X = [data(:,1:(n-1)) ones(m,1)];
    A = Diag_Y*X;
    B = [eye(m) -eye(m)];
    b = ones(m,1);
    C = 0.1;

    tic
    cvx_begin quiet
        %cvx_precision low
        variables z1(n-1) z2(1) v1(m) v2(m);
        obj = 0.5*(z1'*z1)+C*sum(v1);
        minimize(obj);
        subject to
            A*[z1;z2]+B*[v1;v2] == b;
            v1 >= 0;
            v2 >= 0;
    cvx_end
    z = [z1;z2];
    v = [v1;v2];
    cvx_toc = toc;
```

ANEXO A.4. SVM_LIN_SC_PMA

```
function [z,v,history] =
svm_lin_SC_PMA(data,rho,lambda,op_dis_f,op_dis_g,MAX_ITER,TOL,ITER_CON,
GRAF_CON)
%clc;
#####
%Minimiza la funci3n: f(x)+g(z) s.t. Ax+Bz=b para
#####
%-----
% INPUT
%-----
% data      : Data a separar linealmente
% MAX_ITER  : N3mero m3ximo de iteraciones
% GRAF_CON  : (1) para graficar y (0) para no graficar
% ITER_CON  : (1) para mostrar conv. y (0) para no mostrar
% rho       : Par3metro de aceleraci3n
% lambda    : Tama3o del paso en cada iteraci3n
% op_dis_f  : Distancia proximal para f
% op_dis_g  : distancia proximal para g
% op_dis_x  : Tipo de distancia proximal
% 1 --> Euclidean
% 2 --> Kullback-Leibler divergence
% 3 --> Itakura-Saito distance
% 4 --> Second-order homogeneous
%
%-----
% OUTPUT
%-----
% z         : Vector soluci3n del hiperplano separador
% v         : Vector soluci3n de las tolerancias
% history   : Historial de convergencia y valor 3ptimo
%-----
if GRAF_CON==1
    [z_cvx,v_cvx,cvx_optval,cvx_status,cvx_toc] = CVX_svm(data);
    t = 1:1:MAX_ITER;
    cont = 1;
    figure;
    subplot(3,1,1)
    grid on
    plot(t,cvx_optval,'--sb');
    title('Convergence "OBJECTIVE"')
    hold on

    subplot(3,1,2)
    plot(t,0,'--sb');
    title('Convergence "z"')
    hold on

    subplot(3,1,3)
    plot(t,0,'--sb');
    title('Convergence "v"')
    hold on
```

```

end
%-----

[m,n] = size(data);
Y = data(:,n);
Diag_Y = diag(Y);
X = [data(:,1:(n-1)) ones(m,1)];
A = Diag_Y*X;
B = [eye(m) -eye(m)];
b = ones(m,1);
C = 0.1;

z0 = ones(size(A,2),1);
v0 = ones(size(B,2),1);
y0 = ones(length(b),1);

%-----
%Introducimos condiciones iniciales para el algoritmo
%-----
theta1 = 1; theta2 = 1; gama1 = 1; gama2 = 1;
options = optimoptions('fmincon','Algorithm','interior-
point','Display','off'); % run interior-point algorithm

if ITER_CON==1
    fprintf('%3s\t%3s\t%3s\t%14s\t\t%14s\t\t%10s\t%10s\n', 'iter',
...
    'N(z_{k})', 'N(v_{k})', '||z_{k}-z_{k-1}||', '||v_{k}-v_{k-
1}||', ...
    '|w_{k}-w_{k-1}|_max','objetivo');
end

t_start = tic;
for k = 1:MAX_ITER

    %Calculamos "p"
    %*****
    lambda=0.5/k;
    p = y0 + lambda*(A*z0 + B*v0 - b);

    %Minimizamos z_k
    %*****
    [z,FVAL,EXITFLAG,OUTPUT_z] = fmincon(@(z) (f(z,n)+
myfun_F(z,z0,A,p,lambda,theta1,op_dis_f)),z0,[],[],[],[],[],[],[],optio
ns);

    %Minimizamos v_k
    %*****
    [v,FVAL,EXITFLAG,OUTPUT_v] = fmincon(@(v) (g(v,C,m)+
myfun_G(v,v0,B,p,lambda,theta2,op_dis_g)),v0,[],[],[],[],zeros(2*m,1),[
],[],options);

    %Actualizamos "y_k"
    %*****
    y = y0 + lambda*rho*(A*z + B*v - b);

```

```

% Cargamos el historial
% *****
history.iter_z(k) = OUTPUT_z.iterations;
history.iter_v(k) = OUTPUT_v.iterations;
history.res_z(k) = norm(z-z0,2);
history.res_v(k) = norm(v-v0,2);
history.res_max(k) = max([norm(z-z0,2),norm(v-v0,2),norm(y-
y0,2)]);
history.objval(k) = f(z,n)+g(v,C,m);
tole = max([norm(z-z0,2),norm(v-v0,2),norm(y-y0,2)]);

%Escribimos el historial
%*****
if ITER_CON==1

fprintf('%4d\t\t%4d\t\t%4d\t\t%9.5f\t\t%9.5f\t\t%9.5f\t\t%9.5f\n',...
        k, history.iter_z(k), history.iter_v(k), ...
        history.res_z(k),history.res_v(k), ...
        history.res_max(k), history.objval(k));
        %figure(5);
        %plot(k,history.res_max(k),'*r')
        %hold on;

end

% Checkeamos el criterio de finalizaci3n
% *****
if (tole < TOL)
    disp("llegue");
    break;
end

% Actualizamos las variables
% *****
vaux = v; z0 = z; v0 = v; y0 = y;

if GRAF_CON==1
    if k>100
        subplot(3,1,1)
        plot(t(1:k),history.objval(1:k),'-r');
        ylim([cvx_optval-max(history.objval((k-100):k))
(cvx_optval+max(history.objval((k-100):k)))]);
        %legend('iteraci3n','sol_real')
        drawnow;

        hold on
        subplot(3,1,2)
        plot(t(1:k),history.res_z(1:k),'-g');
        ylim([0 max(history.res_z((k-100):k))]);
        drawnow;

        hold on
        subplot(3,1,3)
        plot(t(1:k),history.res_v(1:k),'-g');
        ylim([0 max(history.res_v((k-100):k))]);
        drawnow;
        cont = cont + 1;
    end
end

```

```

else
    subplot(3,1,1)
    plot(t(1:k),history.objval(1:k),'-r');
    drawnow;

    hold on
    subplot(3,1,2)
    plot(t(1:k),history.res_z(1:k),'-g');
    drawnow;

    hold on
    subplot(3,1,3)
    plot(t(1:k),history.res_v(1:k),'-g');
    drawnow;
end

end

end

toc(t_start);
end

function f = f(z,n)
    z1 = z(1:(n-1));
    f = 0.5*(z1'*z1);
end

function g = g(v,C,m)
    v1 = v(1:m);
    g = C*sum(v1);
end

function F = myfun_F(z,z0,A,p,lambda,theta,op_dis)
    F = p'*A*z + (1/lambda)*distancia(z,z0,theta,op_dis);
end

function G = myfun_G(v,v0,B,p,lambda,theta,op_dis)
    G = p'*B*v + (1/lambda)*distancia(v,v0,theta,op_dis);
end

function d = distancia(x,y,theta,op_dis)
    n = length(x);
    if op_dis == 1 % Euclidean (vale para Rn)
        d0 = 0;
    end

    if op_dis == 2 % Kullback-Leibler divergence (Vale para Octante
positivo)
        d0 = sum(x.*log(x./y)-x+y);
        if d0 == -Inf || isnan(d0)
            d0 = Inf;
        end
    end

    if op_dis == 3 % Itakura-Saito distance (Vale para Octante

```

```

positivo)
    d0 = sum(x./y - log(x./y) - ones(n,1));
    if d0 == -Inf || isnan(d0)
        d0 = Inf;
    end
end

    if op_dis == 4 % Second-order homogeneous (Vale para Octante
positivo)
        sigma = 0.001; v = 0.01;
        d0 = sum((v/2)*((x-y).*(x-y))+sigma*((y.*y).*log(y./x) + x.*y -
y.*y));
        if d0 == -Inf || isnan(d0)
            d0 = Inf;
        end
    end

    d = d0 + (theta/2)*norm(x-y,2)^2;
end

```

ANEXO B. Algoritmo Proxiales

ANEXO B.1. (PMAPD)

Algoritmo Multiplicador Proximal con Distancias Proxiales (PMAPD)

Paso 0. Elegir dos pares $(d_0, H) \in \mathcal{F}_+(\bar{C})$, $(d'_0, H'_0) \in \mathcal{F}_+(\bar{K})$ que satisfagan la condición (Iviii) y defina (d, H) , (d', H') dadas por (1) – (2) y (3) – (4) respectivamente.

Tomar tres sucesiones $a_k \geq 0$, $b_k \geq 0$ y $\lambda_k > 0$ y elija un punto de partida arbitrario $(x^0, z^0, y^0) \in C \times K \times \mathbb{R}^m$.

Paso 1. Para $k = 0, 1, 2, \dots$, calcule $p^{k+1} \in \mathbb{R}^m$ por

$$p^{k+1} = y^k + \lambda_k(Ax^k + Bz^k - b).$$

Paso 2. Encontrar $(x^{k+1}, v^{k+1}) \in C \times \mathbb{R}^n$ y $(z^{k+1}, \xi^{k+1}) \in K \times \mathbb{R}^p$ tal que

$$\begin{aligned} v^{k+1} &\in \partial_{a_k} f^k(x^{k+1}), & v^{k+1} + \lambda_k^{-1} \nabla_1 d(x^{k+1}, x^k) &= 0 \\ \xi^{k+1} &\in \partial_{b_k} g^k(z^{k+1}), & \xi^{k+1} + \lambda_k^{-1} \nabla_1 d'(z^{k+1}, z^k) &= 0 \end{aligned}$$

Donde las funciones $f^k: \mathbb{R}^n \rightarrow (-\infty, +\infty]$ y $g^k: \mathbb{R}^p \rightarrow (-\infty, +\infty]$ se define por $f^k(x) = f(x) + \langle p^{k+1}, Ax \rangle$ y $g^k(z) = g(z) + \langle p^{k+1}, Bz \rangle$, respectivamente.

Paso 3. Calcular

$$y^{k+1} = y^k + \lambda_k(Ax^{k+1} + Bz^{k+1} - b).$$

Criterio de parada:

Si $x^{k+1} = x^k$, $z^{k+1} = z^k$ y $y^{k+1} = y^k$ luego se detiene. De lo contrario a $k := k + 1$, y regreso al Paso 1.

ANEXO B.2. (RIPADM)

Algoritmo del Método de Direcciones Alternantes con Regularización Proximal Interior inexacto

Considere el problema (P)

$$\min_{x,z} \{f(x) + g(z) | Ax + Bz = b, x \in \bar{C}\}, \quad (P)$$

Y suponga que $\text{dom}(f) \cap C \neq \emptyset$. Sea $d \in \mathcal{D}(C)$ y λ un escalar positivo. A partir de un punto $(x^0, z^0, y^0) \in C \times \mathbb{R}^m \times \mathbb{R}^p$, generamos la secuencia $\{(x^k, z^k, y^k)\} \subset C \times \mathbb{R}^m \times \mathbb{R}^p$, y secuencias de errores $\{a^k\}$ y $\{b^k\}$ a través de los siguientes pasos:

Paso 1. Encontrar $(x^{k+1}, a^{k+1}) \in C \times \mathbb{R}^n$ con $r^{k+1} \in \partial f(x^{k+1})$ resolviendo:

$$a^{k+1} = r^{k+1} + A^t[y^k + \lambda(Ax^{k+1} + Bz^k - b)] + \frac{1}{2\lambda} \nabla_1 d(x^{k+1}, x^k)$$

Donde el error a^{k+1} satisface las condiciones

$$\|a^{k+1}\| \leq \varepsilon_{k+1}$$

$$\|a^{k+1}\| \|x^{k+1}\| \leq \eta_{k+1}$$

Paso 2. Encontrar $(z^{k+1}, b^{k+1}) \in \mathbb{R}^m \times \mathbb{R}^m$ con $\bar{r}^{k+1} \in \partial g(z^{k+1})$ resolviendo:

$$b^{k+1} = \bar{r}^{k+1} + B^t[y^k + \lambda(Ax^{k+1} + Bz^{k+1} - b)] + \frac{1}{\lambda}(z^{k+1} - z^k)$$

Donde el error b^{k+1} satisface las condiciones

$$\|b^{k+1}\| \leq \bar{\varepsilon}_{k+1}$$

$$\|b^{k+1}\| \|z^{k+1}\| \leq \bar{\eta}_{k+1}$$

Paso 3. Calcular

$$y^{k+1} = y^k + \rho\lambda(Ax^{k+1} + Bz^{k+1} - b)$$

Los escalares positivos $\varepsilon_k, \eta_k, \bar{\varepsilon}_k, \bar{\eta}_k > 0, \rho\lambda \in \left(0, \frac{\sqrt{5}+1}{2}\right)$ cumplen las siguientes condiciones:

$$\sum_{k=1}^{\infty} \varepsilon_k < \infty, \sum_{k=1}^{\infty} \eta_k < \infty, \sum_{k=1}^{\infty} \bar{\varepsilon}_k < \infty, y \sum_{k=1}^{\infty} \bar{\eta}_k < \infty$$

ANEXO C. Aplicación del Problema (P)

Mostramos otro ejemplo del problema (4.1) en inversa dispersa selección de covarianza.

Selección de covarianza inversa dispersa (SICS): Gráfico gaussiano modelos son de cierto interés en el aprendizaje estadístico [10], y están modelados por el siguiente problema de optimización convexa:

$$\min_{X \in \mathcal{S}_+^n} \{\langle \mathcal{S}, X \rangle - \text{Indet}(X) + \rho \|X\|_1 : X \in \mathcal{S}_+^n\}$$

Donde $\rho > 0$, $\mathcal{S} \in \mathcal{S}_+^n$ denota la matriz de covarianza empírica y $\|X\|_1$ la l_1 -norma de la matriz X definida por

$$\|X\|_1 := \sum_{i=1}^n \sum_{j=1}^m |x_{ij}|.$$

Definición

$$f(X) = \langle \mathcal{S}, X \rangle - \text{Indet}(X) \text{ y } g(X) = \rho \|X\|_1,$$

Obtenemos $\min_{X \in \mathcal{S}_+^n} \{f(X) + g(X) : X \in \mathcal{S}_+^n\}$, se puede reescribir como:

$$\min_{X \in \mathcal{S}_+^n} \{f(X) + g(X) : X - Y = 0, X, Y \in \mathcal{S}_+^n\}$$

Este modelo es un caso especial de (4.1) cuando $\mathbb{V}_1 = \mathbb{V}_2 = \mathcal{S}^n$, $k_1 = k_2 = \mathcal{S}_+^n$, y

$\mathbb{A}, \mathbb{B}: \mathcal{S}^n \rightarrow \mathfrak{R}^{\bar{m}}$, donde $\bar{m} = \frac{1}{2}n(n+1)$, $\mathbb{A}(X) = \text{svec}(X)$, y $\mathbb{B}(Y) = -\text{svec}(Y)$ con

$$\text{svec}(X) = (x_{11}, \sqrt{2}x_{12}, \dots, \sqrt{2}x_{1n}, x_{22}, \sqrt{2}x_{23}, \dots, \sqrt{2}x_{2n}, \dots, x_{nn})$$

ANEXO D. Producción Científica

ANEXOS D.1. Artículos Publicados

[IEEE.org](#) | [IEEE Xplore](#) | [IEEE SA](#) | [IEEE Spectrum](#) | [More Sites](#)

[SUBSCRIBE](#) | [Cart](#) | [Create Account](#) | [Personal Sign In](#)

IEEE Xplore[®]
[Browse ▾](#)
[My Settings ▾](#)
[Help ▾](#)
[Institutional Sign In](#)

[ADVANCED SEARCH](#)

[Conferences](#) > [2020 IEEE Engineering Interna...](#)

A Systematic Literature Review on Support Vector Machines Applied to Classification

Publisher: IEEE
 [Cite This](#)
[PDF](#)

[Miguel Angel Cano Lengua](#); [Erik Alex Papa Quiroz](#)
[All Authors](#)

2
Paper
Citations

246
Full
Text
Views

[R](#)
[↔](#)
[©](#)
[📁](#)
[🔔](#)

Abstract

Document Sections

- 1. Introduction
- II. Research Methodology
- III. Study Analysis
- IV. Analysis of Results
- » Conclusions

Abstract:

This paper aims to identify the current state of the art of the latest research related to support vector machines through a literature review system according to the methodology proposed by Kitchenham and Charter, in order to answer the following research questions: Q1: In which research areas are they used? Q2: What are the main applications related with classification? Q3: What optimization methods or algorithms are used in SVMs?

Published in: 2020 IEEE Engineering International Research Conference (EIRCON)

Date of Conference: 21-23 October 2020

INSPEC Accession Number: 20193229

Date Added to IEEE Xplore: 17 November 2020

DOI: 10.1109/EIRCON51178.2020.9254028

► ISBN Information:

Publisher: IEEE

Conference Location: Lima, Peru

1. Introduction

Need Full-Text
access to IEEE Xplore for your organization?
[REQUEST A FREE TRIAL >](#)

More Like This

Twin Support Vector Machines for Pattern Classification
IEEE Transactions on Pattern Analysis and Machine Intelligence
Published: 2007

Weighted least squares twin support vector machines for pattern classification
2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)
Published: 2010

[Show More](#)

A Systematic Literature Review on Support Vector Machines Applied to Classification

Miguel Angel Cano Lengua
Facultad de Ingeniería de Sistemas e Informática
Universidad Nacional Mayor de San Marcos
Lima, Perú
mcanol@unmsm.edu.pe

Erik Alex Papa Quiroz
Universidad Nacional Mayor de San Marcos
Universidad Privada del Norte
Lima, Perú
erikpapa@gmail.com

Abstract—This paper aims to identify the current state of the art of the latest research related to support vector machines through a literature review system according to the methodology proposed by Kitchenham and Charter, in order to answer the following research questions: Q1: In which research areas are they used? Q2: What are the main applications related with classification? Q3: What optimization methods or algorithms are used in SVMs?

Keywords— Vector Support Machines, optimization algorithms, classification, hyperplane, linear and nonlinear separation.

I. INTRODUCTION

The classification-regression method of Vector Support Machines (SVMs) was developed in the 90's, within the field of Computational Science [1]. Although it was originally developed as a binary classification method, its applications have extended to multiple classification and regression problems. SVMs have turned out to be one of the best classifiers for a wide range of situations, which is why it is considered one of the benchmarks in the field of statistical learning and machine learning.

In this paper, using the methodology of Kitchenham and Charter [2], we present a review of the literature on SVMs. The main contributions of this paper are the followings: We will know the applications of SVMs in diverse areas of research; the recent applications of SVMs in classification and the optimization algorithms or optimization methods used for this class of machines.

This research is organized by the following sections: In Section 2, we present the proposed methodology. In Section 3, we analyze the information. In Section 4, we analyze the results and in Section 5, we present the conclusions of the work. Finally, we give the sources of references.

II. RESEARCH METHODOLOGY

The systematic review of the literature that we will consider is the procedure proposed by Kitchenham and Charter [2]. This methodology identifies, evaluates, and interprets all available research relevant to particular questions, subject area, or phenomenon of interest. It will be used to devote its structure is accordance with the objectives of our research. This methodology has been recently used by several researches, see [25,26]. However, there are other methodologies, for example [3] and [4], which do not fit to the field of engineering. The proposed methodology involves the following phases:

- **Review planning:** In this phase the research questions are asked and the review protocol is defined.

- **Carrying out the review:** In this phase, the plan is executed and the main studies are selected, according to the inclusion and exclusion criteria.

- **Review report:** In this phase, the statistics, analysis of the selected documents, and discussions are presented.

II.1 Review planning

We search papers that answer the following questions:

Q1: In which research areas SVMs are used?

Q2: What are the latest related applications?

Q3: What optimization methods or algorithms are used in SVMs?

The search was carried out in the following databases: Web of Science, Science Direct, IEEE Xplore Digital Library and ACM Digital Library, Scopus and Springer Link. The review covers the period from 1992 to 2020.

It was used the following stream search: TITLE-ABS-KEY ("areas where SVM are used") OR TITLE-ABS-KEY ("Applications related to classification") OR TITLE-ABS-KEY ("Optimization algorithms used in SVMs"), which have been applied on the title, abstract and keywords. Then, the selection and exclusion criteria showed in Tables 1 and 2 were applied.

II.2 Carrying out the review:

Table 1: Inclusion Criteria

Criterion	Justification
Research focus	papers with the text support vector machines as a keyword
Quantitative information	Articles must be published in indexed journals
Importance factor	SVM linked to classification and applied to Artificial Intelligence
Language	English

Table 2: Exclusion Criteria

Criterion	Justification
Type of publication	Books, dissertations, book chapters between 1990-2000 (Not updated)




[Journals](#)
[Magazines](#)
[Proceedings](#)
[Books](#)
[SIGs](#)
[Conferences](#)
[People](#)

[Conference](#)
[Proceedings](#)
[Upcoming Events](#)
[Authors](#)
[Affiliations](#)
[Award Winners](#)

[Home](#) > [Conferences](#) > [ICMSSP](#) > [Proceedings](#) > [ICMSSP 2020](#) > [A Proximal Multiplier Method for Convex Separable Symmetric Cone Optimization](#)

RESEARCH ARTICLE

[Twitter](#)
[LinkedIn](#)
[Facebook](#)
[Email](#)

A Proximal Multiplier Method for Convex Separable Symmetric Cone Optimization

Authors:

 [Erik Alex Papa Quiroz](#),
  [Julio López Luis](#),
  [Miguel Cano Lengua](#)
[Authors Info & Claims](#)

ICMSSP 2020: Proceedings of the 2020 5th International Conference on Multimedia Systems and Signal Processing • May 2020

• Pages 92–97 • <https://doi.org/10.1145/3404716.3404734>

Published: 08 July 2020

[Publication History](#)

 0
  22






ICMSSP 2020: Proceedings of the...
A Proximal Multiplier Method for Convex...
Pages 92–97

[← Previous](#)
[Next →](#)

ABSTRACT

References

ABSTRACT

This work is devoted to the study of a proximal decomposition algorithm for solving convex symmetric cone optimization with separable structures. The algorithm considered is based on a decomposition method and proximal distances. Under suitable assumptions, we prove that each limit point of the primal-dual sequences generated by the algorithm solves the problem. Finally, the global convergence is established.



A Proximal Multiplier Method for Convex Separable Symmetric Cone Optimization

Erik Alex Papa Quiroz
Universidad Privada del Norte,
Lima, Perú and Universidad
Nacional Mayor de San Marcos,
Lima, Perú
erikpapa@gmail.com

Julio López Luis
Universidad Diego Portales, Chile
julio.lopez@udp.cl

Miguel Cano Lengua
Universidad Privada Norbert
Wiener, Lima, Perú and
Universidad Nacional Mayor de
San Marcos, Lima, Perú.
miguelangel.cano@unmsm.
edu.pe

ABSTRACT

This work is devoted to the study of a proximal decomposition algorithm for solving convex symmetric cone optimization with separable structures. The algorithm considered is based on a decomposition method and proximal distances. Under suitable assumptions, we prove that each limit point of the primal-dual sequences generated by the algorithm solves the problem. Finally, the global convergence is established.

CCS Concepts

• Mathematics of computing → Quasiconvex programming and unimodality

Keywords

Decomposition method; Euclidean Jordan algebra; Proximal distance; Symmetric cone optimization.

1. INTRODUCTION

Let V_1 and V_2 be two finite dimensional vectorial spaces on the real field $\mathbb{R}$ endowed with the inner product $\langle \cdot, \cdot \rangle_{V_1}$ and $\langle \cdot, \cdot \rangle_{V_2}$ respectively. Then consider the following Euclidean Jordan algebra $V_1 = (V_1, \circ_1, \langle \cdot, \cdot \rangle_{V_1})$ and $V_2 = (V_2, \circ_2, \langle \cdot, \cdot \rangle_{V_2})$, where $\circ_1$ and $\circ_2$ denote the Jordan product in V_1 and V_2 , respectively. In this work, we consider the following convex symmetric cone optimization (SCP) problem with separable structure:

$$\min \{f(x) + g(z) : Ax + Bz = b, x \in K_1, z \in K_2\}, \quad (1)$$

where $f : V_1 \rightarrow \mathbb{R} \cup \{+\infty\}$ and $g : V_2 \rightarrow \mathbb{R} \cup \{+\infty\}$ are closed proper convex function, $A : V_1 \rightarrow \mathbb{R}^m$ and $B : V_2 \rightarrow \mathbb{R}^m$ are two linear mapping, $b \in \mathbb{R}^m$, and $K_1 := \{x \circ_1 x : x \in V_1\}$ and $K_2 := \{z \circ_2 z : z \in V_2\}$ denoting the sets of square elements (symmetric cones) in V_1 and V_2 , respectively.

This problem provides a simple unified framework for various existing optimization problems such as nonlinear, second-order cone, and semidefinite programming, and hence has extensive

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

JCMSSP 2020, May 28–30, 2020, Chengdu, China
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7741-6/20/04... \$15.00

DOI: <https://doi.org/10.1145/3404716.3404734>

applications in engineering, economics, game theory, management science, and other fields; see [1]–[5] and references therein. In particular, certain class of problem of (1) have drawn recently a lot of attention due to their emerging applications of practical interest being these the support vector machines and the sparse inverse covariance selection.

The idea of the predictor corrector proximal multiplier (PCPM) for solving (P) without conical constraints reads as follow:

$$\begin{aligned} p^{k+1} &= y^k + \lambda_k (Ax^k + Bz^k - b), \\ x^{k+1} &= \arg \min_{x \in \mathbb{R}^n} \left\{ f(x) + \langle p^{k+1}, Ax \rangle + \frac{1}{2\lambda_k} \|x - x^k\|^2 \right\}, \\ z^{k+1} &= \arg \min_{z \in \mathbb{R}^n} \left\{ g(z) + \langle p^{k+1}, Bz \rangle + \frac{1}{2\lambda_k} \|z - z^k\|^2 \right\}, \\ y^{k+1} &= y^k + \lambda_k (Ax^k + Bz^k - b) \end{aligned}$$

where $\{\lambda_k\}_{k \in \mathbb{N}}$ is a sequence of positive parameters.

Recently, Sarmiento *et al.* [6] developed an extension of the (PCPM) method for solving (1), when the Euclidean Jordan algebra is the well-know Euclidean vectorial space, by using regularized proximal distances [7]. In order to consider a broad range of applications, as for example separable semidefinite and second-order cone optimization problems (see also the Section 4), which cannot be covered by the work [6], the purpose of this paper is to extend the (PCPM) method to solve the (CP) problems. Specifically, first we define a proximal distance with respect to the interior of the symmetric cone K . This distance can be generated by an appropriate class of continuously differentiable strict convex functions on $\mathbb{R}_{++}$. Secondly, we propose a proximal decomposition algorithm using proximal distance for solving (1). Then, under some mild assumptions, we establish the global convergence of the proposed algorithm.

This paper is organized as follows. Section II recalls some basic notions and reviews some essential results on Euclidean Jordan algebras. In Section III, we introduce the class of proximal distances defined in the symmetric cone. Finally, in Section IV, we describe the algorithm with proximal distance for solving (1) and establish its global convergence.

2. PRELIMINARIES

In this subsection, we briefly describe some concepts, properties, and results from Euclidean Jordan algebras needed in this paper. The theory may be found in [8].

A Euclidean Jordan algebra is a triple $(V, \circ, \langle \cdot, \cdot \rangle_V)$, where $(V, \circ, \langle \cdot, \cdot \rangle_V)$ is a finite dimensional space over the real field $\mathbb{R}$ equipped with an inner product $\langle \cdot, \cdot \rangle_V$, and the Jordan product


Seventh Sense Research Group®

[Home](#)
[Articles](#)
[Authors](#)
[Editors](#)
[Reviewers](#)

International Journal of Engineering Trends and Technology

A Symmetric Cone Proximal Multiplier Algorithm



© 2023 by IJETT Journal
 Volume-71 Issue-1
 Year of Publication : 2023
 Author :Erik Alex Papa Quiroz, Miguel Angel Cano Lengua, Julio Cesar Lopez Luis, Rolando Ichpas Tapia
 DOI :10.14445/22315381/IJETT-V71I1P223



How to Cite?

Erik Alex Papa Quiroz, Miguel Angel Cano Lengua, Julio Cesar Lopez Luis, Rolando Ichpas Tapia. "A Symmetric Cone Proximal Multiplier Algorithm," — *International Journal of Engineering Trends and Technology* , vol.71, no. 1, pp. 257-270, 2023. Crossref, <https://doi.org/10.14445/22315381/IJETT-V71I1P223>

Abstract

This paper introduces a proximal multipliers algorithm to solve separable convex symmetric cone minimization problems subject to linear constraints. The algorithm is motivated by the method proposed by Sarmiento et al. (2016, optimization v.65, 2, 501-537), but we consider in the finite-dimensional vectorial spaces, further to an inner product, a Euclidean Jordan Algebra. Under some natural assumptions on convex analysis, it is demonstrated that all accumulation points of the primal-dual sequences generated by the algorithm are solutions to the problem and assuming strong assumptions on the generalized distances; we obtain the global convergence to a minimize point. To show the algorithm's functionality, we provide an application to find the optimal hyperplane in Support Vector Machine (SVM) for binary classification.

Keywords

Symmetric convex cone optimization, Separable techniques, Proximal distances, Proximal method of multipliers, Support vector machine.

Original Article

A Symmetric Cone Proximal Multiplier Algorithm

Erik Alex Papa Quiroz¹, Miguel Angel Cano Lengua², Julio Cesar Lopez Luis³, Rolando Ischpas Tapia⁴

¹ Universidad Nacional Mayor de San Marcos, Universidad Privada del Norte, Universidade Federal de Goiás, Lima, Perú, Goiânia, Brazil

² Universidad Nacional Mayor de San Marcos, Universidad Tecnológica del Perú, Lima, Perú

³ Universidad Diego Portales, Chile

⁴ Intendencia Nacional de Estrategias y Riesgos, Lima, Perú

mcanol@unmsm.edu.pe

Received:

Revised:

Accepted:

Published: (Size9)

Abstract - This paper introduces a proximal algorithm of multipliers to solve separable convex symmetric cone minimization problems subject to linear constraints. The algorithm is motivated from the method given by Sarmiento et al. (2016, Optimization v.65, 2, 501-537) but we consider in the finite dimensional vectorial spaces, further to an inner product, an Euclidean Jordan Algebra. Under some natural assumptions on convex analysis, it is demonstrated that all accumulation points of the primal-dual sequences generated by the algorithm are solutions of the problem and assuming stressed assumptions on the generalized distances we obtain the global convergence to a minimize point. To show the functionality of the algorithm we give an application to find optimal hyperplanes in Support Vector Machine (SVM) for binary classification.

Keywords - Symmetric convex cone optimization, separable techniques, proximal distances, proximal method of multipliers, support vector machine.

1. Introduction

Consider V_1 and V_2 be two linear vectorial spaces of finite dimensions on $\mathbb{R}$, $\mathbb{R}$ is considered as a real field. On this spaces we define two inner products: $\langle \cdot, \cdot \rangle_{V_1}$ for V_1 and $\langle \cdot, \cdot \rangle_{V_2}$ for V_2 with the Jordan product $\circ_1$ and $\circ_2$ respectively. Based in these tools we can define the Euclidean Jordan algebra $V_1 = (V_1, \circ_1, \langle \cdot, \cdot \rangle_{V_1})$ and $V_2 = (V_2, \circ_2, \langle \cdot, \cdot \rangle_{V_2})$, respectively, see subsection 2.1 for a strict definition of this concept.

In this paper, our interest is to introduce an optimization algorithm for solving the convex symmetric cone optimization problem (CSCO problem from now on) considering variables and linear constraints:

$$\min \{f(x) + g(z) : Ax + Bz = b, x \in k_1, z \in k_2\}, \quad (1.1)$$

where $f : V_1 \rightarrow \mathbb{R} \cup \{+\infty\}$ and $g : V_2 \rightarrow \mathbb{R} \cup \{+\infty\}$ are proper, closed and convex functions (possible nonsmooth), $A : V_1 \rightarrow \mathbb{R}^m$ and $B : V_2 \rightarrow \mathbb{R}^m$ are linear applications, $b \in \mathbb{R}^m$, and $k_1 := \{x \circ_1 x : x \in V_1\}$ and $k_2 := \{z \circ_2 z : z \in V_2\}$ are the sets of square elements (symmetric cones) in V_1 and V_2 ,

respectively.

The model (1.1) recovers a wide variety of applications in the fields of science and engineering (for example, in economy, game theory, management science, see for instance [1, 2, 16, 17, 25] and the references of those papers. In particular, includes a lot of applications of current interest (for more details, see 1 and 2 of Section 3).

Many investigations, taking advantage of the separable structure of the object function, have introduced several decomposition methods to solve the problem (1). Between the most recognized methods we can find the alternating direction of multipliers method [9], the partial inverse method [22], the predictor-corrector proximal multiplier (PCPM) method [7]. In this present investigation, we will focus on the (PCPM) using proximal distances.

The steps of the (PCPM) method to solve (1) without conic constraints are the following



This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

ANEXOS D.2. Artículos Aceptados

Indonesian Journal of Electrical Engineering and Computer Science
Vol. 99, No. 1, Month 2099, pp. 1~1x
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v99.i1.pp1-1x

□ 1

Design of a Optimization Algorithm for Binary Classification

Miguel Angel Cano Lengua¹, Erik Alex Papa Quiroz², Marco Antonio Alvarado Cifuentes³, Carlos Antonio Alvarado Cifuentes⁴

¹Universidad Nacional Mayor de San Marcos,
Universidad Tecnológica del Perú, Lima, Perú

²Universidad Nacional Mayor de San Marcos, Lima, Perú
Universidad Federal de Goiás, Goiania, Brazil

³Universidad Privada del Norte, Lima, Perú

^{3,4}Universidad Tecnológica del Perú, Lima, Perú

Article Info	ABSTRACT
Article history: Received month. Revised month Accepted month	In the present work, the design of a system to classify data is carried out, using the Scrum methodology. The validation was carried out by expert judgment, having favorable results in terms of different criteria such as integrity, ease of use, innovation, and scalability. Regarding the development of the functional elements of the system, it was obtained; He developed the architecture of the system, the database, and the prototypes, among other points considered. From the implementation of the system, the equation of a classifying plane in three-dimensional space will be obtained, as well as the number of internal iterations that the algorithm develops, the estimated execution time, and the graph of the plane. This system is based on a recently introduced symmetric cone proximal multiplier algorithm to solve separable optimization problems, this algorithm made an application for classification-related support vector machines.
Keywords: Binary Classification Proximal Multiplier Support Vector Machine Proximal Distances Scrum Methodology	

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Miguel Angel Cano Lengua
Universidad Nacional Mayor de San Marcos
Universidad Tecnológica del Perú
Lima, Perú
Email: mcanol@unmsm.edu.pe

1. INTRODUCTION

At present, due to the demands that exist in different areas such as marketing for customer segmentation, in medicine for the classification of cancer cell images, in the personalizing preferences, etc. For this, efficient classification algorithms are required.

The different classification models start from a data set which has a set of characteristics, and we know the class to which it belongs by calling this training or learning set, a set of rules is created which allows us to validate with a set of different data.

Classification techniques arise thanks to the advancement of science. In the field of artificial intelligence, one of the important disciplines is learning machines, which are divided into two areas: supervised learning machines and unsupervised learning machines, as can be seen in the Figure 1.