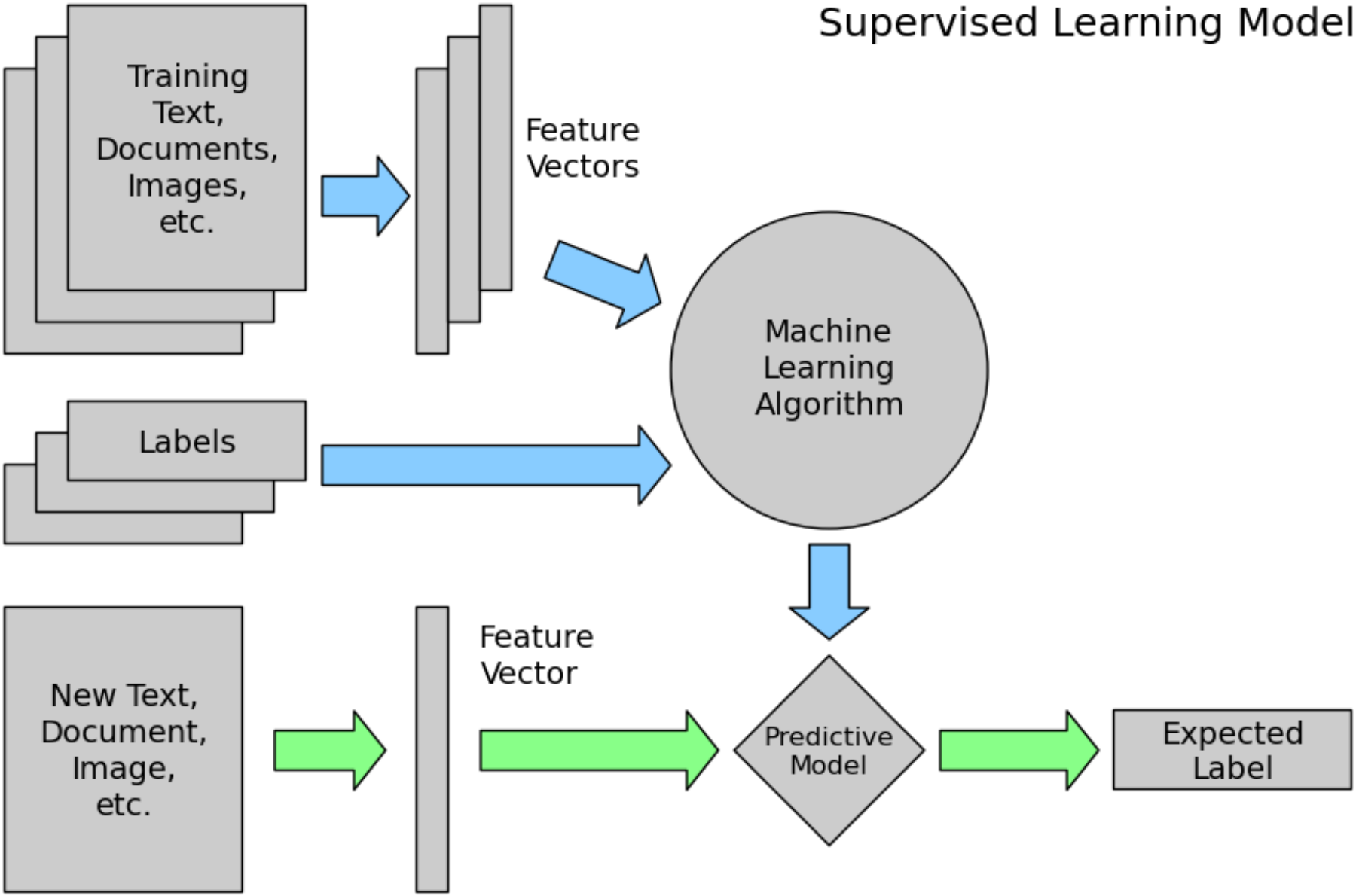
Aprendizaje Supervisado I: Regresión

July 28, 2018

Aprendizaje supervisado?

El aprendizaje supervisado no es más que, dado un input de *features* $\{x_1, x_2, \ldots, x_m\}$, el ajuste de un modelo que aproxime la función de $f(x_1, x_2, \ldots, x_m)$ mediante el aporte de los valores de la función conocida, que se conocen como *labels o targets* $y = f(x_1, x_2, \ldots, x_m)$. Es aquí donde entra la supervisión. Una vez ajustada la función a nivel óptimo, lo que se pretende es hacer predicciones del target de nuevos inputs.


Supervised Learning Model
Training Text, Documents, Images, etc.
Feature Vectors
Machine Learning Algorithm
Labels
New Text, Document, Image, etc.
Feature Vector
Predictive Model
Expected Label

regresión

La regresión no es más que el aprendizaje supervisado donde lo que se pretende predecir es una variable **continua**.
Existen muchos ejemplos en la naturaleza en lo que hacer esto puede ser útil:

- Uso de regresión por parte de una compañía farmacéutica para evaluar la estabilidad de un ingrediente activo en un medicamento para predecir su vida útil a fin de cumplir con las regulaciones impuestas e identificar una fecha de vencimiento adecuada para el medicamento.
- Una compañía de tarjetas de crédito aplica métodos de regresión para predecir las ventas de tarjetas de regalo y así mejorar las proyecciones de ingresos anuales.
- Una compañía de seguros puede utilizar herramientas de regresión para determinar la probabilidad de que exista un problema real cuando se presenta un reclamo de seguro de hogar o coche, con el fin de desalentar a los clientes de presentar reclamaciones excesivas.

regresión lineal

El modelo de regresión lineal se puede escribir de la siguiente forma:

$$y \to f(x_1, x_2, \ldots x_m) = \beta_0 + \sum_{j=1}^{m} \beta_j x_j \tag{1}$$

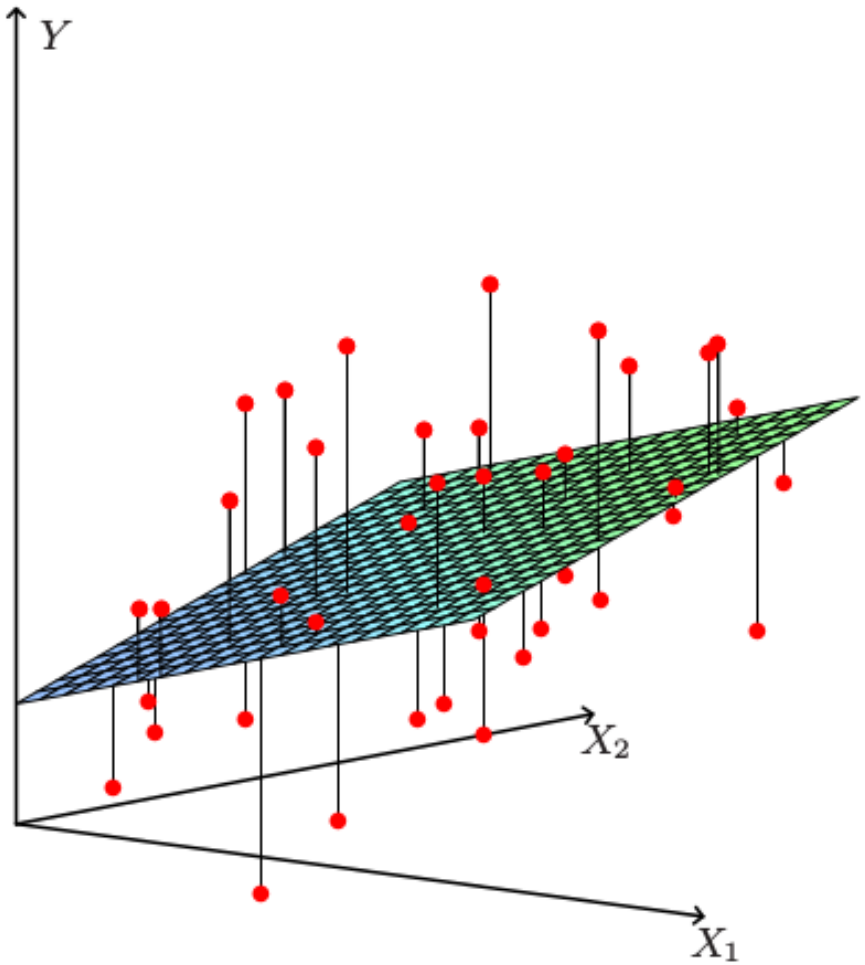
β_j son los coeficientes que debemos ajustar y x_j las diferentes observaciones de la variable j. Estas variables a su vez pueden ser:

- datos cuantitativos individuales (Edad, sexo, altura, etc)
- funciones de los anteriores tipos de datos , por ejemplo, $\log(x)$, $\sqrt{x}$...
- potencias de una sola variable dando lugar a un desarrollo exponencial $f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \ldots$
- Interacción entre variables $f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_1 x_2 + \beta_3 x_1 x_2 x_3 + \ldots$

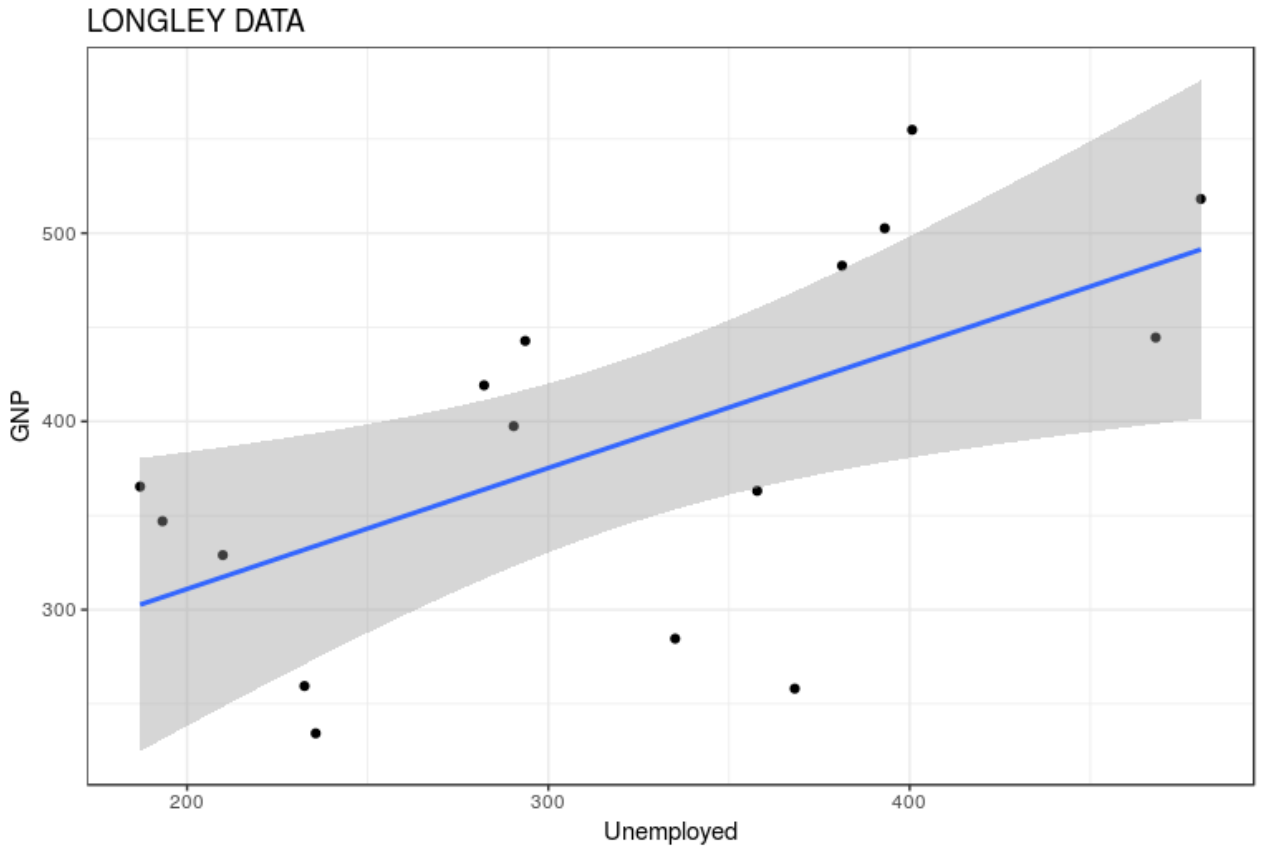
En nuestros datos cuando hacemos regresión, lo que queremos es que los datos predichos sean lo mayormente posible iguales a sus valores observados.

$$RSS = \sum_{i=1}^{N} (y_i - f(\mathbf{x_i^T}))^2$$
$$= \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{m} x_{ij}\beta_j)^2$$

Visto esto, lo que queremos es minimizar esta función de arriba, ya que esto significaría que la distancia x-f(x) sea mínima, o lo que es lo mismo, que la predicciones se parezcan lo mayormente posible a y.


Y
X2
X1

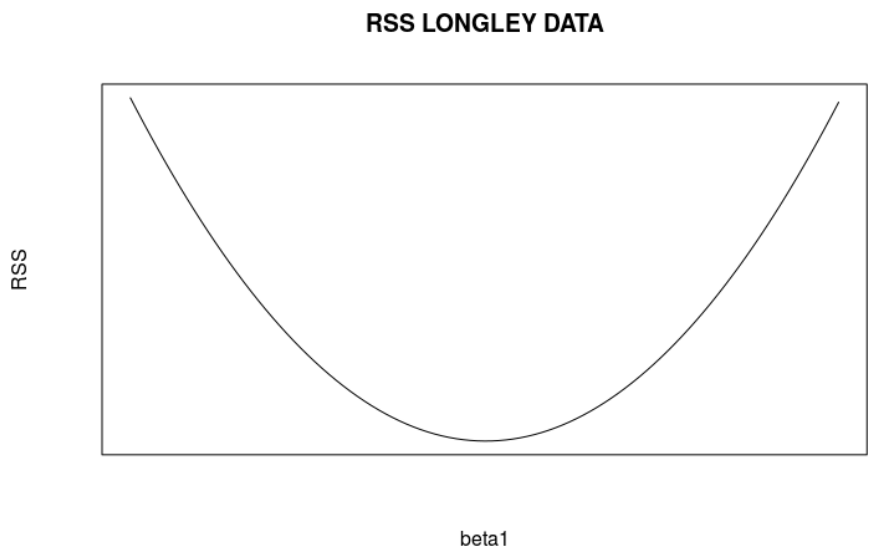
Veamos primero, para simplificar, el caso univariado. La función que tenemos por tanto es del tipo $f(x) = \beta_0 + \beta_1 x$



Para este caso, el error es muy fácil:

$$RSS = \sum_{i}^{N} (y_i - \beta_0 - x_{i1}\beta_1)^2 \tag{2}$$

De esta forma, cambiando β lo predicho se acerca cada vez más a su valor esperado y el error va disminuyendo. Se trata de un problema de optimización convexo.



En el caso más general, minimizando RSS, podemos encontrar los valores de β

$$\frac{dRSS'}{d\beta_0} = -2\sum_{i}^{N}(y_i - \beta_0 - \sum_{j=1}^{m} x_{ij}\beta_j) = 0 \tag{3}$$

$$\frac{dRSS'}{d\beta_k} = -2\sum_{i}^{N} x_{ik}(y_i - \beta_0 - \sum_{j=1}^{m} x_{ij}\beta_j) = 0 \tag{4}$$

o de forma matricial

$$2X^T(X\beta - \mathbf{y}) = 0 \tag{5}$$

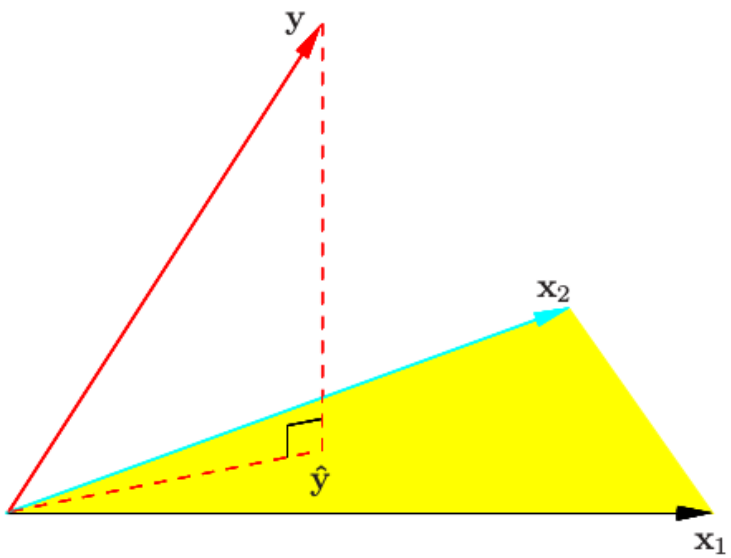
Si lo hacemos para todo los β's nos da un sistema de ecuacions, cuya solución **única** es

$$\boxed{\beta^{fit} = (X^T X)^{-1} X^T \mathbf{y}} \tag{6}$$

Entonces, si sustituimos la solución para los β's

$$\hat{\mathbf{y}} = X\beta^{fit} = X(X^T X)^{-1} X^T \mathbf{y} = H\mathbf{y}, \tag{7}$$

donde $H = X(X^T X)^{-1} X^T$ se suele conocer como hat matrix.



En scikit el modelo de regresión lineal resolviendo esto está implentada con el nombre **linear_ model.LinearRegression**

Gradiente desciente

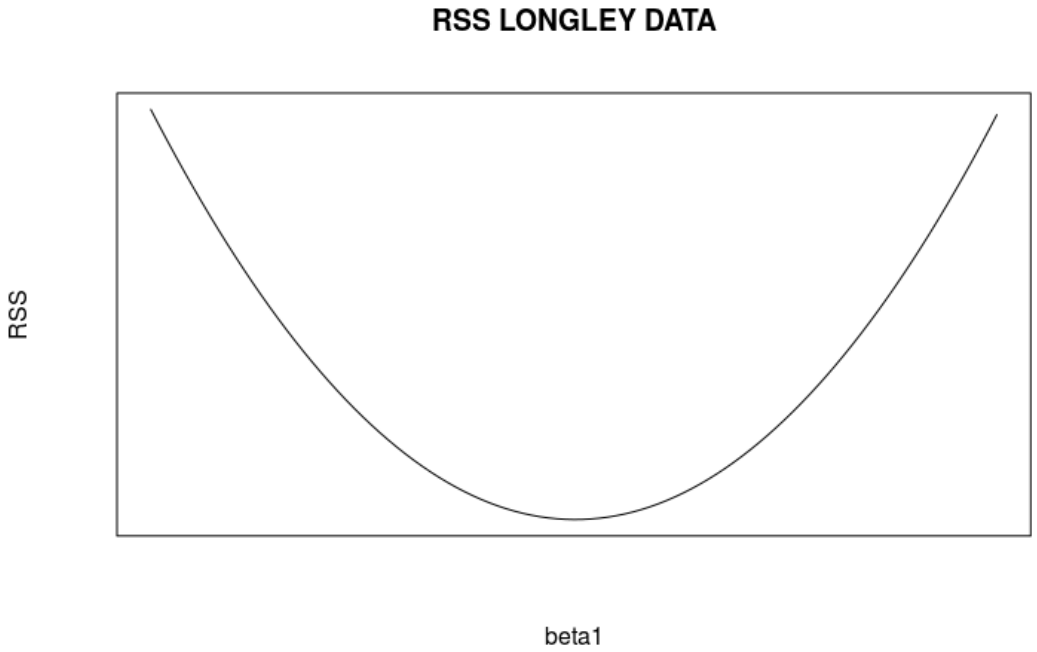
- Si $X^T X$ es invertible, tenemos una solución exacta para los coeficientes β.
- Calcular esta solución puede ser muy lenta computacionalmente, sobre todo, cuando el número de observaciones es alto.
- Volvamos a recordar otra vez las ecuaciones de antes:

$$RSS = \sum_{i}^{N}(y_i - \beta_0 - \sum_{j=1}^{m} x_{ij}\beta_1)^2 \tag{8}$$

$$\frac{dRSS'}{d\beta_0} = -2\sum_{i}^{N}(y_i - \beta_0 - \sum_{j=1}^{m} x_{ij}\beta_j) \tag{9}$$

$$\frac{dRSS'}{d\beta_k} = -2\sum_{i}^{N} x_{ik}(y_i - \beta_0 - \sum_{j=1}^{m} x_{ij}\beta_j) \tag{10}$$

Una forma de encontrar el mínimo de RSS es, sabiendo sus derivadas (que nos dan la **pendiente** sobre una curva), usar éstas para ir moviéndonos por RSS hasta alcanzar el mínimo.



$$\beta_k \rightarrow \beta_k \pm \alpha \frac{\partial RSS}{\partial \beta_k}$$

GRADIENTE DESCENDIENTE

Gradiente desciente

Este algoritmo aparece en muchos otros métodos de machine learning. Destaca por:

- Uno tiene que elegir un paso α
- A diferencia de antes, escala muy bien con el número de observaciones.
- Necesita muchas iteraciones
- Las variables tienen que tener el mismo orden de magnitud (función *preprocessing* en scikit)
- Es muy sensible a las condiciones iniciales, lo que siginifica que puede acabar en un mínimo local
- Sensible al paso α

En scikit está implentada con el nombre **linear_ model.SGDRegressor**

Regularización

El problema es $X^T X$ puede no ser invertible y por lo tanto las β's no estarían univocamente definida. Esto puede ocurrir en los siguientes casos:

- Usando variables redundantes, que tengan dependencias lineales. Por ejemplo, un cambio de escala o punto de referencia. $x_1 = time(s)$ y $x_2 = time(h)$
- Cuando hay más variables que observaciones. Para estos casos, habría que añadir eliminar variables.

La regularización, que añade restricciones sobre las variables, soluciona todos estos problemas.

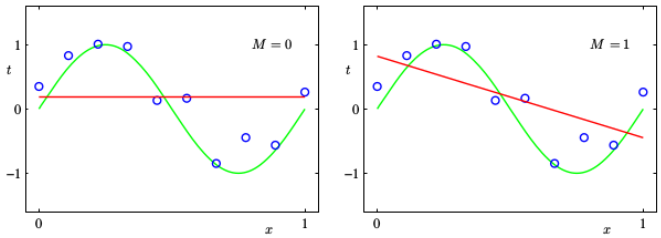
Métodos de regularización

Consideremos una **regresión polinomial**, de tal forma que nuestra función predictora es del tipo:

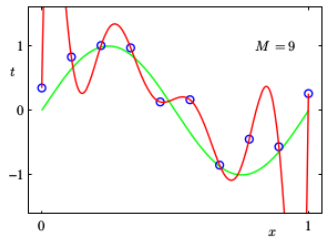
$$y \to f(x) = \beta_0 + \sum_{j=1}^{m} \beta_j x^j \tag{11}$$

Como cambiará el ajuste según añadimos más y más potencias?

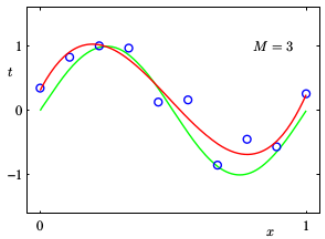
- Pocas potencias (variables) hace que no ajusten bien la curva a las observaciones. Se dice que en este caso, el modelo sufre de mucho **BIAS**



- Muchas potencias (variables) hacen la curva se ajuste demasiado bien a los puntos, de manera muy compleja. Se dice que en este caso, el modelo sufre de mucho **OVERFITTING**



- Lo ideal es siempre encontrar un equilibrio entre bias y overfitting



- El exceso de bias suele deberse a falta de variables. La solución pasa por añadir más variables para hacer el ajuste. Esto no suele ser un problema hoy en día
- El problema de overfitting suele ser más preocupante, ya que encontramos resultados demasiado optimistas y que no son generalizables.
- Una solución para evitar overfitting es eliminar variables.
- La otra consiste en añadir restricciones a las variables. Esto se conoce como **regularización**

- Lo que tenemos que hacer mediante la regularización es controlar la importancia de las variables en la formula del error

$$RSS(\beta) \rightarrow RSS(\beta) + Q(\beta) \tag{12}$$

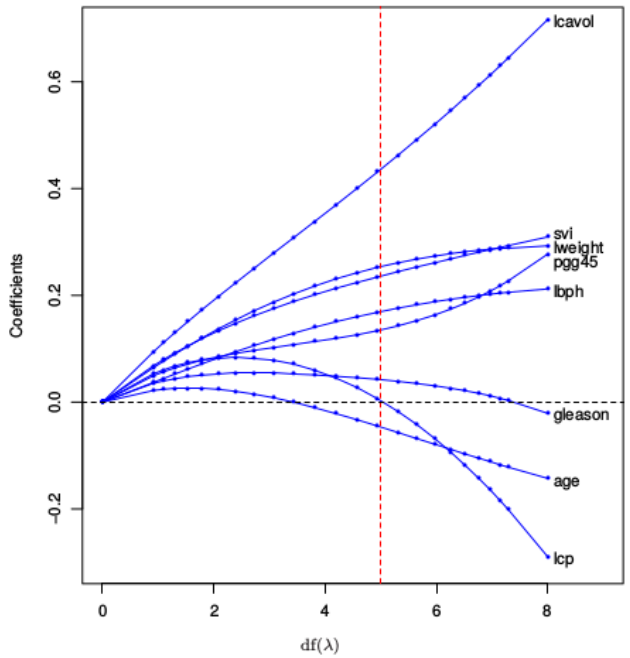
- Según el tipo de regularizador, esto nos da un algoritmo de regresión diferente

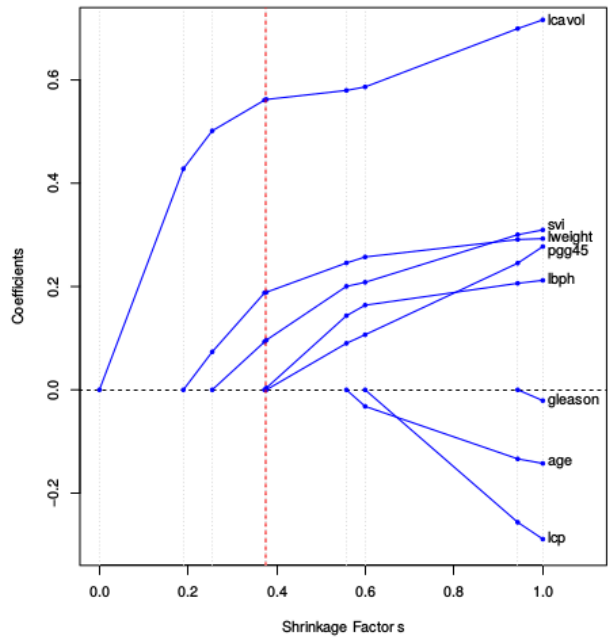
$$Q^{ridge}(\beta) = \lambda \sum_{j=1}^{m} \beta_j^2 \quad \textbf{linear\_model.Ridge} \tag{13}$$

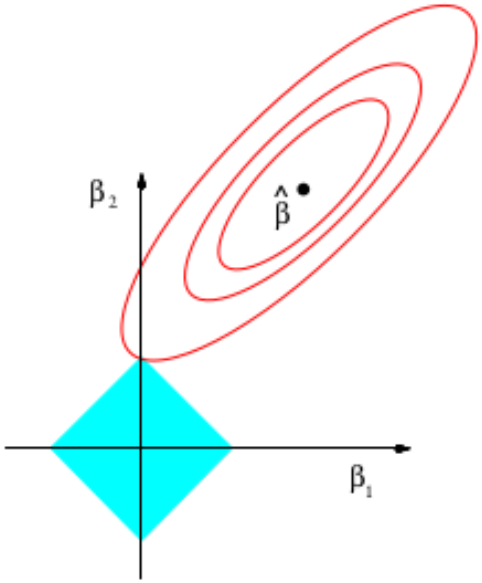
$$Q^{lasso}(\beta) = \lambda \sum_{j=1}^{m} |\beta_j| \quad \textbf{linear\_model.Lasso} \tag{14}$$

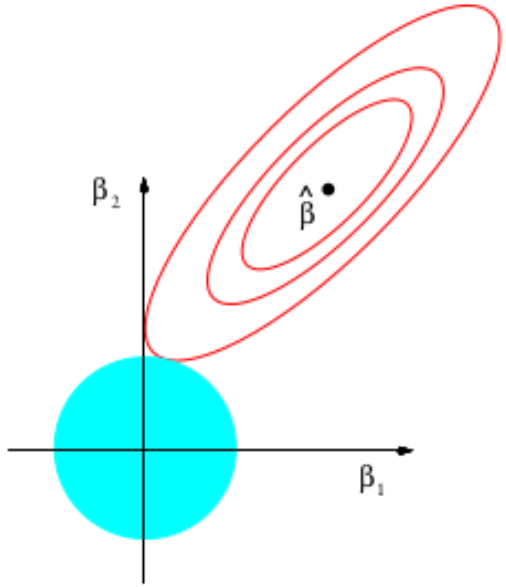
$$Q^{elasticNet}(\beta) = \lambda_1 \sum_{j=1}^{m} \beta_j^2 + \lambda_2 \sum_{j=1}^{m} |\beta_j| \quad \textbf{linear\_model.ElasticNet} \tag{15}$$

La constante de regularización λ cambiar los coeficientes β









Métricas en regresión

¿Qué medidas tenemos para decir que un modelo está bien ajustado (calculando sus coeficientes β's) o que su predicción en nuevos datos es óptimo?

- Varianza explicada $(y^{true}, y^{pred}) = 1 - \frac{Var(y^{true} - y^{pred})}{Var(y^{true})}$ **metrics.explained_ variance_ score**
- Error absoluto medio $M(y_{true}, y_{pred}) = \frac{1}{N}\sum_{i=1}^{N} |y_{true} - y_{pred}|$ **metrics.mean_ absolute_ error**
- Error cuadrado medio $RSS(y_{true}, y_{pred}) = \frac{1}{N}\sum_{i=1}^{N}(y_{true} - y_{pred})^2$ **metrics.mean_ squared_ error**
- Error absoluto mediano $median(|y_1^{true} - y_1^{pred}|, \ldots, |y_N^{true} - y_N^{pred}|)$ **metrics.mean_ absolute_ error**
- $R^2(y_{true}, y_{pred}) = 1 - \frac{\sum_{i=1}^{N}(y_{true} - y_{pred})^2}{\sum_{i=1}^{N}(y_{true} - <y>)^2}$ **metrics.mean_ r2_ score**