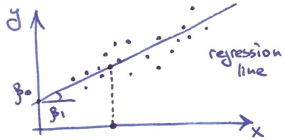
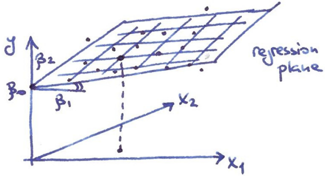
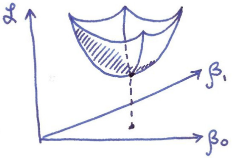
Multiple Linear Regression

Simple vs. multiple linear regression

Multiple linear regression has >1 predictor.







EBERHARD KARLS UNIVERSITÄT TÜBINGEN

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:

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

Another way to write this (sometimes useful):

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

Gradient:

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

Matrix algebra is useful!

Gradient:

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

Setting it to zero to derive the analytical solution:

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

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

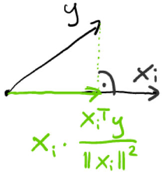
Multiple linear regression: $\hat{\beta} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$.

Prediction $\hat{\mathbf{y}}$ is orthogonal projection

The prediction $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \sum_i \mathbf{x}_i \hat{\beta}_i$ lies in the subspace of $\mathbb{R}^n$ spanned by the $\mathbf{x}_i$'s. To minimize the loss we need the point of that subspace that is closest to $\mathbf{y}$, its orthogonal projection.

The projection of $\mathbf{y}$ on any of the $\mathbf{x}_i$ is given by

$$\frac{\mathbf{x}_i(\mathbf{x}_i^\top \mathbf{y})}{\|\mathbf{x}_i\|^2}.$$

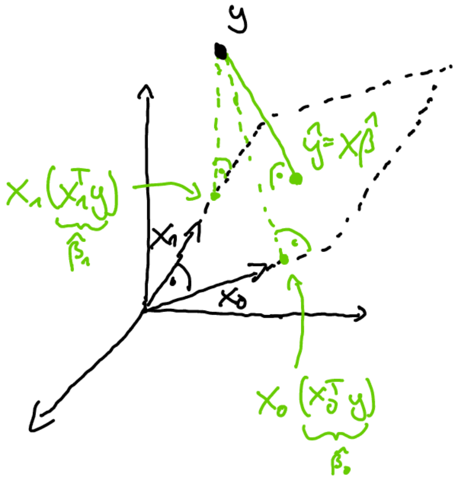


If all features $\mathbf{x}_i$ are orthogonal and have norm 1 (aka are *orthonormal*), we can just add the individual projections

$$\hat{\mathbf{y}} = \sum_{i=0}^{p} \mathbf{x}_i(\mathbf{x}_i^\top \mathbf{y}) = \mathbf{X}\mathbf{X}^\top \mathbf{y} \qquad \hat{\boldsymbol{\beta}} = \mathbf{X}^\top \mathbf{y}, \text{ i.e. } \forall i \;\; \hat{\beta}_i = \mathbf{x}_i^\top \mathbf{y},$$

and we can compute the regression coefficients independently.

Prediction $\hat{\mathbf{y}}$ is orthogonal projection



Orthogonal projection for two orthonormal features $\mathbf{x}_0$ and $\mathbf{x}_1$.

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

The features $\mathbf{x}_i$ are orthonormal if and only if $\mathbf{X}^\top\mathbf{X} = \mathbf{I}$. Otherwise, individual projections are off by $\|\mathbf{x}_i\|^2$ and we exaggerate shared directions of feature vectors.

To correct, we need to "divide by" $\mathbf{X}^\top\mathbf{X}$, making the general projection matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$, called the *hat matrix* due to $\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}$.

Indeed, now the error vector is orthogonal to each $\mathbf{x}_i$:

$$\begin{aligned}
\mathbf{X}^\top(\mathbf{y} - \hat{\mathbf{y}}) &= \mathbf{X}^\top(\mathbf{y} - \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}) \\
&= (\mathbf{X}^\top - \mathbf{X}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top)\mathbf{y} \\
&= (\mathbf{X}^\top - \mathbf{X}^\top)\mathbf{y} = 0 \qquad (1)
\end{aligned}$$

Interpretation of orthonormal features

Exercise: If $\mathbf{x}_0$ is proportional to $(1, \ldots, 1)^\top$, then the features $\mathbf{x}_i$, $i \geq 0$, are orthonormal if and only if

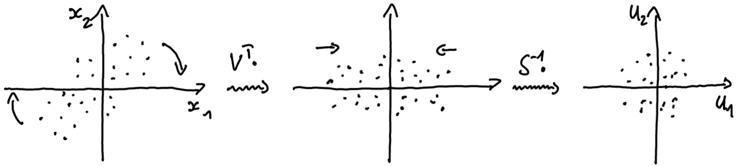
- $\mathbf{x}_i$ is centered for all $i \geq 1$
- $\mathbf{x}_i$ and $\mathbf{x}_j$ are uncorrelated for all $i \neq j \geq 1$
- $\mathbf{x}_i$ has norm 1 for all $i \geq 0$

But what if they are not? We can transform them such that they are!

Singular value decomposition (SVD)

Consider the rows $\mathbf{x}^{(i)}$ of $\mathbf{X}$ as point cloud in $\mathbb{R}^p$ (omitting $\mathbf{x}_0$ and assuming centered features as well as full rank for $\mathbf{X}$).

Feature correlation means a (noisy) linear relationship in the point cloud. The feature norm becomes the variance of the cloud in that direction.



$\mathbf{x}^{(i)} = \begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \end{pmatrix}$	$\mathbf{V}^\top \mathbf{x}^{(i)}$	$\mathbf{S}^{-1}\mathbf{V}^\top \mathbf{x}^{(i)}$
$\mathbf{X}$	$\mathbf{XV}$	$\mathbf{U} \coloneqq \mathbf{XVS}^{-1}$

The columns of $\mathbf{U}$ are the transformed features.

Centering features*

Centering the features $\mathbf{x}_i$ does not change the model, as it can be absorbed in the intercept coefficient:

$$
\begin{aligned}
f(\mathbf{x}) &= \beta_0 + \sum_{i=1}^{p} \beta_i x_i \\
&= \beta_0 + \sum_{i=1}^{p} \beta_i (x_i - \overline{\mathbf{x}_i} + \overline{\mathbf{x}_i}) \\
&= \left(\beta_0 + \sum_{i=1}^{p} \beta_i \overline{\mathbf{x}_i}\right) + \sum_{i=1}^{p} \beta_i (x_i - \overline{\mathbf{x}_i}).
\end{aligned} \tag{2}
$$

SVD formally

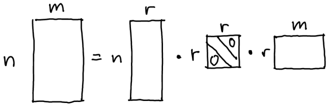
Non-trivial fact:

For any (not necessarily square) matrix $\mathbf{X}$ of shape $n \times m$ and rank r, there exist matrices

- $\mathbf{U}$ of shape $n \times r$ (*left singular vectors*)
- $\mathbf{S}$ of shape $r \times r$ (*singular values*)
- $\mathbf{V}$ of shape $m \times r$ (*right singular vectors*)

such that $\mathbf{U}$ and $\mathbf{V}$ have orthogonal columns ($\mathbf{U}^\top\mathbf{U} = \mathbf{I}$, $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$), $\mathbf{S}$ is diagonal with positive entries on the diagonal, and

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



Solution in transformed features

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

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 :)

Dependent features*

$\mathbf{X}^\top\mathbf{X}$ has shape $(p+1)\times(p+1)$ and the same rank as $\mathbf{X}$. So the inverse of $\mathbf{X}^\top\mathbf{X}$ exists exactly if the features $\mathbf{x}_i$'s are linearly independent.

Otherwise, we still have the formulae

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

yet $\hat{\boldsymbol{\beta}}$ is not unique; adding any vector in $\ker(\mathbf{X})$ yields a valid solution.

Indeed, the orthonormal columns of $\mathbf{U}$ span the same subspace as the columns of $\mathbf{X}$ and $\mathbf{U}\mathbf{U}^\top$ is the projection to that subspace.

Effect of correlated features

Perfect correlation reduces the rank of the design matrix as the two correlated features are linearly dependent.

High correlation leads to small singular values, i.e., $\mathbf{X}^\top\mathbf{X}$ is “nearly not invertible”. This leads to

- numerical problems
- huge regression coefficients
- high uncertainty