UNIVERSIDAD MICHOACANA DE SAN NICOLÁS DE HIDALGO
FACULTAD DE INGENIERA ELÉCTRICA
DIVISIÓN DE ESTUDIOS DE POSGRADOS

COMPARACIÓN DE LOS ALGORITMOS DE LASSO Y LARS EN EL PROBLEMA DE SELECCIÓN DE VARIABLES Y SU APLICACIÓN EN SERIES DE TIEMPO

TESIS

Para obtener el grado de Maestro en Ciencias en Ingeniería Eléctrica

Presenta
Eric Iturbide Díaz

Director de tesis
Dr. Jaime Cerda Jacobo

Co-director de tesis
Dr. Mario Graff Guerrero

Mayo 2013 Morelia, Michoacán.

COMPARACIÓN DE LOS ALGORITMOS DE LASSO Y LARS EN EL PROBLEMA DE SELECCIÓN DE VARIABLES Y SU APLICACIÓN EN SERIES DE TIEMPO

TESIS

Que para obtener el grado de

MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA

presenta

Eric Iturbide Díaz

Jaime Cerda Jacobo

Director de Tesis

Mario Graff Guerrero

Co-Director de Tesis

Universidad Michoacana de San Nicolás de Hidalgo

Mayo 2013

Resumen

En esta tesis, se presentan dos algoritmos de selección de variables; Selección de Operadores y Contracción del Menor Absoluto (LASSO por sus siglas en ingles "Least Absolute Shrinkage and Selection Operator") y Regresión por el Menor Ángulo (LARS por sus siglas en ingles "Least Angle Regression") para predecir series de tiempo. Estas técnicas son aplicadas en el modelo lineal.

En este trabajo, se utiliza validación cruzada para obtener el mejor modelo final. Esta técnica tiene como objetivo encontrar los parámetros idóneos para evitar el bajo-aprendizaje y sobre-aprendizaje. Los resultados muestran que LASSO y LARS tiene un poder predictivo superior o igual que Mínimos Cuadrados Ordinarios en términos de error promedio en el conjunto de validación. Para este fin se utilizaron 4,004 series de tiempo diferentes que fueron tomadas de las competiciones llamadas M1 y M3 de series de tiempo.

Finalmente, es bien sabido que LASSO y LARS se comportan de manera similar, sin embargo, los resultados obtenidos muestran diferencias en la precisión de las predicciones. LARS es el mejor modelo de acuerdo a estos experimentos.

Abstract

In this thesis, we present two algorithms of variable selection; Least Absolute Shrinkage and Selection Operator (LASSO) and Least Angle Regresion (LARS) for forecasting time series. These techniques are applied in the linear model.

In this work, we used cross validation to obtain the best model. It aims to find the ideal number of parameters to avoid under-fitting and over-fitting. The results corroborate that LASSO and LARS obtain better models than Ordinary Least Squares models in terms of mean square error in the validation set. To this end, we used 4,004 different time series taken from the M1 and M3 time series competition.

Finally, it is well known that LASSO and LARS behave similarly; however, the results obtained highlight their differences in terms of forecasting accuracy. The LARS is best model for these experiments.

Contenido

Lista de Figuras

Lista de Tablas

Lista de Algoritmos

Lista de Símbolos

L_q	Normas.
L_1	Norma 1, absolutos.
L_2	Norma 2, cuadrados.
p	Número de variables.
n	Número de observaciones o muestras.
X	Matriz de p variables y n observaciones.
y	Respuesta real.
$\hat{y}$	Respuesta estimada.
r	Residuo $(y - \hat{y})$.
λ	Factor de regularización.
t	Cota superior.
$\|\|$	Valor absoluto.
β	Vector de coeficientes de regresión.
ϵ	Ruido aleatorio.
$M1$	Competencia de series de tiempo con 1,001 series de tiempo diferentes.
$M3$	Competencia de series de tiempo con 3,003 series de tiempo diferentes.
∞	Infinito.
∇	Derivada.
^{T}	Traspuesta.
$\emptyset$	Conjunto vacío.
γ	Dirección equiangular.

Lista de Publicaciones

Aplicación de métodos de selección de variables para la predicción de series de tiempo.
Eric Iturbide, Jaime Cerda Jacobo y Mario Graff.
Organización IEEE. XIV ROPEC 2012. Colima, México. Noviembre 2012.
ISBN: 978-607-95476-6-0

A Comparison between LARS and LASSO for Initialising the Time-Series Forecasting Auto-Regressive Equations.
Eric Iturbide, Jaime Cerda Jacobo and Mario Graff.
CIIECC 2013 Organizing Committee, San Luis Potosi, Mexico.
The 2013 Iberoamerican Conference on Electronics Engineering and Computer Science (CIIECC 2013).

Capítulo 1

Introducción

Este primer capítulo ofrece una descripción general del problema que se resuelve en esta tesis: la predicción de series de tiempo mediante algoritmos que basan su funcionamiento en la selección de variables. Para ello, en primer lugar presenta el problema a tratar que es la predicción de series de tiempo. En segundo lugar, explica la manera de abordar el problema de predicción de series de tiempo mediante la selección de variables aplicando el modelo lineal y en especial los modelos autoregresivos para series de tiempo. Por último, introduce brevemente el contenido de cada uno de los capítulos presentados en este trabajo.

1.1. Predicción de series de tiempo

Hoy en día, la predicción es una actividad crucial en áreas tan diversas como la economía, ingeniería, ciencias biológicas, ciencias de la salud, actividades empresariales y ambientales [Bai98, Andreou02, Aggarwal99, Bardet12, Braun00, Ding08]. En general podemos entender que una predicción es el aviso de algo que va a suceder por lo cual resulta útil comprender el conocimiento anticipado de un hecho futuro, es decir, este conocimiento permitiría prepararse de forma adecuada para hacerle frente a un determinado problema [Ye09].

Por lo general un problema a predecir se puede representar mediante una serie de tiempo, la cual consta de un conjunto de datos numéricos que se obtienen en períodos regulares o secuenciales a través del tiempo, también conocidas como series temporales o series cronológicas [Gershenfeld93]. La unidad de tiempo puede ser: segundo, hora, día, mes, trimestre, semestre, estación del año, año o cualquier período que se pueda considerar de interés. Un ejemplo, podría ser la medición de la velocidad del viento en un intervalo de interés. Con la finalidad de predecir la cantidad de energía que producirá una planta eólica en los siguientes periodos.

Una representación gráfica de una serie de tiempo se muestra en la Figura 1.1.

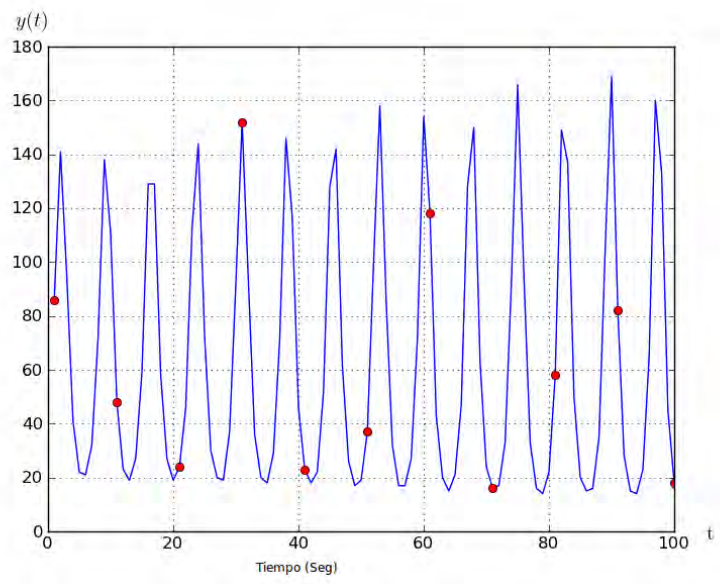


Figura 1.1: Serie de tiempo.

La serie de tiempo de la Figura 1.1, también se puede ver desde el punto de vista de una tabla (ver la Tabla 1.1), de acuerdo a los puntos marcados en la Figura 1.1, el tiempo de interés es cada 10 segundos.

Tabla 1.1: Datos de una serie de tiempo

$y(t)$	86	48	24	152	23	37	118	16	58	82	18
t	1	11	21	31	41	51	61	71	81	91	101

1.2. Descripción del problema

El problema consiste en identificar los valores futuros de algún intervalo de tiempo a partir de una serie de muestras proporcionadas. Por ejemplo, si tomamos como referencia la Figura 1.1 o la Tabla 1.1 es evidente que interesa conocer el valor que tomará $y(t)$ en el segundo 111, 121 y así sucesivamente. En este trabajo se pronosticarán 4,004 series de tiempo diferentes tomadas de las competencia M1 y M3 [Makridakis82, Makridakis00].

Para resolver el problema se parte de los siguientes puntos; primer punto, convertir una serie de tiempo en forma matricial. Segundo punto, utilizar técnicas de selección de variables para obtener la respuesta. Tercer punto, evitar el sobre-aprendizaje o bajo-aprendizaje y por último, comparar las diferentes técnicas para obtener el mejor modelo en términos de precisión. estabilidad y velocidad en cálculos.

1.3. Predicción de series de tiempo mediante la selección de variables

El problema de la selección de variables es especialmente interesante debido a que cada día incrementa el número de problemas que tienen enormes cantidades de información que se desea analizar y para ello es necesario realizar un proceso previo de reducción de datos para poder ser abordado de manera eficiente. La idea principal de este trabajo consiste en resolver el problema de predicción de series de tiempo mediante modelos de selección de variables, donde la elección de las variables necesarias a entrar al modelo sea sencillo y al mismo tiempo exista un balance entre precisión, estabilidad y tiempos de complejidad aceptables.

En el presente trabajo se destaca la contribución de aplicar técnicas de aprendizaje supervisado en especial validación cruzada en conjunto con técnicas de selección de variables que obtienen la solución en tiempo polinomial. Estas técnicas basan su funcionamiento en mínimos cuadrados ordinarios (OLS por sus siglas en ingles "Ordinary Least Squares") [Wikipedia11].

La primer metodología utilizada es la Selección de Operadores y Contracción del Menor Absoluto (LASSO por sus siglas en ingles "Least Absolute Shrinkage and Selection Operator") [Tibshirani94]. La segunda metodología es Regresión por el Menor Ángulo (LARS por sus siglas en ingles "Least Angle Regression") [Efron04].

Como se había mencionado anteriormente (ver 1.2 Descripción del problema) se pronosticarán 4,004 series de tiempo diferentes. Para transformar los datos de las series de tiempo en forma matricial se aplicará el modelo autoregresivo que a continuación se presenta.

1.4. Modelo Autoregresivo

Este modelo es conocido por su sencillez pues depende únicamente de los p valores previos de una serie de tiempo [Tzu-Kuo Huang11]. La formulación general de los modelos autoregresivos tiene la siguiente formulación:

$$y_t = \sum_{i}^{p} \beta_i y_{t-i} + \epsilon \tag{1.1}$$

donde:

- y_t es la serie de tiempo bajo investigación.
- $\{\beta_1, \beta_2, ..., \beta_p\}$ son los coeficientes auto-regresivos a estimar.
- p es el orden del modelo, este valor debe ser mucho menor que la longitud de la serie de tiempo. Es decir p es el primer valor de la respuesta, y sucesivamente los valores $p+1, p+2, ...$, hasta llegar al último valor de la serie de tiempo.

- ϵ es el error aleatorio.

Una vez que se tiene la formulación de una serie de tiempo en forma matricial, lo que es equivalentemente a una formulación lineal por cada observación. Este modelo de regresión lineal a continuación se introduce.

1.5. Modelo de regresión lineal

La regresión lineal es un método de organización de datos. Es una técnica estadística para investigar la relación funcional entre dos o más variables, ajustando algún modelo matemático. A veces es apropiado mostrar los datos como puntos en una gráfica, es decir, tratar de trazar una línea recta a través de los datos. La regresión lineal es un algoritmo para la elaboración de dicha línea [Weisberg05]. Para obtener cada uno de los puntos u observaciones se establece la siguiente ecuación

$$\text{observación} = \text{modelo} + \text{error aleatorio} \tag{1.2}$$

Los modelos de regresión lineal utilizan la ecuación (1.2) dejando el modelo como una función lineal. El objetivo consiste, en generar una respuesta y de la que se obtiene de la siguiente forma [Weisberg05]:

$$y = \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p + \epsilon \tag{1.3}$$

La ecuación (1.3) indica que la respuesta y se genera como una combinación lineal de las variables explicativas mas un error ϵ.

1.6. Objetivos

La presente tesis busca obtener una solución alternativa a los problemas de regresión mediante técnicas de selección de variables.

1. Analizar LASSO y LARS en la predicción de series de tiempo.
2. Presentar un análisis profundo en el problema de selección de variables.
3. Maximizar los resultados con técnicas de aprendizaje supervisado.
4. Obtener el mejor algoritmo para predecir series de tiempo.

1.7. Descripción de Capítulos

El Capítulo 2 presenta los trabajos relacionados con LASSO y LARS resaltando los nuevos avances que se han venido desarrollado con estas técnicas de selección de variables y también se mencionan algunas aplicaciones. Además se presentarán los modelos de selección de variables tradicionales y las técnicas de regularización. El Capítulo 3 presenta los fundamentos de las metodologías basadas en la selección de variables, muestra un análisis profundo de LASSO y LARS. Por otra parte introduce la relevancia que tiene el aprendizaje supervisado en este trabajo. El Capítulo 4 muestra los resultados obtenidos para las 4,004 diferentes series de tiempo. Por último el Capítulo 5 presenta las conclusiones finales de esta tesis y trabajos futuros.

1.8. Resumen

Este capítulo introduce el problema de la predicción de series de tiempo y menciona las técnicas de selección de variables que se utilizarán para pronosticar las 4,004 series de tiempo diferentes. Introduce los modelos autoregresivos que generan las matrices requeridas por las técnicas de selección de variables utilizadas en este trabajo. Además muestra los alcances de este trabajo destacando los objetivos y por último describe el contenido de cada uno de los capítulos.

Capítulo 2

Estado del arte

Este capítulo presenta los nuevos avances relacionados a las diferentes metodologías presentadas en este trabajo (LASSO y LARS). Comienza con una breve revisión de los inicios de la selección de variables, en seguida presenta la técnicas de regularización como una alternativa en el problema de selección de variables. También presenta los principales métodos de selección de variables relacionados con este trabajo. Para finalizar este apartado, menciona algunas aplicaciones de LASSO y LARS en diversas áreas.

2.1. Inicios de la selección de variables

En 1986 Geroge Box utilizó el término Factor Sparsity [Box86] para describir un modelo que tiene un subconjunto de estimadores iguales a cero y donde el resto del conjunto son los únicos necesarios para dar una respuesta.

A partir de lo mencionando anteriormente se puede establecer una relación con el principio de parsimonia de Guillermo de Ockham también conocido como Navaja de Ockham [Delgado-Trejos09]. Seguramente en más de una ocasión se nos ha complicado demasiado un problema, cuando la solución se podía haber encontrado por el lado más sencillo. El principio de parsimonia establece que la solución más simple suele

ser la mejor [Delgado-Trejos09].

Por lo establecido anteriormente, un modelo parsimonioso es muy interesante, principalmente cuando existen una cantidad inmensa de datos. La solución se puede encontrar de una forma sencilla haciendo selección de variables, porque a menudo este problema se complica por la enorme cantidad de variables que modelan los datos. Lo que se busca es encontrar un modelo que ocupe el menor número de variables para establecer una relación de los datos. Además el modelo debe ser capaz de obtener un buen rendimiento en términos generales; precisión, estabilidad.

Pero decidir qué variables explicativas deben incluirse en el modelo no siempre es trivial y viene acompañado de las siguientes preguntas.

- ¿Cómo obtener un modelo sencillo?
- ¿Cómo seleccionar las variables más relevantes?
- ¿Cómo descartar las variables que no aportan información importante a la respuesta?

Para responder correctamente cada una de estas preguntas es necesario tomar muchos factores en consideración y a lo largo de este trabajo se resolverán. En particular en el Capítulo 3, se mencionan todos los fundamentos de las técnicas basadas en la selección de variables analizadas en esta tesis.

Por otra parte, las técnicas de selección de variables vistas en este trabajo se combinan con validación cruzada una técnica de aprendizaje supervisado. El problema de aprendizaje supervisado ha sido altamente desarrollado en la teoría y en la práctica a lo largo de la historia. [Mitchell97, Hastie01, Vapnik95]. Uno de los factores más importantes en las técnicas de aprendizaje supervisado es el conjunto de entrenamiento proporcionado ya que de él depende la mejor generalización del modelo de regresión. Por tal razón es conveniente contar con un amplio conjunto de observaciones, pero en muchas ocasiones esto no es posible. Esta técnica es de vital importancia en la co-

rrecta selección del mejor subconjunto de variables a conformar el modelo final para pronosticar las series de tiempo.

A continuación se presentan los métodos que están estrechamente ligados con LASSO, los cuales se conocen como métodos de regularización. Estos métodos de regularización funcionan regularizando los estimadores mediante el control del crecimiento de los coeficientes.

2.2. Técnicas de regularización en modelos lineales

En términos generales, la regularización busca convertir los problemas mal condicionados, a uno bien condicionado, por ejemplo el caso $p >> n$, sería un problema mal condicionado [Pötscher09].

Una formulación puede realizarse de la siguiente manera.

$$f(\beta) = \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q \tag{2.1}$$

La regularización toma efecto sobre el tamaño del valor de las β, que a su vez depende de λ.

En la Figura 2.1 se puede ver la región factible de dos variables, donde la función objetivo es la norma.

Para evitar que la regularización varíe frente a cambios de escala de las variables, éstas deben ser estandarizadas a media 0 y desviación estándar 1. El valor de λ afecta directamente el crecimiento o disminución de los valores de los coeficientes a estimar. El valor de λ se ajusta variando desde cero hasta un valor lo suficientemente grande que provoca que los coeficientes lleguen a ser iguales a cero. Caso contrario sucede cuando λ es muy pequeño, es decir, el factor de regularización es tan pequeño que no afecta a los valores de los coeficientes de tal manera que tiende a producir los mismos valores que OLS. Se establece que cuando $\lambda \longrightarrow \infty$ tiende a producir los coeficientes iguales a cero y si $\lambda \longrightarrow 0$ tienden a ser los coeficientes de OLS.

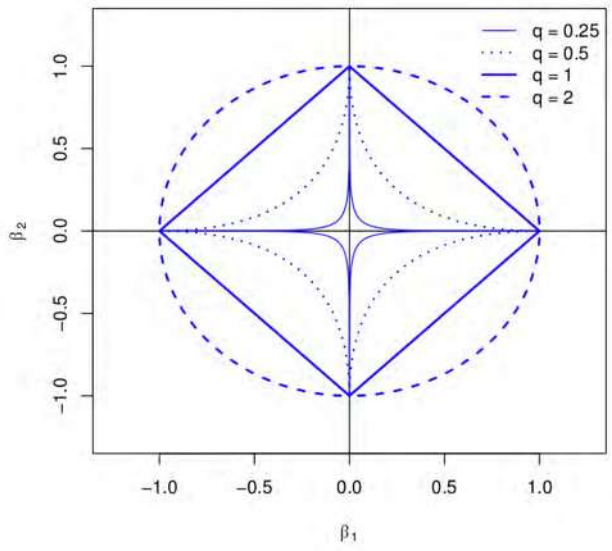


Figura 2.1: Región factible de la función L_q, donde $0 < q \leq 2$ en dos dimensiones.

El primer modelo desarrollado como una técnica de regularización y relacionado con OLS es la regresión ridge [Hoerl70]. Este método consiste en minimizar la función cuadrática de los residuos tal como lo hace mínimos cuadrados pero con una constante no negativa que controla la compensación del ajuste del modelo, sujeta por la sumatoria de los cuadrados de los coeficientes de regresión, o equivalentemente a la norma L_2. La regularización cuadrática L_2 induce a una contracción hacia cero de los coeficientes pero nunca llegan a ser cero por lo cual no se considera un modelo de selección de variables. Otro método es agrupación de LASSO [Yuan06] introduce la norma L_1 en la norma L_2, sin ser una combinación de ambas. A la combinación de la norma L_1 y L_2 se le conoce como redes elasticas [Zou05a] la principal idea de unirlos es por que esta técnica fomenta un efecto de agrupación, donde las variables fuertemente correlacionadas tienden a estar dentro o fuera del modelo sin importar cual es la seleccionada. Agrupación de LASSO también fue propuesta en la versión bayesiana [Raman09]. Todas las metodologías son establecidas a partir de los modelos lineales [Roth08]. Otro modelo relacionado a LASSO es la selección de componentes y suavizado en los operadores (COSSO por sus siglas en ingles) [Lin07]. LASSO es

muy utilizado cuando los datos son de altas dimensiones [Zou05b, Wu09]. Igualmente existen trabajos relacionados mediante la versión bayesiana logística [Park05]. Este método sugiere que los estimadores se pueden interpretar como a posteriori cuando los parámetros independientes se distribuyen idénticamente [Park08]. El comienzo de este tipo de metodologías se remontan en los años 1985 [Chen98, Donoho06].

A continuación se presentan las principales técnicas de selección de variables que son una estrategia diferente a la de regularización.

2.3. Principales técnicas de selección de variables

Los métodos por etapas son técnicas de selección de variables, donde el procedimiento se basa en seleccionar el mejor modelo de manera secuencial incluyendo o excluyendo una sola variable en cada paso según criterios de evaluación.

Básicamente tres algoritmos son los más conocidos; Eliminación hacia atrás, Selección hacia adelante y Selección por etapas.

Estos métodos producen un modelo parsimonioso y a la vez eficiente sobre la base de un conjunto de datos. Pero la eficiencia está muy lejos de ocurrir principalmente porque estas técnicas excluyen un gran número de posibles combinaciones.

Como se menciono anteriormente estos procedimientos realizan su función por etapas, utilizando un determinado criterio para decidir sobre la inclusión o exclusión de una determinada variable y a continuación se comentan.

2.3.1. Eliminación hacia atrás

Esta método comienza con todas las variables en el modelo y por cada paso va excluyendo una variable [Draper81]. Su funcionamiento es el siguiente. Por cada etapa, elimina una variable del modelo, selecciona aquella variable que tiene el valor absoluto más pequeño entre las variables incluidas aún en el modelo (este criterio de la elección de la variable a salir del modelo puede variar respecto al criterio de cada investigador).

El procedimiento finaliza si llega a la etapa k, establecida como un número fijo de terminación.

Los principales inconvenientes de este método son: una variable que ha sido eliminada del modelo, nunca puede volver a entrar al modelo y excluye una variable por etapa excluyendo posibles combinaciones.

2.3.2. Selección hacia adelante

Esta metodología de selección de variables comienza con el modelo vació y va incluyendo al modelo aquella variable que cumpla con una serie de condiciones [Draper81].

- La variable que tiene el valor absoluto más grande entre las variables no incluidas aún en el modelo.
- La variable que produce la mayor reducción del error viendo la respuesta.
- La variable que tiene la correlación parcial más alta en valor absoluto con la respuesta, tomando en cuenta las variables ya incluidas en el modelo.

Finaliza cuando no hay variables significativas para entrar al modelo o cuando se han incluido todas las variables. Este algoritmo tiene un gran problema ya que una vez que una variable entra al modelo, ya no puede salir.

2.3.3. Selección por etapas

Este algoritmo es una combinación de Eliminación hacia atrás y Selección hacia adelante, basa su funcionamiento en la selección de la variable de acuerdo a las variables más correlacionadas con la respuesta [Draper81].

El algoritmo comienza con el modelo vacío y por cada paso agrega una variable y si es el caso existe la posibilidad de eliminarla, su funcionamiento se establece a partir de los siguientes pasos.

La variable que entra al modelo es la que presente mayor correlación con la respuesta.

Se elimina una variable si ocasiona un error mayor al anterior o bien se pueden tomar una serie de reglas igual a la eliminación hacia atrás.

El procedimiento termina si se cumple alguna de las siguientes condiciones.

- Si la última variable incluida no sea significativa.
- Cuando se hayan incorporado al modelo todas las posibles variables disponibles.
- Si llega a la etapa k, establecida como un número fijo de terminación.

A diferencia de los algoritmos Eliminación hacia atrás y Selección hacia adelante este algoritmo si puede eliminar una variable ya en el modelo. La crítica principal a estos métodos de selección radica en todas las posibles combinaciones que se hacen para ajustar el modelo final, lo que indica una gran demanda de cálculos. A partir de los inconvenientes mencionados surge la necesidad de crear nuevos procedimientos de selección de variables que no realicen un proceso discreto explorando cada variable descartada o seleccionada.

Un método que llamo la atención de los investigadores es Regresión hacia adelante por etapas que a continuación se aborda.

2.3.4. Regresión hacia adelante por etapas

Este algoritmo basa su funcionamiento en la selección de una a variable a la vez, seleccionando aquella variable que tiene la correlación absoluta más grande con la respuesta, a este conjunto de variables se le llama conjunto activo [Efron04].

Los pasos a realizar a continuación se presentan:

1. Comienza con $r = y - \hat{y}$ y $\beta_1, \beta_2, ..\beta_p = 0$.
2. Encuentra la variable x_j más correlacionada con r.

3. Actualiza $\beta_j = \beta_j + \alpha_j$, el valor de $\alpha_j = \ell.signo[corr(r, x_j)]$, donde ℓ es un paso pequeño entre 0 y 1.

4. Se repite el paso 2) y 3) hasta que ninguna variable pueda entrar a partir de un criterio.

Por otra parte a continuación se presentan algunas aplicaciones de LASSO y LARS.

2.4. Aplicaciones de LARS y LASSO

Cabe mencionar que una importante revista en estadística [Efron04] en 2004 dedica 92 páginas a LARS. El artículo va seguido de un debate sobre las ventajas y desventajas de LARS [Weisberg04], este trabajo se realizo con la colaboración de varios expertos en el tema. Por tal razón se considera como uno de los actuales métodos que goza de una destacada popularidad e incluso se encuentran aplicaciones en diversas áreas que en este apartado se comentarán brevemente. El número creciente de citas que tienen estos dos algoritmos nos dan una pauta del interés que han despertado estas técnicas. LARS y LASSO se han aplicado sobre todo en la solución de problemas reales, muchas de ellos en la área de visión computacional (reconocimiento facial) y en la biomédicina [Yang10, Singaraju12]. Donde principalmente se estudia el DNA.

En la página web $http : //www.eecs.berkeley.edu/\ yang/software/$ se pueden encontrar diferentes algoritmos desarrollados en C, C++, matlab, plataforma móvil, C/CUDA de las aplicaciones al reconocimiento facial principalmente mediante este tipo de técnicas. En las páginas web $http : //www.eecs.berkeley.edu/$ y $http : //yima.csl.illinois.edu/$ se encuentra cualquier cantidad de artículos relacionados a este tema.

Por otra parte, LARS ha sido aplicado en la industria de los sistemas inteligentes a las empresas Sun Microsystems y getgoing para la personalización de aplicaciones

Web [Gluhovsky11]. El primer objetivo es encontrar el ciclo de vida de un considerado cliente potencial. El segundo objetivo es presentar los registros adecuados para cada usuario y posiblemente se abran las oportunidades de ventas. Todo esto se realiza dependiendo del cliente, es necesario identificarlo como primer instancia para poder personalizar la página web a su necesidad. Para este fin, primero se aplica LARS para clasificar un usuario en una de las siguientes categorías: programador, administrador de sistemas, socio minorista, socio mayorista, negocio pequeño, negocio grande, inicio, estudiantes, etc. Es importante mencionar que el conjunto de entrenamiento fue alimentado en acciones de usuarios dentro y fuera de la web de la empresa. Como son visitas, contenido, descargas de software, los registros, la interacción de ventas, etc.

En el enfoque de personalización, donde el modelo de LARS es ajustado para predecir la probabilidad de éxito de un artículo en el contexto de un usuario. Por último los resultados arrojan que sólo el 30 por ciento del total de usuarios involucrados en el estudio obtuvieron una predicción errónea mientras que el resto fueron predicciones exitosas.

2.5. Resumen

Este capítulo comienza con una breve introducción de los inicios de la selección de variables, En seguida se comentaron los métodos de regularización más populares y algunas técnicas de selección de variables que frecuentemente se utilizan para resolver el problema de regresión. Por último se presentan algunas aplicaciones de LARS y LASSO en diversas áreas.

Capítulo 3

Fundamentos de los modelos de selección de variables para la predicción de series de tiempo

Este capítulo presenta los fundamentos de las técnicas basadas en la selección de variables y la importancia de este proceso previo a obtener el modelo final para pronosticar las series de tiempo. Presenta el funcionamiento y algoritmos de las técnicas de LASSO-puro, LASSO-umbral y LARS. Además presenta un análisis del problema de sobre-aprendizaje y la implicación directa que tiene esto con la selección del mejor subconjunto de variables a modelar el modelo final.

3.1. Fundamentos

Una de las cuestiones más importantes en la selección de un modelo, es la correcta selección del subconjunto de variables que modelarán los datos. La selección correcta de este modelo es de vital importancia, ya que de él depende el buen rendimiento y la precisión en la obtención de la respuesta.

Con lo establecido anteriormente, el objetivo que debe plantearse en todo método

de selección de variables es satisfacer los siguiente puntos.

- Precisión en la predicción.
- Modelos interpretables.
- Estabilidad.
- Baja complejidad en tiempo.

En este trabajo se demuestra que los modelos obtenidos con LASSO y LARS cumplen con los puntos anteriores. Son interpretables porque se reduce el número de variables que conforman el modelo, se tiene mayor estabilidad debido a que no se toman en cuenta las variables irrelevantes o redundantes y tienen un poder predictivo superior a OLS o en el peor de los casos igual. Además LASSO-umbral y LARS tienen un tiempo de complejidad polinomial.

Por otra parte, una variable es irrelevante cuando el conocimiento del valor de la misma no aporta información alguna que despeje incertidumbre sobre la respuesta. Una variable es redundante cuando su valor puede ser determinado a partir de otras variables predictivas.

El objetivo sigue siendo el mismo, lo que se pretende es crear modelos parsimoniosos, es decir, la idea subyacente de la simplicidad; si se dispone de dos modelos que explican suficientemente bien los datos, se debe escoger el modelo más simple de los dos la cual suele ser la mejor solución. Esto trae beneficios directos derivados de la correcta selección de variables y a continuación se enumeran.

1. Reducción en el costo de adquisición de los datos, debido a que el volumen de información a manejar para inducir el modelo es menor.
2. Mejor comprensión del modelo, ya que no se induce en el modelo un gran número de variables.

3. Inducción más rápida del modelo, derivado de la posibilidad de indagar en la naturaleza de distintos casos, debido al tamaño más manejable de cada instancia, la tarea de identificar patrones en ciertos subconjuntos se vuelve más fácil.

4. Mejora en la precisión del modelo, el hecho de que no existan variables redundantes y/o variables irrelevantes hace que su comportamiento sea óptimo.

5. Se pueden abordar los modelos más complejos, debido a la reducción de los datos.

6. Se evita el bajo ajuste o sobre ajuste que producen los modelos complejos [Hawkins04].

Encontrar la solución al problema de selección de variables se convierte en un verdadero reto porque se tienen que realizar 2^p combinaciones, donde p es el número de variables. Esto significa que en la práctica, encontrar la solución óptima cuando el tamaño del problema es grande es prácticamente imposible en cuestión de tiempo. Sin embargo, las metodologías presentadas en este trabajo (LASSO y LARS) realizan los cálculos en tiempo polinomial mediante la selección de variables pero sin hacer las combinaciones de variables. Estas técnicas realizan procedimientos similar a la técnica de mínimos cuadrados.

Antes de comenzar a introducir estas metodologías que basan su funcionamiento en la selección de variables, vamos a presentar OLS ya que de este método parten dichas técnicas.

3.2. Mínimos cuadrados ordinarios

Mínimos cuadrados ordinarios, es un método de regresión clásico. Este método estadístico de estimación de coeficientes desconocidos ajusta los datos a partir de una población muestral.

Esta técnica minimiza el residuo de los errores al cuadrado, este error es la diferencia de la respuesta y y la estimada $\hat{y}$. [Hayashi00]. Consiste en determinar los parámetros β_j de tal manera que los residuos sean mínimos. Trata de buscar cueles son los coeficientes $\{\beta_1, ..., \beta_p\}$ que hacen que la combinación lineal 3.1 aproxime óptimamente al vector respuesta y.

$$\hat{y} = \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p \tag{3.1}$$

La Figura 3.1 muestra la proyección de y sobre el subespacio generado por las columnas $x_1...x_p$. para dos vectores x_1 y x_2.

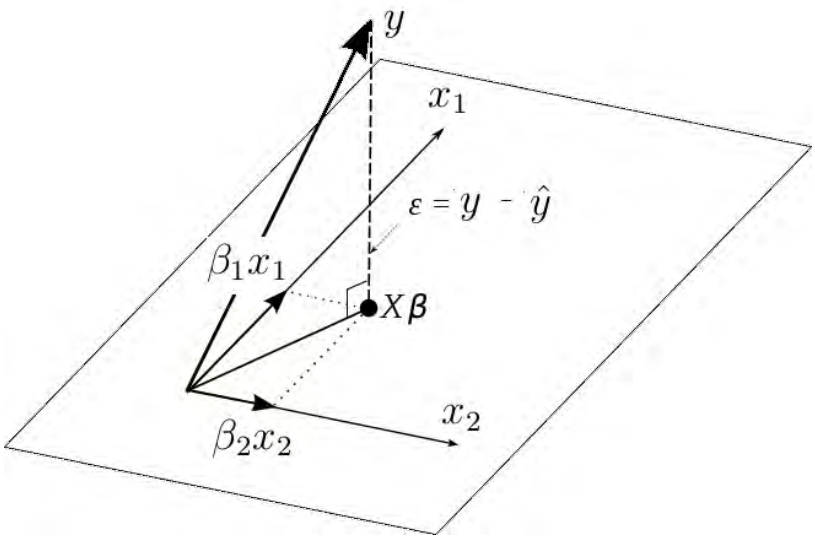


Figura 3.1: Proyección de y sobre dos combinaciones lineales dado los estimadores $\hat{\beta}_1$ y $\hat{\beta}_2$, donde $\hat{\epsilon}$ representa el error entre y y $\hat{y}$

Desde el punto de vista de optimización, OLS busca minimizar la ecuación (3.2).

$$f(\beta_j) = \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 \tag{3.2}$$

El conjunto de valores de β_j que minimiza la sumatoria de los errores al cuadrado son llamados estimadores OLS, obviamente, cuanto menor es el error, mejor es el ajuste. La función $f(\beta_j)$ es convexa, por lo tanto tiene un mínimo global.

$$\text{Minimizar } \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 \tag{3.3}$$

Si escribimos la ecuación (3.3) en forma matricial, se convierte en

$$\left\|y - X\beta\right\|^2 = (y - X\beta)^T(y - X\beta) = y^T y - 2y^T X\beta + X^T X\beta^2 \tag{3.4}$$

Asumiendo que X es no-singular y que $X'X$ es definida positiva, se puede derivar la ecuación (3.4) con respecto a β e igualar a cero, se obteniendo lo siguiente.

$$-2X^T y + 2X^T X\beta = 0 \tag{3.5}$$

Si se despeja β de la ecuación (3.5) se obtienen los estimadores de mínimos cuadrados ordinarios, obteniendo una solución única mediante la formula cerrada.

$$\beta = (X^T X)^{-1} X^T y \tag{3.6}$$

Sin embargo, la pseudo-inversa $X^T X$ puede ser singular, es decir su determinante es igual a cero. Si este problema ocurre, no es posible calcularse debido a que no se puede invertir. Para solucionar este problema, se pueden utilizar algoritmos iterativos de aproximación [Ruhe92, Campbell09].

OLS es el mejor método linealmente insesgado y la solución es obtenido mediante una formula cerrada. Pero tiene desventajas que a continuación se mencionan: Primero, no produce modelos sencillos lo cual lleva a la estimación de todos los coeficientes violando el principio de parsimonia; segundo, a causa de lo anterior se producen modelos sobre-ajustados. Por si fuera poco, OLS depende de la singularidad de la matriz de correlación. Especialmente el problema de la seudo-inversa ocurre cuando existe dependencia lineal o cuando el número de variables es más grande que el número de observaciones (no es de rango completo).

Por tal razón, otra opción viable para obtener los coeficientes es haciendo selección de variables. Una de estas técnicas es LASSO una técnica de regulariazación que surge

a partir de OLS. A diferencia de OLS, LASSO logra evitar muchos problemas que se venían presentando en OLS [Tibshirani94]. Una de las principales ventajas de LASSO es que al ser un algoritmo de selección de variables se obtienen modelos sencillos que enriquecen la interpretación de los datos. También obtiene una gran presición en la predicción, principalmente porque evita el sobre-ajuste que se presenta con OLS. A gran medida evita el sobre-ajuste por la combinación con técnicas de aprendizaje supervisado convirtiéndose en una alternativa muy viable en problemas de regresión de grandes dimensiones. A continuación se presenta LASSO y las diferentes soluciones propuestas en este trabajo.

3.3. LASSO

LASSO [Tibshirani94] es a la vez un método de selección de variables y de contracción. Esta técnica no hace selección de variables puramente, sin embargo, si se aplica un umbral logra ser una técnica de selección de variables. El funcionamiento de LASSO se basa en la minimización de los errores al cuadrado similar a OLS pero sujeto a la restricción que tiene como cota superior un valor constante t que controla la sumatoria de los valores absolutos de los estimadores de regresión. Esto es equivalente a la norma L_1 y se define como sigue.

$$\begin{aligned} &\text{Minimizar } \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 \\ &\text{Sujeto a.} \\ &\sum_{j=1}^{p} |\beta_j| \leq t \end{aligned} \tag{3.7}$$

Donde $t \geq 0$ es un parámetro a optimizar. El valor de esta cota superior comienza desde cero hasta el valor total de la sumatoria de los coeficientes absolutos de OLS. Es este trabajo, este valor se encuentra con validación cruzada y juega un papel importante en el buen rendimiento del modelo.

Como se puede observar en la ecuación (3.7), LASSO es un problema de programación cuadrática [Boyd04] y la solución puede obtenerse con los métodos de punto interior [Curtisy05, Gondzio12, Anstreicher90] y el método del conjunto activo [Gupta11, Gratton11]. Sin embargo resolverlo con este tipo de algoritmos implica muchos cálculos e implica un proceso más complejo.

Resolver el problema de programación cuadrática nos lleva a calcular las derivadas de los absolutos que a continuación se define.

$$\nabla f(|x|) = \frac{x}{|x|}\nabla x \tag{3.8}$$

Asimismo nuestro principal objetivo es generar modelos parsimoniosos, es decir, el valor de algunos estimadores son iguales a cero y la definición de la ecuación (3.8) muestra una indeterminación si $x = 0$. La representación de la restricción del problema 3.7 puede ser visto como un conjunto de ecuaciones lineales, donde se realizan todas las posibles combinaciones de los signos para cada β. Esto quiere decir que el conjunto factible se pueden ver como un poliedro, compuesto por un conjunto de restricciones lineales de todas las posibles combinaciones de los signos de β (ver la ecuación (3.9)).

La ecuación (3.9) representa las combinaciones de los signos de cada una de las β para el caso de tres variables.

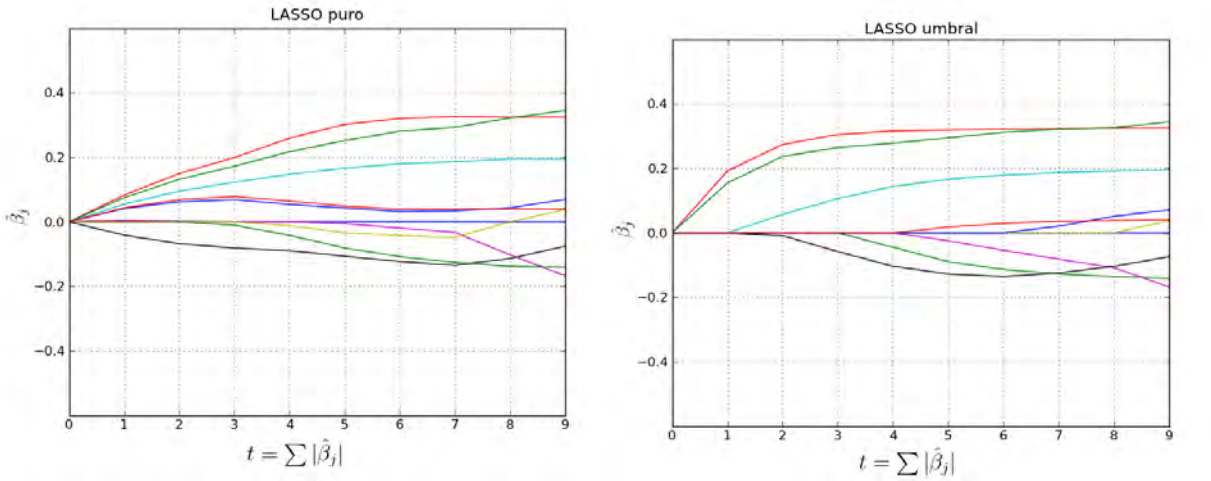
$$
\begin{aligned}
+\beta_1 + \beta_2 + \beta_3 &\leq t \\
+\beta_1 - \beta_2 + \beta_3 &\leq t \\
+\beta_1 + \beta_2 - \beta_3 &\leq t \\
+\beta_1 - \beta_2 - \beta_3 &\leq t \\
-\beta_1 + \beta_2 + \beta_3 &\leq t \\
-\beta_1 + \beta_2 - \beta_3 &\leq t \\
-\beta_1 - \beta_2 + \beta_3 &\leq t \\
-\beta_1 - \beta_2 - \beta_3 &\leq t
\end{aligned}
\tag{3.9}
$$

Es claro que minimizar la función cuadrática sujeta a estas restricciones será la solución de la ecuación (3.7), sin embargo, es evidente que este problema de programación cuadrática tiene 2^p restricciones, lo que provoca que el procedimiento no sea trivial sobre todo cuando existe un número muy grande de variables.

Con la finalidad de evitar el problema anteriormente mencionado, LASSO se resolvió con una librería de optimización de python llamado scipy, que utiliza un método de optimización numérico donde todo el procedimiento se realiza mediante aproximaciones lineales haciendo interpolación de newton de primer grado [Jones , Powell98].

La solución de este problema le llamaremos LASSO-puro. Este algoritmo no resulta ser un método de selección de variables puramente, esto quiere decir que en cuanto incrementa el valor de la cota superior varios coeficientes inmediatamente dejan de ser cero, especialmente cuando son vectores muy influyentes en la respuesta, sin embargo los vectores con menor relevancia si tienden a ser cero, pero en pocas ocasiones llegan a ser cero.

A partir de lo establecido anteriormente se concluye que LASSO-puro no genera modelos sencillos.



(a) *Lasso puro* **(b) *Lasso umbral***

Figura 3.2: Comparación de LASSO-puro y LASSO-umbral en la obtención de los valores del vector de los estimadores.

La Figura 3.2 representa los valores del vector de los estimadores a lo largo de la selección de variables de LASSO. Es decir al inicio todos los valores de los coeficientes de los estimadores β son iguales a cero. Recordar que la función cuadrática está sujeta a una restricción controlada por una cota superior t. Al inicio el valor de t es igual a cero pero el valor incrementa lo cual repercute directamente a los valores de los coeficientes estimados.

La Figura 3.2 se lee de la siguiente manera; La Figura 3.2 (a) representa los valores del vector estimador de LASSO-puro, es decir, el algoritmo que se resuelve mediante optimización sin aplicar el umbral que en seguida se explicará a detalle. La Figura 3.2 (b) representa los valores a lo largo de la selección de variables aplicando el umbral. Como conclusión de esta Figura 3.2 se destaca la importancia que LASSO-umbral realiza selección de variables una a la vez, mientras el algoritmo LASSO-puro introduce todas las variables y tiende a reducir el valor de los estimadores por el hecho de estar controlado por el valor de t. Esto se debe a la aplicación de un umbral y la suposición de un diseño ortogonal en la matriz. Asimismo se menciona que el valor de t varia desde 0 hasta el valor de la sumatoria de los valores de los estimadores de

mínimos cuadrados. A continuación se presenta el algoritmo LASSO-puro.

3.3.1. Algoritmo LASSO-Puro

Este algoritmo comienza con los valores de los coeficientes iguales a cero y $t = 0$. La Linea 3, ValorMax es el criterio de terminación debe ser menor o igual que los valores absolutos de los estimadores de mínimos cuadrados. La Linea 4 y Linea 5, fórmula el problema de optimización de LASSO. La Linea 6, resuelve el problema de optimización, donde el número máximo de iteraciones es el primer criterio de convergencia y el algoritmo es un método numérico que no hace derivadas simplemente simula la dirección del gradiente mediante aproximaciones lineales. La Linea 7, incrementa el valor t. La Linea 8, guarda los valores de los estimadores con la finalidad de hacer validación cruzada y encontrar la mejor generalización a partir del valor t.

Algoritmo 1 Algoritmo LASSO-Puro

LASSO-PURO$(X, y, t, Incremento, ValorMax)$

1 $betas \leftarrow 0$
2 $RutaBetas \leftarrow []$
3 **mientras** $t \leq ValorMax$
4 $\quad Fobjetivo \leftarrow (y - (X.betas))^2$
5 $\quad restriccion \leftarrow t - Sum(Abs(betas))$
6 $\quad betas \leftarrow Minimizar(Fobjetivo, restriccion)$
7 $\quad t \leftarrow Incremento + t$
8 $\quad RutaBetas.append(betas)$
9 **regresar** $RutaBetas$

Como se mencionó anteriormente esta solución no produce modelos parsimoniosos.

LASSO logra ser una modelo de selección de variables cuando se un umbral como término de regularización [Friedman10, Wu08]. Resulta interesante que las librerías de (pyhton y R) solucionan LASSO con coordenada descendente aplicando dicho umbral en el problema de regularización de LASSO.

LASSO se puede ver como un método de regularización [Wu08], donde el funcionamiento depende de λ, este valor controla la contracción o incremento de los coeficientes y tiene la siguiente forma.

$$f(\beta_j) = \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{3.10}$$

Donde $\lambda \geq 0$ es un parámetro a optimizar y en este trabajo se obtiene con validación cruzada.

La solución de la ecuación (3.7) y la ecuación (3.10) resulta ser una correspondencia uno a uno entre λ y t. Esto es, si $\beta_j(\lambda)$ minimiza la ecuación (3.10) también resuelve la ecuación (3.7) con $t = \sum_{j=1}^{p} |\beta_j(\lambda)|$ [Wu08]. Resolver LASSO con coordenada descendente resulta más conveniente por su simplicidad y porque fijando el parámetro λ se convierte en un problema sin restricciones.

Tibshirani [Tibshirani94] propuso la solución de LASSO para un diseño ortogonal, donde la matriz X es ortogonal [Chen98, Donoho06]. Esto con la finalidad de hacer selección de variables, de tal manera que los valores de β_j lleguen a ser igual a cero mediante la aplicación de un umbral. A continuación se presenta el procedimiento para obtener la solución.

Si LASSO se resuelve suponiendo que la matriz X es ortonormal la solución de la ecuación (3.10) es la ecuación (3.11). Donde $(|\beta_j| - \lambda)^+$ es la parte positiva de la resta de los valores en caso contrario se toma el valor 0.

$$\beta_j(\lambda) = signo(\beta_j)(|\beta_j| - \lambda)^+ = \begin{cases} \beta_j - \lambda, & si \quad \beta_j > 0 \quad y \quad \lambda < |\beta_j|, \\ \beta_j + \lambda, & si \quad \beta_j < 0 \quad y \quad \lambda < |\beta_j|, \\ 0, & \lambda \geq |\beta_j|. \end{cases} \tag{3.11}$$

Esta solución se divide en dos partes; la primer parte corresponde en la obtención de los estimadores de la función cuadrática, donde β_j se obtiene con mínimos cuadrados ordinarios y la segunda parte es aplicar el umbral donde λ juega un papel muy importante como un término de regularización y el valor de λ es equivalentemente a la condición $\sum_{j=1}^{p} |\beta_j| \leq t$. Cuando t es suficientemente pequeño, λ es grande de modo que algunos de los coeficientes llegan a ser iguales a cero.

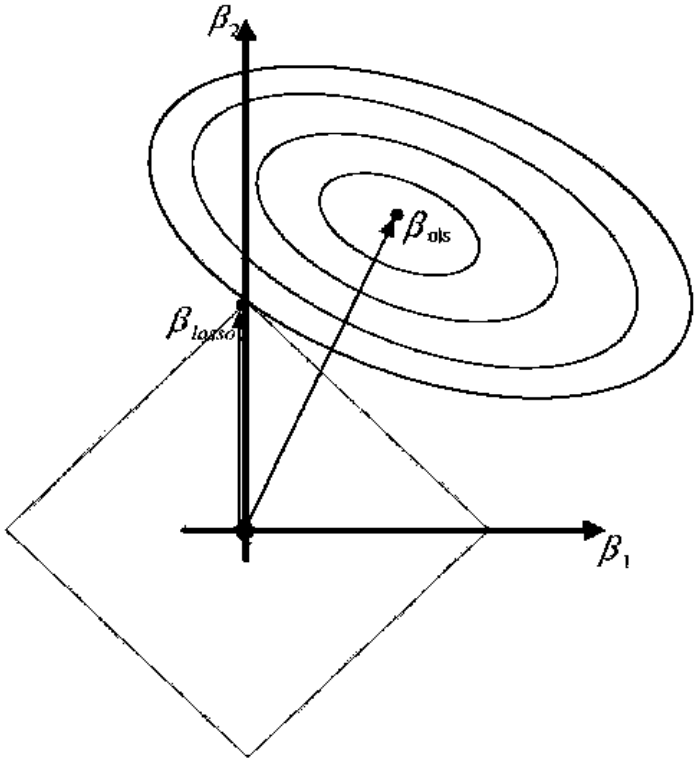


Figura 3.3: Representación geométrica de LASSO en dos dimensiones.

La Figura 3.3 es la representación geométrica de LASSO, donde el contorno de la

elipse son los estimadores de la función cuadrática obtenida por OLS y el cuadrado rotado es la región factible de la sumatoria de la restricción $\beta_1 + \beta_2 \leq t$, es un caso especial de dos variables de todas las posibles restricciones $\sum_{j=1}^{p} |\beta_j| \leq t$. Claramente se puede apreciar que la elipse toca el cuadrado exactamente cuando una variable es igual a cero y la otra es diferente a cero.

A continuación se desarrolla el procedimiento para obtener la ecuación (3.11), donde el problema se reducirá al caso más básico con la finalidad de mostrar la obtención del umbral. Si recordamos $I = X^T X$ es una propiedad de las matrices ortogonales, entonces la solución OLS (ver la ecuación (3.6)) se simplifica como $\beta^{OLS} = X^T y$.

A partir de lo establecido anteriormente la ecuación (3.10) se puede escribir de la siguiente forma.

$$f(\beta) = \frac{1}{2}(y - X\beta)^T(y - X\beta) + \lambda|\beta| \tag{3.12}$$

y si desarrollamos los productos obtenemos.

$$f(\beta) = \frac{1}{2}y^T y - X^T y\beta + \frac{1}{2}\beta^2 + \lambda|\beta| \tag{3.13}$$

Sustituyendo $\beta^{OLS} = X^T y$ de la ecuación 3.13 obtenemos.

$$f(\beta) = \frac{1}{2}y^T y - \beta^{OLS}\beta + \frac{1}{2}\beta^2 + \lambda|\beta| \tag{3.14}$$

Formalmente, el valor absoluto de todo número real β se define como sigue (ver ecuación (3.15)):

$$|\beta| = \begin{cases} \beta, & si \quad \beta \geq 0 \\ -\beta, & si \quad \beta < 0 \end{cases} \tag{3.15}$$

Por otra parte, la función a optimizar respecto a β es $f(\beta)$ (ver la ecuación (3.14)). Si derivamos la ecuación (3.14) respecto a β y se iguala a cero, obtenemos lo siguiente

$$\nabla f(\beta) = \beta - \beta^{OLS} + \frac{\beta}{|\beta|}\lambda = 0 \tag{3.16}$$

Si despejamos β de la ecuación (3.16).

$$\beta = \beta^{OLS} - \frac{\beta}{|\beta|}\lambda \tag{3.17}$$

En este punto se consideran dos casos, el primer caso consiste en obtener el valor del estimador β considerando las siguientes consideraciones $\beta^{OLS} > 0$ y $\beta \geq 0$ obteniendo lo siguiente.

$$\beta = \beta^{OLS} - \lambda = (\beta^{OLS} - \lambda)^{+} \tag{3.18}$$

Donde $(\beta^{OLS} - \lambda)^{+}$ indica que es la parte positiva de la citada resta en caso contrario el valor que toma es cero. El segundo caso consiste en las siguientes consideraciones $\beta^{OLS} < 0$ y $\beta \leq 0$, entonces la solución es.

$$\beta = \beta^{OLS} + \lambda \tag{3.19}$$

A partir de los dos casos establecidos anteriormente se puede generalizar la siguiente ecuación.

$$\beta_j(\lambda) = signo(\beta_j^{OLS})(|\beta_j^{OLS}| - \lambda)^{+} \tag{3.20}$$

Si no corresponde a ninguno de los casos anteriores, $\beta = 0$ cuando $\lambda \geq |\beta|$. Entonces la solución de $f(\beta)$ converge a la solución de LASSO.

Por otra parte la solución para el caso multivariable consiste en la misma idea, fijar el parámetro de penalización o regularización λ y optimizar sucesivamente respecto de cada parámetro β_j, dejando los restantes parámetros β_k, $k \neq j$ fijos en sus valores actuales. La actualización se repite para cada una de las variables hasta que converge.

3.3.2. Algoritmo LASSO-Umbral

Este algoritmo recibe como parámetros X, y y MaxIteraciones. La Linea 5 y Linea 7 hacen el ciclo por coordenada. La linea 8 y Linea 9 representan la obtención de la

función cuadrática OLS (ver la ecuación (3.21)), esto se obtiene derivando la función respecto a β e igualando a cero.

$$-X'(y - X\beta) = \beta_{parcial} \tag{3.21}$$

Una vez obtenido el valor de $\beta_{parcial}$ se aplica el umbral (ver Linea 10).

Algoritmo 2 Algoritmo LASSO-Umbral

LASSO-Umbral$(X, y, MaxIteraciones)$

1 $\beta \leftarrow 0$
2 $ite \leftarrow 0$
3 **mientras** $ite \leq MaxIteraciones$
4 $\quad i \leftarrow 0$
5 $\quad$ **mientras** $i \leq n$
6 $\quad\quad j \leftarrow 0$
7 $\quad\quad$ **mientras** $j \leq m$
8 $\quad\quad\quad r_{ij} \leftarrow y_i - \sum_{k \neq i} X_{ik}\beta_k$
9 $\quad\quad\quad \beta_j \leftarrow \sum_{i=1}^{N} r_{ij}X_{ij}$
10 $\quad\quad\quad \beta_j \leftarrow signo(\beta_j)(Max(|\beta_j| - \lambda, 0))$
11 $\quad\quad\quad j \leftarrow j + 1$
12 $\quad\quad i \leftarrow i + 1$
13 $\quad ite \leftarrow ite + 1$
14 **regresar** β

3.4. LARS

Least Angle Regression (LAR) toma el nombre de LARS porque a partir de este método se obtiene LASSO y forward stepwise regression mediante un algoritmo extremadamente eficiente en términos de tiempo de complejidad. LARS es un nuevo procedimiento de selección de variables [Efron04], y puede ser visto como una versión mejorada de Regresión hacia adelante por etapas.

Regresión hacia adelante por etapas construye el modelo secuencialmente, agregando una variable a la vez, la variable seleccionada es la que tiene la correlación absoluta más grande con la respuesta, después hace el ajuste lineal por regresión de mínimos cuadrados con las variables incluidas en el conjunto activo, y así sucesivamente hasta obtener k variables en el modelo o los residuos sean ceros.

LARS es una estrategia similar en donde el primer paso es identificar la variable más correlacionada, es decir, dado un conjunto de vectores se selecciona el vector que tenga la correlación absoluta más grande con la respuesta. En vez de adaptarse a esta variable completamente, LARS incrementa continuamente el valor del vector estimador hasta el punto donde la dirección del vector tenga la misma correlación con el residuo actual. Este nuevo coeficiente entra al conjunto activo, y el proceso continúa.

La siguiente variable a entrar al modelo se toma de acuerdo a la correlación absoluta más grande con la respuesta y una vez que esta variable es seleccionada a entrar al conjunto activo. LARS se mueve hacia la dirección equiangular para después hacer el ajuste lineal por mínimos cuadrados. Esta dirección equiangular hace que las variables en el modelo tengan la misma correlación con el residuo actual y puede ser visto como una democracia para las variables que aún faltan por entrar al modelo.

Este procedimiento se repite sucesivamente k veces y al finalizar se puede obtener la misma solución que OLS si entran todas las variables al modelo. En este trabajo la constante k se determina mediante validación cruzada y juega un papel muy

importante en el funcionamiento del modelo.

Por otra parte para implementar la estrategia equiangular [Efron04], describe el álgebra necesaria como sigue. Suponemos que nos encontramos en el paso k, donde A_k es el conjunto activo de variables en el modelo y B_{A_k} son los coeficientes de los vectores de las variables.

Esto quiere decir que hay $k-1$ valores diferentes a cero y el coeficiente de la variable a entrar en el k paso es igual a cero. Si $r_k = y - X_{A_k} B_{A_k}$ es el residuo $k-$esimo, entonces la dirección para este paso es

$$\varphi_k = (X'_{A_k} X_{A_k})^{-1} X'_{A_k} r_k \tag{3.22}$$

donde los coeficientes resultan ser $\beta_{A_k} = \beta_{A_k} + \alpha.\varphi_k$. El parámetro α indica el tamaño el cual hace que el residuo actual sea igualmente correlacionado con las variables en el conjunto activo y el otro competidor a entrar.

Debido a la linealidad por tramos del algoritmo y la información de las variables, el tamaño de paso puede ser calculada exactamente al comienzo de cada paso.

Si el vector de ajuste en el comienzo de este paso es $\hat{f}_k$, entonces

$$\hat{f}_k(\alpha) = f_k + \alpha.\mu_k \tag{3.23}$$

donde $\mu_k = X_{A_k}\varphi_k$ es la nueva dirección de ajuste.

Una vez que se conoce como se van obteniendo los coeficientes β, LARS construye la respuesta estimada como sigue:

$$\hat{\mu} = X\beta \tag{3.24}$$

Como se ha mencionado anteriormente, LARS encuentra en cada paso la variable más correlacionada y se obtiene de la ecuación (3.25). Esta variable entra al conjunto activo y forma parte del modelo.

$$\hat{c} = X'(y - \hat{\mu}) \tag{3.25}$$

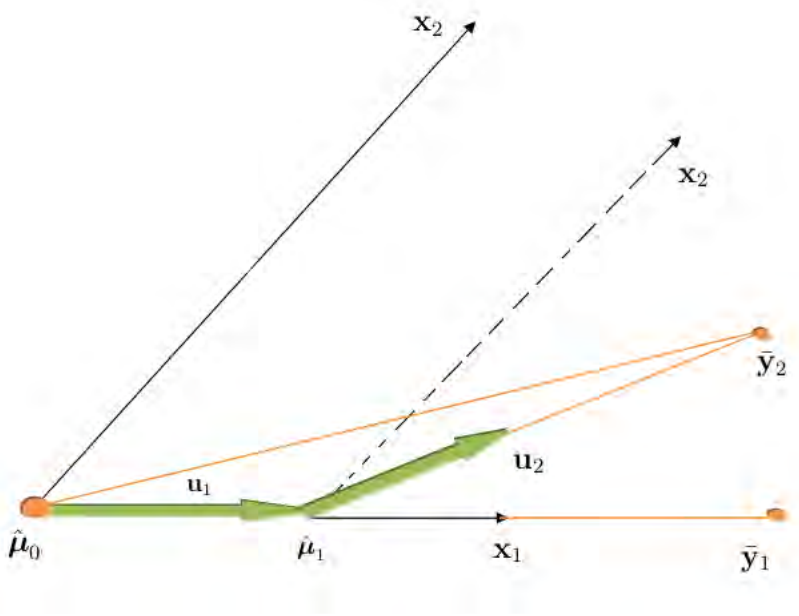


Figura 3.4: Representación geométrica de LARS para 2 variables

La Figura 3.4 muestra la proyección de $\hat{y_2}$ sobre el espacio lineal generado por x_1 y x_2. El algoritmo comienza con $\hat{\mu_0} = 0$. Una vez que se encuentra la variable más correlacionada con la respuesta, entonces $\hat{\mu_1}$ se mueve en la dirección de x_1 (la variable primer variable seleccionada a conformar el modelo). Cuando entra la segunda variable x_2, la dirección de $\hat{\mu_2}$ hace un ángulo medio entre las dos variables y sigue una dirección en medio de las dos variables. Esta proyección es generada por la dirección equiangular. Esto hace una correlación exactamente igual entre las variables en el modelo. Se puede ver claramente que las u va marcando la dirección equiangular parándose un paso atrás de lo que seria equivalente con mínimos cuadrados (ver $\hat{y}$). También es interesante ver la linea punteada de x_2. Donde la linea x_1 y linea x_2 punteada forman el ángulo medio que es equivalente a tener la misma correlación entre las variables en el modelo y la variable nueva a entrar.

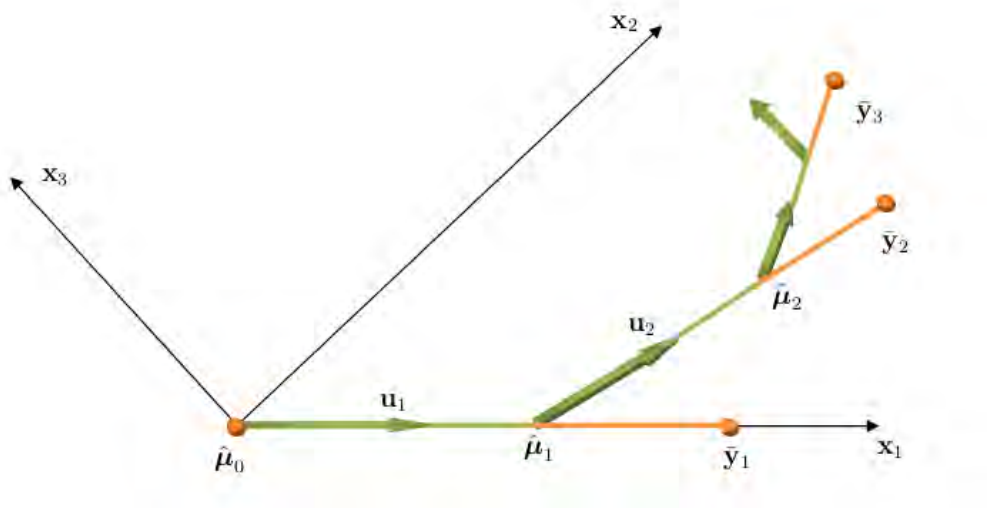


Figura 3.5: Representación geométrica de LARS para 3 variables

La Figura 3.5 representa la trayectoria de la estimación de la respuesta para más de 2 variables específicamente 3 variables. Como se observa LARS sigue la misma idea basada en la dirección equiangular que hace quedar un paso atrás de mínimos cuadrados. La $\hat{y}$ representa la estimación de mínimos cuadrados y $\hat{\mu}$ es la estimación de LARS.

3.4.1. Algoritmo LARS

El algoritmo LARS (ver el Algoritmo 3) funciona como a continuación se presenta.

1. Este algoritmo recibe como parámetros X, y y k número de etapa a parar.

2. Como primer paso es normalizar los datos con media 0 y desviación estándar 1.

3. Comenzar con $r = y - \hat{y}$ y $\beta = 0$.

4. Encontrar la variable x_j más correlacionada con r.

5. Mover $\beta_j = 0$, hacia su coeficiente de mínimos cuadrados (x_j, r), hasta que otra variable x_k tenga la misma correlación con el residuo tal como x_j.

6. Mover β_j y β_k en la dirección definida por los coeficientes de mínimos cuadrados del residuo actual en (x_j, x_k), hasta que otra variable tenga la misma correlación con el residuo actual.

7. Continúa en este camino hasta que k variables entren al modelo, si $k = p$ se obtendrá la solución de mínimos cuadrados.

Algoritmo 3 Algoritmo LARS

LARS(X, y, k)

1 $\hat{\mu} \leftarrow 0$
2 $\beta \leftarrow 0$
3 $i \leftarrow 0$
4 **mientras** $i < k$ or $A^c = \emptyset$
5 $\hat{c} = X'r$ y $C = Max_j|\hat{c}|$
6 $A \leftarrow \{j : C\}$
7 $X_A \leftarrow (...x_j...)_{j \in A}$
8 $\hat{y}_{i+1} \leftarrow (X_A{}'X_A)^{-1}X_A{}'y$ y $a = X_A{}'(\hat{y}_{i+1} - \hat{\mu})$
9 $\gamma \leftarrow min^+\{\frac{C-\hat{c}_j}{C-a_j}, \frac{C+\hat{c}_j}{C+a_j}\}$ donde $j \in A^c$
10 $\hat{\mu}_{i+1} \leftarrow \hat{\mu}_i + \gamma(\hat{y}_{i+1} - \hat{\mu}_i)$
11 $\beta \leftarrow (\beta + \gamma(X_A{}'X_A)^{-1}) * singo(A)$
12 $i \leftarrow i + 1$
13 **regresar** β

3.5. Selección de los valores estimados a lo largo del modelado

Una de los principales ventajas de las técnicas analizadas en este trabajo que basan su funcionamiento en la selección de variables es la interpretación que se puede obtener a lo largo del modelado. Esto implica un análisis sencillo mediante la observación del comportamiento de los valores del vector de estimadores por cada iteración que se realiza en el modelado.

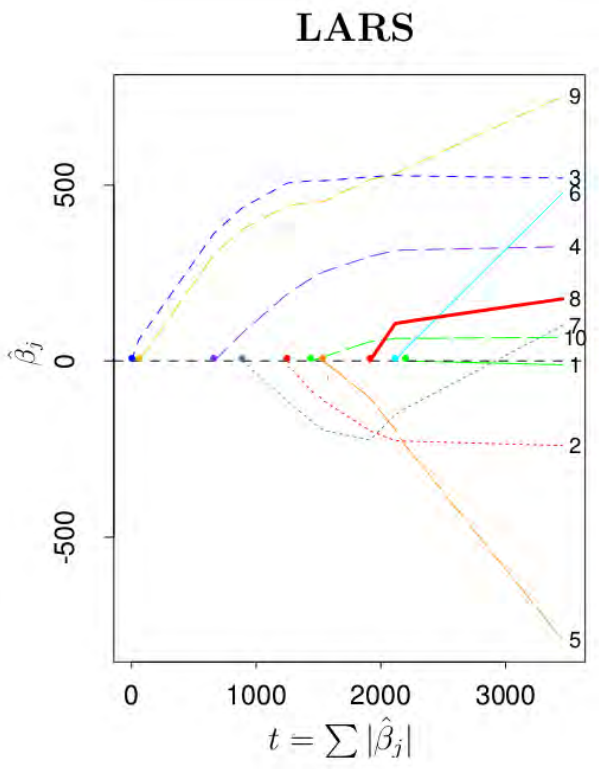


Figura 3.6: Camino del valor de los estimadores en el proceso de la selección de variables en el ejemplo de la diabetes

La Figura 3.6 representa el camino de las variables que entran al modelo en el ejemplo del estudio de la diabetes, donde hay 10 variables que representan la edad, peso, tipo de sangre, etc. Este estudio se realiza con la finalidad de determinar que variables son las más influyentes en la adquisición de esta enfermedad.

La Figura 3.6 se lee de la siguiente manera; la primer variable a entrar al modelo es la variable No. 3 la cual es la más influyente, en seguida entra la variable 9 y así sucesivamente hasta que entran todas las variables al modelo. Recordar que el punto inicial es cuando todas los estimadores tienen valores iguales a cero, por esa razón las variables se representan con un punto cuando dejan de ser cero. Lo que se puede apreciar es el cambio del valor que van tomando los estimadores a lo largo del modelado, es decir, interesa conocer cual es la relación que existe entre las variables. Un aspecto interesante se puede observar cuando una variable entra al modelo, el efecto que puede ocasionar en el resto de las variables que ya están en el modelo, esto se refleja directamente en un incrementando o decremento de los valores de los estimadores.

Siguiendo con el análisis de la Figura 3.6, se puede observar que cuando entra la variable 6, esta variable provoca gran alteración principalmente en dos variables la variable 5 y la variable 9. Esta alteración consiste en el incremento del valor. De la misma manera la variable 7 cuyo valor, al entrar la variable 8 cambia dramáticamente.

A partir de lo establecido anteriormente se observa la importancia que tiene este tipo de gráficas en el análisis de datos. Por otra parte la importancia de ambos métodos de selección de variables (LASSO y LARS) requieren seleccionar el mejor subconjunto de variables. Esto quiere decir que tienen que escoger el número ideal de variables para conformar el modelo y así poder predecir los valores futuros de interés. A continuación se da una introducción de este procedimiento y los problemas que puede ocasionar cuando no se toma en consideración este aspecto.

3.6. Número de variables a introducir en el modelo

La selección del número de variables que deben introducirse en el modelo no debería suponer un problema, dado que es el objetivo del estudio y automáticamente las variables que son de mayor interés son identificadas fácilmente. Sin embargo, no

siempre es tan evidente y se plantea encontrar relaciones posibles para evaluar una respuesta de la forma más deseable. Puede ocurrir que aunque el objetivo de nuestro estudio esté bien definido, no dispongamos de información previa o simplemente no se cuenta con la información necesaria. Si a este hecho se le añade un número importante de variables, construir un modelo de regresión resulta en un proceso laborioso y complicado. Es conveniente incluir en el modelo solamente aquellas variables que consideremos especialmente importantes o influyentes e incluso variables de las que hayamos tenido conocimiento de su influencia a través de estudios previos. Además en muchas situaciones se desea conocer si todas las variables deben de entrar en el modelo de regresión, en el caso opuesto, se quiere saber que variables no deben entrar en el modelo de regresión.

Por otra parte no existe garantía que al incluir todas o pocas variables se obtendrán mejores resultados. Esto se debe a que algunas variables dan información útil, sin embargo otras variables redundantes proporcionan ruido no deseado. Por lo tanto es de vital importancia seleccionar una técnica adecuada.

A menudo suele suceder que en el conjunto de entrenamiento obtenemos excelentes resultados sin imaginar que para predecir los datos del conjunto de validación los resultados obtenidos no son los deseados. Un buen modelo se mide en términos del error obtenido en el conjunto de validación y no en el conjunto de entrenamiento. Sin embargo, las técnicas de medición de los errores que a menudo se utilizan dan resultados muy engañosos [Hawkins04]. Esto puede conducir al fenómeno de sobreajuste, es decir, un modelo puede adaptarse a los datos de entrenamiento excelentemente, pero se comportan muy mal en la predicción de nuevos datos, lo cual es muy común que ocurra con OLS. A continuación se analiza este problema también conocido como sobreaprendizaje.

3.6.1. SobreAprendizaje

En aprendizaje supervisado, el sobreaprendizaje (en ingles overfitting) se produce cuando un modelo describe el error aleatorio o ruido en lugar de la relación de entrada-salida [Alpaydin09]. El sobre aprendizaje generalmente ocurre cuando el modelo es excesivamente complejo, es decir, por lo general ocurre cuando un modelo obtiene la respuesta a partir de demasiadas variables. El propósito de evitar el problema de sobre ajuste en el conjunto de entrenamiento, subyace de la idea de obtener un modelo robusto, capaz de generalizar los datos del conjunto de validación que se busca conocer. El sobre-aprendizaje a menudo existe porque el criterio utilizado para entrenar el modelo no es el correcto. En particular, los datos para entrenar son incorrectos o los datos sólo generalizan unos casos sin poder tener un entrenamiento general y adecuado. Sin embargo, su eficiencia está determinada no solo por su desempeño en los datos de entrenamiento, sino por su capacidad para desempeñarse bien en los datos que no conoció en el conjunto de entrenamiento.

Un modelo simple de aprendizaje puede predecir perfectamente los datos de entrenamiento simplemente por la memorización de los datos de entrenamiento en su totalidad, pero tal modelo fallará drásticamente a la hora de hacer predicciones a cerca de nuevos datos o no conocidos en el conjunto de entrenamiento, ya que el modelo simple no ha aprendido a generalizar en absoluto.

Con el fin de evitar el sobre-ajuste es necesario utilizar técnicas adicionales como la validación cruzada [Arlot10] para dividir correctamente los datos y generalizar el modelo. A continuación se presenta la técnica de validación cruzada.

3.6.2. Validación cruzada

En la construcción de modelos [Alpaydin09], el objetivo principal debe ser la construcción de un modelo que prediga con mayor precisión nuevos datos que nos interese conocer. Sin embargo, en la práctica a menudo se comenten errores, por

ejemplo seleccionar el mejor modelo porque es el que presenta mejor ajuste en los datos de entrenamiento. Por lo general el uso de esta medida de error sin utilizar técnicas adecuadas en la división de los datos puede conducir a la selección de un modelo inferior al deseado.

Lo que se busca con validación cruzada es evitar precisamente ese sobre aprendizaje, la Figura 3.7 muestra la idea general de lo que se quiere evitar con validación cruzada.

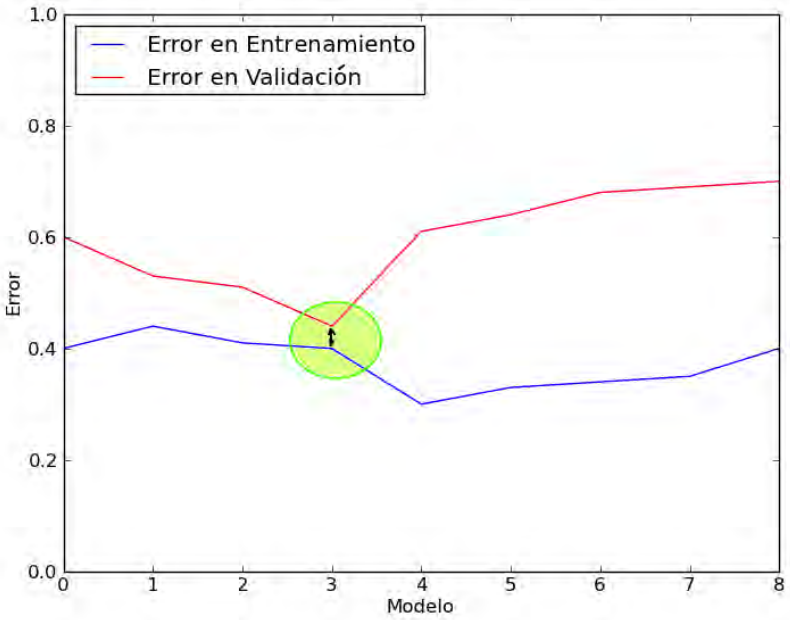


Figura 3.7: Error en el conjunto de entrenamiento y conjunto de validación para evitar el Sobre-Aprendizaje.

Frecuentemente el modelo seleccionado no es el ideal para predecir los datos, lo que se busca es generalizar dicho modelo y escoger el mejor modelo a partir de validación cruzada para predecir adecuadamente los datos, es decir obtener el menor error posible en el conjunto de validación. Como se puede observar claramente en la Figura 3.7, la representación del circulo verde nos muestra que el modelo ideal es el número 3 sin ser el punto con menor error en el entrenamiento. Es entonces nuestro principal

objetivo identificar el modelo 3 aplicando validación cruzada. El caso erróneo sería pensar que el mejor modelo es el 4 porque tiene el menor error en el conjunto de entrenamiento, es importante recordar que la linea roja es el error de la predicción de los datos del conjunto de validación no utilizado en el entrenamiento. Asimismo, la linea azul representa el error del modelo obtenido del conjunto de entrenamiento (ver la Figura 3.7) este proceso en ocasiones suele ser muy engañoso porque lo más lógico sería escoger el menor error.

Por otra parte, para evitar lo mencionado anteriormente la idea principal es dividir los datos en dos grupos. Un conjunto suficientemente grande para entrenamiento (aproximadamente 80 por ciento del total de los datos) y el resto para validar nuestro modelo al finalizar de obtener el mismo. Un grupo se usa para entrenar el modelo, el segundo grupo se usa para medir el error del modelo resultante. Por ejemplo, si tuviéramos 1,000 observaciones, podríamos usar 700 para construir el modelo y las restantes 300 muestras para medir el error del modelo. Como se puede ver en la Figura 3.8.

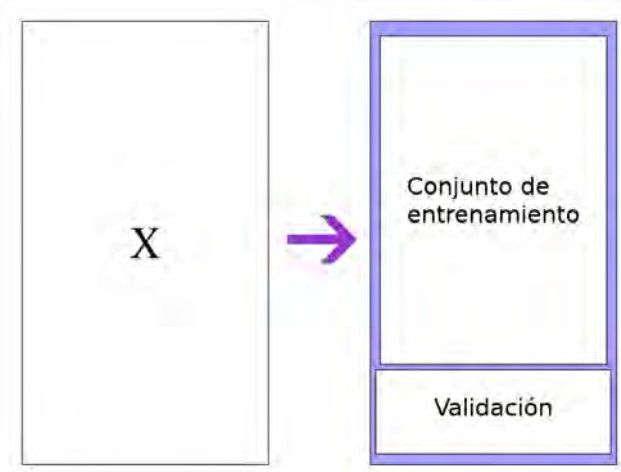


Figura 3.8: Conjunto de datos

La técnica de validación cruzada funciona dividiendo los datos de entrenamiento hasta en un conjunto de k bloques [Alpaydin09]. Por ejemplo, en el caso de 5-fold la

validación cruzada con 100 puntos de datos, se creará 5 bloques conteniendo cada uno 20 puntos de datos. Entonces la construcción del modelo y el proceso de estimación de error se repite 5 veces validando todos los datos en algún momento. Cada vez que los cuatro bloques se combinan forman un grupo de 80 puntos de datos los cuales son utilizados para entrenar el modelo. De la misma manera el quinto grupo de 20 puntos que no se utilizó para construir el modelo se utiliza para estimar el error de predicción de validación.

En el caso de la validación cruzada de 5-fold acabaría con 5 errores que se promedian para obtener una estimación más robusta del error de predicción de entrenamiento, es entonces cuando se elige el mejor modelo que representa ese menor error MSE por lo general los resultados arrojan que frecuentemente elige el mejor modelo, pero en muchas ocasiones no es así.

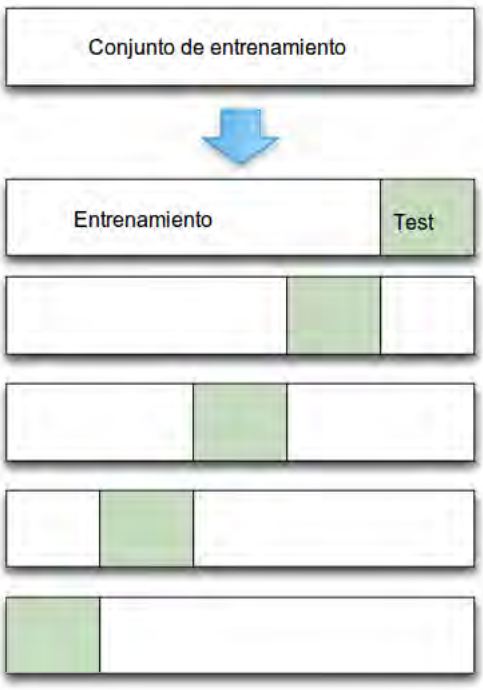


Figura 3.9: Validación cruzada 5-fold

La Figura 3.9 muestra el conjunto de entrenamiento (bloques en color blanco)

y a continuación se divide en 5 grupos de datos para el caso de 5-fold, tomando 4 bloques (blancos) para el conjunto de entrenamiento para determinar el modelo y el otro bloque (verde) para hace la predicción y obtener un error que será promediado por cada uno de los conjuntos en 5 ocasiones, cabe destacar que los conjuntos nunca se repiten y que el orden no importa es decir se puede hacer una mezcla e ir tomando datos de diferentes grupos siempre y cuando cumplan con la condición que no se deben repetir, la Figura 3.9 no es la única forma de dividir los datos.

3.7. Resumen

En este capítulo se presentaron los fundamentos de las técnicas basadas en la selección de variables para resolver el problema de regresión. Se presentaron los tres algoritmos que se utilizarán para pronosticar las 4,004 diferentes series de tiempo. Así como los algoritmos LASSO-puro, LASSO-umbral y LARS, cabe señalar que también se utilizará el modelo de OLS para corroborar los resultados.

Se presentó el vital papel que juega la selección de variables en problemas enormes. También se presenta la interpretación que se puede obtener a lo largo del modelado.

Se presenta la importancia que tiene utilizar técnicas de aprendizaje supervisado en especial validación cruzada. Esta técnica evita el sobreaprendizaje y ayuda a obtener respuestas estables por tal motivo juega un papel muy importante en este trabajo.

Capítulo 4

Resultados

Este capítulo muestra los resultados obtenidos de la predicción de las 4,004 series de tiempo diferentes con los algoritmos de selección de variables (LASSO y LARS). Los resultados se obtuvieron por cada una de las 4,004 series en el conjunto de entrenamiento y en el conjunto de validación. El modelo final se determina a partir de los datos del conjunto de entrenamiento, una vez establecido ese modelo, prosigue a pronosticar los datos de validación. Para finalizar se realiza una comparación entre los diferentes algoritmos para obtener el mejor método que modelo las series de tiempo.

4.1. Parámetros a optimizar

Los dos modelos de selección de variables LARS y LASSO tienen parámetros que se pueden optimizar con técnicas de aprendizaje supervisado. En este trabajo se aplicó validación cruzada 5-fold [Arlot10].

Parámetros a optimizar:

- LASSO-puro, el parámetro a optimizar es el valor de t.
- LASSO-umbral, el parámetro a optimizar es el valor de λ.
- LARS debe seleccionar la mejor etapa k.

Estos parámetros son de vital importancia, pues este valor determina el número de variables seleccionadas. A continuación se presentan los resultados obtenidos en la predicción de series de tiempo.

4.2. Resultados en la predicción de las 4,004 series de tiempo

Los algoritmos son LASSO-umbral, LASSO-puro, LARS y OLS. Para el modelo auto-regresivo se utilizó una ventana de 5 y para optimizar los parámetros de LASSO y LARS se utilizó validación cruzada con 5-fold.

Las 4,004 series de tiempo utilizadas son las proporcionada por las competencias M1 y M3, donde cada serie de tiempo tiene un conjunto de entrenamiento y otro de validación. Cinco valores de los estimadores β son los que se necesitan encontrar a partir de las técnicas de selección de variables y OLS.

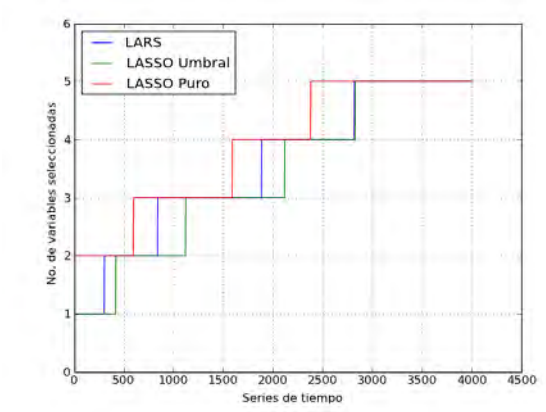


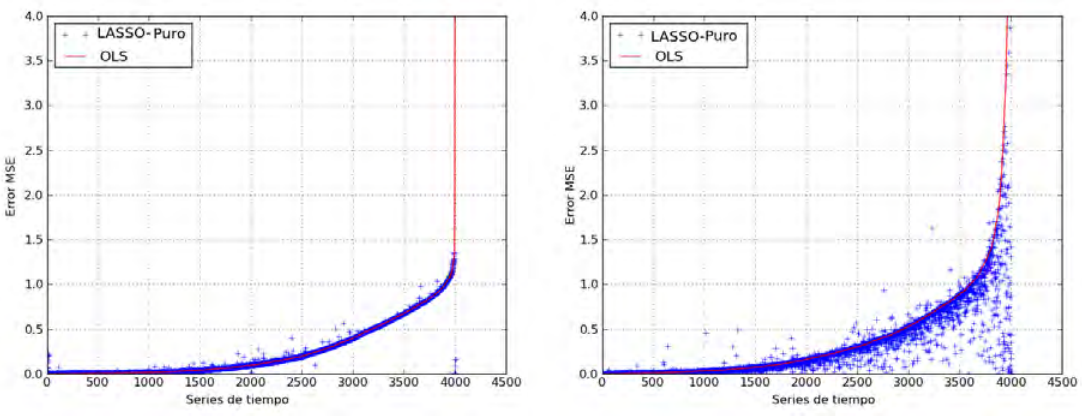
Figura 4.1: Número de variables seleccionadas en el modelo final para modelar cada una de las 4,004 series de tiempo diferentes.

La Figura 4.1 indica el número de variables seleccionadas en el modelo final para predecir cada una de las series de tiempo mediante LASSO y LARS. Esto quiere decir que sólo k variables son necesarias para obtener la respuesta $\hat{y}$. LASSO-puro tiene

más coeficientes diferentes a cero, mientras que el resto de los algoritmos muestran un número similar de variables. Esta selección de variables se realiza en el conjunto de entrenamiento. Es claro que en el modelado de mas de 3,000 series de tiempo no fue necesario introducir todas las variables en el modelo, donde basto introducir alrededor de tres variables (ver la Figura 4.1).

Otro detalle que llama la atención en LASSO-umbral y en LARS es que aproximadamente 300 series de tiempo utilizan únicamente una variable esto quiere decir que el resto de las variables no son necesarias para obtener una respuesta y esto se debe principalmente por dos razones; La primera porque son fuertemente correlacionadas entre si. La segunda porque los vectores son dependientes. Dichos algoritmos excluyen las variables que tienen citados problemas. Esto viene de la mano con la generación de modelos parsimoniosos.

Por otra parte, hasta este punto sólo conocemos el número de variables que se necesitan en cada metodología para predecir cada una de las series de tiempo, sin conocer como se comportará el modelo en la predicción tanto en el conjunto de entrenamiento como en el conjunto de validación, siendo este último conjunto el que interesa conocer y así medir el rendimiento del modelo.



(a) ***LASSO puro entrenamiento.*** (b) ***LASSO puro validación.***

Figura 4.2: Error obtenido en la predicción de las 4,004 series de tiempo en el conjunto de entrenamiento y conjunto de validación con el algoritmo LASSO puro.

Analicemos el comportamiento de LASSO-puro en el conjunto de entrenamiento y conjunto de validación (ver la Figura 4.2) la cual se lee de la siguiente manera:

- La linea roja es el error MSE obtenido en la predicción de las 4,004 series de tiempo por OLS, por tal razón el eje de la x termina en 4,004 por las series de tiempo y el eje de la y se gráfico hasta un error de 4.

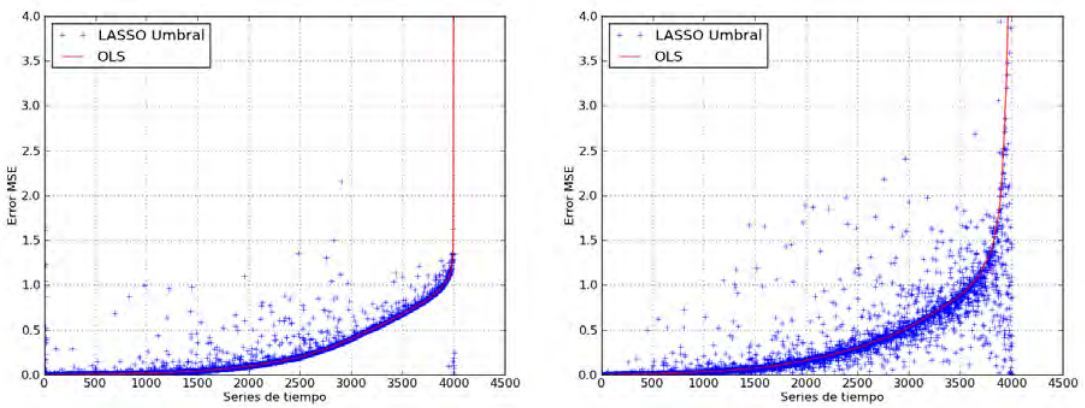
- Los puntos azules son los errores obtenidos con LASSO-puro, si los puntos azules están en la parte inferior de la linea roja quiere decir que LASSO-puro fue mejor en la predicción de los datos que OLS y si los puntos se encuentran en la parte superior de la linea roja, LASSO-puro fue peor modelo que OLS.

La Figura 4.2 representa el resultado del algoritmo LASSO puro, donde la Figura 4.2 a) representa el comportamiento en el conjunto de entrenamiento y la Figura 4.2 b) representa el comportamiento en el conjunto de validación. Como se puede observar en mencionadas gráficas, la linea roja es el error MSE obtenido de la predicción de cada una de las 4,004 series de tiempo con el algoritmo OLS. Esta linea se toma como referencia para todos los algoritmos, esta linea se obtuvo a partir del ordenamiento del menor error al mayor error obtenido en la predicción de los dos conjuntos de datos para cada una de las 4,004 series de tiempo. Todo esto se realizo con la finalidad de obtener una mejor interpretación en términos del mejor y peor algoritmo.

A partir de la Figura 4.2 se puede concluir lo siguiente; La Figura 4.2 a) principalmente la linea roja que presenta un pico muy elevado de error para algunas series de tiempo a partir de la serie número 3,980, esto se debe porque se presentó el problema $p >> n$ en aproximadamente 20 series de tiempo y LASSO-puro evita, siendo la primer ventaja. Para las series de tiempo bien acondicionadas el algoritmo LASSO-puro resulto ser muy parecido a la solución OLS. Esto se puede observar en la Figura 4.2 a), donde se puede ver una similitud muy estrecha, es decir, los puntos azules casi forman una linea roja igual a la de OLS. Hasta este momento podríamos afirmar que los resultados en el conjunto validación serían muy parecidos, pero los resultados ob-

tenidos por LASSO-puro dicen otra cosa (ver la Figura 4.2 b)). La solución obtenida con LASSO-puro resulto ser mucho mejor que la de OLS, esto se puede observar porque existen muchos puntos azules por debajo de la linea roja, es decir, lo que indica que tiene un error MSE menor al de OLS en la predicción del conjunto de validación. Se establece que con el simple hecho de reducir los vectores estimadores se evita el problema de sobre-aprendizaje que ocurre en OLS. Esto trae como consecuencia una reducción de los valores de los estimadores hacia cero e incluso llegan a ser iguales a cero. Es muy interesante ver que en la mayoría de las ocasiones el error MSE de la predicción con LASSO-puro se encuentran en la parte inferior de la linea de OLS, mientras que OLS en muy pocas series resulto ser mejor.

Se concluye que LASSO-puro es muy superior a OLS en la predicción de conjunto de validación y muy similares en la predicción del conjunto de entrenamiento. LASSO-puro presenta las siguientes tres propiedades; tiene una gran estabilidad, no tiene problemas para las matrices mal condicionadas (caso especial $p >> n$) y goza de un buen poder predictivo en el conjunto de validación convirtiéndose en una alternativa muy viable en el problema de regresión.



(a) *LASSO umbral entrenamiento.* **(b) *LASSO umbral validación.***

Figura 4.3: Error obtenido en la predicción de las 4,004 series de tiempo en el conjunto de entrenamiento y conjunto de validación con el algoritmo LASSO-umbral.

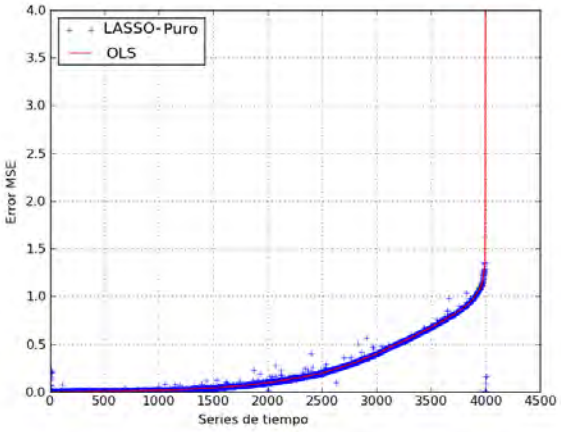
A continuación se presenta LASSO-umbral este método es puramente de selec-

ción de variables lo cual se logro aplicando el umbral y suponiendo que la matriz es ortogonal. Los resultados obtenidos por este algoritmo se pueden ver en la Figura 4.3.

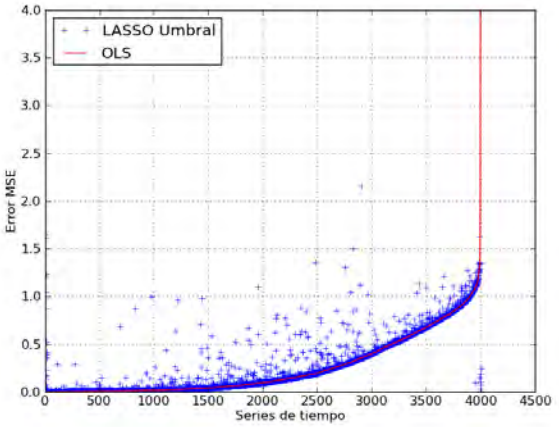
La Figura 4.3 representa el error MSE obtenido con el algoritmo LASSO-umbral y la interpretación de la gráfica es lo mismo que la Figura 4.3 anteriormente presentada. Asimismo se concluye que el algoritmo LASSO-umbral no es muy recomendable para matrices mal condicionadas, especialmente cuando los vectores no son perpendiculares es decir no se obtiene una matriz ortogonal ya que de esta suposición se obtiene el umbral. Este umbral contrae los coeficientes a cero, lo cual hace que el algoritmo simule hacer selección de variables cuando propiamente no lo es. Esto se demuestra con el algoritmo LASSO-puro donde si se obtienen resultados superiores a LASSO-umbral.

Los resultados de LASSO-umbral se deben a dos situaciones principalmente. Primero LASSO-umbral esta diseñado para matrices ortogonales y estos experimentos no fueron transformados mediante alguna técnica a matrices ortogonales, a consecuencia de este hecho se perdió poder predictivo, pero se gano en velocidad de cálculos ya que se resuelve este algoritmo con coordenada descendente. Segundo el problema pudo ocurrir por la mala elección de λ, aún siendo la equivalencia del valor de la cota superior t de LASSO-puro. El valor de λ se escoge a partir de una lista de valores, aunque la elección de este valor fue el equivalente a t no se obtuvieron los mismos resultados. Sin embargo LASSO-puro efectivamente realiza selección de variables haciendo modelos son sencillos, es decir, sólo ocupa las variables necesarias. Continuando con la interpretación de la Figura 4.3. La Figura 4.3 a) es el resultado de la predicción de las series de tiempo del conjunto de entrenamiento tal como se puede observar OLS obtiene mejores resultados que LASSO-umbral, sin embargo el resultado de la Figura 4.3 b) en el conjunto de validación no se esperaban tantos puntos por encima de la linea roja de OLS.

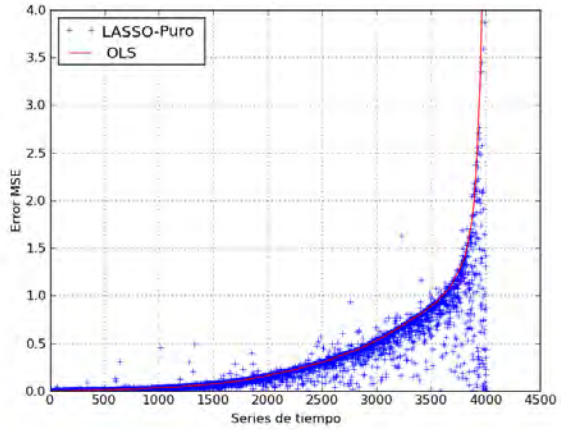
A continuación se presentan las gráficas de LASSO-umbral y LASSO-puro (ver la Figura 4.4) con la finalidad de ver la diferencia entre ambos algoritmos.



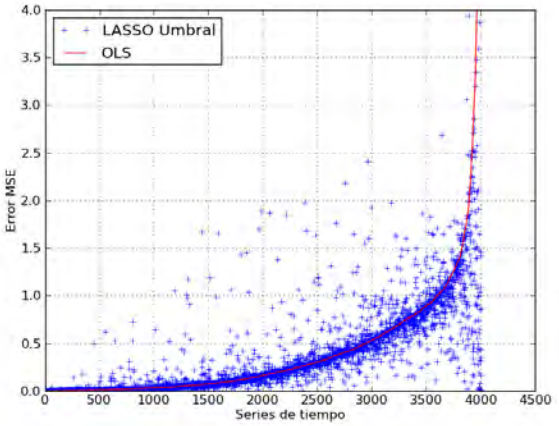
(a) ***LASSO puro entrenamiento.***



(b) ***LASSO umbral entrenamiento.***



(c) ***LASSO puro validación.***



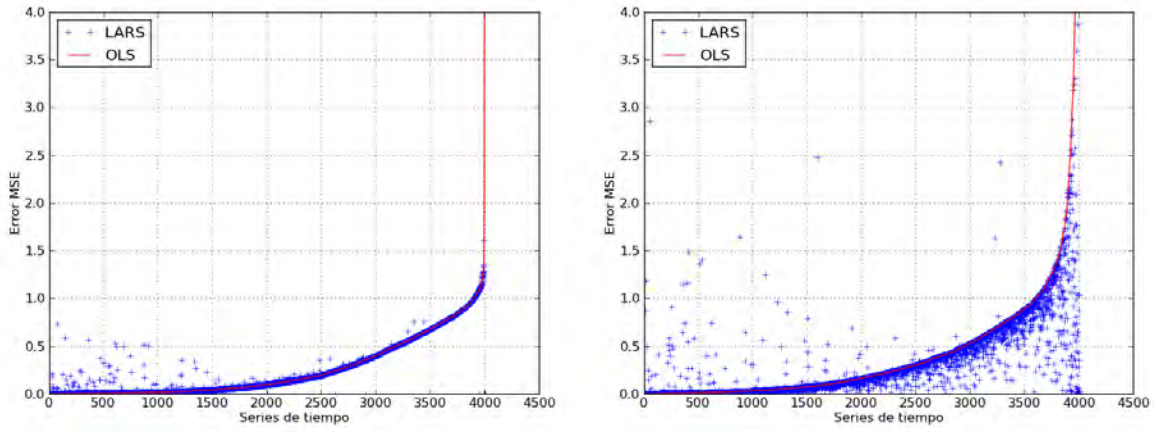
(d) ***LASSO umbral validación.***

Figura 4.4: Comparación de los dos algoritmos LASSO-puro y LASSO-umbral en la predicción de las 4,004 series de tiempo en el conjunto de entrenamiento y conjunto de validación.

Como se puede observar en la Figura 4.4 a) y la Figura 4.4 b) los dos algoritmos no obtienen resultados similares, es claro porque se pueden ver diferencias notables por la existencia de muchos puntos arriba de la linea roja para LASSO-umbral (ver la Figura 4.4 b)), sin embargo LASSO-puro tiene una cantidad menor de puntos en la parte inferior de la linea roja de referencia (ver la Figura 4.4 a)). Además la Figura 4.4 c) y la Figura 4.4 d) que representan los errores obtenidos en el conjunto de validación siguen la misma tendencia del conjunto de entrenamiento, es decir, no se obtienen

resultados superiores, pero LASSO-puro aún así supero a OLS. Se concluye que no es recomendable utilizar LASSO-umbral como modelo de selección de variables sin antes asegurarse de la ortogonalidad de las matrices.

Por otra parte, a continuación se presenta el algoritmo LARS, el cual obtiene resultados similares o superiores a LASSO-puro en términos del menor error en validación y se puede observar en la Figura 4.5.



(a) *LARS entrenamiento.* **(b) *LARS validación.***

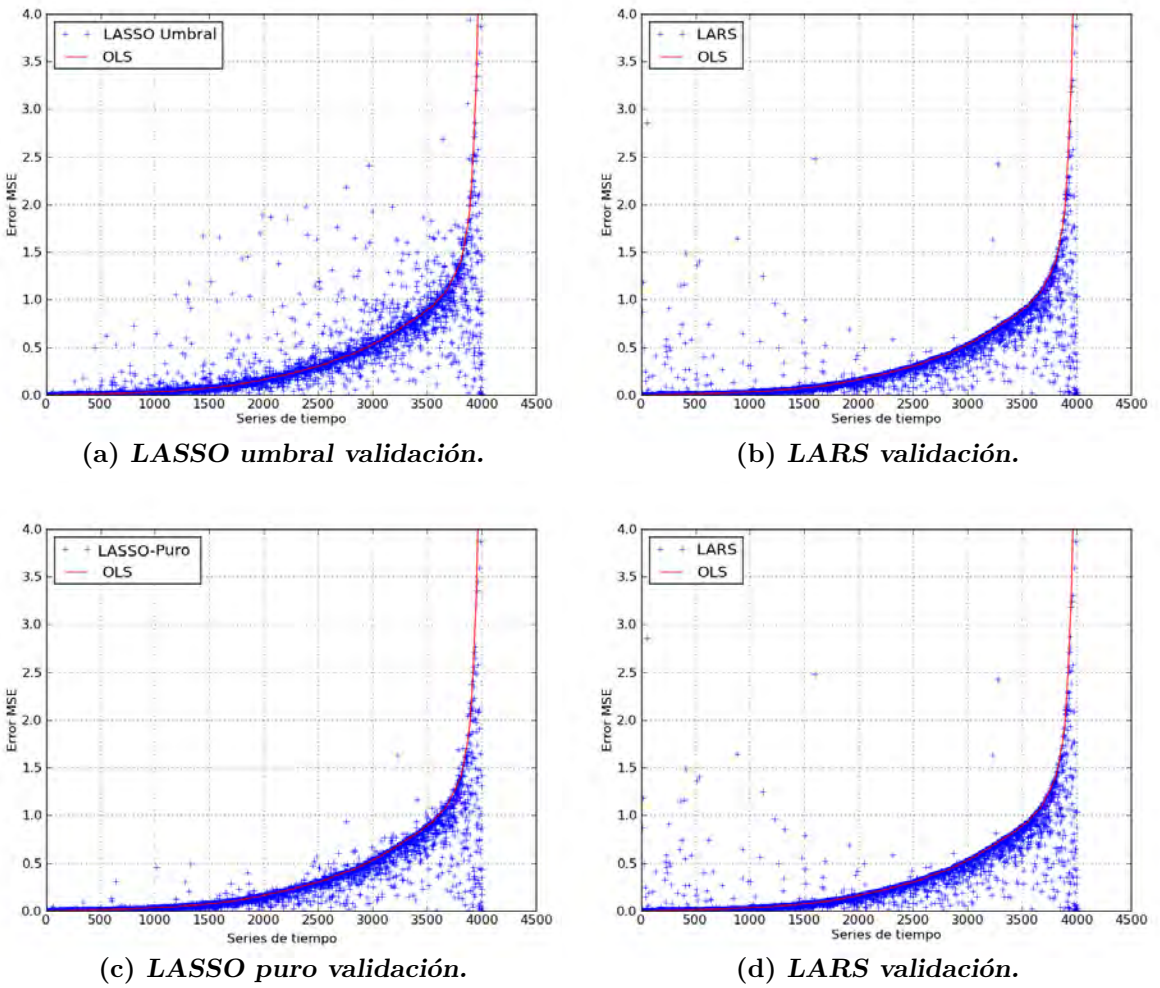
Figura 4.5: Error obtenido en la predicción de las 4,004 series de tiempo en el conjunto de entrenamiento y conjunto de validación con el algoritmo LARS.

LARS es en esencia el algoritmo de selección de variables que mejores resultados obtiene en comparación a LASSO-puro como se puede observar en la Figura 4.5. La Figura 4.5 a) muestra que LARS obtiene errores similares que OLS o peores en el conjunto de entrenamiento, es decir, forma una linea muy parecida a la linea roja de OLS y existen muchos puntos por debajo de la linea de OLS. En la Figura 4.5 b), la predicción de las series de tiempo en el conjunto de validación los resultados son aún mejores que los obtenidos por el LASSO-puro.

A pesar que existen puntos en la parte superior de la linea roja existe una gran cantidad de puntos en la parte inferior de la linea roja. Además las series de tiempo mal acondicionadas que representan el sobre-pico de la linea roja por la predicción

incorrecta de OLS, se ve que LARS supera ese problema excluyendo las variables innecesarias e introduciendo al modelo sólo las requeridas por el algoritmo de LARS. Estos puntos se pueden ver en la parte de abajo de dicho sobre-pico (ver a partir de la serie No. 3980). A partir de estos resultados LARS se establecen las siguientes propiedades; Genera modelos sencillos haciendo selección de variables. Los resultados por lo general son precisos. La estabilidad es otra propiedad de LARS.

A continuación se discute las diferencias entre los algoritmos LARS y LASSO.



(a) *LASSO umbral validación.* **(b) *LARS validación.***

(c) *LASSO puro validación.* **(d) *LARS validación.***

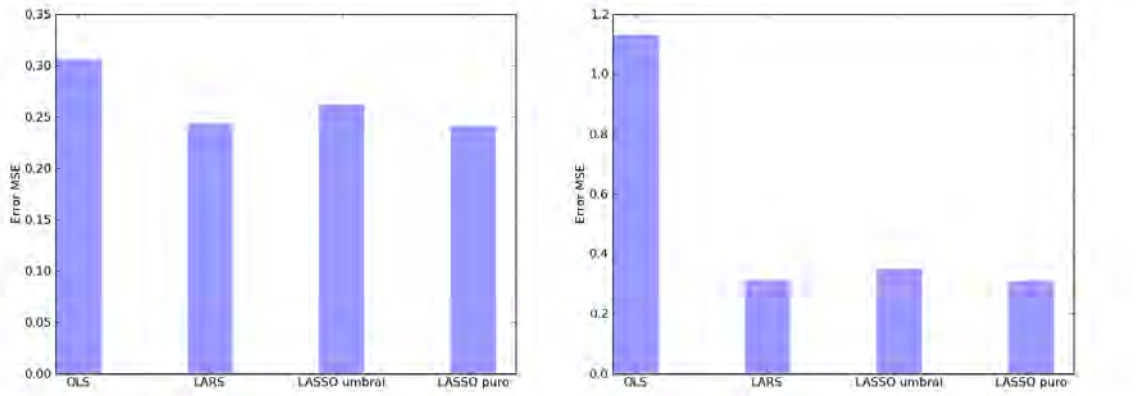
Figura 4.6: Comparación de los dos algoritmos de LASSO con LARS en la predicción de las 4,004 series de tiempo en el conjunto de de validación.

La Figura 4.6 a) y la Figura 4.6 b) representan la comparación de los algoritmos LASSO-umbral y LARS. Se puede observar una gran diferencia entre ambos especialmente interesante porque los dos algoritmos modelan las series de tiempo mediante selección de variables (ver la Figura 4.1). Como se ha venido comentando los resultados de LASSO-umbral carece de un buen poder predictivo, caso contrario sucede con el algoritmo de LARS que es un algoritmo que introduce una variable al modelo por cada etapa del proceso similar a LASSO-umbral. LARS obtiene los mejores resultados en este trabajo como se puede observar en la Figura 4.6 c) y la Figura 4.6 d) donde supera al algoritmo LASSO-puro, sin embargo este resultado se puede discutir porque LASSO-puro es mas estable que LARS.

A partir de las comparaciones anteriores se establecen las siguientes conclusiones: primero LARS es el mejor algoritmo de acuerdo a la mejor precisión en comparación a los tres algoritmos, es decir, el error MSE obtenido en la predicción de las series de tiempo en el conjunto de validación fue menor en mayor número de series de tiempo que LASSO-puro el algoritmo que se posiciona en segundo lugar. A pesar de esta superioridad el algoritmo más estable fue LASSO-puro. Esto se puede ver por la cantidad de puntos que se pueden ver en la parte superior de la linea roja sobre todo en el rango de las predicciones de las series de tiempo cero a la dos mil los puntos están muy dispersos y con un error considerable. Por otro lado, el algoritmo LASSO-puro no presenta ese problema siendo por este hecho un algoritmo más estable. Sin embargo LARS resulta ser el mejor algoritmo para este trabajo sobre todo porque tiene un balance en la sencillez del modelo y buen poder predictivo.

Por otra parte se realiza un promedio de los errores MSE para cada algoritmo (ver la Figura 4.7), en el conjunto de entrenamiento y en el conjunto de validación quitando las series de tiempo donde OLS tiende a tener problemas por dependencia de los vectores.

Para finalizar con este apartado se busca obtener el mejor algoritmo en la predicción de las 4,004 series de tiempo, esta comparación se realiza respecto al error



(a) ***Error promedio en entrenamiento.*** **(b)** ***Error promedio en validación.***

Figura 4.7: Error promedio en la predicción de las series de tiempo en el conjunto de entrenamiento y conjunto de validación de los 4 algoritmos.

en la predicción en el conjunto de validación. Esto quiere decir que si el algoritmo tiene menor error en la predicción de tal serie de tiempo, sumara un punto a su favor mientras que los otros tres algoritmos suman cero y así sucesivamente hasta contar las 4,004 series de tiempo. La Tabla 4.1 muestra el resultado de tal comparación con todos los algoritmos. Los resultados muestran que LARS fue el mejor algoritmo seguido de LASSO-puro lo cual nos indica que a pesar de no ser precisamente un método de selección de variable obtiene muy buenos resultados.

Tabla 4.1: Comparación de los algoritmos en términos del menor error en la predicción de las 4,004 series de tiempo en el conjunto de validación

Modelo	Conjunto de validación
LARS	1726
LASSO puro	1020
LASSO umbral	737
OLS	521

La Figura 4.8 representa la comparación obtenida de la Tabla 4.1, donde se puede ver claramente los porcentajes de cada uno de los 4 algoritmos.

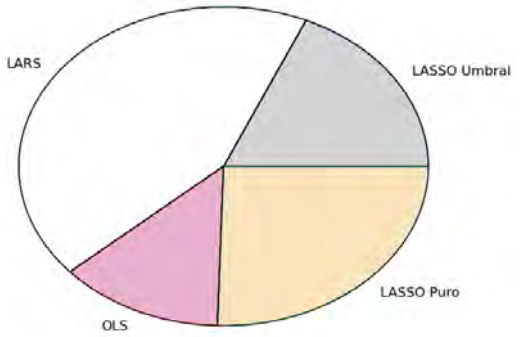


Figura 4.8: Gráfica de los resultados de la Tabla 4.1.

4.3. Resumen

Este capítulo muestra los resultados de las predicciones de las 4,004 series de tiempo diferentes mediante los algoritmos OLS, LASSO-puro, LASSO-umbral y LARS. Este apartado comienza con los parámetros de ajuste que requieren cada uno de los algoritmos propuestos en este trabajo. Es una gran sorpresa el resultado obtenido por LASSO-puro por el problema que implica no reducir exactamente a cero el valor de los coeficientes cuando sólo tienden a ser muy cercanos a cero. Es muy interesante el resultado obtenido tanto en el conjunto de entrenamiento como en el conjunto de validación cuando parecía tender al mismo problema de OLS, esto quiere decir que se evito ese problema sin la necesidad de hacer selección de variables.

Se concluye que para estas 4,004 series de tiempo los resultados de los tres algoritmos propuestos superan a OLS en términos de la comparación del menor error MSE obtenido en la predicción en el conjunto de validación. Para finalizar se realiza una comparación para determinar el mejor algoritmo en la predicción del conjunto de validación. LARS es el mejor algoritmo de los cuatro utilizados para pronosticar las 4,004 series de tiempo.

Capítulo 5

Conclusiones

5.1. Conclusiones

Las técnicas de selección de variables han sido capaces de modelar 4,004 series de tiempo diferentes. En este trabajo se ha mostrado experimentalmente que LARS y LASSO son técnicas automáticas de selección de variables y que producen modelos sencillos. Se comparó LARS y LASSO (puro y umbral) en el problema de la selección de variables. Los resultados muestran que LARS y LASSO puro son los mejores modelos en términos del menor error obtenido en la predicción del conjunto de validación. OLS es el mejor algoritmo en la predicción de las 4,004 series de tiempo en el conjunto de entrenamiento y LASSO-umbral resulto ser el peor algoritmo en el conjunto de entrenamiento. OLS fue el modelo que peor ajusto a los datos en el conjunto de validación. El resultado de LASSO-puro tiene un rendimiento similar a OLS en la predicción del conjunto de entrenamiento y en la predicción del conjunto de validación obtiene resultados muy superiores a OLS. El algoritmo LASSO-puro resalta la importancia de la reducción de los valores del vector de estimadores debido a la formulación de LASSO como problema de optimización.

Por otra parte LARS y LASSO son dos enfoques de selección de variables muy interesantes, que ofrecen velocidad, facilidad de interpretación, estabilidad en la pre-

dicción, una buena presentación gráfica del seguimiento de los diferentes valores que van tomando los estimadores a lo largo del modelado. Además estos algoritmos pueden ser combinados con técnicas de aprendizaje supervisado.

Otro aspecto muy importante es que las técnicas de LASSO y LARS resuelven el problema mal condicionadas ($p >> n$) [Buhlmann06]. Por último estas técnicas (OLS, LARS y LASSO-umbral) se encuentran disponibles en dominios públicos y lo encontramos en python-Sklearn [Pedregosa11] y R [R Development Core Team08].

5.2. Trabajos futuros

Los modelos de selección de variables presentados en este trabajo son técnicas adecuadas en el modelado de datos y así se comprobó en este estudio, sin embargo, estas técnicas son muy interesantes cuando el estudio involucra muchas variables. En este trabajo especialmente en la predicción de las 4,004 series de tiempo sólo se modelo para 5 variables, sin embargo, si se comprobó el funcionamiento para un caso especifico en ingeniería eléctrica que se realizo en conjunto con el maestro Isidro Lazaro Castillo donde nos proporciono una matriz de 250 variables y más de 1,000 observaciones. Los resultados arrojaron que no era necesario introducir las 250 variables al modelo, donde LARS únicamente utilizo 64 variables en el modelo. Este modelo evito muchos problemas, como el sobre-ajuste principalmente, la interpretación, la inestabilidad y sobre todo disminuyo considerablemente la complejidad.

Por otra parte, la solución con OLS presentaba sobre-ajuste con los datos de entrenamiento, es decir, la respuesta obtenida en el conjunto de entrenamiento superaba la de LARS. La predicción en el conjunto de validación la respuesta no era la deseable. A partir de esto se plantea como trabajos futuros lo siguiente:

- Dar seguimiento al trabajo realizado con el Maestro Isidro Lazaro Castillo y probar con otros experimentos donde la predicción fue resuelta con OLS.

- El segundo trabajo a futuro es comparar LASSO y LARS con técnicas más robustas y conocidas por su buen poder predictivo. Como pueden ser el suavizado exponencial (ver [Hyndman02]), Modelos ARIMA (Procesos Auto-regresivos integrados con promedios móviles, ver [Shanmugam97]), tendencia y estacionalidad por componentes (ver [De Livera11]) e incluso con los nuevos enfoques de LASSO y LARS vistos en el estado de arte. Todos estos algoritmos forman parte de las librerias de python-Sklearn [Pedregosa11] y R [R Development Core Team08].

- Como tercer trabajo sería resolver ARIMA con LARS o LASSO, actualmente se resuelve con OLS.

- Utilizar LARS como un algoritmo de clasificación.

- Aplicar LARS como un algoritmo de clasificación.

- Aplicar LARS y LASSO-puro en la industria (caso especial seguimiento a fallas y detección de perdida de espesor en la empresa de acero TERNIUM).

Referencias

[Aggarwal99] Aggarwal, R., Inclan, C., y Leal, R. Volatility in emerging stock markets. *The Journal of Financial and Quantitative Analysis*, 34(1):33–55, 1999. URL `http://www.jstor.org/stable/2676245`

[Alpaydin09] Alpaydin, E. *Introduction to Machine Learning*. The MIT Press, second edition, december 2009. ISBN 026201243X.

[Andreou02] Andreou, E. y Ghysels, E. Detecting multiple breaks in financial market volatility dynamics. *Journal of Applied Econometrics*, 17(5):579–600, 2002.

[Anstreicher90] Anstreicher, K. M., den Hertog, D., Roos, C., y Terlaky, T. A long step barrier method for convex quadratic programming. *ALGORITHMICA*, 10:365–382, 1990.

[Arlot10] Arlot, S. y Celisse, A. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010. doi:10.1214/09-SS054.

[Bai98] Bai, J. y Perron, P. Estimating and testing linear models with multiple structural changes. *Econome-*

trica, 66(1):pp. 47–78, 1998.
URL `http://www.jstor.org/stable/2998540`

[Bardet12] Bardet, J.-M. y Chakry Kengne, W. Monitoring procedure for parameter change in causal time series. *ArXiv e-prints*, 2012.

[Box86] Box, G. E. y Meyer, R. An analysis for unreplicated fractional factorials. *Technometrics*, 28:11–18, 1986. ISSN 0040-1706. doi:10.2307/1269599.

[Boyd04] Boyd, S. y Vandenberghe, L. *Convex Optimization*. Cambridge University Press, mar. 2004. ISBN 0521833787.

[Braun00] Braun, J. V., Braun, R. K., y Muller, H. G. Multiple changepoint fitting via quasilikelihood, with application to dna sequence segmentation. *Biometrika*, 87(2):301–314, 2000.
URL `http://www.jstor.org/stable/2673465`

[Buhlmann06] Buhlmann, P. Boosting for high-dimensional linear models. *The Annals of Statistics*, 34(2):559–583, abr. 2006. ISSN 0090-5364. doi: 10.1214/009053606000000092. Mathematical Reviews number (MathSciNet): MR2281878; Zentralblatt MATH identifier: 1095.62077.

[Campbell09] Campbell, S. L. y Meyer, C. D. *Generalized Inverses of Linear Transformations*. SIAM, mar. 2009. ISBN 9780898716719.

[Chen98] Chen, S. S., Donoho, D. L., Michael, y Saunders, A. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998.

[Curtisy05] Curtisy, F. y Nocedalz, J. Steplength selection in interior-point methods for quadratic programming, 2005.

[De Livera11] De Livera, A. M., Hyndman, R. J., y Snyder, R. D. Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, 106(496):1513–1527, dic. 2011. ISSN 0162-1459, 1537-274X. doi: 10.1198/jasa.2011.tm09771.

[Delgado-Trejos09] Delgado-Trejos, E. Editorial. *Tecno Lógicas*, 0(22), 2009.

[Ding08] Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., y Keogh, E. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proc. VLDB Endow.*, 1(2):1542–1552, ago. 2008. ISSN 2150-8097.

[Donoho06] Donoho, D. L. y Elad, M. On the stability of the basis pursuit in the presence of noise. *Signal Process.*, 86(3):511–532, mar. 2006. ISSN 0165-1684. doi: 10.1016/j.sigpro.2005.05.027.

[Draper81] Draper, N. y Smith, H. *Applied Regression Analysis*. Wiley, New York, NY, 2ª ed$^{\underline{\text{ón}}}$., 1981.

[Efron04] Efron, B. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004. ISSN 0090-5364. doi: 10.1214/009053604000000067.

[Friedman10] Friedman, J., Hastie, T., y Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1–22, 2010. ISSN 1548-7660. PMID: 20808728 PMCID: PMC2929880.

[Gershenfeld93] Gershenfeld, N. A. y Weigend, A. S. The future of time series: Learning and understanding. *En* A. Weigend y N. Gershenfeld, eds., *Time Series Prediction: Forecasting the Future and Understanding the Past*, págs. 1–70. Addison-Wesley, 1993.

[Gluhovsky11] Gluhovsky, I. Multinomial least angle regression with application to web personalization. *Proceedings of the World Congress on Engineering*, 1, 2011.

[Gondzio12] Gondzio, J. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587–601, 2012.

[Gratton11] Gratton, S., Toint, P., y Troltzsch, A. An active set trust-region method for derivative-free nonlinear bound-constrained optimization. *Optimization Methods and Software*, 26(4-5):875–896, 2011. Preliminary version available as CERFACS Technical Report TR/PA/10/70.

[Gupta11] Gupta, A., Bhartiya, S., y Nataraj, P. S. V. A novel approach to multiparametric quadratic programming. *Automatica*, 47(9):2112–2117, sep. 2011. ISSN 0005-1098. doi:10.1016/j.automatica.2011.06.019.

[Hastie01] Hastie, T., Tibshirani, R., y Friedman, J. H. *The elements of statistical learning: data mining, inference, and prediction: with 200 full-color illustrations*. New York: Springer-Verlag, 2001.

[Hawkins04] Hawkins, D. M. The Problem of Overfitting. *Journal of Chemical Information and Computer Sciences*, 44(1):1–12, 2004.

[Hayashi00] Hayashi, F. Bekaert-hodrick93. Instructional Stata datasets for econometrics bhodrick93, Boston College Department of Economics, 2000.

[Hoerl70] Hoerl, A. E. y Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[Hyndman02] Hyndman, R. J., Koehler, A. B., Snyder, R. D., y Grose, S. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18(3):439–454, jul. 2002. ISSN 0169-2070. doi:10.1016/S0169-2070(01)00110-8.

[Jones] Jones, E., Oliphant, T., Peterson, P., et al. SciPy: Open source scientific tools for Python, 2001–.
URL `http://www.scipy.org/`

[Lin07] Lin, Y. y Zhang, H. H. Component selection and smoothing in multivariate nonparametric regression. *Ann. Stat.*, 34(math.ST/0702659. IMS-AOS-AOS0201. 5):2272–2297, Feb 2007.

[Makridakis82] Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E., y Winkler, R. The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of Forecasting*, 1(2):111–153, 1982. ISSN 1099-131X. doi:10.1002/for.3980010202.

[Makridakis00] Makridakis, S. y Hibon, M. The m3-competition: results, conclusions and implications. *International Journal of Forecasting*, 16(4):451–476, oct. 2000. ISSN 0169-2070. doi:10.1016/S0169-2070(00)00057-1.

[Mitchell97] Mitchell, T. M. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1ª ed$^{\underline{\text{ón}}}$., 1997. ISBN 0070428077, 9780070428072.

[Park05] Park, T. y Casella, G. The bayesian lasso. Inf. téc., 2005.

[Park08] Park, T. y Casella, G. The bayesian lasso. *Journal of the American Statistical Association*, 103:681–686, 2008.

[Pedregosa11] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer,

P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., y Duchesnay, E. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[Pötscher09] Pötscher, B. M. y Leeb, H. On the distribution of penalized maximum likelihood estimators: The lasso, scad, and thresholding. *J. Multivar. Anal.*, 100(9):2065–2082, oct. 2009. ISSN 0047-259X. doi: 10.1016/j.jmva.2009.06.010.

[Powell98] Powell, M. J. D. Direct search algorithms for optimization calculations. 7:287–336, 1998. ISSN 0962-4929 (print), 1474-0508 (electronic). doi: http://dx.doi.org/10.1017/S0962492900002841.

[R Development Core Team08] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.
URL `http://www.R-project.org`

[Raman09] Raman, S., Fuchs, T. J., Wild, P. J., Dahl, E., y Roth, V. The bayesian group-lasso for analyzing contingency tables. *En Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, págs. 881–888. ACM, New York, NY, USA, 2009. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553487.

[Roth08] Roth, V. y Fischer, B. The group-lasso for generalized linear models: uniqueness of solutions and efficient algorithms. *En Proceedings of the 25th international conference on Machine learning*, ICML '08, págs. 848–855. ACM, New York, NY, USA, 2008. ISBN 978-1-60558-205-4. doi:10.1145/1390156.1390263. URL `http://doi.acm.org/10.1145/1390156.1390263`

[Ruhe92] Ruhe, A. Rational krylov algorithms for nonsymmetric eigenvalue problems, ii: Matrix pairs, 1992.

[Shanmugam97] Shanmugam, R. Introduction to time series and forecasting. *Technometrics*, 39(4):426–426, 1997.

[Singaraju12] Singaraju, D., Elhamifar, E., Tron, R., Yang, A. Y., y Sastry, S. S. On the lagrangian biduality of sparsity minimization problems. *CoRR*, abs/1201.3674, 2012.

[Tibshirani94] Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[Tzu-Kuo Huang11] Tzu-Kuo Huang, J. S. Learning auto-regressive models from sequence and non-sequence data. *En NIPS 2011*. 2011.

[Vapnik95] Vapnik, V. N. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0-387-94559-8.

[Weisberg04] Weisberg, S. Discussion of "least angle regression" by

efron et al. *The Annals of Statistics*, págs. 490–494, 2004.

[Weisberg05] Weisberg, S. *Applied Linear Regression*. Wiley, Hoboken NJ, 3ª ed$^{\underline{\text{ón}}}$., 2005.
URL `http://www.stat.umn.edu/alr`

[Wikipedia11] Wikipedia, S. *Least Squares: Ordinary Least Squares, Linear Least Squares, Non-Linear Least Squares, Coefficient of Determination, Total Least Squares*. General Books, 2011. ISBN 9781233136537.
URL `http://books.google.com.mx/books`

[Wu08] Wu, T. T. y Lange, K. Coordinate descent algorithms for lasso penalized regression. mar. 2008.
URL `http://arxiv.org/abs/0803.3876`

[Wu09] Wu, T. T., Chen, Y. F., Hastie, T., Sobel, E., y Lange, K. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721, mar. 2009. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp041.

[Yang10] Yang, A. Y., Ganesh, A., Zhou, Z., Sastry, S., y Ma, Y. A review of fast l1-minimization algorithms for robust face recognition. *CoRR*, abs/1007.3753, 2010.

[Ye09] Ye, L. y Keogh, E. Time series shapelets: a new primitive for data mining. *En Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, págs.

947–956. ACM, New York, NY, USA, 2009. ISBN 978-1-60558-495-9. doi:10.1145/1557019.1557122. URL `http://doi.acm.org/10.1145/1557019.1557122`

[Yuan06] Yuan, M., Yuan, M., Lin, Y., y Lin, Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.

[Zou05a] Zou, H. y Hastie, T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.

[Zou05b] Zou, H. y Hastie, T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.