DESARROLLO DE UNA ESTRATEGIA DE CLASIFICACIÓN SUPERVISADA BASADA EN KERNEL EN UN ESCENARIO DE MÚLTIPLES ANOTADORES

DANIA GISELA MARTINEZ FIGUEROA

ERNESTO ALEJANDRO SOLARTE PAZ

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
SAN JUAN DE PASTO
2019

DESARROLLO DE UNA ESTRATEGIA DE CLASIFICACIÓN SUPERVISADA BASADA EN KERNEL EN UN ESCENARIO DE MÚLTIPLES ANOTADORES

DANIA GISELA MARTINEZ FIGUEROA

ERNESTO ALEJANDRO SOLARTE PAZ

Trabajo de grado para optar por el título de ingeniero electrónico

Director
PhD. Edgardo Javier Revelo Fuelagán
Ingeniero electrónico

Co-director
Mg. Jaime Andrés Riascos Salas
Ingeniero Mecatrónico

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
SAN JUAN DE PASTO
2019

NOTA DE RESPONSABILIDAD

“La Universidad de Nariño no se hace responsable por las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor.”

Acuerdo 1. Artículo 324. Octubre 11 de 1966, emanado del honorable Consejo Directivo de la Universidad de Nariño.

NOTA DE ACEPTACIÓN:

_____
_____
_____
_____
_____
_____

_____
Firma del presidente del jurado

_____
Firma del jurado

_____
Firma del jurado

San Juan de Pasto, 5 de noviembre de 2019

AGRADECIMIENTO

“Agradezco a cada una de las personas que en algún momento de mi proceso académico me han ayudado de una manera u otra. A aquellos profesores que nos enseñaron e hicieron por nosotros todo lo que estaba a su alcance; especialmente a nuestro director de trabajo de grado Javier Revelo Fuelagán y a nuestro co-director Jaime Riascos Salas. También agradezco a la Universidad de Nariño por abrirle sus puertas a los jóvenes soñadores de nuestra región, con anhelos de un mejor futuro. Finalmente, agradezco a nuestros amigos por haber afrontado junto a nosotros cada uno de los retos de esta bella ingeniería”.

Dania Gisela Martínez Figueroa.

“Agradezco a mis padres, que siempre me han apoyado a lo largo de este camino, por todos los valores que en mi inculcaron y por haberme brindado todo lo que he necesitado para ser un profesional y una buena persona. A mis primos que han sido hermanos para mí y me han acompañado en los buenos y malos momentos. Quiero agradecer a mis profesores que me hicieron crecer como profesional y como persona. Finalmente, a mis compañeros y amigos de pregrado quienes me enseñaron muchas cosas y en los momentos más difíciles estuvieron conmigo para salir juntos adelante”.

Ernesto Alejandro Solarte Paz.

DEDICATORIA

“A mi hermosa madre María Eugenia Figueroa, quien ha sido mi mejor ejemplo de fortaleza, paciencia, ternura y amor; ella es luz y sin ella no habría podido afrontar cada duro momento de mi vida. A mi querido padre Enrique Martínez por haber hecho todo por protegerme, por enseñarme a ser rigurosa en cada tarea que me propongo realizar y por haber sido tan incondicional a lo largo de este proceso. A mi hermano Norman Martínez, a quien admiro inmensamente y quien ha sido mi compañero de mi vida siempre. Es mi familia la razón por la que sigo en pie, ustedes me han mostrado el amor más grande que puede existir y por ustedes seguiré fuerte en este camino”.

Dania Gisela Martínez Figueroa.

“A mis padres, mis motores de vida, esto es suyo, recogen lo que en mí sembraron. A mis abuelos y en especial mis abuelas que soñaban verme cumplir esta meta y aunque no pueden acompañarme físicamente quiero que se sientan felices y orgullosas donde quiera que estén”.

Ernesto Alejandro Solarte Paz.

RESUMEN

En el campo del reconocimiento de patrones, uno de los aspectos más importantes para la clasificación supervisada es la base de datos de entrada, la cual debe estar correctamente etiquetada. En algunos casos, las propiedades intrínsecas de los objetos pueden generar confusión a la hora de obtener buenas etiquetas, por lo que se recurre a consultar la opinión de varios expertos para dicho propósito; por ejemplo, en medicina, para algunas enfermedades complejas es necesario consultar a varios profesionales y así, tener mayor certeza del diagnóstico médico. La dificultad que esto representa es que no siempre existe consenso entre ellos o incluso, algunos no cuentan con la experiencia en el área de interés para dar buenas etiquetas. Es por ello que, en este trabajo de grado se creó un método capaz de regular la influencia de cada uno de los etiquetadores en la creación del modelo de predicción, de forma que se garantice un adecuado aprendizaje del sistema inteligente. Para dicho fin, se entrenó uno de los métodos más conocidos en aprendizaje supervisado, *Support Vector Machine* (SVM), con matrices kernel de tipo supervisado, que contienen información tanto de las etiquetas de cada anotador como de los datos de entrenamiento. La introducción de dichas matrices al clasificador se realizó por medio de la suma ponderada de las mismas, con factores de ponderación obtenidos con métodos de análisis de relevancia de variables aplicados a las matrices de cada experto. Los experimentos realizados en un entorno de múltiples expertos mostraron un menor porcentaje de error para los kernel con función exponencial, así como mejor precisión y AUC (área bajo la curva). Este método, además, constituye un gran avance dentro de este campo puesto que permite el trabajo con bases de datos en un escenario de múltiples expertos gracias a las funciones kernel usadas.

ABSTRACT

In the field of pattern recognition, one of the most important aspects for supervised classification is the input database, which must be correctly labeled. In some cases, the intrinsic properties of the objects can generate confusion to obtain good labels, so it is used to consult the opinion of several experts for this purpose. For example, in medicine, for some complex diseases it is necessary to consult several professionals and thus have greater certainty of the medical diagnosis. The difficulty that this represents is that there is not always consensus among them or even, some do not have the experience in the area of interest to give good labels. The objective of this thesis is to create a method that allows regulating the influence of each of the taggers in the creation of the prediction model, in order to ensure adequate learning of the intelligent system. To that end, one of the most well-known methods in supervised learning, Support Vector Machine (SVM), was trained with supervised kernel matrices, which contain information on both the labels of each scorer and the training data. The introduction of these matrices to the classifier was carried out by means of their weighted sum, with weighting factors obtained by methods of analysis of relevance of variables applied to the matrices of each expert. Experiments performed in a simulated multi-expert environment showed a lower error rate for kernels with exponential function, as well as better accuracy and AUC. This method also constitutes a great advance in this field since it allows working with databases in a scenario of multiple experts thanks to the used kernel matrices.

CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

GLOSARIO

Clase: marca o etiqueta de objetos que tienen características comunes.

Características: Se refiere a los atributos que describen las instancias del conjunto de datos.

Algoritmo: Conjunto definido de reglas o procesos que llevan a la solución de un problema en un número determinado de pasos.

Anotador: ente que asigna las etiquetas de los datos según su criterio. También referido en este documento como experto o etiquetador.

Clasificación: crear modelos para separar datos de diferentes clases y de la misma forma predecir las clases de nuevos datos.

Modelo de predicción: regla de decisión que un algoritmo construye para predecir etiquetas nuevas a partir de un conjunto de entrenamiento.

SVM: clasificador de tipo supervisado basado en vectores de soporte, cuyo principal propósito es crear un modelo de predicción maximizando el margen de decisión.

Funciones Kernel: permiten transformar datos de un espacio de características de determinada dimensión a un espacio de dimensión mayor, generalmente infinita.

Curvas ROC: representación gráfica del desempeño de un clasificador, que expone la tasa de verdaderos positivos frente a la tasa de falsos positivos. Muy útil para la visualización de los resultados.

AUC: área bajo la curva ROC, es un indicador que se puede interpretar como la probabilidad de que el clasificador prediga nuevas etiquetas correctamente.

INTRODUCCIÓN

Diversos campos de la ciencia y la tecnología están implementando técnicas de aprendizaje automático para analizar gran cantidad de información proveniente de diferentes casos de estudio, como se explica en el libro *Pattern Recognition*[1]. Una definición del aprendizaje automático dada en Redes neuronales y aprendizaje automático[2] indica que es una rama de la inteligencia artificial que se encarga del desarrollo de algoritmos computacionales, que buscan darle a una máquina la capacidad de "aprender", propia de un ser humano; en otras palabras, la máquina se programa de manera que logre crear modelos de predicción y tomar decisiones autónomamente. Esta disciplina tuvo sus inicios varias décadas atrás; sin embargo, el interés en esta área ha resurgido gracias a los volúmenes de datos disponibles, al procesamiento computacional más económico y poderoso y al almacenaje de datos más asequible.

Dentro del aprendizaje de máquina se encuentra el reconocimiento de patrones, el cual según Bishop[3] se ocupa de la detección automática de información importante en un conjunto de datos, difícilmente perceptible para el ser humano en un escenario normal. Esta metodología se ha usado para abordar un sinfín de problemas: detección de fraude en el uso de tarjetas de crédito[4], segmentación de imágenes[5], identificación de emociones[6], estudio de aguas subterráneas para predecir variaciones del nivel de agua en acuíferos[7], para brindar soporte a los profesionales en el diagnóstico de enfermedades[8], entre otros.

Una de las tareas principales del reconocimiento de patrones es la clasificación. En ella se desarrollan algoritmos capaces de crear reglas de decisión que separen conjuntos de datos en varias clases diferentes. Uno de los métodos más conocidos según Ricardo Henao[9], son las máquinas de vectores de soporte (SVM por sus siglas en inglés *Support Vector Machine*) que constituyen un tipo de clasificador

[1] THEODORIDIS. Sergios; KOUTROUMBAS- Konstantinos. Pattern recognition. Segunda edición. San Diego: Elsevier, 2003.
[2] Haykin. Simon. Redes neuronales y aprendizaje automático. Tercera edición. Hamilton: Prentice Hall PTR, 1994.
[3] Bishop. Christopher. Pattern Recognition and Machine Learning. New York: Springer-Verlag, 2006.
[4] LODHIA, ZeeshanAhmad; RASOOL, Akhtar; HAJELA, Gaurav. A survey on machine learning and outlier detection techniques. En: International journal of computer science and network security, 2017, vol. 17, no 5, p. 271-276.
[5] SONG, Mingjun; CIVCO, Daniel. Road extraction using SVM and image segmentation. En: Photogrammetric Engineering & Remote Sensing, 2004, vol. 70, no 12, p. 1365-1371.
[6] BALEÓN, Emmanuel. Sistema de visión artificial para el reconocimiento de emociones faciales aplicado al estudio de expresiones del usuario. *Repositorio nacional conacyt*, 2017.
[7] TAORMINA, Riccardo; CHAU, Kwok-Wing; SETHI, Rajandrea. Artificial neural network simulation of hourly groundwater levels in a coastal aquifer system of the Venice lagoon. En: Engineering Applications of Artificial Intelligence, 2012, vol. 25, no 8, p. 1670-1676.
[8] CAIPE, Angela; MUÑOZ, Jorge; PELUFFO, Diego. Machine Learning for the prediction of preterm pregnancy using EHG signals. Trabajo de grado (Ingeniero electrónico). Pasto, 2016. Universidad de Nariño. Facultad de ingeniería. Departamento de Electrónica.
[9] HENAO, Ricardo; HURTADO, Jorge; CASTELLANOS, German. Selección de hiperparámetros en máquinas de soporte vectorial utilizando adaptación de matriz de covarianza. En: Scientia et technica, 2005, vol. 1, no 27.

supervisado muy versátil. Henao además, afirma que las SVM han sido muy importantes dentro de la clasificación debido a que se fundamentan sobre unas bases matemáticas muy sólidas dadas por Vapnik en 1995.

Esta técnica de aprendizaje de máquina creada por Vapnik[10] ha sido escogida como base para este trabajo de grado debido a que presentan muchas ventajas frente a otros tipos de clasificador, por ejemplo: el espacio de búsqueda de su problema de optimización tiene un sólo mínimo global, el entrenamiento es muy eficiente, es muy robusto para la generalización y tiene menos necesidad de técnicas heurísticas para su entrenamiento.

Ahora bien, tanto las SVM como muchos otros algoritmos de clasificación se construyen generalmente a partir de un conjunto de entrenamiento con un vector de etiquetas confiable. A menudo dicho vector es entregado por un experto que se basa en su propio criterio para definir la etiqueta de cada elemento del conjunto. Sin embargo, existen situaciones donde esta labor se encarga a varios expertos cuya confiabilidad es incierta, quienes podrían entorpecer el proceso de entrenamiento y, subsecuentemente la clasificación. Este problema generalmente se conoce como "aprendiendo de múltiples profesores", "múltiples anotadores", "múltiples expertos", "múltiples etiquetadores", etc; como se expone en el trabajo de Yan[11].

Un ejemplo de esta situación se presenta en la clasificación de páginas web, donde se busca determinar cuáles podrían ser de mayor interés para los usuarios y cuáles han sido manipuladas por programas informáticos para aumentar su popularidad. La información que se requiere para el entrenamiento se puede obtener directamente de los comentarios de los usuarios, o simplemente extrayendo información del número de clics de las páginas; de modo que, cada usuario representa un etiquetador diferente. Otro caso muy familiar se observa en la medicina, donde a menudo se presentan diferencias entre varios profesionales frente al diagnóstico de determinada enfermedad. Tales casos evidencian la necesidad de desarrollar técnicas de clasificación capaces de dar etiquetas con precisión, incluso en casos donde no hay consenso entre los etiquetadores.

Los trabajos que se han desarrollado frente a este inconveniente han abordado el caso desde varios enfoques: en el trabajo de Sheng se usa *repeated labeling* para mejorar la calidad de las etiquetas, a manera de preprocesamiento[12]; en la propuesta de Dekel se asume que cada experto debería contribuir

[10] BOSER, Bernhard; GUYON, Isabelle; VAPNIK, Vladimir. A training algorithm for optimal margin classifiers. En: Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory. 2003. p. 144-152

[11] YAN, Yan. Active learning from crowds. En: ICML. 2011. p. 1161-1168.

[12] SHENG, Victor; PROVOST, Foster; IPEIROTIS, Panagiotis. Get another label? Improving data quality and data mining using multiple, noisy labelers. En: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008. p. 614-622.

aproximadamente el mismo número de vectores de soporte del clasificador SVM[13], de manera que esta consideración conforma una restricción que solo afecta a etiquetadores maliciosos y de baja calidad; Yan aplica técnicas de *active learning* para identificar los etiquetadores más aptos para dar etiquetas, basándose en un modelo probabilístico[14]; por su parte, Murillo propone asignarle a cada experto un factor de penalización, que aumente la función objetivo de SVM cuando las etiquetas de cada experto sean incorrectas[15]; en el trabajo de Guan se usan redes neuronales, en las que los expertos son asignados a unos pesos[16]; finalmente, en el trabajo de Gil se presenta un enfoque basado en *kernel annotator relevance análisis[17] (KAAR).*

Aunque los resultados reportados en dichos trabajos han sido favorables, la exploración de matrices kernel para la resolución de este problema ha sido muy escasa. Es por ello que en esta investigación se exploró un método capaz de sobrellevar el problema de múltiples etiquetadores para datos no linealmente separables mediante matrices kernel, desarrollado sobre un clasificador SVM multi-clase. En este sentido, se adoptó un criterio de confiabilidad que consiste en asignarle a cada experto un factor de ponderación que regula la influencia de este mismo sobre el clasificador. Este proceso se hace bajo la suposición de que no todos los etiquetadores poseen la misma credibilidad, algunos tendrán una mejor percepción sobre los datos por lo que entregarán mejores etiquetas, y otros podrían no tener mucha experiencia y darán etiquetas ruidosas.

El enfoque propuesto en este trabajo de grado es similar al de Gil en el sentido de que se realiza una combinación lineal de matrices kernel por medio de unos factores de ponderación. Sin embargo, dichos factores son calculados bajo diferentes criterios. Además, el enfoque de este trabajo de grado propuso usar factores de ponderación para regular la influencia de cada anotador en la construcción del modelo, mientras que en el trabajo de Gil se usaron para la predicción de nuevas etiquetas. Es por tal razón que en dicho trabajo se deben entrenar un número de clasificadores igual al número de anotadores, lo cual supone una limitación no deseable.

El método propuesto fue analizado con diferentes experimentos cuyos resultados se compararon con los de un clasificador SVM convencional. En dicho análisis, fue

[13] DEKEL, Ofer; SHAMIR, Ohad. Good learners for evil teachers. En: Proceedings of the 26th annual international conference on machine learning. ACM, 2009. p. 233-240.
[14] YAN, Yan. Active learning from crowds. En: ICML. 2011. p. 1161-1168
[15] MURILLO, Santiago. Metodología para el aprendizaje de máquina a partir de múltiples expertos en procesos de clasificación de bioseñales. Trabajo de grado (Master of Engineering - Industrial Automation). Manizales, 2013. Universidad Nacional de Colombia-Sede Manizales. Facultad de Ingeniería y Arquitectura. Departamento de ingeniería Eléctrica, Electrónica y Computación.
[16] GUAN, Melody Y., et al. Who said what: Modeling individual labelers improves classification. En: Thirty-Second AAAI Conference on Artificial Intelligence. 2018.
[17] GIL, J.; ALVAREZ, A.; OROZCO, A. Learning from multiple annotators using kernel alignment. En: Pattern Recognition Letters, 2018, vol. 116, p. 150-156.

el kernel exponencial el que mejor desempeño mostró, tanto para el cálculo de factores de ponderación como para la clasificación. En este sentido, se logró proporcionar una técnica novedosa para manejar el escenario de múltiples expertos con un enfoque de matrices kernel, ideal en el trabajo con datos no separables linealmente.

El trabajo desarrollado se expone a lo largo de este documento. Primero se explica los conceptos básicos sobre clasificación supervisada, máquinas de vectores de soporte, funciones kernel y el problema de múltiples expertos; luego se expone el método desarrollado en esta investigación y finalmente los experimentos que permitieron evaluar el desempeño del método desarrollado.

Modalidad: Investigación.

Línea: Procesamiento de datos.

Este trabajo es de carácter investigativo debido a que propone un nuevo enfoque para la clasificación supervisada en un escenario de múltiples expertos, basado en un análisis del estado del arte que permitió determinar la escasa exploración de métodos kernel para la solución de este problema.

Descripción del problema

Dentro del marco de la clasificación supervisada, es de gran importancia tener un conjunto de datos de entrenamiento adecuado que garantice la confiabilidad de las etiquetas. La tarea de etiquetado puede llegar a ser costosa y a menudo requiere del conocimiento de varios expertos especializados en el campo de aplicación de los datos. Además, dicha tarea no es trivial dado que cada etiquetador tiene (subjetivamente) un diferente enfoque. Particularmente, esta última circunstancia, presenta una mayor complejidad inherente a cada proceso donde se involucren las capacidades sensoriales del ser humano, debido a que los juicios que emite un experto pueden diferir a los de otros según su experiencia, su percepción, sus ideas, etc. En algunas ocasiones, puede ocurrir que algunos anotadores no tengan la experticia requerida o incluso, tengan malas intenciones, como se expone en el trabajo *Good learners for evil teachers*[13].

Es debido a este problema que se necesitaron crear métodos de aprendizaje de máquina capaces de construir una regla de decisión basada en la información útil de cada etiquetador. Por su parte, en el trabajo de Murillo se muestra un ejemplo de cómo las SVM facilitan la regulación del impacto de cada experto en la construcción del hiperplano de decisión, gracias al planteamiento matemático que manejan[15]; optimizar el margen de decisión. En este sentido, los modelos de SVM permiten la extensión tanto para el escenario de múltiples anotadores como para el problema

de múltiples clases[18]; además, estos clasificadores permiten el manejo de datos no separables linealmente por medio de funciones kernel[19]. Tales funciones permiten llevar los datos del espacio de características de entrada a un espacio de alta dimensión, donde sí sea posible separarlos con un hiperplano lineal.

En el desarrollo de la teoría de SVM, el conjunto de entrenamiento fue introducido al proceso mediante un *kernel* lineal, cuya forma conserva la dimensión de los datos de entrada y fue suficiente para clasificar datos de naturaleza sencilla[3]. No obstante, fue posible implementar un método basado en el uso de funciones *kernel* no lineales que permitieron mapear datos más complejos a un espacio de alta dimensión, donde fue posible encontrar un hiperplano de clasificación adecuado; esta técnica es conocida como el "*kernel tric"*[10]. Es por eso que la estructura de la SVM fue reformulada de tal forma que permita introducir dichas funciones[20].

Usualmente, en tareas de clasificación donde un único anotador provee el vector de etiquetas requerido, se usan matrices kernel de carácter no supervisado, como en el trabajo desarrollado por Jiang[21]. No obstante, en el escenario de múltiples expertos se requirió mejorar la representación de la información para que el clasificador tenga un buen rendimiento, incluso en presencia de posibles etiquetas ruidosas, es por ello que se vio la necesidad de implementar matrices kernel supervisadas[17]. En este trabajo, tomando como base las SVM para la clasificación multi-clase, se exploran diferentes funciones kernel con un enfoque supervisado; en ese mismo orden de ideas, se crearon unas *máscaras* propuestas en este trabajo cuyo propósito es relacionar las etiquetas dadas por cada experto con los valores dados por el kernel.

El método propuesto en esta investigación se logró desarrollar con la herramienta computacional MATLAB®. En este software se implementó primero un clasificador SVM multi-clase con kernel; luego, se crearon las máscaras con las etiquetas de cada experto, para obtener las matrices kernel supervisadas y, finalmente se ingresaron dichas matrices al clasificador mediante la suma ponderada de las mismas.

Justificación

La mayoría de las investigaciones sobre técnicas de aprendizaje supervisado se basan en la suposición de que un único experto puede proporcionar la supervisión requerida[14]. Sin embargo, la opinión de un solo experto puede ser insuficiente para

[18] HSU, Chih-Wei; LIN, Chih-Jen. A comparison of methods for multiclass support vector machines. En: IEEE transactions on Neural Networks, 2002, vol. 13, no 2, p. 415-425.
[19] RAKOTOMAMONJY, Alain. SimpleMKL. En: Journal of Machine Learning Research, 2008, vol. 9, no Nov, p. 2491-2521.
[20] NALEPA, Jakub; KAWULOK, Michal; DUDZIK, Wojciech. Tuning and Evolving Support Vector Machine Models. En: International Conference on Man–Machine Interactions. Springer, Cham, 2017. p. 418-428.
[21] JIANG, Hao. Stationary Mahalanobis kernel SVM for credit risk evaluation. En: Applied Soft Computing, 2018, vol. 71, p. 407-417.

crear una verdad única y la supervisión puede estar sujeta a más de un experto. Esta necesidad se da en la mayoría de casos por las ambigüedades que pueden generar las características de algunos objetos. Un ejemplo es presentado por Murillo en su trabajo de grado[15], donde se explica la dificultad para lograr identificar hipernasalidad y soplos cardíacos mediante bio-señales de voz y fonocardiografía.

Para un experto, coincidir con el criterio de los otros para obtener una verdad única es algo bastante complejo[13]. Lograr que los etiquetadores estén tan bien preparados que todos entreguen el mismo dictamen sobre algún evento requiere de un riguroso e intensivo proceso de enseñanza, con el que ellos lleguen a ser capaces de dar juicios correspondientes con la realidad, lo que implicaría grandes costos económicos. El enfoque de múltiples expertos puede evitar la necesidad de tales esfuerzos, pues pretende identificar las mejores cualidades de cada etiquetador, de forma que se identifique cuáles son aptos para aportar en la construcción de una base de datos sólida y en la clasificación.

La revisión de bibliografía sobre problema de múltiples expertos explicada en más detalle en la sección Escenario de múltiples expertos indicó que se han propuesto diversos enfoques dentro del marco de la clasificación. Sin embargo, son bastantes limitados los estudios que manejan este problema por medio de matrices kernel. Es por ello que, en el presente trabajo se desarrolló una solución basada en kernel supervisado, que adicionalmente permite superar el problema de muestras no separables linealmente.

Objetivo general

Desarrollar un enfoque de clasificación multi-clase, usando funciones kernel y máquinas de vectores de soporte, orientado a resolver problemas de reconocimiento de patrones que involucren múltiples expertos.

Objetivos específicos

- Implementar en programación un clasificador multi-clase basado en máquinas de vectores de soporte que permita evaluar diferentes funciones kernel de forma no sesgada.
- Diseñar una estrategia de clasificación supervisada multi-clase usando funciones kernel supervisadas en un escenario de múltiples expertos.
- Comparar el desempeño de funciones kernel supervisadas, en términos de clasificación, versatilidad y representación, con el fin de seleccionar aquellas que sean las más adecuadas para generar extensiones a múltiples expertos.

Contribuciones de este trabajo

Dentro de los retos de la clasificación supervisada, uno de especial interés es el caso de múltiples anotadores, que puede surgir debido a varias circunstancias; entre ellas, la dificultad en el proceso de etiquetado de algunos objetos debido a las propiedades intrínsecas los mismos, los altos costos que implica obtener la supervisión de etiquetadores con el experiencia y experticia que requieren algunas aplicaciones; la imposibilidad de obtener consenso entre anotadores.

Este trabajo de grado presentó una metodología que logró un buen desempeño de clasificación en medio de múltiples anotadores, teniendo como base un clasificador convencional (máquinas de vectores de soporte). La inclusión de máscaras creadas a partir de las etiquetas de cada anotador combinadas con las matrices kernel, proporcionaron un enfoque novedoso que, se desenvolvió efectivamente con datos no linealmente separables.

Marco teórico

En esta sección se explican varios conceptos básicos sobre el aprendizaje automático, las máquinas de vectores de soporte, funciones kernel, entre otros; muy importantes para la presente investigación.

Aprendizaje automático

Desde hace algunos años, muchas áreas de la ciencia y la ingeniería han venido explorando técnicas de aprendizaje automático para el análisis de grandes cantidades de información. La investigación de algoritmos computacionales continúa trabajando por el desarrollo de métodos más eficientes y menos costosos, que logren extraer información esencial de bases de datos complejas. Un claro ejemplo de esta situación se evidencia en el trabajo de Diller, donde se incorporan redes de aprendizaje automático en modelos de pronóstico de enfermedades cardíacas congénitas, con base en datos recopilados de más de 10.000 pacientes y 44.000 informes médicos[22].

El aprendizaje automático es una rama de la inteligencia artificial que ha tenido cabida en diversos campos, desde detección de agentes contaminantes en

[22] DILLER, Gerhard-Paul. Machine learning algorithms estimating prognosis and guiding therapy in adult congenital heart disease: data from a single tertiary centre including 10 019 patients. En: European heart journal, 2019, vol. 40, no 13, p. 1069-1077.

productos alimenticios[23], hasta optimización de motores de búsqueda online[13]. Su principal objetivo es desarrollar métodos que permitan crear modelos de predicción a partir de información proveniente del entorno; en otras palabras, se busca darles a las máquinas la capacidad de "aprender", simulando el entendimiento necesario para predecir comportamientos futuros[24].

El aprendizaje de máquina conserva una estrecha relación con el reconocimiento de patrones, reflejada en sus campos de acción. En efecto, Bishop afirma en su libro[3] *Pattern recognition and machine Learning:* "El reconocimiento de patrones tiene su origen en la ingeniería, mientras que el aprendizaje automático surgió de la informática. Sin embargo, estas actividades se pueden ver como dos facetas del mismo campo, y juntas han experimentado un desarrollo sustancial en los últimos diez años".

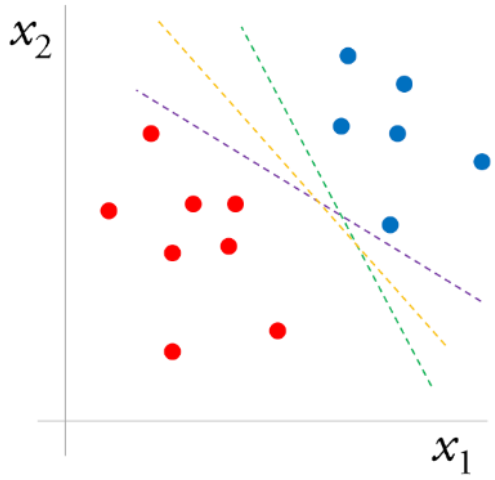
Reconocimiento de patrones

El reconocimiento de patrones es una metodología que desarrolla algoritmos computacionales capaces de extraer conclusiones importantes de grandes bases de datos. Algunas de las tareas de las cuales se ocupa son: clasificación, regresión, agrupamiento y modelado descriptivo[25]. Sin embargo, esta investigación se concentró en la clasificación, una que pretende encontrar reglas de decisión que discriminen datos de diferentes clases y predigan etiquetas de nuevos datos.

Para poder entender mejor el concepto de clasificación, se muestra en la Figura 1 una cantidad de puntos de dos colores diferentes; cada punto sobre el plano representa un objeto o muestra y su color representa a su vez la etiqueta, o sea, la clase a la cual pertenece; la línea punteada expone la regla de decisión que separa correctamente los objetos de las dos diferentes clases. El reto para los algoritmos computacionales es encontrar una regla de decisión óptima, a partir de un conjunto de datos $X \in \mathbb{R}^{nxm}$, donde n es el numero de datos y m es el número de características. Es importante aclarar que los datos del ejemplo se grafican en el plano bidimensional puesto que tienen dos características, es decir, $m = 2$. En otros casos, los datos de un conjunto pueden contener muchas más.

[23] BISGIN, Halil. Comparing SVM and ANN based machine learning methods for species identification of food contaminating beetles. En: Scientific reports, 2018, vol. 8, no 1, p. 6532.
[24] CONWAY, Drew; WHITE, John. Machine learning for hackers. Sebastopol: O'Reilly Media, 2012.
[25] FLACH, Peter. Machine learning: the art and science of algorithms that make sense of data. New York: Cambridge University Press, 2012

Figura 1. Representación de un conjunto de datos en un plano bidimensional.



Fuente: Esta investigación.

Las características son clave en el proceso de definir la etiqueta de algún objeto; por ejemplo, si la muestra es un vehículo, una característica podría ser su tamaño, según el cual el objeto puede ser clasificado como bicicleta o como carro; en otro conjunto de datos puede ser que un elemento sea clasificado como fruta o vegetal, según sus niveles de vitamina C, la cantidad de proteína, y muchas otras propiedades. Aunque estos ejemplos expongan objetos que son bastante familiares, muchos problemas actuales tratan con datos que no son tan fáciles de diferenciar a simple vista, como señales eléctricas, sísmicas, entre otros.

En general, los métodos de clasificación pueden dividirse de acuerdo con la forma de aprendizaje que usan: aprendizaje por refuerzo, donde el algoritmo aprende a base de ensayo y error; aprendizaje supervisado, donde la máquina “aprende con la ayuda de un profesor”, y aprendizaje no supervisado, donde no es necesario ningún “profesor”, como explica el libro de Haykin[2]. Los algoritmos de aprendizaje supervisado extraen verdades de muestras observadas, o sea, cuyas etiquetas son conocidas[26].

Clasificación supervisada

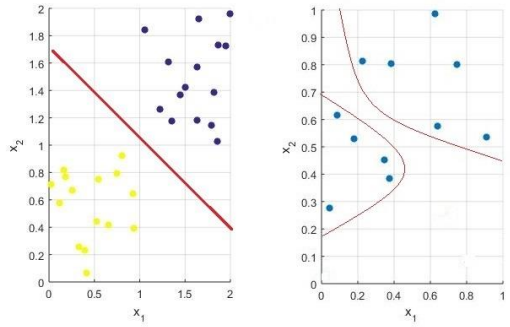
En este tipo de clasificación el algoritmo se entrena bajo la supervisión de un “profesor"; es decir, dado un conjunto de datos, en él se incluye un vector de etiquetas dadas con anterioridad por un experto. Por el contrario, en la clasificación no supervisada no se cuenta con ningún “profesor”; o sea, el conjunto de

[26] BZDOK, Danilo; KRZYWINSKI, Martin; ALTMAN, Naomi. Points of significance: Machine learning: supervised methods. 2018.

entrenamiento no cuenta con un vector de etiquetas, por lo que el algoritmo crea grupos de la mejor manera posible, de acuerdo a la similitud en las características de los datos.

Para comprender la idea de clasificación supervisada, se presenta en la Figura 2 un ejemplo de entrenamiento supervisado (izquierda), y no supervisado (derecha). Como se aprecia en la imagen, la clasificación supervisada es aquella donde se conoce la clase a la cual pertenece cada muestra, es decir, la etiqueta de cada dato (representada por los diferentes colores en los puntos); mientras que, en el enfoque no supervisado, no se cuenta con la marca de clase y por tanto no se crean reglas de decisión, sino que se forman grupos.

Figura 2. Representación de dos tipos de aprendizaje según el conjunto de entrenamiento. a) En el aprendizaje supervisado se usa una base de datos previamente etiquetada para entrenar el clasificador. b) En el aprendizaje no supervisado no se conocen etiquetas del conjunto.



(a) (b)

Fuente: Esta investigación.

Las técnicas existentes dentro del aprendizaje automático, específicamente dentro de la clasificación supervisada, varían bastante en su estructura; algunas se basan en funciones de densidad de probabilidad[27], otras se inspiran en algunos procesos de la naturaleza –bioinspirados- (por ejemplo, la habilidad cognitiva del cerebro humano) [28] y muchas otras técnicas. Uno de los métodos que es objeto de especial interés en este trabajo, es el clasificador basado en vectores de soporte.

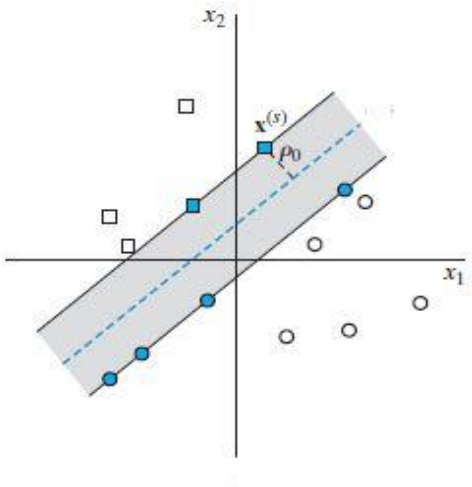
27 JI, Haijin, et al. Empirical Studies of a Kernel Density Estimation Based Naive Bayes Method for Software Defect Prediction. En: IEICE TRANSACTIONS on Information and Systems, 2019, vol. 102, no 1, p. 75-84.

28 ABID, Faroudja; HAMAMI, Latifa. A survey of neural network based automated systems for human chromosome classification. En: Artificial Intelligence Review, 2018, vol. 49, no 1, p. 41-56.

Máquinas de vectores de soporte

Las máquinas de vectores de soporte (SVM por sus siglas en inglés) son un tipo de clasificador supervisado que según Haykin[2], es quizás el más elegante de todos los métodos de aprendizaje *kerne*l. Entre muchas de las ventajas que presentan se encuentran: su gran eficiencia, gran robustez en la extensión al caso multi-clase, poco uso de técnicas heurísticas, solución única, etc. Estas máquinas logran construir, a partir de un conjunto de entrenamiento, una regla de decisión capaz de predecir la clase de nuevos datos. Para entender cómo funciona, se presentan en la Figura 3 dos conjuntos de diferentes clases, la clase A representada por cuadros y la clase B representada por círculos. El área sombreada resalta los elementos que se encuentran más cercanos al hiperplano de decisión (línea punteada), estos se conocen como *vectores de soporte* y la distancia desde éstos hasta el hiperplano se define como *margen de separación*. La idea central del clasificador es encontrar un hiperplano o regla de decisión que logre clasificar nuevos datos correctamente, maximizando dicho margen.

Figura 3. Hiperplano de decisión entre dos conjuntos de diferente clase.



Fuente: Esta investigación

Sea $\{(x_i, d_i)\}_{i=1}^{n}$ un conjunto de entrenamiento, donde n es el número de elementos, x_i es una muestra y d_i es su respectiva etiqueta (con un valor de +1 para los datos de la clase A y -1 para los datos de la clase B). La ecuación del hiperplano deseado se define como:

$$w^T x + b = 0, \tag{1}$$

donde w es un vector de pesos y b es el sesgo. Encontrar estos parámetros es el objetivo principal del clasificador. Note que los datos de la clase $d_i = +1$ cumplen que $w^T x + b \geq 0$ pues se ubican de un lado del hiperplano, y los datos de la clase $d_i = -1$ cumplen que $w^T x + b < 0$ puesto que están del otro lado, como explica Haykin[2].

En la Figura 3 se pueden observar los vectores de soporte resaltados en azul. Una de las características principales de SVM es que en dichos puntos se establece que:

$$w_*^T x + b_* = 1 \text{ para } d = +1, \tag{2}$$

$$w_*^T x + b_* = -1 \text{ para } d = -1, \tag{3}$$

donde w_* y b_* son los valores óptimos. Por lo tanto, todos los datos cumplen con:

$$d_i(w^T x_i + b) \geq 1, \tag{4}$$

esta condición presentada en una forma tan compacta, facilita el planteamiento del problema, según Bishop[3]. Entonces, los pesos w y el sesgo b deben hacer que el hiperplano separe perfectamente los datos de las diferentes clases; pero, también deben lograr maximizar el margen de separación. Por medio de un análisis geométrico, se puede definir la distancia entre el hiperplano y los vectores de soporte, ρ_o, como:

$$\rho_o = \frac{1}{\| w \|}. \tag{5}$$

Note que ρ_o es la mitad del margen de separación. Consecuentemente, minimizar $\|w\|$ equivale a maximizar ρ_o y por ende el margen. Esto conduce a un problema de optimización cuadrática cuyas restricciones están dadas por la ecuación (4) y (5). De esta forma el problema se plantea así:

$$\min_w \ f(w) = \frac{1}{2} w^T w \ ; \ s.a. \ d_i(w^T x + b) \geq 1, \tag{6}$$

donde los valores a encontrar son los w_* óptimos. Es importante notar que la función objetivo $f(w)$ es una función convexa y las restricciones son lineales, donde existirán tantos pesos w como características tengan los datos el conjunto de entrenamiento. Además, x_i no es una variable de optimización, puesto que todos los datos de entrenamiento son conocidos. La ecuación en (6) se conoce como problema primal.

Este problema puede ser resuelto por medio de los multiplicadores de Lagrange y las condiciones de Karush-Kuhn-Tucker (KKT). Consecuentemente, se inicia formando el lagrangiano así:

$$L(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i [d_i (w^T x_i + b) - 1], \tag{7}$$

donde $\{\alpha_i\}_{i=1}^{n}$ son los multiplicadores de Lagrange. Luego, verificando las condiciones KKT se obtiene:

$$\alpha_i > 0 \, , \tag{8}$$

$$w = \sum_{i=1}^{N} \alpha_i d_i x_i, \tag{9}$$

$$\sum_{i=1}^{N} \alpha_i d_i = 0, \tag{10}$$

$$\alpha_i [d_i (w_i^T x_i + b) \; - 1] = 0. \tag{11}$$

En este punto, aunque es posible encontrar todos los parámetros con diferentes técnicas de optimización, es conveniente llevar el problema a la formulación dual, puesto que se facilitan considerablemente los cálculos, como explica Smith[29], y los datos entran a la función objetivo como un producto interno (circunstancia que será de mucha ayuda en la introducción de funciones kernel). La función del problema dual, definida como $Q(\alpha)$ se logra remplazando las ecuaciones (9) y (10) en (7), dando como resultado:

$$\max_{\alpha} \; Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \, \alpha_j d_i d_j \, \boldsymbol{x_i x_j} \, ; \tag{12}$$

$$s.a \sum_{i=1}^{N} \alpha_i d_i = 0, \; \alpha_i > 0.$$

Es importante resaltar que esta función objetivo depende solo de los multiplicadores $\{\alpha_i\}$, como explica Boser[10]. Encontrando estos parámetros se puede calcular el resto con las ecuaciones (9) y (11). En la práctica, es conveniente expresar el problema de (12) en forma matricial para facilitar los procesos computacionales, aprovechando el hecho de que se trata de un problema de programación cuadrática.

[29] SMITH, Baxter. Lagrange multipliers tutorial in the context of support vector machines. En: Memorial University of Newfoundland St. John's, Newfoundland, Canada, 2004, p. 17.

La predicción de la etiqueta d_k de un nuevo dato x_k se realiza mediante la función signo, así:

$$d = sign(w^T x_k + b). \tag{13}$$

Margen suave

El método usado por SVM es muy preciso para datos separables linealmente. No obstante, datos de esa forma son bastante inusuales. Por lo general, los datos de una clase se traslapan con los de la otra en el espacio de características. Esto dificulta bastante el entrenamiento del clasificador; de hecho, el algoritmo no podría converger, ya que no existen parámetros que hagan que el hiperplano logre cumplir con cada una de las restricciones. Es por ello que en la práctica se agregan unas variables de escasez ξ_i a la función objetivo en (6), así:

$$\min_{w,\xi} \ f(w) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{N}\xi_i \tag{14}$$

donde C es un parámetro de compensación, definido previamente según los requerimientos. El procedimiento para llegar a la formulación dual es similar al que se hizo anteriormente, de donde resulta:

$$\max_{\alpha} \ Q(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\,\alpha_j d_i d_j\,\boldsymbol{x_i x_j}\,; \tag{15}$$

$$s.a\ \sum_{i=1}^{N}\alpha_i d_i = 0,\ 0 \le \alpha_i \le C.$$

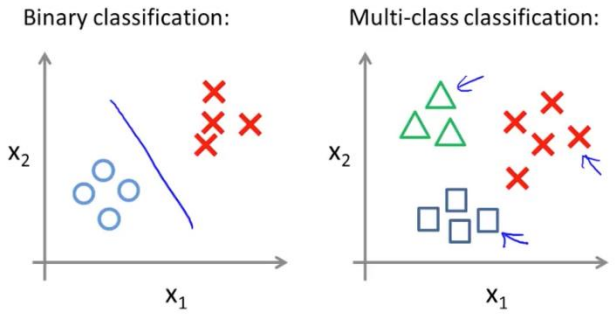
El problema dual no cambia considerablemente. De este modo se logra tratar con el problema de datos traslapados. Este proceso se conoce como SVM de margen suave.

Caso multi-clase

Otra situación que se presenta con mucha frecuencia en el reconocimiento de patrones es el caso multi-clase. Este concepto hace referencia al número de clases en las que puede ser clasificado un conjunto de elementos. Por ejemplo, aplicaciones sencillas pueden requerir que un correo sea clasificado como correo

importante y permanezca en la bandeja de entrada, o como correo no deseado y se envíe a la carpeta de spam; esta sería una tarea bi-clase, ya que solo hay dos opciones. No obstante, muchas otras aplicaciones requieren clasificación entre varias clases. Por ejemplo, un algoritmo de reconocimiento de manuscritos debería poder reconocer diferentes letras, llegando a necesitar más clases[30]. La Figura 4 muestra un ejemplo simple de cómo se vería un conjunto con dos clases y otro con tres.

Figura 4. Representación sencilla de la diferencia entre la clasificación bi-clase y multi-clase.



Fuente: https://danielmoralesp.gitbooks.io/inteligencia-artificial/content/2-machine-learning/34-coursera-andrew-ng-machine-learning/343-week-3/3433-clasificacion-multiclase.html

Debido a que la SVM fue diseñada originalmente para clasificar datos únicamente en dos clases, se han desarrollado diferentes métodos para extenderla a varias clases; por ejemplo, métodos basados en arboles de decisión, métodos que usan la combinación de diferentes clasificadores binarios *como* el *one against one* y el *one against all*, entre otros[18]. El método uno contra todos (*one against all*) se encuentra entre los más usados y es muy sencillo de implementar como explica Liu en su trabajo sobre el clasificador SVM multi-clase[31]. Es por ello que este método fue escogido para el desarrollo de este trabajo y su funcionamiento se explica a continuación.

Método uno contra todos (*one against all*)

Dado un conjunto de entrenamiento que cuenta con p número de clases, este método crea p clasificadores bi-clase; en la Figura 5 se muestra un ejemplo para

[30] BAHLMANN, Claus; HAASDONK, Bernard; BURKHARDT, Hans. Online handwriting recognition with support vector machines-a kernel approach. En: Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition. IEEE, 2002. p. 49-54.
[31] LIU, Yi; ZHENG, Yuan F. One-against-all multi-class SVM classification using reliability measures. En: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005. IEEE, 2005. p. 849-854.

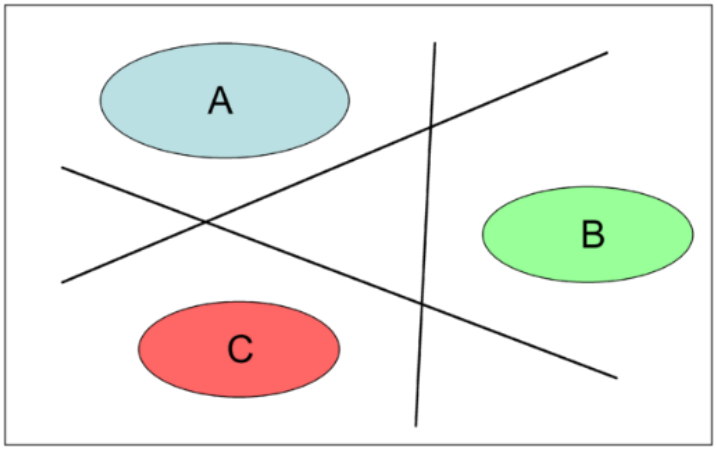
$p = 3$. Cada uno de los p hiperplanos se crea tomando el conjunto de entrenamiento completo. Para el primer clasificador se toman los datos de una sola clase como positivos y los datos de todas las clases restantes como negativos[31]. El siguiente clasificador toma los elementos de otra clase como positivos y los elementos de las clases restantes como negativos. Este procedimiento se repite con cada clase, hasta encontrar todas las p reglas decisión.

Una vez construidas estas reglas de decisión, para clasificar un nuevo dato x_k, se usa el siguiente criterio:

$$d_k \equiv \arg\max_i \left((\boldsymbol{w}_*^{\boldsymbol{i}})^T x_k + \boldsymbol{b}_*^{\boldsymbol{i}}\right), \tag{16}$$

donde w_i y b_i, $i = \{1,2,3,\dots p\}$, hacen referencia a cada regla de decisión obtenida por cada clasificador bi-clase[18].

Figura 5. Reglas de decisión creadas en el método uno contra todos.



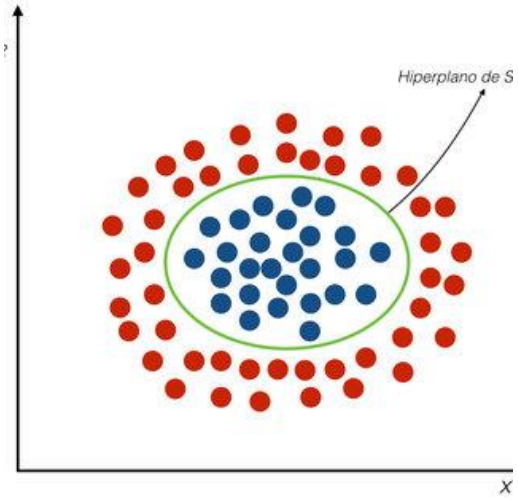
Fuente: http://courses.media.mit.edu/2006fall/mas622j/Projects/aisen-project/

Funciones kernel

Hasta el momento se han considerado datos linealmente separables (con cierto traslape), donde una línea recta es suficiente para separar los datos de las diferentes clases. No obstante, existen casos donde una línea no es capaz de realizar esta tarea, puesto que los datos se ubican de una forma más compleja en el espacio de características. Un ejemplo de esta situación se presenta en la Figura 6, donde fácilmente se puede observar que una regla de decisión lineal sería totalmente inútil para la clasificación. Debido a ello, se introducen a continuación las funciones kernel para abordar este problema.

El clasificador SVM permite introducir matrices kernel que transforman los datos de entrada de un espacio de características de dimensión m_o a un espacio de dimensión infinita, donde es posible encontrar un hiperplano de decisión adecuado. Este proceso se conoce como el *kernel trick,* como se menciona en el trabajo de Rakotomamonji[19].

Figura 6. Ejemplo de un conjunto de entrenamiento no separable linealmente.



Fuente: https://www.researchgate.net/figure/Figura-4-Datos-linealmente-separables_fig2_283345149

Estas matrices son usadas para transformar el conjunto de datos $\{\boldsymbol{x}_n\}$ de una dimensión m a un espacio de más alta dimensión v, como se explica el libro de Haykin[2]. Entonces, sea $\varphi_j(\boldsymbol{x})$: $\mathbb{R}^m \rightarrow \mathbb{R}^v$ una función no lineal, cada dato tiene su imagen en el nuevo espacio, así:

$$\varphi_j(\boldsymbol{x}) = \boldsymbol{x_j}. \tag{17}$$

Es aquí donde el producto interno entre $\boldsymbol{x_i x_j}$ de la ecuación (15) es de mucha utilidad, ya que permite cambiar este producto por el producto de las funciones $\varphi_i(\boldsymbol{x})\varphi_j(\boldsymbol{x})$. Introduciendo este término a la ecuación (15) se obtiene:

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\, \alpha_j d_i d_j\, \varphi_i(\boldsymbol{x})\varphi_j(\boldsymbol{x}). \tag{18}$$

Si finalmente se define $\mathrm{k}(\boldsymbol{x_i}, \boldsymbol{x_j})$: $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$, tal que $\mathrm{k}(\boldsymbol{x_i}, \boldsymbol{x_j}) = \varphi_i(\boldsymbol{x})\varphi_j(\boldsymbol{x})$, entonces la función objetivo resultaría en:

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\, \alpha_j d_i d_j \mathrm{k}(\boldsymbol{x_i}, \boldsymbol{x_j})\,. \tag{19}$$

Como se mencionó anteriormente, el problema es más fácil de resolver con notación matricial, computacionalmente hablando. Es por ello que se requiere crear una matriz kernel K^{nxn} , así:

$$\boldsymbol{K} = \begin{pmatrix} \mathrm{k}(\boldsymbol{x_1}, \boldsymbol{x_1}) & \dots & \mathrm{k}(\boldsymbol{x_n}, \boldsymbol{x_1}) \\ \vdots & \ddots & \vdots \\ \mathrm{k}(\boldsymbol{x_1}, \boldsymbol{x_n}) & \dots & \mathrm{k}(\boldsymbol{x_n}, \boldsymbol{x_n}) \end{pmatrix} \tag{20}$$

donde n es el número de elementos del conjunto de entrenamiento. Note que esta matriz es simétrica ya que $\mathrm{k}(\boldsymbol{x_i}, \boldsymbol{x_j}) = \mathrm{k}(\boldsymbol{x_j}, \boldsymbol{x_i})$. Además, ya que cada función mide la similitud entre dos datos, la diagonal principal estará conformada solo por unos.

Características de las matrices kernel

Para formar una matriz kernel no es necesario calcular explícitamente $\varphi_j(\boldsymbol{x})$ en ningún punto; simplemente se necesita encontrar cada elemento que $\mathrm{k}(\boldsymbol{x_i}, \boldsymbol{x_j})$ mediante las funciones kernel. Entre las más comunes se encuentran:

- Lineal $\mathrm{k}(\boldsymbol{x_i}, \boldsymbol{x_j}) = \boldsymbol{x_i^T x_j}$,
- Polinomial $\mathrm{k}(\boldsymbol{x_i}, \boldsymbol{x_j}) = (\boldsymbol{x_i^T x_j} + 1)^e$,
- Exponencial $\mathrm{k}(\boldsymbol{x_i}, \boldsymbol{x_j}) = exp(-\frac{1}{2\sigma^2}\|\boldsymbol{x_i} - \boldsymbol{x_j}\|^2)$,

donde la potencia p y el ancho σ^2 son constantes definidas dependiendo de la naturaleza de los datos. Adicionalmente, según el Teorema de Mercer para que una matriz kernel sea válida debe cumplir con las siguientes condiciones :

- ✓ La matriz debe ser semi-definida positiva.
- ✓ La función kernel usada debe expresar una medida de similitud entre los datos.

Muchos tipos de matrices pueden ser creadas con base en estas dos condiciones y aunque la medida de similitud usada con más frecuencia es la distancia euclidiana, ésta no es una limitación, como lo demuestra el trabajo de Jiang[21].

Kernel trick

Emplear funciones kernel supone dos valiosas ventajas. Por un lado, aunque se asume que el espacio de características al introducir el kernel es de dimensión infinita, se calculan una cantidad finita de $\{\alpha_i\}$ gracias a que estos dependen

únicamente del número de datos de entrenamiento, mas no del número de características[3]. Por otro lado, aunque la cantidad de los pesos $\boldsymbol{w}$ sería infinita, en ningún momento de la clasificación es necesario calcularlos explícitamente. Por estas razones, las máquinas de vectores de soporte han sido referidas también como máquinas kernel[32].

Escenario de múltiples expertos

Normalmente, un modelo de clasificación se construye a partir de un conjunto de entrenamiento, donde se asume que toda la información contenida en él es correcta. No obstante, esto puede no ser totalmente cierto, puesto que las propiedades intrínsecas de algunos objetos pueden afectar las bases de datos, y la tarea de conformar un conjunto fiable debe ser otorgada a varios anotadores (un problema también conocido como aprendiendo de multitudes, múltiples expertos, etiquetadores, fuentes, profesores, etc)[33] , quienes bajo su propio criterio crean un vector de etiquetas requerido para el conjunto de entrenamiento. El problema de aprender de estas múltiples fuentes es poder determinar cuáles de ellas están dando un juicio correcto acerca de un objeto y cuáles no[34].

A continuación, se presenta un breve resumen de la principal bibliografía encontrada entorno al problema de "aprender de múltiples expertos" y de qué métodos se han propuesto para solucionarlo:

- En el trabajo de Sheng se explica que construir una base de datos de calidad implica costos considerablemente altos[12]; pero, prescindir de etiquetadores expertos para disminuir dichos costos resulta en etiquetas ruidosas provenientes de etiquetadores sin experiencia. Por tanto, se propone mejorar la calidad de estas etiquetas ruidosas mediante etiquetado repetido (*repeated labeling),* con el fin de mejorar a su vez, la precisión en el aprendizaje supervisado. Este trabajo tiene un carácter de preprocesamiento.

- En el trabajo de Dekel se propone un algoritmo supervisado construido con base en SVM, cuyo propósito es superar el problema de etiquetas proporcionadas por un grupo de etiquetadores[13]; teniendo en cuenta que, no se tiene ninguna información previa sobre ninguno de ellos. Para tal propósito mideron la influencia de cada maestro por el efecto acumulativo de los ejemplos que controla, gracias a la característica de SVM de identificar cuáles muestras son vectores de soporte y

[32] PLÖTZ, Thomas; FINK, Gernot A. Markov models for offline handwriting recognition: a survey. En: International Journal on Document Analysis and Recognition (IJDAR), 2009, vol. 12, no 4, p. 269.
[33] WANG, Xin; BI, Jinbo. Bi-convex optimization to learn classifiers from multiple biomedical annotations. En: IEEE/ACM transactions on computational biology and bioinformatics, 2016, vol. 14, no 3, p. 564-575.
[34] TORKKOLA, Kari; TUV, Eugene. Ensemble learning with supervised kernels. En: European Conference on Machine Learning. Springer, Berlin, Heidelberg, 2005. p. 400-411

cuáles no. Esto significa que cada experto debería contribuir aproximadamente el mismo número de vectores de soporte. De modo que, el algoritmo usa esta observación como una restricción. Tal análisis, muestra que es probable que esta restricción afecte solo a maestros maliciosos y de baja calidad.

- En el trabajo de Yan se explica que obtener datos para el aprendizaje puede ser más fácil y menos costoso que obtener sus respectivas etiquetas[14]. El aprendizaje activo (*active Learning*) propone identificar de manera inteligente cuales muestras no son muy útiles, de manera que se eliminen del conjunto y así disminuir el costo de etiquetado. El trabajo desarrollado por Yan se propuso dirigir este concepto al caso de múltiples etiquetadores, para identificar los que son más aptos para dar etiquetas, basándose en un modelo probabilístico

- En el trabajo de grado de Murillo se desarrolla un método basado en máquinas de vectores de soporte multi-clase que agrega un factor de penalización en la función objetivo, que logra estimar el desempeño de los etiquetadores[15]. De esta forma, se pretendió disminuir la función objetivo al incluir etiquetas de buenos evaluadores y aumente en el caso contrario.

- En el trabajo realizado por Zhang[35] se presenta un estudio detallado de las técnicas desarrolladas en los últimos años para el problema de "aprender de múltiples fuentes" (*learning from crowdsourced data*) incluido el aprendizaje activo, el aprendizaje de transferencia emergente y métodos de inferencia de la verdad en el *crowdsourcing*. Además, se revisaron las definiciones de etiquetado colaborativo, calidad de la etiqueta y calidad del modelo de aprendizaje.

- En el trabajo de Guan se propuso modelar cada experto individualmente con una red neuronal compartida que tiene conjuntos separados de salidas para cada experto[36]; al tiempo que se aprenden pesos promedio para combinar sus predicciones modeladas.

- En el trabajo de Gil se presenta un enfoque un enfoque basado en *kernel annotator relevance análisis[17]* (KAAR). Con este kernel se calculó la relevancia de cada anotador como una comparación promedio entre las características de entrada y las etiquetas de los expertos. A su vez, se predijo una nueva etiqueta de muestra como una combinación convexa de clasificadores que adoptan la codificación basada en el KAAR desarrollado.

[35] ZHANG, Jing; WU, Xindong; SHENG, Victor S. Learning from crowdsourced labeled data: a survey. En: Artificial Intelligence Review, 2016, vol. 46, no 4, p. 543-576.

[36] GUAN, Melody Y., et al. Who said what: Modeling individual labelers improves classification. En: Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

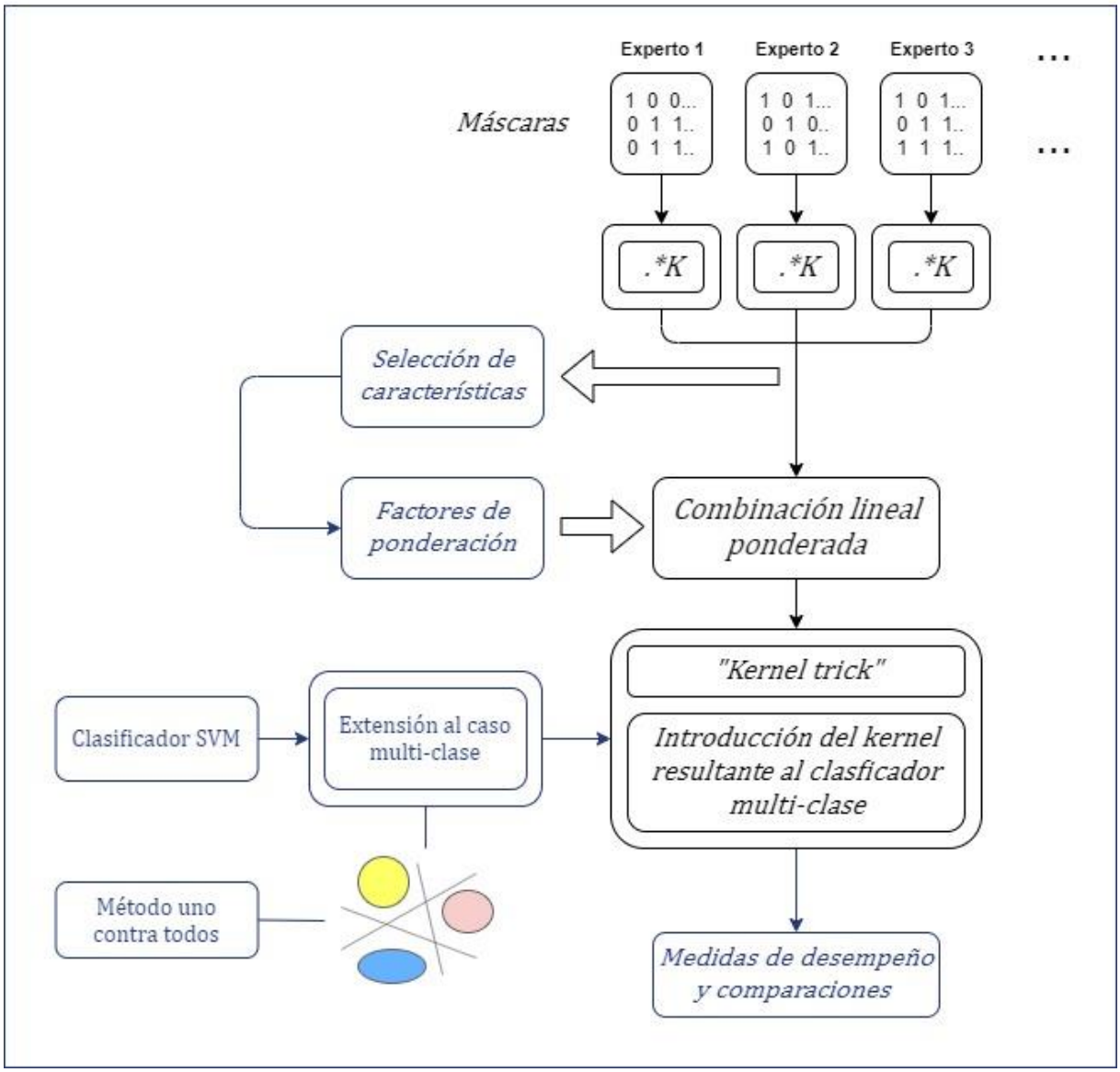
Comentarios sobre trabajos anteriores

A pesar de que el escenario de múltiples anotadores ha sido abordado desde diversos enfoques, son bastante escasos los trabajos realizados mediante matrices kernel. El más cercano puede ser el de Gil[17]. Aunque su trabajo presenta ciertos conceptos semejantes a los usados en este trabajo, el esquema de clasificación desarrollado presenta grandes diferencias. La presente investigación propone un nuevo enfoque en el que se construye una matriz a partir de las etiquetas de cada experto y la matriz kernel de los datos de entrenamiento; posteriormente, estas matrices entran al clasificador SVM por medio de la suma ponderada de las mismas, cuyos factores de ponderación se obtuvieron de un análisis de relevancia de variables al que se someten los expertos. La ventaja de este método es que adicionalmente, es posible tratar con datos no linealmente separables, gracias a las funciones kernel exploradas.

1. DESARROLLO DE LA INVESTIGACIÓN

En esta sección se explica el método basado en matrices kernel que se desarrolló en este trabajo para abordar el problema de clasificar a partir de múltiples expertos. En la Figura 7 se presenta el diagrama del proceso seguido.

Figura 7. Diagrama de la metodología desarrollada en este trabajo.



Fuente: Esta investigación.

1.1. CLASIFICACIÓN MULTI-CLASE CON KERNEL

Inicialmente, se implementó el clasificador SVM multi-clase con el kernel trick basado en el método uno contra todos, considerando un solo experto. Para ello, se tiene el vector de etiquetas $\boldsymbol{d_v} \in \mathbb{R}^{nx1}$ con $d_v = \{1,2,3 \dots p\}$, donde p es el número de clases y n el número de datos. Cabe recordar que este método se basó en clasificadores bi-clase donde el conjunto de etiquetas debe contener simplemente 1 para los datos de una clase y -1 para los de la clase restante. Por ello se hizo necesario transformar el vector $\boldsymbol{d_v}$ a una matriz $\boldsymbol{d} \in \mathbb{R}^{nxp}$, de modo que cada clasificador use la columna de la matriz correspondiente. Por ejemplo, suponiendo que el experto entrega este vector de etiquetas d_v:

$$d_v = \begin{pmatrix} 1 \\ 2 \\ 2 \\ 3 \\ 1 \end{pmatrix}, \tag{21}$$

debido a que el vector contiene tres etiquetas diferentes, la matriz $\boldsymbol{d}$ tendrá tres columnas. En la primera columna de la matriz se asignó un 1 a los datos de la clase 1 de d_v y un -1 a los datos de las clases 2 y 3. Igualmente, en la segunda columna se asignó un valor de 1 a los datos de la clase 2 y un -1 a los datos de la clase 1 y 3. Este proceso se realizó para todas las clases existentes y la matriz resultante es:

$$d_v = \begin{pmatrix} 1 \\ 2 \\ 2 \\ 3 \\ 1 \end{pmatrix} \Rightarrow d = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ 1 & -1 & -1 \end{pmatrix}.$$

Luego, para entrenar los clasificadores bi-clase, cada uno de ellos usa la columna de la matriz que corresponda; o sea, el primer clasificador que separa los datos de la clase 1 de todos los demás datos, se creó a partir de la primera columna; el segundo clasificador con la segunda columna y así sucesivamente. De esta manera, se creó un número de reglas de decisión igual al número de clases que tengían los datos.

Ahora, el problema de optimización a resolver para un clasificador bi-clase es:

$$\max_{\alpha} \; Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \, \alpha_j d_i d_j \; \mathbf{k}(x_i, x_j). \tag{22}$$

Convenientemente, se expresó esta función objetivo en forma matricial, puesto que facilita la implementación del algoritmo computacional. Para ello se usó una matriz $\boldsymbol{H} \in \mathbb{R}^{nxn}$ conformada por la matriz kernel y el vector de etiquetas, $\boldsymbol{H} = \boldsymbol{d^T K d}$; además de un vector columna $\alpha \in \mathbb{R}^{nx1}$ conteniendo los multiplicadores de Lagrange. De esta forma se obtuvo:

$$\max_{\alpha} \mathbf{1}\alpha + \frac{1}{2}\alpha^T \boldsymbol{H}\alpha. \tag{23}$$

Note que la matriz $\boldsymbol{H}$ debe ser semi-definida positiva para que el problema tenga una solución única.

1.1.1. Predicción de nuevas etiquetas

El problema en (23) se resolvió para cada uno de los p clasificadores bi-clase. De modo que se obtuvieron p vectores $\boldsymbol{\alpha}$, uno por cada clasificador. Una vez resueltos todos, la predicción de nuevas etiquetas para $\{x_k\}$ se realizó con el criterio de (16), pero con la modificación del kernel, así:

$$d \equiv \arg\max_{j} \left(\sum_{i=1}^{N} \alpha_i^j d_i^j \boldsymbol{K}(x_i, x_k) + b^j \right), \tag{24}$$

para $j = \{1,2,3, \dots, p\}$. De esta forma la clasificación multi-clase con kernel quedó lista para la siguiente etapa.

1.2. CASO DE MÚLTIPLES EXPERTOS

En esta subsección se muestra el método desarrollado para el manejo de múltiples expertos. Para comenzar, se asumió un conjunto de datos de entrenamiento $\{x_i\}_{i=1}^{n}$ y una matriz $\boldsymbol{D} \in \mathbb{R}^{nxL}$ que contiene las etiquetas entregadas por L expertos. Luego, se creó la matriz kernel $\boldsymbol{K} \in \mathbb{R}^{nxn}$ a partir de los datos de entrenamiento $\{x_n\}$. El elemento k_{ij} representa la medida de similitud entre los datos $\boldsymbol{x_i}$ y $\boldsymbol{x_j}$. Luego, se crearon L matrices denominadas en este trabajo como *máscaras*; cada una de ellas codificó el vector de etiquetas de cada experto en una matriz que servirá luego para relacionar las etiquetas con la matriz kernel.

1.2.1. Máscaras

Las máscaras $M_l \in \mathbb{R}^{nxn}$ son matrices que fueron creadas a partir de las etiquetas de los expertos y están compuestas por unos y ceros. Para cada elemento de la máscara m_{ij} el valor de 1 indica que el dato x_i fue etiquetado con la misma clase que el dato x_j y el valor de 0 indica que los datos x_i y x_j fueron etiquetados en

diferentes clases. Por ejemplo, suponiendo que $D_l = [1, 2, 2, 3, 1, 3]'$ es el vector de etiquetas entregado por el experto l, su respectiva máscara será:

$$D_l = \begin{pmatrix} 1 \\ 2 \\ 2 \\ 3 \\ 1 \\ 3 \end{pmatrix} \Rightarrow M_l = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \quad (25)$$

Entonces, para integrar la matriz kernel K con las máscaras M_l se propuso la siguiente estrategia:

- Crear una matriz $\boldsymbol{E_l}$ para cada experto que resulte del producto Hadamard $\boldsymbol{E_l = M_l \circ K}$. De esta forma, si un elemento de la máscara tiene el valor de 1, es decir, el experto afirmó que dos datos pertenecen a la misma clase, entonces se le asigna la medida de similitud de la matriz kernel. Si por el contrario un elemento de la máscara tiene el valor 0, entonces el elemento resultante también será 0. Este método equivale a poner en duda la posición del experto al afirmar que dos datos pertenecen a una misma clase.

1.3. CRITERIO DE CONFIABILIDAD

Hasta el momento, se ha implementado el clasificador SVM multi-clase con funciones kernel y se han creado las matrices correspondientes a cada uno de los expertos; además, se ha logrado relacionar las etiquetas dadas por los expertos con la medida de semejanza entre los datos dada por el kernel, mediante las matrices $\boldsymbol{E_l}$; sin embargo, aún no se abordado la forma en la que estas matrices afectan al clasificador. Cabe recordar que, el objetivo principal es encontrar un modelo de clasificación multi-clase en un escenario de múltiples expertos, en el cual no se conoce la *verdad* sobre los datos, o sea, no se cuenta con un vector de etiquetas único y verdadero sino con una matriz de etiquetas.

Para manejar este problema se propuso el siguiente criterio:

- Cada uno de los etiquetadores tiene cierto grado de confiabilidad que será expresada por medio de unos parámetros $\eta \in \mathbb{R}^{Lx1}$, con los cuales se va a controlar la influencia de cada experto dentro del proceso de entrenamiento del clasificador.

Los parámetros en η son clave para la siguiente etapa del método desarrollado.

1.3.1. Combinación lineal ponderada de las máscaras

En este punto se logró integrar el problema del clasificador multi-clase con kernel y el caso de múltiples etiquetadores. Entonces, observe que en la ecuación (23) se usó una matriz kernel para encontrar el vector $\boldsymbol{\alpha}$ de cada clasificador bi-clase. Esta matriz kernel es única para el caso de un solo etiquetador. No obstante, en este caso habrá tantas matrices como etiquetadores; la introducción de todas estas matrices a la ecuación se logra mediante la combinación lineal de las máscaras con los factores de ponderación $\boldsymbol{\eta_l}$, para la primera y segunda opción de máscara respectivamente:

$$\boldsymbol{K}(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sum_{l=1}^{L} \boldsymbol{\eta_l}\, \boldsymbol{E_l}, \tag{26}$$

De este modo, la matriz kernel resultante se ingresó a la función objetivo de cada clasificador bi-clase.

Ahora, es importante notar que en las ecuaciones (23) y (24) el vector de etiquetas que se requiere debería ser único. Obviamente esto supone un problema ya que en el caso de múltiples expertos se dispone de una matriz (formada por los vectores de etiquetas de cada experto) en lugar de un vector. Este problema se manejó aplicando el criterio del voto mayoritario a la matriz D para obtener un vector de etiquetas de referencia $\boldsymbol{dref} \in \mathbb{R}^{nx1}$. En este vector, la etiqueta de determinada muestra es aquella que más se presenta entre todos los etiquetadores; por ejemplo, si un elemento es etiquetado por seis expertos y tres de los ellos afirman que pertenece a la clase A, dos de ellos dicen que pertenece a la clase B y solo uno dice que pertenece a la clase C, entonces, en el vector de referencia se dirá que esa muestra pertenece a la clase A.

1.3.2. Cálculo de los factores de ponderación

Como se explicó anteriormente, los parámetros $\{\boldsymbol{\eta_l}\}$ indican el grado de confiabilidad de cada experto y se usaron para controlar la influencia de cada uno de ellos en la creación del modelo de predicción. Para calcular estos factores se propuso darle al problema un enfoque de selección de características y análisis de relevancia de variables, ya que estas técnicas son capaces de asignarle un porcentaje de importancia a las variables de una matriz.

La selección de características es un conjunto de técnicas dentro del aprendizaje automático que, dado un conjunto de muestras, pretenden identificar información redundante e irrelevante para caracterizar las muestras de la mejor manera

posible[37]. Aunque el objetivo principal de estas técnicas es mejorar la eficiencia del clasificador, fue posible hacer una analogía que permita aplicar el mismo concepto al cálculo de los factores de ponderación, ya que cada experto jugaría el papel de una característica. Siguiendo esta línea de ideas, se creó una matriz $\boldsymbol{V} \in \mathbb{R}^{n^2 xL}$, donde cada columna $\boldsymbol{V_l}$ contiene la vectorización de la matriz $\boldsymbol{E_l}$, expuesta en la sección 1.2.1. Esto se hizo con el fin de que, al aplicar la selección de características se obtenga un porcentaje de importancia para cada columna de la matriz, y a su vez para cada experto.

Las técnicas de selección de características pueden ser de dos tipos:

- $Wrapper$: en este caso se usa un algoritmo de aprendizaje para evaluar las características, de modo que las características que mejores resultados de clasificación tengan serán las de más porcentaje de importancia; cabe resaltar que para tal tarea se usan las etiquetas de las muestras. En consecuencia, usar estas técnicas sobre la matriz $\boldsymbol{V}$ donde cada columna representa un experto, no tendría sentido puesto que el algoritmo asumiría que cada fila representa una muestra y tiene una respectiva etiqueta, lo cual no es cierto para $\boldsymbol{V}$. Por tanto, un método de este tipo no es adecuado para encontrar los factores de ponderación[38].
- $Filter$: estos métodos analizan las propiedades intrínsecas de los datos, y aunque muchos de estos métodos utilizan las etiquetas del conjunto, existen otros tantos que no. Es por esto que se consideró para este propósito usar una técnica del tipo $filter$ llamada $Laplacian\ scores$, en la cual no se usan las etiquetas de la matriz, ideal para este caso.

1.3.3. Laplacian Scores

Esta es una técnica de selección de características no supervisada basada en $eigenmapas\ laplacianos$. Dada la matriz $\boldsymbol{V}$ donde cada columna representa un experto, el algoritmo logró asignarle un porcentaje a cada columna según su importancia dentro del conjunto. Su funcionamiento se puede resumir brevemente así: se toma el conjunto completo y se construye un grafo con un número de nodos igual al número de datos, se crean aristas entre los k nodos más cercanos y

[37] DASH, Manoranjan; LIU, Huan. Feature selection for classification. En: Intelligent data analysis, 1997, vol. 1, no 1-4, p. 131-156.

[38] AMARI, Shun-ichi; WU, Si. Improving support vector machine classifiers by modifying kernel functions. En: Neural Networks, 1999, vol. 12, no 6, p. 783-789.

posteriormente se construye una matriz S donde cada elemento es asignado con el valor $S_{ij} = exp(-\frac{1}{t}\|\boldsymbol{x_i} - \boldsymbol{x_j}\|^2)$ si los datos $\boldsymbol{x_j}$ y $\boldsymbol{x_j}$ tienen una arista conectándolos entre sí o $S_{ij} = 0$ si no lo están; t es una constante. Luego, definiendo f_r como cada una de las columnas de V y L_r como el puntaje asignado para cada f_r, se tiene que:

$$D = diag(S\mathbf{1}), \mathbf{1} = [1, \cdots, 1]_T, L = D - S, \quad (27)$$

$$\widetilde{f_r} = f_r - \frac{{f_r}^T D\mathbf{1}}{\mathbf{1}^T D\mathbf{1}}\mathbf{1}, L_r = \frac{\widetilde{f_r} L \widetilde{f_r}}{{\widetilde{f_r}}^T D \widetilde{f_r}}. \quad (28)$$

De esta forma se calcularon los factores de ponderación para cada experto. Este cálculo surgió de la idea de que una "buena" característica es aquella en la que dos que tienen una arista entre si se encuentran cercanos[39]. En este trabajo se usó la función *Laplacian* que ya está implementada en Matlab, en la librería de *feature selection* llamada *FSlib*.

[39] HE, Xiaofei; CAI, Deng; NIYOGI, Partha. Laplacian score for feature selection. En: Advances in neural information processing systems. 2006. p. 507-514.

2. MARCO EXPERIMENTAL

Después de diseñar la estrategia de clasificación con funciones kernel en el escenario de múltiples expertos, el último de los objetivos específicos era evaluar el desempeño del clasificador. Es importante notar que para este propósito no era necesario tener bases de datos etiquetadas por múltiples expertos reales. Esto es debido a que, se requiere tener una referencia sólida que permita hacer comparaciones acertadas; por tanto, se usan bases de datos con un solo vector de etiquetas y el escenario de múltiples expertos es simulado.

Entonces, las muestras de la base de datos se colocaron en una matriz $\boldsymbol{X} \in \mathbb{R}^{nxm}$ donde n es el número de muestras y m es el número de características. Para simular los múltiples anotadores se creó la matriz de etiquetas $\boldsymbol{D} \in \mathbb{R}^{nxL}$, para un número de etiquetadores L definidos por el usuario, la cual contienía en la primera columna el vector de etiquetas original y en el resto de columnas vectores obtenidos al corromper la primera columna. En Matlab, se usó la función *rand* para cambiar un determinado porcentaje de valores de un vector, simulando las malas etiquetas que los expertos poco confiables proveen.

Una vez creado el entorno propicio para las diferentes pruebas, se describen a continuación las diferentes bases de datos que se usaron, las medidas de desempeño y los experimentos que se desarrollaron.

2.1. BASES DE DATOS

- **Biomedic:** estos datos surgieron en un estudio para desarrollar métodos de detección para identificar portadores de un trastorno genético raro. Se realizaron cuatro mediciones m1, m2, m3, m4 en muestras de sangre. Debido a que la enfermedad es rara, solo hay unos pocos portadores de la enfermedad de los que hay datos disponibles: 209 observaciones (134 para "normales" y 75 para "portadores").
- **Fisher Iris:** Esta base de datos se encuentró en las librerías de Matlab y contiene 150 muestras de flores: 50 muestras de cada una de las tres clases de flores (Iris setosa, Iris virginica e Iris versicolor). Las cuatro características que componen cada muestra son: largo y ancho tanto del sépalo como del pétalo.
- **Wine recognition data:** Esta base de datos fue obtenida del Machine Learning Repository es el resultado de un análisis químico de vinos hechos en una misma región de Italia, pero por tres cultivadores de viñedos diferentes. Tiene

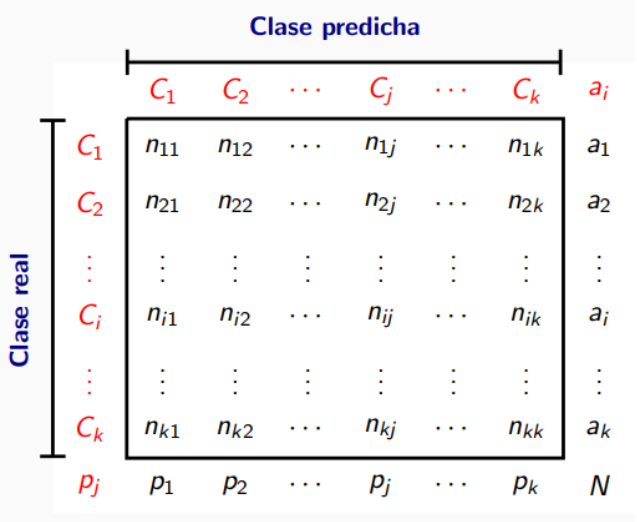
59 elementos de la primera clase, 71 de la segunda y 48 de la tercera. El análisis determinó las cantidades de 13 componentes encontrados en cada uno de los tres tipos de vinos, estas son: alcohol, ácido málico, ceniza, alcalinidad de cenizas, magnesio, fenoles totales, flavonoides, fenoles no flavonoides, proantocianinas, intensidad de color, Hue, OD280 / OD315 de vinos diluidos y prolina.

2.2. MEDIDAS DE DESEMPEÑO

A continuación, se presentan las medidas de desempeño que fueron de ayuda para comprobar la efectividad del clasificador desarrollado:

- $Matriz\ de\ confusion$: esta matriz es ampliamente usada en el aprendizaje automático ya que es la base para calcular otras medidas y permite ver qué clases está confundiendo el clasificador. Suponiendo un número de clases k, se presenta en la Figura 8 el esquema para construirla; a_i es el número real de ejemplos que pertenecen a la clase C_i, p_j el número real de ejemplos que el clasificador clasifica como perteneciente a la clase C_i, n_{ij} el número de ejemplos de la clase C_i que se clasifican como pertenecientes a la clase C_j, y N el número total de ejemplos.

Figura 8. Matriz de confusión para k clases.



Fuente:
https://campusvirtual.ull.es/ocw/pluginfile.php/15312/mod_resource/content/8/evaluacion-de-clasificadores.pdf

- $Error\ Estándar\ E_s$: es la proporción entre los datos mal clasificados y el total de datos.

- $Precisión$: es la cantidad de datos que realmente pertenecen a una clase sobre el número de datos que el clasificador asignó a esa clase:

$$Precisión(C_i) = \frac{número\ de\ datos\ pertenecientes\ a\ la\ clase\ i}{número\ de\ datos\ clasificados\ a\ la\ clase\ i}. \quad (29)$$

- $Sensitividad\ o\ True\ positive\ (TP)\ rate$: Un verdadero positivo (TP) es una situación donde la prueba sobre un dato da como resultado la pertenencia a la clase a la que efectivamente pertenece:

$$TP(C_i) = \frac{número\ de\ decisiones\ TP}{número\ de\ datos\ pertenecientes\ a\ la\ clase\ i}. \quad (30)$$

- $Especificidad\ o\ False\ positive\ (FP)\ rate$: Un falso positivo (FP) es el caso donde la prueba sobre un dato da como resultado pertenencia a la clase, pero en realidad el dato no pertenece a ésta:

$$FP(C_i) = \frac{número\ de\ decisiones\ FP}{número\ de\ datos\ pertenecientes\ a\ la\ clase\ i} \quad (31)$$

- $Curva\ ROC$: esta curva característica operativa del receptor, ROC por sus siglas en inglés (Receiver Operating Characteristic), permite representar gráficamente el desempeño del clasificador. Para construirla sobre el plano bidimensional se ubica la especificidad en el eje x y la sensitividad en el eje y. De este gráfico se puede observar que entre mayor sea el área bajo esta curva mejor es el desempeño del clasificador con respecto a cada clase.

2.3. SVM CONVENCIONAL COMO PUNTO DE REFERENCIA

Antes de desarrollar los diferentes experimentos con el clasificador propuesto, se probaron las bases de datos con una SVM convencional con kernel y con el experto (vector de etiquetas) original. Éste fue la base para comparar el desempeño del método propuesto.

Adicionalmente, para todos los experimentos se consideraron:

- Parámetro de compensación $C = 1$.
- Parámetro de la función kernel exponencial $\sigma = 1$.
- Parámetro de la función kernel polinomial $e = 2$.

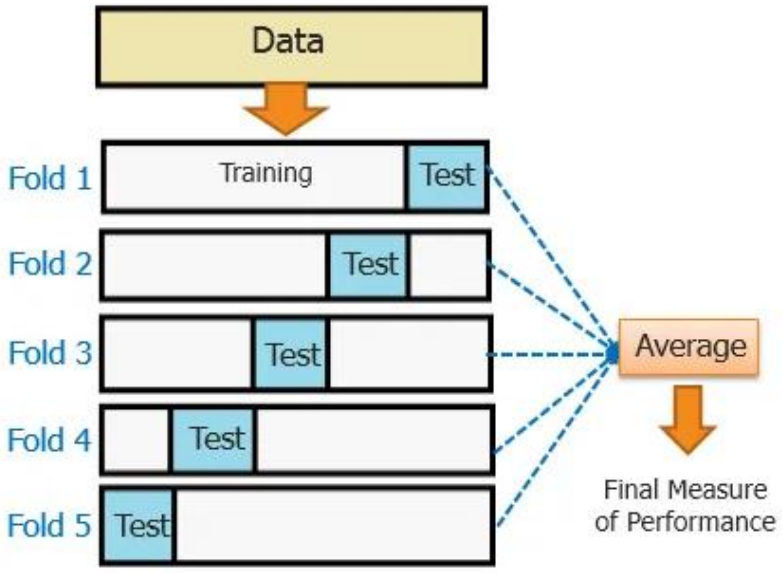
2.4. EXPERIMENTOS

Inicialmente se consideraron pruebas que evidenciaran el desempeño de la extensión multi-clase de la SVM. Luego, un primer problema de interés era verificar si los factores de ponderación mejoraban la clasificación o no; para ello se consideró hacer unas pruebas con los factores obtenidos con Laplacian score y otras dejándolos todos con el mismo valor, lo que equivale a considerar que todos los expertos tienen el mismo porcentaje de importancia. Un segundo caso de interés era comparar las diferentes funciones kernel: lineal, polinomial y exponencial.

Para cada uno de los se experimentos realizados se utilizaron dos tipos de validación:

- $Hold\text{-}out$: se dividieron las bases de datos en un 70% para entrenar y el 30% restante para validar. Los conjuntos obtenidos fueron definidos como $Xtrain$ y $Xtest$ respectivamente.
- $Cross\text{-}validation$: se hicieron diez particiones de las bases de datos (parámetro $cv = 10$), de manera que en la primera iteración se usaron las nueve partes para entrenar y la restante para validar. En la siguiente iteración se usaron nueve partes diferentes para entrenar y la restante para validar. Al final se realizó un promedio de las medidas de desempeño. En la Figura 9 se muestra un esquema de este tipo de validación.

Figura 9. Esquema de las diferentes particiones realizadas en el método $Cross\text{-}validation$ para entrenar y validar el clasificador para un parámetro cv=5.



Fuente: https://blog.contactsunny.com/data-science/different-types-of-validations-in-machine-learning-cross-validation

A continuación, se presentan los experimentos propuestos dentro del marco de pruebas:

- Experimento 1: se entrenó el clasificador con la base de datos bi-clase de Biomedic y factores de ponderación iguales para todos los expertos.
- Experimento 2: se entrenó el clasificador con la base de datos bi-clase de Biomedic y con los factores de ponderación obtenidos de Laplacian Score.
- Experimento 3: se entrenó el clasificador con las bases de datos multi-clase de IRIS y Wine recognition data, con los factores de ponderación iguales para todos los expertos.
- Experimento 4: se entrenó el clasificador con las bases de datos multi-clase de IRIS y Wine recognition data, con factores de ponderación obtenidos de Laplacian Score.

Para todos estos eventos se analizaron diferentes números de expertos $L = \{3,5,7\}$, donde se les asignó el siguiente porcentaje de error (per) a cada anotador correspondientemente de la siguiente manera $per = \{20,5,30,70,10,50,30\}$ los tres tipos de funciones kernel y los dos tipos de validación. En la Tabla 1 se presenta un resumen de las diferentes configuraciones para todos los experimentos.

Tabla 1. Resumen de las diferentes configuraciones para cada experimento.

	Clasificador		Factores de ponderación	
Experimento	Bi-clase Biomedic	Multi-clase Iris, Wine data	Iguales	Laplacian Score
1	X		X	
2	X			X
3		X	X	
4		X		X

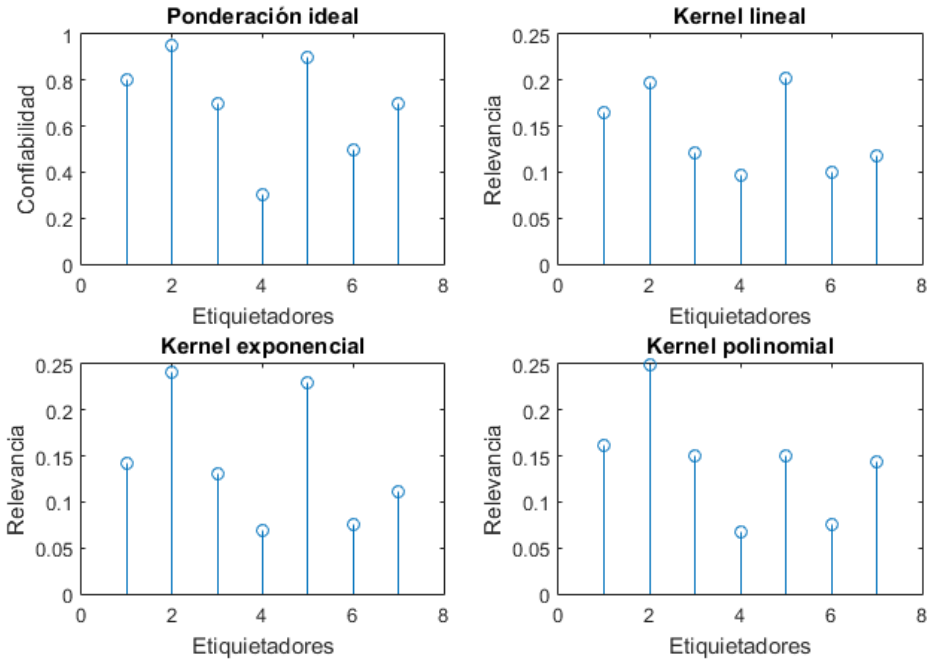
3. RESULTADOS

En esta sección se presentan los resultados obtenidos para la SVM de referencia (con el vector original) y para los diferentes experimentos, teniendo en cuenta que en cada uno de ellos se realizaron 10 iteraciones para obtener las diferentes medidas de desempeño.

3.1. Prueba de los factores de ponderación

Antes de comenzar a analizar el resultado de la técnica que se propuso en este trabajo de grado, se verificó que los factores de ponderación obtenidos mediante *Laplacian Scores* fueran acordes con el porcentaje de error introducido a los etiquetadores. Con este fin, en la Figura 10 se grafican algunos de los parámetros obtenidos, para cinco etiquetadores y la base de datos IRIS. En ella se observó que los tres tipos de kernel logran identificar cuáles expertos han sido más corruptos. Sin embargo, el kernel lineal no logra hacer una diferenciación clara entre ellos.

Figura 10 Factores de ponderación obtenidos con diferentes funciones kernel.



En la parte superior izquierda se encuentra el parámetro esperado según el porcentaje de error introducido en cada etiquetador (confiabilidad esperada); en la parte superior derecha se presentan los parámetros encontrados con un kernel lineal; en la parte inferior izquierda para un kernel exponencial y en la parte inferior derecha para un kernel polinomial.

3.2. SVM de referencia

En esta sección se presentan los resultados de haber aplicado una SVM convencional a las tres bases de datos propuestas. Como se explicó en la sección

MARCO EXPERIMENTAL, esto se hizo con el fin de tener un punto de referencia para comparar con el método propuesto en este trabajo de grado. En la Tabla 2, Tabla 3 y Tabla 4 se registran el error estándar de este clasificador para los tres diferentes kernel.

Tabla 2 Resultados del clasificador SVM convencional para la base de datos Biomedic.

	Error estándar	
Tipo de métrica	$Holdout$	$Crossvalidation$
Kernel lineal	0.6786	0.6686
Kernel exponencial	**0.3214**	**0.3314**
Kernel polinomial	0.3214	0.5686

Tabla 3 Resultados del clasificador SVM convencional para la base de datos IRIS.

	Error estándar	
Tipo de métrica	$Holdout$	$Crossvalidation$
Kernel lineal	0.6667	0.667
Kernel exponencial	**0.0356**	**0.0467**
Kernel polinomial	0.4756	0.42

Tabla 4 Resultados del clasificador SVM convencional para Wine dataset.

	Error estándar	
Tipo de métrica	$Holdout$	$Crossvalidation$
Kernel lineal	0.6731	0.6684
Kernel exponencial	**0.0962**	**0.0618**
Kernel polinomial	0.6731	0.6886

3.3. Experimento 1

Este experimento fue desarrollado con la base de datos bi-clase *Biomedic dataset.* La clasificación se realizó para un número de expertos $L = \{3,5,7\}$, haciendo pruebas para las funciones kernel lineal, exponencial y polinomial. Además, en este experimento se consideró los factores de ponderación iguales para todos los expertos. En la Tabla 5 se registra el error estándar del clasificador tanto para el método de validación $Holdout$ y como para el de $Crossvalidation$. En ella se pudo observar que es el kernel exponencial el que menor error registra.

Tabla 5. Errores estándar para el experimento 1.

	Error estándar					
Tipo de métrica	$\boldsymbol{Holdout}$			$\boldsymbol{Crossvalidation}$		
Kernel \ # de expertos	*L=3*	*L=5*	*L=7*	*L=3*	*L=5*	*L=7*
Kernel lineal	0,5789	0,6579	0,6578	0,6212	0,6686	0,6678
Kernel exponencial	0,2333	**0,2561**	0,211	0,2285	0,2274	**0,211**
Kernel polinomial	0,6578	0,6596	0,6579	0,6685	0,6686	0,6679

En la Tabla 6 y Tabla 7 se presentan las medidas de $True\ postive\ rate\ (TP)$, $False\ positive\ rate\ (FP)$ y $precisión\ (P)$ para las clase 1 y 2 respectivamente. En estas tablas es importante notar que, aunque el kernel lineal presentó un TP alto en algunos casos, esto no significa que éste haya presentado mejor desempeño puesto que su FP es igualmente alto. Por tanto, podría considerarse que el kernel exponencial presenta mejores resultados, puesto que son buenos tanto para FP como para TP.

Tabla 6. Medidas de desempeño para la clase 1 con diferentes números de expertos.

Resultados para la clase 1		$\boldsymbol{Crossvalidation}$			$\boldsymbol{Holdout}$		
		TP	FP	P	TP	FP	P
$\boldsymbol{L=3}$	Lineal	0.2334	0.0475	0.9226	0.1257	0.0294	0.9886
	Exponencial	0.9833	0.025	0.8117	0.96	0.039	0.77
	Polinomial	0	0	0.5	0	0	0.5

$L = 5$	Lineal	0.1573	0.2132	0.6765	0	0	1
	Exponencial	0.9267	0.4611	0.7994	0.9164	0.5671	0.7792
	Polinomial	0	0.76	0.8493	0	0.56	0.9
$L = 7$	Lineal	0.1866	0.3747	0.8762	0	0	1
	Exponencial	0.9984	0.5984	0.8164	0.9972	0.6536	0.7572
	Polinomial	0	0.2674	0.8682	0	0	1

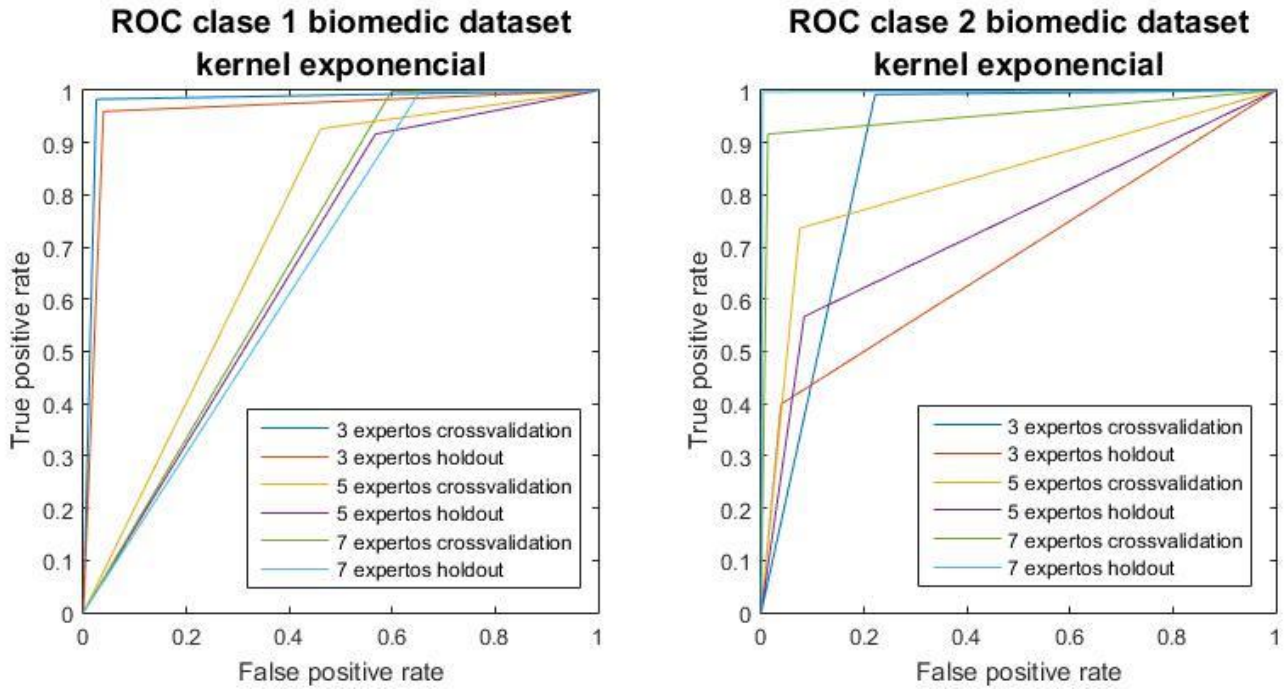
Tabla 7 Medidas de desempeño para la clase 2 con diferentes números de expertos.

Resultados para la clase 2		$Crossvalidation$			$Holdout$		
		TP	FP	P	TP	FP	P
$L = 3$	Lineal	0.9623	0.9167	0.3928	0.9706	0.8743	0.4121
	Exponencial	0.9924	0.2213	0.9473	0.3994	0.0387	0.9522
	Polinomial	0.9429	0.9874	0.4827	1	1	0.5
$L = 5$	Lineal	0.8445	0.7562	0.2973	1	1	0.3421
	Exponencial	0.7358	0.0748	0.9152	0.5671	0.0836	0.8961
	Polinomial	0.9277	0.8721	0.2702	0.9944	1	0.3409
$L = 7$	Lineal	0.9273	0.8564	0.2811	1	1	0.3
	Exponencial	0.9165	0.013	0.9983	0.9972	0.0028	0.9947
	Polinomial	0.9134	0.9374	0.2895	1	1	0.3421

Posteriormente, la Figura 11 se muestran las curvas Roc para el kernel exponencial puesto que es el que mejores resultados obtuvo. En ella se puedo notar que el mejor AUC lo tiene el caso con 3 expertos, lo cual es entendible debido a que cómo se explicó en la sección

MARCO EXPERIMENTAL, el entorno de múltiples expertos es simulado, por lo que al añadir más expertos, se añadió más ruido al modelo de predicción.

Figura 11 Curvas ROC para el experimento 1.



3.4. Experimento 2

El experimento fue desarrollado con la base de datos bi-clase *Biomedic dataset.* Para la clasificación se realizaron pruebas para un número de expertos $L = \{3,5,7\}$, con las diferentes funciones kernel: lineal, exponencial y polinomial. Además, en este experimento se usaron los factores de ponderación obtenidos mediante $Laplacian\ Score$ para todos los expertos. En la Tabla 8 se registra el error estándar para los dos métodos de validación. En ella se puedo observar el buen desempeño del algoritmo con el kernel Exponencial, puesto que presenta menor error.

Tabla 8 Errores estándar para el experimento 2.

	Error estándar

Tipo de métrica	$\boldsymbol{Holdout}$			$\boldsymbol{Crossvalidation}$		
Kernel \ # de expertos	*L=3*	*L=5*	*L=7*	*L=3*	*L=5*	*L=7*
Kernel lineal	0,5842	0,6579	0,6579	0,6686	0,6686	0,6686
Kernel exponencial	0,2473	0,228	**0,2052**	0,239	0,2118	**0,1846**
Kernel polinomial	0,6578	0,6596	0,6579	0,6687	0,6686	0,6686

En la Tabla 9 y

Tabla 10 se presentan las medidas de $True\ postive\ rate\ (TP)$, $False\ positive\ rate\ (FP)$ y $precisión\ (P)$ para las clase 1 y 2, donde se observó nuevamente que un buen balance entre estas tres medidas lo tiene el kernel exponencial.

Tabla 9 Medidas de desempeño para la clase 1 con diferentes números de expertos.

Resultados para la clase 1		$\boldsymbol{Crossvalidation}$			$\boldsymbol{Holdout}$		
		TP	FP	P	TP	FP	P
$\boldsymbol{L=3}$	Lineal	0.3574	0.0353	0.8362	0.1157	0.0176	0.9923
	Exponencial	0.9364	0.5283	0.7551	0.9891	0.7091	0.7392
	Polinomial	0.1336	0.2543	0.8365	0	0	1
$\boldsymbol{L=5}$	Lineal	0.4257	0.0163	0.9264	0	0	1
	Exponencial	0.9732	0.5645	0.8267	0.9718	0.5968	0.7707
	Polinomial	0.013	0.2551	0.7240	0	0.0056	0.9000
$\boldsymbol{L=7}$	Lineal	0.2873	0.1673	0.8274	0	0	1
	Exponencial	0.9328	0.3761	0.6923	0.9889	0.5791	0.7755
	Polinomial	0.0332	0.1344	0.9873	0	0	1

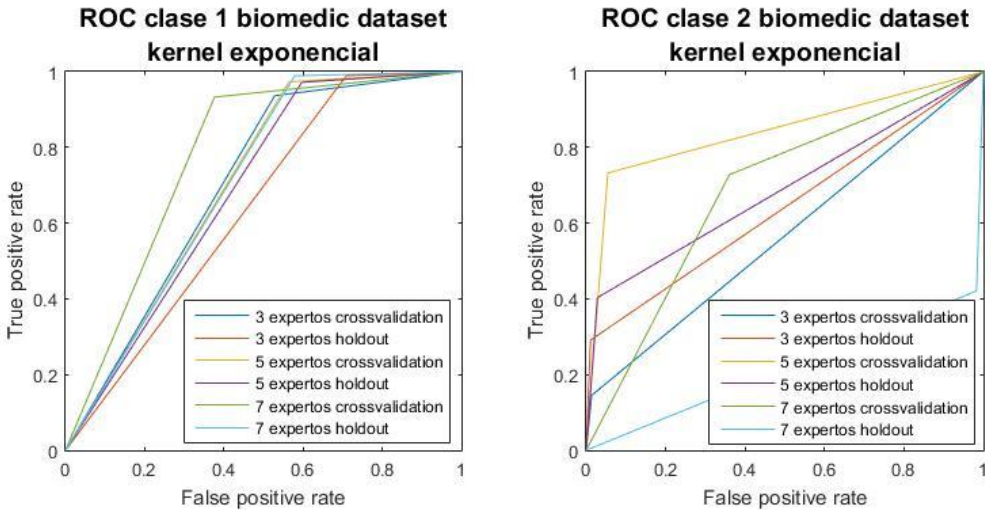
Tabla 10 Medidas de desempeño para la clase 2 con diferentes números de expertos.

Resultados para la clase 2		$\boldsymbol{Crossvalidation}$			$\boldsymbol{Holdout}$		
		TP	FP	P	TP	FP	P
$\boldsymbol{L=3}$	Lineal	0.9378	0.7364	0.4622	0.9824	0.8843	0.3973
	Exponencial	0.1451	0.0138	0.9635	0.2909	0.0109	0.9774
	Polinomial	0.9026	0.9834	0.3037	1	1	0.3421
$\boldsymbol{L=5}$	Lineal	0.9942	0.9963	0.3524	1	1	0.3421
	Exponencial	0.7325	0.0544	0.9467	0.4032	0.0282	0.9398
	Polinomial	1	1	0.3166	0.9994	1	0.3409
$\boldsymbol{L=7}$	Lineal	0.9332	0.9641	0.3690	1	1	0.3421
	Exponencial	0.7281	0.3607	0.9183	0.4209	0.9826	0.9826
	Polinomial	1	0.9893	0.3074	1	1	0.3421

Posteriormente, en la **Figura 12** Curvas ROC para el experimento 2.Figura 12 se muestran las curvas ROC para el kernel exponencial puesto que es el que mejores resultados obtuvo. En ella se puede notar que el mejor AUC lo tuvo el caso con 3 expertos, lo cual concuerda con lo que se explicó en la sección

MARCO EXPERIMENTAL, ya que al añadir más expertos se añade más ruido al modelo de predicción.

Figura 12 Curvas ROC para el experimento 2.



3.5. Experimento 3

El experimento fue desarrollado con dos bases de datos multi-clase: *IRIS* y *Wine dataset.* Para la clasificación se realizó pruebas para un número de expertos $L = \{3,5,7\}$, haciendo pruebas para las funciones kernel lineal, exponencial y polinomial. Además, en este experimento se consideró los factores de ponderación iguales para todos los expertos. En la Tabla 11 y la Tabla 12 se registra el error estándar del clasificador con las bases de datos Iris y Wine respectivamente para los dos métodos de validación.

Tabla 11 Errores estándar en el experimento 3 con la base de datos IRIS.

	Error estándar					
Tipo de métrica	***Holdout***			***Crossvalidation***		
Kernel \ # de expertos	*L=3*	*L=5*	*L=7*	*L=3*	*L=5*	*L=7*
Kernel lineal	0,6244	0,62	0,6355	0,5533	0,6333	0,6333
Kernel exponencial	**0,0555**	0,0755	0,0688	0,0667	0,0599	**0,0533**
Kernel polinomial	0,6578	0,66	0,6155	0,6667	0,64	0,6267

Tabla 12 Errores estándar en el experimento 3 con la base de datos Wine dataset.

	Error estándar					
Tipo de métrica	***Holdout***			***Crossvalidation***		
Kernel \ # de expertos	*L=3*	*L=5*	*L=7*	*L=3*	*L=5*	*L=7*
Kernel lineal	0,5962	0,6056	0,6151	0,6294	0,6059	0,6529
Kernel exponencial	0,0792	0,1094	**0,0849**	0,0941	0,1294	**0,1294**
Kernel polinomial	0,6415	0,6358	0,6226	0,6177	0,5706	0,5411

En la Tabla 13, Tabla 14 y

Tabla 15 se presentan las diferentes medidas de desempeño para las clases 1, 2 y 3 respectivamente de la base de datos IRIS. En estos resultados se pudo observar que la función exponencial tuvo un muy buen desempeño ya que logró obtener la puntuación más alta en TP y la mas baja en FP para la clase 1, y un muy buen balance para las 2 clases restantes.

Tabla 13 Medidas de desempeño para la clase 1 de la base de datos IRIS, experimento 3.

Resultados para la clase 1		***Crossvalidation***			***Holdout***		
		TP	FP	P	TP	FP	P
$\boldsymbol{L=3}$	Lineal	0.5	0.25	0.8167	0.5	0.4	0.7267
	Exponencial	1	0	1	1	0	1
	Polinomial	0.24	0.26	0.7879	0.35	0.3	0.7956
$\boldsymbol{L=5}$	Lineal	0.1	0	1	0.1	0	1
	Exponencial	1	0	1	1	0	1
	Polinomial	0.28	0.19	0.8690	0.1286	0.0969	0.9295
$\boldsymbol{L=7}$	Lineal	0.1	0	1	0	0	1
	Exponencial	1	0	1	1	0.0063	0.9856

	Polinomial	0.24	0.14	0.8889	0.4	0.2141	0.8052

Tabla 14 Medidas de desempeño para la clase 2 de la base de datos IRIS, experimento 3.

Resultados para la clase 2		$\boldsymbol{Crossvalidation}$			$\boldsymbol{Holdout}$		
		TP	FP	P	TP	FP	P
$\boldsymbol{L=3}$	Lineal	0.1	0.08	0.9385	0	0	**1**
	Exponencial	0.94	0.07	0.8929	0.9767	0.0688	0.8820
	Polinomial	0.1	0.05	0.95	0.1	0.1037	0.8311
$\boldsymbol{L=5}$	Lineal	0.7	0.6	0.5667	0.3	0.3037	0.7022
	Exponencial	0.96	0.007	0.8976	0.9778	0.0984	0.8369
	Polinomial	0.28	0.25	0.8111	0.2857	0.2446	0.7177
$\boldsymbol{L=7}$	Lineal	0.3	0.3	0.8	0.2	0.2387	0.7756
	Exponencial	0.94	0.05	0.9262	0.9706	0.0784	0.8692
	Polinomial	0.58	0.53	0.6162	0.1	0.1124	0.7444

Tabla 15 Medidas de desempeño para la clase 3 de la base de datos IRIS, experimento 3.

Resultados para la clase 3		$\boldsymbol{Crossvalidation}$			$\boldsymbol{Holdout}$		
		TP	FP	P	TP	FP	P
$\boldsymbol{L=3}$	Lineal	0.74	0.5	0.633	0.6	0.5529	0.6268
	Exponencial	0.86	0.03	0.9548	0.8774	0.0125	0.9730
	Polinomial	0.66	0.69	0.3774	0.5909	0.5750	0.6075
$\boldsymbol{L=5}$	Lineal	0.3	0.35	0.7	0.6909	0.65	0.565
	Exponencial	0.86	0.02	0.9667	0.8252	0.0118	0.9733
	Polinomial	0.52	0.52	0.5788	0.5886	0.6646	0.4046
$\boldsymbol{L=7}$	Lineal	0.7	0.65	0.55	0.8	0.7538	0.5042

	Exponencial	0.9	0.03	0.95	0.8460	0.0156	0.9681
	Polinomial	0.3	0.27	0.75	0.6775	0.5826	0.5701

En la Tabla 16, Tabla 17 y Tabla 18 se presentan las diferentes medidas de desempeño para las clases 1, 2 y 3 respectivamente de la base de datos *Wine dataset*. Cabe resaltar que, nuevamente, aunque los kernel lineal y polinomial lograron en ocasiones un TP alto, su FP también es alto, contrario al desempeño del kernel polinomial donde el TP es alto y el FP fue bajo como se desearía.

Tabla 16 Medidas de desempeño para la clase 1 de la base de datos Wine data, experimento 3.

Resultados para la clase 1		$\boldsymbol{Crossvalidation}$			$\boldsymbol{Holdout}$		
		TP	FP	P	TP	FP	P
$\boldsymbol{L=3}$	Lineal	1	0.9659	0.3556	1	0.8889	0.3738
	Exponencial	1	0.0902	0.8655	0.9868	0.0785	0.8724
	Polinomial	0.9	0.8364	0.4368	1	0.9750	0.347
$\boldsymbol{L=5}$	Lineal	1	0.8909	0.378	1	0.9056	0.3659
	Exponencial	1	0.1614	0.7944	1	0.133	0.8135
	Polinomial	0.8667	0.6636	0.4864	0.5909	0.5567	0.388
$\boldsymbol{L=7}$	Lineal	1	1	0.3471	1	0.9017	0.3803
	Exponencial	1	0.1447	0.8057	1	0.0986	0.8396
	Polinomial	0.8	0.6068	0.4615	0.8310	0.7666	0.4458

Tabla 17 Medidas de desempeño para la clase 2 de la base de datos Wine data, experimento 3.

Resultados para la clase 2		$\boldsymbol{Crossvalidation}$			$\boldsymbol{Holdout}$		
		TP	FP	P	TP	FP	P
$\boldsymbol{L=3}$	Lineal	0	0	1	0.014	0.003	0.9667

	Exponencial	0.7875	0.01	0.9875	0.8077	0.0105	0.9877
	Polinomial	0.0393	0	1	0.0182	0	1
$L = 5$	Lineal	0	0	1	0.0725	0.0207	0.9333
	Exponencial	0.6911	0	1	0.7284	0.0029	0.9941
	Polinomial	0.1286	0.13	0.8733	0.2353	0.2427	0.6495
$L = 7$	Lineal	0	0	1	0	0	1
	Exponencial	0.7589	0.05	0.9349	0.7946	0.0033	0.9955
	Polinomial	0.2893	0.19	0.8162	0.2353	0.2427	0.6495

Tabla 18 Medidas de desempeño para la clase 3 de la base de datos Wine data, experimento 3

Resultados para la clase 3		$Crossvalidation$			$Holdout$		
		TP	FP	P	TP	FP	P
$L = 3$	Lineal	0.1	0	1	0.2108	0.0122	0.9688
	Exponencial	0.975	0.0385	0.9	0.9917	0.308	0.9245
	Polinomial	0.2	0.1	0.8067	0.0397	0	1
$L = 5$	Lineal	0.2	0.0308	0.955	0.0969	0	1
	Exponencial	1	0.0308	0.92	0.9941	0.0281	0.9271
	Polinomial	0.325	0.0769	0.75	0.1944	0.1887	0.8078
$L = 7$	Lineal	0	0	1	0.1504	0.023	0.9625
	Exponencial	0.8750	0.0077	0.98	0.9917	0.0231	0.9418
	Polinomial	0.25	0.0462	0.9194	0.1875	0.1027	0.920

Adicionalmente, en la Figura 13 y Figura 14 se presentan las curvas Roc para las bases de datos IRIS y Wine dataset respectivamente. En estas se observó que el desempeño del clasificador es muy parecido para los diferentes números de expertos. En la base de datos IRIS se pudo observar que la clase que se identifican muy bien las clases 1 y 3, mientras que en la 2 presenta mayores errores. En la

base de datos Wine dataset la clase que tiene más dificultad de identificar presentó es la 2, sin embargo, el número de expertos no parece afectar la clasificación.

Figura 13 Curvas ROC para el experimento 3 con la base de datos IRIS.

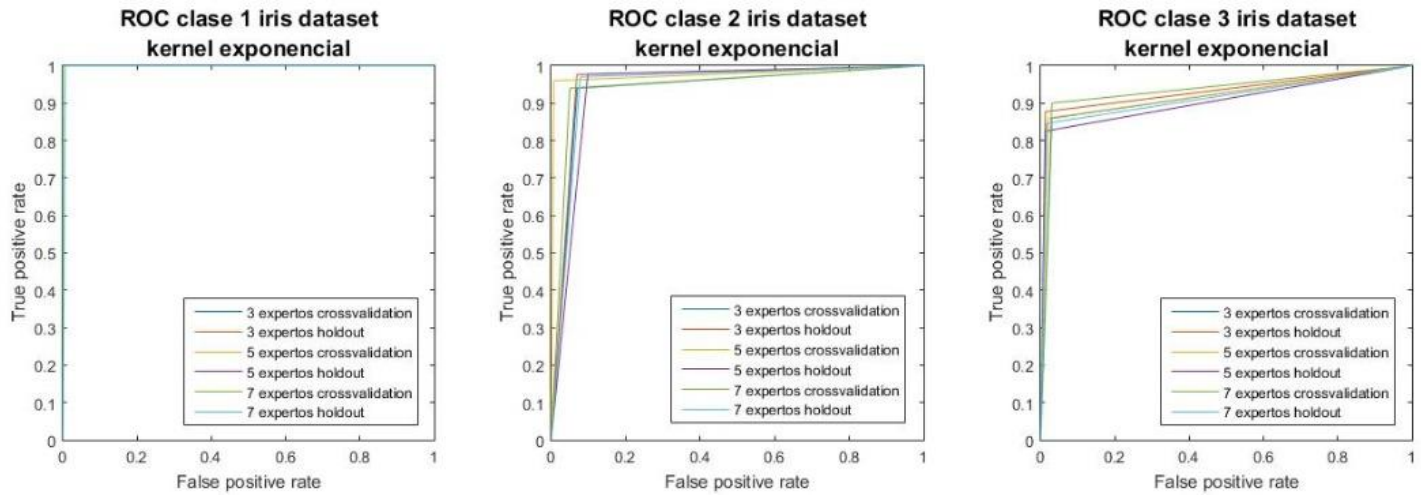
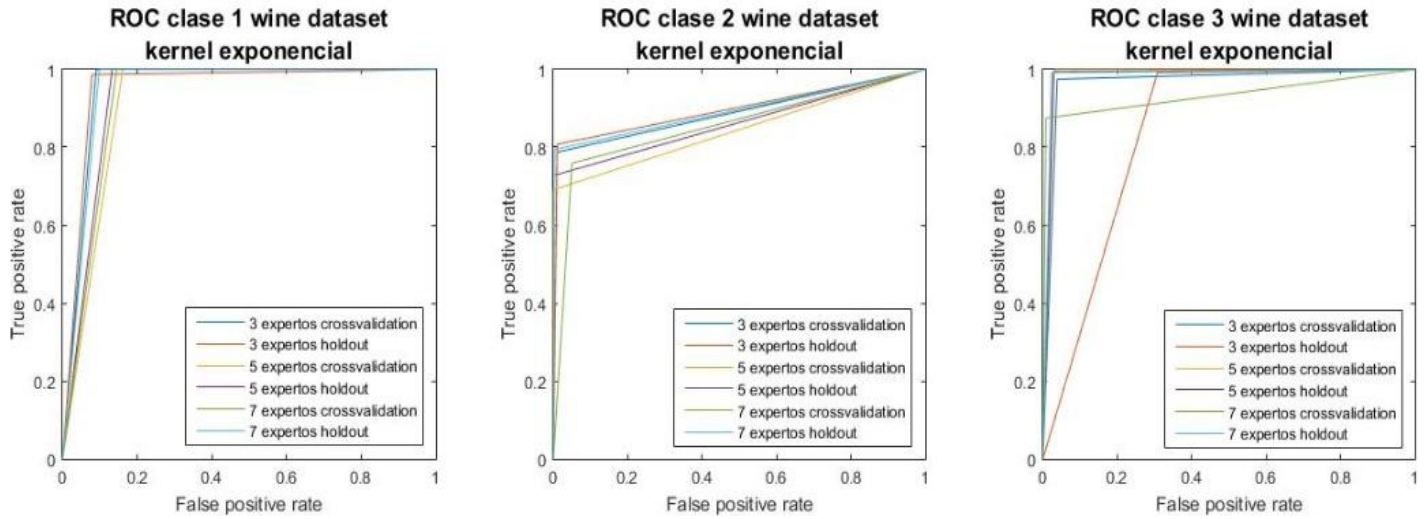


Figura 14 Curvas ROC para el experimento 3 con la base de datos Wine dataset.



3.6. Experimento 4

El experimento fue desarrollado con dos bases de datos multi-clase: *IRIS* y *Wine dataset.* Para la clasificación se realizaron pruebas para un número de expertos $L =$

$\{3,5,7\}$, haciendo pruebas para las funciones kernel lineal, exponencial y polinomial. Además, en este experimento se consideró los factores de ponderación obtenidos mediante $Laplacian\ Score$ para todos los expertos. En la Tabla 19 y la Tabla 20 se registra el error estándar del clasificador con las bases de datos Iris y Wine respectivamente para los dos métodos de validación.

Tabla 19 Errores estándar en el experimento 4 con la base de datos IRIS.

	Error estándar					
Tipo de métrica	***Holdout***			***Crossvalidation***		
Kernel \ # de expertos	*L=3*	*L=5*	*L=7*	*L=3*	*L=5*	*L=7*
Kernel lineal	0,6022	0,6222	0,6533	0,6	0,6533	0,6667
Kernel exponencial	0,0622	0,0711	0,0667	0,0733	0,0667	0,0399
Kernel polinomial	0,6266	0,7067	0,6289	0,6533	0,62	0,6467

Tabla 20 Errores estándar en el experimento 4 con la base de datos Wine Dataset.

	Error estándar					
Tipo de métrica	***Holdout***			***Crossvalidation***		
Kernel \ # de expertos	*L=3*	*L=5*	*L=7*	*L=3*	*L=5*	*L=7*
Kernel lineal	0,5509	0,6245	0,6226	0,5706	0,5941	0,6529
Kernel exponencial	**0,0773**	0,1037	0,1132	**0,1059**	0,1294	0,1059
Kernel polinomial	0,6377	0,6339	0,5773	0,6176	0,5941	0,5823

En la

Tabla **21**, Tabla 22 y Tabla 23 se presentan las diferentes medidas de desempeño para las clases 1, 2 y 3 respectivamente de la base de datos IRIS. En estos resultados se hizo el mismo análisis que en los anteriores experimentos, donde el kernel exponencial sigue presentando resultados bueno para las tres medidas de desempeño.

Tabla 21 Medidas de desempeño para la clase 1 de la base de datos IRIS, experimento 4.

Resultados para la clase 1		$\boldsymbol{Crossvalidation}$			$\boldsymbol{Holdout}$		
		TP	FP	P	TP	FP	P
$\boldsymbol{L=3}$	Lineal	0.5	0.3	0.8	0.5	0.4	0.7267
	Exponencial	1	0	1	1	0	1
	Polinomial	0.34	0.26	0.8074	0.3929	0.3	0.7956
$\boldsymbol{L=5}$	Lineal	0.14	0.09	0.9357	0.1	0	1
	Exponencial	1	0	1	1	0.0030	0.9923
	Polinomial	0.34	0.2	0.8667	0.2	0.0969	0.9295
$\boldsymbol{L=7}$	Lineal	0.1	0.1	0.9333	0	0	1
	Exponencial	1	0	1	1	0.0061	0.9857
	Polinomial	0.1	0	1	0.03	0	1

Tabla 22 Medidas de desempeño para la clase 2 de la base de datos IRIS, experimento 4.

Resultados para la clase 2		$\boldsymbol{Crossvalidation}$			$\boldsymbol{Holdout}$		
		TP	FP	P	TP	FP	P
$\boldsymbol{L=3}$	Lineal	0.1	0.05	0.95	0.2	0.1645	0.8790
	Exponencial	0.94	0.08	0.881	0.8717	0.0757	0.9711
	Polinomial	0.04	0.05	0.9286	0.2	0.1926	0.8730
$\boldsymbol{L=5}$	Lineal	0.6	0.57	0.5121	0.3	0.3074	0.7022
	Exponencial	0.94	0.07	0.8976	0.8550	0.0897	0.9778
	Polinomial	0.3	0.25	0.8167	0.2294	0.2778	0.6160

$L = 7$	Lineal	0.1	0.05	0.95	0.4	0.4	0.6733
	Exponencial	0.94	0.04	0.9429	0.8727	0.0748	0.9711
	Polinomial	0.16	0.16	0.776	0.2	0.1926	0.8730

Tabla 23 Medidas de desempeño para la clase 3 de la base de datos IRIS, experimento 4.

Resultados para la clase 3		$Crossvalidation$			$Holdout$		
		TP	FP	P	TP	FP	P
$L = 3$	Lineal	0.6	0.55	0.6167	0.4579	0.3529	0.75557
	Exponencial	0.84	0.03	0.9548	0.8647	0.0155	0.9675
	Polinomial	0.66	0.67	0.3825	0.5	0.4595	0.5931
$L = 5$	Lineal	0.3	0.32	0.7	0.6818	0.65	0.5647
	Exponencial	0.86	0.03	0.95	0.8353	0.0118	0.9733
	Polinomial	0.5	0.48	0.5886	0.4594	0.7037	0.3589
$L = 7$	Lineal	0.8	0.85	0.3667	0.6	0.6	0.613
	Exponencial	0.92	0.02	0.9667	0.8551	0.0155	0.9671
	Polinomial	0.8	0.81	0.3833	0.8	0.7915	0.3142

En la Tabla 24, Tabla 25 y Tabla 25 se presentan las diferentes medidas de desempeño para las clases 1, 2 y 3 respectivamente de la base de datos *Wine dataset*. Nuevamente, en estos resultados se puedo observar que la función exponencial tiene un mejor desempeño.

Tabla 24 Medidas de desempeño para la clase 1 de la base de datos Wine data, experimento 4.

Resultados para la clase 1		$Crossvalidation$			$Holdout$		
		TP	FP	P	TP	FP	P
$L = 3$	Lineal	1	0.8803	0.3985	0.8176	0.7373	0.5020

	Exponencial	1	0.1265	0.8367	0.9809	0.0761	0.8792
	Polinomial	0.9233	0.8273	0.4352	1	0.9645	0.3496
$L = 5$	Lineal	1	0.8636	0.3978	1	0.9502	0.3538
	Exponencial	1	0.1538	0.8069	0.9839	0.1178	0.8269
	Polinomial	0.7	0.5636	0.4315	0.5496	0.5295	0.369
$L = 7$	Lineal	1	1	0.3471	0.9524	0.8605	0.4123
	Exponencial	1	0.1356	0.82	1	0.1245	0.8088
	Polinomial	0.9	0.7705	0.3672	0.7006	0.6013	0.4689

Tabla 25 Medidas de desempeño para la clase 2 de la base de datos Wine data, experimento 4.

Resultados para la clase 2		$Crossvalidation$			$Holdout$		
		TP	FP	P	TP	FP	P
$L = 3$	Lineal	0.075	0	1	0.2	0.1408	0.9085
	Exponencial	0.7446	0	1	0.8089	0.0102	0.9863
	Polinomial	0	0.01	0.9	0.0227	0.032	0.9833
$L = 5$	Lineal	0	0	1	0	0	1
	Exponencial	0.6875	0	1	0.7516	0.0089	0.9841
	Polinomial	0.2143	0.17	0.85	0.2686	0.2692	0.6601
$L = 7$	Lineal	0	0	1	0	0	1
	Exponencial	0.7446	0	1	0	0	1
	Polinomial	0.1571	0.1367	0.8271	0.3662	0.2687	0.7989

Tabla 26 Medidas de desempeño para la clase 3 de la base de datos Wine data, experimento 4.

	$Crossvalidation$	$Holdout$

Resultados para la clase 3		TP	FP	P	TP	FP	P
$\boldsymbol{L=3}$	Lineal	0.2	0	1	0.2571	0	1
	Exponencial	1	0.0308	0.92	1	0.0312	0.9264
	Polinomial	0.25	0.1	0.806	0.0439	0.0023	0.9
$\boldsymbol{L=5}$	Lineal	0.25	0.0385	0.93	0.1237	0	1
	Exponencial	1	0.0385	0.9	1	0.0285	0.9291
	Polinomial	0.325	0.1615	0.6919	0.2031	0.1936	0.7955
$\boldsymbol{L=7}$	Lineal	0	0	1	0.1875	0.0711	0.9357
	Exponencial	1	0.0231	0.94	1	0.0405	0.8978
	Polinomial	0.125	0.077	0.975	0.1468	0.0159	0.9549

Las curvas ROC para este experimento se muestran en la Figura 15 y Figura 16. En ellas se evidenció una leve diferencia para los diferentes números de expertos. Mientras que en la base datos de IRIS el mejor resultado fue para $L = 3$, en Wine dataset fue para $L = 5$.

Figura 15 Curvas ROC para el experimento 4 con la base de datos de IRIS

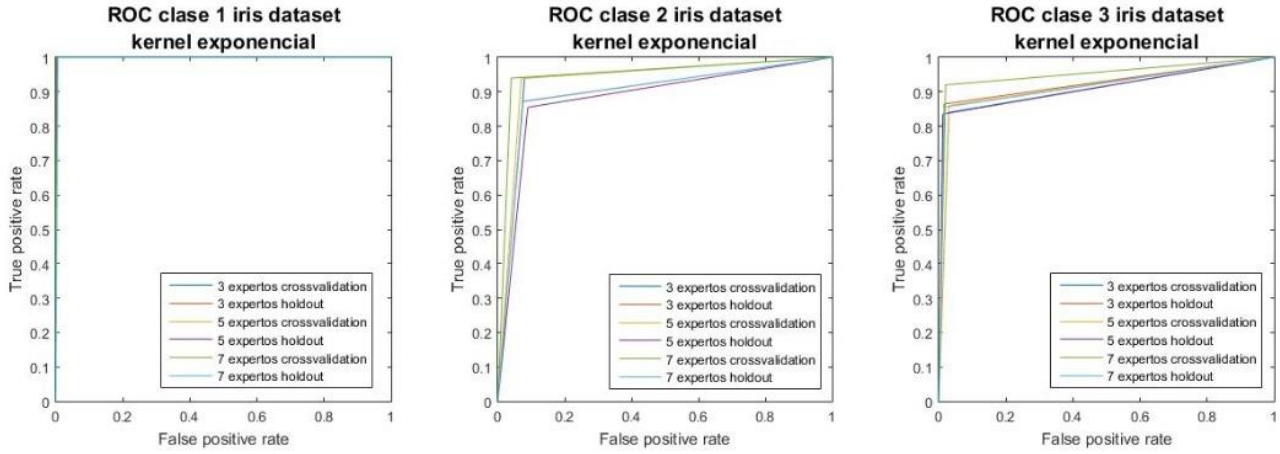
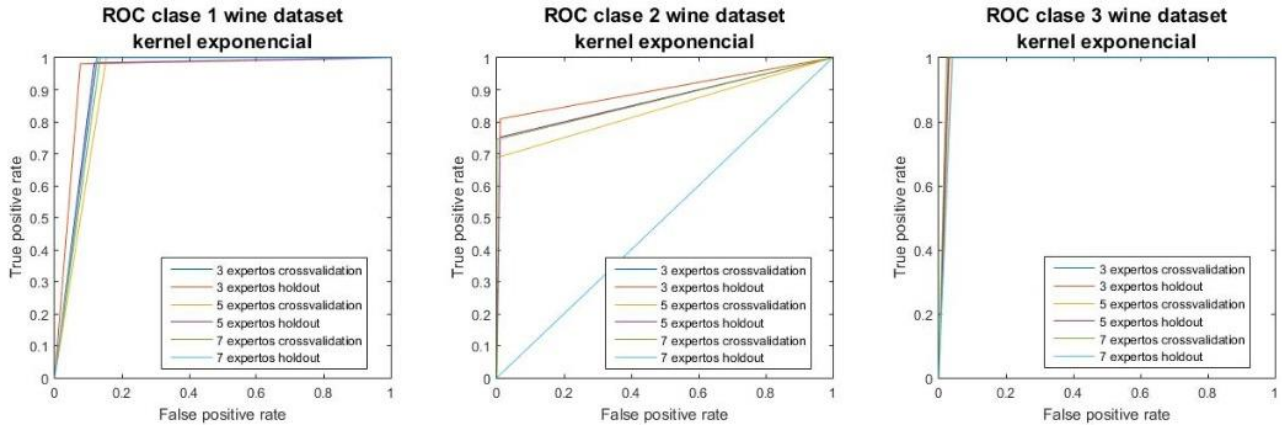


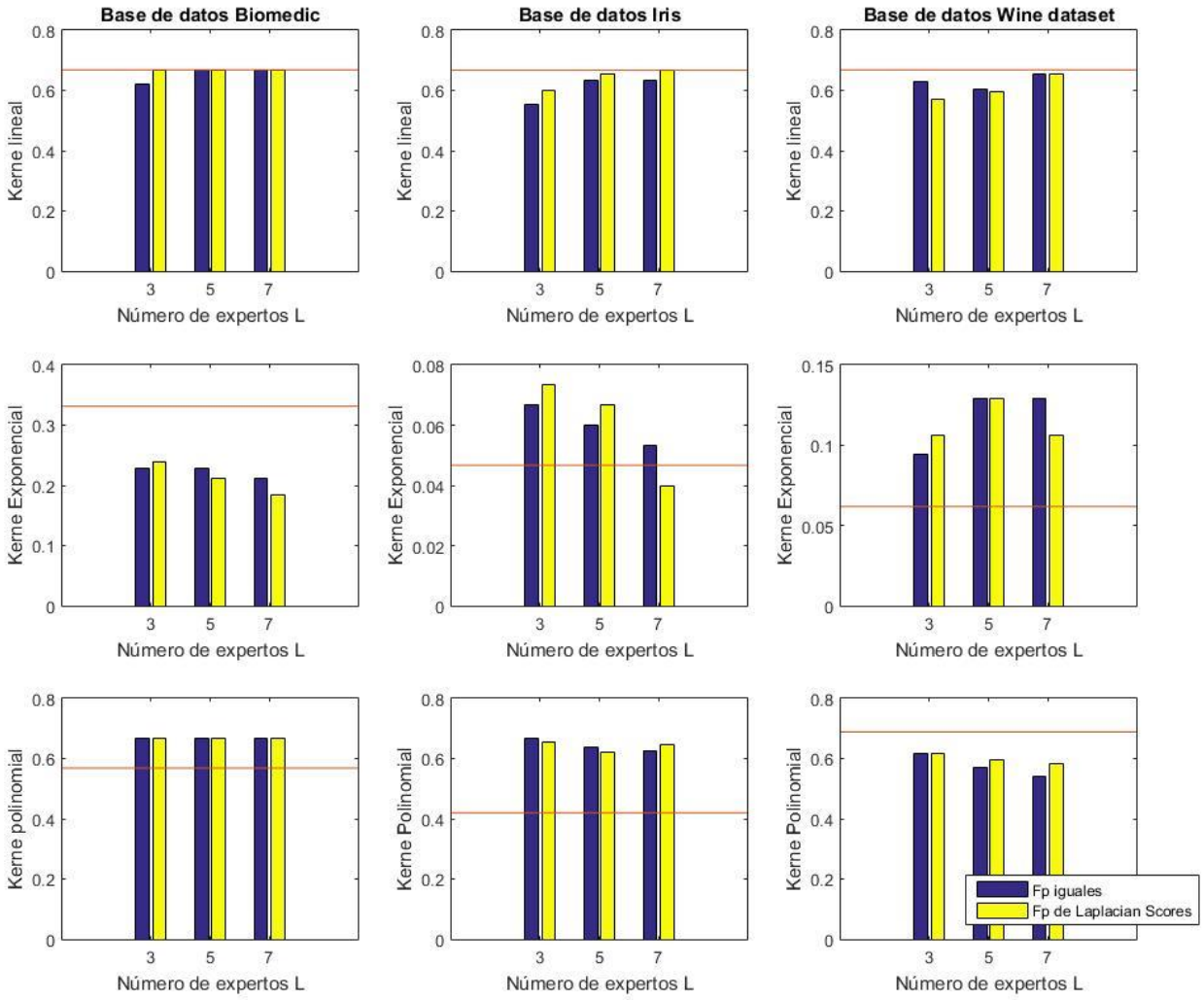
Figura 16 Curvas ROC para el experimento 4 con Wine dataset.



3.7. Gráfica de comparación de los 4 experimentos con el SVM de referencia

Los resultados de los cuatro experimentos permitieron analizar muy bien el desempeño del clasificador en las diferentes configuraciones propuestas; sin embargo, en un análisis global se puede visualizar en la Figura 17, donde se presenta el error de clasificación para cada base de datos con factores de ponderación iguales (barras azules) y con factores de ponderación obtenidos por Laplacian Scores (barras amarillas) comparados con el error del SVM de referencia (línea recta).

Figura 17 Gráfico comparativo entre el desempeño del método propuesto con el del clasificador SVM convencional, para las tres bases de datos.



En la Figura 17 se puede apreciar que el kernel lineal presentó errores altos tanto para la SVM convencional como para el método propuesto. Esto es coherente debido a que la función lineal no es muy útil para datos no separables linealmente. De igual forma, el kernel polinomial presentó errores bastante altos y solo es capaz de mejorar el desempeño del SVM convencional con la *Wine dataset*. Con respecto al kernel exponencial, se observó que el método propuesto logró errores comparables con una SVM convencional incluso con la matriz de etiquetas corrupta. Adicionalmente, no se registraron cambios notables para ninguno de los experimentos cuando se usaron los factores de ponderación iguales a cuando se usaron factores de ponderación obtenidos por *Laplacian Scores*. Esto pudo deberse a que la matriz kernel en sí es una representación de la calidad de cada etiquetador,

ya que como se explicó en el desarrollo de la investigación, el proceso de multiplicar la máscara con la función kernel, pone en duda la opinión del experto.

4. TRABAJO FUTURO

A lo largo del desarrollo de este trabajo de grado surgieron algunas ideas sobre aspectos que pueden ser analizados en detalle en futuro trabajos, con el fin de mejorar el desempeño del enfoque propuesto. Estas son:

- Probar otros métodos de análisis de relevancia de variables para el cálculo de los factores de ponderación de las matrices kernel, con el fin de lograr mejores resultados. De forma que se haga un análisis exhaustivo para determinar si incluir los factores de ponderación mejora o no el desempeño del clasificador. Esto es importante debido a que el cálculo de dichos parámetros implica un coste computacional adicional.
- Investigar en nuevas técnicas para crear el vector de etiquetas de referencia en lugar del voto mayoritario y analizar si se obtienen mejores medidas de desempeño. Entre ellas podría considerarse usar los mismos factores de ponderación.
- Es importante tener en cuenta que el método fue desarrollado con base en una SVM de margen suave, cuyo parámetro de compensación no fue calibrado exhaustivamente; es decir, no se usó optimización robusta (gradiente descendiente estocástico) ni heurística suficiente (algoritmos genéticos, validación cruzada, etc.). Es por esto que se recomienda hacer ajustes en dicho parámetro.

5. CONCLUSIONES

Dentro del desarrollo de este trabajo de grado se logró obtener un método de clasificación supervisada que funciona adecuadamente en un escenario de múltiples expertos, basado en matrices kernel. Este enfoque novedoso permitió tomar como base el clasificador SVM sin la necesidad de modificar su estructura principal y conservando una de sus más notables cualidades de no usar técnicas heurísticas.

Además, la máscara propuesta demostró una buena representación de la calidad de etiquetado de cada experto, siendo una herramienta adecuada para obtener los factores de ponderación, aplicando técnicas de *feature selection.* En este mismo análisis, se observó que las tres funciones kernel obtuvieron parámetros de ponderación acordes a los esperados.

La aplicación de esta metodología a bases de datos complejas como las que se usaron en este trabajo, demostró que es más conveniente usar el kernel de tipo exponencial, ya que es el que mejores medidas de desempeño presenta a lo largo de los experimentos realizados. Los resultados además mostraron que el algoritmo es robusto frente al número de etiquetadores ruidosos, así como también, frente a bases de datos multi-clase.

BIBLIOGRAFIA

THEODORIDIS. Sergios; KOUTROUMBAS- Konstantinos. Pattern recognition. Segunda edición. San Diego: Elsevier, 2003.

Haykin. Simon. Redes neuronales y aprendizaje automático. Tercera edición. Hamilton: Prentice Hall PTR, 1994.

Bishop. Christopher. Pattern Recognition and Machine Learning. New York: Springer-Verlag, 2006.

LODHIA, ZeeshanAhmad; RASOOL, Akhtar; HAJELA, Gaurav. A survey on machine learning and outlier detection techniques. En: International journal of computer science and network security, 2017, vol. 17, no 5, p. 271-276.

SONG, Mingjun; CIVCO, Daniel. Road extraction using SVM and image segmentation. En: Photogrammetric Engineering & Remote Sensing, 2004, vol. 70, no 12, p. 1365-1371.

BALEÓN, Emmanuel. Sistema de visión artificial para el reconocimiento de emociones faciales aplicado al estudio de expresiones del usuario. *Repositorio nacional conacyt*, 2017.

TAORMINA, Riccardo; CHAU, Kwok-Wing; SETHI, Rajandrea. Artificial neural network simulation of hourly groundwater levels in a coastal aquifer system of the Venice lagoon. En: Engineering Applications of Artificial Intelligence, 2012, vol. 25, no 8, p. 1670-1676.

CAIPE, Angela; MUÑOZ, Jorge; PELUFFO, Diego. Machine Learning for the prediction of preterm pregnancy using EHG signals. Trabajo de grado (Ingeniero electrónico). Pasto, 2016. Universidad de Nariño. Facultad de ingeniería. Departamento de Electrónica.

HENAO, Ricardo; HURTADO, Jorge; CASTELLANOS, German. Selección de hiperparámetros en máquinas de soporte vectorial utilizando adaptación de matriz de covarianza. En: Scientia et technica, 2005, vol. 1, no 27.

BOSER, Bernhard; GUYON, Isabelle; VAPNIK, Vladimir. A training algorithm for optimal margin classifiers. En: Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory. 2003. p. 144-152

[1]YAN, Yan. Active learning from crowds. En: ICML. 2011. p. 1161-1168.

SHENG, Victor; PROVOST, Foster; IPEIROTIS, Panagiotis. Get another label? improving data quality and data mining using multiple, noisy labelers. En: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008. p. 614-622.

DEKEL, Ofer; SHAMIR, Ohad. Good learners for evil teachers. En: Proceedings of the 26th annual international conference on machine learning. ACM, 2009. p. 233-240.

YAN, Yan. Active learning from crowds. En: ICML. 2011. p. 1161-1168

MURILLO, Santiago. Metodología para el aprendizaje de máquina a partir de múltiples expertos en procesos de clasificación de bioseñales. Trabajo de grado (Master of Engineering - Industrial

Automation). Manizales, 2013. Universidad Nacional de Colombia-Sede Manizales. Facultad de Ingeniería y Arquitectura. Departamento de ingeniería Eléctrica, Electrónica y Computación.

GUAN, Melody Y., et al. Who said what: Modeling individual labelers improves classification. En: Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

GIL, J.; ALVAREZ, A.; OROZCO, A. Learning from multiple annotators using kernel alignment. En: Pattern Recognition Letters, 2018, vol. 116, p. 150-156.

HSU, Chih-Wei; LIN, Chih-Jen. A comparison of methods for multiclass support vector machines. En: IEEE transactions on Neural Networks, 2002, vol. 13, no 2, p. 415-425.

RAKOTOMAMONJY, Alain. SimpleMKL. En: Journal of Machine Learning Research, 2008, vol. 9, no Nov, p. 2491-2521.

NALEPA, Jakub; KAWULOK, Michal; DUDZIK, Wojciech. Tuning and Evolving Support Vector Machine Models. En: International Conference on Man–Machine Interactions. Springer, Cham, 2017. p. 418-428.

JIANG, Hao. Stationary Mahalanobis kernel SVM for credit risk evaluation. En: Applied Soft Computing, 2018, vol. 71, p. 407-417.

DILLER, Gerhard-Paul. Machine learning algorithms estimating prognosis and guiding therapy in adult congenital heart disease: data from a single tertiary centre including 10 019 patients. En: European heart journal, 2019, vol. 40, no 13, p. 1069-1077.

BISGIN, Halil. Comparing SVM and ANN based machine learning methods for species identification of food contaminating beetles. En: Scientific reports, 2018, vol. 8, no 1, p. 6532.

CONWAY, Drew; WHITE, John. Machine learning for hackers. Sebastopol: O'Reilly Media, 2012.

FLACH, Peter. Machine learning: the art and science of algorithms that make sense of data. New York: Cambridge University Press, 2012

BZDOK, Danilo; KRZYWINSKI, Martin; ALTMAN, Naomi. Points of significance: Machine learning: supervised methods. 2018.

JI, Haijin, et al. Empirical Studies of a Kernel Density Estimation Based Naive Bayes Method for Software Defect Prediction. En: IEICE TRANSACTIONS on Information and Systems, 2019, vol. 102, no 1, p. 75-84.

ABID, Faroudja; HAMAMI, Latifa. A survey of neural network based automated systems for human chromosome classification. En: Artificial Intelligence Review, 2018, vol. 49, no 1, p. 41-56.

SMITH, Baxter. Lagrange multipliers tutorial in the context of support vector machines. En: Memorial University of Newfoundland St. John's, Newfoundland, Canada, 2004, p. 17.

BAHLMANN, Claus; HAASDONK, Bernard; BURKHARDT, Hans. Online handwriting recognition with support vector machines-a kernel approach. En: Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition. IEEE, 2002. p. 49-54.

LIU, Yi; ZHENG, Yuan F. One-against-all multi-class SVM classification using reliability measures. En: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005. IEEE, 2005. p. 849-854

PLÖTZ, Thomas; FINK, Gernot A. Markov models for offline handwriting recognition: a survey. En: International Journal on Document Analysis and Recognition (IJDAR), 2009, vol. 12, no 4, p. 269.

WANG, Xin; BI, Jinbo. Bi-convex optimization to learn classifiers from multiple biomedical annotations. En: IEEE/ACM transactions on computational biology and bioinformatics, 2016, vol. 14, no 3, p. 564-575.

TORKKOLA, Kari; TUV, Eugene. Ensemble learning with supervised kernels. En: European Conference on Machine Learning. Springer, Berlin, Heidelberg, 2005. p. 400-411

ZHANG, Jing; WU, Xindong; SHENG, Victor S. Learning from crowdsourced labeled data: a survey. En: Artificial Intelligence Review, 2016, vol. 46, no 4, p. 543-576.

GUAN, Melody Y., et al. Who said what: Modeling individual labelers improves classification. En: Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

DASH, Manoranjan; LIU, Huan. Feature selection for classification. En: Intelligent data analysis, 1997, vol. 1, no 1-4, p. 131-156.

AMARI, Shun-ichi; WU, Si. Improving support vector machine classifiers by modifying kernel functions. En: Neural Networks, 1999, vol. 12, no 6, p. 783-789.

HE, Xiaofei; CAI, Deng; NIYOGI, Partha. Laplacian score for feature selection. En: Advances in neural information processing systems. 2006. p. 507-514.