Integrantes:

Arturo Zuñiga Blanco

Jaime Gómez Marín

Jhordan Calderón Vega

Renzo Arenaza Ramos

n Aplicada

RESUMEN

El presente trabajo de investigación tiene como objetivo principal dar a conocer los conceptos teóricos y prácticos de la regresión Ridge, y como una consecuencia de esta la regresión Lasso, estos métodos son conocidos como técnicas de regularización que surgen como una alternativa para datos que presentan multicolinealidad en los predictores, los cuales pueden perjudicar la confiabilidad del modelo, es así que con estas técnicas se busca un modelo que se ajuste bien a los datos y que, a la vez, sea posible buscar un equilibrio entre bondad de ajuste y sencillez.

Los modelos de regresión lineal múltiple requieren de supuestos, como linealidad, homocedasticidad, independencia, normalidad en los errores y que las variables explicativas se obtengan sin errores de medida y si alguno de estos supuestos no se cumple, los estimadores por Mínimos Cuadrados Ordinarios no serían los más adecuados a utilizar.

Tabla de contenido

1. DESARROLLO TEÓRICO

1.1. Introducción

En muchas situaciones se dispone de un conjunto grande de posibles variables explicativas, una posible pregunta sería saber si todas las variables deben entrar en el modelo de regresión y, en caso negativo, saber qué variables deben entrar y cuáles no. en otras palabras, debemos seleccionar un subconjunto de variables entre todas las variables candidatas a ser explicativas de la variable dependiente, subconjunto que resulte suficientemente explicativo.

En general, si se incluyen cada vez más variables en un modelo de regresión, el ajuste a los datos mejora, aumenta la cantidad de parámetros a estimar, pero disminuye su precisión individual (mayor varianza) y, por tanto, la de la función de regresión estimada, produciéndose un sobreajuste. Por el contrario, si se incluyen menos variables de las necesarias en el modelo, las varianzas se reducen, pero los sesgos aumentarán obteniéndose una mala descripción de los datos.

Por otra parte, algunas variables predictoras pueden perjudicar la confiabilidad del modelo, especialmente si están correlacionadas con otras.

De esta manera, el objetivo de los métodos de selección de variables es buscar un modelo que se ajuste bien a los datos y que, a la vez, sea posible buscar un equilibrio entre bondad de ajuste y sencillez.

En la práctica, no obstante, la selección del subconjunto de variables explicativas de los modelos de regresión se deja en manos de procedimientos más o menos automáticos.

1.2. Modelo de regresión lineal

El modelo muestral

$$y = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p + \varepsilon$$

Modelo matricial

$\underline{Y} = X\underline{\beta} + \underline{\varepsilon}$ Donde β es un vector p-dimensional

Los coeficientes de regresión $\beta_0, \beta_1, ..., \beta_p$ son desconocidos, luego deben ser estimados. Para realizar un análisis de regresión lineal múltiple se realizan las siguientes consideraciones sobre los datos:

- Linealidad
- Homocedasticidad
- Independencia
- Normalidad
- Las variables explicativas se obtienen sin errores de medida

Si los datos cumplen estas hipótesis entonces pasamos a estimar los coeficientes por el método de los mínimos cuadrados, MCO, ya que los estimadores de los parámetros son insesgados y tienen mínima varianza.

1.2.1. Supuestos en la regresión lineal

Además de suponer que $\underline{Y} = X\beta + \underline{\varepsilon}$ y que la matriz X es no aleatoria, requeriremos lo siguiente:

1.- $E[\underline{\varepsilon}] = \underline{0}$

2.- $E[\underline{\varepsilon\varepsilon}'] = \sigma^2 I_n$

3.- $rango(X) = p < N$

El supuesto 1) no implica pérdida de generalidad ni supone ninguna restricción, al menos en el caso en que X tiene entre sus columnas una cuyos valores sean constantes (y esto suele suceder; típicamente, la primera columna está formada por “unos”).

El supuesto 2), bastante más restrictivo, dado que requiere que las perturbaciones sean incorreladas (covarianzas cero) y homocedásticas (de idéntica varianza).

El supuesto 3) simplemente fuerza la independencia lineal entre las (p) columnas de X. El requerimiento $N > p$ excluye de nuestra consideración el caso $N = p$, pues entonces $\underline{y} = X\hat{\beta}$ es un sistema de ecuaciones lineales determinado, y tiene siempre solución para algún vector $\hat{\beta}$ que hace los residuos nulos. Las estimaciones del vector β se obtendrían entonces resolviendo dicho sistema.

1.2.2. Estimación por mínimos cuadrados

El método MCO tiene como objetivo minimizar la suma de los cuadrados de los residuos:

$$RSS(\beta) = \|Y - X\beta\|_2^2 = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij} \right)^2$$

El estimador de β por MCO viene dado por:

$$\hat{\beta}^{MCO} = \left(X'X\right)^{-1} X'\underline{y}\ ,\ \varepsilon \square N\left(0, \sigma^2 I_n\right),\ \hat{\beta}^{MCO} = N_p\left(\beta, \sigma^2\left(X'X\right)^{-1}\right)$$

Suponemos que $rango(X) = p$, entonces $(X'X)$ es de rango completo y, por tanto, posee inversa. Este estimador es único en el caso que las columnas de X formen un conjunto linealmente independiente, y bajo el supuesto anterior de normalidad $\varepsilon \square N\left(0, \sigma^2 I_n\right)$, $\hat{\beta}^{MCO}$ es el estimador de mínima varianza de β en la clase de estimadores lineales e insesgados (teorema de Gauss-Markov).

1.2.3. Problemas con MCO

Hay dos razones por las que el método de mínimos cuadrados podría no ser adecuado para estimar modelos con variables no relevantes, es decir, con poca capacidad predictora. Estas son:

- **Baja precisión en las predicciones**. El estimador a menudo presenta poco sesgo pero gran varianza, lo cual se traduce en un pobre poder predictivo sobre nuevas observaciones. La existencia de al menos, una variable que pudiese ser expresada como combinación lineal del resto, conduce a que el determinante de $(X'X)$ sea nulo y por tanto no exista su inversa.

Si no hay variables que sean combinación lineal de otras, pero están fuertemente correlacionadas, provoca inestabilidad en la solución del estimador dado que el determinante de $(X'X)$ será muy cercano a cero. (Rojo Abuín, 2007)

- **Falta de interpretabilidad.** Si se utiliza un gran número de predictores (necesario para tener bajo sesgo ante un problema más o menos complejo), sería deseable determinar un pequeño subconjunto de éstos con fuerte poder explicativo y predictivo, ya que con $p \square n$ el estimador no estará bien definido.

Esta desventaja de MCO también está vinculada a la existencia de predictores fuertemente correlacionadas.

Al estimar los parámetros de regresión por el método de mínimos cuadrados ordinarios, puede que alguna de estas estimaciones sea “casi” cero y por tanto, la variable correspondiente a dicho coeficiente tendría muy poca influencia en el modelo, sin embargo, es poco común que estas estimaciones lleguen a tomar exactamente el valor cero. Por tanto, no nos sirve como método de selección de variables. De este modo, necesitaremos de otros métodos para seleccionar variables.

1.3. Selección de variables

Uno de las cuestiones más importantes a la hora de encontrar el modelo de ajuste más adecuado para explicar la variabilidad de una característica cuantitativa es la correcta especificación del llamado modelo teórico. En muchas situaciones se dispone de un conjunto grande de posibles variables explicativas, una posible pregunta sería saber si todas las variables deben entrar en el modelo de regresión y, en caso negativo, saber qué variables deben entrar y cuáles no.

En otras palabras, debemos seleccionar un subconjunto de variables entre todas las variables candidatas a ser explicativas de la variable dependiente, subconjunto que resulte suficientemente explicativo.

En general, si se incluyen cada vez más variables en un modelo de regresión, el ajuste a los datos mejora, aumenta la cantidad de parámetros a estimar, pero disminuye su precisión individual (mayor varianza) y, por tanto, la de la función de regresión estimada, produciéndose un sobreajuste. Por el contrario, si se incluyen menos variables de las necesarias en el modelo, las varianzas se reducen, pero los sesgos aumentarán obteniéndose una mala descripción de los datos.

Por otra parte, algunas variables predictoras pueden perjudicar la confiabilidad del modelo, especialmente si están correlacionadas con otras.
De esta manera, el objetivo de los métodos de selección de variables es buscar un modelo que se ajuste bien a los datos y que, a la vez, sea posible buscar un equilibrio entre bondad de ajuste y sencillez.

En la práctica, no obstante, la selección del subconjunto de variables explicativas de los modelos de regresión se deja en manos de procedimientos más o menos automáticos. Los procedimientos más usuales son los siguientes:

- **MÉTODOS DE MÍNIMOS CUADRADOS PENALIZADOS**

Se basan en los mínimos cuadrados ordinarios, pero añadiendo una penalización en la función objetivo, para forzar que alguna componente del vector de parámetros $\beta = (\beta_0, \beta_1, ..., \beta_p)$ sea cero y de esta manera conseguir estimación de los parámetros y selección de variables conjuntamente. (estos métodos son Ridge, Lasso y Elastic Net)

- **ALGORITMOS**

Procedimientos secuenciales de selección de variables basados en criterios que conjugan la optimalidad del ajuste y la reducción de la dimensión del espacio predictor. El procedimiento termina cuando se satisface una regla de parada establecida entre los algoritmos de selección de variables podemos señalar tres delos más usuales:

- **Selección hacia adelante:**

Procedimiento de selección de variables en el que las variables se introducen secuencialmente en el modelo. La primera variable que se considerará introducir en la ecuación será la que tenga mayor correlación, positiva o negativa, con la variable dependiente. Dicha variable se introducirá en la ecuación sólo si cumple el criterio de entrada. Si se introduce la primera variable, a continuación, se considerará la variable independiente cuya correlación parcial sea la mayor y que no esté en la ecuación. El procedimiento termina cuando ya no quedan variables que cumplan el criterio de entrada.

- ***Selección hacia atrás***

Procedimiento de selección de variables en el que se introducen todas las variables en la ecuación y después se van excluyendo una tras otra. Aquella variable que tenga la menor correlación parcial con la variable dependiente será la primera en ser considerada para su eliminación. Si satisface el criterio de eliminación, se eliminará. Tras haber excluido la primera variable, se pondrá a prueba aquella variable, de las que queden en la ecuación, que presente una correlación parcial más pequeña. El procedimiento termina cuando ya no quedan en la ecuación variables que satisfagan el criterio de eliminación.

- **Regresión paso a paso:**

Este procedimiento es una combinación de los dos anteriores. Comienza como el de introducción progresiva, pero en cada etapa se plantea si todas las variables introducidas deben de permanecer en el modelo.

Cuando se aplica este tipo de procedimientos tenemos que tener en cuenta cuál será la condición para suprimir o incluir un término. Para ello podemos considerar dos criterios:

- **Criterios de significación:**

En un método de eliminación hacia atrás se suprimirá el término que resulte menos significativo, y en un método de selección hacia adelante se añadirá el término que al añadirlo al modelo resulte más significativo. Un criterio de significación puede ser la significación de cada coeficiente.

- **Criterios globales:**

Una medida global de cada modelo, de modo que tenga en cuenta el ajuste y el exceso de parámetros. Como criterios destacamos el Criterio de Información de Akaike, AIC (Akaike, 1974) definido por

$$AIC = -2\ln(L) + k$$

Donde k es el número de parámetros en el modelo y L es el máximo valor de la función de verosimilitud en el modelo ajustado y el Criterio de Información de Bayes, BIC (Schwars, 1978) definido por

$$BIC = -2\ln L + k\ln(n)$$

Se trata de buscar un modelo cuyo AIC o BIC sea pequeño, ya que en ese caso habría una verosimilitud muy grande y pocos parámetros.

*OJO. **<u>Consecuencias de los métodos de selección de variables</u>**

Dos inconvenientes de estos métodos son que realizan un proceso discreto de exploración del espacio de modelos (cada variable es seleccionada o descartada) y trae consigo una fuerte inestabilidad, pequeños cambios en el conjunto de datos pueden producir grandes modificaciones en los resultados y que resultan inaplicables cuando el número de variables p es similar o incluso superior al número de observaciones n.

Para ello utilizamos los métodos de mínimos cuadrados penalizados citados anteriormente.

1.4. Regresión sesgada

Como ya se sabe, los estimadores MCO son los de mínima varianza en la clase de los estimadores lineales insesgados. Cualesquiera otros que consideremos, si son lineales y de varianza menor, habrán de ser sesgados.

Si consideramos adecuado como criterio en la elección de un estimador $\hat{c}$ su error cuadrático medio, $ECM \overset{def}{=} E\left[\hat{c}-c\right]^2$, y reparamos en que:

$$E\left[\hat{c}-c\right]^2 = E\left[\hat{c}-E\left[\hat{c}\right]+E\left[\hat{c}\right]-c\right]^2$$

$$= E\left[\hat{c}-E\left[\hat{c}\right]\right]^2 + E\left[E\left[\hat{c}\right]-c\right]^2 + \underbrace{2E\left[\hat{c}-E\left[\hat{c}\right]\right]\left[E\left[\hat{c}\right]-c\right]}_{=0}$$

$$= \mathrm{var}\left(\hat{c}\right) + \left(sesgo\left(\hat{c}\right)\right)^2$$

Podemos plantearnos la siguiente pregunta: ¿Es posible reducir el ECM en la estimación tolerando un sesgo? Si la respuesta fuera afirmativa, podríamos preferir el estimador resultante que, aunque sesgado, tendría un ECM menor, producido por una disminución en la varianza capaz de compensar el segundo sumando.

Los métodos de regresión sesgada se contemplan a veces como alternativas a los métodos de selección de variables en situaciones de acusada multicolinealidad.
Nos ocuparemos de procedimientos "Ad-hoc" para reducir la varianza de los estimadores.

No se profundiza más sobre la regresión sesgada en este trabajo, se puede obtener más información acerca de este tema en el capítulo 10 del trabajo de (Tusell, 2011)

1.5. Problema de colinealidad

La colinealidad es uno de los mayores inconvenientes que nos podemos encontrar en un análisis de regresión. Si en un modelo de regresión lineal múltiple alguna variable independiente es combinación lineal de otras, el método MCO es irresoluble, debido a que, en ese caso, la matriz $(X'X)$ es singular, es decir, su

determinante es cero y no se puede invertir. A este fenómeno se le denomina colinealidad. (Castro, 2013)

Otro modo, por tanto, de definir la colinealidad es cuando alguno de los coeficientes de correlación simple o múltiple entre algunas de las variables independientes es 1, es decir, cuando algunas variables independientes están correlacionadas entre sí.

En la práctica, esta colinealidad exacta raras veces ocurre, pero sí surge con cierta frecuencia la llamada "casi" colinealidad, o por extensión, simplemente colinealidad en que alguna variable es "casi" combinación lineal de otra u otras, o dicho de otro modo, algunos coeficientes de correlación simple o múltiple entre las variables independientes están cercanos a 1, aunque no llegan a dicho valor.

En este caso la matriz $(X'X)$ es "casi" singular, es decir su determinante no es cero, pero es muy pequeño. Como para invertir una matriz hay que dividir por su determinante, en esta situación surgen problemas de precisión en la estimación de los coeficientes, ya que los algoritmos de inversión de matrices pierden precisión al tener que dividir por un número muy pequeño, siendo además inestables.

Además, como la matriz de varianzas de los estimadores es proporcional a $(X'X)$, resulta que en presencia de colinealidad los errores estándar de los coeficientes son grandes (no hay precisión en sentido estadístico). Es importante señalar que el problema de multicolinealidad, en mayor o menor grado, se plantea porque no existe información suficiente para conseguir una estimación precisa de los parámetros del modelo. (Anonimo)

1.6. Detección de la colinealidad

A la hora de plantear modelos de regresión lineal múltiple conviene estudiar previamente la existencia de colinealidad.

- **Factor de inflación de la varianza**

Como medida de la misma hay varios estadísticos propuestos, los más sencillos son los coeficientes de determinación o cuadrados de los coeficientes de correlación múltiple de cada variable independiente con todas las demás, es decir:

$$R_i{}^2 = R^2{}_{xi|x_1\ldots,x_{i=1},x_{i+1,\ldots,}x_n}, \quad i = 1, \ldots, n$$

y, relacionados con ellos, el factor de inflación de la varianza (FIV) y la tolerancia (T), definidos como

$$FIV_i = \frac{1}{1 - R_i^2}, \qquad T_i = \frac{1}{FIV_i} = 1 - R_i^2$$

Una regla empírica, citada por (Kleinbaum, 1988), consiste en considerar que existen problemas de colinealidad si algún FIV es superior a 10, que corresponde a algún $R_i{}^2 > 0{,}9 \text{y} T_i < 0{,}1$.

Aunque puede existir colinealidad con FIV bajos, además puede haber colinealidades que no impliquen a todas las variables independientes y que, por tanto, no son bien detectadas por el FIV.

- **Número de condición**

Este procedimiento de detección de la multicolinealidad es el más adecuado entre los actualmente disponibles, según afirman Judge et al. [5]. El número de condición, $k(X)$, es igual a la raíz cuadrada de la razón entre la raíz característica más grande (λ_{max}) y la raíz característica más pequeña (λ_{min}) de la matriz ($X'X$), es decir,

$$k(X) = \sqrt{\frac{\lambda_{max}}{\lambda_{min}}}$$

Si la matriz ($X'X$) es de dimensión $n \times n$ se obtienen n raíces características, pudiéndose calcular para cada una de ellas un índice de condición denido de la siguiente forma:

$$ic(\lambda_i) = \sqrt{\frac{\lambda_{max}}{\lambda_i}}$$

El número de condición mide la sensibilidad de las estimaciones mínimo- cuadráticas ante pequeños cambios en los datos. De acuerdo con los estudios realizados, tanto

con datos observados como con datos simulados, el problema de la multicolinealidad es grave cuando el número de condición toma un valor entre 20 y 30. Naturalmente, si este indicador superase el valor de 30, el problema sería ya manifiestamente grave. Estos valores vienen generalmente referidos a regresores medidos con escala de longitud unidad (es decir, con los regresores divididos por la raíz cuadrada de la suma de los valores de las observaciones), pero no centrados. Parece que no es conveniente centrar los datos (es decir, restarles sus correspondientes medias), ya que esta operación oscurece cualquier dependencia lineal que implique al término independiente. (Carrasco C., 2013)

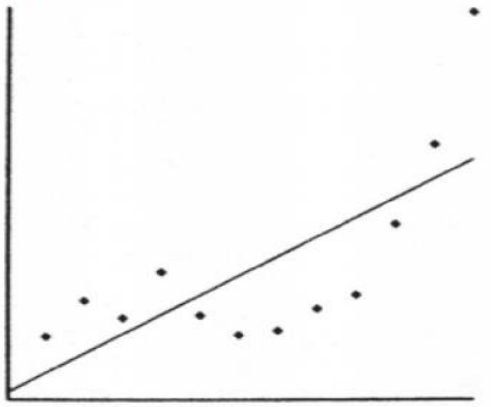
1.7. Overfitting

Overfitting o Sobreajuste es un problema clásico al desarrollar un modelo estadístico,

Consideremos por ejemplo una serie dispersa de puntos (y1, x1),…, (yn, xn) en un plano, a los cuales queremos ajustar un polinomio. Buscamos ni más ni menos f(x). Tenemos varias opciones:

Buscar un polinomio muy simple, como por ejemplo un modelo de regresión lineal (Fig.1).

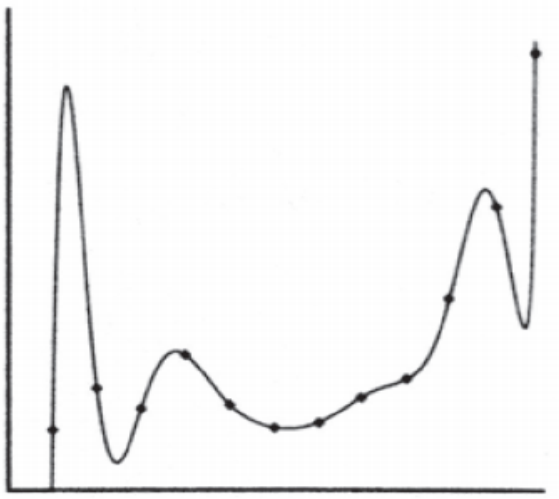
Figura 1. Ajuste a los puntos de un modelo de regresión lineal.



Es el más sencillo de los modelos y, sin embargo, no se ajusta bien, ni es factible que pueda predecir algo en el futuro.

Buscar un polinomio complejo de grado cercano a (n-1) (Fig. 2).

Figura 2. Ajuste a los puntos de un polinomio complejo de grado n-1.



Este modelo es más complejo y se ajusta a los datos; sin embargo, en algunos lugares se "escapa" y es poco factible que prediga algo en el futuro: está sobre ajustado.

Se define la paradoja del sobreajuste como: los modelos complejos contienen más información sobre los datos de la muestra, pero menos información sobre los datos futuros (predicción). El problema del sobreajuste en estadística no es menor y lleva a algunos problemas graves en investigación:

- Algunas relaciones que parecen estadísticamente significativas son solamente ruido.
- La complejidad del modelo estadístico es muy grande para la cantidad de datos que tenemos.
- El modelo en general no es replicable y predice mal. (Nannen , 2003)

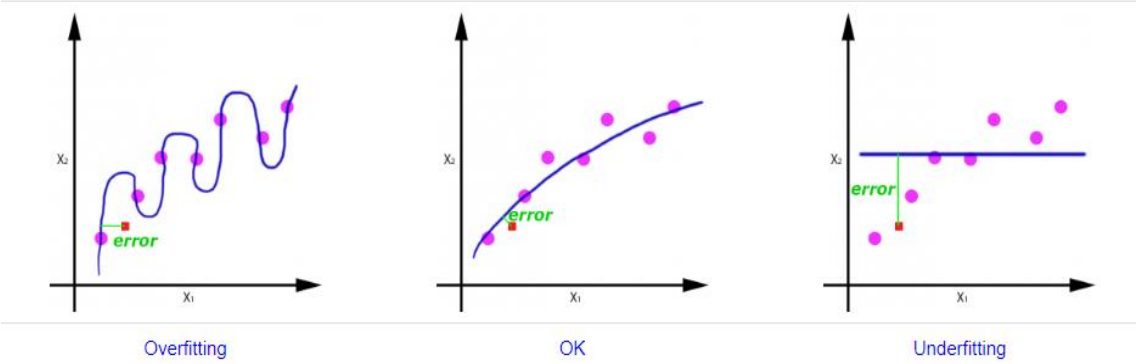
1.8. Underfitting

El underfitting o subajuste se da cuando existe un exceso de generalización del modelo, el cual, prácticamente ignora todas o la mayoría de las muestras de entrenamiento.

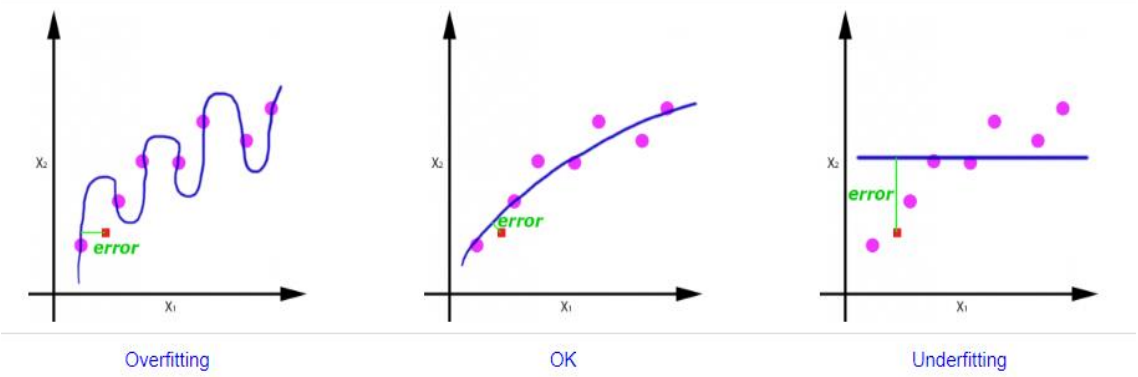
Burnham y Anderson dicen que un modelo no ajustado ignoraría alguna estructura importante replicable (es decir, replicable conceptualmente en la mayoría de las otras muestras) en los datos y, por lo tanto, no identificaría los efectos que realmente fueron respaldados por los datos. En este caso, el sesgo en los estimadores de parámetros es a menudo sustancial, y la varianza muestral se

subestima, ambos factores dan como resultado una cobertura pobre del intervalo de confianza. Los modelos no ajustados tienden a perder importantes efectos de tratamiento en entornos experimentales.

A continuación un ejemplo visual para diferenciar Overfitting y Underfitting



Veamos por ejemplo para el caso de la regresión, el error que se cometería si nos llegase un nuevo dato de entrada (punto rojo), para cada uno de las funciones obtenidas:



Se puede apreciar como el error cometido con un dato de test, es mayor en el caso en el que se produce overfitting y underfitting, y es menor en el caso en el que se ha generalizado de forma correcta, aunque evidentemente tiene un pequeño error. (Nannen , 2003)

1.9. Regresión Ridge

El método de Regresión Ridge, al cual denominaremos a partir de ahora RR, permite detectar la multicolinealidad dentro de un modelo de regresión del tipo

$$Y = X^t\hat{\beta} + \varepsilon$$

Fue propuesto por Arthur E. Hoerl y Robert W. Kennard en 1970 y es usado para trabajar con modelos que presentan sesgo.

La idea del método es simple y consiste en que dado que la matriz (X^tX) es altamente condicionada o cercana a singular es posible agregar constantes positivas a los elementos de la diagonal para asegurar que la matriz resultante no sea altamente condicionada. El vector de coeficientes de RR está dado por:

$$\beta_R = \left(X^tX + K^\delta\right)^{-1}X^tY$$

El cual se puede reescribir como $\beta_R = z\hat{\beta}$, donde $\hat{\beta}$ es el estimador ordinario de MC dado por $\hat{\beta} = (X^tX)^{-1}X^tY$ y $Z = [I + K^\delta(X^tX)^{-1}]^{-1}$es la matriz que transforma a $\hat{\beta}$ en $\hat{\beta}_R$, es decir, β_R es la solución al problema de optimización:
Min:

$$(\beta_R - \hat{\beta})X^tX(\beta_R - \hat{\beta})$$

Sujeto a:

$$\beta_R\beta_R \leq r^2$$

Y la matriz de covarianza es dada por:

$$V\left(\tilde{B}\right) = (X'X + kI)^{-1}X'X\ (X'X + kI)^{-1}$$

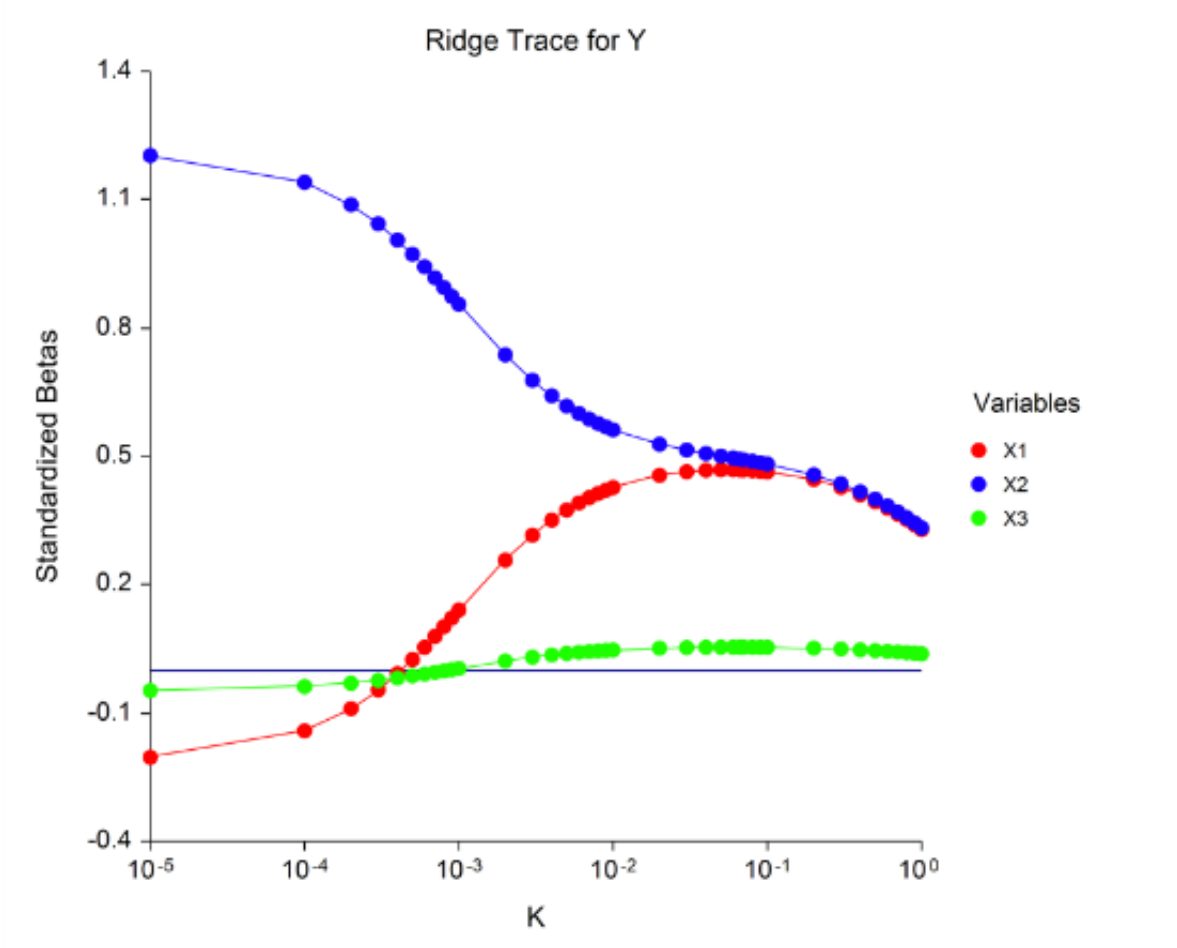
En base a esto se puede apreciar que existe un valor de k para el cual el valor de cuadrado medio del error (la varianza más el sesgo cuadrado) del estimador Ridge es menor que el estimador de mínimos cuadrados. Desafortunadamente, el valor apropiado de k depende de los verdaderos coeficientes (los cuales son estimados) y una solución analítica que garantice el valor óptimo para la regresión Ridge. (James, Witten, Hastie, & Tibshirani, 2013)

Seleccionando un valor K

Ridge Trace

Uno de los principales obstáculos en usar la regresión Ridge es el escoger un valor k apropiado. Hoerl and Kennard, los inventores de la regresión Ridge, sugirieron usar un gráfico el cual ellos llamaron la *traza Ridge.*

Este gráfico muestra los coeficientes de la regresión Ridge en función de k. Cuando se visualiza esta gráfica, el analista tiene la posibilidad de escoger el valor de k en base a los coeficientes de regresión estabilizados. A menudo, los coeficientes de regresión variarán ampliamente para pequeños valores de k y luego se estabilizarán. Escoger el valor más pequeño posible para k (lo cual introduce un pequeño sesgo) para el cual los coeficientes de regresión parezcan que permanecen constantes. Es necesario tomar en cuenta que incrementar el valor de k eventualmente conllevará a los coeficientes de regresión a cero. A continuación, se muestra una gráfica de la traza Ridge.



En este ejemplo, los valores de k son mostrados en una escala logarítmica, Se ha dibujado una línea vertical en los valores de k seleccionados, los cuales son 0.006. En base a esto se puede comentar lo siguiente:

En primer lugar, los ejes verticales contienen los puntos para la solución de mínimos cuadrados. Estos son indicados como 0.000001. Esto se realiza con la finalidad de que estos coeficientes puedan visualizarse. En realidad, el logaritmo de zero es menos infinito, por este motivo, los valores de los mínimos cuadrados no pueden ser mostrados en el eje horizontal cuando la escala es logarítmica. Se está mostrando un amplio rango de valores. Se puede apreciar que agregar k tiene poco impacto hasta que k toma un valor cercano a 0.0001. Esta acción parece tener impacto cuando el valor es cercano a 0.006.

1.10. Regresión Lasso

Es un tipo de regresión linear que utiliza un reductor. El reductor es donde el valor de los datos se encoge hacia un punto central como la media.

Este método realiza selección de variables y regularización para mejorar la exactitud e interpretabilidad del modelo estadístico producido por este.

El procedimiento que ofrece Lasso estimula modelos simples, dispersos (por ejemplo, modelos con menores parámetros. Este tipo particular de regresión es recomendado para modelos que muestran altos grados de multicolinealidad o cuando uno desea automatizar ciertas partes del modelo de selección, como la eliminación de selección de variables o parámetros.

Lasso es el acrónimo de Least Absolute Shrinkage and Selecction Operator.

1.10.1. Regularización

Regresión Lasso realiza una regularización L1, la cual agrega una penalidad igual al valor absoluto de la magnitud de los coeficientes. Este tipo de regularización puede resultar en modelos dispersos con pocos coeficientes. Algunos coeficientes pueden volverse cero y ser eliminados del modelo. A mayor penalidad el valor del coeficiente tiende a cero lo cual es ideal para producir modelos más simples. En otro sentido, regularizaciones L2 (por ejemplo, regresión Ridge) no realiza una

eliminación de coeficientes o modelos dispersos. Esto hace a la regresión Lasso de lejos más sencilla de interpretar en comparación con la de Ridge.

1.10.2. Desempeño de la regresión Lasso

Las soluciones que plantea la regresión Lasso son problemas de programación cuadrática para las cuales es mejor usar alguna herramienta. En si el objetivo del algoritmo es poder minimizar la siguiente expresión:

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

Esto es lo mismo que minimizar la suma de cuadrados con la restricción de $\Sigma\ |B_j \leq$ s. Algunas de las βs son reducidas exactamente a cero, resultando en un modelo de regresión que es más simple de interpretar. (Harrell, 2015)

Un parámetro de afinación, λ controla la fortaleza de la penalidad de L. El parámetro λ es básicamente la cantidad a ser reducida:

- Cuando λ = 0, no se elimina ningún parámetro. El estimado es el mismo hallado en la regresión lineal.
- Cuando λ aumenta, muchos de los coeficientes son inicializados en cero y eliminados (teóricamente, cuando λ = ∞, todos los coeficientes son eliminados)
- Cuando λ aumenta, el sesgo aumenta
- Cuando λ decrece, la varianza aumenta

Si un intercepto es incluido en el modelo, este se mantiene sin modificaciones.

1.11Elección del parámetro λ

Como puede observarse todas estas técnicas de mínimos cuadrados penalizados dependen de un parámetro de penalización λ, que controla la importancia dada a la penalización en el proceso de optimización. Cuanto mayor es λ mayor es la penalización en los coeficientes de regresión y más son contraídos éstos hacia cero.

La elección de este parámetro involucra un balance entre los componentes de sesgo y varianza del ECM al estimar β .

Una propuesta inicial y que continúa siendo sugerida por algunos autores es la utilización de una traza Ridge para determinar λ. Consiste en graficar simultáneamente los coeficientes de regresión estimados en función de λ, y elegir el valor más pequeño del parámetro para el cuál se estabilizan dichos coeficientes. Un método más automático, pero intensivo computacionalmente, consiste en estimar λ mediante validación cruzada (en general se recomiendan utilizar ambos métodos y comparar resultados).

El método de validación cruzada consiste en dividir el modelo en un set de entrenamiento ***(training set)*** para ajustar un modelo y un set de prueba ***(test set)*** para evaluar su capacidad predictiva, mediante el error de predicción u otra medida.

La forma en que se aplica la validación cruzada es mediante la división del conjunto de datos disponibles de manera aleatoria en k subconjuntos o pliegues de igual tamaño y mutuamente excluyentes.

Uno de los subconjuntos se utiliza como datos de prueba y el resto $(K-1)$ como datos de entrenamiento. El proceso de validación cruzada es repetido durante k iteraciones, con cada uno de los posibles subconjuntos de datos de prueba. Finalmente se realiza la media aritmética de los resultados de cada iteración para obtener un único resultado. Este método es muy preciso puesto que evaluamos a partir de K combinaciones de datos de entrenamiento y de prueba, pero aun así tiene una desventaja, y es que es lento desde el punto de vista computacional.

La validación cruzada con $k = 10$ es una de las más utilizadas, pero hay que tener en cuenta el número de observaciones del que disponemos. (Wright, 2009)

Nuestro valor del parámetro será el que nos dé el mínimo error.

2. APLICACIÓN

Lectura de datos Cáncer

Este dataset tiene 10 variables y 97 observaciones y se empleó en un estudio para determinar que variables influyen en la presencia de un antígeno prostático específico para detectar el cáncer de próstata. (Vaquerizo, 2014) y (Guerra de la Corte, 2016)

3. RESULTADOS

3.1. Usando R

```
cancer<-read.table(file
="https://web.stanford.edu/~hastie/ElemStatLearn//datasets/prostate.data", header=TRUE)
str(cancer)
```

```
## 'data.frame':    97 obs. of  10 variables:
##  $ lcavol : num  -0.58 -0.994 -0.511 -1.204 0.751 ...
##  $ lweight: num  2.77 3.32 2.69 3.28 3.43 ...
##  $ age    : int  50 58 74 58 62 50 64 58 47 63 ...
##  $ lbph   : num  -1.39 -1.39 -1.39 -1.39 -1.39 ...
##  $ svi    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ lcp    : num  -1.39 -1.39 -1.39 -1.39 -1.39 ...
##  $ gleason: int  6 6 7 6 6 6 6 6 6 6 ...
##  $ pgg45  : int  0 0 20 0 0 0 0 0 0 0 ...
##  $ lpsa   : num  -0.431 -0.163 -0.163 -0.163 0.372 ...
##  $ train  : logi  TRUE TRUETRUETRUETRUETRUE ...
```

Cargar estos paquetes para el ejercicio

```
library(car)
library(MASS)
```

Donde:
* lcavol : log-cancer volume.
* lweight : log-prostate weight.
* age : age of patient.
* lbhp : log-amount of benign hyperplasia.
* svi : seminal vesicle invasion.
* lcp : log-capsular penetration.
* gleason : Gleason Score, check.
* pgg45 : percent of Gleason scores 4 or 5.
*lpsa is the response variable, log-psa.

Después de haber cargado la data e instalado los paquetes, vamos a seccionar la data de los pacientes.

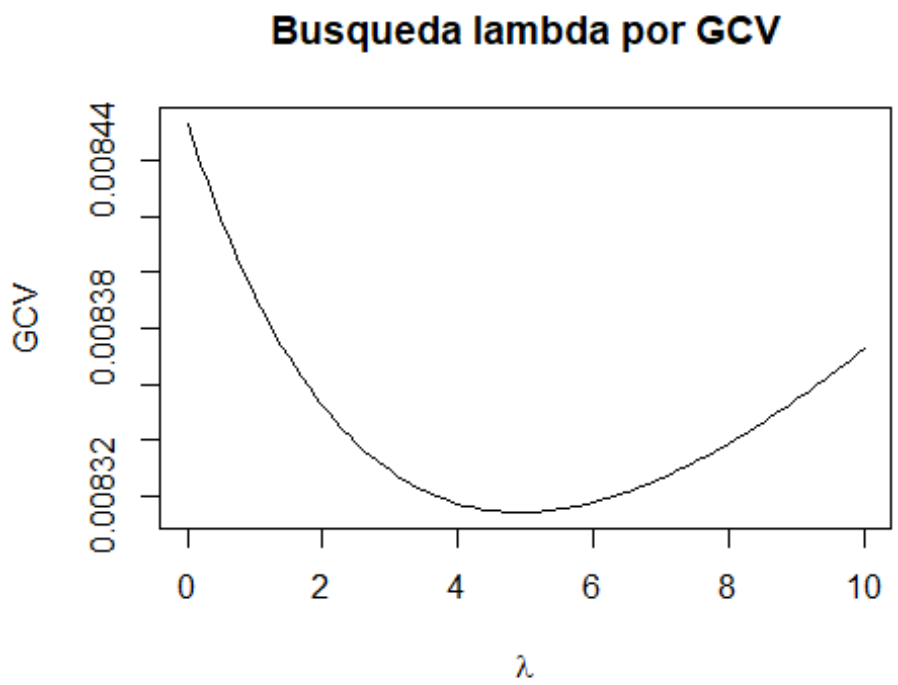
```
train =subset(cancer,train=="TRUE")
test =subset(cancer,train=="FALSE")
modelo_mco<-lm(lpsa~. , data=train[,-10])
summary(modelo_mco)

##
## Call:
## lm(formula = lpsa ~ ., data = train[, -10])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.64870 -0.34147 -0.05424  0.44941  1.48675
##
## Coefficients:
##             EstimateStd. Error t value Pr(>|t|)
## (Intercept)  0.429170   1.553588   0.276  0.78334
## lcavol       0.576543   0.107438   5.366 1.47e-06 ***
## lweight      0.614020   0.223216   2.751  0.00792 **
## age         -0.019001   0.013612  -1.396  0.16806
## lbph         0.144848   0.070457   2.056  0.04431 *
## svi          0.737209   0.298555   2.469  0.01651 *
## lcp         -0.206324   0.110516  -1.867  0.06697 .
## gleason     -0.029503   0.201136  -0.147  0.88389
## pgg45        0.009465   0.005447   1.738  0.08755 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7123 on 58 degreesoffreedom
## Multiple R-squared:  0.6944, Adjusted R-squared:  0.6522
## F-statistic: 16.47 on 8 and 58 DF,  p-value: 2.042e-12
```

La sospecha multicolinealidad se puede analizar con el VIF que obtenemos con la funciónVIF de la librería car:

```
library(car)

## Loadingrequiredpackage: carData

vif<-vif(modelo_mco)
vif

##   lcavol lweight age lbph svi lcp gleason    pgg45
## 2.318496 1.472295 1.356604 1.383429 2.045313 3.117451 2.644480 3.313288
```

Tenemos 2 valores por encima de 3.
Vamos a realizar el modelo de regresión ridge para encontrar ese sesgo que me reduzca la varianza:

```
library(MASS)
modelo_ridge<-lm.ridge(lpsa~., data=train[,-10], lambda =seq(0,10,0.1))
plot(seq(0,10,0.1), modelo_ridge$GCV, main="Busqueda lambda por GCV",
type="l",xlab=expression(lambda), ylab="GCV")
```

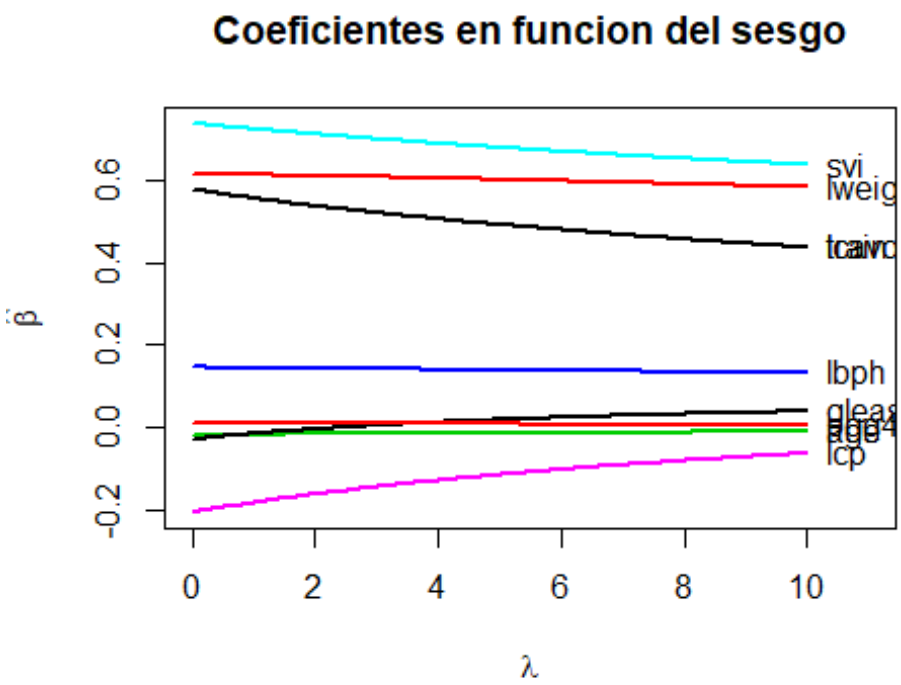


Vemos que el lambda esta muy próximo a 5 para saber el valor óptimo empleamos la función select:

```
select(lm.ridge(lpsa~., data=train[,-10], lambda =seq(0,10,0.1)))

## modified HKB estimatoris 3.355691
## modified L-W estimatoris 3.050708
## small est value of GCV  at 4.9
```

El valor que minimiza el GCV es 4.9 ya estamos en disposición de crear el modelo. Si deseamos ver gráficamente como se modifican los parámetros de nuestro modelo en función de lambda podemos realizar lo siguiente:

```
matplot(seq(0,10,0.1), coef(modelo_ridge)[,-1], xlim=c(0,11),
type="l",xlab=expression(lambda), ylab=expression(hat(beta)),  lty=1, lwd=2,
main="Coeficientes en funcion del sesgo")
text(rep(10, 9), coef(modelo_ridge)[length(seq(0,10,0.1)),-1], colnames(train)[-9],pos=4)
```



Para verificar se hallara **CME** de ambos modelos para compararlo

```
ajuste_mco<-predict(modelo_mco,test)
sum((test$lpsa-ajuste_mco)^2)

## [1] 15.63822
```

Para el modelo ridge

```
#Modelo con lambda optimo
modelo_ridge<-lm.ridge(lpsa~., data=train[,-10], lambda =4.9)
coefficients(modelo_ridge)

##                    lcavol                  lweight        agel                  bph
##  0.096814771  0.492787412  0.601103227 -0.014821787  0.138019854
##        svi                lcp                  gleason       pgg45
##  0.679632580 -0.116790333  0.017113954  0.007081258

#estos objetos no admiten predict
coeficientes <-as.vector(coef(modelo_ridge))
matriz <-as.matrix(test[,-9:-10])
matriz <-cbind(rep(1,length=nrow(test)),matriz)
ajuste_contraida<-matriz%*%coeficientes
sum((test$lpsa-ajuste_contraida)^2)

## [1] 14.8323
```

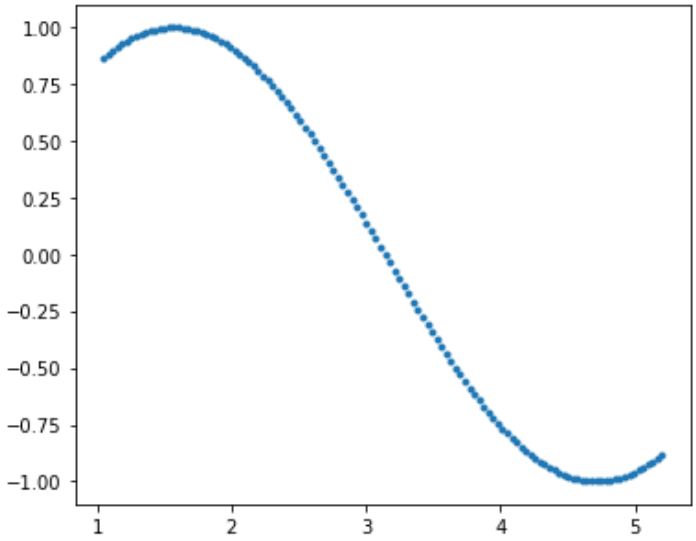
3.2. Usando Python

A continuación se presenta el uso de la regresión Ridge usando el lenguaje de programación Python,

Para tal fin se presenta el análisis de una curva sinusoidal

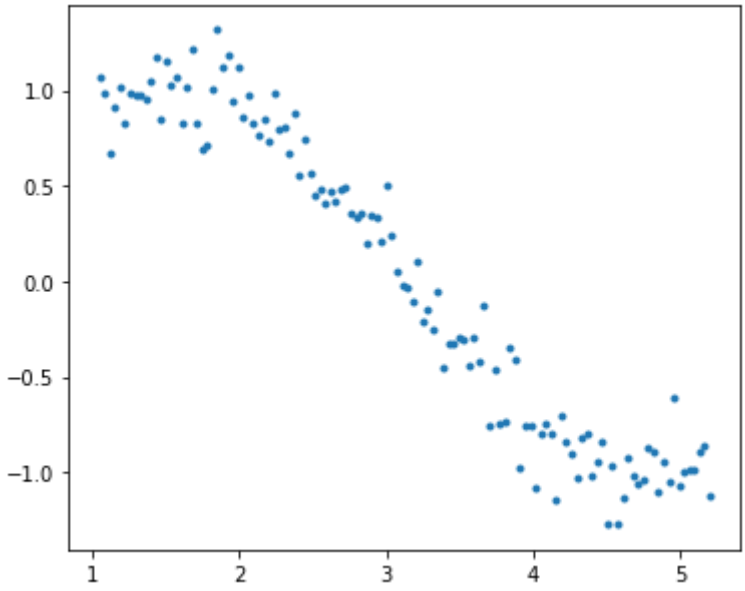
```
#Importando librerias
import numpy as np
import pandas as pd
import random
import matplotlib.pyplot as plt
%matplotlib inline
from matplotlib.pylab import rcParams
rcParams['figure.figsize'] = 6, 5
```

```
#Define un rango de 60 a 300 grados en radianes
x = np.array([i*np.pi/180 for i in range(60,300,2)])
#Define una semilla
np.random.seed(10)
y = np.sin(x)
data = pd.DataFrame(np.column_stack([x,y]),columns=['x','y'])
plt.plot(data['x'],data['y'],'.')
```



Para tal fin se agrega ruido a los datos a analizar

```
#Agregamos ruido a la gráfica sinusoidal
y = np.sin(x) + np.random.normal(0,0.15,len(x))
data = pd.DataFrame(np.column_stack([x,y]),columns=['x','y'])
plt.plot(data['x'],data['y'],'.')
```



Creamos variables predictoras que no tengan multicolinealidad

```
for i in range(2,16):          # Potencia de 2 a 16
   colname = 'x_%d'%i          # Nueva variable para las potencias
   data[colname] = data['x']**i  # Crea una nueva columna para las potencias
print(data.head())
```

```
          x         y       x_2       x_3       x_4       x_5       x_6  \
0  1.047198  1.065763  1.096623  1.148381  1.202581  1.259340  1.318778
1  1.082104  0.990239  1.170949  1.267089  1.371122  1.483697  1.605515
2  1.117011  0.666984  1.247713  1.393709  1.556788  1.738948  1.942424
3  1.151917  0.912288  1.326913  1.528495  1.760699  2.028180  2.336296
4  1.186824  1.020384  1.408551  1.671702  1.984016  2.354677  2.794587

        x_7       x_8       x_9      x_10      x_11      x_12      x_13  \
0  1.381021  1.446202  1.514459  1.585938  1.660790  1.739176  1.821260
1  1.737334  1.879977  2.034331  2.201357  2.382098  2.577678  2.789316
2  2.169709  2.423588  2.707173  3.023942  3.377775  3.773011  4.214494
3  2.691220  3.100062  3.571015  4.113514  4.738429  5.458278  6.287485
4  3.316683  3.936319  4.671717  5.544505  6.580351  7.809718  9.268760

        x_14       x_15
0   1.907219   1.997235
1   3.018331   3.266148
2   4.707635   5.258479
3   7.242662   8.342948
4  11.000386  13.055521
```

Creamos una función para el análisis de la regresión lineal

```python
#Importa el modelo de la regresion lineal desde scikit-learn.
from sklearn.linear_model import LinearRegression

def linear_regression(data, power, models_to_plot):
    #Inicializa predictores:
    predictors=['x']
    if power>=2:
        predictors.extend(['x_%d'%i for i in range(2,power+1)])

    #Define el modelo
    linreg = LinearRegression(normalize=True)
    linreg.fit(data[predictors],data['y'])
    y_pred = linreg.predict(data[predictors])

    #Verifica que potencia sera visualizado
    if power in models_to_plot:
        plt.subplot(models_to_plot[power])
        plt.tight_layout()
        plt.plot(data['x'],y_pred)
        #plt.plot(data['x'],data['y'],'.')
        plt.title('Plot for power: %d'%power)

    #Returna los resultados
    rss = sum((y_pred-data['y'])**2)
    ret = [rss]
    ret.extend([linreg.intercept_])
    ret.extend(linreg.coef_)
    return ret
```

Se define la matriz donde se almacenara los coeficientes de la regresión lineal

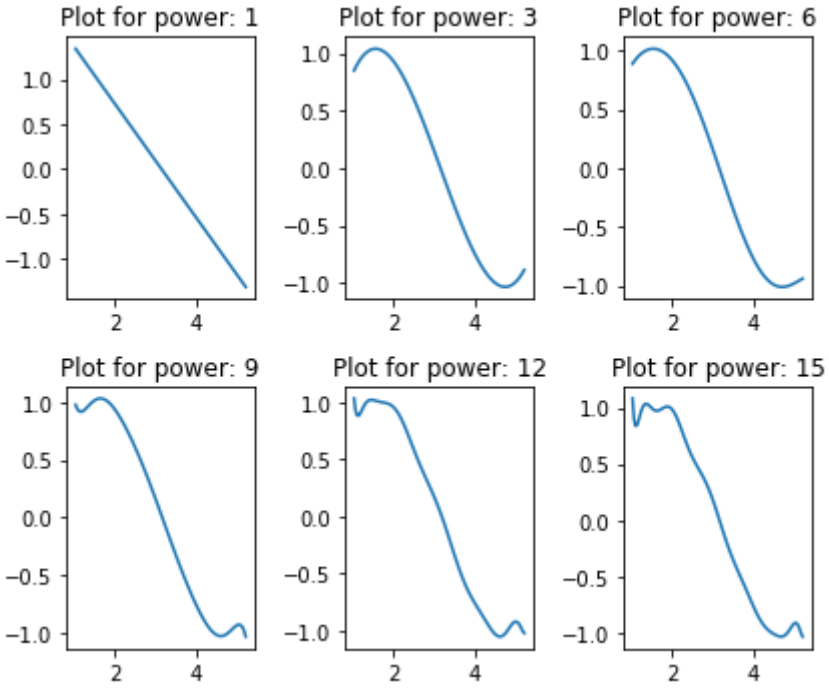
```
#Inicializa un dataframe para almacenar los valores de los coeficientes
col = ['rss','intercept'] + ['coef_x_%d'%i for i in range(1,16)]
ind = ['model_pow_%d'%i for i in range(1,16)]
coef_matrix_simple = pd.DataFrame(index=ind, columns=col)
#print(coef_matrix_simple)
```

```
#Define los modelos que seran visualizadas, solo pueden ser 6
models_to_plot = {1:231,3:232,6:233,9:234,12:235,15:236}
print(models_to_plot)
```

```
{1: 231, 3: 232, 6: 233, 9: 234, 12: 235, 15: 236}
```

Se ejecuta la regresión lineal y se grafica 6 casos

```
#Iteraccion a traves de todas las potencias posibles
for i in range(1,16):
   coef_matrix_simple.iloc[i-1,0:i+2] = linear_regression(data, power=i,
models_to_plot=models_to_plot)
```



Se muestran los coeficientes

```
#Muestra el resultados de los coeficientes
pd.options.display.float_format = '{:,.2g}'.format
coef_matrix_simple
```

	rss	intercept	coef_x_1	coef_x_2	coef_x_3	coef_x_4	coef_x_5	coef_x_6	coef_x_7	coef_x_8	coef_x_9	coef_x_10	coef_x_11
model_pow_1	6.5	2	-0.64	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
model_pow_2	6.4	1.9	-0.57	-0.012	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
model_pow_3	2.5	-1	2.9	-1.2	0.13	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
model_pow_4	2.5	-0.86	2.7	-1.1	0.1	0.0024	NaN	NaN	NaN	NaN	NaN	NaN	NaN
model_pow_5	2.4	0.85	-0.86	1.6	-0.87	0.17	-0.01	NaN	NaN	NaN	NaN	NaN	NaN
model_pow_6	2.4	-0.54	2.6	-1.8	0.8	-0.27	0.048	-0.0031	NaN	NaN	NaN	NaN	NaN
model_pow_7	2.4	15	-42	51	-32	12	-2.4	0.27	-0.012	NaN	NaN	NaN	NaN
model_pow_8	2.4	13	-35	42	-26	8.5	-1.5	0.12	0.0017	-0.00057	NaN	NaN	NaN
model_pow_9	2.4	54	-1.9e+02	3e+02	-2.6e+02	1.4e+02	-48	11	-1.6	0.13	-0.0046	NaN	NaN
model_pow_10	2.4	-0.95	40	-1.3e+02	1.9e+02	-1.6e+02	83	-28	6.1	-0.84	0.066	-0.0023	NaN
model_pow_11	2.4	2.3e+02	-1e+03	2.1e+03	-2.4e+03	1.9e+03	-9.7e+02	3.5e+02	-90	16	-1.8	0.12	-0.0035
model_pow_12	2.3	1.8e+03	-8.9e+03	2e+04	-2.6e+04	2.3e+04	-1.3e+04	5.7e+03	-1.7e+03	3.7e+02	-56	5.5	-0.33
model_pow_13	2.3	6e+03	-3.2e+04	7.8e+04	-1.1e+05	1.1e+05	-7.1e+04	3.4e+04	-1.2e+04	3.1e+03	-5.8e+02	77	-6.8
model_pow_14	2.3	1.1e+04	-6.2e+04	1.6e+05	-2.4e+05	2.5e+05	-1.8e+05	9.4e+04	-3.7e+04	1.1e+04	-2.4e+03	4e+02	-46
model_pow_15	2.3	3.9e+02	5.8e+03	-3.9e+04	1.1e+05	-1.7e+05	1.7e+05	-1.3e+05	6.8e+04	-2.7e+04	8.1e+03	-1.8e+03	3e+02

Se observa que los coeficientes se van incrementando si aumenta la cantidad de variables predictoras

Se crea una función para el análisis de la regresión ridge

```
#Importa el modelo de la regresion Ridge desde scikit-learn.
from sklearn.linear_model import Ridge
def ridge_regression(data, predictors, alpha, models_to_plot={}):
    #Define el modelo
    ridgereg = Ridge(alpha=alpha,normalize=True)
    ridgereg.fit(data[predictors],data['y'])
    y_pred = ridgereg.predict(data[predictors])

    #Verifica que potencia sera visualizado
    if alpha in models_to_plot:
        plt.subplot(models_to_plot[alpha])
        plt.tight_layout()
        plt.plot(data['x'],y_pred)
        #plt.plot(data['x'],data['y'],'.')
        plt.title('Plot for alpha: %.3g'%alpha)

    #Returna los valores
    rss = sum((y_pred-data['y'])**2)
    ret = [rss]
    ret.extend([ridgereg.intercept_])
    ret.extend(ridgereg.coef_)
    return ret
```

Se crea un conjunto de valores alpha para el análisis de ridge

```
#Inicializa predictores:
predictors=['x']
predictors.extend(['x_%d'%i for i in range(2,16)])

#Define los diferentes valores de alpha a ser probados
#alpha_ridge = [1e-15, 1e-10, 1e-8, 1e-4, 1e-3,1e-2, 1, 5, 10, 20]
alpha_ridge = [0.1, 0.2, 0.3, 0.4, 0.5, 1, 3, 5, 10, 20]


#Inicializa un dataframe para almacenar los valores de los coeficientes
col = ['rss','intercept'] + ['coef_x_%d'%i for i in range(1,16)]
ind = ['alpha_%.2g'%alpha_ridge[i] for i in range(0,10)]
coef_matrix_ridge = pd.DataFrame(index=ind, columns=col)

#print(coef_matrix_ridge)
```

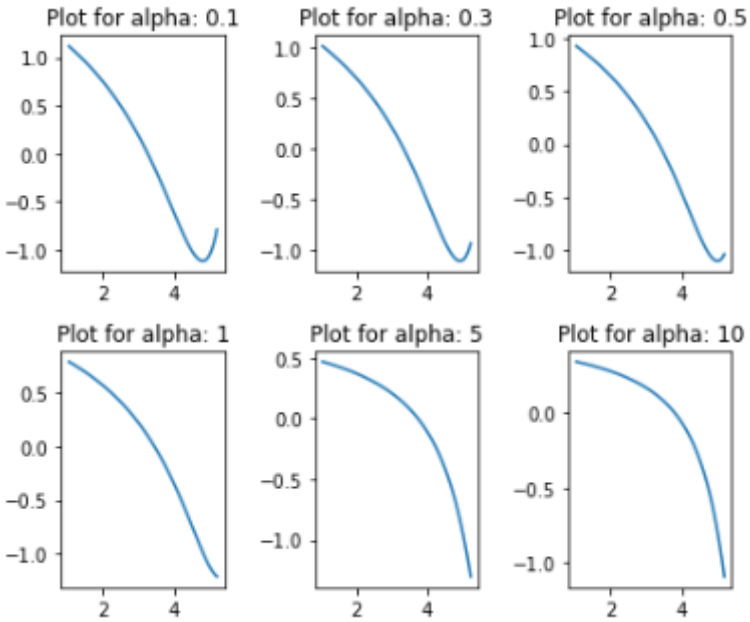
```
#Define los modelos que seran visualizadas, solo pueden ser 6
```

```
#models_to_plot = {1e-15:231, 1e-10:232, 1e-4:233, 1e-3:234, 1e-2:235, 5:236}
models_to_plot = {0.1:231, 0.3:232, 0.5:233, 1:234, 5:235, 10:236}
print(models_to_plot)
```

```
{0.1: 231, 0.3: 232, 0.5: 233, 1: 234, 5: 235, 10: 236}
```

Se ejecuta el análisis ridge

```
for i in range(10):
   coef_matrix_ridge.iloc[i,] = ridge_regression(data, predictors, alpha_ridge[i],
models_to_plot)
```



Se muestra los coeficientes de los análisis

```
#Muestra el resultados de los coeficientes
pd.options.display.float_format = '{:,.2g}'.format
coef_matrix_ridge
```

	rss	intercept	coef_x_1	coef_x_2	coef_x_3	coef_x_4	coef_x_5	coef_x_6	coef_x_7	coef_x_8	coef_x_9	coef_x_10	coef_x_11	coef_x_
alpha_0.1	4.2	1.4	-0.23	-0.034	-0.0051	-0.00071	-8.8e-05	-8.6e-06	-3.3e-07	1.4e-07	5.5e-08	1.4e-08	3.2e-09	6.7e-10
alpha_0.2	4.8	1.3	-0.22	-0.03	-0.0045	-0.00065	-8.7e-05	-1e-05	-8.4e-07	1.5e-08	3e-08	1e-08	2.6e-09	5.9e-10
alpha_0.3	5.6	1.3	-0.2	-0.028	-0.0042	-0.00062	-8.5e-05	-1e-05	-1e-06	-3.9e-08	1.9e-08	7.8e-09	2.2e-09	5.2e-10
alpha_0.4	6.5	1.2	-0.19	-0.026	-0.004	-0.00059	-8.2e-05	-1.1e-05	-1.1e-06	-7.1e-08	1.1e-08	6.1e-09	1.8e-09	4.6e-10
alpha_0.5	7.4	1.1	-0.18	-0.025	-0.0037	-0.00056	-8e-05	-1.1e-05	-1.2e-06	-9.4e-08	5e-09	4.8e-09	1.6e-09	4e-10
alpha_1	12	0.97	-0.14	-0.019	-0.003	-0.00047	-7e-05	-1e-05	-1.3e-06	-1.5e-07	-1.2e-08	6.8e-10	6.4e-10	2.1e-10
alpha_3	22	0.67	-0.082	-0.012	-0.0019	-0.00031	-5e-05	-8.1e-06	-1.3e-06	-2e-07	-3e-08	-4.3e-09	-5.8e-10	-6.9e-11
alpha_5	28	0.54	-0.06	-0.0086	-0.0014	-0.00024	-4.1e-05	-7e-06	-1.2e-06	-2e-07	-3.2e-08	-5.3e-09	-8.7e-10	-1.4e-10
alpha_10	37	0.39	-0.038	-0.0056	-0.00097	-0.00017	-3e-05	-5.3e-06	-9.4e-07	-1.7e-07	-3e-08	-5.3e-09	-9.4e-10	-1.7e-10
alpha_20	47	0.26	-0.023	-0.0035	-0.00061	-0.00011	-2e-05	-3.7e-06	-6.7e-07	-1.2e-07	-2.3e-08	-4.1e-09	-7.6e-10	-1.4e-10

Se busca el mejor valor de alpha para el análisis de regresión de ridge

```
from sklearn.model_selection import GridSearchCV
RR=Ridge()
parameters1= [{'alpha': alpha_ridge}]
Grid1 = GridSearchCV(RR, parameters1, scoring="neg_mean_absolute_error")
Grid1.fit(data[predictors],data['y'])
BestRR=Grid1.best_estimator_
BestRR.alpha
```

```
0.5
```

Se define el modelo

```
ridgeBest = Ridge(alpha = BestRR.alpha, normalize = True)
ridgeBest.fit(data[predictors],data['y'])
```

```
Ridge(alpha=0.5, copy_X=True, fit_intercept=True, max_iter=None,
   normalize=True, random_state=None, solver='auto', tol=0.001)
```

El intercepto del modelo es:

```
ridgeBest.intercept_
```

```
1.1491142877800886
```

El coeficientes de las variables predictoras son:

```
ridgeBest.coef_
```

```
array([ -1.79010447e-01,  -2.45231550e-02,  -3.73781617e-03,
        -5.59892385e-04,  -7.98810467e-05,  -1.05469228e-05,
        -1.20775925e-06,  -9.41432446e-08,   4.96305117e-09,
         4.78814015e-09,   1.55596110e-09,   4.04233820e-10,
         9.50694032e-11,   2.11386993e-11,   4.53494511e-12])
```

4. CONCLUSIONES

- Como se ha podido observar a lo largo del presente trabajo, una alternativa válida para tratar el problema de multicolinealidad es la regresión Ridge.
- La Regresión Ridge produce predicciones mas precisas que los modelos obtenidos por MCO + selección clásica" de variables.
- Al aumentar Lambda (mayor penalización) los coeficientes estimados se contraen hacia cero, ninguno de ellos vale exactamente cero por lo cual no se produce selección de variables. Todas las variables originales permanecen en el modelo final.
- Para trabajar la regresión Ridge existen dos librerías: la librería car y la librería glmnet, para el presente trabajo se utilizó el primero con la función lm.ridge.
- La regresión Lasso es una técnica de regresión lineal regularizada Estable, que sirve también para selección de variables.

5. BIBLIOGRAFÍA

Akaike, H. (1974). *A new look at the statistical model identification.* IEEE Transactions on Automatic Control.

Anonimo. (s.f.). *Multicolinealidad.* disponible en https://www.uv.es/uriel/material/multicolinealidad3.pdf.

Carrasco C., M. (2013). *Tecnicas de regularizacion en regresion: implementacion y aplicaciones.* España: Universida de Sevilla.

Castro, S. (2013). *Estimacion y seleccion de variables en grandes dimensiones RR, GNN, Lasso, Elastic Net.* http://www.iesta.edu.uy/wp-content/uploads/2014/05/CursoPosgrado_Aprendizaje_Automatico_SCastro_2013.pdf: Curso de Posgrado .

Guerra de la Corte, A. (2016). España: universidad de Sevilla.

Harrell, F. (2015). *regression Modeling Strategies.* U.S.A: Springer.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning with applications in R.* U.S.A.: Springer.

Kleinbaum, D. G. (1988). *Applied regression analysis and other multivariate methods.* PWS-KEMT Publishing Co.

Nannen , V. (2003). *The paradox of overfitting.* Obtenido de http://volker.nannen.com/work/mdl

Rojo Abuín, J. M. (2007). *Regresión lineal múltiple.* Madrid: Instituto de Economía y Geografía.

Schwars, G. E. (1978). *Estimating the dimension of a model.* . Annals of Statistics.

Tusell, F. (2011). *Análisis de Regresión. Introducción Teórica y Práctica basada en R.* Disponible en: http://www.et.bs.ehu.es/~etptupaf/nuevo/ficheros/estad3/nreg1.pdf.

Vaquerizo, R. (2014). *Analisis y decision.* Obtenido de http://analisisydecision.es/regresion-ridge-o-contraida-con-r/

Wright, D. (2009). *Modern Regression Techniques Using R.* U.S.A: SAGE publications.

Algunos links de ayuda

- https://www.r-bloggers.com/ridge-regression-and-the-lasso/

- http://webcache.googleusercontent.com/search?q=cache:fV4k5zodQ00J:conferencias.unc.edu.ar/index.php/xclatse/clatse2012/paper/download/654/98+&cd=4&hl=es&ct=clnk&gl=pe
- https://stats.stackexchange.com/questions/108529/comparing-ols-ridge-and-lasso
-