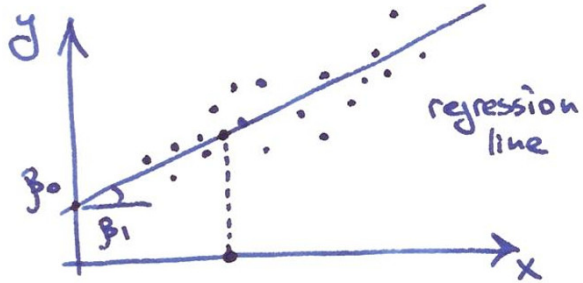
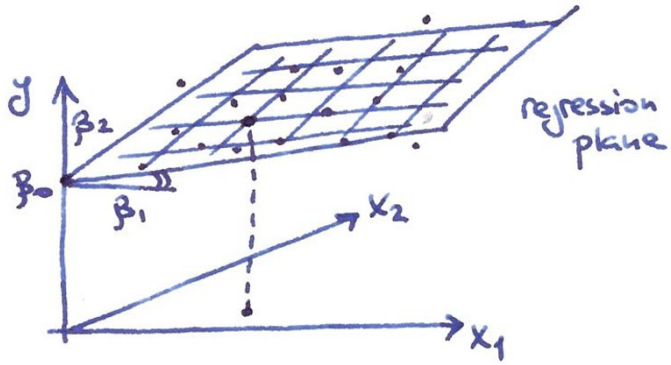
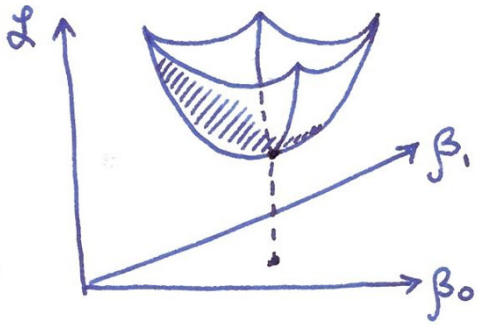
Simple vs. multiple linear regression

Multiple linear regression has >1 predictor.







Multiple linear regression

The model:

$$f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p.$$

It is convenient to define $x_0 \equiv 1$. Then:

$$f(x) = \vec{\beta} \cdot \vec{x} = \boldsymbol{\beta}^\top \mathbf{x} = \begin{pmatrix} \beta_0 & \cdots & \beta_p \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_p \end{pmatrix}$$

The loss and the gradient

Using this notation, the mean-squared-error loss function becomes:

$$\mathcal{L}(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \boldsymbol{\beta}^\top \mathbf{x}^{(i)}\right)^2.$$

Partial derivatives:

$$\frac{\partial \mathcal{L}}{\partial \beta_k} = -\frac{2}{n}\sum_{i=1}^{n}\left(y^{(i)} - \boldsymbol{\beta}^\top \mathbf{x}^{(i)}\right)x_k^{(i)}.$$

Gradient:

$$\nabla \mathcal{L} = -\frac{2}{n}\sum_{i=1}^{n}\left(y^{(i)} - \boldsymbol{\beta}^\top \mathbf{x}^{(i)}\right)\mathbf{x}^{(i)}.$$

Introducing *design matrix*

Let us collect all vectors $\mathbf{x}^{(i)}$ into one matrix of size $n \times (p+1)$:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \vdots \\ \mathbf{x}^{(n)} \end{pmatrix} = \begin{pmatrix} x_0^{(1)} & x_1^{(1)} & \cdots & x_p^{(1)} \\ x_0^{(2)} & x_1^{(2)} & \cdots & x_p^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_0^{(n)} & x_1^{(n)} & \cdots & x_p^{(n)} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \cdots & \mathbf{x}_p \end{pmatrix}.$$

Let us also collect all y values into a *response vector*:

$$\mathbf{y} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{pmatrix}.$$

Matrix multiplication is useful!

Given $\mathbf{X}$ and $\boldsymbol{\beta}$, how to compute predicted values $\hat{\mathbf{y}}$?

$$\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta} = \begin{pmatrix} x_0^{(1)} & x_1^{(1)} & \cdots & x_p^{(1)} \\ x_0^{(2)} & x_1^{(2)} & \cdots & x_p^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_0^{(n)} & x_1^{(n)} & \cdots & x_p^{(n)} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} = \begin{pmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \\ \vdots \\ \hat{y}^{(n)} \end{pmatrix}.$$

Matrix calculus is useful!

Now we can write:

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \boldsymbol{\beta}^\top \mathbf{x}^{(i)}\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left([\mathbf{y}]_i - [\mathbf{X}\boldsymbol{\beta}]_i\right)^2 = \frac{1}{n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2.$$

Another way to write this (sometimes useful):

$$\frac{1}{n}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}).$$

Gradient:

$$\nabla\mathcal{L} = -\frac{2}{n}\sum_{i=1}^{n}(y^{(i)} - \boldsymbol{\beta}^\top \mathbf{x}^{(i)})\mathbf{x}^{(i)} = -\frac{2}{n}\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}).$$

Matrix algebra is useful!

Gradient:

$$\nabla \mathcal{L} = -\frac{2}{n}\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}).$$

Setting it to zero to derive the analytical solution:

$$-\frac{2}{n}\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = \mathbf{0}$$
$$\mathbf{X}^\top\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}^\top\mathbf{y}$$
$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$$

Baby linear regression: $\hat{\beta} = \sum x_i y_i / \sum x_i^2$.

Multiple linear regression: $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$.

The role of $(\mathbf{X}^\top \mathbf{X})^{-1}$

If $(\mathbf{X}^\top \mathbf{X})^{-1} = \mathbf{I}$, then $\hat{\boldsymbol{\beta}} = \mathbf{X}^\top \mathbf{y}$. When does this hold?

The element ij of this matrix is given by $\mathbf{x}_i^\top \mathbf{x}_j = \sum_{k=1}^{n} x_i^{(k)} x_j^{(k)}$. So in this case the features are *orthonormal* (*orthogonal* and have norm 1).

If all features have mean zero, are uncorrelated, and have variance equal to 1, then $\hat{\beta}_i = \mathbf{x}_i^\top \mathbf{y}$ (for $i \neq 0$), i.e. regression coefficient for each predictor can be computed independently (by p separate regressions).

Exercise: if all features have mean zero, then $\hat{\beta}_0 = \frac{1}{n} \sum_i y_i$. Note that *centering* features does not affect the model:

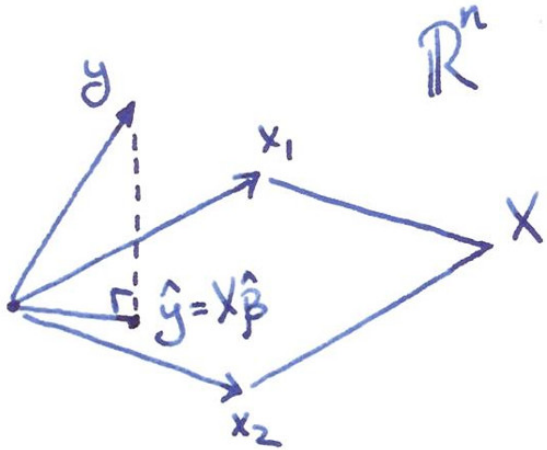
$$f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p.$$

Predicted values and the *hat matrix*

It is useful to write down the formula for $\hat{\mathbf{y}}$:

$$\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}.$$

It can be understood as the orthogonal projection matrix in the n-dimensional space!



$\|y - X\hat{\beta}\|^2$ minimal $\Rightarrow \perp$

$X^T(y - X\hat{\beta}) = 0$ $\Rightarrow \perp$

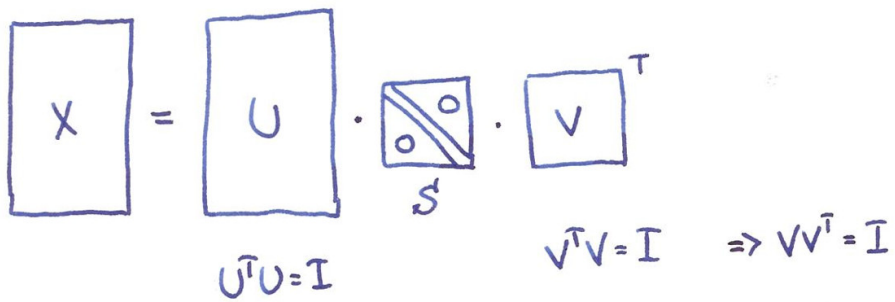
Singular value decomposition (SVD)

When $\mathbf{X}$ has orthonormal columns, i.e. $\mathbf{X}^\top\mathbf{X} = \mathbf{I}$, then everything simplifies a lot. What if it does not? We can transform it such that it does!

Non-trivial fact: any matrix $\mathbf{X}$ can be written as

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^\top,$$

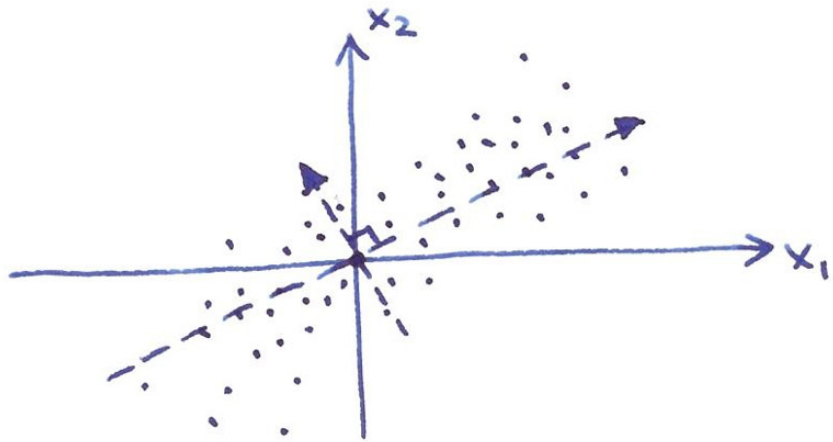
where $\mathbf{U}$ and $\mathbf{V}$ have orthonormal columns (left/right *singular vectors*) and $\mathbf{S}$ is diagonal (with *singular values*).



Geometry of SVD

Let us assume here that all the features and the response are centered (and $\mathbf{X}$ does not contain the column of 1s).

Consider SVD of $\mathbf{X} = \mathbf{USV}^\top$.



SVD is useful!

$$
\begin{aligned}
\hat{\mathbf{y}} = \mathbf{H}\mathbf{y} &= \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} = \\
&= \mathbf{U}\mathbf{S}\mathbf{V}^\top(\mathbf{V}\mathbf{S}\mathbf{U}^\top\mathbf{U}\mathbf{S}\mathbf{V}^\top)^{-1}\mathbf{V}\mathbf{S}\mathbf{U}^\top\mathbf{y} = \\
&= \mathbf{U}\mathbf{S}\mathbf{V}^\top(\mathbf{V}\mathbf{S}^2\mathbf{V}^\top)^{-1}\mathbf{V}\mathbf{S}\mathbf{U}^\top\mathbf{y} = \\
&= \mathbf{U}\mathbf{S}\mathbf{V}^\top\mathbf{V}\mathbf{S}^{-2}\mathbf{V}^\top\mathbf{V}\mathbf{S}\mathbf{U}^\top\mathbf{y} = \\
&= \mathbf{U}\mathbf{U}^\top\mathbf{y}
\end{aligned}
$$

SVD is useful!

$$\hat{\mathbf{y}} = \mathbf{H}\mathbf{y} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} = \mathbf{U}\mathbf{U}^\top\mathbf{y}$$

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^\top\mathbf{y}$$

Note: the formula $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$ is good for mathematical analysis but terrible for computations. Never program your linear regression solver like that :)

The lesson: strong correlations between features $\Rightarrow$ small singular values in $\mathbf{X}^\top\mathbf{X} \Rightarrow$ possible numerical problems, estimated coefficients blowing up, high estimation uncertainty.