

ANÁLISIS DEL COMPORTAMIENTO DEL ALGORITMO SVM PARA DIFERENTES KERNEL EN AMBIENTES CONTROLADOS

J. D. L. CABRERA^{1,*}, A. PEREIRA-TOLEDO²

¹Universidad Central "Marta Abreu" de Las Villas, ²Universidad de Sancti Spiritus "José Martí Pérez"
josedaniellc@uclv.cu *

Recepción 17/01/2017 – Aprobación 18/10/2018

DOI: 10.15628/holos.2018.5563

RESUMEN

En el presente trabajo se realiza una investigación del comportamiento de la técnica de aprendizaje automático SVM en diferentes ambientes controlados, usando cinco *kernel*. Primero se analiza el comportamiento de los clasificadores ante datos con valores perdidos. Luego se prueban en ambientes con valores ruidosos. La última tarea es el análisis de su

comportamiento una vez que se adicionan atributos irrelevantes. Para validar los resultados se realizan pruebas estadísticas no paramétricas. Se encontró que la técnica SVM es muy robusta ante los tres ambientes antes mencionados y el *kernel* polinómico arrojó los mejores resultados de clasificación.

PALABRAS CLAVE: aprendizaje automático, SVM, *kernel*, estudio experimental.

A PERFORMANCE ANALYSIS OF THE SVM TECHNIQUE FOR DIFFERENT KERNELS

ABSTRACT

This paper reports the results of a series of experiments on the machine learning technique known as SVM, for the comparison of five different kernels. First, we compare and analyze their classification performance on databases with missing values. Then, we run a similar

experiment with noisy values databases. Finally, we analyze the results from adding irrelevant features to the databases. The SVM technique turned out to be very robust in the three created environments, and the polynomial kernel showed the best classification results.

KEYWORDS: machine learning, SVM, kernel, experimental study.

1 INTRODUCCIÓN

Las técnicas de aprendizaje automático en la actualidad se utilizan en muchas ramas de la investigación (Rao & Govindaraju, 2013), siendo una de las tareas iniciales realizar un estudio experimental sobre el comportamiento de algunos de estos métodos en los problemas particulares que se desea resolver. Dentro de las técnicas más populares y de mejores resultados en el aprendizaje automático reportada en la literatura está la conocida como máquina de soporte vectorial (*support vector machine*, SVM) (Fernández-Delgado, Cernadas, Barro, & Amorim, 2014; Wu et al., 2008). Si se usa como palabra de búsqueda *Support Vector Machines* en el Google Scholar, aparecen aproximadamente 62 600 artículos relacionados solamente desde el 2016, lo que brinda un aproximado de la popularidad de esta técnica. No obstante, algunos investigadores tienden a tratar este algoritmo de clasificación en problemas en los cuales no tienen un adecuado comportamiento, de ahí que en ocasiones los resultados obtenidos no estén en concordancia con sus verdaderas potencialidades. Por tanto, conocer su comportamiento en determinados ambientes es de vital importancia para identificar sus potencialidades y fallos.

Uno de los parámetros que se suele variar en los SVM es su función *kernel*, la que hace una transformación de los datos de entrada a otro espacio de mayor dimensión como se explica en la sección 3. Los estudios experimentales reportados en la literatura han estado orientados principalmente a comparar diferentes familias de clasificadores (Chorowski, Wang, & Zurada, 2014) en lugar de comparar diferentes *kernel* de un SVM. Además, se seleccionan en la investigación bases de datos que permiten determinar su comportamiento en distintos ambientes controlados. En las Tabla 1 Tabla 2 y Tabla 3 en la sección 3 se describen los conjuntos de datos usados los cuales contienen: datos ruidosos, valores nulos y atributos irrelevantes. Por tanto, un estudio experimental sobre el desempeño de una SVM con varios *kernel* en ambientes controlados ayudaría a dilucidar el comportamiento de esta técnica de clasificación, como se aborda en las secciones 4.1, 4.2 y 4.3. Con el objetivo de validar los resultados obtenidos se usará una modificación de la metodología que se propone en (Demsar, 2006), la cual ha sido muy influyente en los estudios sobre comparación de múltiples algoritmos con múltiples bases de datos

2 REVISIÓN BIBLIOGRÁFICA

La literatura revisada sobre comparaciones de diferentes SVM, comprende un período de tiempo de más de 20 años, que va desde 1995 hasta 2017. En tal período se encontraron 12 artículos que contienen estudios experimentales en los que se compara alguna variante de SVM con otros algoritmos, o entre variantes de SVM (Amari & Wu, 1999; Chorowski et al., 2014; Cortes & Vapnik, 1995; Fernández-Delgado et al., 2014; Glick, Jenkins, Nettles, Hitchings, & Davies, 2006; Hsu & Lin, 2002; Jayadeva, Khemchandani, & Chandra, 2007; Lee & Mangasarian, 2001; Mangasarian & Kou, 2007; Meyer, Leisch, & Hornik, 2003; Scholkopf et al., 1997; Weston & Watkins, 1998). En 8 de ellos se comparan variantes de SVM surgidas de modificaciones de diferente naturaleza a los algoritmos seleccionados, mientras que menos del 50% de ellos se dedica específicamente a la comparación de variantes de *kernel*. En otros cinco artículos, la comparación siempre se realiza entre SVM y clasificadores de otras familias. En solo uno de ellos se compara tanto entre *kernel* como con otras familias, y es también el único que usa la

metodología estadística para evaluar la comparación propuesta por (Demsar, 2006). En cuanto a la selección de los datos de entrenamiento y prueba, tres artículos confirman el uso de bases de datos orientadas a un dominio de aplicación específico, inclusive hasta algunos muestran los resultados a partir de una sola base de datos. El resto usa múltiples bases de datos artificiales o reales, con distintos dominios de aplicación.

Por ello, la realización de un estudio experimental para la comparación de varios *kernel* de SVM en la que la metodología de evaluación es la propuesta por (Demsar, 2006), y en la que se busque la capacidad de generalización ante problemas diferentes; no es solo una oportunidad, sino una necesidad para las investigaciones en comunidad de Aprendizaje Automático.

3 MATERIALES Y MÉTODOS

La idea de SVM está basada en la teoría de (Vapnik & Chervonenkis, 1964), pero se publica por primera vez en (Cortes & Vapnik, 1995), donde se demuestra la alta capacidad de generalización utilizando transformaciones de entrada polinomiales y se compara con otras técnicas clásicas de aprendizaje automático. A partir de un conjunto de muestras de entrenamiento, cada una con su clase asociada, el SVM se entrena para construir un modelo que asigne un nuevo ejemplo dentro de una categoría u otra, por lo cual se considera como una técnica de aprendizaje supervisado. La idea central del método es encontrar un hiperplano de separación óptima entre dos clases a través de la maximización de los márgenes entre las clases. Los puntos en las fronteras de las clases que sirven para formar los márgenes, se conocen como vectores de soporte. En el centro de los márgenes se encuentra el hiperplano de separación óptimo, ver Figura 1. Los SVM usan el modelo matemático $y = mx + \gamma$ que se manipula para permitir divisiones en el dominio lineal. Esta técnica puede ser dividida en dos modelos: lineales y no-lineales (Hastie, Tibshirani, Friedman, & Franklin, 2005). Se llaman entonces SVM lineales si el dominio de los datos puede ser dividido linealmente (una línea recta, un hiperplano) en el dominio original. En el caso de que los datos no sean linealmente separables y deban ser transformados hacia un espacio llamado espacio de rasgos, donde el dominio de los datos es divisible linealmente para separar las clases, entonces este es llamado SVM no lineal (Schölkopf et al., 1999). Matemáticamente cuando se habla de SVM lineales se usa la ecuación lineal $y = wx^T + \gamma$, mientras que para los SVM no lineales se usa la ecuación no lineal $y = w\theta x^T + \gamma$ donde la función *kernel* es no lineal. Las técnicas de clasificación usando SVM incluyen la parametrización y la optimización de objetivos. Estos objetivos son principalmente dependientes de la estructura topológica del dominio de los datos. Es decir, las clases pueden ser linealmente separables o linealmente no separables. Los SVM se desarrollaron para clasificación binaria (CORTES; VAPNIK, 1995). No obstante, en problemas reales muchas veces se requiere discriminar en más de dos categorías. Lo más conveniente para abordar problemas multiclases es usar varios clasificadores SVM (DUAN; KEERTHI, 2005). Una técnica común es descomponer los problemas en una serie de problemas binarios tal que puedan ser aplicados los SVM convencionales (WANG; XUE, 2014).

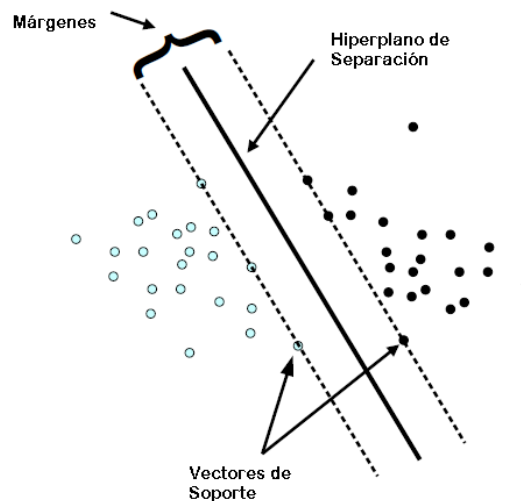


Figura 1. Ejemplo de clasificación usando un SVM donde los datos son linealmente separables

3.1 Descripción de las bases de datos y los parámetros de los clasificadores

A diferencia del estudio comparativo realizado en (Fernández-Delgado et al., 2014), aquí se analiza el comportamiento de la familia de clasificadores SVM en diferentes ambientes. Primeramente, se analizará el comportamiento de los clasificadores usando bases de datos que contienen valores perdidos. Los clasificadores evaluados no están orientados a trabajar con datos con valores perdidos, por tanto, estos serán tratados como cero, lo que no debe sesgar el resultado de la comparación (Fernández-Delgado et al., 2014). Otra de las pruebas realizadas está enfocada a determinar el comportamiento en ambientes ruidosos. Se analizará además el desempeño usando bases de datos que no contienen atributos irrelevantes y se observará cómo se comportarán los clasificadores tras la adición gradual de atributos irrelevantes.

Las bases de datos seleccionadas en todos los casos se limitan a problemas de clasificación binarios. El número de muestras en estas bases de datos es diferente entre una clase y otra, por lo que se consideran desbalanceadas. No obstante, el problema del desbalance se comienza a tratar cuando la relación entre clases es mayor que el 90 %, esto se conoce como desbalance crítico (García & Herrera, 2009). En estos casos un clasificador estándar tiende a aprender más de la clase mayoritaria y a ignorar la clase minoritaria. En los SVM los vectores de soporte se construyen solo para los casos de las clases que se encuentran en la frontera. Los casos que están lejos de la frontera pueden ser eliminados sin afectar mucho la calidad de la clasificación siendo bastante exacta en datos con cierto desbalance (Wang & Xue, 2014). No obstante, los SVM son sensitivos en el caso de que el desbalance sea grande, apareciendo un sesgo hacia la clase mayoritaria y bajando la calidad de las medidas de desempeño en la clase minoritaria (Tang, Zhang, Chawla, & Krasser, 2009).

Otro aspecto importante tenido en cuenta para seleccionar las bases de datos fue la relación entre el número de atributos y el número de casos. En (Foster, Koprowski, & Skufca, 2014) se realiza un llamado a sensibilizarse con las debilidades y engaños que acarrea en los algoritmos de clasificación no tener una adecuada relación casos/atributos, donde se plantea que el número adecuado de esta relación debe ser como mínimo 10. Esto no se cumple en todas las bases de datos disponibles en (Bache & Lichman, 2013). Por tanto, se seleccionan solo aquellas bases de datos donde existan al menos 10 casos por rasgos.

En la Tabla 1 se muestra la descripción de las bases de datos seleccionadas que contienen valores perdidos (VP). El tipo de atributos puede ser nominal (No) o numérico (Nu). En la tabla se observa además el porcentaje de la clase mayoritaria (%CM), observándose que en ningún caso es superior a 90 %.

Tabla 1 Descripción de las bases de datos usadas en la tarea de clasificación con valores perdidos

Base de Casos	#atri(Nu/No)	# casos	#clases	% VP	#Casos por clase	% CM
credit-a	15 (6/9)	690	2	0.53	307/383	55.36
wisconsin breast cancer	9 (9/0)	699	2	0.25	458/241	65.52
bands (cylinder bands)	39(18/21)	540	2	4.74	228/312	57.77
breast	9 (0/9)	286	2	0.34	201/85	70.27
Housevotes	16 (0/16)	435	2	5.63	267/168	61.37
mammographic	5 (5/0)	960	2	3.75	516/444	53.75
Horse Colic	22(15/7)	368	2	22.78	232/136	63.04
Chronic Kidney Disease	24 (13/11)	400	2	10.42	250/150	62.5
Hepatitis	19 (6/13)	155	2	5.67	32/123	79.35

Tabla 2 Descripción de las bases de datos donde los ejemplos de entrenamiento contienen un 15 % y un 20 % de ruido.

Base de casos	#atri(Nu/No)	# casos	#clases	VP	%rui train	AI	#Casos por clase	% CM
heart-15cn01-nc	13 (13/0)	270	2	no	15	no	137/133	50.74
heart-20cn01-nc	13 (13/0)	270	2	no	20	no	137/133	50.74
ionosphere-15cn01-nc	33 (33/0)	351	2	no	15	no	189/162	53.85
ionosphere-20cn01-nc	33 (33/0)	351	2	no	20	no	189/162	53.85
pima-15cn01-nc	8 (8/0)	768	2	no	15	no	420/348	54.69
pima-20cn01-nc	8 (8/0)	768	2	no	20	no	420/348	54.69
sonar-15cn10-nc	60 (60/0)	208	2	no	15	no	111/97	53.37
sonar-20cn10-nc	60 (60/0)	208	2	no	20	no	111/97	53.37
wdbc-15cn10-nc	30 (30/0)	569	2	no	15	no	301/268	52.90
wdbc-20cn10-nc	30 (30/0)	569	2	no	20	no	301/268	52.90

La tarea de clasificación estándar consiste en hacer generalizaciones de un conjunto de ejemplos de entrenamiento. El conocimiento aprendido desde este conjunto puede ser aplicado sobre ejemplos no observados para predecir sus clases. Si el ruido afecta a los atributos o a las clases del conjunto de entrenamiento es más difícil hacer una acertada predicción de las clases sobre los nuevos ejemplos. Las bases de datos utilizadas para evaluar la calidad de los clasificadores con casos ruidosos se extrajeron de (Alcalá et al., 2010). El esquema que se adoptó para introducir ruido a las bases de datos es el propuesto por Zhu en (Zhu, Wu, & Chen, 2003). Se seleccionaron cinco bases de datos de las que se muestra su descripción en la Tabla 2. Las bases de datos contienen cinco particiones listas para ser usadas en tareas de clasificación usando validación cruzada con $k = 5$. Se tienen cinco conjuntos de entrenamiento y cinco de prueba; y existen cuatro bases asociadas, pues se afectaron las bases de datos con diferentes niveles de ruido (5 %, 10 %, 15% y 20 %). Para este estudio se seleccionan las bases que tienen 15% y 20% de ruido. Los conjuntos de prueba no contienen ruido. Esto sigue el enfoque de entrenar con los datos ruidosos y realizar la etapa de prueba con datos limpios. La forma de manejarlo fue usando el *Knowledgeflow*, herramienta gráfica de WEKA inspirada en los diagramas de flujo que aumenta las posibilidades de análisis y procesamiento de datos.

Tabla 3 Descripción de las bases de datos que fueron contaminadas con atributos irrelevantes

Base de casos	#atts(Nu/No)	#casos	#clases	VP	CR	AI	#Casos por clases	%CM
bupa	6(6/0)	345	2	no	no	no	200/145	57.97
haberman	3(3/0)	306	2	no	no	no	225/81	73.53
heart (heart-statlog)	13(13/0)	270	2	no	no	no	150/120	55.56
Iris	4(4/0)	100	2	no	no	no	50/50	50
pima	8(8/0)	786	2	no	no	no	500/286	63.61
saheart	9(8/1)	462	2	no	no	no	302/160	65.37
tic-tac-toe	9(0/9)	768	2	no	no	no	626/142	81.51
wdbc	30(30/0)	569	2	no	no	no	357/212	62.74
wine	13(13/0)	119	2	no	no	no	71/48	59.66

En la Tabla 3 se muestra la descripción de las bases de datos originales, las cuales serán contaminadas con atributos irrelevantes. La primera prueba se realiza con las bases de datos originales, luego se adicionan de cinco en cinco atributos numéricos generados aleatoriamente con una distribución normal en el intervalo [0,1]. Estas bases de datos no contienen casos ruidosos (CR), ni atributos irrelevantes (AI), tampoco valores perdidos (VP). El porcentaje de la clase mayoritaria (%CM) para ninguna de las bases de datos está por encima del 90 %.

La plataforma de software para el aprendizaje automático usada fue WEKA (Cunningham & Denize, 1994). La implementación del clasificador SVM que está disponible en WEKA es la propuesta en (Platt, 1999). Esta implementación reemplaza todos los valores perdidos y transforma los atributos nominales en binarios. Además, normaliza todos los atributos por defecto (en este caso los coeficientes en la entrada están basados en los datos normalizados, no sobre los originales lo que es importante para interpretar el clasificador). Por otra parte, los problemas multiclases se resuelven usando la clasificación por pares uno contra uno. Para obtener un estimado adecuado de probabilidad se usa la opción que ajusta el modelo de regresión lineal en la salida del SVM. En el caso de multiclases para estimar las probabilidades se usa el método de acople por pares descrito en (Hastie, Tibshirani, & others, 1998). Además, se usa la biblioteca libSVM (Chang & Lin, 2012) y se establece el parámetro “tipo” a nu-SVM y el valor por defecto para el resto. Se utilizaron los cuatro *kernel* disponibles. Esto nos proporciona cinco clasificadores que quedan listados en la Tabla 4.

Tabla 4 Relación de clasificadores usados en el experimento, contienen cinco funciones *kernel* y dos bibliotecas

<i>kernel</i>	biblioteca	Ecuación
lineal	libsvm	$K(x_i, x_j) = x_i^T x_j$
polinomial	libsvm	$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
rbf	libsvm	$K(x_i, x_j) = e^{-\gamma x_j - x_i ^2}, \gamma > 0$
sigmoide	libsvm	$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$
pilinomial normalizado	smo	$K(x_i, x_j) = \frac{(\gamma x_i^T x_j + r)^d}{\sqrt{(\gamma x_i^T x_j + r)^d (\gamma x_j^T x_i + r)^d}}, \gamma > 0$

En este trabajo no se usan técnicas de pre-procesamiento, transformación de los datos ni selección de atributos. El objetivo no es encontrar el mejor comportamiento de los clasificadores para cada base de datos, sino comparar el desempeño de los clasificadores de forma general sobre las bases de datos usadas.

3.2 Análisis estadístico

El problema de comparar múltiples algoritmos en múltiples bases de datos, es común en la rama de aprendizaje automático. Varias son las publicaciones que abordan sobre el tema y proponen diferentes técnicas estadísticas para tratarlo (Demsar, 2006; Dietterich, 1998; García, Fernández, Luengo, & Herrera, 2010; García & Herrera, 2008; Santafe, Inza, & Lozano, 2015). En este trabajo se usa el lenguaje R, con el paquete SCMAMP (Calvo & Santafé, 2016) para aplicar las pruebas estadísticas. La metodología seguida se explica a continuación.

Según la metodología aplicada, primero se aplica el omnibus test. Se propone en la literatura el test de Friedman con la extensión de Iman y Davempport cuando se comparan más de cinco algoritmos. En el caso que ocupa, con solo cinco algoritmos, se propone la utilización de la prueba de los rangos alineados de Friedman o la prueba de Quade (García et al., 2010), siendo seleccionada la primera por ser más potente. Si existiesen diferencias significativas, es decir, el valor p (p value) es menor que 0.05, entonces podría decirse que al menos uno de los algoritmos se comporta diferente. Para saber cuál y si su comportamiento es mejor o peor, se usa una prueba post hoc. El test post hoc que se selecciona es el método de Finner, que en general es una buena elección debido a su simplicidad y potencia.

En cuanto a la evaluación del desempeño de los algoritmos, se usa la técnica de validación cruzada, la cual divide las n muestras a clasificar en k grupos ordenados aleatoriamente. Se utilizan $k-1$ grupos para entrenamiento y el grupo restante para la evaluación. Con este proceso siempre se toma un subconjunto diferente para la evaluación hasta clasificar todos los k . Esta técnica es simple y garantiza conocer el grado de generalización adquirido por los clasificadores. En el caso de los datos ruidosos se usa $k = 5$, ya que los datos estaban particionados de esa forma. Para entrenar se usan los ruidosos y para clasificar, los limpios. Todo ello se realizó a través del Knowledgeflow, mientras que en los otros experimentos se usó el *Experimenter* de WEKA, esta vez con $k=10$ y repitiendo el experimento 10 veces.

El AUC (*Area Under Curve*) fue la métrica seleccionada para cuantificar el comportamiento de los clasificadores. Se deriva del cálculo del área bajo la curva ROC (*Receiver Operating Characteristics*), la cual es resultado de trazar la tasa de verdaderos positivos contra la tasa de falsos negativos. Diferentes estudios han usado el AUC como métrica para conocer el desempeño de los clasificadores y demostrado sus ventajas (Bradley, 1997; Demsar, 2006), aunque otros han señalado sus deficiencias (Lobo, Jiménez-Valverde, & Real, 2008). Esta medida puede ser engañosa principalmente para conjuntos de datos desbalanceados donde la clase minoritaria representa menos del 10% del conjunto de datos. En ninguna de las bases de datos seleccionadas se tiene un desbalance crítico.

4 RESULTADOS Y DISCUSIÓN

4.1 Datos Ruidosos

El primer experimento está basado en el conjunto de datos ruidosos, siguiendo el esquema de entrenar con los datos ruidosos y probar con los no contaminados. La Figura 2 muestra el comportamiento de los clasificadores para las 5 bases de datos con 15% y 20% de ruido. La Figura 2a muestra en el eje de las abscisas cada una de las bases de datos, donde el 1er número hace referencia a la base de datos y luego está acompañado del porcentaje de ruido que contiene. Cada una de las barras se asocia a un clasificador como se muestra en la leyenda. Se observa que el comportamiento de los clasificadores para 15% de ruido es muy similar que para el 20% (ver Figura 2a). Esto se justifica porque los vectores de soporte están asociados a los casos que están en la frontera de cada clase y una vez que se modifica el *kernel*, se desplaza la disposición espacial de los rasgos y por tanto es otro conjunto de datos el que se intenta resolver. Pero es meritorio destacar que se mantiene prácticamente constante el comportamiento cuando se aumenta del 15% al 20% de ruido en los conjuntos de entrenamiento. Por otro lado, la Figura 2b muestra el resultado de aplicar el test de Finner como prueba post hoc, pues el *p-value* resultó ser menor que 0.05 luego de aplicar la prueba de los rangos alineados de Friedman, evidenciando que existen diferencias significativas entre los clasificadores usados. Los resultados se encuentran ordenados de forma creciente, siendo el *kernel* polinomial normalizado el de mejor posición de forma general. Las líneas horizontales que se aprecian en esta figura representan los *kernel* que tuvieron un comportamiento similar de acuerdo al AUC, es decir, no existen diferencias estadísticamente significativas. Se observa que no existen diferencias significativas entre los *kernel* polinomial normalizado, polinomial, lineal y rbf como tampoco existen diferencias entre el sigmoide y el rbf. Los de mejor comportamiento aparecen agrupados más a la izquierda.

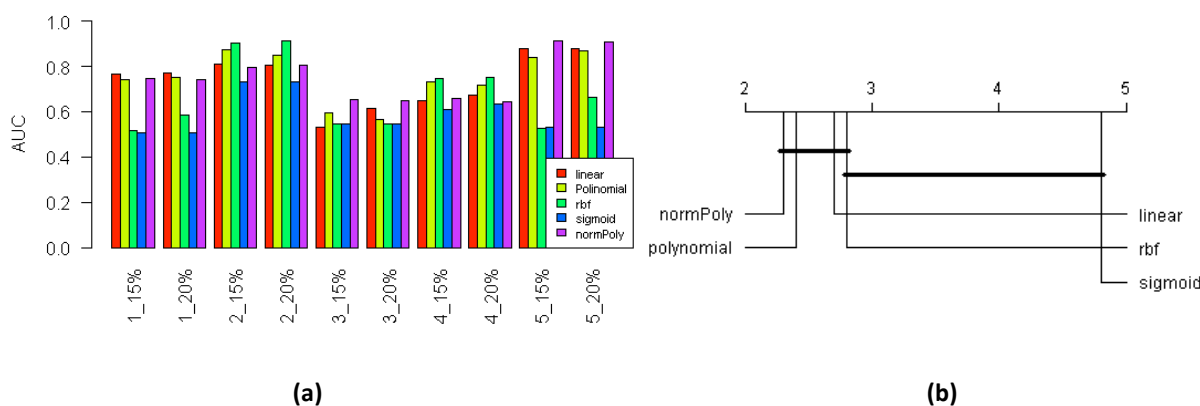


Figura 2. Comportamiento de los cinco clasificadores para datos ruidosos. En (a) se observan las 10 bases de datos usadas, y en las ordenadas el área bajo la curva ROC (AUC). En (b) se muestra el resultado de aplicar el test de Finner.

4.2 Valores nulos

Un análisis similar al de la sección anterior se realizó para el caso de los valores nulos. En este caso se obtuvo un *p-value* = 0.96, que representa la no existencia de diferencias significativas entre los algoritmos de clasificación evaluados. La gráfica del comportamiento se muestra en la Figura 3, donde no se observa un claro ganador entre los algoritmos que compiten.

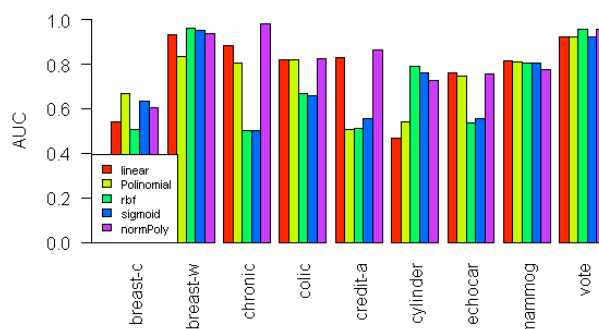


Figura 3. Comportamiento de los cinco clasificadores en las bases de datos con valores nulos. En las abscisas las nueve bases de datos y en las ordenadas el área bajo la curva ROC (AUC).

4.3 Atributos irrelevantes

Similar a los análisis anteriormente realizados la siguiente prueba está orientada a observar el comportamiento de los clasificadores una vez que se adicionan atributos irrelevantes. La prueba tiene cuatro etapas, en la primera se realiza la clasificación usando las bases de datos originales, luego se comienzan a agregar atributos irrelevantes de cinco en cinco, como se explicó en la sección 3.1.

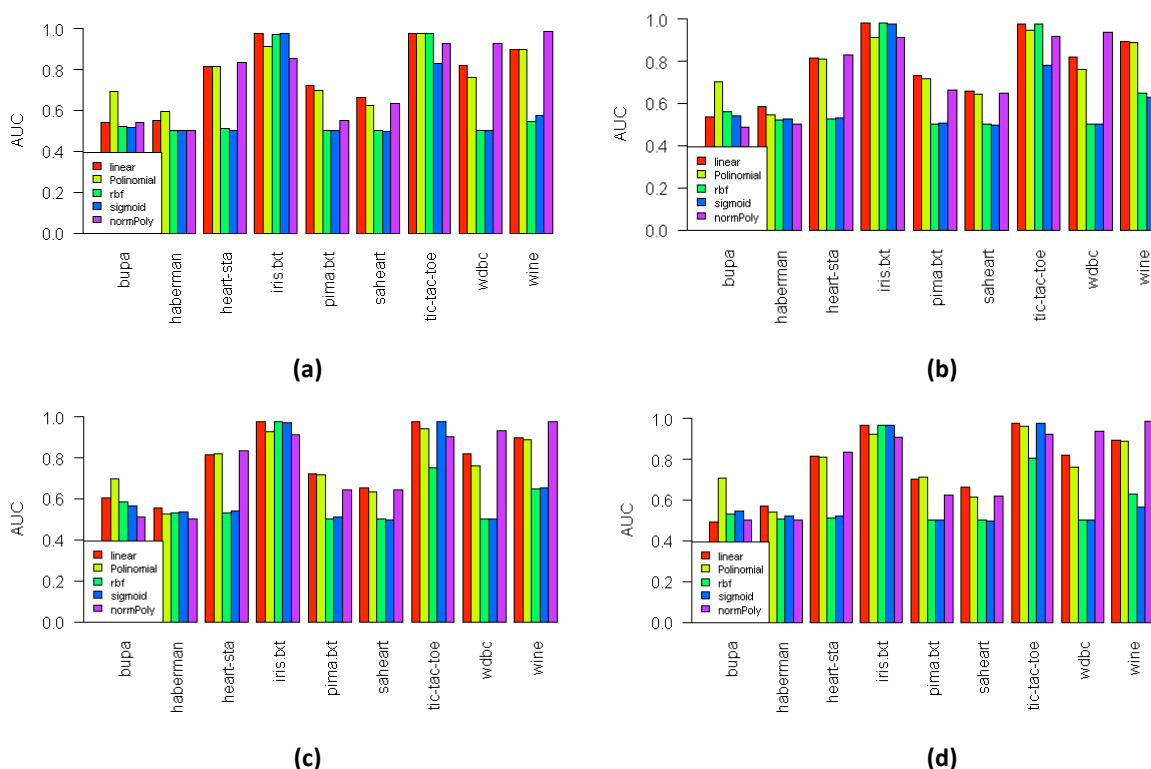


Figura 4. Comportamiento de los cinco clasificadores. En (a) las bases de datos originales, en (b) luego de adicionar cinco atributos irrelevantes, en (c) luego de adicionar diez atributos irrelevantes y en (d) luego de adicionar quince atributos irrelevantes.

En la Figura 4a se muestra el comportamiento de los cinco clasificadores para las nueve bases de datos originales, lo que refleja una visión de cómo es su comportamiento antes de ser contaminados por los atributos irrelevantes. Luego de aplicar el test de los rangos alineados de Friedman, se obtuvo un $p\text{-value} = 0.0004843$, lo cual indica que existen diferencias entre los

algoritmos. En la Figura 5a se observa el resultado del test pos hoc donde los clasificadores SVM con *kernel* lineal, polinomial y polinomial normalizado no tienen diferencias significativas entre ellos y se comportan mejor que el resto. Tampoco existen diferencias entre el rbf y el sigmoide, los cuales son los de peor comportamiento.

Un análisis similar para el caso de las bases de datos luego de adicionar cinco atributos irrelevantes arroja un $p\text{-value} = 0.001424$. El comportamiento de los clasificadores se observa en Figura 4b. Por otro lado, el resultado del test post hoc se muestra en Figura 5b donde se observa que los SVM con *kernel* lineal, polinomial y polinomial normalizado no tienen diferencias estadísticamente significativas, pero se comportan mejor que el rbf y el sigmoide. Estos últimos no presentan diferencias significativas entre ellos, pero sí con los tres primeros.

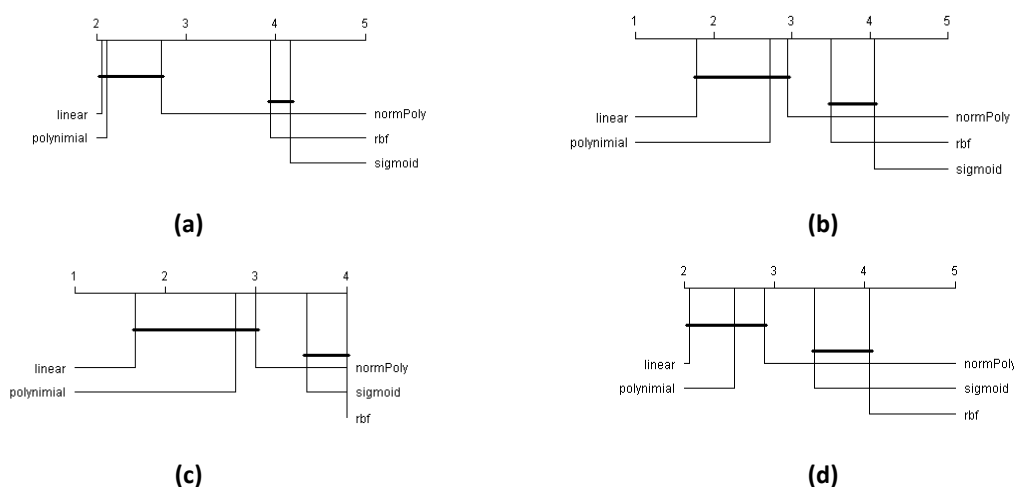


Figura 5. Gráfico basado en el test de Finner. En (a) usando las bases de datos originales, en (b) luego de adicionar cinco atributos irrelevantes, en (c) luego de aplicar diez atributos irrelevantes y en (d) luego de aplicar quince atributos irrelevantes.

Cuando se analizan las bases de datos luego de adicionar diez atributos irrelevantes, se obtiene un $p\text{-value} = 0.001292$, igualmente se concluye que existen diferencias entre algunos de los algoritmos evaluados y se realiza el test *post hoc*. Los resultados son similares a los obtenidos para los casos anteriores (ver Figura 4c y Figura 5c), los *kernel* lineal, polinomial y polinomial normalizado son los de mejores resultados en ese orden, aunque no existen diferencias estadísticamente significativas entre ellos pero si entre estos tres y los kernels sigmoide y rbf.

Los resultados obtenidos para el caso de las bases luego de adicionar 15 atributos irrelevantes son similares a los anteriores tres análisis. Se obtiene un nivel de significación estadística menor que 0.05, es decir, un $p\text{-value} = 0.001431$. En la Figura 4d se muestra el comportamiento para cada una de las bases de datos en base al AUC obtenido para cada *kernel*. Por otro lado, la Figura 5d muestra el resultado del test de Finner. Se observa que igualmente los kernel lineal, polinomial y polinomial normalizado son los de mejores desempeños en ese orden y que no existen diferencias significativas entre estos tres. No obstante, si existen diferencias entre estos y los *kernel* sigmoide y rbf.

Otro análisis interesante que muestra el desempeño de los *kernel* estudiados se observa en la Figura 6. Se aprecia en dicha figura el comportamiento en el tiempo para cada uno de los *kernel* en las cuatro etapas. Se aprecia que el comportamiento es muy similar observando cada

kernel desde la base de datos sin atributos irrelevante y una vez se comienzan a adicionar estos. Se evidencia, por tanto, muy buena robustez ante atributos irrelevantes por parte de los SVM.

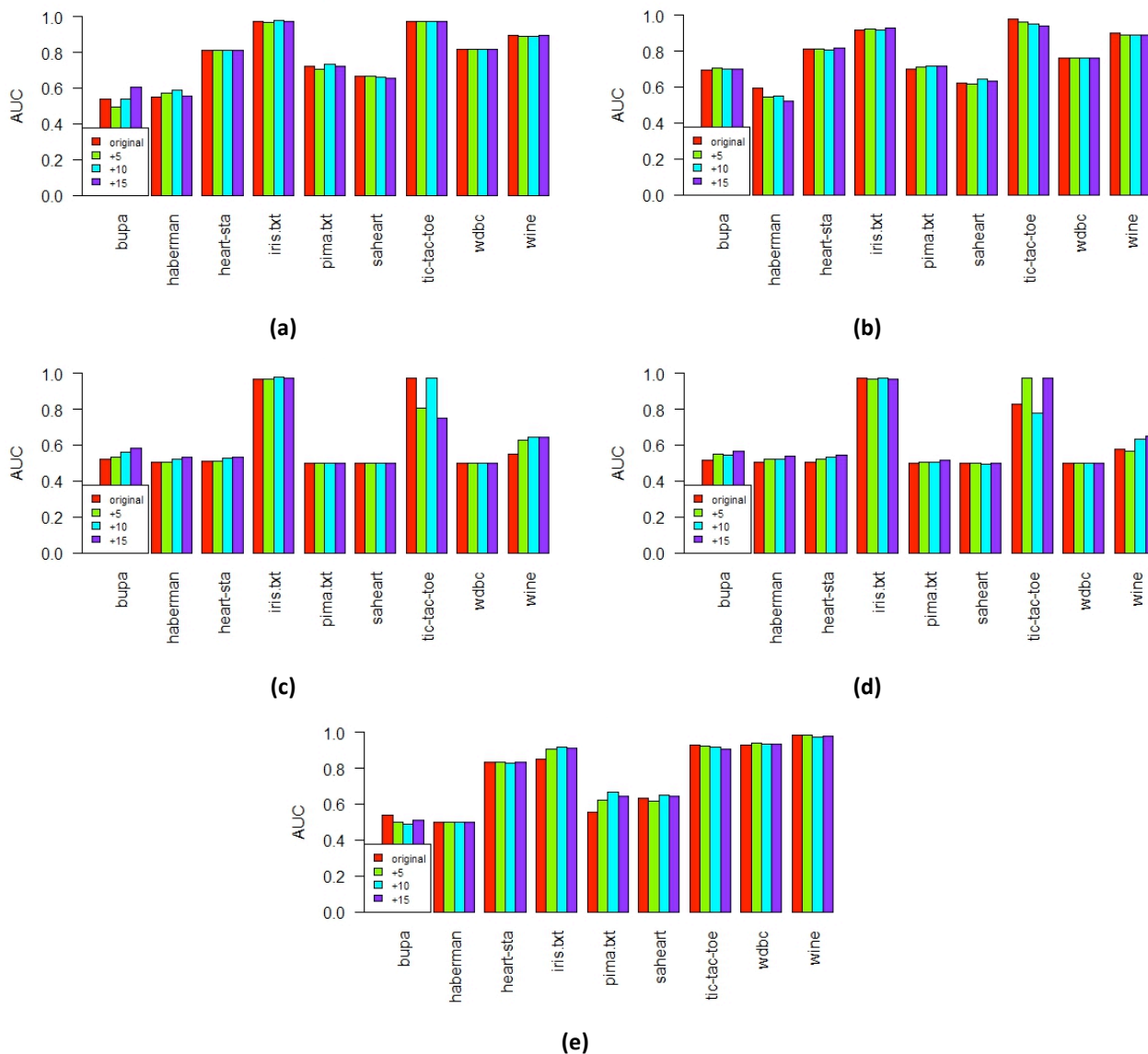


Figura 6. Comportamiento histórico de los clasificadores, en las abscisas las cuatro versiones de las bases de datos usadas, en las ordenadas el AUC. En (a) el kernel lineal, en (b) el kernel polinomial, en (c) el kernel rbf, en (d) el kernel sigmoide y en (e) el kernel polinomial normalizado.

5 CONCLUSIONES

En este trabajo se realizó una evaluación del comportamiento de clasificadores pertenecientes a la familia de los SVM usando cinco *kernel* diferentes. La evaluación se realizó en ambientes controlados, los conjuntos de datos usados contenían casos ruidosos, atributos irrelevantes y valores nulos.

Los resultados obtenidos muestran que los *kernel* analizados toleran de manera similar los niveles de ruido generados, siendo el *kernel* sigmoide el de peor comportamiento. En cuanto al desempeño en las bases de datos con valores nulos, no existen diferencias significativas entre ninguno de los *kernel*. Además, se evidenció gran robustez por parte de todos los *kernel* ante la presencia de atributos irrelevantes mostrando fluctuaciones mínimas en su medida de desempeño. En este último experimento los *kernel* sigmoide y rbf mostraron diferencias significativas con respecto al resto, mostrando un peor comportamiento pereira.

6 REFERENCIAS

- Alcalá, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., & Herrera, F. (2010). *Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework*. *Journal of Multiple-Valued Logic and Soft Computing*, 17(255–287), 11.
- Amari, S., & Wu, S. (1999). *Improving support vector machine classifiers by modifying kernel functions*. *Neural Networks*, 12(6), 783–789. [https://doi.org/10.1016/S0893-6080\(99\)00032-5](https://doi.org/10.1016/S0893-6080(99)00032-5)
- Bache, K., & Lichman, M. (2013). *UCI machine learning repository*.
- Bradley, A. P. (1997). *The use of the area under the ROC curve in the evaluation of machine learning algorithms*. *Pattern Recognition*, 30(7), 1145–1159.
- Calvo, B., & Santafé, G. (2016). *scmamp: Statistical Comparison of Multiple Algorithms in Multiple Problems*. *The R Journal*, 8(1), 248–256.
- Chang, C.-C., & Lin, C.-J. (2012). *LIBSVM: a library for support vector machines*. Retrieved from <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- Chorowski, J., Wang, J., & Zurada, J. M. (2014). *Review and performance comparison of SVM-and ELM-based classifiers*. *Neurocomputing*, 128, 507–516.
- Cortes, C., & Vapnik, V. (1995). *Support-vector networks*. *Machine Learning*, 20(3), 273–297.
- Cunningham, S. J., & Denize, P. (1994). *A tool for model generation and knowledge acquisition*. In *Selecting Models from Data* (pp. 471–478). Springer. Retrieved from http://link.springer.com/chapter/10.1007/978-1-4612-2660-4_48

- Demsar, J. (2006). *Statistical Comparisons of Classifiers over Multiple Data Sets*. *J. Mach. Learn. Res.*, 7, 1–30.
- Dietterich, T. G. (1998). *Approximate statistical tests for comparing supervised classification learning algorithms*. *Neural Computation*, 10(7), 1895–1923.
- Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). *Do we need hundreds of classifiers to solve real world classification problems?* *The Journal of Machine Learning Research*, 15(1), 3133–3181.
- Foster, K. R., Koprowski, R., & Skufca, J. D. (2014). *Machine learning, medical diagnosis, and biomedical engineering research-commentary*. *BioMedical Engineering OnLine*, 13(1), 94.
- García, S., Fernández, A., Luengo, J., & Herrera, F. (2010). *Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power*. *Information Sciences*, 180(10), 2044–2064.
- García, S., & Herrera, F. (2008). *An Extension on “Statistical Comparisons of Classifiers over Multiple Data Sets” for all Pairwise Comparisons*. *Journal of Machine Learning Research*, 9(Dec), 2677–2694.
- García, S., & Herrera, F. (2009). *Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy*. *Evolutionary Computation*, 17(3), 275–306.
- Glick, M., Jenkins, J. L., Nettles, J. H., Hitchings, H., & Davies, J. W. (2006). *Enrichment of High-Throughput Screening Data with Increasing Levels of Noise Using Support Vector Machines, Recursive Partitioning, and Laplacian-Modified Naive Bayesian Classifiers*. *Journal of Chemical Information and Modeling*, 46(1), 193–200. <https://doi.org/10.1021/ci050374h>
- Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2005). *The elements of statistical learning: data mining, inference and prediction*. *The Mathematical Intelligencer*, 27(2), 83–85.

- Hastie, T., Tibshirani, R., & others. (1998). *Classification by pairwise coupling*. *The Annals of Statistics*, 26(2), 451–471.
- Hsu, C.-W., & Lin, C.-J. (2002). *A comparison of methods for multiclass support vector machines*. *IEEE Transactions on Neural Networks*, 13(2), 415–425. <https://doi.org/10.1109/72.991427>
- Jayadeva, Khemchandani, R., & Chandra, S. (2007). *Twin support vector machines for pattern classification*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5), 905–910.
- Lee, Y.-J., & Mangasarian, O. L. (2001). *SSVM: A Smooth Support Vector Machine for Classification*. *Computational Optimization and Applications*, 20(1), 5–22. <https://doi.org/10.1023/A:1011215321374>
- Lobo, J. M., Jiménez-Valverde, A., & Real, R. (2008). *AUC: a misleading measure of the performance of predictive distribution models*. *Global Ecology and Biogeography*, 17(2), 145–151.
- Mangasarian, O. L., & Kou, G. (2007). *Feature Selection for Nonlinear Kernel Support Vector Machines*. In *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)* (pp. 231–236). <https://doi.org/10.1109/ICDMW.2007.30>
- Meyer, D., Leisch, F., & Hornik, K. (2003). *The support vector machine under test*. *Neurocomputing*, 55(1–2), 169–186. [https://doi.org/10.1016/S0925-2312\(03\)00431-4](https://doi.org/10.1016/S0925-2312(03)00431-4)
- Platt, J. C. (1999). *12 fast training of support vector machines using sequential minimal optimization*. *Advances in Kernel Methods*, 185–208.
- Rao, C. R., & Govindaraju, V. (2013). *Handbook of Statistics: Machine Learning: Theory and Applications* (Vol. 31). Newnes. Retrieved from <https://www.google.com/books?hl=es&lr=&id=WhqLgjO5HQgC&oi=fnd&pg=PR1&dq=review+machine+learning+applications&ots=nLClo5clrb&sig=aJkc9mgZm0TzZg8Vw6imuXunH-w>

- Santafe, G., Inza, I., & Lozano, J. A. (2015). *Dealing with the evaluation of supervised classification algorithms*. *Artificial Intelligence Review*, 44(4), 467–508.
- Schölkopf, B., Mika, S., Burges, C. J., Knirsch, P., Müller, K.-R., Rätsch, G., & Smola, A. J. (1999). *Input space versus feature space in kernel-based methods*. *Neural Networks, IEEE Transactions On*, 10(5), 1000–1017.
- Scholkopf, B., Sung, K.-K., Burges, C. J. C., Girosi, F., Niyogi, P., Poggio, T., & Vapnik, V. (1997). *Comparing support vector machines with Gaussian kernels to radial basis function classifiers*. *IEEE Transactions on Signal Processing*, 45(11), 2758–2765.
<https://doi.org/10.1109/78.650102>
- Tang, Y., Zhang, Y.-Q., Chawla, N. V., & Krasser, S. (2009). *SVMs modeling for highly imbalanced classification*. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions On*, 39(1), 281–288.
- Vapnik, V., & Chervonenkis, A. (1964). *A note on one class of perceptrons*. *Automation and Remote Control*, 25(1).
- Wang, Z., & Xue, X. (2014). *Multi-class support vector machine*. In *Support Vector Machines Applications* (pp. 23–48). Springer. Retrieved from http://link.springer.com/chapter/10.1007/978-3-319-02300-7_2
- Weston, J., & Watkins, C. (1998). *Multi-class support vector machines*. Citeseer. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.50.9594&rep=rep1&type=pdf>
- Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., ... others. (2008). *Top 10 algorithms in data mining*. *Knowledge and Information Systems*, 14(1), 1–37.
- Zhu, X., Wu, X., & Chen, Q. (2003). *Eliminating class noise in large datasets*. In *ICML (Vol. 3, pp. 920–927)*. Retrieved from <http://www.aai.org/Papers/ICML/2003/ICML03-119.pdf>