

**ACERCAMIENTO A LAS MÁQUINAS DE SOPORTE VECTORIAL Y SUS
APLICACIONES EN PROYECTOS DE GRADO DEL PROGRAMA DE
INGENIERÍA DE SISTEMAS Y COMPUTACION DE LA UNIVERSIDAD
TECNOLOGICA DE PEREIRA.**

MÓNICA ANDREA JIMÉNEZ MORALES

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS: ELÉCTRICA, ELECTRÓNICA, FÍSICA Y
CIENCIAS DE LA COMPUTACIÓN
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PEREIRA
2012**

**ACERCAMIENTO A LAS MÁQUINAS DE SOPORTE VECTORIAL Y SUS
APLICACIONES EN PROYECTOS DE GRADO DEL PROGRAMA DE
INGENIERÍA DE SISTEMAS Y COMPUTACION DE LA UNIVERSIDAD
TECNOLOGICA DE PEREIRA.**

MÓNICA ANDREA JIMÉNEZ MORALES

**TRABAJO DE GRADO
MONOGRAFIA**

**DOCUMENTO PROYECTO DE GRADO PRESENTADO COMO REQUISITO
PARCIAL PARA OPTAR AL TÍTULO DE INGENIERO(A) DE SISTEMAS Y
COMPUTACIÓN**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS: ELÉCTRICA, ELECTRÓNICA, FÍSICA Y
CIENCIAS DE LA COMPUTACIÓN
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PEREIRA
2012**

Nota de aceptación:

Firma del presidente del jurado

Firma jurado

Firma jurado

Pereira 16 de noviembre del 2012

DEDICATORIA

Para mis sobrinos, Juan David Jiménez, Miguel Ángel Brochero Jiménez, Ana Sofía Jiménez, Emanuel Brochero Jiménez, Gabriela Jiménez y Luciana Jiménez.

Luchen por sus sueños sin perder el horizonte, su meta es superarnos.

AGRADECIMIENTOS

A mis padres Guillermo Jiménez y Mariela Morales, por su enorme sacrificio y apoyo a lo largo de todo este proceso que al fin culmina.

A mis hermanos, Ing. Electricista David Ricardo Jiménez por motivarme a la Ingeniería y darme el primer acercamiento a los sistemas, el Economista Juan Guillermo Jiménez, y la Rehabilitadora Cardiopulmonar Lina María Jiménez por el apoyo a lo largo de este proceso.

Y al Ing. Hugo Humberto Morales por brindarme tiempo y el apoyo que necesitaba para culminar con este proyecto.

CONTENIDO

	Pág.
1. GENERALIDADES	15
1.1. FORMULACIÓN DEL PROBLEMA.....	15
1.2. JUSTIFICACIÓN.....	15
1.3. OBJETIVOS.....	16
1.3.1. Objetivos específicos	16
1.4. INTRODUCCIÓN	16
2. MARCO DE REFERENCIA	18
2.1. MARCO CONCEPTUAL	18
2.1.1. Máquinas de soporte vectorial	18
2.1.2. Margen y el hiperplano	19
2.1.3. Clasificación de vectores de soporte lineales y no lineales	20
2.1.4. Kernel	21
3. MARCO TEORICO	22
3.1. TEORÍA APRENDIZAJE COMPUTACIONAL.....	22
3.1.1. Machine Learning	22
3.1.2. Métodos basados en estrategias de aprendizaje.....	23
3.2. APRENDIZAJE AUTOMÁTICO	23
3.2.1. Generador.....	23
3.2.2. Sistema.....	24
3.2.3. Máquinas de aprendizaje.....	24
3.2.4. Supervisado	24
3.2.5. No supervisado.....	24
3.3. MACHINE VISION	25

3.3.1. Medición de características	25
3.3.2. Clasificación de patrones.....	25
4. TEORIA DE LAS SVM	26
4.1. HISTORIA DE LAS SVM	26
4.2. PREÁMBULO MATEMÁTICO	26
4.2.1. Dimensión Vapnik-Chervonenkis(VC).....	27
4.2.2. Condición Kurush-Kuhn-Turcker (KKT)	27
4.2.3. Teorema de Mercer	28
4.3. MINIMIZACIÓN DE RIESGO ESTRUCTURAL	29
4.3.1. Reconocimiento de patrones	30
4.3.2. Minimización del riesgo empírico.....	30
4.3.3. Minimización de riesgo estructural.....	30
4.3.4. Confidence interval	33
4.4. INTRODUCCIÓN A LOS VECTORES DE SOPORTE	33
4.4.1. Ejemplo de una máquinas de vector características.....	33
4.5. GENERALIZACIÓN DEL LÍMITE.....	34
4.5.1. Margen de distribución basada en limites.....	35
4.6. SUPPORT VECTOR MACHINE(SVM)	36
4.6.1. SVM lineales.....	37
4.6.2. Minimización del vector proyección y maximización del margen	40
4.6.3. SVM no lineales.....	42
4.7. MULTIVLASES EN SVM	53
4.7.1. One-against-all (OAASVM).....	53
4.7.2. Pairwise SVM	54

4.7.3. Error correcting output code(ECOC).....	54
4.7.4. All-at-once SVM.....	54
4.7.5. Grafica aciclica dirigido SVM(DAGSVM)	55
4.7.6. One-against-one(OAOSVM)	55
4.8. DISTINTOS TIPOS DE MÁQUINAS DE SOPORTE VECTORIAL	55
4.8.1. SVM transductiva(TSVM)	56
4.8.2. Arboles de SVM (BTSMV)	56
4.8.3. Algoritmo de las SVM basada en función objetivo.....	57
4.8.4. Método de reconocimiento dinámico con SVM.....	58
4.9. COMPARACION DE SVM CON DISTINTOS KERNEL.....	59
4.9.1. Comparación RBF y polinomial	59
4.9.2. Comparación kernel lineal, polinomial y RBF	61
4.10. COMPARACION ENTRE DISTINTOS TIPOS DE SVM	62
4.10.1. Comparación entre SVM y métodos de reconocimiento dinámico....	62
4.10.2. Problema de clasificación de expresión de genes	63
4.10.3. Comparación OAASVM y OAOSVM.....	64
4.11. COMPARACION DE LAS SVM CON LAS REDES NEURONALES.....	65
4.11.1. Comparación entre SVM con redes neuronales con datos diversos	65
4.11.2. Clasificación con muestras pequeñas.....	65
4.11.3. Ventajas y desventajas frente a las redes neuronales.....	66
4.12. METODOS DE ENTRENAMIENTO.....	67
4.12.1. Técnicas de descomposición.....	67
4.13. METODOS DE CLASIFICACION	69
4.13.1. Ejemplos de clasificación.....	69

4.14.	MÁQUINAS DE SOPORTE VECTORIAL PARA LA REGRESIÓN	71
4.14.1.	Calcular la calidad de la estimación.....	73
5.	ESTADO DEL ARTE	74
5.1.	EXTRACCIÓN DE OBJETOS EN UN VIDEO	74
5.2.	DETECCIÓN DE ROSTRO BASADO EN SEGMENTACIÓN DEL COLOR DE LA PIEL	76
5.3.	EXPLORANDO LA ROTACIÓN INVARIANTE CON EL SVM CLASIFICADOR PARA LA DETECCIÓN DE MICROCLASIFICACIONES.....	78
5.4.	DETECCIÓN HUMANA CON PC-SVM	81
5.5.	ESTUDIO DE SVM COMPARADO CON OTROS MÉTODOS DE CLASIFICACIÓN DE TEXTO	83
5.6.	MAPREDUCE BASADO EN SVM PARALELO PARA FILTRAR SPAM EN LARGA ESCALA.....	86
6.	APLICACIONES FUTURAS EN EL AREA DE INGENIERÍA DE SISTEMAS	
6.1.	RECONOCIMIENTO DE PATRONES ESTÁTICOS OFF-LINE (VOCALES Y CONSOANTES DEL LENGUAJE DE SEÑAS)	91
6.1.1.	La problemática	91
6.1.2.	Señales del lenguaje de señas colombiano.....	92
6.1.3.	Etapas de desarrollo.....	94
6.2.	RECONOCIMIENTO DE PATRONES ESTATICOS ON-LINE (VOCALES Y CONSONANTES LE LENGUAJE DE SEÑAS)	99
6.2.1.	Detección de las muestras y procesado	99
6.2.2.	Clasificación.....	101
6.3.	ANOTACIONES.....	101
7.	CONCLUSIONES	102
8.	BIBLIOGRAFÍA.....	104

LISTA DE FIGURAS

	Pág.
Figura 1. Limite de decisión lineal	18
Figura 2. Margen	19
Figura 3. Set de entrenamiento en dos dimensiones, “+” pertenece a clase positiva y “-” la clase negativa.....	19
Figura 4. Ejemplos de hiperplano, a) el hiperplano que se aprendió por las SVM, los círculos corresponden a vectores de soporte, b) el aprendizaje de las SVM con un punto puede cambiar.....	20
Figura 5. Las SVM aplicaba a datos no separables en dos dimensiones, los vectores de soporte están indicada por círculos.....	21
Figura 6. Estructura básica de un sistema de aprendizaje.....	22
Figura 7. Una máquina de aprendizaje.....	24
Figura 8. Organización de un sistema de machine visión	25
Figura 9. Separación de dos clases	27
Figura 10. Estructura de un set de funciones anidados en un sub set de funciones visión	31
Figura 11. El límite de riesgo visión	32
Figura 12. Contribución de los puntos de datos a la generalización del error	36
Figura 13. Error de riesgo funcional	37
Figura 14. Ortogonalidad de los puntos	38
Figura 15. Distribución de las clases y su regla de decisión.....	39
Figura 16. Lados del hiperplano	40
Figura 17. SVM no lineal.....	42

Figura 18. Variables de holgura.....	43
Figura 19. Espacio de características a un espacio de alta dimensión.....	46
Figura 20. Procedimiento de mapeado alta dimensión y luego hallar producto punto.....	46
Figura 21. Solucionando el problema de optimización usando Kernels	47
Figura 22. Ejemplo de transformación no lineal	49
Figura 23. Transformada a otros espacios de Hilbert	52
Figura 24. Limite con uno contra todos	54
Figura 25. Limite de clases con pairwise SVM.....	54
Figura 26. Diagrama de flujo método de reconocimiento dinámico.....	58
Figura 27. Comparación de los límites de decisión	60
Figura 28. Comparación del mejor limite de decisión	61
Figura 29. Selección de candidatos de vectores de soporte	68
Figura 30. Ejemplos de pocos puntos y la ecuación hiperplano	69
Figura 31 Regresión lineal con una banda de ξ – intensiva.....	71
Figura 32. Regresión no lineal	71
Figura 33. Variables de holgura.....	72
Figura 34. Fases de la extracción.....	74
Figura 35. Comparación SVM con SOFT SVM.....	75
Figura 36. Secuencia de frames	75
Figura 37. Modelo del rostro.....	76
Figura 38. Imágenes de la detección	77
Figura 39. Mamografía con calcificaciones.....	78

Figura 40. Imágenes con microcalcificaciones (primera fila) y fondo de radiografía (segunda fila).	80
Figura 41. Estructura identificación humana	81
Figura 42. Resultado del PC-SVM.	83
Figura 43. Clasificador de texto	84
Figura 44. Grafica de los resultados	86
Figura 45. Diseño MapReduce y SMO paralelo.....	87
Figura 46. Diseño general.....	89
Figura 47. Resultado de los nodos.	90
Figura 48. Porcentaje de personas por deficiencia en estructuras o funciones corporales	91
Figura 49. Vocales del lenguaje de señas	92
Figura 50. Vocales del lenguaje de señas	92
Figura 51. Consonantes estáticas señas	93
Figura 52. Consonantes móviles.....	94
Figura 53. Secuencia de patrones para decir “buenos días”	94
Figura 54. Escala de grises de las vocales de lengua de señas.....	95
Figura 55. Distintas dimensiones	95
Figura 56. Imagen binarizada	96
Figura 57. Histograma de la seña para la letra a	97
Figura 58. Histograma de la seña para la letra e	97
Figura 59. Histograma de la seña para la letra i	98

Figura 60. Histograma de la seña para la letra o	98
Figura 61. Histograma de la seña para la letra u	99
Figura 62. Secuencia hand –tracking	100
Figura 63. Secuencia hand – tracking II.....	100
Figura 64. Secuencia hand –tracking III.....	101

LISTA DE TABLAS

	Pág.
Tabla 1. Kernel conocidos	51
Tabla 2. Comparación entre SVM	55
Tabla 3. Comparación con precisión de la predicción.....	60
Tabla 4. Comparación con la precisión de las predicciones	61
Tabla 5. Comparación entre los núcleos de Kernel.....	62
Tabla 6. Comparación	62
Tabla 7. Comparación entre tipos de SVM	64
Tabla 8. Comparación con Kernels múltiples	64
Tabla 9. Resultado de la comparación	65
Tabla 10. Comparación entre SVM y redes neuronales	65
Tabla 11. Comparación de tiempo entre ambos métodos.....	66
Tabla 12. Resultado de la comparación entre las tres técnicas.	80
Tabla 13. Comparación entre métodos	85
Tabla 14. Resultados del entrenamiento	90

1. GENERALIDADES

1.1. FORMULACIÓN DEL PROBLEMA

A mediados de los 90, se comenzó a utilizar el concepto de las Máquinas de Soporte Vectorial (SVM, en inglés Support Vector Machine) para el desarrollo de programas de reconocimiento, regresión y la detección.

El perceptron que llevó a las Redes Neuronales Artificiales, fue el primer método de reconocimiento, muy conocido en el ámbito académico y también muy utilizado en desarrollo de programas de reconociendo y de detección, también se desarrollaron otros métodos como la transformada de Wavelet, las cadenas de Markov y los procesos Gaussianos para la regresión, entre otras técnicas.

Las Máquinas de Soporte Vectorial es una técnica ampliamente utilizada en el mundo, pero muy poco conocida e incluso desconocida para el ámbito académico de pregrado en el programa de Ingeniería de Sistemas y Computación¹ de la Universidad Tecnológica de Pereira, como otra opción a parte de las Redes Neuronales.

Los proyecto de grados de la carrera de Ingeniería de Sistemas y Computación, no se presenta ninguno² con base a las Maquinas de Soporte Vectorial enfocados a la construcción de software de reconocimiento y clasificación de patrones, por ese motivo esta monografía busca dar a conocer la teoría, estado de arte y aplicaciones de las Máquinas de Soporte Vectorial.

1.2. JUSTIFICACIÓN

A nivel académico se realizara un documento el cual se describa el funcionamiento de las Máquina de Soporte Vectorial, su fundamento teórico, matemático y el estado del arte; también se analizara sus aplicaciones, y se recomendara pautas para el desarrollo futuro de software de reconocimiento estático de patrones con esta metodología para la elaboración de proyectos de grado futuro en el programa de Ingeniería de Sistemas y Computación.

¹ <http://isc.utp.edu.co/archivos/materia0843136is943-sistemas-expertos.pdf>.

² <http://recursosbiblioteca.utp.edu.co/tesisdigitales/ingsistemas.html>.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Desarrollar una monografía la cual se muestre a fondo la estructura de las Máquinas de Soporte Vectorial y como se pueden aplicar en proyectos de grado en el programa de Ingeniería de Sistemas y Computación.

1.3.2. OBJETIVOS ESPECÍFICOS

- Realizar un estudio de las SVM, con base a las fuentes bibliográficas consultadas.
- Realizar un estudio sobre el estado del arte y sus aplicaciones en Ingeniería.
- Elaborar el documento de la monografía y las conclusiones de la investigación.
- Aportar ejemplos de proyectos de grados con esta metodología.

1.4. INTRODUCCION

Las Máquinas de Soporte Vectorial son relativamente nuevas en comparación con otras metodología de reconocimiento de patrones, en este trabajo monográfico presenta un recorrido teórico y matemático de los fundamentos de las Máquinas de Soporte Vectorial, sus aplicaciones a nivel general y las aplicaciones en el programa de Ingeniería de Sistemas y Computación.

Las Redes Neuronales, las Redes de Bayer como en los otros métodos, se parte de datos iniciales, estos datos deben ser trabajados como vectores; se debe tener en cuenta que las máquinas soportan un numero finito de datos, a más datos el tiempo computacional aumenta.

Las Máquinas de Soporte Vectorial se fundamentan en la teoría de minimización de riesgo estructural, las cuales permiten limitar el modelo de las máquinas para que el error en estas sea mínimo o igual a cero si el número de datos en una muestra llegase a aumentar, también se fundamenta en la teoría de las dimensiones Vapnik-Chervonenkis (VC), [NORWAK, 2009] las cuales permiten la separación entre las clases.

Para la construcción de las máquinas, se parte de la transformación de los datos en vectores de dimensión n , estos vectores son distribuidos en un espacio, este espacio inicial se llama espacio de entrada, de acuerdo a su distribución se presentan dos tipos de máquinas, las lineales y no lineales, los datos son ingresados por conjuntos llamados clases, donde solo se soportan dos y son identificadas como $(+1, -1)$, una vez los datos distribuidos se realizan un mapeo, en las máquinas lineales se aplica el producto punto en cada uno de los vectores del espacio de características o en el caso de las máquinas no lineales se usa un *Kernel* o núcleo, aumentándose la dimensionalidad de los vectores, este nuevo espacio es llamado espacio de características, a su vez se realiza una maximización del espacio de separación entre las clases, llamado margen y una línea de separación llamada hiperplano, donde los puntos de soporte están al límite del margen, a diferencia de otros métodos esta conversión es más veloz debido al potencial de paralelismo del método y reduciendo así el tiempo computacional.

Fuentes consultadas [LIPO, 2005], [BISHOP, 2006], [MARTINEZ, 2008], [GARCIA, 2005], [ABE, 2005], [SHAONING y KASASABOV, 2004], [SHI y HU, 2007] hay un completo abanico de tipos de Máquinas de Soporte Vectorial, pero debido a que los problemas son distintos, el efecto de las máquinas en su resolución puede cambiar como se verá en este trabajo, datos iguales pueden provocar un resultado distinto si se usa *Kernel* distintos.

Pero también en la literatura [LIPO, 2005], [CHAMANSEMNI y YASHWANT, 2011], [LI, GUO y LI, 2008] se encontró comparaciones entre las Máquinas de Soporte Vectorial y las Redes Neuronales, bajo el mismo set de datos se encontró que las máquinas dan mejor resultado que las Redes Neuronales.

Las máquinas en su base solo pueden soportar dos clases de datos en su espacio de entrada, pero debido a su versatilidad pueden trabajar con más de dos clases usando más de cuatro tipos de técnicas, donde se resuelven de a pares hasta terminar de mapear todas las clases.

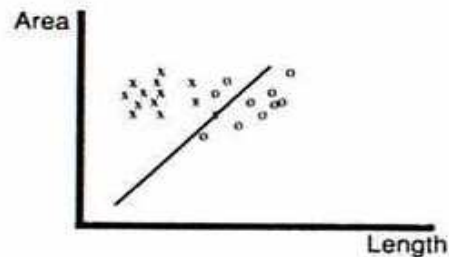
2. MARCO DE REFERENCIA

2.1. MARCO CONCEPTUAL

2.1.1. Máquina de soporte vectorial³: El reconocimiento estático, se basa en el uso de los ejes coordenados X y Y, donde las muestras de conocimiento se encuentran en los ejes, estas son llamadas “set de entrenamiento”; Los datos desconocidos llegan en forma de un vector función, este dato se compara con el set de entrenamiento, donde se analizan las muestras más cercanas, basadas en la distancia euclidiana entre los puntos.

Para crear las reglas de decisión, se necesita una función la cual consta de un set de números que caracteriza alguna propiedad de la imagen, hay muchas formas de determinar las reglas de decisión, y los parámetros que caracterizan cada regla; para representar estas reglas se utiliza una línea llamada línea de límite de decisión.

Figura 1. Limite de decisión lineal.



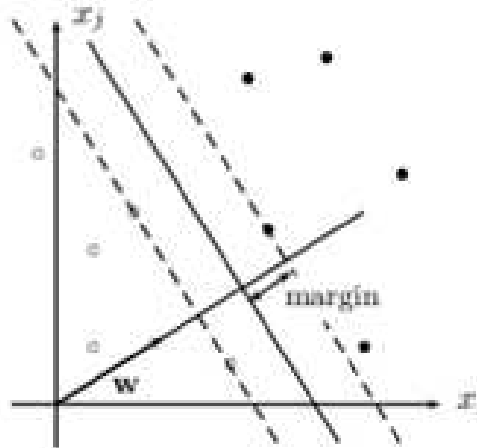
Fuente: Wesley E. QI, Hairong, Machine Vision, Cambridge University Press.

En la figura 1, observamos los puntos X y O, los cuales representan dos clases, lo que se busca determinar es, si el objeto desconocido es el resultado de las muestras que se encuentran en el lado izquierdo del límite de decisión, eligiendo el más cercano; Esta técnica es muy utilizada en máquinas lineales debido a su potencial de paralelismo aumentando la velocidad de análisis.

³ SNYDER, Wesley E. QI, Hairong. Machine Vision, Cambridge University Press. 434 p.

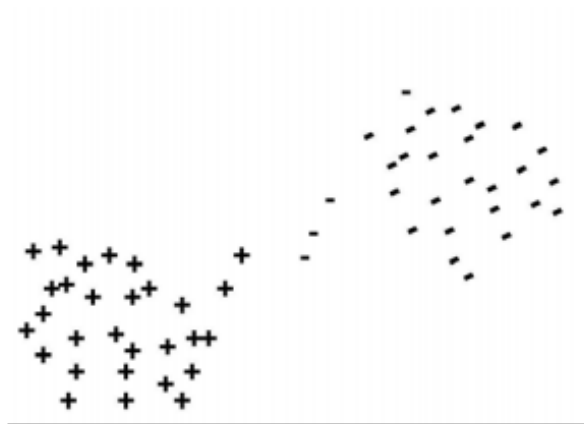
2.1.2. Margen y el hiperplano⁴: el margen es la distancia más cercana del punto del hiperplano de aprendizaje, las SVM se centra en el punto más cercano para el hiperplano y escoge el plano que separa los datos con el margen máximo.

Figura 2. Margen.



Fuente: n. SEBE, ira COHER, ashutosh GARG y HUANG Thomas, Machine Learning in computer vision, Springer.

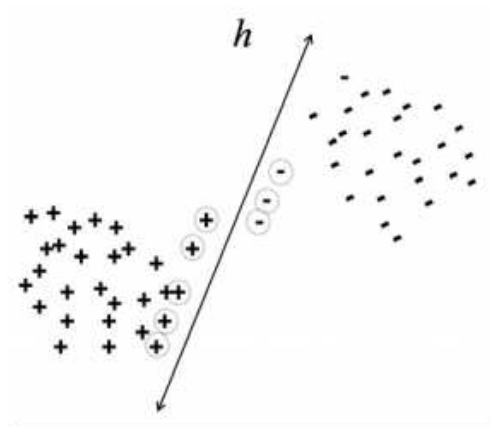
Figura 3. Set de entrenamiento en dos dimensiones, "+" pertenece a clase positiva y "-" la clase negativa.



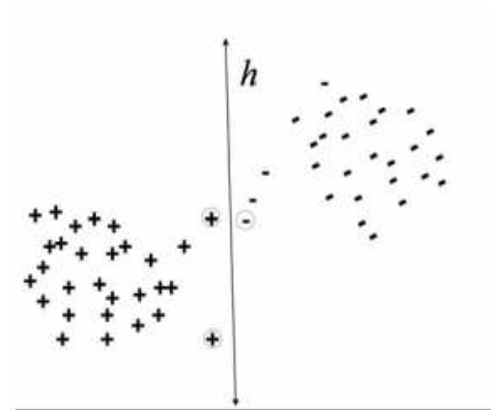
Fuente: n. SEBE, ira COHER, ashutosh GARG y HUANG Thomas, Machine Learning in computer vision, Springer.

⁴ SEBE N, COHER ira, GARG ashutosh y HUANG Thomas, Machine Learning in computer vision, Springer, 2005, 243p.

Figura 4. Ejemplos de hiperplano, a) el hiperplano que se aprendió por las SVM, los círculos corresponden a vectores de soporte, b) el aprendizaje de las SVM con un punto puede cambiar.



(a)



(b)

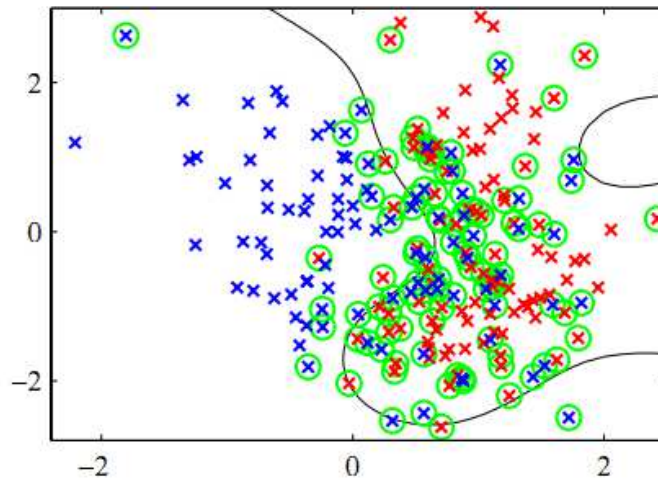
Fuente: n. SEBE, ira COHER, ashutosh GARG y HUANG Thomas, Machine Learning in computer vision, Springer.

La separación óptima debe ser esencial, ya que permite elegir de manera correcta los vectores de soporte y realizar una buena clasificación de los datos del espacio de características.

2.1.3. Clasificación de vectores de soporte lineales y no lineales⁵: las Máquinas de Soporte Vectorial trabajan sobre dos tipos de casos, los separables y no los separables, en casos separables se pueden utilizar clasificadores lineales como la figura 4, en el caso de las no separables se utilizan *Kernel*.

⁵ TONG simon, KOLLER daphne, Computer Science Department, Stanford University, Support Vector Machine Active Learning with Applications to Text Classification, 2001.

Figura 5. Las SVM aplicaba a datos no separables en dos dimensiones, los vectores de soporte están indicada por círculos.



Fuente: M. BISHOP Chistopher, Cambridge, Pattern recognition and machine learning.

2.1.4. Kernel⁶: para evitar la complejidad la información se emplea un *Kernel*, este proyecta los datos no lineales a un espacio característico de mayor dimensión, el cual no solo permite aumentar la capacidad computacional, también la linealidad de las clases en ese espacio.

los *Kernel* más usados son: lineales con producto punto, polinómicas, Gaussiana, perseptron y multicuadrática inversa.

⁶ LIPO Wang, Nanyang Technological University, Support Vector Machine: theory and applications.

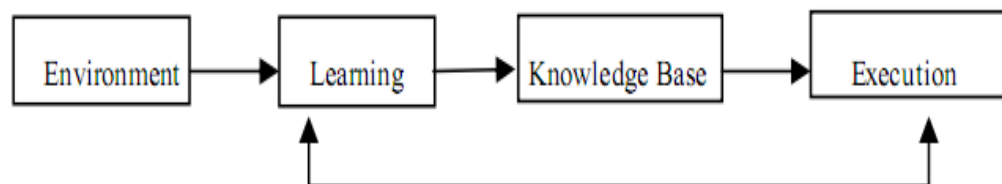
3. MARCO TEÓRICO

3.1. TEORÍA DEL APRENDIZAJE COMPUTACIONAL⁷

Busca algoritmos por los cuales una computadora pueda modificar su comportamiento en forma autónoma con base a ejemplos, como lo haría un humano.

3.1.1. Machine Learning (Máquinas de aprendizaje) estudia como la computadora puede simular o realizar el comportamiento humano, busca establecer modelos computacionales o entender modelos de acuerdo al estudio de mecanismos humanos a través de la sicología, la ciencia cognitiva y el estudio metodológico.

Figura 6. Estructura básica de un sistema de aprendizaje.



Fuente: XUE ming. ZHU changiun , A Study and application on machine learning of artificial intelligence.

El entorno provee cierta información a la sección de aprendizaje del sistema y esta revisa la librería de conocimiento para usar esta información, para mejorar la parte de ejecución.

El factor más importante para el diseño del sistema de aprendizaje es seleccionar la información que se proveerá al sistema y si dicha información es de calidad, también se tiene en cuenta la creación de reglas correctamente constituidas pueden hacer que el sistema sea eficiente, pero unas reglas incorrectas pueden modificar o borrar información de la base de conocimiento.

⁷ XUE ming. ZHU changiun. 2009 INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE. A Study and application on machine learning of artificial intelligence. 3p.

El segundo factor importante que puede afectar el diseño del sistema es la base de conocimiento, el conocimiento puede ser expresado en varias formas, instancias, vector de características, producción de reglas de reconociendo y reglas semánticas.

3.1.2. Métodos basados en estrategias de aprendizaje

Rote Learning (Aprendizaje de memoria) es el más simple de las máquinas de aprendizaje, donde todo conocimiento es guardado, donde la información se recupera por transferencia.

Explanation-Based Learning (Aprendizaje basado en explicación) la estructura de la explicación satisface el objetivo conceptual de basado en ejemplos.

Learning from instruction (Aprendizaje de instrucción) una persona obtiene información del ambiente y trasforma ese conocimiento en expresiones que en su interior repasa y combinan generando un nuevo conocimiento con el conocimiento original.

Learning by deduction (Aprendizaje por deducción) se puede obtener conocimiento útil a partir de procesos de razonamiento.

Learning of analogy (Aprendizaje por analogía) es un método útil y eficaz y se pueden describir claramente la similitud entre los objetos.

3.2. APRENDIZAJE AUTOMÁTICO⁸

Un esquema de aprendizaje consta de tres componentes, un generador de vectores de entrada aleatoria, un sistema que proporciona salida al vector y una máquina de aprendizaje que estima una relación.

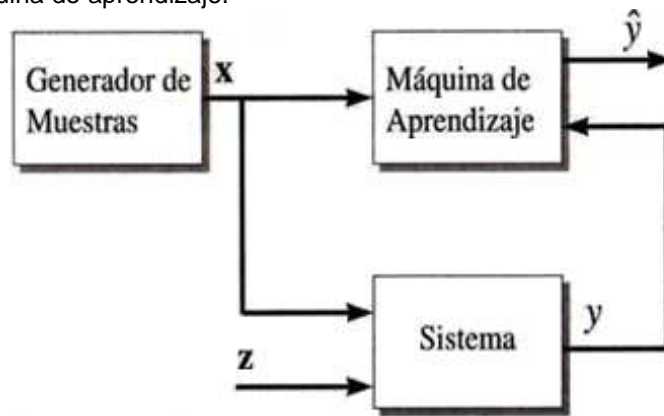
3.2.1. Generador este componente produce vectores aleatorios y determina una función de densidad probabilística $p(x)$.

⁸ PAJARES, Gonzalo; DE LA CRUZ, Jesús M. UNIVERSIDAD COMPLUTENSE DE MADRID. Visión por computador imágenes digitales y aplicaciones. Alfaomega Ra-Ma. 764 p.

3.2.2. Sistema se determinan las dependencias entre variables de entrada x y salida y a partir de muestras pasadas.

3.2.3. Máquinas de aprendizaje se obtienen los valores observados en x , pero esta obtención no es perfecta y se presenta incertidumbres z , de aquí se ve una dependencia estadística en los datos, esta máquina es capaz de interpretar un conjunto de funciones $f(x,w)$ donde w es un conjunto de parámetros para unir con un conjunto de funciones implementadas antes a máquina para el proceso de aprendizaje.

Figura 7. Una maquina de aprendizaje.



Fuente: GONZALO PAJARES, visión por computador imágenes digitales y aplicaciones.

Para el aprendizaje de una máquina existen dos tipos el aprendizaje supervisado y el no supervisado.

3.2.4. Supervisado para este aprendizaje se utiliza una relación conocida a partir de muestras conocidas.

3.2.5. El no supervisado para este aprendizaje solo hay muestras de entrada pero no hay de salida durante el aprendizaje.

Existen diferentes tipos de metodologías para el reconocimiento de patrones, como la utilización del clasificador Bayesiano, Redes Neuronales, métodos sintácticos, y reconocimiento basado en apariencia.

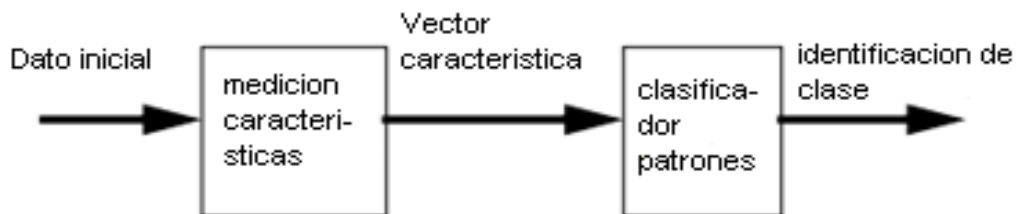
3.3. MACHINE VISION

Las *machine vision* [SNYDER y QI, 2004] es el proceso por el cual una computadora, percibe una información a través del ambiente por medio de una imagen, la procesa e identifica que elementos hay en la imagen.

Sus aplicaciones son diversas una de ellas el reconocimiento automático de objetivo e inspección industrial.

La *machine vision* incluye dos componentes, la clasificación de patrones y la medición de características.

Figura 8. Organización de un sistema de machine visión.



Organizacion de un sistema de machine vision

Fuente: Wesley E. QI, Hairong, Machine Vision, Cambridge University Press y modificado.

3.3.1. Medición de características: se centra en la imagen procesándola pixel por pixel, con base a los pixeles se puede medir una imagen o un parte de esas extrae un set de medición.

3.3.2. Clasificación de patrones: se define como el proceso de tomar una decisión con base a las medidas, donde esas medidas son obtenidas de un objeto desconocido, se define también como el proceso de asignación de incógnitas a una clase, y posee la misma definición que el reconocimiento de patrones con la diferencia que el reconocimiento se extiende a incluir el proceso de hacer las mediciones.

4. TEORIA DE LAS SMV

Las Máquinas de Soporte Vectorial son conformadas por un conjunto de teorías y conceptos que se explicaran en detalle en este capítulo, el concepto inicial es de un vector de características en un espacio de entrada, donde un conjunto de vectores es mapeado a una dimensión mayor, sea por la multiplicación de los productos internos o por un *Kernel* en el caso de los no separables, asegurando la minimización del espacio característico y la maximización del margen, donde los puntos más cercanos a la línea de decisión son los puntos de soporte; las Máquinas de Soporte Vectorial parten de la teoría de minimización de riesgo estructural la cual nos garantiza que el error en el diseño es mínimo.

Hay otra variante de las Máquinas de Soporte Vectorial llamada las RMV (*Relevance Machine Vector*), que usa el mismo concepto de las Maquinas de Soporte Vectorial pero para aplicarlo a la regresión lineal y no lineal.

4.1. HISTORIA DE LAS SVM⁹

Las Máquinas de Soporte Vectorial, es un término inventado por Vladimir Vapnik y su grupo de colaboradores en los laboratorios de la AT&T en 1992.

Los conceptos básicos de aprendizaje fueron introducidos desde 1960, en ese tiempo se utilizaba el concepto de hiperplano y el *Kernel* interpretado geométricamente como el producto interno en el espacio característico, al igual que la separabilidad para el reconocimiento de patrones, pero no fue hasta que en 1992 se unieran estos términos con el clasificador de margen máximo, construyendo los conceptos básico de las Máquinas de Soporte Vectorial y en 1995 fueron introducidos los clasificadores de margen blando.

4.2. PREAMBULO MATEMATICO

Las Máquinas de Soporte Vectorial tienen un fuerte soporte matemático entre los cuales se usan de distintos teoremas, se tratara lo más importante de cada una y sus aplicaciones en la construcción de las Máquinas de Soporte Vectorial.

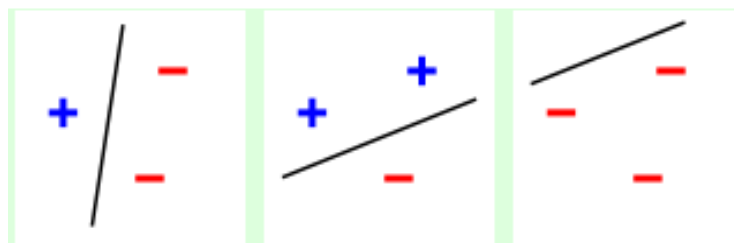
⁹ <http://kernelsvm.tripod.com/> visitada el día 22/06/12.

4.2.1. Dimensión Vapnik-Chervonenkis (VC)¹⁰: Esta teoría se aplica en la generalización de límite, donde en un conjunto finito de datos de entrenamiento, se debe encontrar una regla la cual reduzca el número de modelos que se puede tener en cuenta, se puede medir el tamaño efectivo de la clase F , usando un coeficiente de separación $S(F, n)$ u otro parámetro.

- Cuando n es pequeño o igual a un valor $VC(F)$, se tiene $S(F, n) = 2^n$ (F se dice que es el cortador de n puntos en ese caso).
- Si n es mayor a $VC(F)$ se tiene $S(F, n) < 2^n$.

El número de $VC(F)$ es llamada la dimensión (VC).

Figura 9. Separación de dos clases.



Fuente: http://es.wikipedia.org/wiki/Dimension_VC.

4.2.2. Condición Karush-Kuhn-Tucker (KKT)¹¹: Son las condiciones necesarias para optimización de los problemas no lineales con restricciones de desigualdad, este método es muy utilizado para resolución de los problemas de optimización de las Máquinas de Soporte Vectorial en los casos linealmente separables, son una generalización del método de los multiplicadores de Lagrange para restricciones.

El problema a optimizar.

$$\text{Min } f(x_1, x_2, \dots, x_n) \quad (4.2.1)$$

¹⁰ NORWAK. R, Lecture 19: The proof of the VC inequality.pdf, 17 de mayo del 2009, 6 p.

¹¹ departamento de matemáticas CSI/ITESM, Condiciones Karush-Kuhn-Tucker, 21 de abril del 2012, 9 p.

Con las restricciones.

$$g_1(x_1, x_2, \dots, x_n) \leq 0 \quad (4.2.2)$$

$$g_2(x_1, x_2, \dots, x_n) \leq 0 \quad (4.2.3)$$

$$\dots$$

$$G_m(x_1, x_2, \dots, x_n) \leq 0 \quad (4.2.4)$$

Ahora para toda desigualdad $g_i \leq 0$ a una restricción de igualdad con un valor si.

$$g_i \leq 0 \longrightarrow g_i + s_i^2 = 0 \quad (4.2.5)$$

Con la técnica de multiplicadores se construye la función.

$$F(x, \lambda, s) = f(x) + \sum_{i=1}^m \lambda_i (g_i + s_i^2) \quad (4.2.6)$$

Se busca la minimización de f.

- Se hacen cero parciales con respecto a las variables x_j ($j = 1, \dots, n$).

$$\frac{\partial F}{\partial x_j} = \frac{\partial f}{\partial x_j} + \sum_{i=1}^m \lambda_i \frac{\partial g_i}{\partial x_j} = 0 \quad (4.2.7)$$

- Se hacen cero parciales con respecto a las variables λ_i ($i = 1, \dots, m$).

$$\frac{\partial F}{\partial \lambda_j} = g_j + s_j^2 = 0 \leftrightarrow g_j \leq 0 \quad (4.2.8)$$

- Se hacen cero las parciales con respecto a las variables s_i ($i = 1, \dots, m$).

$$\frac{\partial F}{\partial s_i} = 2\lambda_i s_i = 0 \leftrightarrow \lambda_i s_i = 0 \leftrightarrow \lambda_i g_i = 0 \quad (4.2.9)$$

Las condiciones deben satisfacer los puntos que minimizan la función respecto a las restricciones.

4.2.3. Teorema de Mercer¹²: para la transformación de un espacio de entrada a un espacio de alta dimensionalidad, de un conjunto de datos de entrenamiento no lineal, el cual se tiene un conjunto X , y K que es una función continua tal que el operador.

¹² VALLE VIDA, Carlos, Vector de soporte y métodos de kernel, universidad técnica Federico santa maría, abril 2009, 94 p.

$$T_K: L_2(X) \leftrightarrow L_2(X) \quad (4.2.10)$$

$$(T_K f)(\cdot) = \int_X K(\cdot, x) f(x) dx \quad (4.2.11)$$

Donde con los valores positivos.

$$\int_{X \times X} K(x, z) f(x) f(z) dx dz \geq 0, \forall f \in L_2(X) \quad (4.2.12)$$

El valor $K(x, z)$ se expande en una serie uniforme de $X \times X$, en términos de T_k , donde las funciones propias Φ_j , normalizando $\|\Phi_j\| = 1$ y los valores de Lagrange positivos mayores e iguales a cero.

$$K(x, z) = \sum_{j=1}^{\infty} \lambda_j \varphi_j(x) \varphi_j(z) \quad (4.2.13)$$

Donde la imagen de un vector x , mapeado a través de un *Kernel*.

$$\sum_{j=1}^{\infty} \sqrt{\lambda_j} \varphi_j(x) \quad (4.2.14)$$

El teorema de *Mercer* es la base para la que los vectores de característica, evitan ser mapeados a un espacio de entrada a través de productos escalares, aumentando la dimensionalidad y el costo computacional, con este teorema solo se necesita conocer el vector, para luego transfórmalos a un espacio de alta dimensión.

4.3. MINIMIZACIÓN DE RIEGO ESTRUCTURAL¹³

Partiendo del modelo de máquina de aprendizaje de la figura 7, se debe garantizar que no haya pérdida o discrepancia en el modelo, por tal razón se busca minimizar el riesgo en el modelo.

$$R(\alpha) = \int L(y, f(x, \alpha)) dF(x, y) \quad (4.3.1)$$

Donde $L(y, f(x, \alpha))$ es el término de pérdida, entre la respuesta de y para obtener una entrada x y la salida de $f(x, \alpha)$, proveniente de la máquina de aprendizaje, donde $F(x, y)$ es la función de probabilidad de distribución, el objetivo de la función $f(x, \alpha_0)$ es minimizar el riesgo funcional.

Hay tres tipos de problemas de aprendizaje: el problema de reconocimiento de patrones, la estimación de la regresión y la estimación de densidad, donde solo se abordara el problema del reconocimiento de patrones.

¹³ VAPNIK, Vladimir, Springer, The Nature of statistical learning theory 1999, 334 p.

4.3.1. Reconocimiento de patrones: Solo se tiene dos valores $y = \{0,1\}$, consideradas en la función de perdida.

$$L(y, f(x, \alpha)) = \begin{cases} 0 & \text{if } y = f(x, \alpha) \\ 1 & \text{if } y \neq f(x, \alpha) \end{cases} \quad (4.3.2)$$

La función (4.3.1) determina la probabilidad de diferentes respuestas dadas por el sistema, con la función donde $f(x, \alpha)$, donde distintas respuestas pueden ser llamadas clasificación de error.

El problema es encontrar una función que minimiza la clasificación de error, donde $F(x,y)$ es desconocido.

4.3.2. Minimización del riesgo empírico (*empirical risk minimization, ERM*): Partiendo de la minimización de riesgo funcional general.

$$R(\alpha) = \int Q(z, \alpha) dF(z), \quad \alpha \in \Lambda \quad (4.3.3)$$

Es remplazado por.

$$R_{emp}(\alpha) = \frac{1}{l} \sum_{i=1}^l Q(z_i, \alpha) \quad (4.3.4)$$

Que es la función de riesgo funcional, con el set de entrenamiento.

$$z_1, \dots, z_l \quad (4.3.5)$$

Este principio es llamado la minimización de riesgo empírico, principio inductivo, donde se define los procesos de aprendizaje para máquinas de aprendizaje basada en este principio, Las ERM en general es un método clásico para solucionar específicos problemas de aprendizaje.

4.3.3. Minimización de riesgo estructural (*structural risk minimization, SRM*): controlar la generalización de las máquinas de aprendizaje, es lo principal en el principio de minimización de riesgo.

Para comenzar se parte usando un pequeña muestra, usando la generalización de una máquinas de aprendizaje, con un set de límites en una función no negativa.

$$R(\alpha_l) \leq R_{emp}(\alpha_l) + \frac{B\varepsilon}{2} \left(1 + \sqrt{1 + \frac{4R_{emp}(\alpha_l)}{B\varepsilon}}\right) \quad (4.3.6)$$

Con los set de una función ilimitada:

$$R(\alpha_l) \leq \frac{R_{emp}(\alpha_l)}{(1-a(p)t\sqrt{\varepsilon})} = \pi r^2 \quad (4.3.7)$$

$$a(p) = \sqrt[p]{\frac{1}{2} \left(\frac{p-1}{p-2}\right)^{p-1}} \quad (4.3.8)$$

$$\varepsilon = 2 \frac{\ln N - \ln n}{\varepsilon} \quad (4.3.9)$$

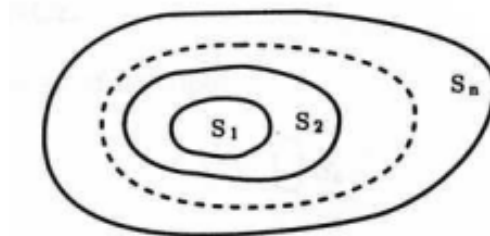
Se necesita reducir el lado derecho de las desigualdades (4.3.6) y (4.3.7) simultáneamente, pero el primer término (4.3.6) depende de una función específica de un set de funciones y el segundo depende de una dimensión VC, por tal razón para minimizar los términos derechos dentro del límite de riesgo en ambos términos se necesita hacer una dimensión VC que controle las variables.

Este es el principio de minimización de riesgo estructural (SRM), minimizar el riesgo estructural para ambos términos.

Se tiene un set S de funciones $Q(z, \alpha)$ donde $\alpha \in \lambda$, con una estructura en una función de sub set anidados $S_k = \{ Q(z, \alpha), \alpha \in \lambda \}$.

$$S_1 \subset S_2 \subset \dots \subset S_n \quad (4.3.10)$$

Figura 10. estructura de un set de funciones anidados en un sub set de funciones.



Fuente: VAPNIK, Vladimir, Springer, The Nature of statistical learning theory.

Para que los elementos de la estructura se cumplan, se debe tener en cuenta dos propiedades.

- La dimensión VC (h_k), para cada set S_k de la función es finita.

$$h_1 \leq h_2 \dots \leq h_n \dots \quad (4.3.11)$$

- Cualquier elemento de S_k de la estructura debe contener.
 - Un set con todos las funciones limites.

$$0 \leq Q(z, \alpha) \leq B_k, \quad \alpha \in \Lambda_k \quad (4.3.12)$$

- o O un set que cumplan la desigualdad

$$\sup_{\alpha \in \Lambda_k} \frac{(\int Q^p(z, \alpha) dF(z))^{\frac{1}{p}}}{\int Q(z, \alpha) dF(z)} \leq t_k, \quad p > 2 \quad (4.3.13)$$

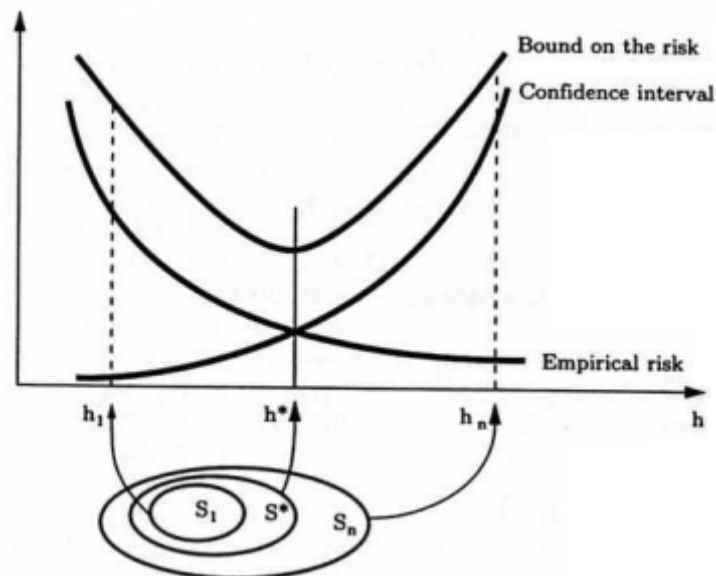
Para un par (p, t_k)

Esta estructura es una estructura admisible.

Las SRM toman la función $Q(z, \alpha)$ para minimizar el riesgo empírico en los sub set S_k que garantizan que el riesgo sea mínimo.

El principio de SMR se define [VAPNIK, 1999] “la compensación entre la calidad de la aproximación para el dato dado y la complejidad de la función aproximación”, si un sub set aumenta, el mínimo del riesgo empírico decrece.

Figura 11. El límite de riesgo



Fuente: VAPNIK, Vladimir, Springer, The Nature of statistical learning theory

El límite de el riesgo es la suma de el riesgo empírico más un *confidence interval* VC, el riesgo empírico decrece con la adición de elementos de la estructura mientras en el confidencial incrementa.

4.3.4. Confidence interval: este es el segundo término a la derecha de la desigualdad

$$R(\lambda) \leq R_{emp}(\lambda) + \sqrt{\left(\frac{h(\log(\frac{2l}{h})+1) - \log(\frac{n}{4})}{l}\right)} \quad (4.3.14)$$

Donde h es de la VC dimensión y es una medida de la capacidad de la máquina de aprendizaje.

4.4. INTRODUCCIÓN A LOS VECTORES DE SOPORTE

La representación grafica de los modelos de conocimiento se realizan a través de diagramas de ejes X, Y, los datos en esta colección de datos son llamados set de entrenamiento, donde un dato es representado por un vector que contiene características de un objeto, se necesita encontrar la distancia euclidiana más cercana entre un límite de decisión y dos clases.

Las características de un objeto es un set de números que se definen para contener las propiedades de una imagen, el clasificador debe estar desarrollado con un método que se implementa una regla de decisión.

4.4.1. ejemplo de una máquinas de vector de características: [SNYDER y QI, 2004], las máquina de soporte inician de un vector de datos de N elementos, donde la dimensionalidad es N y se busca reducirla.

Para reducir la dimensionalidad en general es d dimensiones, para $c-1$ dimensiones, donde C son diferentes clases con un set de entrenamiento X_i para ejemplos de cada clase, este tipo de aprendizaje es supervisado.

Paras las SVM se define una matriz de dispersión de clase, la cual es la medida total de la distancia entre las clases y el total de las medidas.

$$S_W = \sum_{i=1}^c S_i \quad (4.4.1)$$

Donde

$$S_i = \sum_{x \in X_i} (x - \mu_i)(x - \mu_i)^T \quad (4.4.2)$$

Donde μ_i es la media de la clase i , si la media varia se utiliza

$$\mu_i = \frac{1}{n_i} \sum_{x \in X_i} x \quad (4.4.3)$$

Y se define la matriz de dispersión.

$$S_B = \sum_{i=1}^c n_i (\mu_i - \mu)(\mu_i - \mu)^T \quad (4.4.4)$$

Donde μ_i es la media de todos los puntos del set de entrenamiento y n_i es el número de muestras de la clase i .

Donde se desea encontrar una proyección del vector de dato x ortogonal al vector y .

$$y = W_x \quad (4.4.5)$$

y es dimensión menor que x , las clases con la mejor separación serán las proyectadas, debido a que el espacio dimensional es d en un espacio $c-1$, esto es una función discriminante lineal, donde se trata de encontrar relaciones lineales entre las variables que mejor discriminen los grupos de objetos dados.

$$Y = W_2^T \quad (4.4.6)$$

y_i es el componente del vector, w_i son las columnas de la matriz W .

4.5. GENERALIZACIÓN DEL LIMITE

Un algoritmo puede ser generalizado y más cuando se usa espacios de alta dimensionalidad en [SEBE, COHEN, GARG y HUANG, 2005], se muestra el caso de las Máquina de Soporte Vectorial, cuando se aplican datos a un espacio de alta generalización es común una relativa distorsión entre la distancia de los puntos proyectados, si se presenta una diferencial entre la dimensionalidad, esta distorsión incurre en una pequeña cantidad de error respecto al espacio original.

Se introduce un dato dependiente, de acuerdo a una proyección D , y esperando una cantidad de errores introducida en el clasificador H , en una proyección en una K proyección.

$$a_k(D, h) = \int_{x \in D} u(x) dD \quad (4.5.1)$$

Donde

$$u(x) = \min \left(3 \exp \left(- \frac{k(v(x))^2}{8(2+|v(x)|)^2} \right), 1 \right) \quad (4.5.2)$$

$v(x)$ es la distancia entre x y el clasificador de hiperplano definido por H , en un clasificador lineal D .

$$P(D, h) = (a_1(D, h), a_2(D, h), \dots) \quad (4.5.3)$$

Este es el perfil de proyección de D, pero esta generalización depende la estimación del perfil de proyección mientras se proyecta a otra dimensión efectiva.

4.5.1. Margen de distribución basada en límites: Otra opción es el uso de márgenes, basado en límites donde el límite depende la distribución geométrica de las distancias de los puntos del clasificador más extremos, esta distancia es el margen de distribución de los datos, lo que significa que mientras los datos más lejos estén, este un clasificador óptimo, solo unos pocos puntos determinan el margen más cercanos a ellos.

Se tiene un clasificador h , basado en el signo de $v(x) = h^T x$, desde que h y x estén normalizadas $|v(x)|$, a través de la distancia geométrica entre x y el hiperplano ortogonal en h , que pasen a través del origen, teniendo una distribución de datos x , esto induce una distribución en la distancia del hiperplano inducido por h , esto se refiere al margen de distribución.

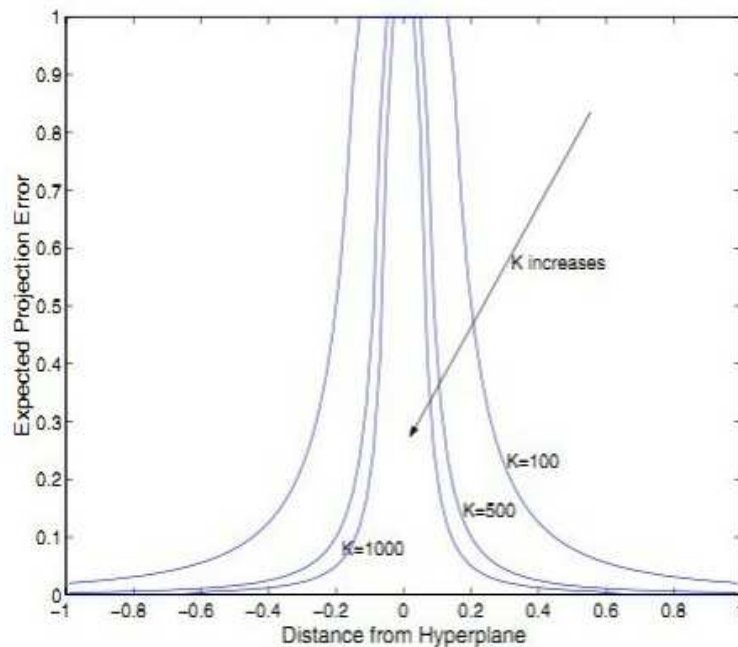
La proyección aleatoria es la imagen debajo de una proyección de datos que están lejos de h , en un espacio original que está lejos de esta imagen en la proyección del espacio (K dimensión), el error radica cuando los puntos de datos de la imagen no consisten con la imagen h , este es un error empírico proyectado, Si decrece la dimensión en el espacio de proyección este implica un decrecimiento de la dimensión VC (Vapnik-Chervonenkis), se debe tener un óptimo límite el cual haya un balance entre ambos cuantificadores y escoger un K , en el espacio de proyección para que el error generalizado disminuya.

Para estimar la probabilidad de error en una dimensión K en la imagen de x , se tiene el punto x , la distancia $v(x)$ de n dimensión del hiperplano.

$$\min \left(3 \exp \left(-\frac{(v(x)^2 K)}{8(2+|v(x)|)^2} \right), \frac{2}{kl^2}, 1 \right) \quad (4.5.4)$$

La función anterior es la distancia del hiperplano para diferentes valores de k .

Figura 12. Contribución de los puntos de datos a la generalización del error.



Fuente: N. Sebe, IRA Cohe, ASHUTOSH Garg, Machine Learning in computer vision, Springer.

En la figura 12, se puede ver que la final todos los puntos contribuirán al error, se puede apreciar que el error disminuye exponencialmente rápido si la función dista del hiperplano.

4.6. SUPPORT VECTOR MACHINE (SVM)

Este concepto se basa en la minimización de riesgo estructural, donde un set de datos se optimiza maximizando el margen, donde el margen es la distancia del punto más cercano al límite de decisión [SNYDER y QI, 2004].

Las SVM no poseen parámetros fijos, pero el número de datos de entrenamiento es el que controla el modelo como se explicó en el tema de minimización del riesgo estructural (SRM), para una buena generalización del modelo hay dos enfoques.

- Elegir una estructura adecuada (orden polinomio), manteniendo el error de estimación fijo (varianza del modelo), reducir el mínimo de error de entrenamiento (riesgo empírico).

- Mantener el valor de error de entrenamiento (riesgo empírico) de cero a un nivel aceptable y minimizar el intervalo de confianza.

En la construcción de las máquinas de aprendizaje hay que saber si se utilizara un modelo de bajo ajuste o un sobre ajuste de los datos de entrenamiento, el modelo estructural ideal es el que coincida con la capacidad de la máquina de aprendizaje, en un modelo básico y el error de riesgo funcional.

$$R = \sum_{i=1}^l L_{\varepsilon}(\text{Datos más cercanos}) + \Omega(l, h) (\text{Capacidad de la máquina}) \quad (4.6.1)$$

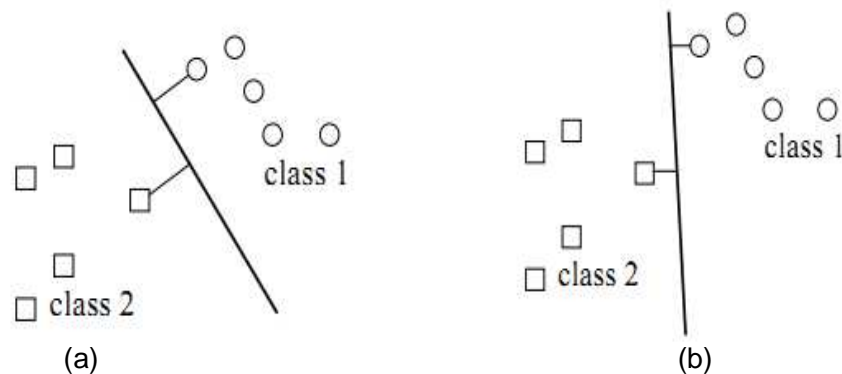
Donde L_{ε} es la función de de perdida de las SVM, por lo general asume valores entre 0 y 1, h es un de la VC dimensión y Ω es la capacidad de la máquina.

Las SVM busca la optimización de parámetros tanto como globales como locales, por tal razón se usa extensivamente los multiplicadores de Lagrange.

Las Máquina de Soporte Vectorial se dividen en lineales donde el conjunto de clases está perfectamente separado y las no lineales las cuales las clases no tienen un límite definido.

4.6.1. SVM lineales: El espacio de entrada se divide por un hiperplano en dos segmentos, las SVM comienza con sets de entrenamiento linealmente separables.

Figura 13. Ejemplo de selección hiperplano, a) una buena elección con un hiperplano con un amplio margen, b) una pobre elección con un hiperplano donde el margen es pequeño.



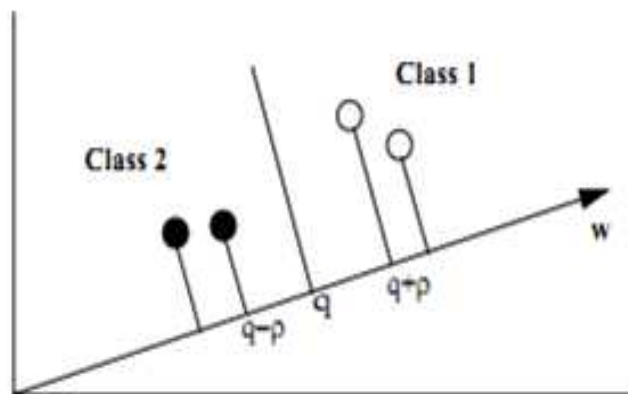
Fuente: Wesley E. Qi, Hairong, Machine Vision, Cambridge University Press.

La figura 13, nos muestran dos clases donde hay puntos cercanos de la clase 1 y la clase 2 al hiperplano, esta distancia entre ambos puntos es el margen, Para las reglas de decisión lineales se usa un dato x , esta muestra se proyecta a un vector unidad W , la regla de decisión para la clase 1 es:

$$w^T x - q > 0 \quad (4.6.2)$$

Donde q es una constante, los puntos más cercanos a la línea de decisión se le realiza una proyección ortogonal, x_1 punto de la clase 1 y x_2 , punto de la clase 2, donde se realiza una proyección ortogonal a w para determinar quien está más cerca de la línea de decisión.

Figura 14. Ortogonalidad de los puntos.



Fuente: Wesley E. Qi, Hairong, Machine Vision, Cambridge University Press.

En la Figura 14 se muestra que las clases se proyectan ortogonalmente a la línea W , así se puede ver los puntos de entrenamiento más cercano a la línea q , con los puntos x_1, x_2 y el mínimo positivo p .

$$\varphi^T x_1 = q + p \quad \varphi^T x_2 = q - p \quad (4.6.3)$$

Ahora se define el valor L como el set de entrenamiento de una clase.

$$L1 \text{ son los puntos de la clase 1, } \varphi^T x_1 - q > p \quad (4.6.4)$$

$$L2 \text{ los puntos de la clase 2, } \varphi^T x_1 - q < p \quad (4.6.5)$$

Ahora se hallaran dos cosas:

- los puntos los más cercanos a q , a estos se le llaman vectores de soporte.
- Un vector ortogonal que proyecte los puntos de soporte la mayor separación posible.

Ahora se mira el comportamiento de las clases en el espacio de entrada, se define $\varphi = w / \|w\|$, en el vector unidad en la dirección w y se reemplaza en la ecuación (4.6.3).

$$\left(\frac{w}{\|w\|}\right)^T x_1 - q \geq p \quad \left(\frac{w}{\|w\|}\right)^T x_2 - q \leq p \quad (4.6.6)$$

Se organiza la ecuación.

$$w^T x_1 - q\|w\| \geq P\|w\| \quad w^T x_2 - q\|w\| \leq P\|w\| \quad (4.6.7)$$

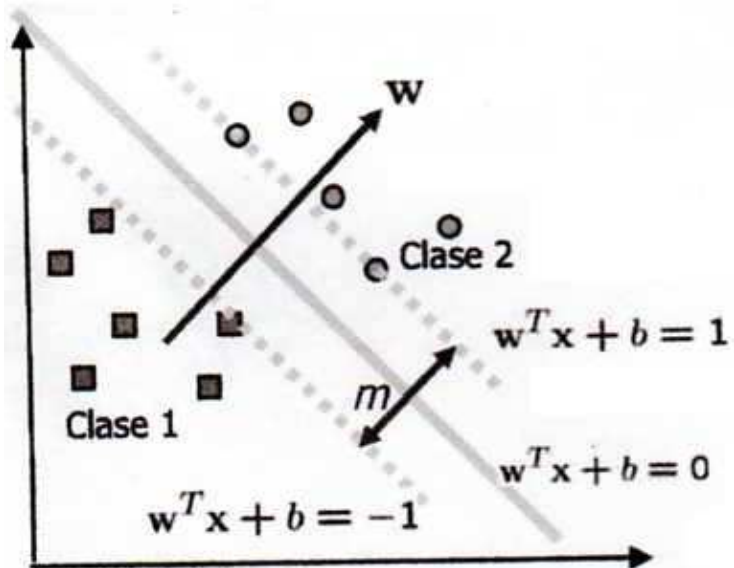
Definimos la constante $b = -q\|w\|$, donde se adiciona a W , la magnitud tiene una particular propiedad.

$$\|w\| = \frac{1}{p} \quad (4.6.8)$$

Ahora tenemos la ecuación que describe el comportamiento de los puntos.

$$w^T x_1 + b \geq 1 \quad w^T x_2 + b \leq 1 \quad (4.6.9)$$

Figura 15. Distribución de las clases y su regla de decisión.



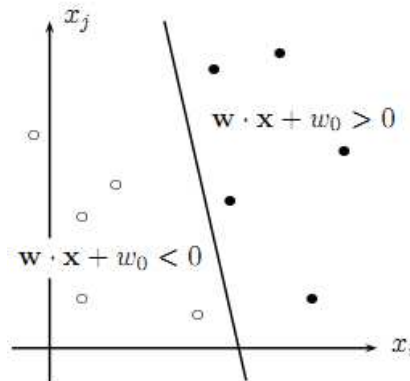
Fuente: Maquinas de soporte vectorial (SVM), Scientia et Techica.

4.6.2. Minimización del vector proyección y maximización del margen ^{14, 15}: en [LIPO, 2005], se explica esta teoría, donde se busca la minimización del vector proyección w donde su magnitud sea mínima, y que el margen de separación sea el máximo, se tiene un set de entrenamiento compuesto de la forma (x_i, y_i) para $i = 1, \dots, n$, donde y_i es ± 1 , se usa la siguiente función.

$$\tilde{y}_i = y_i(w \cdot x + w_0) \quad (4.6.10)$$

Que es conocido, como la función de margen y esta compuesta por, w el peso del vector, w_0 el set de entrenamiento, x el dato, donde $\tilde{y}_i > 0$ son puntos correctamente clasificados, los mal clasificados están en distancia negativa al hiperplano.

Figura 16. Lados del hiperplano.



Fuente: Gaussian process for machine learning.

Luego se considera un margen geométrico definido como, $y_D = \min y_i$ teniendo la separación canónica del hiperplano es el margen $1/\|w\|$.

$$\text{Minimiza} \quad \frac{1}{2} \|w\|^2 \quad w, w_0 \quad (4.6.11)$$

$$\text{Sujeto a} \quad y_i(w \cdot x_i + w_0) \geq 1 \quad \text{Para todo } i=1, \dots, n \quad (4.6.12)$$

¹⁴ RASMUSSEN, Carl Edward, WILLIAMS, Christopher, Cambridge, Gaussian Processes for machine Learning, The MIT press 2006, 226 p.

¹⁵ BISHOP. M, Chistopher, Cambridge, Pattern recognition and machine learning, 2006, 701 p.

la minimización se realiza bajo la Minimización vectorial de margen duro (*hard-margin*), para la optimización de este problema usamos los multiplicadores de Lagrange $\alpha = (\alpha_1, \dots, \alpha_m)$

$$Q(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^M \alpha_i \{y_i (w^T x_i + b) - 1\} \quad (4.6.13)$$

Para la solución óptima se minimiza con respecto a w y b , pero maximiza con respecto a α , aquí se aplica la condición Karush-Kuhn-Tucker (KKT).

$$\frac{\partial Q(w, b, \alpha)}{\partial w} = 0 \quad (4.6.14)$$

$$\frac{\partial Q(w, b, \alpha)}{\partial b} = 0 \quad (4.6.15)$$

$$\alpha_i \{y_i (w^T x_i + b) - 1\} = 0 \quad \text{para } i = 1, \dots, M \quad (4.6.16)$$

$$\alpha_i \geq 0 \quad \text{para } i = 1, \dots, M \quad (4.6.17)$$

En la ecuación (4.6.14) y (4.6.15) se reducen los términos en w y b , en (4.6.13) se igualan a cero y en la (4.6.17) que todos los multiplicadores de Lagrange serán positivos, el set de entrenamiento con datos X_i con α_i , son llamados vectores de soporte, se usa la ecuación (4.6.13), y se sustituye con (4.6.18) y (4.6.19), se obtiene.

$$w = \sum_{i=1}^M \alpha_i y_i x_i \quad (4.6.18)$$

$$\sum_{i=1}^M \alpha_i y_i = 0 \quad (4.6.19)$$

Ahora se usa la condición de representación doble para la maximización de la margen.

$$Q(\alpha) = \sum_{i=1}^M \alpha_i - \frac{1}{2} \sum_{i,j=1}^M \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (4.6.20)$$

Con las restricciones

$$\sum_{i=1}^M y_i \alpha_i = 0 \quad \alpha_i \geq 0 \quad \text{para } i = 1, \dots, M \quad (4.6.21)$$

Si la solución existe entonces es linealmente separable y también existe una solución óptima global, si los valores de el primer y segundo término de la función objetivo coinciden con la solución óptima, si existe es llamada la dualidad de ceros, ahora se toma la función de decisión donde los datos son asociados con un α positivo.

$$D(x) = \sum_{i \in S} \alpha_i y_i x_i^T x + b \quad (4.6.22)$$

Donde S , es el índice de los vectores de soporte, usando de nuevo las condiciones de KKT, se puede obtener el valor de la constante b que es el valor umbral.

$$b = y_i - w^T x_i \quad (4.6.23)$$

Para tener un mejor cálculo se usa un número mayor de vectores de soporte en la ecuación.

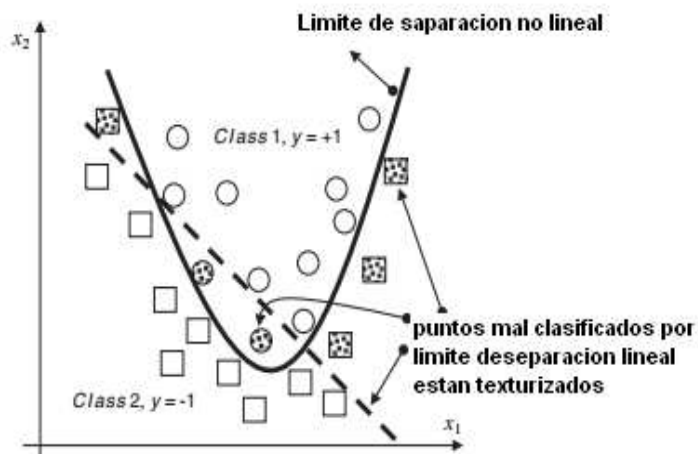
$$b = \frac{1}{|S|} \sum_{i \in S} (y_i - w^T x_i) \quad (4.6.24)$$

Un dato desconocido puede ser clasificado dentro de la clase 1 si $D(x) > 0$ o pertenece a la clase 2 si $D(x) < 0$, la ecuación proyección óptima del vector es la ecuación (4.6.18), los puntos más cercanos son los vectores de soporte, pero no todos los puntos son usados para dar la solución final.

Se debe tener en cuenta que a mayor número de set de entrenamiento la dimensión aumenta, por tal razón se debe elegir un set a priori el cual se disminuya la complejidad computacional, la única manera de que los puntos de entrenamiento x_i y los puntos prueba x_0 sean calculados es que estén en termino de producto interno, pero usando el truco del *Kernel* así se evita el uso de productos internos y se obtiene un resultado equivalente al espacio característico.

4.6.3. SVM no lineales: En los casos reales los datos no son linealmente separables, se procura que el error de entrenamiento sea mínimo y que el margen sea máximo [LIPO, 2005].

Figura 17. SVM no lineal.

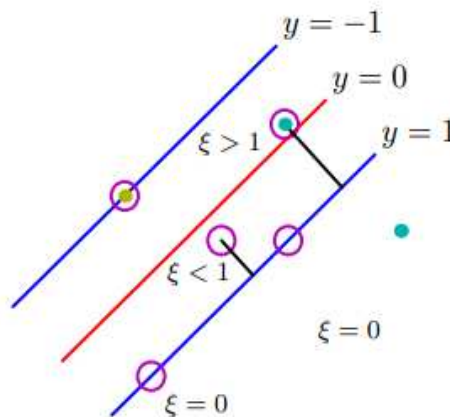


Fuente: Support vector machine theory and applications studies in fuzzines and soft computing y modification.

4.6.2.1. Margen suave (Soft – margin): en *hard-margin* se tiene un conjunto de datos linealmente separable, pero cuando los datos no son separables no existe una solución factible, el *soft-margin* [LIPO, 2005], [BISHOP, 2006], se aplica para clases que no sean separables y se usan las variables de holgura no negativas ξ_n para permitir la separabilidad.

$$t_n y(x_n) \geq 1 - \xi_n \quad n = 1, \dots, N \quad (4.6.25)$$

Figura 18. Variables de holgura.



Fuente: M. BISHOP Christopher, Cambridge, Pattern recognition and machine learning.

En ciertos casos la distancia del límite de decisión a los puntos de entrenamiento son iguales a los puntos de soporte, este es un problema ya que genera una pobre generalización, para corregir esto se usa una variable de holgura $\xi_n \geq 0$ donde $n = 1, \dots, N$, para cada dato de entrenamiento, el cual incrementa la distancia del límite, permitiendo si un dato es mal clasificado el error es cero y si es correcto se optimiza el parámetro para maximizar el margen.

Se define $\xi_n = 0$ por los puntos de datos, si están en el lado correcto del margen límite y $\xi_n = |t_n - y(x_n)|$ para otros puntos, así para puntos de dato en el límite de decisión $y(x_n) = 0$ puede tener $\xi_n = 1$ y puntos $\xi_n > 1$ pueden ser mal clasificados.

Con las variables de holgura con restricciones satisface $\xi_n \geq 0$, los datos de puntos $\xi_n = 0$ son correctamente clasificados, si están en el margen o en un lugar correcto del margen puntos que se encuentran $0 < \xi_n \leq 1$ dentro del margen, pero el correcto sitio del límite de decisión, donde los puntos $\xi_n > 1$ están en el lado equivocado del límite de decisión y son mal clasificados, pero esta holgura

permite que algunos puntos de entrenamiento puede ser mal clasificados, ya que los errores aumenta linealmente con ξ .

Para la optimización de este margen se necesita minimizar.

$$Q(w, b, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^M \xi_i^p \quad (4.6.26)$$

Sujeto a las restricciones

$$y_i(w^T x_i + b) \geq 1 - \xi_i \quad \text{para } i = 1, \dots, M \quad (4.6.27)$$

Se realiza el mismo procedimiento del caso anterior con la diferencia de dos variables mas ξ y β , donde C es el parámetro de la margen, obtenemos.

$$Q(w, b, \xi, \alpha, \beta) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^M \xi_i - \sum_{i=1}^M \alpha_i (y_i(w^T x_i + b) - 1 + \xi_i) - \sum_{i=1}^M \beta_i \xi_i \quad (4.6.28)$$

Aplicamos la condición KKT, para este caso

$$\frac{\partial Q(w, b, \xi, \alpha, \beta)}{\partial w} = 0 \quad (4.6.29)$$

$$\frac{\partial Q(w, b, \xi, \alpha, \beta)}{\partial b} = 0 \quad (4.6.30)$$

$$\frac{\partial Q(w, b, \xi, \alpha, \beta)}{\partial \xi} = 0 \quad (4.6.31)$$

$$\alpha_i (y_i(w^T x_i + b) - 1 + \xi_i) = 0 \quad \text{para } i = 1, \dots, M \quad (4.6.32)$$

$$\beta_i \xi_i = 0 \quad \text{para } i = 1, \dots, M \quad (4.6.33)$$

$$\alpha_i \geq 0, \xi_i \geq 0, \beta_i \geq 0 \quad \text{para } i = 1, \dots, M \quad (4.6.34)$$

Al igual que el caso anterior aplicamos una reducción de la ecuación principal (4.6.28) con respecto a (4.6.29) y (4.6.31), donde se obtiene las siguientes ecuaciones.

$$w = \sum_{i=1}^M \alpha_i y_i x_i \quad (4.6.35)$$

$$\sum_{i=1}^M \alpha_i y_i = 0 \quad (4.6.36)$$

$$\alpha_i + \xi_i = C \quad \text{para } i = 1, \dots, M \quad (4.6.37)$$

Aquí se sustituye la ecuación (4.6.35) y (4.6.36) en la principal (4.6.28), donde se aplica el problema doble para llegar a la ecuación a maximizar.

$$Q(\alpha) = \sum_{i=1}^M \alpha_i - \frac{1}{2} \sum_{i,j=1}^M \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (4.6.38)$$

Sujeto a las restricciones

$$\sum_{i=1}^M y_i \alpha_i = 0 \quad C \leq \alpha_i \leq 0 \quad \text{para } i = 1, \dots, M \quad (4.6.39)$$

En este caso los vectores de soporte pueden exceder C, en el caso de α_i , se presentan tres casos.

- $\alpha_i = 0$ y $\xi_n = 0$ entonces x es correctamente clasificado.
- $0 < \alpha_i < C$ y $\xi_n = 0$ o $\alpha_i = C$ y $\xi_n > 0$, y $y_i(w^T x_i + b) - 1 + \xi_i = 0$ y que $y_i(w^T x_i + b) = 1$, es x un vector de soporte.
- $\xi_n > 1$ entonces x está mal clasificado.

la función de decisión es la misma del *hard-margin* (4.6.22) y el b es el mismo (4.6.24).

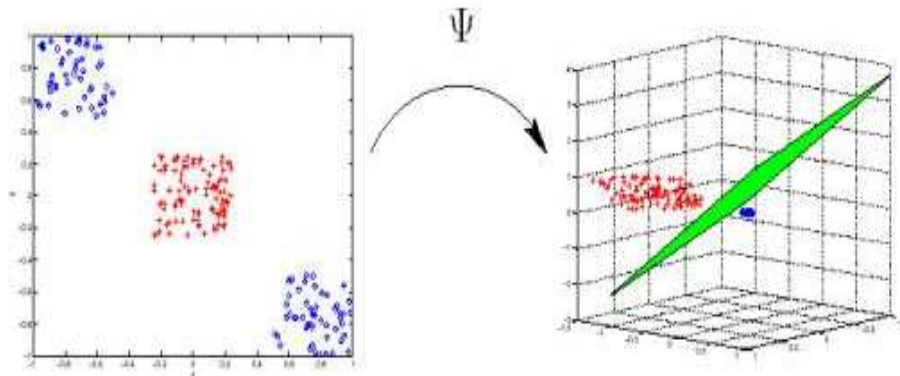
No importa si se realiza un tratamiento a las SVM *soft-margin* o *hard-margin*, sigue siendo un problema de programación cuadrática, Como este es un problema de multiplicación de matrices se tiene una complejidad $O(n^3)$, el objetivo de las SVM, es encontrar el hiperplano de separación en el espacio de características, ese espacio es transformar a una alta dimensión un espacio de entrada X.

4.6.2.2. Transformador de altas dimensiones: con el uso de un transformador no lineal, de alta dimensión para buscar que los datos entrantes por el espacio de entrada sea separables.

$$y_i = \vartheta(x_i) \quad (4.6.40)$$

Por ejemplo, si $x = [x_1, x_2]^T$ que está en una dimensión, si se aumenta la dimensión es 2 es $y_i = [x_1^2 x_2^2 x_1 x_2 x_2 1]^T$ esta expansión incrementa la dimensionalidad del espacio, incrementando el promedio de clases que pueden ser separables en un alto espacio de dimensión.

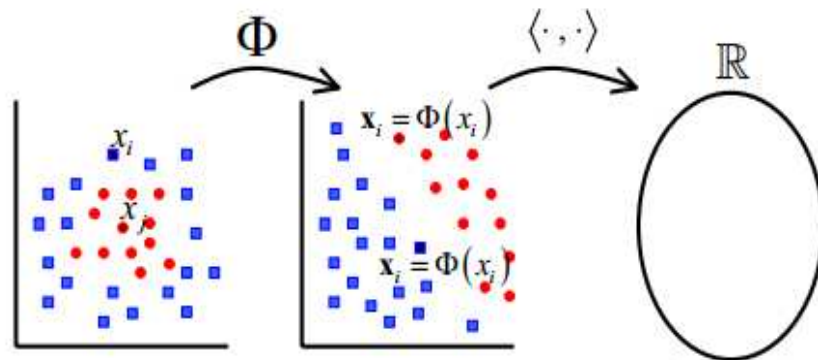
Figura 19. Espacio de características a un espacio de alta dimensión.



Fuente: <http://kernelsvm.tripod.com/>

Las SVM no solo se busca la minimización de las clases, y la transformación en un espacio mayor, también la búsqueda de un hiperplano de óptimo donde no es necesario conocer los vectores en sí mismos, solo se necesitan los valores escalares, no siendo necesario mapearlos a una alta dimensión y tomar su producto interno.

Figura 20. Procedimiento de mapeo a alta dimensión y luego hallar producto punto.



Fuente: GARCÍA días ELKIN Eduardo, tesis maestría en ingeniería electrónica, Boosting support vector machines.

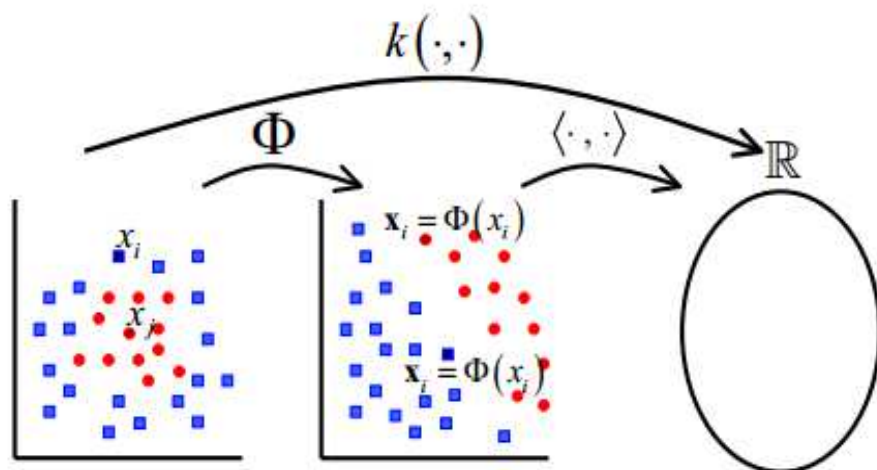
Como se ve en la figura 20, el procedimiento de mapeo a un espacio de alta dimensión y luego producto interno, este proceso es costoso computacionalmente, \mathbb{R} es un conjunto no vacío.

4.6.2.3. La maldición de la dimensionalidad: [LIPO, 2005], hay que tener en cuenta que cuando se aumenta la dimensión el costo computacional también

aumenta, por ejemplo tenemos una superficie de decisión correspondiente a un polinomio de grado 2, en un espacio de características $f = n(n+3)/2$, y se toma un espacio de entrada de dimensión 256, la dimensión del espacio de características es de $f = 33152$ si se aumenta el grado de polinomio 4 o 5 con el mismo espacio de dimensión, el hiperplano de decisión puede tener un espacio de dimensión hasta de un billón, para evitar esta explosión de dimensionalidad se hace que los puntos de entrenamiento aparezcan en forma de producto escalar $x_i^T x_j$,

4.6.2.4. Truco del Kernel^{16, 17, 18}: permite calcular el producto escalar sin necesidad de la representación explícita de los vectores en el espacio de característica, pero hay que seguir una serie de pasos para llegar a la aplicación de un *Kernel*.

Figura 21. Solucionando el problema de optimización usando Kernel.



Fuente: GARCÍA días ELKIN Eduardo, tesis maestría en ingeniería electrónica, Boosting support vector machines.

¹⁶ MARTÍNEZ Manuel Ramón, , introducción a los métodos Kernel, Universidad autónoma de Madrid, 29 de abril de 2008, 31 p.

¹⁷ ATIENZA, Felipe Alonso, tesis doctoral, estudio de los mecanismos de las arritmias cardiacas mediante modelado y procesado robusto digital de señal, Universidad Carlos III de Madrid, mayo de 2008, 268 p.

¹⁸ GARCÍA DÍAS, Elkin Eduardo, tesis maestría en ingeniería electrónica, Boosting support vector machines, universidad de los andes, 2005, 59 p.

Para usar el teorema de Mercer, se parte de $\Phi(x)$ que es una transformación a un espacio Hilbert de dimensión superior.

$$R^d \rightarrow H \quad (4.6.41)$$

$$x \rightarrow \varphi(x) \quad (4.6.42)$$

Para llegar a un espacio de Hilbert, se parte de un espacio vectorial dotado con productos puntos, este es un espacio pre-Hilbert.

$$g(\cdot) = \sum_{j=1}^m \beta_j k(\cdot, x_j) \quad (4.6.43)$$

Para convertirlo a un espacio Hilbert, se completa usando la secuencia de Cauchy.

$$(z_i)_i := (z_i)_{i \in N} = (z_1, z_2, \dots) \quad (4.6.44)$$

Aplicándolo en un espacio normalizado X' .

$$\|z_n - z_{n'}\| < \varepsilon \quad (4.6.45)$$

Para todo ε mayor de 0 y converge cuando n tiende a infinito, siendo X' un espacio de Hilbert de *kernel* reproducido, donde funciones $f : X \rightarrow R$, dotado de un producto punto (\cdot, \cdot) y la norma.

$$\|f\| := \sqrt{\langle f, f \rangle} \quad (4.6.46)$$

Existe una función $k : X \times X \rightarrow R$ donde.

1. K cumple con la reproducción.

$$\langle k(x, \cdot), k(x', \cdot) \rangle = k(x, x') \quad (4.6.47)$$

2. K se expande a X' , donde X es completado con todas las secuencias de Cauchy.

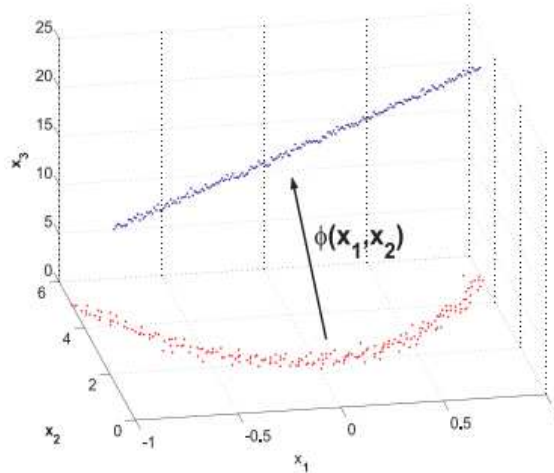
Ya definido los espacios de Hilbert, ahora se describe la estructura de de los *Kernel*, eligiendo adecuadamente $\Phi(\cdot)$ se puede construirse una máquina lineal en el espacio de características como un espacio de entrada, a un espacio de mayor dimensionalidad los productos escalares pueden ser expresados como *Kernel*.

$$K(x_i, x_j) = \Phi^T(x) \Phi(x) \quad (4.6.48)$$

La función de *Kernel* $K(x_n, x_m)$, [LIPO, 2005], es una función de espacio entrada, evita el mapeo $\Phi(x)$. el cual se requiere hacer un producto escalar de la espacio característico $\Phi^T(x) \Phi(x)$, lo que se hace es calcular directamente el *Kernel* con

los vectores de datos de entrenamiento dado en el espacio característico F, por tal razón no es necesario conocer el valor del mapeo $\Phi(x)$.

Figura 22. Ejemplo de transformación no lineal.



Fuente: Felipe Alonso, Tesis doctoral, estudio de los mecanismos de las arritmias cardiacas mediante modelado y procesado robusto digital de señal.

Para la transformación a un espacio de características, los pesos de la maquina se expresan como.

$$w = \sum_{i=0}^n (\alpha_i - \hat{\alpha}_i) \varphi(x_i) \quad (4.6.49)$$

Y se aplica el truco de *Kernel* a la separación óptima del hiperplano.

$$f(x) = \sum_{i=0}^n (\alpha_i - \hat{\alpha}_i) K(x_i, x) + b \quad (4.6.50)$$

Una de las condiciones de Mercer, para que las cuales el núcleo del *Kernel* represente un producto escalar es que el *kernel* sea una función simétrica definida que satisfaga.

$$\int K(x, y) g(x) g(y) dx dy \geq 0 \quad (4.6.51)$$

Siendo $g(x)$ cualquier función intangible.

$$\int g(x) dx < \infty \quad (4.6.52)$$

4.6.2.5. Kernel de Mercer: la función de *Kernel* se define $K(x_n, x_m)$, [LIPO, 2005], donde se evalúa un par de puntos, X_n y X_m son puntos del set de entrenamientos.

Para encontrar un hiperplano de separación de una alta dimensión se define.

$$y_i = \vartheta(x_i), \vartheta: (R^d \rightarrow R^m) \quad (4.6.53)$$

Donde $m > d$, recordar que se debe sacar el producto interno de los elementos A, donde el nuevo vector A es.

$$A = [y_i y_j \vartheta^T(x_i) \vartheta(x_j)] \quad (4.6.54)$$

Ahora se define un operador *Kernel* $K(x_i, x_j)$, para hallar el hiperplano optimo se necesita una transformación lineal ϑ , el producto interno y la función de núcleo.

Para hallar el valor de umbral b, en una función de *Kernel* se parte de un problema dual el cual se maximiza la función

$$Q(\alpha) = \sum_{i=1}^M \alpha_i - \frac{1}{2} \sum_{i,j=1}^M \alpha_i \alpha_j y_i y_j H(x_i, x_j) \quad (4.6.55)$$

Sujeto a las restricciones

$$\sum_{i=1}^M y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C \quad \text{para } i = 1, \dots, M \quad (4.6.56)$$

Sigue siendo un problema de programación cuadrática porque $\alpha = 0$ y es una solución factible, nótese que la ecuación (4.6.55) es igual a la expresión (4.6.38), la única diferencia es que no se trabaja con productos internos sino directamente con el *Kernel*, ahora usando las condiciones KKT se tiene.

$$\alpha_i (y_i (\sum_{j=1}^M y_j \alpha_j H(x_i, x_j) + b) - 1 + \xi_i) = 0 \quad \text{para } i = 1, \dots, M \quad (4.6.57)$$

$$(C - \alpha_i) \xi_i = 0 \quad \text{para } i = 1, \dots, M \quad (4.6.58)$$

$$\alpha_i \geq 0, \quad \xi_i \geq 0 \quad \text{para } i = 1, \dots, M \quad (4.6.59)$$

La función decisión es dada por

$$D(x) = \sum_{i \in S} \alpha_i y_i H(x_i, x) + b \quad (4.6.60)$$

Donde b es dado por

$$b = y_j - \sum_{i \in S} \alpha_i y_i H(x_i, x_j) \quad (4.6.61)$$

Donde X_j es un vector de soporte ilimitado

$$b = \frac{1}{U} \sum_{j \in U} (y_j - \sum_{i \in S} \alpha_i y_i H(x_i, x_j)) \quad (4.6.62)$$

Donde U es un set de índices de vectores de soporte ilimitados, donde se desconoce las clases a clasificar.

$$x \in \begin{cases} \text{clase 1} & \text{si } D(x) > 0 \\ \text{clase 2} & \text{si } D(x) < 0 \end{cases} \quad (4.6.63)$$

Si $D(x) = 0$ es un punto mal clasificado.

4.6.2.6. Tipos de Kernel: se tiene varios tipos de *Kernel* [LIPO, 2005].

Tabla 1. Kernel conocidos.

Función de Kernel	Tipo de clasificación
$K(x, x_i) = (x^T X_i)$	Lineal, Kernel de producto punto
$K(x, x_i) = [(x^T x_i) + 1]^d$	Polinomio completo de grado d
$K(x, x_i) = e^{\frac{1}{2}} [(x - x_1)^T \sum (x - x_i)]$	Gausiana RBF
$K(x, x_i) = \tanh[(x^T x_i) + b]$	Perceptron
$K(x, x_i) = \frac{1}{\sqrt{\ x - x_i\ ^2 + \beta}}$	Función de inverso multicuadrático

Fuente: Support vector machine theory and applications studies in fuzzines and soft computing.

Kernel lineal: si el problema es linealmente separable en un espacio de entrada y no se necesita un mapeo en un espacio de alta dimensión.

$$H(x, x') = x^T x' \quad (4.6.64)$$

Kernel polinomial: con un grado d donde es un numero natural.

$$H(x, x') = (x^T x' + 1)^d \quad (4.6.65)$$

Donde si $d=1$ es equivalente al kernel lineal, si $d=2$

$$H(x, x') = 1 + 2x_1x'_1 + 2x_2x'_2 + 2x_1x'_1x_2x'_2 + x_1^2x'^2_1 + x_2^2x'^2_2 = g^T(x)g(x) \quad (4.6.66)$$

Donde

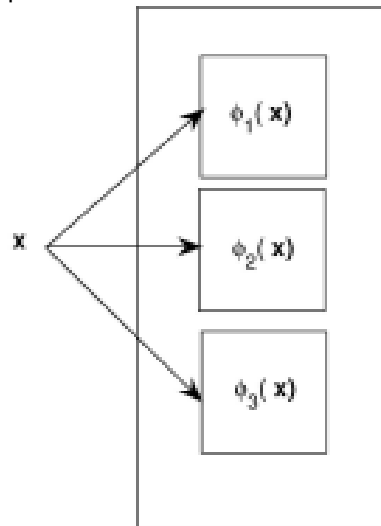
$$g(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2x_2^2)^T \quad (4.6.67)$$

Kernel función de base radial

$$K(a, b) = \exp\left(-\frac{(a-b)^T(a-b)}{2\sigma^2}\right) \quad (4.6.68)$$

4.6.2.7. Kernels compuestos: usando los espacios de Hilbert [MARTINEZ, 2008] se pueden realizar una trasformada de los datos a distintos espacios de Hilbert con varios *Kernel*.

Figura 23. Transformada a otros espacios de Hilbert.



Fuente: MANUEL MARTÍNEZ Ramón, introducción a los métodos Kernel.

Si un vector

$$x = \begin{pmatrix} x^a \\ x^b \end{pmatrix} \quad (4.6.69)$$

Se transforma a la forma

$$\varphi(x) = \begin{pmatrix} \varphi^a(x^a) \\ \varphi^b(x^b) \end{pmatrix} \quad (4.6.70)$$

En un espacio de Hilbert compuesto por dos espacios H^a y H^b , su producto escalar es

$$\varphi(x_1)^T \varphi(x_2) = \varphi(x_1^a)^T \varphi(x_2^a) + \varphi(x_1^b)^T \varphi(x_2^b) = K(x_1^a, x_2^a) + K(x_1^b, x_2^b) \quad (4.6.71)$$

Donde se puede ver que la suma de dos *Kernel* suma es un *kernel*.

4.7. MULTICLASES EN SVM ¹⁹

Las Máquina de Soporte Vectorial solo permite el uso de dos clases, pero es posible abordar mas clases, creando un clasificador multiclases [LIPO, 2005] [GARCIA, 2005].

Para construir esto se necesita un modelo K^{th} donde es $y_k(x)$, el cual es entrenado con los datos positivos es la clase C_k y los negativos es la clase $K-1$, este es el enfoque uno versus el resto.

$$y(x) = \max_k y_k(x) \quad (4.7.1)$$

Pero al usar varios clasificadores traen muchos problemas.

- No se garantiza que haya una escala adecuada para los valores $y_k(x)$.
- El desbalance entre las instancias, si se tiene 10 clases con igual número de ser de entrenamiento, pero realizando una clasificación independiente de una clase, se realiza una comparación del 90% de datos negativos y 10% de datos positivos, pero la simetría original es perdida, modificando el valor positivo +1 y negativo $1/(K-1)$.

Se puede usar la definición de Weston y Watkins [BISHOP, 2006], la cual define una función para entrenar las K simultáneamente, se basa en la maximización de la margen para una de las márgenes, sin embargo es muy lento el entrenamiento, si se soluciona K separadamente sobre cada dato N , tiene un costo $O(KN^2)$, una optimización simple $(K-1)N$, se resuelve con un costo total de $O(K^2N^2)$.

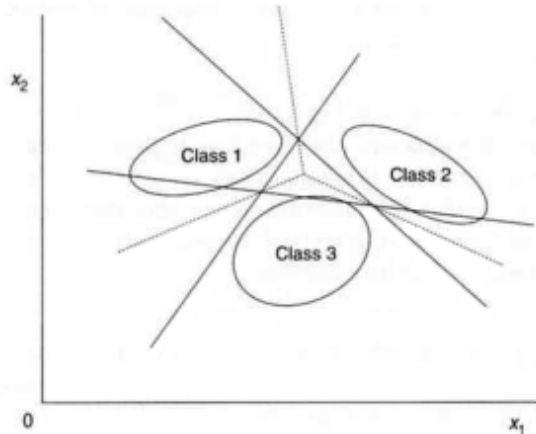
Hay cuatro arquitecturas para solucionar el problema de las multiclases, pero la explicación profunda de cada una se encuentra por fuera del alcance de este trabajo monográfico así que se da un resumen de cada una.

4.7.2. One-against-all (Uno contra todos)(OAASVM): un problema de n clases, se convierte en un problema de dos clases, la clase n ésima es separada del resto de las clases, pero existen regiones inclasificables si se usan funciones de decisión discretas.

Para los datos del set de prueba, se mejora la generalización de habilidad de los SVM sobre otras SVM con una función de decisión discreta es grande.

¹⁹ ABE shibeo, support vector machine for pattern classification, Kobe University, 2005, 343 p.

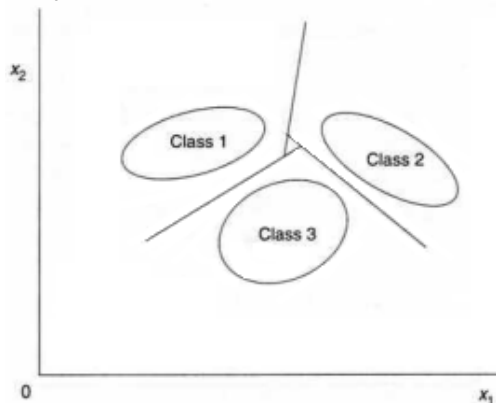
Figura 24. Limite con uno contra todos.



Fuente: ABE shibeo, support vector machine for pattern classification.

4.7.3. Pair wise SVM : convierte un problema n clases en un problema $n(n-1)/2$ problema de dos clases, para cubrir todos los pares de clases, pero también existen las regiones inclasificables.

Figura 25. Limite de clases con pairwise SVM.



Fuente: ABE shibeo, support vector machine for pattern classification.

4.7.4. Error-correcting output code (ECOC) (error correctivo en código de salida): este método se usa para resolver regiones inclasificables, es una extensión de uno contra todos, pero introduciendo una omisión a las salidas, para esta estructura la optimización es necesaria.

4.7.5. All-at-once SVM (todos de una vez): se determina las funciones de decisión con base a las regiones inclasificables resueltas, para eso la formulación original se determina con el número de variables es $M \times (n-1)$ donde M es el

numero de de set de entrenamiento y n es el numero de clases, pero el entrenamiento se dificulta si el numero de set de entrenamientos es grande.

En la tabla 2, se puede ver la comparación entre los cuatro tipos, n son las clases y M son los datos de entrenamiento, k es el largo del las palabras de código.

Tabla 2. Comparación entre SVM.

	Uno contra todos	Par wise	ECOC	Todos contra uno
función de decisión	n	$N(n-1)/2$	k	n
Variable	M	$M_i + M_j$	M	M
equivalencias	1	1	1	n

Fuente: ABE shibeo, support vector machine for pattern classification y modificado

4.7.6. Grafica aciclica dirigido SVM (DAGSVM): el método usa un entrenamiento construido por un clasificador binario $M \times (M-1)/2$ donde M es la clase, el set de prueba la evaluación de la función de decisión binaria empieza por el nodo raíz y se mueve de izquierda a derecha dependiendo de los valores de salida.

4.7.7. One- against-one (uno contra uno) (OAOSVM): es un clasificador binario $M \times (M - 1)/2$, usando las combinaciones pair –wise de las clases M , el numero de ejemplos para el entrenamiento es pequeño, mientras solo los ejemplo de dos con todas las clases M es considerable.

4.8. DISTINTO TIPOS DE MÁQUINA DE SOPORTE VECTORIAL ^{20, 21}

Existen una gran cantidad de tipo de Máquina de Soporte Vectorial, donde se pretende construir una SVM de acuerdo a ciertas características, proporcionando un mejoramiento en el método, pero aumentando el número de tipos de SVM, por tal razón se tomo un grupo de estas maquinas y se realizo una breve descripción de cada una.

²⁰ SHAONING Pang, KASABOV nikola, inductive vs trasductive infrencia, global vs local model: SVM, TSVM, and SVMT for gene expression classification problems, Auckland university of tecnology, 2004, 6 p.

²¹ SHI Guangzhi, HU junchuan, a dynamic recognition method study using the support vector machine, navy submarine academy, 2007, 5 p.

4.8.2. SVM transductiva (TSVM): estas SVM el aprendizaje incluye la información del set de prueba S en el proceso de entrenamiento, para casos lineales.

$$f_l = L(S_{entrenamiento}, S_{test}) \quad (4.8.1)$$

Donde.

$$S_{entrenamiento} = (\bar{x}'_1, y'_1), (\bar{x}'_2, y'_2), \dots, (\bar{x}'_n, y'_n) \quad (4.8.2)$$

En los casos separables se busca las etiquetas y_1^*, \dots, y_n^* en los dato de prueba, en el hiperplano (w, b) , donde se mínima sobre.

$$(y'_1, y'_2, \dots, y'_n, \bar{w}, b): \frac{1}{2} \bar{w}^T \bar{w} \quad (4.8.3)$$

Con las restricciones.

$$y_i(\bar{w} \cdot \bar{x}_i + b) \geq 1 \quad (4.8.4)$$

$$y'_i(\bar{w} \cdot \bar{x}'_i + b) \geq 1 \quad (4.8.5)$$

En los casos no separables, se minimiza.

$$(y'_1, y'_2, \dots, y'_n, \bar{w}, b, \xi_1, \dots, \xi_n, \xi'_1, \dots, \xi'_n): \frac{1}{2} \bar{w}^T \bar{w} + C(\sum_{i=1}^n \xi_i)^k + C(\sum_{j=1}^n \xi'_j)^k \quad (4.8.6)$$

Con las restricciones

$$\begin{aligned} y_i(\bar{w} \cdot \varphi(\bar{x}_i) + b) &\geq 1 - \xi_i \\ y'_i(\bar{w} \cdot \varphi(\bar{x}'_i) + b) &\geq 1 - \xi'_i \end{aligned} \quad (4.8.7)$$

Donde C es el factor de ejemplo de prueba y $C^* \xi^*$, es el termino enésimo de el ejemplo de prueba a la función objetivo.

4.8.3. Arboles de SVM (BTSMV): Se basan en el principio de divide y conquista, usando una clase binaria en clúster específicos, hay dos diseño de arboles, lo primero es dividir los datos en dos datos subconjuntos basados en las características globales como la composición y características propias.

Algoritmo 1: entrenamiento del árbol SVM

Funcion SVMarbol_entrenamiento(un set de entrenamiento X){
If (X contiene el mismo número de miembros de la clase){
Marca el fin de la rama de SVM del árbol T;


```

    Return;
}
    Clase_especifica_cluster(x: x1, x2) /*partición X dentro de los subset x1 y x2*/
    T= anexar_nodo(T); /*SVM entrenamiento sobre una partición de datos*/
    Entrenamiento_SVM_clasificador(x:x1,x2) /*recursiva partición y entrenamiento/
arbolSVM_entrenamiento(x1);
arbolSVM_entrenamiento(x2);
}

```

Dependiendo del resultado de la regla de decisión hecho por el nodo raíz, se recorre hacia abajo por el nodo hijo, y se repite hasta el terminal.

Algoritmo 2. Árbol de prueba

```

Funcion arbolSVM_prueba(árbol SVM T, entrada del dato x){
Corriente =1; /*set de nodo corriente para la raíz de nodo de T*/
/*se busca en el árbol hasta el nodo terminal es encontrado*/
While (Tcorriente es un nodo interno){
    Test_SVM_Tcorriente(x)
Next = usca_proximo_nodo(T,corriente);
Corriente =next;
} /*Retorna la etiqueta de nodo terminal
Return etiqueta (Tcorriente)
}

```

4.8.4. Algoritmo de las SVM basada en función objetivo: Este método posee el siguiente algoritmo

Paso 1: la característica objetivo x (espacio de entrada), es la entrada de las SVM basada de la característica objetivo, el valor umbral c y la función h(.) esta dada.

Paso 2: se calcula la distancia r_i, entre la característica objetivo x y la muestra de entrenamiento x_i, y calcular la escala r_c

$$r_c = [f^{-1}(c)]^{-1} \quad (4.8.8)$$

Paso 3: calcule la función penalidad C_i del set de entrenamiento A_x, si la distancia de r_i es menor que A_x, la muestra de entrenamiento x_i es seleccionado en el set de entrenamiento de A_x.

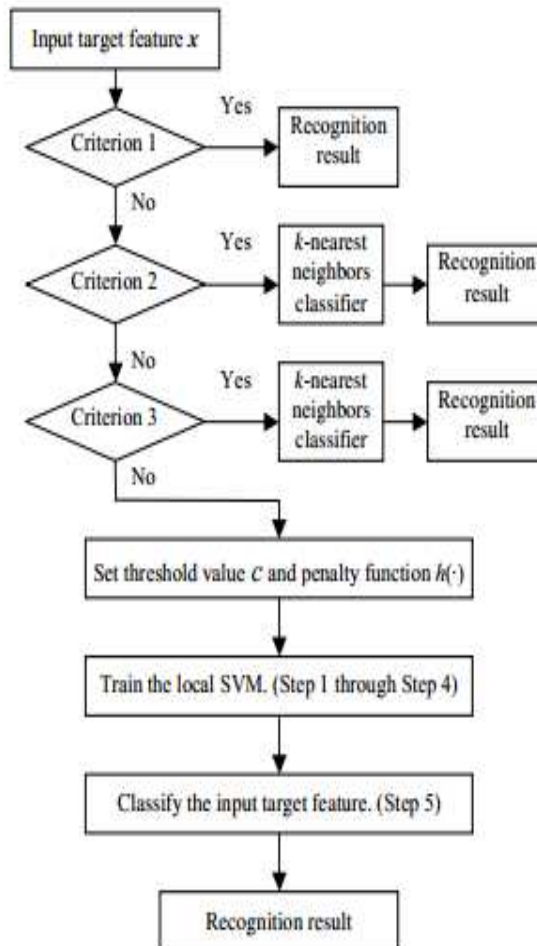
Paso 4: sustituye el dinámico set de entrenamiento Ax y la función de penalidad C_i , en la función de núcleo (4.6.39) y (4.6.40).

Paso 5: clasificar la característica objetivo x y obtener el resultado del reconocimiento.

Paso 6: para otras características objetivo repetir el paso 1 al 5.

4.8.5. Método de reconocimiento dinámico con SVM: para este método se tiene el siguiente algoritmo representado en el diagrama de flujo

Figura 26. Diagrama de flujo método de reconocimiento dinámico.



Fuente: SHI Guangzhi, HU junchuan, a dynamic recognition method study using the support vector machine.

Se tiene el objetivo es clase M, donde se tiene la clase 1,..., y clase m,..., que tiene el numero de muestras N_1, N_m .

Criterio 1: si el L más cercano a la muestra que pertenece a la clase m la característica objetivo es de la clase m.

Criterio 2: si $n \leq L$ se selecciona el método de vecindario k-cercano como característica objetivo.

Criterio 3: si $n > L$ y

$$\frac{\max(N_m)}{\min(N_m)} \geq L_2 \quad (4.8.9)$$

4.9. COMPARACION DE SVM CON DISTINTOS KERNEL

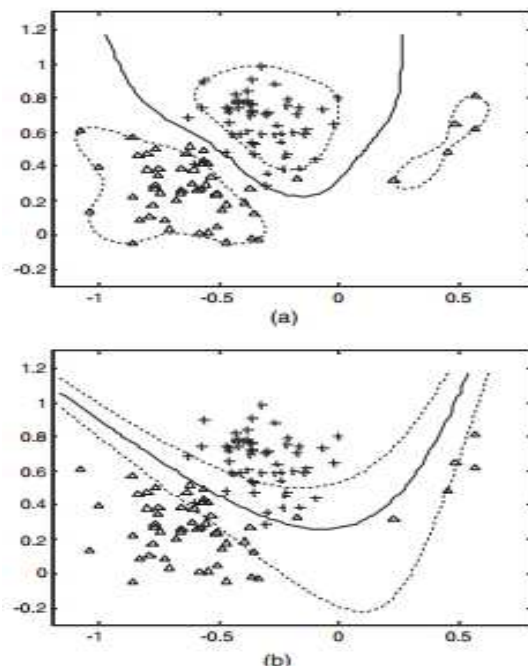
Algunos *Kernel* presentan mejor desempeño que otros *Kernel*, se presentara algunas comparaciones y su resultado [LIPO, 2005].

4.9.2. Comparación RBF y polinomial: En la figura 27, se usaran un set de datos de entrenamiento compuesto por 55 muestras de la clase 1 y 55 muestras de la clase 2, note que la clase 2 tiene dos clúster el mayor de 50 muestras y el menor 5 muestras y un set de prueba de 1100 muestras usadas para la estimación del error.

La clase 1 "+", es generado una distribución Gausiana con el centro en (-0.3, 0.7) y varianza 0.03.

La clase 2 "Δ", tiene dos Gaussianas una en (-0.7, 0.3) y (0.4, 0.7) con varianza 0.03.

Figura 27. Comparación de los límites de decisión



Fuente: Support vector machine theory and applications studies in fuzzines and soft computing

En las figura 27. Se tiene a) el set con un *Kernel* RBF con $p = 0.2$ y la b) un Kernel polinomial de grado $d = 3$, en la tabla 3 se puede ver los resultados donde el polinomial tiene más porcentaje de vectores de soporte que el RBF, pero posee un error levemente mayor.

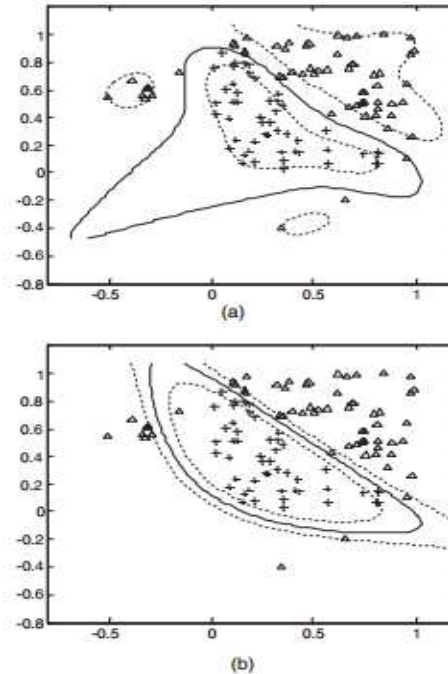
Tabla 3. Comparación con precisión de la predicción.

	Error	Proporción(%vector soporte)
RBF(C = 1, p=0.2)	0.0582	(25.5%)
POLY(C=1,d=3)	0.0673	(26.4%)

Fuente: Support vector machine theory and applications studies in fuzzines and soft computing y modificado.

En la figura 28. Se tiene un conjunto de datos, con un set de entrenamiento de 110 muestras de las cuales 50 muestras de la clase 1 y 60 muestras de la clase 2, la clase 2, tiene 3 distribuciones 50 muestras, 8 muestras y 2 muestras, y un set de prueba de 1100 ejemplos usados para la estimación del error, Clase 1 "+". Es una distribución uniforme en un triángulo con vértices (0,0), (1,0), (0,1), Clase 2 "Δ", se tiene tres distribuciones uniformes en un espacio triangular, se tiene una varianza de 0.01.

Figura 28. Comparación del mejor limite de decisión.



Fuente: Support vector machine theory and applications studies in fuzzines and soft computing.

En la figura 28. Se puede observar en a) un *Kernel* RBF con $p = 0.2$ en b) un *Kernel* polinomial de grado $d= 3$.

En la tabla 4, se puede ver que en este caso el porcentaje de los vectores de soporte aumenta y posee un error menor a comparación de la polinomial de grado 3 que posee un porcentaje de vectores de soporte del 16.4%.

Tabla 4. Comparación con la precisión de las predicciones.

	Error	Proporción(%vector soporte)
RBF(C = 1, p=0.2)	0.0291	(37.3%)
POLY(C=100,d=3)	0.0473	(16.4%)

Fuente: Support vector machine theory and applications studies in fuzzines and soft computing y modificado.

4.9.3. Comparación kernel lineal, polinomial y RBF: Se utilizaran un set de dato de hipotiroides con 4 clases con 30 atributos, el número de muestras son 3772 de las cuales se dividen en 3167 muestras de entrenamiento y 605 datos de prueba.

Tabla 5. Comparación entre los núcleos de Kernel.

Kernel	C	Precisión		Tiempo computacional	
		Entrenamiento	Test	Entrenamiento	Test
Lineal	10	93.8%	92.5%	58.8s	52.9s
	40	93.1%	93.8%	92.2s	93.3s
	60	97.2%	96.4%	40.74s	40.21s
	80	93.4%	92.9%	123.9s	134.6s
Polinomial	10	95.4%	94.2%	416.7s	413.6s
	40	96.6%	95.1%	656.5s	620s
	60	99.1%	96.9%	139.54s	134.5s
	80	97.3%	95.2%	711.6s	783.7s
RBF	10	92.4%	93.1%	152.9s	155.8s
	40	93.1%	92.2%	135.3s	139.1s
	60	95.5%	96.1%	50.4s	79.49s
	80	92.3%	92%	186s	187.7s

Fuente: multi-class support vector machine (SVM) classifiers- an application in hypothyroid detection and classification y modificado.

En la tabla 5, se puede ver el resultado de los *Kernel* con el conjunto de datos, en la fila del *Kernel* las variaciones del valor C, con valor 60 se puede ver que tiene un valor superior al resto de valores, en el caso del polinomial y el RBF se tienen el mismo resultado, para esa aplicación es el valor de C ideal, Y en general el *Kernel* polinomial dio un mejor resultado comparado con los otros *Kernel*, aunque el único problema con este *Kernel* es que tiene el mayor tiempo de los otros dos.

4.10. COMPARACION ENTRE DISTINTOS TIPOS DE SVM

4.10.1. Comparación entre SVM y método de reconocimiento dinámico: se tiene un grupo de entrenamiento A con un grupo de de 1200 muestras de objetivo, B 1000 muestras, C 750 muestras, D 540 muestras, [SHI and HU, 2007].

Tabla 6. Comparación.

Objetivo clasificación	Precisión de reconocimiento (%)	
	SVM tradicional	Método de reconocimiento dinámico
A	86.3	87.3
B	82.5	83.6
C	72.3	76.5
D	71.7	75.3

Fuente: Dynamic recognition method study using the support vector machine y modificado.

Se puede apreciar en la tabla 6, un aumento en el reconocimiento con un mayor número de muestras.

4.10.2. Problema de clasificación de expresión de genes: En el área de la bioinformática, realizar una identificación de genes en una muestra donde se clasifica dos o más clases, (maligno o benigno) es muy importante, en este caso se utilizarán tres tipos de máquinas de soporte vectorial las SVM, TSVM y SVMT, [GARCIA, 2005].

4.10.2.1. Set de datos: Se usan distintos tipos de set de datos de cáncer, linfoma, leucemia, colon.

Los datos de linfoma es una colección de 77 muestras de linfocitos malignos, 58 muestras de células B largas linfomas (DLBCL), 19 muestras de linfoma folicular (FL) y contiene 6817 genes.

Los datos de leucemia es una colección de 27 muestras leucemia limfoblástico (ALL), 11 muestras de leucemia mieloblástico (AML), de una muestra 6817.

El set de datos del colon es una colección de 22 muestras normales y 40 muestras con cáncer, obtenidas de un grupo de 2000 genes.

4.10.2.2. Pasos experimentales y resultados: se desarrolló la siguiente metodología en la realización de este análisis.

1. Se define las clases objetivo.
2. Se identifica los sub sets y se dividen tres objetivos.
 - 2.1. Minimizar el tamaño de los sub set en el clasificador.
 - 2.2. Minimizar el número de desajustes en las muestras de datos entrenamiento.
 - 2.3. Minimizar el número de desajustes en las muestras de pruebas.
3. Filtrar y normalizar los datos, eliminando los genes que no presenten mucha variación.
4. Construir el clasificador, para cada set de variables y definir clases, utilizando una muestra y usando el resto para el ser de entrenamiento, teniendo en cuenta el modelo de debe minimizar el error.
5. Evaluar el resultado.

Tabla 7. Comparación entre tipos de SVM.

	Marcador de selección de genes	SVM	TSVM	Árbol SVM
Linfoma	DLBCL vs FL; 6432 genes, 77 muestras	55.8%	57.1%	77.9%
	DLBCL vs FL; 30 genes, 77 muestras	92.2%	93.5%	92.2%
	Curable vs fatal; 6432 genes, 58 muestras	53.3%	55.1%	72.4%
	Curable vs fatal; 13 genes, 58 muestras	72.4%	79.3%	70.7%
Leucemia	ALL vs AML: 7219 genes, 72 muestras	55.6%	52.8%	78.3%
	ALL vs AML: 3859 genes, 72 muestras	94.4%	95.8%	100%
	ALL vs AML: 27 genes, 72 muestras	98.6%	98.6%	90.3%
Cáncer de colon	Normal vs cáncer: 2000 genes, 62 muestras	79.0%	72.6%	80.7%
	Normal vs cáncer: 12 genes, 62 muestras	100%	100%	98.4%

Fuente: inductive vs trasductive infrencia, global vs local model: SVM, TSVM, and SVMT for gene expression classification problems y modificado.

En la tabla 7, se puede ver el desempeño de los TSVM y SVMT, es superior al resultado de las SVM, la técnica TSVM tiene mejor desempeño con un pequeño número de genes, muestras de el árbol tiene un mejor desempeño con un alto número de genes.

4.10.3. Comparación entre OAASVM y OAOSVM: ambos son técnicas de *Kernel* multiclases, se utiliza el mismo set de datos del apartado anterior con 4 clases de hipoteroides, [SHAONING y KASABOV, 2004].

Tabla 8. Comparación con Kernel múltiples.

Método	Precisión		Tiempo computacional	
	Entrenamiento	Prueba	Entrenamiento	Prueba
OAASVM	95.6%	95.3%	212.4s	227.8s
OAOSVM	96.1%	94.48%	79.3s	81s

Fuente: multi-class support vector machine (SVM) classifiers- application in hypothyroid detection and classification.

Una comparación entre los métodos uno contra uno (OAOSVM) y uno contra todos (OAASVM), se presenta un mejor desempeño en el método de uno contra uno a demás que el tiempo computacional es mucho menor.

4.11. COMPARACION DE LAS SVM CON LAS REDES NEURONALES ^{22, 23}

4.11.1. Comparación entre las SVM con redes neuronales con datos diversos: se presenta un estudio de las SVM con las perceptron, con base a un set de datos sintéticos, rostros estándar y varios datasets sacados del repositorio UCI ML, cuya página oficial es <http://archive.ics.uci.edu/ml/> [SEBE, COHEN, GARG y HUANG, 2005].

Tabla 9. Resultado de la comparación.

Set de dato	Perceptron	SVM
Datos sintéticos	73.25%	78.50%
Detección de rostros	89.5%	91.5%
Hongos(UCI)	83.61%	87.96%
Hígado(UCI)	56.88%	60.94%
Ionosfera(UCI)	81.90%	84.84%
Corazón(UCI)	82.37%	83.04%

Fuente: de Machine Learning in Computer y modificado.

En la tabla 9 se puede observar que se tiene mayor porcentaje de éxito frente al perceptron, las SVM poseen una ventaja, ya que están manejando valores globales y no soluciones locales como en el caso de las redes, también hay que tener en cuenta que los datos son los que determinan el éxito del diseño y los resultados de este, además de su entrenamiento más sencillo, de ahí su ventaja sobre las redes.

4.11.2. Clasificación con muestras pequeñas: con una muestra pequeña de 10, 20, 50, seleccionada como set de entrenamiento y 900 muestras de set de prueba [LI, GOU y LI, 2008].

Tabla 10. Comparación entre SVM y redes neuronales.

Entrenamiento(muestras de prueba)	SVM	RN
10(900)	72.56%	65.67%
20(900)	81.44%	67.33%
50(900)	87.89%	72.85%

Fuente: Application research of support vector machine in network security risk evaluation y modificado.

²² CHAMASEMANI falah fereshteh, YASHWANT pradas singh, multi-class support vector machine (SVM) classifiers- an application in hypothyroid detection and classification, university multimedia, 2011, 6 p.

²³ LI cong-cong, GUO ai-ling, LI dan, application research of support vector machine in network security risk evaluation, international symposium on intelligent information technology application workshops, 2008, 4 p.

En la tabla 10 se realiza una comparación entre las SVM y redes neuronales con pocas muestras de entrenamiento, se puede observar que el SVM es superior a las redes neuronales, a medida de que se aumenta la cantidad el porcentaje de vectores de soporte.

Tabla 11. Comparación de tiempo entre ambos métodos.

Entrenamiento(muestra de prueba)	SVM		RN	
	Tiempo entrenamiento	Tiempo de prueba	Tiempo entrenamiento	Tiempo de prueba
10(900)	0.0097	0.0184	0.6075	0.0214
20(900)	0.0171	0.0389	0.6305	0.0255
50(900)	0.1913	0.1179	0.6641	0.0412

Fuente: Application research of support vector machine in network security risk evaluation y modificado.

En la tabla 11, se puede ver la diferencia entre tiempos de prueba entre ambos métodos, el tiempo de las SVM es muy menor que las redes neuronales en tiempo de entrenamiento, pero a medida que las muestras sube el tiempo de prueba en las SVM aumenta considerablemente mientras que las redes neuronales aumenta levemente.

4.11.3. Ventajas y desventajas frente a las redes neurales: aunque las redes son muy conocidas, las SVM tiene ventajas respecto a ellas aunque también presentan desventajas [LIPO, 2005].

4.11.3.1. Ventajas

- **Maximización en la generalización:** las redes neuronales en su clasificación de patrones usa el error de la suma de las raíces entre las salidas mientras las salidas de entrenamiento deseadas son minimizadas, en cuanto a su generalización cuando los datos de entrenamiento son escasos y linealmente separables la habilidad de generalización de deteriora considerablemente, mientras que en las SVM se entrena maximizando el margen y la habilidad de generalización no se deteriora mucho bajo esas condiciones.
- **No hay locales mínimos:** las redes neuronales son conocidos por tener numerosos locales mínimos, mientras él las SVM son formulado como problemas cuadráticos donde la solución es una solución global.
- **Robustez de los datos atípicos:** las redes neuronales son vulnerables en los datos atípicos porque usan el error de suma de las raíces, para evitar eso se eliminan antes del entrenamiento algunos mecanismos para evitar esto se

incorporan en el entrenamiento, en las SVM el parámetro de la margen C controla la mal clasificación del error, si el valor es mayor a C este valor es suprimido, si es pequeño se deja, esta es la propiedad de C.

4.11.3.2. Desventajas

- **La extensión en problemas de multiclases:** aunque las SVM usan funciones de decisión directa, el problema de las multiclases no es sencillo porque se tienen muchas formulaciones.
- **Largo tiempo de entrenamiento:** en las SVM están asociadas a problema dual, donde el número de variables es igual al número de datos de entrenamiento, así un número grande de datos de entrenamiento, resolviendo por problema dual puede traer dificultades en el tamaño de la memoria y datos de entrenamiento.
- **Selección de parámetro:** las SVM se necesita seleccionar un *Kernel* apropiado y sus parámetros y un set de valor para el parámetro de margen C, para escoger los parámetros óptimos es un problema llamado modelo de selección, donde se debe estimar la generalización a través de la repetición de SVM, pero el tiempo consumido es mucho debido a la cantidad de índices estimados para la generalización.

4.12. MÉTODOS DE ENTRENAMIENTO

Para evitar que el excesivo tiempo de entrenamiento se realiza una preselección de candidatos de vectores de soporte.

4.12.1. Técnicas de descomposición: para reducir el número de variables de entrenamiento se realiza unas técnicas llamadas chunking [LIPO, 2005], y este procedimiento es:

1. Un set de puntos F , para un set de punto de entrenamiento para W , donde F es un integrador positivo.
2. Resolver el sub problema para α_w .

3. Borrar las variables excepto los candidatos de vectores de soporte adicionada de F que no satisfagan las condiciones KKT y regresar al paso 2, si no es así termine el algoritmo.

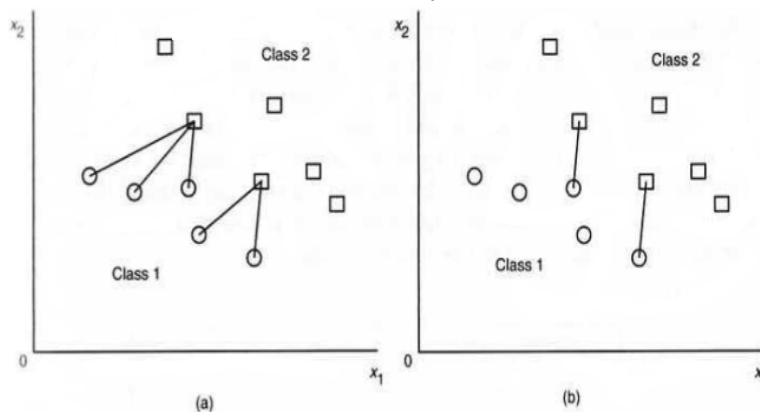
Sin embargo ese algoritmo incrementa los procesos de iteración, un procedimiento para disminuir eso es la chuking de tamaño fijo.

1. Selecciona W puntos para el set de datos de entrenamiento.
2. Resolver el sub problema α_w .
3. Si existe $\alpha_j (\in \alpha_N)$ que no satisface las condiciones KKT remplace por $\alpha_i (\in \alpha_N)$ con α_j e ir al paso 2, si no termine el algoritmo.

Hay una segunda técnica la cual realiza el mismo procedimiento con mejores resultados.

1. Se selecciona un dato de la clase uno y se compara con la distancia mínima con los puntos de la clase 2.
2. Se selecciona el par con la distancia mínima.
3. Repite los pasos anteriores.
4. Si no se obtiene los datos suficientes, repetir los pasos anteriores más veces.

Figura 29. Selección de candidatos de vectores de soporte.



Fuente: Support vector machine for pattern classification.

En la figura 29, la parte a) se selecciona un dato de la clase 1 y un par con la clase 2 con la mínima distancia, b) se selecciona el par con la mínima distancia.

4.13. METODO DE CLASIFICACION²⁴

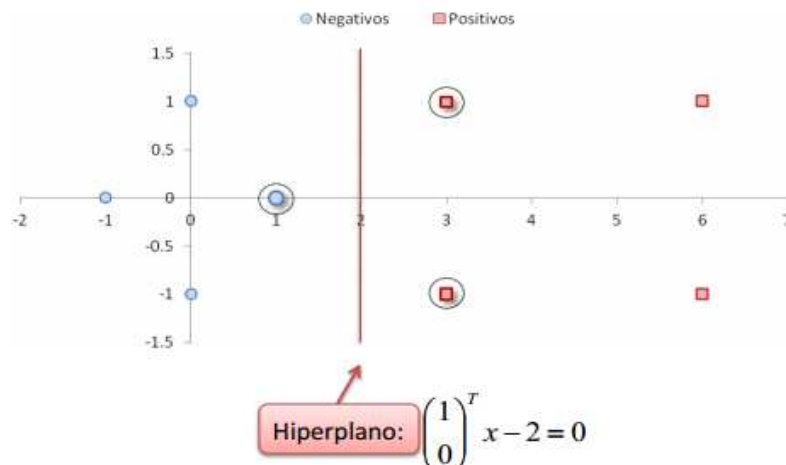
Cuando se va a ingresar un dato en el sistema a clasificar se usa el signo de la función

$$f(x) = \sum_{i=1}^{Ns} \alpha_i y_i \varphi(s_i) \cdot \varphi(x) + b = \sum_{i=1}^{Ns} \alpha_i y_i K(s_i, x) + b \quad (4.13.1)$$

Donde el resultado de la función $f(x)$, un numero negativo, se clasifica como -1 o clase equivalente, si se obtiene un numero positivo se clasifica +1 o su clase equivalente.

4.13.1. Ejemplo de clasificación: para la clasificación de un punto se parte conociendo la ecuación del hiperplano.

Figura 30. Ejemplos de pocos puntos y la ecuación hiperplano



Fuente: ESTRADA betsy mary, MERA Carlos Andrés, support vector machine

Para este ejemplo se parte de la ecuación del hiperplano discriminante, calculada conociendo los valores α , Donde los puntos del espacio de característica son pares ordenados (x,y) y un bias 1 de entrada (aumento de dimensión).

$$w = \sum_i \alpha_i \tilde{s}_i = -3.5 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \quad (4.13.2)$$

²⁴ ESTRADA betsy mary, MERA Carlos Andrés, support vector machine, universidad nacional de Colombia, 60p

Se va a clasificar el punto $x = (1,1)$

$$f(1,1) = \sigma(-3.5 \varphi \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot \varphi \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 0.75 \varphi \begin{pmatrix} 3 \\ 1 \end{pmatrix} \cdot \varphi \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 0.75 \varphi \begin{pmatrix} 3 \\ -1 \end{pmatrix} \cdot \varphi \begin{pmatrix} 1 \\ 1 \end{pmatrix}) \quad (4.13.3)$$

$$f(1,1) = \sigma(-3.5 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}) \quad (4.13.4)$$

$$f(1,1) = \sigma\left(\begin{pmatrix} -3.5 \\ 0 \\ -3.5 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 2.25 \\ 0.75 \\ 0.75 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 2.25 \\ -0.75 \\ 0.75 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}\right) \quad (4.13.5)$$

$$f(1,1) = \sigma(-7.0 + 3.75 + 2.25) \quad (4.13.6)$$

$$f(1,1) = \sigma(-1.0) \quad (4.13.7)$$

Como se ve en la ecuación (4.13.7) nos da un -1, por tal razón consideramos es punto como elemento de la clase negativa.

Ahora se clasifica el punto $x = (5,-1)$

$$f(5,-1) = \sigma(-3.5 \varphi \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot \varphi \begin{pmatrix} 5 \\ -1 \end{pmatrix} + 0.75 \varphi \begin{pmatrix} 3 \\ 1 \end{pmatrix} \cdot \varphi \begin{pmatrix} 5 \\ -1 \end{pmatrix} + 0.75 \varphi \begin{pmatrix} 3 \\ -1 \end{pmatrix} \cdot \varphi \begin{pmatrix} 5 \\ -1 \end{pmatrix}) \quad (4.13.8)$$

$$f(5,-1) = \sigma(-3.5 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ -1 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ -1 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ -1 \\ 1 \end{pmatrix}) \quad (4.13.9)$$

$$f(5,-1) = \sigma\left(\begin{pmatrix} -3.5 \\ 0 \\ -3.5 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ -1 \\ 1 \end{pmatrix} + \begin{pmatrix} 2.25 \\ 0.75 \\ 0.75 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ -1 \\ 1 \end{pmatrix} + \begin{pmatrix} 2.25 \\ -0.75 \\ 0.75 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ -1 \\ 1 \end{pmatrix}\right) \quad (4.13.10)$$

$$f(5,-1) = \sigma(-21.0 + 11.25 + 12.75) \quad (4.13.11)$$

$$f(5,-1) = \sigma(3.0) \quad (4.13.12)$$

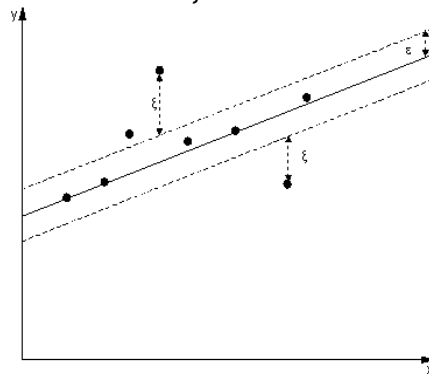
Como se ve en la ecuación (4.13.12), da un 3, por tal razón es un punto que pertenece a la clase positiva.

4.14. MÁQUINA DE SOPORTE VECTORIAL PARA LA REGRESION (RVM)⁹

La regresión es un método estadístico para encontrar las relaciones entre dos variantes y un término aleatorio.

Esta aplicación de los SVM se centra en la regresión lineal y usando una función llamada ξ – intensiva, la cual hace caso omiso a los errores que están dentro de una determinada área.

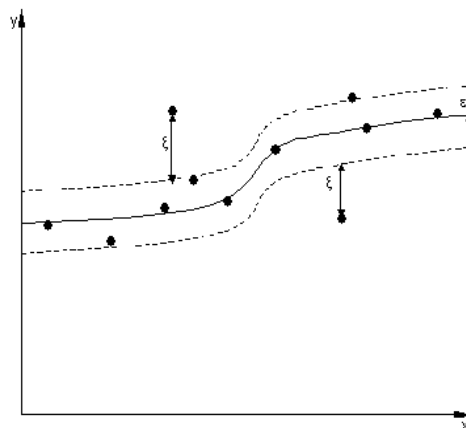
Figura 31 Regresión lineal con una banda de ξ – intensiva.



Fuente: <http://kernelsvm.tripod.com/>.

Los datos entrenamiento que están por fuera de la banda tienen un costo de error, los datos que están dentro de la banda sus valores son ceros.

Figura 32. Regresión no lineal.

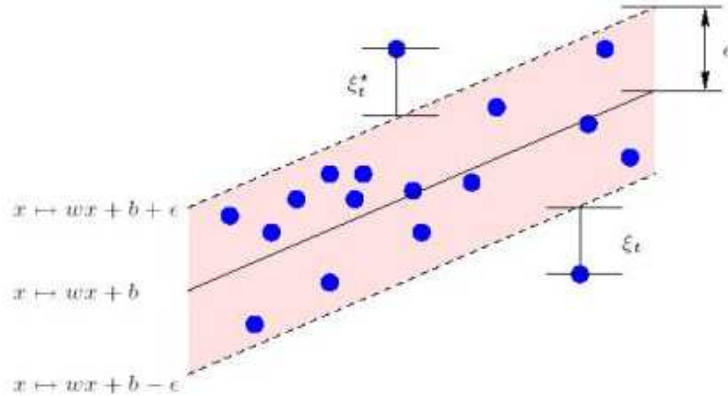


Fuente: <http://kernelsvm.tripod.com/>.

⁹ <http://kernelsvm.tripod.com/> visitada el día 22/06/12

La ventaja de usar las Máquina de Soporte Vectorial para este problema es que, se puede obtener la solución con un subconjunto de datos y con el uso de la función ξ – intensiva, se garantiza que hay un mínimo global y se garantiza la generalización del límite.

Figura 33. Variables de holgura.



Fuente: <http://kernelsvm.tripod.com/>.

Las variables de holgura ϵ , selecciona un grupo de puntos de datos.

La aplicación de la Máquina de Soporte para la regresión es la misma mecánica inicial que la clasificación, se comienza con un conjunto de puntos puestos en un espacio de característico.

$$f(x, w) = \sum_{j=1}^m w_j g_j(x) + b \quad (4.14.1)$$

Donde $g_i(x)$, $j = 1, \dots, m$, se denota el set de transformación lineal, b es una tendencia.

La formula de Máquina de Soporte Vectorial para la regresión se minimiza la función.

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi'_i) \quad (4.14.2)$$

Sujeto a

$$y_i - f(x_i, w) \leq \epsilon + \xi'_i \quad (4.14.3)$$

$$f(x_i, w) - y_i \leq \epsilon + \xi_i \quad (4.14.4)$$

$$\xi_i, \xi'_i \geq 0, \quad i = 1, \dots, n \quad (4.14.5)$$

Recordar que $1/2 \|w\|^2$ es el margen de separación de un hiperplano canónico, donde $\xi_i, \xi'_i \geq 0, i = 1, \dots, n$ son las variables de holgura y C determina el equilibrio entre el complejo del modelo y las desviaciones toleradas en la optimización.

La optimización del problema se convierte en un problema dual.

$$f(x) = \sum_{i=1}^{n_{gr}} (\alpha_i - \alpha'_i) K(x_i, x) \quad (4.14.6)$$

Sujeto a

$$0 \leq \alpha'_i \leq C \quad (4.14.7)$$

$$0 \leq \alpha_i \leq C \quad (4.14.8)$$

Donde n_{rs} es el número de vectores de soporte y

$$K(x, x_i) = \sum_{j=1}^m g_j(x) g_j(x_i) \quad (4.14.9)$$

Es la función de *Kernel*, esta función es determinada por la misma naturaleza del problema, teniendo en cuenta la distribución del dominio de los valores de entrada de los datos de entrenamiento (x).

Sea las Máquina de Soporte Vectorial para la regresión o SVR, es un problema de programación cuadrática.

4.14.1. Calcular la calidad de la estimación: se usa la función $L(y, f(x, w))$, donde en problemas de clasificación es una función de pérdida que tiene los valores 0 o 1, donde se utiliza la ϵ sensibilidad en la función de pérdida.

$$L_\epsilon(y, f(x, w)) = \begin{cases} 0 & \text{if } |y - f(x, w)| \leq \epsilon \\ |y - f(x, w)| - \epsilon & \text{otros} \end{cases} \quad (4.14.10)$$

Con el riesgo empírico

$$R_{emp}(w) = \frac{1}{n} \sum_{i=1}^n L_e(y_i, f(x_i, w)) \quad (4.14.11)$$

Aquí se usa el modelo de minimización riesgo estructural, el cual tiene un modelo con una VC dimensión minimizado, cuando el valor VC dimensión es baja, la probabilidad de error es baja.

5. ESTADO DEL ARTE

Las Máquinas de Soporte Vectorial poseen una gran variedad de tipos distintos, para solucionar u optimizar distintos tipos problemas de detección y clasificación, en la práctica se plantean optimización de algoritmos realizando híbridos de las SVM con otras metodologías como las *wavelet*, redes neuronales y cadenas de *markov*.

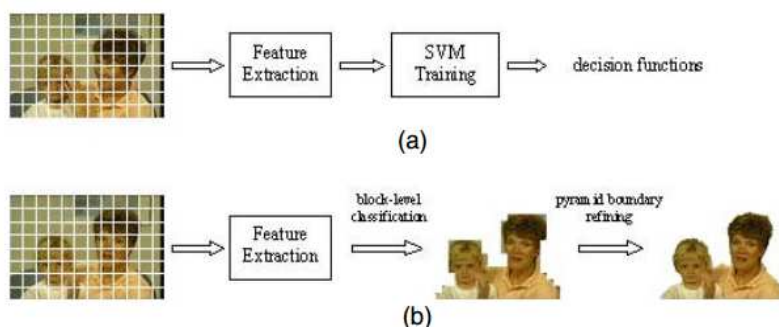
En la página web de la *IEEE Xplorer*, se presentan los últimos trabajos sobre Máquina de Soporte Vectorial desarrolladas, existen alrededor de 4485 trabajos sobre las Máquina de Soporte Vectorial, entre estas publicaciones se encuentran para conferencias, magazines y artículos de fácil acceso en los últimos 2 años.

La aplicación de las Máquina de Soporte Vectorial es extensa, se pueden aplicar en el campo de la medicina hasta la seguridad de las redes, las Máquina son más utilizadas en la clasificación de muestras y reconocimiento; En el caso de la visión artificial, se usan para el reconocimiento de objetos, rostros, caracteres, placa de autos, en interfaces humano computador; En la seguridad informática, se puede analizar el tipo de correo entrante y verificar si es spam o no correo spam; también se puede usar en simulación y predicción de la economía; las SVM se pueden aplicar a muchísimas áreas.

5.1. EXTRACCIÓN DE OBJETOS EN UN VIDEO²⁶

La utilización de Máquina de Soporte Vectorial puede ser utilizado en el proceso de extracción de un imagen en un video, donde las muestras son extraídas de un video de secuencias con el estándar MPEG-4, en esta extracción se utilizara un *soft SVM* y SVM, para luego mirar su desempeño.

Figura 34. Fases de la extracción.



Fuente : Soft SVM and its application to video object extraction.

²⁶ LUI Yi, ZHENG Yuan, soft SVM and its application to video object extraction, the Ohio state university, 2005, 4 p.

En la figura 34, en la parte a) la fase de entrenamiento b) fase de extracción, la idea base es descomponer la imagen en pequeños bloques.

Se usan las SVM, para clasificar las imágenes del fondo, la fase de entrenamiento consiste en dividir los bloques del objeto con los bloques del fondo dependiendo a qué clase de pixel en el centro del bloque pertenecen, en el proceso de aprendizaje separa el objeto del fondo, en la fase de extracción se dividen los *frames* en bloques y es evaluado por una función de decisión respecto al pixel central y por último se aplica un algoritmo pirámide refinado de límite para pulir el borde.

Para usar las SVM se necesita normalizar los datos, entre los pixeles del objeto y el fondo, en tamaño de bloques de $L \times M$.

$$y_i = \frac{\#pixeles\ del\ objeto - \#pixeles\ de\ fondo}{L \times M} \quad (5.1.1)$$

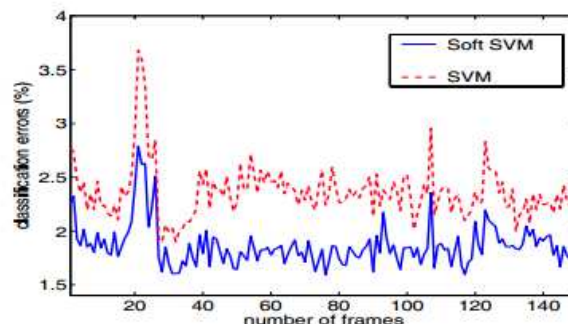
En la figura 35 se ve el resultado de la comparación de un *soft SVM* con un *kernel RBF*, comparado con un SVM, donde se ven las diferencia entre ambas máquina, *soft SVM* (b), es mas pulida que las SVM (a).

Figura 35. Comparación SVM con SOFT SVM.



Fuente: Soft SVM and its application to video object extraction.

Figura 36. Secuencia de frames.



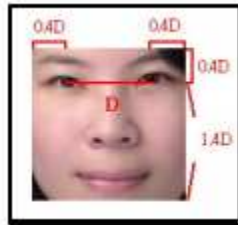
Fuente: Soft SVM and its application to video object extraction.

En la figura 36, se puede ver que las SVM posee un mayor porcentaje de error que las soft SVM, siendo estas mejor para la extracción de características.

5.2. DETECCIÓN DE ROSTRO BASADO EN SEGMENTACION DEL COLOR DE LA PIEL²⁷

Lo primero es elegir los candidatos a través del color de la segmentación de la piel, la localización de la cara usando bloques y propiedades de la cara y detector de ojos.

Figura 37. Modelo del rostro.



Fuente: Face detection based on skin color segmentation and SVM classification.

Para reducir la complejidad, se debe reducir el espacio de los colores y la intensidad, para tal razón se debe normalizar el espacio RGB (*Red, Green, Blue*) y HSV (*Hue, Saturation, Value*), con solo dos valores $r = R/(R+G+B)$ y $g = G/(R+G+B)$, un pixel es etiquetado como un pixel de piel si cumple con esas restricciones.

$$R > G \quad (5.2.1)$$

$$|R-G| \geq 11 \quad (5.2.2)$$

$$0,33 \leq r \leq 0,6 \quad (5.2.3)$$

$$0,25 \leq g \leq 0,37 \quad (5.2.4)$$

$$0,12 \leq S \leq 0,7 \quad (5.2.5)$$

$$0,3 \leq V \leq 1 \quad (5.2.6)$$

$$340 \leq H \leq 359 \text{ o } 0 \leq H \leq 50 \quad (5.2.7)$$

²⁷ HWEI-JEN Lin, SHWU-HUEY Yen, JIN-PIN Yeh, Face detection based on skin color segmentation and SVM classification, Tamkang university, 2008, 2 p.

Para la localización de los bloque de la cara, se usa una región máxima 30 x 30, un radio de aspecto es entre 0.8 y 2.6, el porcentaje de pixeles de piel en la caja limite sea mayor al 40%, el resto es una región no rostro.

Para la detección de ojos, se usa un radio de aspecto entre 0.2 y 1.67, el porcentaje de pixeles de piel mayor o igual a 30% y que el bloque de a cara entre 0.28 y 0.4.

En la figura 37. Se puede ver la distribución de características del rostro, utilizando los bloques de los ojos y calculando la distancia horizontal D, entre los dos centroides, se aplican los criterios anteriores.

Se aplican las SVM, para la clasificación, para eso se usa el Kernel función de base radial

$$K(x_i, x_j) = \exp(-\gamma(x_i - x_j)^2) \quad (5.2.8)$$

Donde

$$\gamma = 20 \quad (5.2.9)$$

Se usan 300 rostros y 300 no rostros en escala de grises para las muestras de entrenamiento, el tamaño de cada muestra es 20 x 20, el valor de gris I, se normaliza

$$\bar{I} = \frac{1}{N} \sum_{i=1}^N I_i \quad I'_I = (I_i - \bar{I}) + 128 \quad (5.2.10)$$

El resultado arrojo de un grupo de 2615 candidatos de rostros, con tamaños variables de 35 x 35 a 370 x 275, se presento un radio de detección del 88.26% al 96.37% con falsa aceptación 5.81% y 3.63%.

Figura 38. Imágenes de la detección.



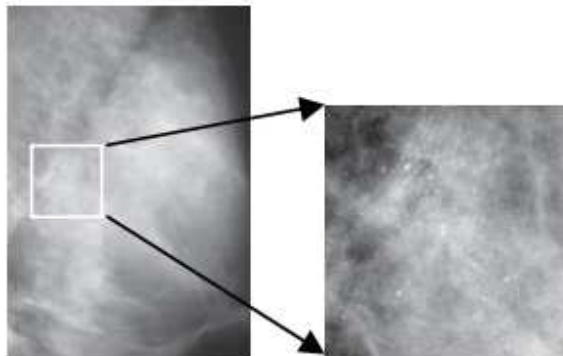
Fuente: Face detection based on skin color segmentation and SVM classification.

5.3. EXPLORANDO LA ROTACION INVARIANTE CON EL SVM CLASIFICADOR PARA LA DETECCION DE MICROCLASIFICACIONES²⁸

Los agrupamientos de las microcalcificaciones (*clustered microcalcifications, MC*), son muy importante para la detección temprana de cáncer de seno en mujeres, las MC son un pequeño deposito de calcio que en las mamografías aparecen como puntos brillantes, y se pueden encontrar entre el 30% y 50% de los casos diagnosticados, los MC individuales son difíciles de encontrar debido a la sutileza en apariencia, la variación en forma, tamaño y tejido circundante.

Se tratara el reconocimiento de estas microcalcificaciones con tres tipos de SVM, las tradicionales, las VSVM (*Virtual Support Vector*) y TV-SVM (*Tangent Vector SVM*), la prueba se realizó con en un set de datos de 200 mamografías que contienen un total de 5211 MC.

Figura 39. Mamografía con calcificaciones.



Fuente. Exploiting totational invariance with SVM classifier for microcalcification detection.

5.3.1. Virtual Support Vector(VSVM): las VSVM. Es una invariancia rotacional en las SVM, lo que significa es que en un conjunto de entrenamiento se copian algunas muestras de ese conjunto y se rotan, estas son llamadas “ejemplos virtuales”, pero a pesar de la teoría es sencilla, lo que se logra es aumentar la complejidad de la máquina ya que se aumenta la cantidad de datos del conjunto de entrenamiento.

Las VSVM constan de los siguientes tres pasos:

- Obtener los vectores de soporte del conjunto de entrenamiento.

²⁸ YANG yan, WANG juan, Yang yongyi, Exploiting totational invariance with SVM classifier for microcalcification detection, Department of Electrical and Computer Engineering, Illinois Institute of Technology, 2012, 4 p.

- Generar “ejemplos virtuales” con los vectores de soporte y luego combinarse con los vectores de soporte originales, formando un nuevo conjunto de entrenamiento.
- Obtener los nuevos vectores de soporte con este nuevo conjunto.

Hay un problema con este método, muchas muestras de entrenamiento se pierden ya que solo se toma un subconjunto óptimo de los vectores de soporte.

5.3.2. Tangent Vector SVM (TV-SVM): Se utiliza un vector tangente dx_i

$$dx_i \equiv \lim_{t \rightarrow 0} \frac{1}{t} (L_t x_i - x_i) \quad (5.3.1)$$

donde:

x_i es la muestra de entrada.

$L_t x_i$ se expresa la rotación de la muestra de entrada con su transformación donde t es el parámetro asociado al ángulo de rotación.

Se incorpora esta propiedad a la función objetivo de las SVM y se modifica de la siguiente forma.

$$J(w, \xi) = \frac{1}{2} (\gamma \|w\|^2 + (1 - \gamma) \sum_{i=0}^N (w^T dx_i)^2) + C \sum_{i=0}^N \xi_i \quad (5.3.2)$$

donde el parámetro γ entre 0 y 1 es un control entre las SVM e invariancias.

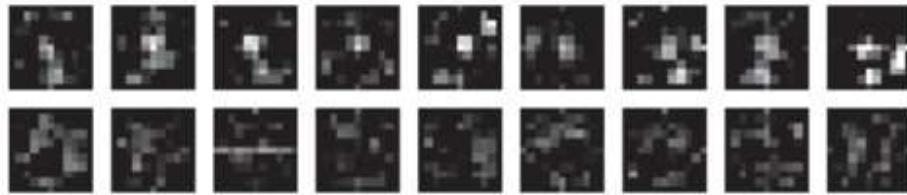
5.3.3. Set de datos: el dataset es una colección de datos del Departamento de Radiología de la Universidad de Chicago, el cual consista de un total de 200 diferentes mamografías, que contienen múltiples MCs, de dimensiones de 1024x1024 o 512x512 pixeles, de una resolución de 0.1 mm/pixel y escala de grises de 10 bits.

En este experimento el dataset es un grupo aleatorio de datos particionado en dos subsets, 50 mamografías por entrenamiento y 150 de prueba.

Las MC simples son extraídas de todas las MC conocidas en cada mamografía, de 5211 MC, 956 MC son extraídas para el set de entrenamiento y 4159 MC son extraídas para prueba.

Para la generación de las muestras virtuales, para SVSM, se aplica una rotación incremental cada 45 grados, de un grupo “MC presente” de nos mas de 40 muestras y un grupo de muestras “MC ausente”, que son muestras de forma aleatoria de la zona de tejido de fondo de la mamografía.

Figura 40. Imágenes con microcalcificaciones (primera fila) y fondo de radiografía (segunda fila).



Fuente: Exploiting totational invariance with SVM classifier for microcalcification detection.

Para las TV-SVM el parámetro γ es:

$$\gamma = \frac{\alpha}{S + \alpha(1-S)} \quad (5.3.3)$$

$$S = \sum_{i=1}^n \|dx_i\|^2 \quad (5.3.4)$$

donde $\alpha = 0.9$

Para esta técnica se uso con un *kernel* RBF Gausiano.

5.3.4. Resultados: De las técnicas consideradas para este caso, se consideró que las TV-SVM es la más efectiva para la aplicación debido a que genera un porcentaje menor de falsos positivo.

Tabla 12. resultado de la comparación entre las tres técnicas.

Clasificador	SVM	VSVM	TV-SVM
Generalización del error	0.0132	0.0108	0.087
Parámetro δ	50	50	25
Parámetro C	10	1	0.1

Fuente: Exploiting totational invariance with SVM classifier for microcalcification detection.

5.4. DETECCIÓN HUMANA CON PC-SVM²⁹

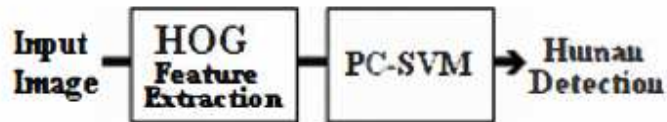
En la detección humana hay una serie de factores que hay que considerar, la apariencia variable, iluminación inconsistente y una gran variedad de poses que se pueden adoptar.

Por ese motivo se usa una técnica llamada histogramas normalizados orientados al gradiente (HOG), el objetivo de este algoritmo es implementar una imagen dividida en ventanas dentro de pequeñas celdas, para cada celda acumulada local hay 1-D histogramas de gradiente de dirección u orientaciones ventaja sobre los pixeles de la celda, para las celda (Cx,Cy) se tiene:

$$HOG(cx, cy) = \frac{|f(cx-1,cy-1)|f(cx,cy-1)|f(cx-1,cy)|f(cx,cy)|}{E(cx-1,cy-1)+E(cx,cy-1)+E(cx-1,cy)+E(cx,cy)} \quad (5.4.1)$$

Donde $f(Cx,Cy)$ indica la magnitud ponderada del gradiente que cada pixel y $E(Cx,Cy)$ la energía de la celda, note que el operador $|$ es la concatenación de características.

Figura 41. Estructura identificación humana.



Fuente: Human detection base on PC-SVM.

5.4.1. Probabilistic constraint SVM (PC-SVM): la diferencia a las SVM tradicionales es la expresión:

$$d_i(w^T x_i + b) \geq F_i^{-1}(\beta_i) \quad (5.4.2)$$

Donde $\beta_i = 1 - \delta$ y $F_i^{-1}(\cdot)$ es el inverso de la función distribución con las variable $u_i - \xi_i$ con $i = 1, \dots, n$.

²⁹ MEYSAM hosseini, seyyed, NASRABADI, abbas, Human detection base on PC-SVM, IEEE, 2010, 4 p.

$$F^{-1}(\beta_i) - d_i(wx_i + b) \leq 0 \quad (5.4.3)$$

Para la optimización se continúa:

$$J(w, b, \alpha, \xi) = \frac{1}{2}w^T w + c \sum_{i=1}^n \xi_i + \sum_{i=1}^n \xi_i [F^{-1}(\beta_i) - d_i(wx_i + b)] \quad (5.4.4)$$

$$\frac{\partial J}{\partial w} = w - \sum_{i=1}^n \alpha_i d_i x_i = 0 \quad (5.4.5)$$

$$\frac{\partial J}{\partial b} = \sum_{i=1}^n \alpha_i d_i = 0 \quad (5.4.6)$$

$$\frac{\partial J}{\partial \xi} = c = 0 \quad i = 1, \dots, n \quad (5.4.7)$$

Para la resolución de este problema pasa a un problema dual.

Maximizar

$$\sum_{i=1}^n \alpha_i F^{-1}(\beta_i) - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j d_i d_j x_i x_j \quad (5.4.8)$$

$$\sum_{i=1}^n d_i \alpha_i = 0 \quad \text{con } i = 1, \dots, n \leq \alpha_i \quad (5.4.9)$$

Usando la teoría de optimización KKT.

$$\sum_{i=0}^n \alpha_i d_i = 0 \quad 0 \leq \alpha_i \quad (5.4.10)$$

$$F^{-1}(\beta_i) - d_i(wx_i + b) \leq 0 \quad i = 1, \dots, n \quad (5.4.11)$$

$$\alpha_i [F^{-1}(\beta_i) - d_i(wx_i + b)] \leq 0 \quad i = 1, \dots, n \quad (5.4.12)$$

Los multiplicadores de lagrange se denotan α_i y es posible la optimización computacional usando el vector peso w_0 .

$$w_0 = \sum_{i=1}^n \alpha_i d_i x_i \quad (5.4.13)$$

$$b = \begin{cases} F^{-1}(\beta_i) - Wx_i \\ -F^{-1}(\beta_i) - Wx_i \end{cases} \quad (5.4.14)$$

$$i=1, \dots, n \quad \xi = 1 - d_i(w^T x_i + b) \quad (5.4.15)$$

5.4.2. Dataset: se tomaron muestras humanas y no humanas, el tamaño de las celdas son de 8x8 pixeles y cada imagen tiene el tamaño de 32x96 pixeles, en cada imagen se genera 33 celdas de características.

En general se tiene 825 vectores humanos y 825 no humanos como vectores de entrenamiento y 800 vectores que son aleatoriamente seleccionados para entrenar el clasificador.

5.4.3. Resultado: comparado con las SVM tradicionales, el resultado de esta prueba arrojo superioridad de este nuevo algoritmo, donde el resultado de reconocimiento fue de un 85% de con las PC-SVM, respecto a un 41% de las SVM tradicionales.

Figura 42. Resultado del PC-SVM.



Fuente: Human detection base on PC-SVM.

5.5. ESTUDIO DE SVM COMPARADA CON OTROS MÉTODOS DE CLASIFICACIÓN DE TEXTO³⁰

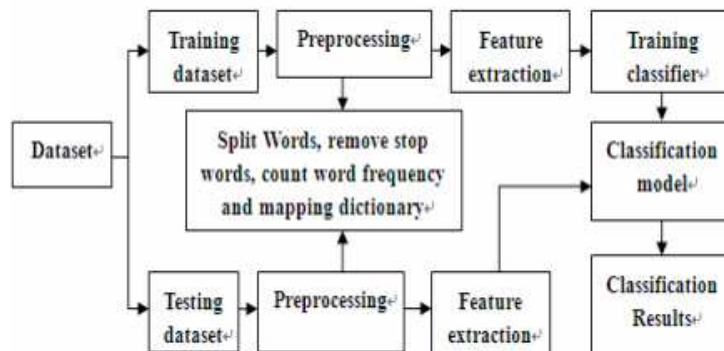
Se desarrolla un procesador de información de texto con las SVM y con otros métodos de clasificación el método de árbol de decisión, el método KNN y el método de Naive bayesianas.

³⁰ ZHIJIE, liu, KUN, lui, XUEQIANG, Lv, SHUICAI, Shi, Study on SVM compared with the other text classification methods, Information Science and Techonologu University, 2010, 4 p.

- El método de árboles de decisión, con una base de datos de muestra se construye un nodo del árbol de decisión y se construye una rama de acuerdo con las propiedades de los diferentes valores del campo y repetir la construcción de próximos nodos y rama, tiene características rápidas y precisas de clasificación.
- Las KNN comienzan con una clase de muestras y cada uno de los datos tienen un tipo de datos estándar, el objetivo es obtener el dato de las muestras, más cercanas de los datos para ser clasificados mediante el cálculo de las distancias de cada muestra de datos a los datos para ser clasificada, si el tipo de datos de la muestra k ocupan mayor parte entonces pertenece a esa clase, pero a mayor número de cálculos la eficiencia se ve reducida.
- El método Naive bayesiana, se basa en el teorema de las bayesianas, y es usado para predecir la posibilidad de miembros de una clase y da la probabilidad de que el texto que pertenece una clase en particular.

5.5.1. El proceso general de un clasificador de texto: el proceso general de clasificación de texto, el cual consta de cuatro módulos, procesador de texto, extracción de características, entrenamiento del clasificador y modelo de entrenamiento.

Figura 43. Clasificador de texto.



Fuente: Study on SVM compared with the other text classification methods.

5.5.2. Extracción de características: Es la parte más importante y reduce el espacio de dimensión del vector texto, simplificando el cálculo.

- En el procesador de texto, se eliminan las palabras divididas, palabras vacías y cuenta la frecuencia de palabras.

- Generalizando diccionario: se cuenta el peso de cada palabra y toma el valor más alto de 7834 palabras para construir un diccionario.
- La extracción de características: a través del mapeo se procesa el texto y se genera el diccionario, para extraer las características del texto para obtener el vector de características.

5.5.3. Obtención de los datos: la escala de los set es de $[-1,1]$, el *Kernel* que se usa es el RBF *Kernel* y se selecciona el mejor parámetro C y d, 700 y 0,5 respectivamente.

Los datos experimentales incluyen 4 clases de documentación, el medio ambiente, deportes, política y arte, los datos de entrenamiento son diferentes para los datos de prueba.

5.5.4. Resultados: se ve que las SVM tienen mejor rendimiento comparada con las otras técnicas de clasificación.

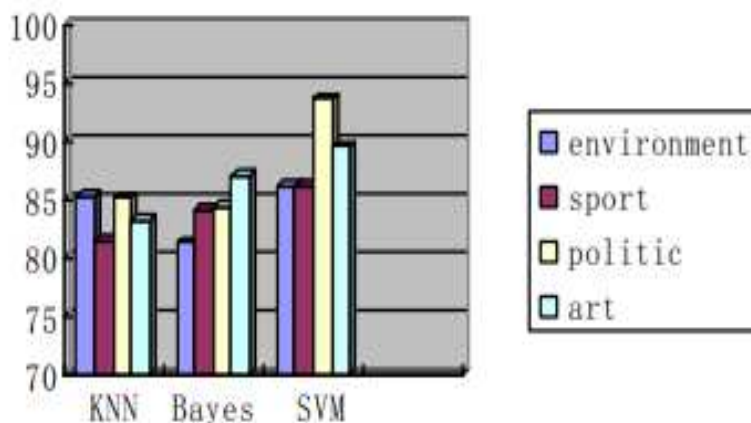
Tabla 13. Comparación entre métodos.

Categoría texto	Entrenamiento/ prueba	SVM	KNN	Naive bayesiana
Ambiente	1800/200	86.26%	85.39%	81.27%
Deportes	1800/200	86.28%	81.51%	84.16%
Política	1800/200	93.78%	85.13%	84.32%
Arte	1800/200	89.58%	83.24%	87.07%

Fuente: Study on SVM compared with the other text classification methods.

En la figura 44 se puede ver que las SVM poseen mejor desempeño que los otros métodos de clasificación, en la clasificación de texto.

Figura 44. Grafica de los resultados.



Fuente: Study on SVM compared with the other text classification methods.

5.6. MAPREDUCE BASADO EN SVM PARALELO PARA FILTRAR SPAM EN LARGA ESCALA³¹

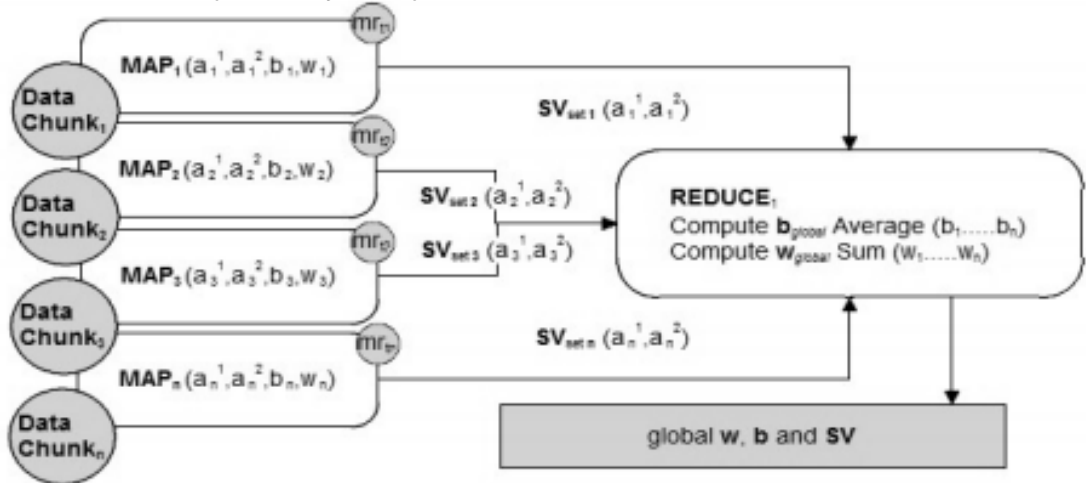
El Spam es una amenaza en la cual el uso de las SVM provee un método novedoso de clasificación del Spam.

5.6.1. MapReduce: Para este desarrollo se utilizó un prototipo de SVM paralelo y un framework computacional llamado MapReduce, lo que permite poner en paralelo procesamiento de conjuntos a escala de internet, lo que se necesita saber es un mapa y una operación para reducirlo, desde la perspectiva lógica los datos se tratan como una clave K, valores pares V, a demás de mapeadores múltiples y reductores pueden ser empleados, a nivel atómico la operación mapa toma un par {K1, V1}, y emite una lista intermedia de parejas {K2, V2}, donde una reducción toma los valores de una misma clave de una lista intermedia y crea otra lista, para la ejecución el mapa debe estar listo y comenzar a reducir operaciones en forma independiente en paralelo, es decir cada función del mapa se ejecuta en paralelo y emite datos respecto a la entrada asociada, mientras el reductor procesa diferentes claves de forma independiente y simultanea.

³¹ GODWIN, caruana, MAOZHEN, Li, MAN Qi, A MapReduce based parallel SVM for large scale Spam Filtering, IEEE, 2011, 4 p.

Los SVM realizan una reducción de los mapas, como se ve en la figura 45, cada parte de datos es reducida por vectores de soporte, en este prototipo la agregación de pesos (w) y el bias (b) se hayan usando suma y promedio.

Figura 45. Diseño MapReduce y SMO paralelo.



Fuente: A MapReduce based parallel SVM for large scale Spam Filtering.

5.6.2. SMO paralelo: este algoritmo permite el paralelismo aplicado al mapeo.

1. **MAP** $\forall j \in \{1, \dots, data_{chunk}\}$
2. **Entrada:** set de entrenamiento de dato x_j , correspondiente a las etiquetas $y_i, \forall i \in \{1, \dots, l\}$
3. **Salida:** peso del vector w_j , cadena α_j, b_j y SV
4. **Inicializar:** $\alpha_i \leftarrow 0, f_i \leftarrow -y_i \forall i \in \{1 \dots l\}$
5. **Calcule:** $b_{high}, I_{high}, b_{low}, I_{low}$
6. **Actualizar:** $\alpha_{I_{high}}$ y $\alpha_{I_{low}}$
7. **Repetir**
8. **Actualizar** $f_i, \forall i \in \{1, \dots, l\}$
9. **Calcule:** $b_{high}, I_{high}, b_{low}, I_{low}$
10. **Actualizar** $\alpha_{I_{high}}$ y $\alpha_{I_{low}}$
11. **Hasta** $b_{low} \leq b_{high} + 2\epsilon$
12. **Actualizar** b_j termino de bias
13. **Guardar** actualización de α_{j1} y α_{j2}
14. **Actualizar** w_j
15. **REDUCE**

16. **Entrada** set de MAP peso de los vectores $w_j \forall_j \in \{1, \dots, data_{chunk}\}$ set de Map bias de $\forall_j \in \{1, \dots, data_{chunk}\}$
17. **Salida** el peso global del vector w_j promedio de b y SV
18. **Calcule** $b_{global} = \sum_{j=1}^{Map} b^{data_{chunk}} / Map_j$
19. **Calcule** $w_{global} = \sum_{j=1}^{Map} b^{data_{chunk}}$

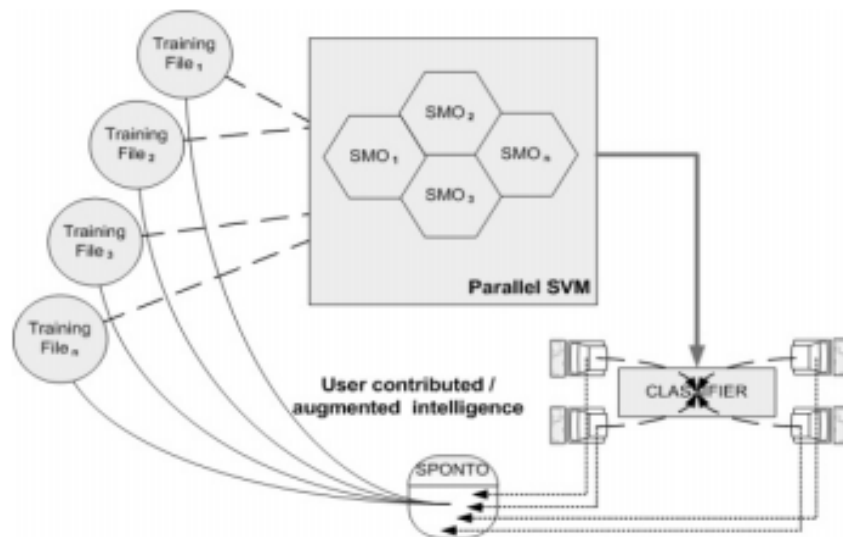
Donde:

Map_j = mapa de MapReduce
 data_{chunk} = dato de entrenamiento asociado con Map_j
 x = elementos de entrenamiento
 y = clase etiquetada para x
 w_j = (Map) local vector peso
 b_j = (Map) local b umbral
 I = set índice de datos de entrenamiento
 α_j = multiplicadores de lagrange
 b_{global} = global b umbral
 w_{global} = global vector de peso

En el algoritmo, en la línea 4 inicializa las estructuras necesarias, primero los multiplicadores de lagrange y luego la función objetivo, de las línea 5 a la 10, las línea 11 verifica la condición de salida, la línea 12 verifica el umbral acordado, las líneas 13 y la 14, se guardan los multiplicadores de lagrange y actualizan el local, la segunda parte, el reductor realiza un cálculo promedio en todas las salidas b respectivas, emitidos por el mapa (Map_j).

También hay un proceso llamado SPONTO (SPamONTOlogy) el cual actúa como base del bucle de retroalimentación para los procesos de formación y clasificación, esto permite mitigar la degradación de la precisión, que se presenta debido a la estrategia de formación, la división de archivos de la estrategia y el cálculo de SVM adoptado.

Figura 46. Diseño general.



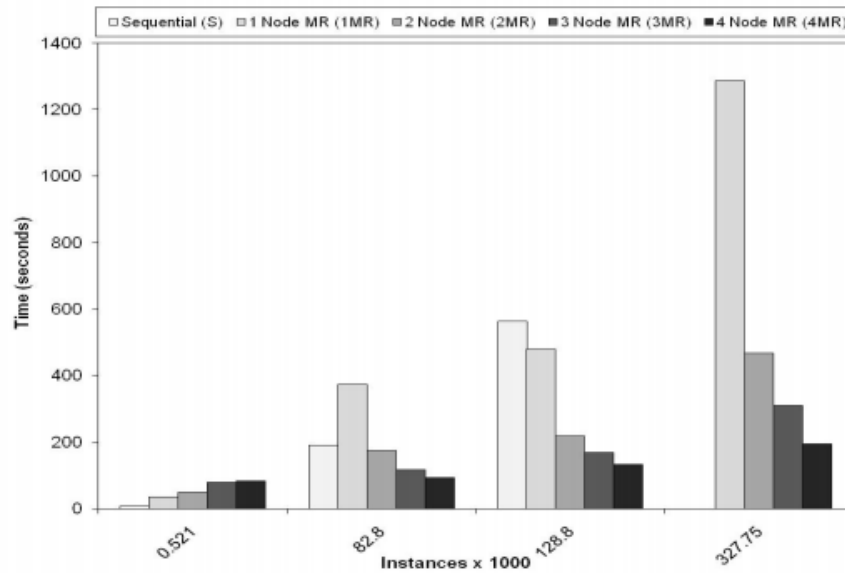
Fuente: A MapReduce based parallel SVM for large scale Spam Filtering.

Para cerrar el ciclo se aumenta el peso de las instancias correctamente clasificadas y se disminuye el peso en los clasificados incorrectamente, fusionar estos casos en la entrada original.

5.6.3. Resultados experimentales: Los datos son extraídos de un SpamBase, se usa un en tren de SMO secuencial, con 128000 instancias en un simple nodo de computador en un tiempo de 563 segundos, con un test secuencias de 327750 instancias falla.

Con 128000 instancias aplicadas a 4 computadoras, estas se procesas en 134 segundos.

Figura 47. Resultado de los nodos.



Fuente: A MapReduce based parallel SVM for large scale Spam Filtering.

En la figura 47. A medida que las instancias crecen, el tiempo aumenta, la tabla 14, desde la primera prueba se ve un resultado es superior al 90%, con un porcentaje de 94.03%, aunque en 4 nodos el resultado baja un poco el resultado sigue siendo alto.

Tabla 14. Resultados del entrenamiento.

	Secuencial	4 Nodos
Correctamente clasificados	94.03%	92.04%
Incorrectamente clasificados	5.97%	7.96%
Tiempo de entrenamiento con 128000 instancias	563.7%	134.5 s

Fuente: A MapReduce based parallel SVM for large scale Spam Filtering.

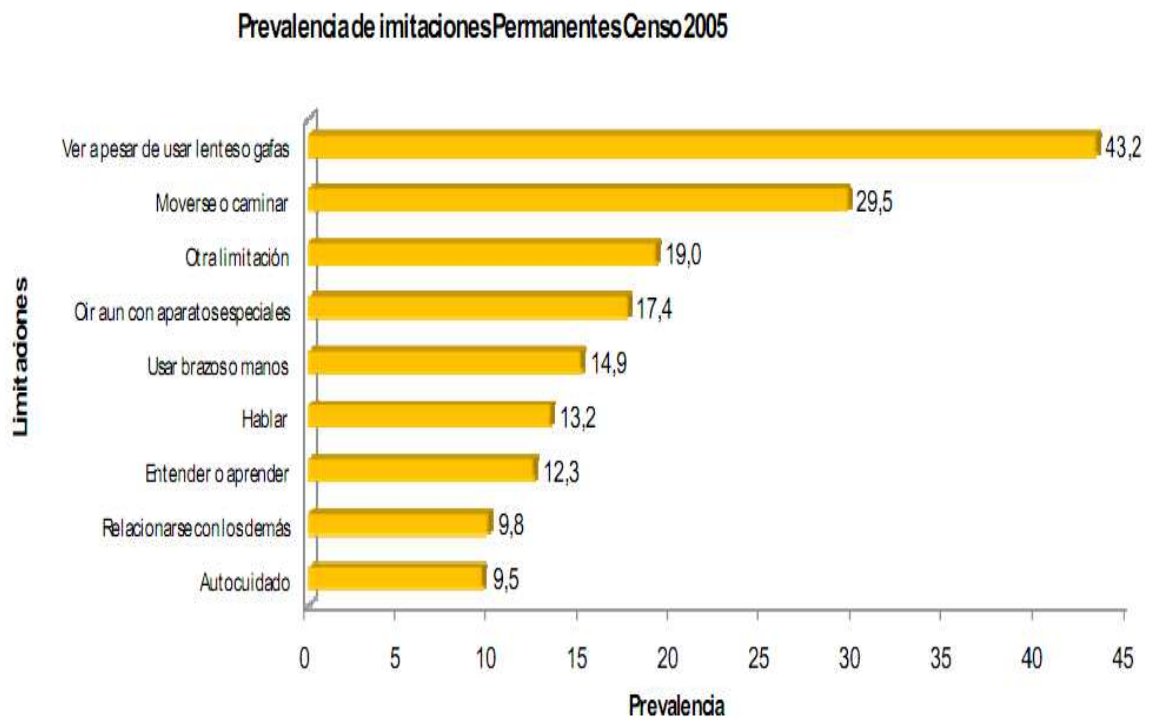
6. APLICACIONES FUTURAS EN EL AREA DE INGENIERÍA DE SISTEMAS

Como ya se vio a lo largo del documento, las Máquina de Soporte Vectorial son muy versátiles para resolver problemas de reconocimiento y clasificación, pero aplicar este método de reconocimiento de patrones a un problema que ha tenido poca visualización, no solo permite crear conciencia de los problemas de las personas no oyentes, también abre un espacio de investigación a nuevas formas de comunicación más directa con personas de esta población.

6.1. RECONOCIMIENTO DE PATRONES ESTATICOS OFF-LINE (VOCALES Y CONSONANTES DEL LENGUAJE DE SEÑAS)

6.1.1. La problemática: Una aplicación la cual podría identificar señas estáticas del lenguaje de señas puede ser muy útil a demás de brindar un apoyo social a una población, la cual la no posee aplicaciones que le faciliten la comunicación con las personas oyentes

Figura 48. Porcentaje de personas por deficiencia en estructuras o funciones corporales.



Fuente: Censo General 2005. Personas con limitaciones permanentes

En la figura 48, se ve el porcentaje de personas con deficiencias en el territorio Colombiano, correspondiendo el sexto lugar con un porcentaje de 13,2 % a las personas con problemas para hablar y un 9,8 % para relacionarse con los demás.

6.1.2. Señas del lenguaje de señas colombiano: para comenzar a desarrollar el problema, se muestra las señas básicas estáticas:

Figura 49. Vocales del lenguaje de señas.



Fuente: http://mail.colombiaaprende.edu.co:8080/recursos/lengua_senas/

Figura 50. Vocales de lenguaje de señas.



Fuente: Autor.

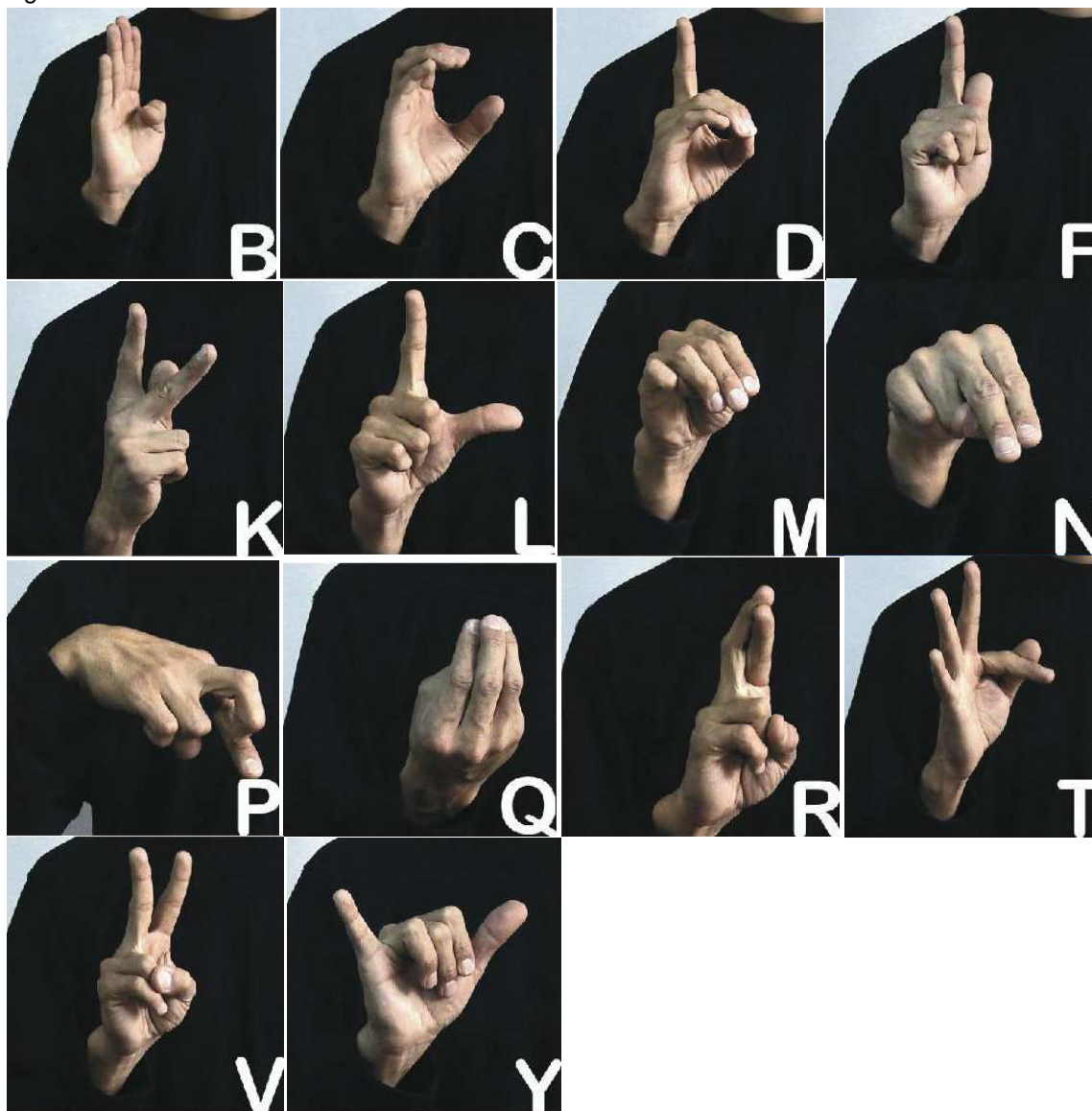
En la figura 49 y 50, se ven las vocales de del lenguaje de señas, estas son básicas y estáticas, pero la mayor parte de las señas son móviles pero permitiendo reconocer estas señas iniciales se da un avance en sensibilizar este lenguajes y a las personas que se comunican a través de el.

Hay que tener en cuenta aspectos de la imagen, como se pueden ver en las figuras anteriores, a pesar de ser la misma seña varia la posición afectando el reconocimiento de esta, para este caso se recomienda para la tomar los datos las siguientes recomendaciones:

- El color de fondo se recomienda negro.

- La distancia de la mano respecto a la cámara debe ser de 40 cm.
- La rotación de la muñeca debe estar entre 0° grados (la mano perpendicular a la cámara) a 90° (paralelo a la cámara).

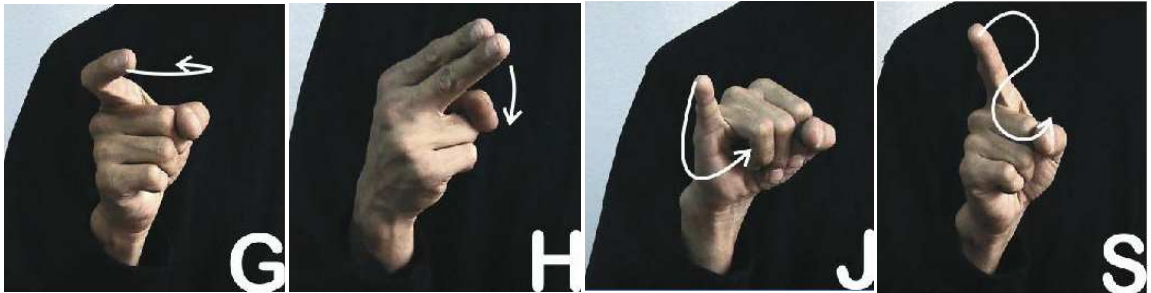
Figura 51. Consonantes estáticas.



Fuente: http://mail.colombiaaprende.edu.co:8080/recursos/lengua_senas/.

En la figura 51, vemos las consonantes las cuales pueden ser reconocida por una Máquina de Soporte Vectorial, si se extraen sus características propias.

Figura 52. Consonantes móviles .



Fuente: http://mail.colombiaaprende.edu.co:8080/recursos/lengua_senas/.

En la figura 52. Presenta el problema principal en el reconocimiento de las señas, debido a que el lenguaje es móvil, en el caso de los números son estáticos hasta el numero 5, de ahí en adelante son móviles, sin contar que la lengua de señas se estructura por secuencia de señas como se ven en la figura 53.

Figura 53. Secuencia de patrones para decir “buenos días”.



Fuente: http://mail.colombiaaprende.edu.co:8080/recursos/lengua_senas/.

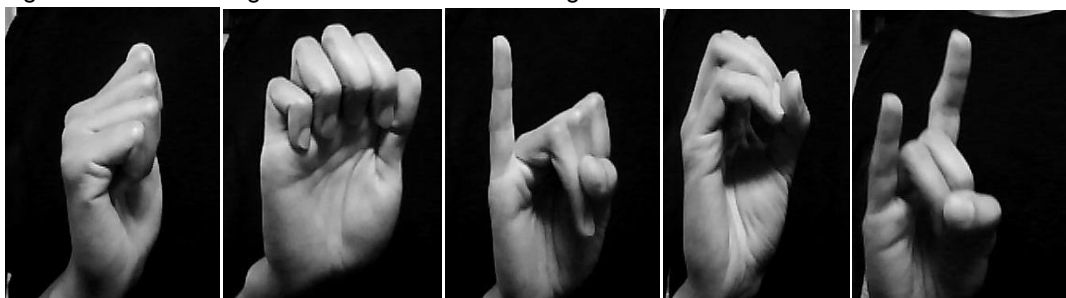
Se plantea que se parta de un sistema que analice una serie de muestras obtenidas con base de fotos siendo un sistema off-line,

6.1.3. Etapas de desarrollo: En este capítulo se dará un esquema de desarrollo de una posible aplicación con Máquina de Soporte Vectorial, como se ve en los apartados anteriores, los datos son representados en un vector de características a priori.

6.1.3.1. Recolección de los datos: se realiza una recolección de los datos en este caso es una conjunto de fotos con las señas de manos de las vocales, para cada seña se realiza una estimación un conjunto de 50 muestras, representando un total 250 muestras basadas en los criterios del apartado.

6.1.3.2. Segmentación: se pule cada una de las imágenes, se pasara escala de grises y solo se deja la mano.

Figura 54. Escala de grises de las vocales de lengua de señas



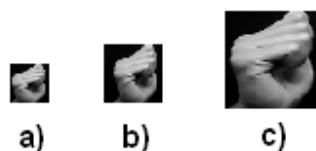
Fuente: Autor.

6.1.3.3. Normalización: para este paso se deciden los criterios para las clases de entrenamiento y prueba, se normaliza respecto a las dimensiones que se trabajara cada clase y sus características

Se debe tener en cuenta la dimensionalidad de las imágenes, y recordando que se necesitan 2 clases de dimensión distinta y características distintas.

Para los datos de entrenamiento son siempre de mayor cantidad, aquí hay que tener en cuenta que la dimensionalidad el dato no puede ser muy grande recordar que en la máquina la dimensionalidad crece.

Figura 55. Distintas dimensiones.



Fuente: Autor.

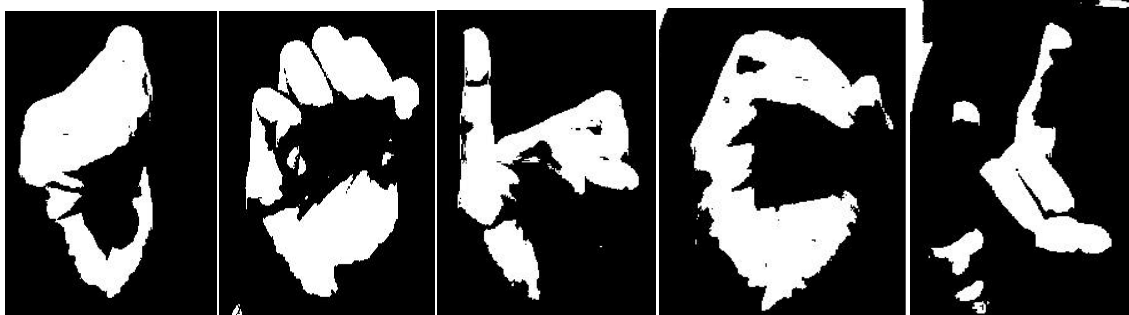
En la figura 55, se puede ver en a) una imagen de 20 x 20 que da una dimensión de 400, b) 30 x 30 de dimensión 900 y c) 50 x 50 de dimensión 2500, debido a la naturaleza de la imagen es recomendable una dimensión media.

Los datos de prueba son siempre de menor cantidad y se recomienda una dimensionalidad menor, siguiendo las recomendaciones anteriores.

6.1.3.4. Extracción de las características: hay varias técnicas para la extracción de las características, se puede realizar una binarización, contorno binario, histogramas, proyección poligonal, etc.

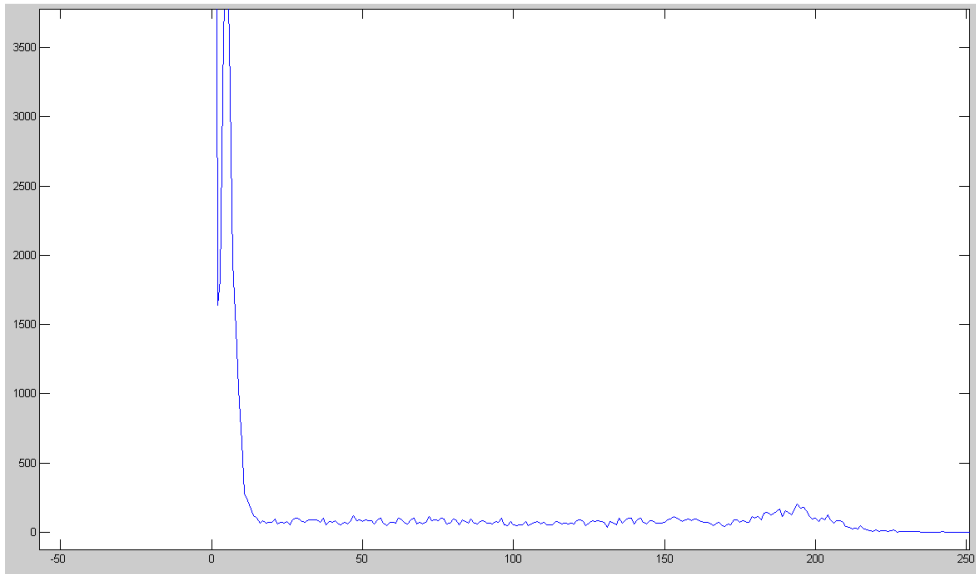
Ahora que ya se definió la dimensionalidad de la imagen se extrae la información de ella, con base a las técnicas anteriores.

Figura 56. Imagen binarizada.



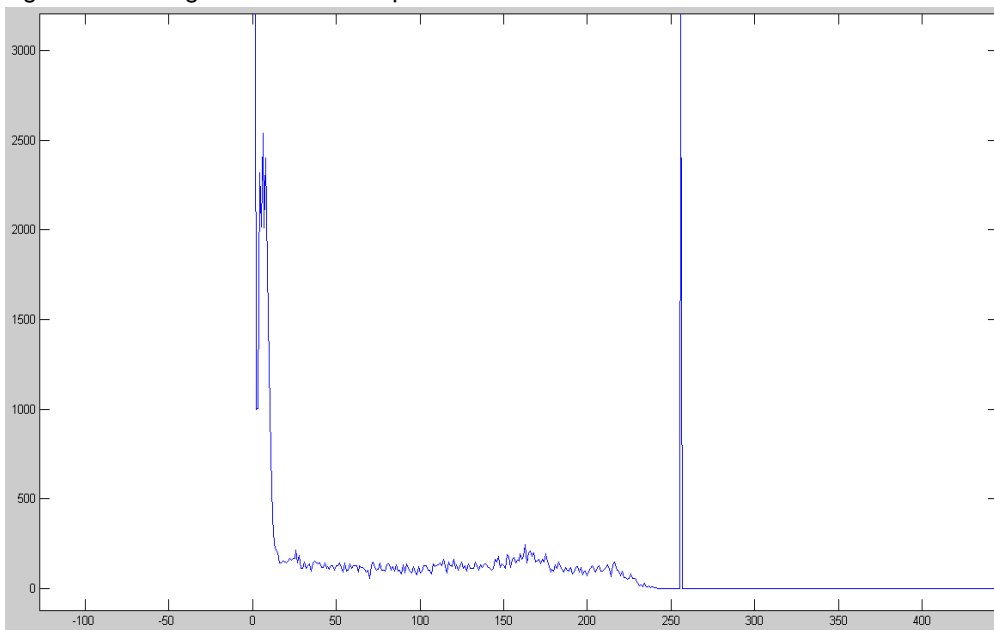
Fuente: Autor.

Figura 57. Histograma de la seña para la letra a.



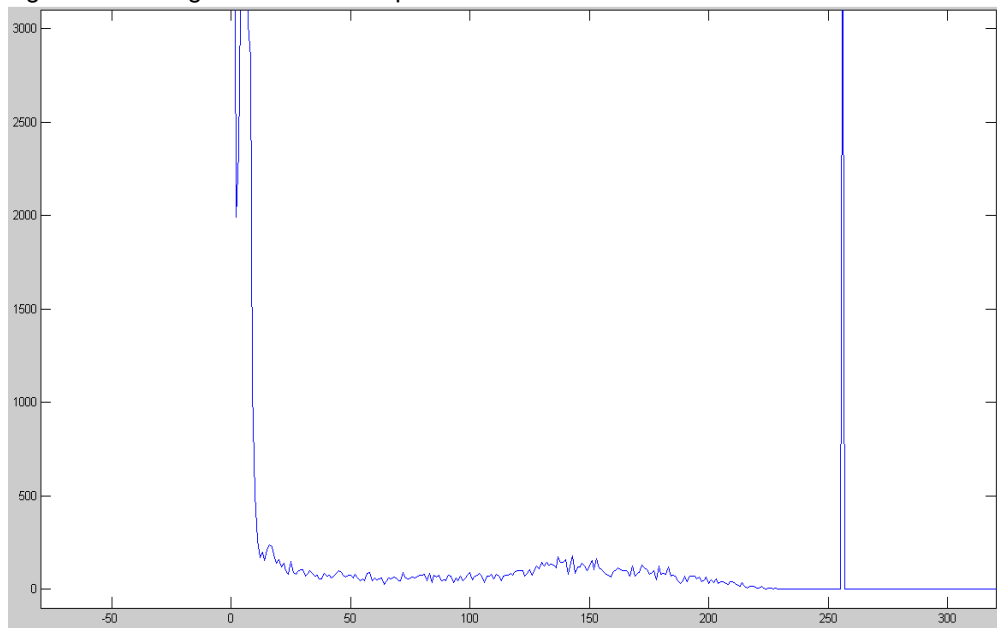
Fuente: Autor.

Figura 58. Histograma de la seña para la letra e



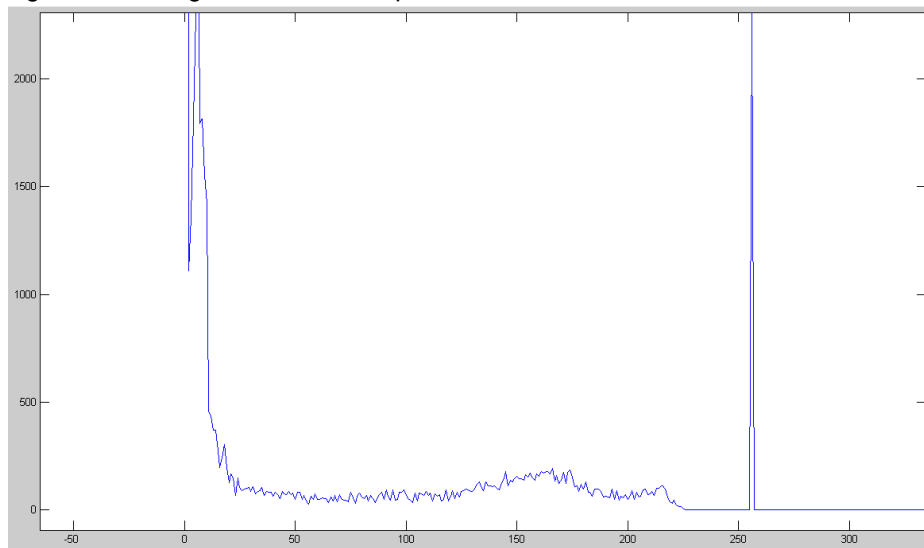
Fuente: Autor.

Figura 59. Histograma de la seña para la letra i.



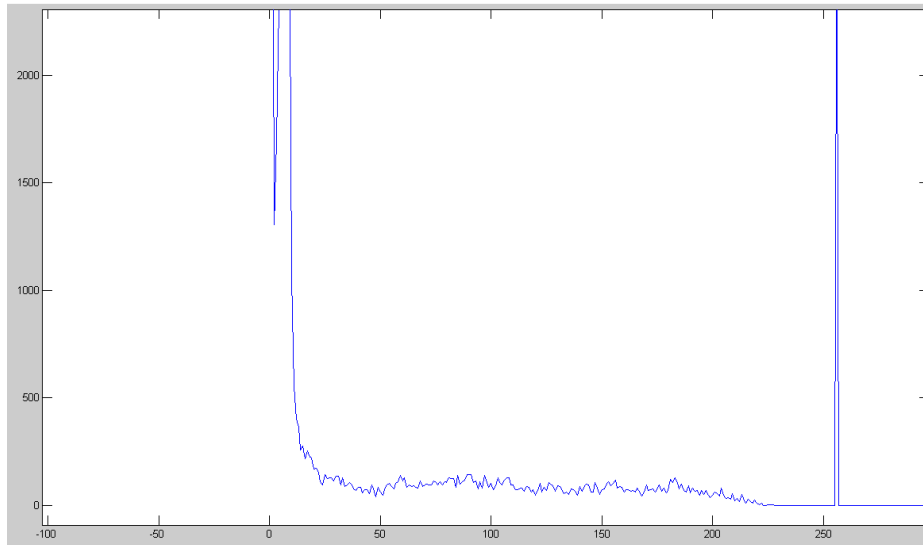
Fuente: Autor.

Figura 60. Histograma de la seña para la letra o.



Fuente: Autor.

Figura 61. Histograma de la seña para la letra u.



Fuente: Autor.

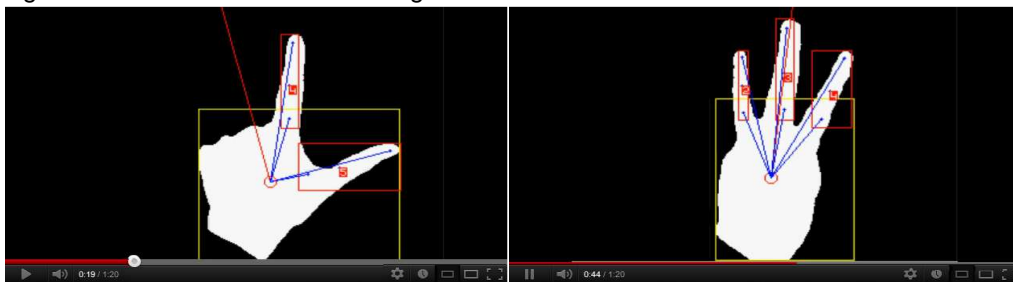
6.1.3.5. Clasificación: una vez obtenidos los datos, se pasa a la etapa del reconocimiento de las Máquina de Soporte Vectorial, se recomienda utilizar *Kernel* polinomiales de grado alto o un *Kernel* de base radial, como se vio en los apartados anteriores tienen mayor desempeño a distintos datos, respecto a otros tipos de *Kernel*.

6.2. RECONOCIMIENTO DE PATRONES ESTATICOS ON-LINE (VOCALES Y CONSONANTES DEL LENGUAJE DE SEÑAS)

El siguiente paso es realizar una aplicación que identifique y reconozca las señas estáticas de forma on-line, para este caso se necesita elaborar un *hand-tracking*.

6.2.1. Detección de las muestras y procesado: *Hand-tracking* es el proceso en que la mano humana es identificada a través de una cámara, se pueden implementar muchos algoritmos los cuales puede realizar la identificación de partes de la mano, se muestra unos ejemplos de *Hand-Tracking*, todos realizados con la librería *openCV*.

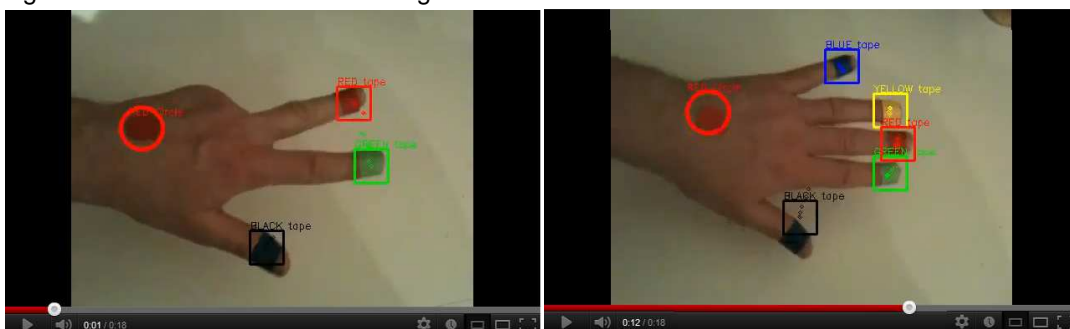
Figura 62. Secuencia hand –tracking I



Fuente: Gesture and fingers recognition.

En la figura 62, se usa una técnica que se basa en el ángulo de cada dedo para el centro de la mano, para cada dedo, se calculan dos ángulos, una línea entre la punta del dedo y el centro de la mano y el otro entre la base del dedo y el centro de la mano, también se tiene en cuenta la distancia entre el centro de la mano y la base del dedo.

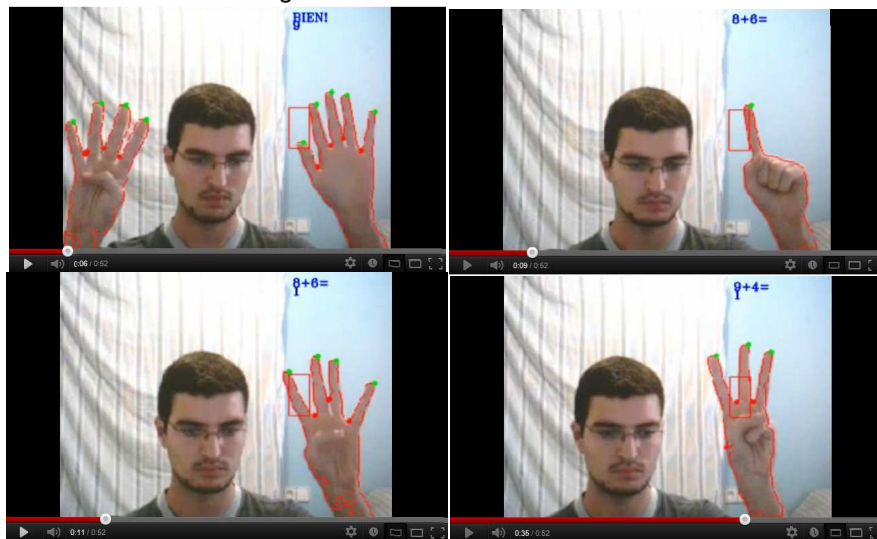
Figura 63. Secuencia hand – tracking II.



Fuente: Motion tracking using open CV.

En la figura 63, en este caso se crea una máscara binaria de un color específico, se toma la posición media de todos los píxeles que quedan en la máscara original, el cuadrado es el centro de esa posición.

Figura 64. Secuencia hand –tracking III.



Fuente: Hand detection.

En la figura 64. Se usa un reconocimiento de objetos, llamado cascada de *Haar*, el filtrado se realizo, con un filtro de color con un pre marcado en el rectángulo.

6.2.2. clasificación: una vez extraída la información en el proceso anterior se procede a realizar la clasificación como se ve en el numeral.

6.3. ANOTACIONES:

A pesar de que se dan ideas iniciales, estos proyectos enfocados a la solución de un problemática social no solo le dan una notoriedad a las SVM, también se construyen las bases para comprender el lenguaje de señas y para permitir avanzar en una la solución computacional, ayudando en un futuro a cerrar la brecha entre los oyentes y los no oyentes.

7. CONCLUSIONES

- Versatilidad, flexibilidad, facilidad, son las palabras que podrían describir las Máquina de Soporte Vectorial, por tal razón presentan una ventaja sobre otras técnicas como las redes neuronales.
- Las Máquina de Soporte Vectorial poseen una gran variedad de tipos, dependiendo de la aplicación algunos tipos son mejores que otros, también es importante tener en cuenta la cantidad de datos a usar, al igual cual es el mejor *Kernel* para una mejor solución del problema.
- Se busca siempre la separabilidad de las clases con la separación optima del margen, también que los datos sean a priori y limitados, también que sus dimensiones sean manejables.
- Aunque las Máquina de Soporte Vectorial poseen un fuerte soporte matemático, entender los conceptos claves de maximización de margen y los *Kernel* es lo más importante de la teoría matemáticas de las Máquina de Soporte Vectorial, con base a ellas se pueden dictar este tema en la materia de Sistemas Expertos como otra opción para resolver problemas de clasificación.
- Como se vio a lo largo del documento se posee una gran variedad de aplicaciones, es cuestión de explotar mas esa metodología de reconocimiento y de clasificación, para avanzar un poco más a nivel académico sobre la optimización de las Máquina de Soporte Vectorial, teniendo aquí una opción de investigación por parte de los estudiantes del programa del pregrado de Ingeniería de Sistemas y Computación.
- Utilizar no solo las Máquina de Soporte Vectorial también otras técnicas de reconocimiento para la aplicación de problemáticas sociales de poblaciones minoritarias debería ser un enfoque a tener en cuenta en el desarrollo de proyectos de grado, hay mas incapacidades físicas que la ceguera.
- Con base a ambas opciones de aplicación en el área de Ingeniería de Sistemas y Computación, se busca que en futuros trabajos se puedan desarrollar dichas ideas, y continuar avanzando en la construcción de un aplicativo el cual reconozca

frases del lenguaje de señas colombiano tomados de cámaras de computadores y posteriormente en teléfonos móviles.

8. BIBLIOGRAFÍA

ABE shibeo, support vector machine for pattern classification, Kobe University, 2005, 343 p.

ATIENZA, Felipe Alonso, tesis doctoral, estudio de los mecanismos de las arritmias cardiacas mediante modelado y procesado robusto digital de señal, Universidad Carlos III de Madrid, mayo de 2008, 268 p.

BETANCOURT Gustavo, Universidad tecnológica de Pereira, LAS MAQUINAS DE SOPORTE VECTORIAL (SVM), Scientia et Technica Año XI, No 27, abril 2005 UTP

BISHOP Chistopher M., Cambrige, Pattern recognition and machine learning, 2006, 701 p.

CHAMASEMANI falah fereshteh, YASHWANT pradas singh, multi-class support vector machine (SVM) classifiers- an application in hypothyroid detection and classification, university multimedia, 2011, 6 p.

Departamento de matemáticas CSI/ITESM, Condiciones Karush-Kuhn-Tucker.pdf, 21 de abril del 2012, 9 p.

ESTRADA betsy mary, MERA Carlos Andrés, support vector machine, universidad nacional de Colombia, 60p

GARCÍA días, ELKIN Eduardo, tesis maestría en ingeniería electrónica, Boosting support vector machines, universidad de los andes, 2005, 59 p.

Gesture and fingers recognition [grabacion sonora], democritus university of thrace, youtube, 2009, 1:20 minutos 360 p, color, http://www.youtube.com/watch?v=6Uw_8Y1RuQQ&feature=related, visitada el día 15 de agosto del 2012

GODWIN, caruana, MAOZHEN, Li, MAN Qi, A MapReduce based parallel SVM for large scale Spam Filtering, IEEE, 2011, 4 p.

Hand detection [grabacion sonora], youtube, 2007, 0:52 minutos 360 p, color, http://www.youtube.com/watch?v=L9xCBvKHDCg&feature=watch_response, visitada el día 15 de agosto del 2012

<http://kernelsvm.tripod.com/> visitada el día 17 de abril del 2012

HWEI-JEN Lin, SHWU-HUEY Yen, JIN-PIN Yeh, Face detection based on skin color segmentation and SVM classification, Tamkang university, 2008, 2 p.

http://mail.colombiaaprende.edu.co:8080/recursos/lengua_senas/, visitada el día 14 de agosto del 2012

MEYSAM hosseini, seyyed, NASRABADI, abbas, Human detection base on PC-SVM, IEEE, 2010, 4 p.

Motion tracking using open CV [grabacion sonora], youtube, 2010, 0:18 minutos 360 p, color, http://www.youtube.com/watch?v=yGtEezvW_KU&feature=related, visitada el día 15 de agosto del 2012

N. Sebe, IRA Cohe, ASHUTOSH Garg, Machine Learning in computer vision, Springer, 2005, 243p

LI cong-cong, GUO ai-ling, LI dan, application research of support vector machine in network security risk evaluation, international symposium on intelligent information technology application workshops, 2008, 4 p.

LUI Yi, ZHENG Yuan, soft SVM and its application to video object extraction, the Ohio state university, 2005, 4 p.

LIPO Wang, Nanyang Technological University, Support vector machine theory and applications studies in fuzzines and soft computing, springer, 2005, 431p

PAJARES, Gonzalo; DE LA CRUZ, Jesús M. UNIVERSIDAD COMPLUTENSE DE MADRID. Visión por computador imágenes digitales y aplicaciones. Alfaomega Ra-Ma. 764 p.

R. Norwak, Lecture 19: The proof of the CV inequality.pdf, 17 de mayo del 2009, 6 p

RASMUSSEN, Carl Edward, WILLIMS, Christopher, Cambrige, Gaussian Processes for machine Learnig, The MIT press 2006, 226 p.

RAMÓN, Manuel Martínez, introducción a los métodos Kernel, Universidad autónoma de Madrid, 29 de abril de 2008, 31 p.

SHAONING Pang, KASABOV nikola, inductive vs trasductive infrencia, global vs local model: SVM, TSVM, and SVMT for gene expression classification problems, Auckland university of tecnology, 2004, 6 p.

SHI Guangzhi, HU junchuan, a dynamic recognition method study using the support vector machine, navy submarine academy, 2007, 5 p.

SNYDER, Wesley E. QI, Hairong. Machine Vision, Cambrige University Press. 434 p.

SIMON Tong, DAPHNE Koller, Computer Science Department, Stanford University, Support Vector Machine Active Learning with Applications to Text Classification, 2001.

VAPNIK, Vladimir, Springer, The Nature of statistical learning theory 1999, 334 p.

VALLE VIDA, Carlos, Vector de soporte y métodos de kernel, universidad técnica Federico santa maría, abril 2009, 94 p.

XUE ming, ZHU changjun. 2009 INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE. A Study and application on machine learning of artificial intelligence. 3p.

YANG yan, WANG juan, Yang yongyi, Exploiting rotational invariance with SVM classifier for microcalcification detection, Department of Electrical and Computer Engineering, Illinois Institute of Technology, 2012, 4 p.

ZHIJIE, liu, KUN, lui, XUEQIANG, Lv, SHUICAI, Shi, Study on SVM compared with the other text classification methods, Information Science and Technology University, 2010, 4 p.