Chapter 1

Ordinary least squares regression

Ordinary least squares is a classical regression technique that dates back to Carl Friedrich Gauß who might have used it in 1801 to rediscover the dwarf planet Ceres that had previously been discovered by the Italian monk Guiseppe Piazzi who was able to track it for 41 days before it got lost in the halo of the sun.

Data generation model

We assume that

$$Y = \theta^\top X + \varepsilon,$$

where

1. $\mathcal{Y} = \mathbb{R}$ and $\mathcal{X} = \{1\} \times \mathbb{R}^n$,
2. Y is a linear function of X with coefficients $\theta \in \mathbb{R}^{n+1}$, and
3. the random noise term ϵ is normally distributed with mean 0 and variance σ^2, i.e., $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. Note that here the noise is not allowed to vary with the covariates X and also not with different instantiations of the random experiment. This assumption is also known as *homoskedasticity*.

That is, we assume that Y is a linear function of X with added stochastic Gaussian noise. This implies that also $Y|X = x$ is a random variable with distribution

$$Y|X = x \sim \mathcal{N}(\theta^\top x, \sigma^2),$$

which means

$$p_x(y;\theta) \;=\; p(y|X=x,\theta) \;=\; \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y-\theta^\top x)^2}{2\sigma^2}\right).$$

Maximum likelihood estimate

The whole data generation model, i.e., the family of densities p_x, is completely specified by $\sigma > 0$ and $\theta \in \mathbb{R}^{n+1}$. Furthermore, the predictor derived from the model (that is only indirectly accessible for us)

$$h : \mathcal{X} \to \mathcal{Y},\; x \mapsto \mathsf{argmax}_{y\in\mathcal{Y}}\; p_x(y) \;=\; \theta^\top x$$

does not depend on σ. Hence, the problem of learning this predictor reduces to estimating θ from the i.i.d. data

$$(x^{(1)},y^{(1)}),\ldots,(x^{(m)},y^{(m)}).$$

A natural estimator for θ is the parameter vector $\hat{\theta}$ that maximizes the probability of the data. This estimator is called the *maximum likelihood estimator* and can be derived as a solution to the following optimization problem

$$\hat{\theta} \;=\; \mathsf{argmax}_{\theta\in\mathbb{R}^{n+1}}\; L(\theta),$$

where

$$L(\theta) \;=\; \prod_{i=1}^{m} p(y^{(i)}|X=x^{(i)},\theta) \;=\; \prod_{i=1}^{m}\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y^{(i)}-\theta^\top x^{(i)})^2}{2\sigma^2}\right)$$

is the likelihood function for the parameter vector θ. The product form of the likelihood function is due to the i.i.d. assumption for the data. An optimum of the likelihood function remains optimal if we apply a monotonically increasing transformation to the likelihood function. Since the likelihood function is the product of exponentials a natural choice for such a transformation is the logarithm. Applying the logarithm to the likelihood function $L(\theta)$ gives us the log-likelihood function $\ell(\theta)$ that reads as

$$\begin{aligned}\ell(\theta) \;=\; \log L(\theta) \;&=\; -\sum_{i=1}^{m}\left(\log(\sqrt{2\pi}\sigma)+\frac{(y^{(i)}-\theta^\top x^{(i)})^2}{2\sigma^2}\right)\\ &=\; -m\log(\sqrt{2\pi}\sigma)-\frac{1}{2\sigma^2}\sum_{i=1}^{m}\left(y^{(i)}-\theta^\top x^{(i)}\right)^2.\end{aligned}$$

Using

$$y = (y^{(1)}, \ldots, y^{(m)})^\top \in \mathbb{R}^m, \quad \theta = (\theta_0, \ldots, \theta_n)^\top \in \mathbb{R}^{n+1}$$

and

$$X = (x^{(1)}, \ldots, x^{(m)}) = \begin{pmatrix} 1 & \ldots & 1 \\ x_1^{(1)} & \ldots & x_1^{(m)} \\ \vdots & \ddots & \vdots \\ x_n^{(1)} & \ldots & x_n^{(m)} \end{pmatrix} \in \mathbb{R}^{(n+1)\times m}$$

we can write $\ell(\theta)$ more compactly as

$$\ell(\theta) = -m \log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}\|y - X^\top\theta\|^2.$$

Hence, we have

$$\begin{aligned}
\hat{\theta} &= \mathsf{argmax}_{\theta\in\mathbb{R}^{n+1}}\ L(\theta) \\
&= \mathsf{argmax}_{\theta\in\mathbb{R}^{n+1}}\ \ell(\theta) \\
&= \mathsf{argmax}_{\theta\in\mathbb{R}^{n+1}}\ -m \log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}\|y - X^\top\theta\|^2 \\
&= \mathsf{argmax}_{\theta\in\mathbb{R}^{n+1}}\ -\frac{1}{2}\|y - X^\top\theta\|^2 \\
&= \mathsf{argmin}_{\theta\in\mathbb{R}^{n+1}}\ \frac{1}{2}\|y - X^\top\theta\|^2.
\end{aligned}$$

A necessary condition for a minimum is the vanishing gradient. We compute

$$\begin{aligned}
\nabla_\theta \frac{1}{2}\|y - X^\top\theta\|^2 &= \nabla_\theta \frac{1}{2}\left(y - X^\top\theta\right)^\top\left(y - X^\top\theta\right) \\
&= \nabla_\theta \frac{1}{2}\left(y^\top y - y^\top X^\top\theta - \theta^\top X y + \theta^\top X X^\top\theta\right) \\
&= \nabla_\theta \frac{1}{2}\left(y^\top y - 2\theta^\top X y + \theta^\top X X^\top\theta\right) \\
&= -Xy + XX^\top\theta,
\end{aligned}$$

and get from the vanishing gradient condition that

$$XX^\top\hat{\theta} = Xy.$$

That is, the maximum likelihood estimate of θ is the solution of a linear system, where $XX^\top$ is the *second moment matrix*. If the system is underdetermined, i.e., if the rank of the second moment matrix is less than $n+1$, then its solution space

is an affine subspace of $\mathbb{R}^{n+1}$, but if the second moment matrix is invertible, i.e., if it has full rank, then the unique solution to the maximum likelihood problem is given as

$$\hat{\theta} = (XX^\top)^{-1}Xy.$$

Remark: argmax ... and argmin ... do not need to be unique in general. When we write something like $\hat{\theta} =$ argmax ..., then we assign an arbitrary value in argmax ... to $\hat{\theta}$ unless stated otherwise.

Data preparation

For allowing linear function that do not need to pass through the origin, we have artificially padded the data points $x^{(i)}, \in [m]$ by an additional component with value 1. Instead we can also introduce the offset term θ_0 explicitly. Let $X \in \mathbb{R}^{n\times m}$ be the data matrix without padding. Then we can write the OLS problem as

$$\hat{\theta} = \text{argmin}_{\theta\in\mathbb{R}^{n+1}} \frac{1}{2}\|y - \theta_0\mathbf{1}_m - X^\top\theta\|^2,$$

where $\mathbf{1}_m \in \mathbb{R}^m$ is the vector whose entries are all 1. The necessary condition for the optimal value of θ_0 is the vanishing derivative (of a convex, quadratic function)

$$\begin{aligned}\frac{d}{d\theta_0}\frac{1}{2}\|y - \theta_0\mathbf{1}_m - X^\top\theta\|^2 &= \frac{d}{d\theta_0}\frac{1}{2}\left(\theta_0^2\mathbf{1}_m^\top\mathbf{1}_m - 2\theta_0\mathbf{1}_m^\top y + 2\theta_0\mathbf{1}_m^\top X^\top\theta\right)\\ &= m\theta_0 - \mathbf{1}_m^\top y + \mathbf{1}_m^\top X^\top\theta,\end{aligned}$$

which gives

$$\theta_0 = \frac{1}{m}\left(\mathbf{1}_m^\top y - \mathbf{1}_m^\top X^\top\theta\right).$$

The entries of the vector $\mathbf{1}_m^\top X^\top \in \mathbb{R}^m$ are given as

$$\mathbf{1}_m^\top X^\top = \left(\sum_{i=1}^m x_1^{(i)}, \ldots, \sum_{i=1}^m x_n^{(i)}\right).$$

This vector becomes the zero vector, if we center the data points, i.e., if we replace $x^{(i)}$ by

$$x^{(i)} - \frac{1}{m}\sum_{j=1}^m x^{(j)}, \quad i \in [m].$$

Hence, after centering, we get

$$\theta_0 = \frac{\mathbf{1}_m^\top y}{m} = \frac{1}{m}\sum_{i=1}^{m} y_i.$$

Now, if we also center the label vector y, then we have $\theta_0 = 0$ and we do no longer need the offset. In the following we assume that the data and label vectors are centered.
It is important to note though that for predictions at $x \in \mathbb{R}^n$ one has to subtract the sample mean also from x, and later one has to add the offset, i.e., the average of the observed label vector, back to the prediction.

An additional data transformation, that becomes important in the next paragraph, is making the scales on which the explanatory variables are measured comparable. This is typically done by standardizing, i.e., scaling the features such that all entries on the diagonal of the second moment matrix become 1, that is, the centered data points $x^{(i)}$ are replaced by

$$\left(\frac{x_1^{(i)}}{\sqrt{\sum_{j=1}^m {x_1^{(j)}}^2}}, \ldots, \frac{x_n^{(i)}}{\sqrt{\sum_{j=1}^m {x_n^{(j)}}^2}}\right)^\top.$$

Alternatively, one can just scale the features simply by replacing $x_j^{(i)}$, $i \in [m], j \in [n]$ with

$$\frac{x_j^{(i)}}{\max_{k=1,\ldots,m}\left\{x_j^{(k)}\right\} - \min_{k=1,\ldots,m}\left\{x_j^{(k)}\right\}}.$$

When computing a prediction at $x \in \mathbb{R}^n$, then x should be transformed in the same way, for instance, when using the first transformation, the sample mean is subtracted from x before the components are each scaled by the inverse of the square root of the corresponding sample variance.

Ridge regression

The second moment matrix $XX^\top$ can be written as

$$XX^\top = \sum_{i=1}^{m} x^{(i)} {x^{(i)}}^\top,$$

where $x^{(i)}{x^{(i)}}^\top$ is a projection matrix that projects any point $x \in \mathbb{R}^n$ onto the one-dimensional subspace spanned by $x^{(i)} \in \mathbb{R}^n$, i.e.,

$$\mathbb{R}^n \to \mathbb{R}^n,\ x \mapsto \left(x^{(i)}{x^{(i)}}^\top\right)x = \left({x^{(i)}}^\top x\right)x^{(i)}.$$

Thus, the matrix $x^{(i)}{x^{(i)}}^\top$ has rank one, and $XX^\top$ has rank at most m. That is, if m is small compared to n, then the second moment matrix does not have full rank and is thus not invertible.
By construction, the second moment matrix is symmetric and positive semi-definite, i.e., it holds for all $x \in \mathbb{R}^n$ that $\left(XX^\top\right)^\top = XX^\top$ and $x^\top XX^\top x \geq 0$. Note that the second moment matrix is the Hessian, i.e., the matrix of second order derivatives of the function $\frac{1}{2}\|y - X^\top\theta\|^2$. Hence, the Hessian of this function is positive semi-definite, which means that the function itself is convex and thus, $\hat{\theta} = (XX^\top)^{-1}Xy$ is indeed a minimum.
Another consequence of positive semi-definiteness is that for any $c > 0$ the matrix $XX^\top + c\mathbb{1}_n$ is positive definite and thus invertible. By replacing the second moment matrix $XX^\top$ by $XX^\top + c\mathbb{1}_n$, we get the following estimate for the parameter vector θ,

$$\hat{\theta} = (XX^\top + c\mathbb{1}_n)^{-1}Xy.$$

This estimate is the solution of the regularized maximum likelihood problem

$$\min_{\theta\in\mathbb{R}^n} \frac{1}{2}\|y - X^\top\theta\|^2 + \frac{c}{2}\|\theta\|^2$$

that is known as *ridge regression*. The ridge regression problem is a strictly convex optimization problem that has a unique solution even if the second moment matrix does not have full rank.

Remarks: The regularization term $\frac{c}{2}\|\theta\|^2$ treats all the explanatory variables the same. Thus it is important that these variables are measured on comparable scales. Here, we took care of that in the data preparation phase where we rescaled the explanatory variables by their sample variances.
Turning to the regularized problem makes sense even when the matrix $XX^\top$ is invertible. The *condition number* of the matrix $XX^\top$, defined as the quotient of the largest eigenvalue $\lambda_{\max}$ divided by the smallest eigenvalue $\lambda_{\min}$ of the matrix, is a measure of the range of scales that is covered by the data. Since floating point number approximations can adapt well only to small ranges of scales, we can run into numerical problems when attempting to compute $\left(XX^\top\right)^{-1}$, if the range of scales is large. After regularization the condition number of the matrix changes to

$$\frac{\lambda_{\max} + c}{\lambda_{\min} + c}$$

which converges to 1 for large values of c. Of course, for large values of c the ridge regression optimization problem hardly depends on the data anymore since

all the weight is on the data independent regularization term $\frac{c}{2}\|\theta\|^2$. Hence, the impact of regularization on the learning problem is making it less dependent on the data and thus also less dependent on small changes in the data. In other words regularization makes the learning problem more *robust/stable*, and thus regularization makes sense also from the learning point of view. We discuss this in more depth in the next chapter.

The practically challenging problem is to figure out the right amount of regularization, i.e., a good value for the regularization parameter c. The idea here is to split the data set into two parts, a training set and a so called validation set. Then a solution $\hat{\theta}$ of the ridge regression problem is computed for several values of $c \geq 0$. The performance of the resulting predictors is then compared on the validation set and the predictor with the best performance is chosen.

Choosing a good value for c is an instance of the more general hyper-parameter selection problem. Models for different values of the hyper-parameters are computed on the training data. Afterwards, the validation data are used to chose the best performing among these models. For validating the chosen model one should actually split the data set into three parts *training*, *validation* and *test*. The model that is chosen on the validation set is then validated on the test set. The reason for using a different part of the data for validation is that selecting good hyper-parameter values is also part of learning the model. Also for this part we run the risk of overfitting which can reflect itself in a much better performance on the validation data than on the test data. We come back to this issue with more explanations in later chapters.