Data Mining and Machine Learning: Fundamental Concepts and Algorithms

datamininingbook.info

Mohammed J. Zaki[1] Wagner Meira Jr.[2]

[1]Department of Computer Science
Rensselaer Polytechnic Institute, Troy, NY, USA

[2]Department of Computer Science
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Chapter 23: Linear Regression

Regression

Given $X_1, X_2, \cdots, X_d$ (*predictor*, *explanatory*, or *independent* variables), and given Y (*response* or *dependent* variable), *regression* aims to predict Y based on X.

That is, the goal is to learn a *regression function* f, such that

$$Y = f(X_1, X_2, \cdots, X_d) + \varepsilon = f(\boldsymbol{X}) + \varepsilon$$

where $\boldsymbol{X} = (X_1, X_2, \cdots, X_d)^T$ is the multivariate random variable comprising the predictor attributes, and ε is a random *error term* that is assumed to be independent of $\boldsymbol{X}$.

Y is comprised of two components, one dependent on X, and the other, coming from the error term, independent of the predictor attributes.

The error term encapsulates inherent uncertainty in Y, as well as, possibly the effect of unobserved, hidden or *latent* variables.

Linear Regression

In *linear regression* the function f is assumed to be linear in $\boldsymbol{X}$, that is

$$f(\boldsymbol{X}) = \beta + \omega_1 X_1 + \omega_2 X_2 + \cdots + \omega_d X_d = \beta + \sum_{i=1}^{d} \omega_i X_i = \beta + \boldsymbol{\omega}^T \boldsymbol{X}$$

β is the true (unknown) *bias* term, ω_i is the true (unknown) *regression coefficient* or *weight* for attribute X_i, and $\boldsymbol{\omega} = (\omega_1, \omega_2, \cdots, \omega_d)^T$ is the true d-dimensional weight vector.

f specifies a hyperplane in $\mathbb{R}^{d+1}$, where $\boldsymbol{\omega}$ is the the weight vector that is normal or orthogonal to the hyperplane, and β is the *intercept* or offset term.

f is completely specified by the $d+1$ parameters comprising β and ω_i, for $i = 1, \cdots, d$.

Linear regression

A common approach to predicting the bias and regression coefficients is to use the method of *least squares*.

Given the training data $\boldsymbol{D}$ with points $\boldsymbol{x}_i$ and response values y_i (for $i = 1, \cdots, n$), we seek values b and $\boldsymbol{w}$, so as to minimize the sum of squared residual errors (SSE)

$$\boxed{SSE = \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \left(y_i - b - \boldsymbol{w}^T \boldsymbol{x}_i\right)^2}$$

In bivariate regression, $\boldsymbol{D}$ comprises a single predictor attribute, $X = (x_1, x_2, \cdots, x_n)^T$, along with $Y = (y_1, y_2, \cdots, y_n)^T$:

$$\boxed{\hat{y}_i = f(x_i) = b + w \cdot x_i}$$

Bivariate Regression

The residual error is $\epsilon_i = y_i - \hat{y}_i$ and the best line that minimizes the SSE:

$$\boxed{\min_{b,w} SSE = \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - b - w \cdot x_i)^2}$$

We differentiate it with respect to b and set the result to 0:

$$\frac{\partial}{\partial b} SSE = -2 \sum_{i=1}^{n} (y_i - b - w \cdot x_i) = 0$$

$$\implies b = \frac{1}{n} \sum_{i=1}^{n} y_i - w \cdot \frac{1}{n} \sum_{i=1}^{n} x_i$$

Therefore, we have

$$\boxed{b = \mu_Y - w \cdot \mu_X}$$

Bivariate Regression

Differentiating with respect to w, we obtain

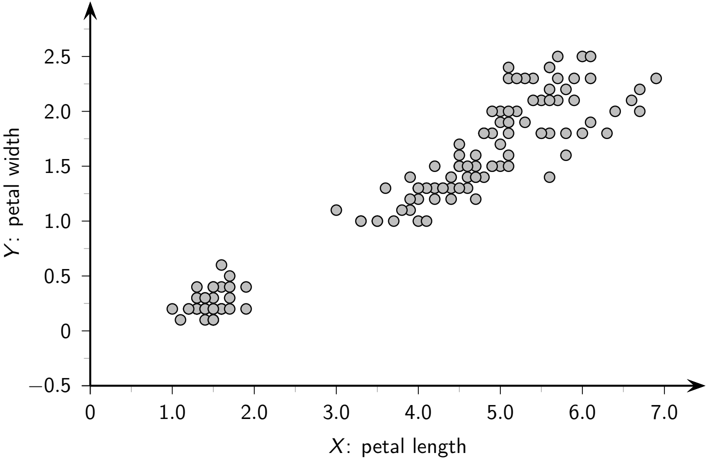
$$
\begin{aligned}
\frac{\partial}{\partial w} SSE &= -2 \sum_{i=1}^{n} x_i (y_i - b - w \cdot x_i) = 0 \\
&\implies \sum_{i=1}^{n} x_i \cdot y_i - b \sum_{i=1}^{n} x_i - w \sum_{i=1}^{n} x_i^2 = 0 \\
&\implies \sum_{i=1}^{n} x_i \cdot y_i - \mu_Y \sum_{i=1}^{n} x_i + w \cdot \mu_X \sum_{i=1}^{n} x_i - w \sum_{i=1}^{n} x_i^2 = 0 \\
&\implies w = \frac{\sum_{i=1}^{n} x_i \cdot y_i - n \cdot \mu_X \cdot \mu_Y}{\sum_{i=1}^{n} x_i^2 - n \cdot \mu_X^2}
\end{aligned}
$$

The regression coefficient w can also be written as

$$
\boxed{w = \frac{\sum_{i=1}^{n} (x_i - \mu_X)(y_i - \mu_Y)}{\sum_{i=1}^{n} (x_i - \mu_X)^2} = \frac{\sigma_{XY}}{\sigma_X^2} = \frac{\text{cov}(X, Y)}{\text{var}(X)}}
$$

Bivariate Regression

Given two attributes `petal length` (X; the predictor variable) and `petal width` (Y; the response variable) in the Iris dataset ($n = 150$).



Bivariate Regression

Example

The mean values for these two variables are

$$\mu_X = \frac{1}{150}\sum_{i=1}^{150} x_i = \frac{563.8}{150} = 3.7587$$

$$\mu_Y = \frac{1}{150}\sum_{i=1}^{150} y_i = \frac{179.8}{150} = 1.1987$$

The variance and covariance is given as

$$\sigma_X^2 = \frac{1}{150}\sum_{i=1}^{150} (x_i - \mu_X)^2 = 3.0924$$

$$\sigma_Y^2 = \frac{1}{150}\sum_{i=1}^{150} (y_i - \mu_Y)^2 = 0.5785$$

$$\sigma_{XY} = \frac{1}{150}\sum_{i=1}^{150} (x_i - \mu_X)\cdot(y_i - \mu_Y) = 1.2877$$

Assuming a linear relationship between the response and predictor variables, we obtain the slope and intercept terms as follows

$$w = \frac{\sigma_{XY}}{\sigma_X^2} = \frac{1.2877}{3.0924} = 0.4164$$

$$b = \mu_Y - w \cdot \mu_X = 1.1987 - 0.4164 \cdot 3.7587 = -0.3665$$

Thus, the fitted regression line is

$$\hat{y} = -0.3665 + 0.4164 \cdot x$$

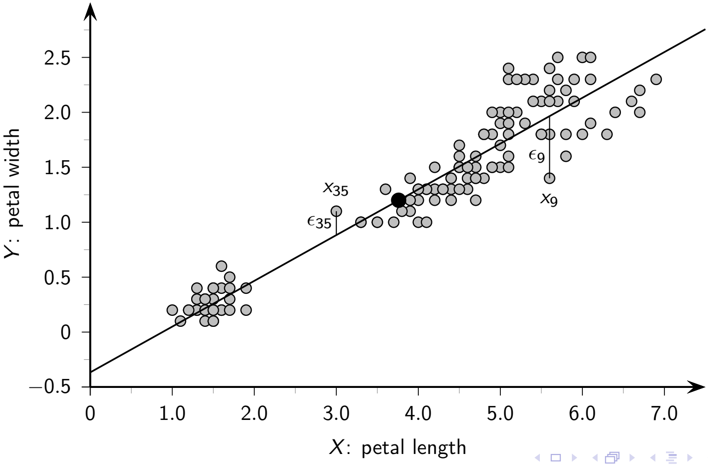
Finally, we can compute the SSE value as follows:

$$SSE = \sum_{i=1}^{150} \epsilon_i^2 = \sum_{i=1}^{150} (y_i - \hat{y}_i)^2 = 6.343$$

Bivariate Regression

Example

`petal length` (X) versus `petal width` (Y). Solid circle (black) shows the mean point; residual error is shown for two sample points: x_9 and x_{35}.

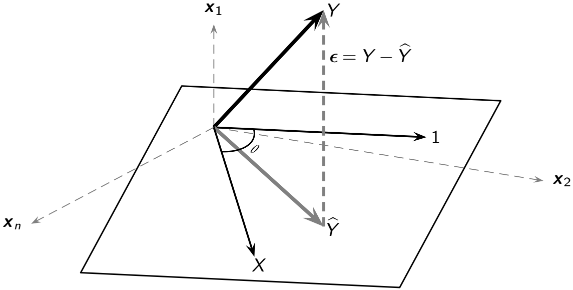


Geometry of Bivariate Regression

We can express the n equations, $y_i = b + w \cdot x_i$ for $i = 1, 2, \cdots, n$, as:

$$\widehat{Y} = b \cdot 1 + w \cdot X$$

where $1 \in \mathbb{R}^n$ is the n-dimensional vector of 1s. $\widehat{Y}$ is a linear combination of 1 and X, i.e., it must lie in the column space spanned by 1 and X, given as $\text{span}(\{1, X\})$. ϵ captures the deviation between Y and $\widehat{Y}$.



Geometry of Bivariate Regression

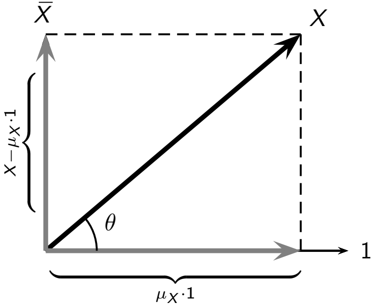
Orthogonal decomposition of X into $\bar{X}$ and $\mu_X \cdot 1$.

Even though 1 and X are linearly independent and form a basis for the column space, they need not be orthogonal.

We can create an orthogonal basis by decomposing X into a component along 1 and a component orthogonal to 1, $\bar{X}$.

$$X = \mu_X \cdot 1 + (X - \mu_X \cdot 1) = \mu_X \cdot 1 + \bar{X}$$

where $\bar{X} = X - \mu_X \cdot 1$ is the centered attribute vector.



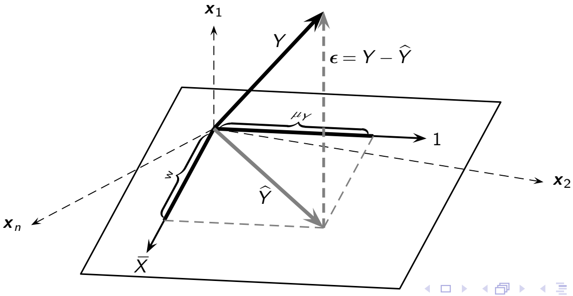
Geometry of Regression

The optimal $\widehat{Y}$ that minimizes the error is the orthogonal projection of Y onto the subspace spanned by 1 and X.

The residual error vector ϵ is thus *orthogonal* to the subspace spanned by 1 and X, and its squared length (or magnitude) equals the SSE value.

Summarizing:

$\mu_Y = \text{proj}_1(Y)$ $\qquad$ $w = \text{proj}_{\bar{X}}(Y)$ $\qquad$ $b = \mu_Y - w \cdot \mu_X$



Geometry of Regression

Example

Let us consider the regression of `petal length` (X) on `petal width` (Y) for the Iris dataset, with $n = 150$. First, we center X by subtracting the mean $\mu_X = 3.759$. Next, we compute the scalar projections of Y onto 1 and $\overline{X}$, to obtain

$$\mu_Y = \text{proj}_1(Y) = \left(\frac{Y^T 1}{1^T 1}\right) = \frac{179.8}{150} = 1.1987$$

$$w = \text{proj}_{\overline{X}}(Y) = \left(\frac{Y^T \overline{X}}{\overline{X}^T \overline{X}}\right) = \frac{193.16}{463.86} = 0.4164$$

Thus, the bias term b is given as

$$b = \mu_Y - w \cdot \mu_X = 1.1987 - 0.4164 \cdot 3.7587 = -0.3665$$

We can compute the SSE value as the squared length of the residual error vector

$$SSE = \|\epsilon\|^2 = \left\|Y - \widehat{Y}\right\|^2 = (Y - \widehat{Y})^T (Y - \widehat{Y}) = 6.343$$

Multiple Regression

Multiple regression: multiple predictor attributes $X_1, X_2, \cdots, X_d$ and a single response attribute Y.

The training data sample $\boldsymbol{D} \in \mathbb{R}^{n \times d}$ comprises n points $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{id})^T$ in a d-dimensional space, along with the corresponding observed response value y_i.

Instead of dealing with the bias b separately from the weights w_i, we can introduce a new "constant" valued attribute X_0 whose value is always fixed at 1.

The predicted response value for an augmented $(d+1)$ dimensional point $\tilde{\boldsymbol{x}}_i$ can be written as

$$\boxed{\hat{y}_i = w_0 x_{i0} + w_1 x_{i1} + w_2 x_{i2} + \cdots + w_d x_{id} = \tilde{\boldsymbol{w}}^T \tilde{\boldsymbol{x}}_i}$$

Multiple Regression

The multiple regression task is to find the *best fitting hyperplane* defined by $\tilde{\boldsymbol{w}}$ that minimizes the SSE:

$$\begin{aligned}
\min_{\tilde{\boldsymbol{w}}} SSE &= \sum_{i=1}^{n} \epsilon_i^2 = \|\boldsymbol{\epsilon}\|^2 = \left\| Y - \widehat{Y} \right\|^2 \\
&= (Y - \widehat{Y})^T (Y - \widehat{Y}) = Y^T Y - 2Y^T \widehat{Y} + \widehat{Y}^T \widehat{Y} \\
&= Y^T Y - 2Y^T (\tilde{\boldsymbol{D}} \tilde{\boldsymbol{w}}) + (\tilde{\boldsymbol{D}} \tilde{\boldsymbol{w}})^T (\tilde{\boldsymbol{D}} \tilde{\boldsymbol{w}}) \\
&= Y^T Y - 2\tilde{\boldsymbol{w}}^T (\tilde{\boldsymbol{D}}^T Y) + \tilde{\boldsymbol{w}}^T (\tilde{\boldsymbol{D}}^T \tilde{\boldsymbol{D}}) \tilde{\boldsymbol{w}}
\end{aligned}$$

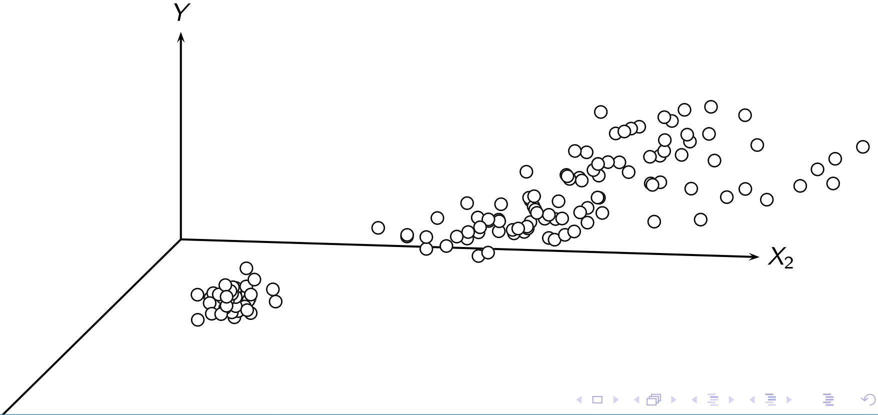
Therefore, the optimal weight vector is given as

$$\boxed{\tilde{\boldsymbol{w}} = (\tilde{\boldsymbol{D}}^T \tilde{\boldsymbol{D}})^{-1} \tilde{\boldsymbol{D}}^T Y}$$

Multiple Regression

Example

Given `sepal length` (X_1) and `petal length` (X_2) on the response attribute `petal width` (Y) for the Iris dataset with $n = 150$ points, we want to learn the multiple regression.



Multiple Regression

Example

We and $X_0 = 1_{150}$ and $\tilde{\boldsymbol{D}} \in \mathbb{R}^{150\times 3}$. We then compute $\tilde{\boldsymbol{D}}^T\tilde{\boldsymbol{D}}$ and its inverse

$$\tilde{\boldsymbol{D}}^T\tilde{\boldsymbol{D}} = \begin{pmatrix} 150.0 & 876.50 & 563.80 \\ 876.5 & 5223.85 & 3484.25 \\ 563.8 & 3484.25 & 2583.00 \end{pmatrix} \quad (\tilde{\boldsymbol{D}}^T\tilde{\boldsymbol{D}})^{-1} = \begin{pmatrix} 0.793 & -0.176 & 0.064 \\ -0.176 & 0.041 & -0.017 \\ 0.064 & -0.017 & 0.009 \end{pmatrix}$$

We also compute $\tilde{\boldsymbol{D}}^T Y$, given as

$$\tilde{\boldsymbol{D}}^T Y = \begin{pmatrix} 179.80 \\ 1127.65 \\ 868.97 \end{pmatrix}$$

The augmented weight vector $\tilde{\boldsymbol{w}}$ is then given as

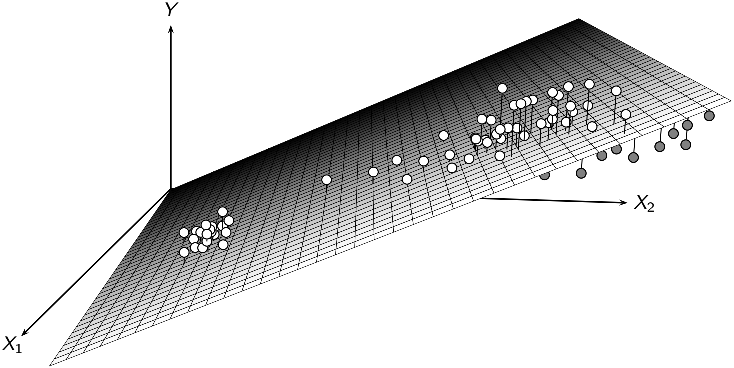
$$\tilde{\boldsymbol{w}} = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix} = (\tilde{\boldsymbol{D}}^T\tilde{\boldsymbol{D}})^{-1} \cdot (\tilde{\boldsymbol{D}}^T Y) = \begin{pmatrix} -0.014 \\ -0.082 \\ 0.45 \end{pmatrix}$$

Therefore $b = w_0 = -0.014$, and $\widehat{Y} = -0.014 - 0.082 \cdot X_1 + 0.45 \cdot X_2$

Multiple Regression

Example

Figure shows the fitted hyperplane and the residual error for each point. Positive residuals (i.e., $\epsilon_i > 0$ or $\hat{y}_i > y_i$) are white, while negative residuals (i.e., $\epsilon_i < 0$ or $\hat{y}_i < y$) are gray. The SSE value for the model is 6.18.



Multiple-Regression Algorithm

The algorithm is based on the QR-factorization, which expresses a matrix as a product of two separate matrices, Q (an orthogonal matrix), and R (an upper/right triangular matrix).

```
**Multiple-Regression** ($\boldsymbol{D}, Y$):
$\tilde{\boldsymbol{D}} \leftarrow \begin{pmatrix} 1 & \boldsymbol{D} \end{pmatrix}$ // augmented data with $X_0 = 1 \in \mathbb{R}^n$
$\{\boldsymbol{Q}, \boldsymbol{R}\} \leftarrow$ QR-factorization($\tilde{\boldsymbol{D}}$) // $\boldsymbol{Q} = \begin{pmatrix} U_0 & U_1 & \cdots & U_d \end{pmatrix}$
$\Delta^{-1} \leftarrow \begin{pmatrix} \frac{1}{\|U_0\|^2} & 0 & \cdots & 0 \\ 0 & \frac{1}{\|U_1\|^2} & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \frac{1}{\|U_d\|^2} \end{pmatrix}$ // squared norms
$\boldsymbol{R}\boldsymbol{w} \leftarrow \Delta^{-1}\boldsymbol{Q}^T Y$ // solve for $\boldsymbol{w}$ by back-substitution
$\widehat{Y} \leftarrow \boldsymbol{Q}\Delta^{-1}\boldsymbol{Q}^T Y$
```

QR-Factorization and Geometric Approach

Example

Consider the multiple regression of `sepal length` (X_1) and `petal length` (X_2) on `petal width` (Y) for the Iris dataset with $n = 150$ points.

The Gram–Schmidt orthogonalization results in the following QR-factorization:

$$\underbrace{\begin{pmatrix} | & | & | \\ X_0 & X_1 & X_2 \\ | & | & | \end{pmatrix}}_{\tilde{\boldsymbol{D}}} = \underbrace{\begin{pmatrix} | & | & | \\ U_0 & U_1 & U_2 \\ | & | & | \end{pmatrix}}_{\boldsymbol{Q}} \cdot \underbrace{\begin{pmatrix} 1 & 5.843 & 3.759 \\ 0 & 1 & 1.858 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{R}}$$

$\boldsymbol{Q} \in \mathbb{R}^{150\times 3}$ and Δ, the squared norms of the basis vectors, and its inverse are

$$\Delta = \begin{pmatrix} 150 & 0 & 0 \\ 0 & 102.17 & 0 \\ 0 & 0 & 111.35 \end{pmatrix} \qquad \Delta^{-1} = \begin{pmatrix} 0.00667 & 0 & 0 \\ 0 & 0.00979 & 0 \\ 0 & 0 & 0.00898 \end{pmatrix}$$

QR-Factorization and Geometric Approach

Example

We can use back-substitution to solve for $\tilde{\boldsymbol{w}}$, as follows

$$\boldsymbol{R}\tilde{\boldsymbol{w}} = \Delta^{-1}\boldsymbol{Q}^T Y$$

$$\begin{pmatrix} 1 & 5.843 & 3.759 \\ 0 & 1 & 1.858 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 1.1987 \\ 0.7538 \\ 0.4499 \end{pmatrix}$$

Back-substitution starts with w_2:

$$w_2 = 0.4499$$

Next, w_1 is given as:

$$w_1 + 1.858 \cdot w_2 = 0.7538$$
$$\implies w_1 = 0.7538 - 0.8358 = -0.082$$

Finally, w_0 can be computed as

$$w_0 + 5.843 \cdot w_1 + 3.759 \cdot w_2 = 1.1987$$
$$\implies w_0 = 1.1987 + 0.4786 - 1.6911 = -0.0139$$

QR-Factorization and Geometric Approach

Example

The multiple regression model is given as

$$\widehat{Y} = -0.014 \cdot X_0 - 0.082 \cdot X_1 + 0.45 \cdot X_2$$

It is also instructive to construct the new basis vectors $U_0, U_1, \cdots, U_d$ in terms of $X_0, X_1, \cdots, X_d$. Since $\tilde{\boldsymbol{D}} = \boldsymbol{QR}$, we have $\boldsymbol{Q} = \tilde{\boldsymbol{D}}\boldsymbol{R}^{-1}$. The inverse of $\boldsymbol{R}$ is also upper-triangular, and is given as

$$\boldsymbol{R}^{-1} = \begin{pmatrix} 1 & -5.843 & 7.095 \\ 0 & 1 & -1.858 \\ 0 & 0 & 1 \end{pmatrix}$$

$\boldsymbol{Q}$ can be written as:

$$\underbrace{\begin{pmatrix} | & | & | \\ U_0 & U_1 & U_2 \\ | & | & | \end{pmatrix}}_{\boldsymbol{Q}} = \underbrace{\begin{pmatrix} | & | & | \\ X_0 & X_1 & X_2 \\ | & | & | \end{pmatrix}}_{\tilde{\boldsymbol{D}}} \underbrace{\begin{pmatrix} 1 & -5.843 & 7.095 \\ 0 & 1 & -1.858 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{R}^{-1}}$$

QR-Factorization and Geometric Approach

Example

This expression allows us to

$$
\begin{aligned}
U_0 &= X_0 \\
U_1 &= -5.843 \cdot X_0 + X_1 \\
U_2 &= 7.095 \cdot X_0 - 1.858 \cdot X_1 + X_2
\end{aligned}
$$

The scalar projections of Y onto U_i are:

$$
\text{proj}_{U_0}(Y) = 1.199 \qquad \text{proj}_{U_1}(Y) = 0.754 \qquad \text{proj}_{U_2}(Y) = 0.45
$$

The fitted response vector $\widehat{Y}$ is given as:

$$
\begin{aligned}
\widehat{Y} &= \text{proj}_{U_0}(Y) \cdot U_0 + \text{proj}_{U_1}(Y) \cdot U_1 + \text{proj}_{U_2}(Y) \cdot U_2 \\
&= 1.199 \cdot X_0 + 0.754 \cdot (-5.843 \cdot X_0 + X_1) + 0.45 \cdot (7.095 \cdot X_0 - 1.858 \cdot X_1 + X_2) \\
&= (1.199 - 4.406 + 3.193) \cdot X_0 + (0.754 - 0.836) \cdot X_1 + 0.45 \cdot X_2 \\
&= -0.014 \cdot X_0 - 0.082 \cdot X_1 + 0.45 \cdot X_2
\end{aligned}
$$

Multiple Regression: Stochastic Gradient Descent

Instead of using the QR-factorization approach to exactly solve the multiple regression problem, we can also employ the simpler stochastic gradient algorithm. The gradient of the SSE objective is given as

$$\nabla_{\tilde{\boldsymbol{w}}} = \frac{\partial}{\partial \tilde{\boldsymbol{w}}} SSE = -\tilde{\boldsymbol{D}}^T Y + (\tilde{\boldsymbol{D}}^T \tilde{\boldsymbol{D}}) \tilde{\boldsymbol{w}}$$

From an initial weight vector $\tilde{\boldsymbol{w}}^0$, we update $\tilde{\boldsymbol{w}}$ as:

$$\tilde{\boldsymbol{w}}^{t+1} = \tilde{\boldsymbol{w}}^t - \eta \cdot \nabla_{\tilde{\boldsymbol{w}}} = \tilde{\boldsymbol{w}}^t + \eta \cdot \tilde{\boldsymbol{D}}^T (Y - \tilde{\boldsymbol{D}} \cdot \tilde{\boldsymbol{w}}^t)$$

where $\tilde{\boldsymbol{w}}^t$ is the estimate of the weight vector at step t. We update the weight vector by considering only one (random) point at each iteration.

$$\begin{aligned} \tilde{\boldsymbol{w}}^{t+1} &= \tilde{\boldsymbol{w}}^t - \eta \cdot \nabla_{\tilde{\boldsymbol{w}}}(\tilde{\boldsymbol{x}}_k) \\ &= \tilde{\boldsymbol{w}}^t + \eta \cdot (y_k - \tilde{\boldsymbol{x}}_k \cdot \tilde{\boldsymbol{w}}^t) \cdot \tilde{\boldsymbol{x}}_k \end{aligned}$$

Multiple Regression: SGD Algorithm

```
Multiple Regression: SGD ($D, Y, \eta, \epsilon$):
$\tilde{D} \leftarrow \begin{pmatrix} 1 & D \end{pmatrix}$ // augment data
$t \leftarrow 0$ // step/iteration counter
$\tilde{w}^t \leftarrow$ random vector in $\mathbb{R}^{d+1}$ // initial weight vector
repeat
    foreach $k = 1, 2, \cdots, n$ (in random order) do
        $\nabla_{\tilde{w}}(\tilde{x}_k) \leftarrow -(y_k - \tilde{x}_k^T \tilde{w}^t) \cdot \tilde{x}_k$ // compute gradient at $\tilde{x}_k$
        $\tilde{w}^{t+1} \leftarrow \tilde{w}^t - \eta \cdot \nabla_{\tilde{w}}(\tilde{x}_k)$ // update estimate for $w_k$
    $t \leftarrow t + 1$
until $\left\| w^t - w^{t-1} \right\| \leq \epsilon$
```

Multiple Regression: SGD

Example

Multiple regression of `sepal length` (X_1) and `petal length` (X_2) on the response attribute `petal width` (Y) for the Iris dataset with $n = 150$ points.

Using the exact approach the multiple regression model was given as

$$\widehat{Y} = -0.014 \cdot X_0 - 0.082 \cdot X_1 + 0.45 \cdot X_2$$

Using SGD we obtain the following model with $\eta = 0.001$ and $\epsilon = 0.0001$:

$$\widehat{Y} = -0.031 \cdot X_0 - 0.078 \cdot X_1 + 0.45 \cdot X_2$$

The results from the SGD approach are essentially the same as the exact method, with a slight difference in the bias term.

The SSE value for the exact method is 6.179, whereas for SGD it is 6.181.

Ridge Regression

For linear regression, $\widehat{Y}$ lies in the span of the column vectors comprising the augmented data matrix $\tilde{\boldsymbol{D}}$.

Often the data is noisy and uncertain, and, therefore, instead of fitting the model to the data exactly, it may be better to fit a more robust model.

Regularization constrains the solution vector $\tilde{\boldsymbol{w}}$ to have a small norm.

Besides minimizing $\left\|Y - \widehat{Y}\right\|^2$, we add a regularization term ($\|\tilde{\boldsymbol{w}}\|^2$):

$$\boxed{\min_{\tilde{\boldsymbol{w}}} \; J(\tilde{\boldsymbol{w}}) = \left\|Y - \widehat{Y}\right\|^2 + \alpha \cdot \|\tilde{\boldsymbol{w}}\|^2 = \left\|Y - \tilde{\boldsymbol{D}}\tilde{\boldsymbol{w}}\right\|^2 + \alpha \cdot \|\tilde{\boldsymbol{w}}\|^2}$$

$\alpha \geq 0$ controls the tradeoff between minimizing the squared norm of the weight vector and the squared error.

Ridge Regression

We differentiate w.r.t. $\tilde{\boldsymbol{w}}$ and set the results to 0 to obtain

$$\tilde{\boldsymbol{w}} = (\tilde{\boldsymbol{D}}^T \tilde{\boldsymbol{D}} + \alpha \cdot \boldsymbol{I})^{-1} \tilde{\boldsymbol{D}}^T Y$$

The matrix $(\tilde{\boldsymbol{D}}^T \tilde{\boldsymbol{D}} + \alpha \cdot \boldsymbol{I})$ is always invertible (or non-singular) for $\alpha > 0$ even if $\tilde{\boldsymbol{D}}^T \tilde{\boldsymbol{D}}$ is not invertible (or singular).

If λ_i is an eigenvalue of $\tilde{\boldsymbol{D}}^T \tilde{\boldsymbol{D}}$, then $\lambda_i + \alpha$ is an eigenvalue of $(\tilde{\boldsymbol{D}}^T \tilde{\boldsymbol{D}} + \alpha \cdot \boldsymbol{I})$. Since $\tilde{\boldsymbol{D}}^T \tilde{\boldsymbol{D}}$ is positive semi-definite it has non-negative eigenvalues. Even if an $\lambda_i = 0$, the corresponding eigenvalue of $(\tilde{\boldsymbol{D}}^T \tilde{\boldsymbol{D}} + \alpha \cdot \boldsymbol{I})$ is $\lambda_i + \alpha = \alpha > 0$.

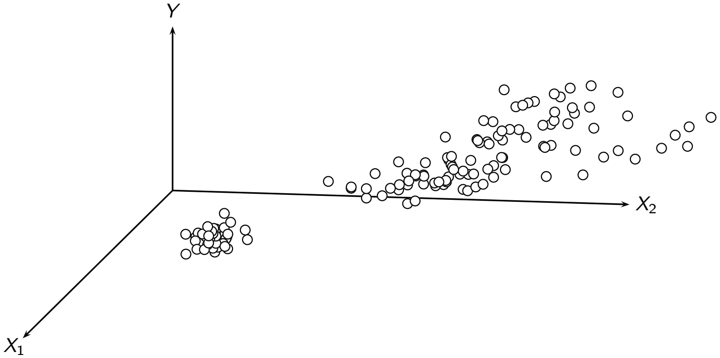
Regularized regression is called *ridge regression* because it adds a "ridge" along the main diagonal of the $\tilde{\boldsymbol{D}}^T \tilde{\boldsymbol{D}}$ matrix, i.e., the solution depends on $(\tilde{\boldsymbol{D}}^T \tilde{\boldsymbol{D}} + \alpha \cdot \boldsymbol{I})$.

If we choose a small positive α we are always guaranteed a solution.

Ridge Regression

Example

Given `sepal length` (X_1) and `petal length` (X_2) on the response attribute `petal width` (Y) for the Iris dataset with $n = 150$ points, we want to learn the ridge regression.



Ridge Regression

Example

The uncentered scatter matrix is given as

$$\tilde{\boldsymbol{D}}^T\tilde{\boldsymbol{D}} = \begin{pmatrix} 150.0 & 563.8 \\ 563.8 & 2583.0 \end{pmatrix}$$

We obtain different lines of best fit for different values of the regularization constant α:

$$\alpha = 0 : \widehat{Y} = -0.367 + 0.416 \cdot X, \quad \|\tilde{\boldsymbol{w}}\|^2 = \left\|(-0.367, 0.416)^T\right\|^2 = 0.308, \quad SSE = 6.34$$

$$\alpha = 10 : \widehat{Y} = -0.244 + 0.388 \cdot X, \quad \|\tilde{\boldsymbol{w}}\|^2 = \left\|(-0.244, 0.388)^T\right\|^2 = 0.210, \quad SSE = 6.75$$

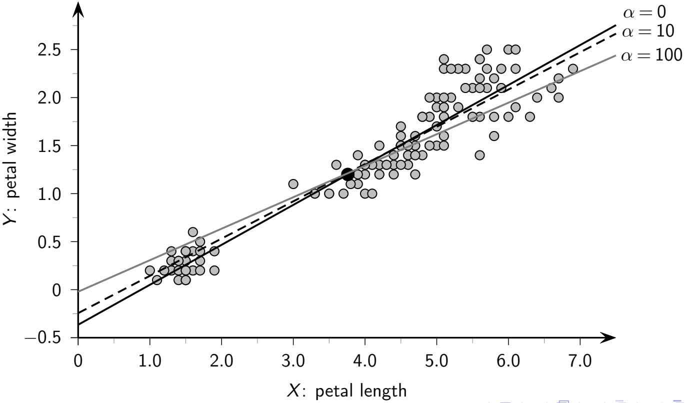
$$\alpha = 100 : \widehat{Y} = -0.021 + 0.328 \cdot X, \quad \|\tilde{\boldsymbol{w}}\|^2 = \left\|(-0.021, 0.328)^T\right\|^2 = 0.108, \quad SSE = 9.97$$

Ridge Regression

Example

As α increases there is more emphasis on minimizing the squared norm of $\tilde{\boldsymbol{w}}$.

Since $\|\tilde{\boldsymbol{w}}\|^2$ is more constrained as α increases, the fit of the model decreases, as seen from the increase in SSE values.



Ridge Regression: Unpenalized Bias Term

Often in L_2 regularized regression we do not want to penalize the bias term w_0, since it simply provides the intercept information.

Consider the new regularized objective where $\boldsymbol{w} = (w_1, w_2, \cdots, w_d)^T$ without w_0:

$$\begin{aligned}\min_{\boldsymbol{w}} \; J(\boldsymbol{w}) &= \left\| Y - w_0 \cdot 1 - \boldsymbol{D}\boldsymbol{w} \right\|^2 + \alpha \cdot \|\boldsymbol{w}\|^2 \\ &= \left\| Y - w_0 \cdot 1 - \sum_{i=1}^{d} w_i \cdot X_i \right\|^2 + \alpha \cdot \left(\sum_{i=1}^{d} w_i^2 \right)\end{aligned}$$

Therefore, we have

$$\boxed{\min_{\boldsymbol{w}} \; J(\boldsymbol{w}) = \left\| \overline{Y} - \overline{\boldsymbol{D}}\boldsymbol{w} \right\|^2 + \alpha \cdot \|\boldsymbol{w}\|^2}$$

where $\overline{Y} = Y - \mu_Y \cdot 1$ is the centered Y, and $\overline{\boldsymbol{D}} = \boldsymbol{D} - 1\boldsymbol{\mu}^T$ is the centered $\boldsymbol{D}$.

We can exclude w_0 from the L_2 regularization objective by centering the response vector and the unaugmented data matrix.

Ridge Regression: Unpenalized Bias

Example

When we do not penalize w_0, we obtain the following lines of best fit for different values of the regularization constant α:

$$\begin{aligned}
\alpha = 0 : \widehat{Y} &= -0.365 + 0.416 \cdot X & w_0^2 + w_1^2 &= 0.307 & SSE &= 6.34 \\
\alpha = 10 : \widehat{Y} &= -0.333 + 0.408 \cdot X & w_0^2 + w_1^2 &= 0.277 & SSE &= 6.38 \\
\alpha = 100 : \widehat{Y} &= -0.089 + 0.343 \cdot X & w_0^2 + w_1^2 &= 0.125 & SSE &= 8.87
\end{aligned}$$

We observe that for $\alpha = 10$, when we penalize w_0, we obtain the following model:

$$\alpha = 10 : \widehat{Y} = -0.244 + 0.388 \cdot X \qquad w_0^2 + w_1^2 = 0.210 \qquad SSE = 6.75$$

As expected, we obtain a higher bias term when we do not penalize w_0.

Ridge Regression: Stochastic Gradient Descent

Instead of inverting the matrix $(\tilde{\boldsymbol{D}}^T \tilde{\boldsymbol{D}} + \alpha \cdot \boldsymbol{I})$ as called for in the exact ridge regression solution, we can employ the stochastic gradient descent algorithm.

The gradient of $\tilde{\boldsymbol{w}}$ multiplied by 1/2 for convenience is:

$$\nabla_{\tilde{\boldsymbol{w}}} = \frac{\partial}{\partial \tilde{\boldsymbol{w}}} J(\tilde{\boldsymbol{w}}) = -\tilde{\boldsymbol{D}}^T Y + (\tilde{\boldsymbol{D}}^T \tilde{\boldsymbol{D}})\tilde{\boldsymbol{w}} + \alpha \cdot \tilde{\boldsymbol{w}}$$

Using (batch) gradient descent, we can iteratively compute $\tilde{\boldsymbol{w}}$ as follows

$$\tilde{\boldsymbol{w}}^{t+1} = \tilde{\boldsymbol{w}}^{t} - \eta \cdot \nabla_{\tilde{\boldsymbol{w}}} = (1 - \eta \cdot \alpha)\tilde{\boldsymbol{w}}^{t} + \eta \cdot \tilde{\boldsymbol{D}}^T (Y - \tilde{\boldsymbol{D}} \cdot \tilde{\boldsymbol{w}}^{t})$$

In SGD, we update the weight vector by considering only one (random) point at each time:

$$\tilde{\boldsymbol{w}}^{t+1} = \tilde{\boldsymbol{w}}^{t} - \eta \cdot \nabla_{\tilde{\boldsymbol{w}}}(\tilde{\boldsymbol{x}}_k) = \left(1 - \frac{\eta \cdot \alpha}{n}\right) \tilde{\boldsymbol{w}}^{t} + \eta \cdot (y_k - \tilde{\boldsymbol{x}}_k \cdot \tilde{\boldsymbol{w}}^{t}) \cdot \tilde{\boldsymbol{x}}_k$$

Ridge Regression: SGD Algorithm

```
Ridge Regression: SGD ($\mathbf{D}, Y, \eta, \epsilon$):
$\tilde{\mathbf{D}} \leftarrow \begin{pmatrix} 1 & \mathbf{D} \end{pmatrix}$ // augment data
$t \leftarrow 0$ // step/iteration counter
$\tilde{\mathbf{w}}^t \leftarrow$ random vector in $\mathbb{R}^{d+1}$ // initial weight vector
repeat
    foreach $k = 1, 2, \cdots, n$ (in random order) do
        $\nabla_{\tilde{\mathbf{w}}}(\tilde{\mathbf{x}}_k) \leftarrow -(y_k - \tilde{\mathbf{x}}_k^T \tilde{\mathbf{w}}^t) \cdot \tilde{\mathbf{x}}_k + \frac{\alpha}{n} \cdot \tilde{\mathbf{w}}$ // gradient at $\tilde{\mathbf{x}}_k$
        $\tilde{\mathbf{w}}^{t+1} \leftarrow \tilde{\mathbf{w}}^t - \eta \cdot \nabla_{\tilde{\mathbf{w}}}(\tilde{\mathbf{x}}_k)$ // update estimate for $w_k$
    $t \leftarrow t + 1$
until $\left\| \mathbf{w}^t - \mathbf{w}^{t-1} \right\| \leq \epsilon$
```

Ridge Regression: SGD

Example

We apply ridge regression on the Iris dataset ($n = 150$), using `petal length` (X) as the independent attribute, and `petal width` (Y) as the response variable.

Using SGD (with $\eta = 0.001$ and $\epsilon = 0.0001$) we obtain different lines of best fit for different values of the regularization constant α:

$$\begin{aligned}
\alpha = 0 : \widehat{Y} &= -0.366 + 0.413 \cdot X & SSE_{SGD} &= 6.37 & SSE_{Ridge} &= 6.34 \\
\alpha = 10 : \widehat{Y} &= -0.244 + 0.387 \cdot X & SSE_{SGD} &= 6.76 & SSE_{Ridge} &= 6.38 \\
\alpha = 100 : \widehat{Y} &= -0.022 + 0.327 \cdot X & SSE_{SGD} &= 10.04 & SSE_{Ridge} &= 8.87
\end{aligned}$$

Kernel Regression

Kernel generalizes linear regression to the non-linear case, i.e., finding a non-linear fit to the data to minimize the squared error, along with regularization.
$\phi(\boldsymbol{x}_i)$ maps the input point $\boldsymbol{x}_i$ to the feature space.

For regularized regression, we have to solve the following objective in feature space:

$$\boxed{\min_{\tilde{\boldsymbol{w}}} \; J(\tilde{\boldsymbol{w}}) = \left\| Y - \widehat{Y} \right\|^2 + \alpha \cdot \|\tilde{\boldsymbol{w}}\|^2 = \left\| Y - \tilde{\boldsymbol{D}}_\phi \tilde{\boldsymbol{w}} \right\|^2 + \alpha \cdot \|\tilde{\boldsymbol{w}}\|^2}$$

The optimal solution is therefore given as

$$\boxed{\boldsymbol{c} = (\tilde{\boldsymbol{K}} + \alpha \cdot \boldsymbol{I})^{-1} Y}$$

where $\boldsymbol{I} \in \mathbb{R}^{n \times n}$ is the $n \times n$ identity matrix, and $\tilde{\boldsymbol{D}}_\phi \tilde{\boldsymbol{D}}_\phi^T$ is the augmented kernel matrix $\tilde{\boldsymbol{K}}$.

Kernel Regression

The expression for the predicted response is:

$$\begin{aligned}\widehat{Y} &= \tilde{\boldsymbol{D}}_\phi \tilde{\boldsymbol{w}} \\ &= \tilde{\boldsymbol{D}}_\phi \tilde{\boldsymbol{D}}_\phi^T \boldsymbol{c} \\ &= \left(\tilde{\boldsymbol{D}}_\phi \tilde{\boldsymbol{D}}_\phi^T\right)\left(\tilde{\boldsymbol{K}} + \alpha \cdot \boldsymbol{I}\right)^{-1} Y \\ &= \tilde{\boldsymbol{K}}\left(\tilde{\boldsymbol{K}} + \alpha \cdot \boldsymbol{I}\right)^{-1} Y\end{aligned}$$

where $\tilde{\boldsymbol{K}}(\tilde{\boldsymbol{K}} + \alpha \cdot \boldsymbol{I})^{-1}$ is the *kernel hat matrix*.

$\alpha > 0$ ensures that the inverse always exists, which is another advantage of using (kernel) ridge regression, in addition to the regularization.

We compute the vector $\tilde{\boldsymbol{K}}_{\boldsymbol{z}}$ comprising the augmented kernel values of $\boldsymbol{z}$ with respect to all of the data points in $\boldsymbol{D}$, and take its dot product with the mixture coefficient vector $\boldsymbol{c}$ to obtain the predicted response.

Kernel Regression Algorithm

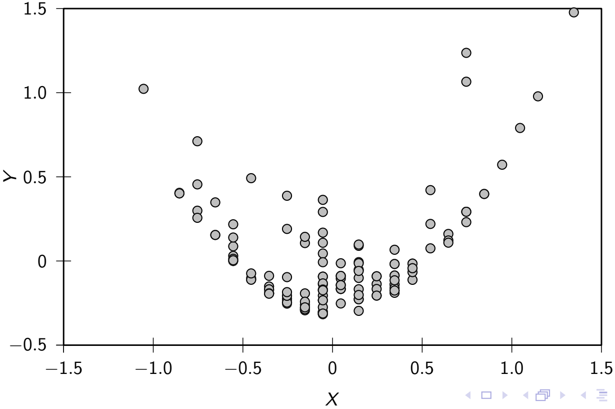
```
Kernel-Regression $(\boldsymbol{D}, Y, K, \alpha)$:
    $\boldsymbol{K} \leftarrow \left\{K(\boldsymbol{x}_i, \boldsymbol{x}_j)\right\}_{i,j=1,\dots,n}$ // standard kernel matrix
    $\tilde{\boldsymbol{K}} \leftarrow \boldsymbol{K} + 1$// augmented kernel matrix
    $\boldsymbol{c} \leftarrow \left(\tilde{\boldsymbol{K}} + \alpha \cdot \boldsymbol{I}\right)^{-1} Y$ // compute mixture coefficients
    $\widehat{Y} \leftarrow \tilde{\boldsymbol{K}} \boldsymbol{c}$
Testing $(\boldsymbol{z}, \boldsymbol{D}, K, \boldsymbol{c})$:
    $\tilde{\boldsymbol{K}}_{\boldsymbol{z}} \leftarrow \left\{1 + K(\boldsymbol{z}, \boldsymbol{x}_i)\right\}_{\forall\, \boldsymbol{x}_i \in \boldsymbol{D}}$
    $\hat{y} \leftarrow \boldsymbol{c}^T \tilde{\boldsymbol{K}}_{\boldsymbol{z}}$
```

Kernel Regression on Iris

Example

Consider the nonlinear Iris dataset obtained via a nonlinear transformation of `sepal length` (A_1) and `sepal width` (A_2) attributes (A_2):

$$X = A_2 \qquad\qquad Y = 0.2A_1^2 + A_2^2 + 0.1A_1A_2$$



Kernel Regression on Iris

Example

We treat Y as the response variable and X is the independent attribute. The points show a clear quadratic (nonlinear) relationship between the them.

The linear fit is

$$\widehat{Y} = 0.168 \cdot X$$

Using the quadratic (inhomogeneous) kernel over X comprising constant (1), linear (X), and quadratic terms (X^2), and $\alpha = 0.1$:

$$\widehat{Y} = -0.086 + 0.026 \cdot X + 0.922 \cdot X^2$$

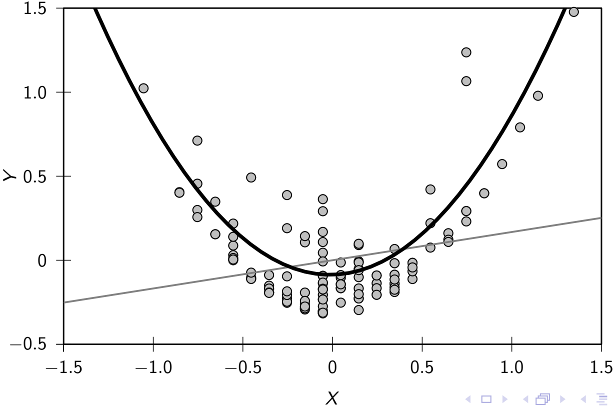
Kernel Regression on Iris

Example

The linear (in gray) and quadratic (in black) fit are shown.

The SSE error is 13.82 for the linear and 4.33 for the quadratic kernel.

The quadratic kernel (as expected) gives a much better fit to the data.

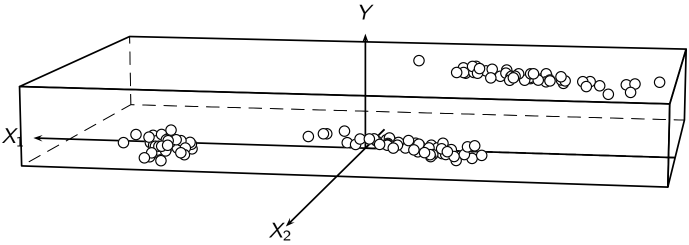


Kernel ridge regression

Example

Consider the Iris principal components dataset, where X_1 and X_2 denote the first two principal components.

The response variable Y is binary, with value 1 corresponding to `Iris-virginica` (points on the top right, with Y value 1) and 0 corresponding to `Iris-setosa` and `Iris-versicolor` (other two groups of points, with Y value 0).

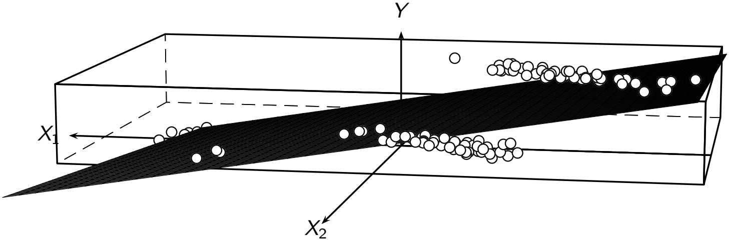


Kernel ridge regression

Example

Figure shows the fitted regression plane using a linear kernel with ridge value $\alpha = 0.01$:

$$\widehat{Y} = 0.333 - 0.167 \cdot X_1 + 0.074 \cdot X_2$$

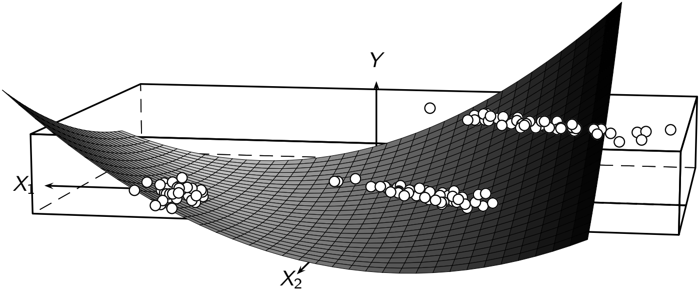


Kernel ridge regression

Example

Figure shows the fitted model when we use an inhomogeneous quadratic kernel with $\alpha = 0.01$:

$$\widehat{Y} = -0.03 - 0.167 \cdot X_1 - 0.186 \cdot X_2 + 0.092 \cdot X_1^2 + 0.1 \cdot X_1 \cdot X_2 + 0.029 \cdot X_2^2$$



The SSE error for the linear model is 15.47, whereas for the quadratic kernel it is 8.44, indicating a better fit for the training data.

L_1 Regression: Lasso

The *Lasso* (*least absolute selection and shrinkage operator*) is a regularization method that aims to sparsify the regression weights.

Lasso uses the L_1 norm for regularization:

$$\boxed{\min_{\boldsymbol{w}} \; J(\boldsymbol{w}) = \frac{1}{2} \cdot \|\overline{Y} - \overline{\boldsymbol{D}}\boldsymbol{w}\|^2 + \alpha \cdot \|\boldsymbol{w}\|_1}$$

where $\alpha \geq 0$ is the regularization constant and

$$\|\boldsymbol{w}\|_1 = \sum_{i=1}^{d} |w_i|$$

We assume that $X_1, X_2, \ldots, X_d$ and Y have been centered.

Centering relieves us from explicitly dealing with the bias term $b = w_0$, since we do not want to penalize b.

L_1 Regression: Lasso

The usage of the L_1 norm leads to *sparsity* in the solution vector.

Ridge regression reduces the value of the regression coefficients w_i, they may remain small but still non-zero.

L_1 regression can drive the coefficients to zero, resulting in a more interpretable model, especially when there are many predictor attributes.

The Lasso objective comprises two parts, the squared error term $\left\|\bar{\boldsymbol{Y}} - \bar{\boldsymbol{D}}\boldsymbol{w}\right\|^2$ which is convex and differentiable, and the L_1 penalty term

$$\alpha \cdot \|\boldsymbol{w}\|_1 = \alpha \sum_{i=1}^{d} |w_i|$$

which is convex but unfortunately non-differentiable at $w_i = 0$.

We cannot simply compute the gradient and set it to zero, as we did in the case of ridge regression.

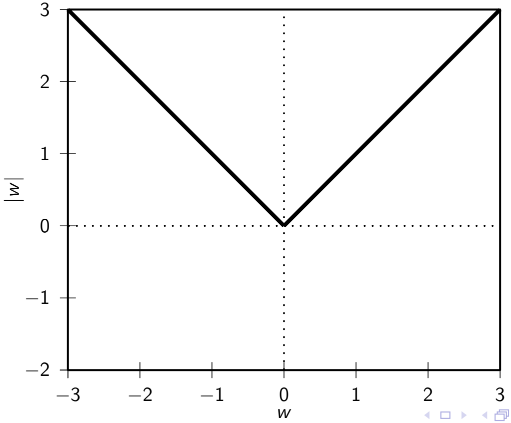
It can be solved via the generalized approach of *subgradients*.

L_1 Regression: Subgradients

Consider the absolute value function $f(w) = |w|$.

When $w > 0$, $f'(w) = +1$, and when $w < 0$, $f'(w) = -1$.

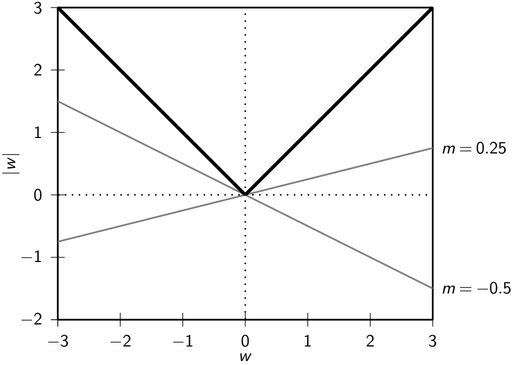
There is a discontinuity at $w = 0$ where the derivative does not exist.



L_1 Regression: Subgradients

Subgradients generalize the notion of a derivative.

For $f(w) = |w|$, the slope m of any line that passes through $w = 0$ that remains below or touches the graph of f is called a subgradient of f at $w = 0$.



Subgradients and Subdifferential

The set of all the subgradients at w is called the *subdifferential*, denoted as $\partial|w|$.

The subdifferential of $f(w) = |w|$ at $w = 0$ is given as $\partial|w| = [-1, 1]$.

Considering all the cases, the subdifferential for $f(w) = |w|$ is:

$$\partial|w| = \begin{cases} 1 & \text{iff } w > 0 \\ -1 & \text{iff } w < 0 \\ [-1, 1] & \text{iff } w = 0 \end{cases}$$

When the derivative exists, the subdifferential is unique and corresponds to the derivative (or gradient).

When the derivative does not exist the subdifferential corresponds to a set of subgradients.

Bivariate L_1 Regression

Consider the bivariate L_1 regression, where we have a single independent attribute $\overline{X}$ and a response attribute $\overline{Y}$ (both centered). The bivariate regression model is given as

$$\hat{y}_i = w \cdot \bar{x}_i$$

The Lasso objective can then be written as

$$\boxed{\min_w \; J(w) = \frac{1}{2}\sum_{i=1}^{n}(\bar{y}_i - w \cdot \bar{x}_i)^2 + \alpha \cdot |w|}$$

We can compute the subdifferential of this objective as follows:

$$\begin{aligned}
\partial J(w) &= \frac{1}{2} \cdot \sum_{i=1}^{n} 2 \cdot (\bar{y}_i - w \cdot \bar{x}_i) \cdot (-\bar{x}_i) + \alpha \cdot \partial|w| \\
&= -\sum_{i=1}^{n} \bar{x}_i \cdot \bar{y}_i + w \cdot \sum_{i=1}^{n} \bar{x}_i^2 + \alpha \cdot \partial|w| \\
&= -\overline{X}^T \overline{Y} + w \cdot \left\|\overline{X}\right\|^2 + \alpha \cdot \partial|w|
\end{aligned}$$

Bivariate L_1 Regression

Corresponding to the three cases for the subdifferential of the absolute value function we have three cases to consider:

Case I ($w > 0$ and $\partial|w| = 1$): $w = \eta \cdot \bar{X}^T\bar{Y} - \eta \cdot \alpha$

Since $w > 0$, $\eta \cdot \bar{X}^T\bar{Y} > \eta \cdot \alpha$ or $|\eta \cdot \bar{X}^T\bar{Y}| > \eta \cdot \alpha$.

Case II ($w < 0$ and $\partial|w| = -1$): $w = \eta \cdot \bar{X}^T\bar{Y} + \eta \cdot \alpha$

Since $w < 0$, $\eta \cdot \bar{X}^T\bar{Y} < -\eta \cdot \alpha$ or $|\eta \cdot \bar{X}^T\bar{Y}| > \eta \cdot \alpha$.

Case III ($w = 0$ and $\partial|w| \in [-1, 1]$): $w \in \left[\eta \cdot \bar{X}^T\bar{Y} - \eta \cdot \alpha, \eta \cdot \bar{X}^T\bar{Y} + \eta \cdot \alpha\right]$

However, since $w = 0$, $|\eta \cdot \bar{X}^T\bar{Y}| \leq \eta \cdot \alpha$.

Then the above three cases can be written compactly as:

$$\boxed{w = \mathcal{S}_{\eta \cdot \alpha}(\eta \cdot \bar{X}^T\bar{Y})}$$

with $\tau = \eta \cdot \alpha$, where w is the optimal solution to the problem.

L_1-Regression Algorithm

```
$L_1$-Regression $(\boldsymbol{D}, Y, \alpha, \eta, \epsilon)$:
$\boldsymbol{\mu} \leftarrow \text{mean}(\boldsymbol{D})$ // compute mean
$\overline{\boldsymbol{D}} \leftarrow \boldsymbol{D} - \mathbf{1} \cdot \boldsymbol{\mu}^T$ // center the data
$\overline{Y} \leftarrow Y - \mu_Y \cdot \mathbf{1}$ // center the response
$t \leftarrow 0$ // step/iteration counter
$\boldsymbol{w}^t \leftarrow$ random vector in $\mathbb{R}^d$ // initial weight vector
repeat
    foreach $k = 1, 2, \cdots, d$ do
        $\nabla(w_k^t) \leftarrow -\overline{X}_k^T (Y - \overline{\boldsymbol{D}} \boldsymbol{w}^t)$ // compute gradient at $w_k$
        $w_k^{t+1} \leftarrow w_k^t - \eta \cdot \nabla(w_k^t)$ // update estimate for $w_k$
        $w_k^{t+1} \leftarrow \mathcal{S}_{\eta \cdot \alpha}\left(w_k^{t+1}\right)$ // apply soft-threshold function
    $t \leftarrow t + 1$
until $\left\| \boldsymbol{w}^t - \boldsymbol{w}^{t-1} \right\| \leq \epsilon$
$b \leftarrow \mu_Y - \left(\boldsymbol{w}^t\right)^T \boldsymbol{\mu}$ // compute the bias term
```

L_1 Regression

Example

We apply L_1 regression to the full Iris dataset with $n = 150$ points, and four independent attributes, namely `sepal-width` (X_1), `sepal-length` (X_2), `petal-width` (X_3), and `petal-length` (X_4).

The Iris type attribute comprises the response variable Y. There are three Iris types, namely `Iris-setosa`, `Iris-versicolor`, and `Iris-virginica`, which are coded as 0, 1 and 2, respectively.

The L_1 regression for α ($\eta = 0.0001$) are shown below:

$$\begin{aligned}
\alpha = 0: \ \widehat{Y} &= +0.19 - 0.11 \cdot X_1 - 0.05 \cdot X_2 + 0.23 \cdot X_3 + 0.61 \cdot X_4 & SSE &= 6.96 & \|\boldsymbol{w}\|_1 &= 0.44\\
\alpha = 1: \ \widehat{Y} &= -0.08 - 0.08 \cdot X_1 - 0.02 \cdot X_2 + 0.25 \cdot X_3 + 0.52 \cdot X_4 & SSE &= 7.09 & \|\boldsymbol{w}\|_1 &= 0.34\\
\alpha = 5: \ \widehat{Y} &= -0.55 + 0.00 \cdot X_1 + 0.00 \cdot X_2 + 0.36 \cdot X_3 + 0.17 \cdot X_4 & SSE &= 8.82 & \|\boldsymbol{w}\|_1 &= 0.16\\
\alpha = 10: \ \widehat{Y} &= -0.58 + 0.00 \cdot X_1 + 0.00 \cdot X_2 + 0.42 \cdot X_3 + 0.00 \cdot X_4 & SSE &= 10.15 & \|\boldsymbol{w}\|_1 &= 0.18
\end{aligned}$$

Note the sparsity inducing effect, for $\alpha = 5$ and $\alpha = 10$, which drives some w_i to 0.

L_1 Regression

Example

We can contrast the coefficients for L_2 (ridge) and L_1 (Lasso) regression by comparing models with the same level of squared error.

For $\alpha = 5$, the L_1 model has $SSE = 8.82$.

We adjust the ridge value in L_2 regression, with $\alpha = 35$ resulting in a similar SSE value. The two models are given as follows:

$$
\begin{aligned}
L_1: \ \widehat{Y} &= -0.553 + 0.0 \cdot X_1 + 0.0 \cdot X_2 + 0.359 \cdot X_3 + 0.170 \cdot X_4 & \|\mathbf{w}\|_1 = 0.156 \\
L_2: \ \widehat{Y} &= -0.394 + 0.019 \cdot X_1 - 0.051 \cdot X_2 + 0.316 \cdot X_3 + 0.212 \cdot X_4 & \|\mathbf{w}\|_1 = 0.598
\end{aligned}
$$

L_2: the coefficients for X_1 and X_2 are small, and therefore less important, but they are not zero.

L_1: the coefficients for X_1 and X_2 are exactly zero, leaving only X_3 and X_4;

Lasso can thus act as an automatic feature selection approach.

Data Mining and Machine Learning: Fundamental Concepts and Algorithms

datamining book.info

Mohammed J. Zaki[1] Wagner Meira Jr.[2]

[1]Department of Computer Science
Rensselaer Polytechnic Institute, Troy, NY, USA

[2]Department of Computer Science
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Chapter 23: Linear Regression