

Machine Learning Course - CS-433

# Regularization: Ridge Regression and Lasso

Oct 7, 2021

minor changes by Martin Jaggi 2021,2020,2019,2018, minor changes by Rüdiger Urbanke 2017,  
changes by Martin Jaggi 2016 ©Mohammad Emtiyaz Khan 2015

Last updated on: October 5, 2021



## Motivation

We have seen that by augmenting the feature vector we can make linear models as powerful as we want. Unfortunately this leads to the problem of overfitting. *Regularization* is a way to mitigate this undesirable behavior.

We will discuss regularization in the context of linear models, but the same principle applies also to more complex models such as neural nets.

## Regularization

Through [regularization](#), we can penalize complex models and favor simpler ones:

$$\min_{\mathbf{w}} \quad \mathcal{L}(\mathbf{w}) + \Omega(\mathbf{w})$$

The second term  $\Omega$  is a [regularizer](#), measuring the complexity of the model given by  $\mathbf{w}$ .

## $L_2$ -Regularization: Ridge Regression

The most frequently used regularizer is the standard Euclidean norm ( $L_2$ -norm), that is

$$\Omega(\mathbf{w}) = \lambda \|\mathbf{w}\|_2^2$$

where  $\|\mathbf{w}\|_2^2 = \sum_i w_i^2$ . Here the main effect is that large model weights  $w_i$  will be penalized (avoided), since we consider them “unlikely”, while small ones are ok. When  $\mathcal{L}$  is MSE, this is called [ridge regression](#):

$$\min_{\mathbf{w}} \quad \frac{1}{2N} \sum_{n=1}^N [y_n - \mathbf{x}_n^\top \mathbf{w}]^2 + \lambda \|\mathbf{w}\|_2^2$$

*Least squares* is a special case of this: set  $\lambda := 0$ .

**Explicit solution for  $\mathbf{w}$ :** Differentiating and setting to zero:

$$\mathbf{w}_{\text{ridge}}^* = (\mathbf{X}^\top \mathbf{X} + \lambda' \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}$$

(here for simpler notation  $\frac{\lambda'}{2N} = \lambda$ )

## Ridge Regression to Fight Ill-Conditioning

The eigenvalues of  $(\mathbf{X}^\top \mathbf{X} + \lambda' \mathbf{I})$  are all at least  $\lambda'$  and so the inverse always exists. This is also referred to as *lifting the eigenvalues*.

*Proof:* Write the Eigenvalue decomposition of  $\mathbf{X}^\top \mathbf{X}$  as  $\mathbf{U} \mathbf{S} \mathbf{U}^\top$ . We then have

$$\begin{aligned}\mathbf{X}^\top \mathbf{X} + \lambda' \mathbf{I} &= \mathbf{U} \mathbf{S} \mathbf{U}^\top + \lambda' \mathbf{U} \mathbf{I} \mathbf{U}^\top \\ &= \mathbf{U} [\mathbf{S} + \lambda' \mathbf{I}] \mathbf{U}^\top.\end{aligned}$$

We see now that every Eigenvalue is “lifted” by an amount  $\lambda'$ .

Here is an alternative proof. Recall that for a symmetric matrix  $\mathbf{A}$  we can also compute eigenvalues by looking at the so-called Rayleigh ratio,

$$R(\mathbf{A}, \mathbf{v}) = \frac{\mathbf{v}^\top \mathbf{A} \mathbf{v}}{\mathbf{v}^\top \mathbf{v}}.$$

Note that if  $\mathbf{v}$  is an eigenvector with eigenvalue  $\lambda$  then the Rayleigh coefficient indeed gives us  $\lambda$ . We can find the smallest and largest eigenvalue by minimizing and maximizing this coefficient. But note that if we apply this to the symmetric matrix  $\mathbf{X}^\top \mathbf{X} + \lambda' \mathbf{I}$  then for any vector  $\mathbf{v}$  we have

$$\frac{\mathbf{v}^\top (\mathbf{X}^\top \mathbf{X} + \lambda' \mathbf{I}) \mathbf{v}}{\mathbf{v}^\top \mathbf{v}} \geq \frac{\lambda' \mathbf{v}^\top \mathbf{v}}{\mathbf{v}^\top \mathbf{v}} = \lambda'.$$

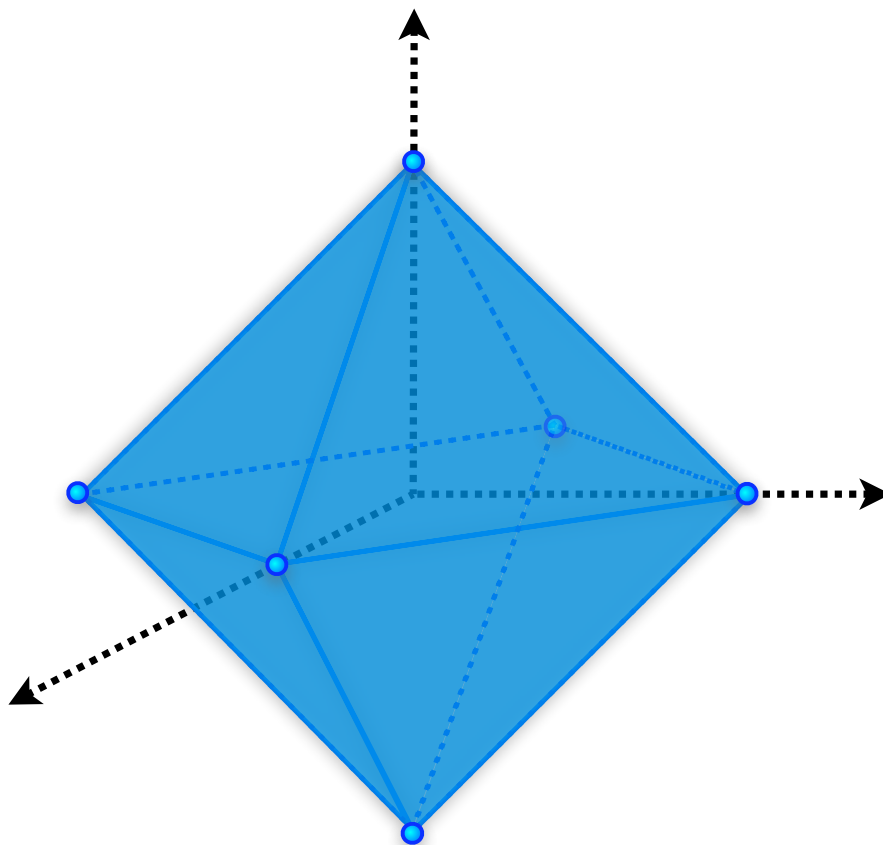
# $L_1$ -Regularization: The Lasso

As an alternative measure of the complexity of the model, we can use a different norm. A very important case is the  $L_1$ -norm, leading to  $L_1$ -regularization. In combination with the MSE cost function, this is known as the **Lasso**:

$$\min_{\mathbf{w}} \quad \frac{1}{2N} \sum_{n=1}^N [y_n - \mathbf{x}_n^\top \mathbf{w}]^2 + \lambda \|\mathbf{w}\|_1$$

where

$$\|\mathbf{w}\|_1 := \sum_i |w_i|.$$



The figure above shows a “ball” of constant  $L_1$  norm. To keep things simple assume that  $\mathbf{X}^\top \mathbf{X}$  is invertible. We claim that in this case the set

$$\{\mathbf{w} : \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 = \alpha\} \tag{1}$$

is an ellipsoid and this ellipsoid simply scales around its origin as we change  $\alpha$ . We claim that for the  $L_1$ -regularization the optimum solution is likely going to be sparse (only has few non-zero components) compared to the case where we use  $L_2$ -regularization.

Why is this the case? Assume that a genie tells you the  $L_1$ -norm of the optimum solution. Draw the  $L_1$ -ball with that norm value (think of 2D to visualize it). So now you know that the optimal point is somewhere on the surface of this “ball”. Further you know that there are ellipsoids, all with the same mean and rotation that describes the equal error surfaces incurred by the first term. The optimum solution is where the “smallest” of these ellipsoids just touches the  $L_1$ -ball. Due to the geometry of this ball this point is more likely to be on one of the “corner” points. In turn, sparsity is desirable, since it leads to a “simple” model.

How do we see the claim that (1) describes an ellipsoid? First look at  $\alpha = \|\mathbf{X}\mathbf{w}\|^2 = \mathbf{w}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w}$ . This is a quadratic form. Let  $\mathbf{A} = \mathbf{X}^\top \mathbf{X}$ . Note that  $\mathbf{A}$  is a symmetric matrix and by assumption it has full rank. If  $\mathbf{A}$  is a diagonal matrix with strictly positive elements  $a_i$  along the diagonal then this describes the equation

$$\sum_i a_i \mathbf{w}_i^2 = \alpha,$$

which is indeed the equation for an ellipsoid. In the general case,  $\mathbf{A}$  can be written as (using the SVD)  $\mathbf{A} = \mathbf{U}\mathbf{B}\mathbf{U}^\top$ , where  $\mathbf{B}$  is a diagonal matrix with strictly positive entries. This then corresponds to an ellipsoid with rotated axes. If we now look at  $\alpha = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$ , where  $\mathbf{y}$  is in the column space of  $\mathbf{X}$  then we can write it as  $\alpha = \|\mathbf{X}(\mathbf{w}_0 - \mathbf{w})\|^2$  for a suitable chosen  $\mathbf{w}_0$  and so this corresponds to a shifted ellipsoid. Finally, for the general case, write  $\mathbf{y}$  as  $\mathbf{y} = \mathbf{y}_\parallel + \mathbf{y}_\perp$ , where  $\mathbf{y}_\parallel$  is the component of  $\mathbf{y}$  that lies in the subspace spanned by the columns of  $\mathbf{X}$  and  $\mathbf{y}_\perp$  is the component that

is orthogonal. In this case

$$\begin{aligned}\alpha &= \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 \\ &= \|\mathbf{y}_{\parallel} + \mathbf{y}_{\perp} - \mathbf{X}\mathbf{w}\|^2 \\ &= \|\mathbf{y}_{\perp}\|^2 + \|\mathbf{y}_{\parallel} - \mathbf{X}\mathbf{w}\|^2 \\ &= \|\mathbf{y}_{\perp}\|^2 + \|\mathbf{X}(\mathbf{w}_0 - \mathbf{w})\|^2.\end{aligned}$$

Hence this is then equivalent to the equation  $\|\mathbf{X}(\mathbf{w}_0 - \mathbf{w})\|^2 = \alpha - \|\mathbf{y}_{\perp}\|^2$ , proving the claim. From this we also see that if  $\mathbf{X}^{\top}\mathbf{X}$  is not full rank then what we get is not an ellipsoid but a cylinder with an ellipsoidal cross-section.



# Additional Notes

## Other Types of Regularization

Popular methods such as [shrinkage](#), [dropout](#) and [weight decay](#) (in the context of neural networks), [early stopping of the optimization](#) are all different forms of regularization.

Another view of regularization: The ridge regression formulation we have seen above is similar to the following constrained problem (for some  $\tau > 0$ ).

$$\min_{\mathbf{w}} \quad \frac{1}{2N} \sum_{n=1}^N (y_n - \mathbf{x}_n^\top \mathbf{w})^2, \quad \text{such that } \|\mathbf{w}\|_2^2 \leq \tau$$

The following picture illustrates this.

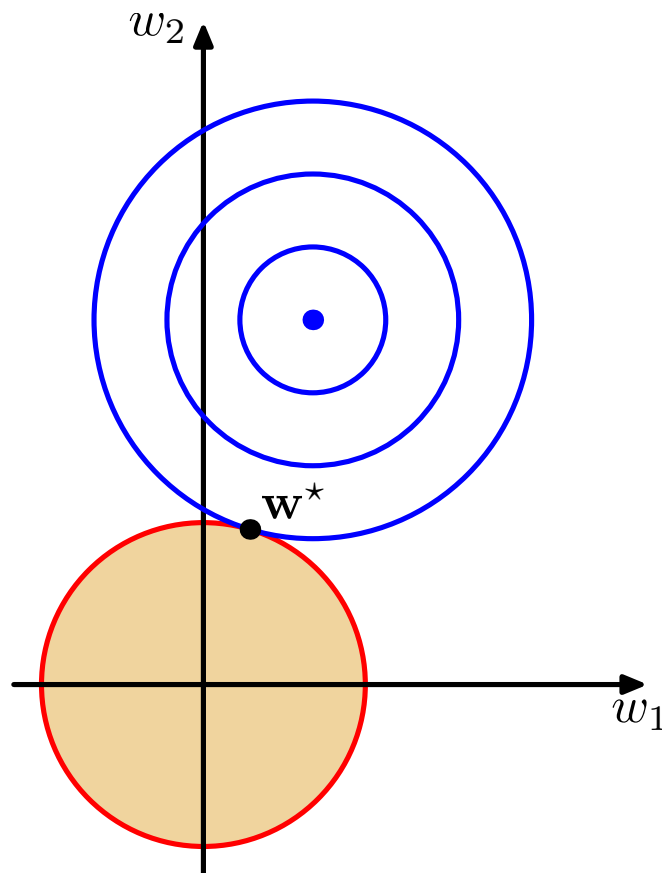


Figure 1: Geometric interpretation of Ridge Regression. Blue lines indicating the level sets of the MSE cost function.

For the case of using  $L_1$  regularization (known as the [Lasso](#), when used with MSE) we analogously consider

$$\min_{\mathbf{w}} \quad \frac{1}{2N} \sum_{n=1}^N (y_n - \mathbf{x}_n^\top \mathbf{w})^2, \quad \text{such that } \|\mathbf{w}\|_1 \leq \tau$$

This forces some of the elements of  $\mathbf{w}$  to be strictly 0 and therefore enforces sparsity in the model (some features will not be used since their coefficients are zero).

- Why does  $L_1$  regularizer enforce sparsity? *Hint:* Draw the picture similar to above, and locate the optimal solution.
- Why is it good to have sparsity in the model? Is it going to be better than least-squares? When and why?

## Ridge Regression as MAP estimator

Recall that classic *least-squares linear regression* can be interpreted as the [maximum likelihood estimator](#):

$$\begin{aligned}\mathbf{w}_{\text{lse}} &\stackrel{(a)}{=} \arg \min_{\mathbf{w}} -\log p(\mathbf{y}, \mathbf{X} | \mathbf{w}) \\ &\stackrel{(b)}{=} \arg \min_{\mathbf{w}} -\log p(\mathbf{X} | \mathbf{w}) p(\mathbf{y} | \mathbf{X}, \mathbf{w}) \\ &\stackrel{(c)}{=} \arg \min_{\mathbf{w}} -\log p(\mathbf{X}) p(\mathbf{y} | \mathbf{X}, \mathbf{w}) \\ &\stackrel{(d)}{=} \arg \min_{\mathbf{w}} -\log p(\mathbf{y} | \mathbf{X}, \mathbf{w}) \\ &\stackrel{(e)}{=} \arg \min_{\mathbf{w}} -\log \left[ \prod_{n=1}^N p(y_n | \mathbf{x}_n, \mathbf{w}) \right] \\ &\stackrel{(f)}{=} \arg \min_{\mathbf{w}} -\log \left[ \prod_{n=1}^N \mathcal{N}(y_n | \mathbf{x}_n^\top \mathbf{w}, \sigma^2) \right] \\ &= \arg \min_{\mathbf{w}} -\log \left[ \prod_{n=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y_n - \mathbf{x}_n^\top \mathbf{w})^2} \right] \\ &= \arg \min_{\mathbf{w}} -N \log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) + \sum_{n=1}^N \frac{1}{2\sigma^2} (y_n - \mathbf{x}_n^\top \mathbf{w})^2 \\ &= \arg \min_{\mathbf{w}} \frac{1}{2\sigma^2} \sum_{n=1}^N (y_n - \mathbf{x}_n^\top \mathbf{w})^2\end{aligned}$$

In step (a) on the right we wrote down the negative of the log of the likelihood. The maximum likelihood criterion chooses that parameter  $\mathbf{w}$  that minimizes this quantity (i.e., maximizes the likelihood). In step (b) we factored the likelihood. The usual assumption is that the choice of the input samples  $\mathbf{x}_n$  does not depend on the model parameter (which only influences the output given the input. Hence, in step (c) we removed the conditioning. Since the factor  $p(\mathbf{X})$  does not depend on  $\mathbf{w}$ , i.e., is a constant wrt to  $\mathbf{w}$ ) we can remove it. This is done in step (d). In step (e) we used the assumption that the samples are iid. In step (f) we then used our assumption that the samples have the form  $y_n = \mathbf{w}_n^\top \mathbf{w} + Z_n$ ,

where  $Z_n$  is a Gaussian noise with mean zero and variance  $\sigma_2$ . The rest is calculus.

*Ridge regression* has a very similar interpretation. Now we start with the posterior  $p(\mathbf{w}|\mathbf{X}, \mathbf{y})$  and chose that parameter  $\mathbf{w}$  that maximizes this posterior. Hence this is called the [maximum-a-posteriori \(MAP\) estimate](#). As before, we take the log and add a minus sign and minimize instead. In order to compute the posterior we use Bayes law and we assume that the components of the weight vector are iid Gaussians with mean zero and variance  $\frac{1}{\lambda}$ .

$$\begin{aligned}
\mathbf{w}_{\text{ridge}} &= \arg \min_{\mathbf{w}} -\log p(\mathbf{w}|\mathbf{X}, \mathbf{y}) \\
&\stackrel{(a)}{=} \arg \min_{\mathbf{w}} -\log \frac{p(\mathbf{y}, \mathbf{X}|\mathbf{w})p(\mathbf{w})}{p(\mathbf{y}, \mathbf{X})} \\
&\stackrel{(b)}{=} \arg \min_{\mathbf{w}} -\log p(\mathbf{y}, \mathbf{X}|\mathbf{w})p(\mathbf{w}) \\
&\stackrel{(c)}{=} \arg \min_{\mathbf{w}} -\log p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w}) \\
&= \arg \min_{\mathbf{w}} -\log \left[ p(\mathbf{w}) \prod_{n=1}^N p(y_n|\mathbf{x}_n, \mathbf{w}) \right] \\
&= \arg \min_{\mathbf{w}} -\log \left[ \mathcal{N}(\mathbf{w} | 0, \frac{1}{\lambda}\mathbf{I}) \prod_{n=1}^N \mathcal{N}(y_n | \mathbf{x}_n^\top \mathbf{w}, \sigma^2) \right] \\
&= \arg \min_{\mathbf{w}} -\log \left[ \frac{1}{(2\pi\frac{1}{\lambda})^{D/2}} e^{-\frac{\lambda}{2}\|\mathbf{w}\|^2} \prod_{n=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y_n - \mathbf{x}_n^\top \mathbf{w})^2} \right] \\
&= \arg \min_{\mathbf{w}} \sum_{n=1}^N \frac{1}{2\sigma^2}(y_n - \mathbf{x}_n^\top \mathbf{w})^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2.
\end{aligned}$$

In step (a) we used Bayes' law. In step (b) and (c) we eliminated quantities that do not depend on  $\mathbf{w}$ .