Machine Learning Course - CS-433

Regularization: Ridge and Lasso

Oct 5, 2017

changes by Martin Jaggi 2016

small changes by Rüdiger Urbanke 2017

Last updated: October 5, 2017

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

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 Ω 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:

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

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

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

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

(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 λ' and so the inverse always exists. This is also referred to as *lifting the eigenvalues*.

Proof: Write the singular-value decomposition of $\mathbf{X}^\top\mathbf{X}$ as $\mathbf{USU}^\top$. We then have

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

We see now that every singular value is "lifted" by an amount λ'.

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 λ then the Rayleigh coefficient indeed gives us λ. 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

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

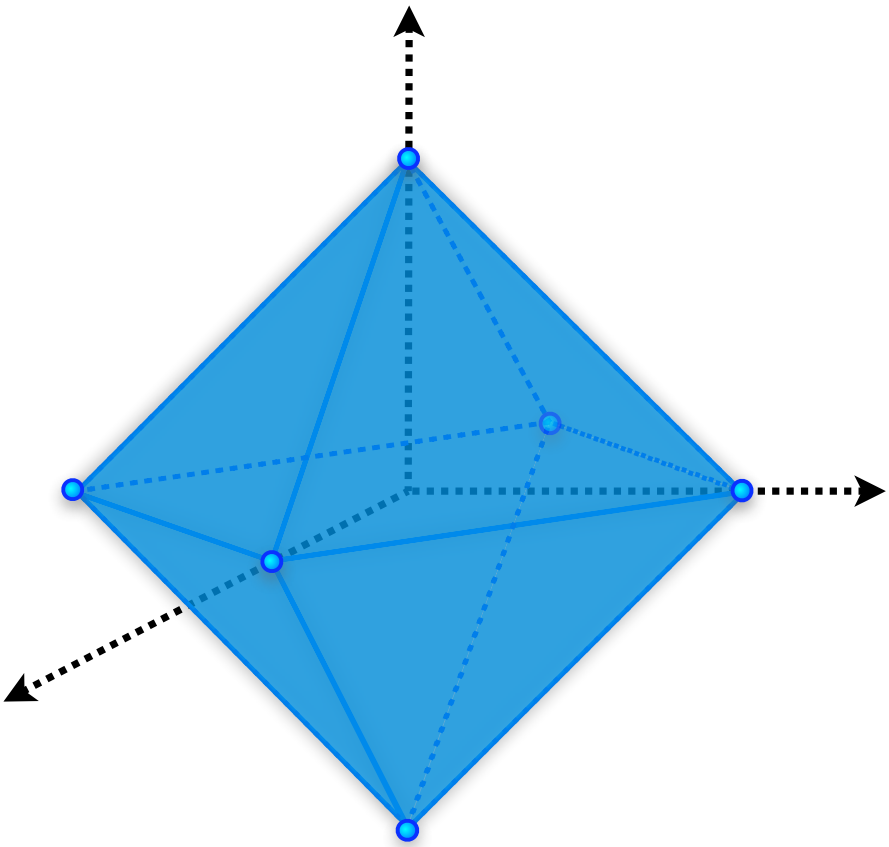
L_1-Regularization: The Lasso

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

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

is an ellipsoide and this ellipsoide simply scales around it's origin as we change α. The optimum solution is not gotten by scaling both, this ellipsoid as well the constant L_1 "ball", so that they both "just touch" in such a way that the sum of the two contributions is minimum. Because of the geometry of the L_1 ball it is now quite likely that the point where they touch is on one of the lower-dimensional faces, i.e., the solution is sparse.

In turn, sparsity is desirable, since it leads to a "simple" model.

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 $\tau > 0$).

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

The following picture illustrates this.

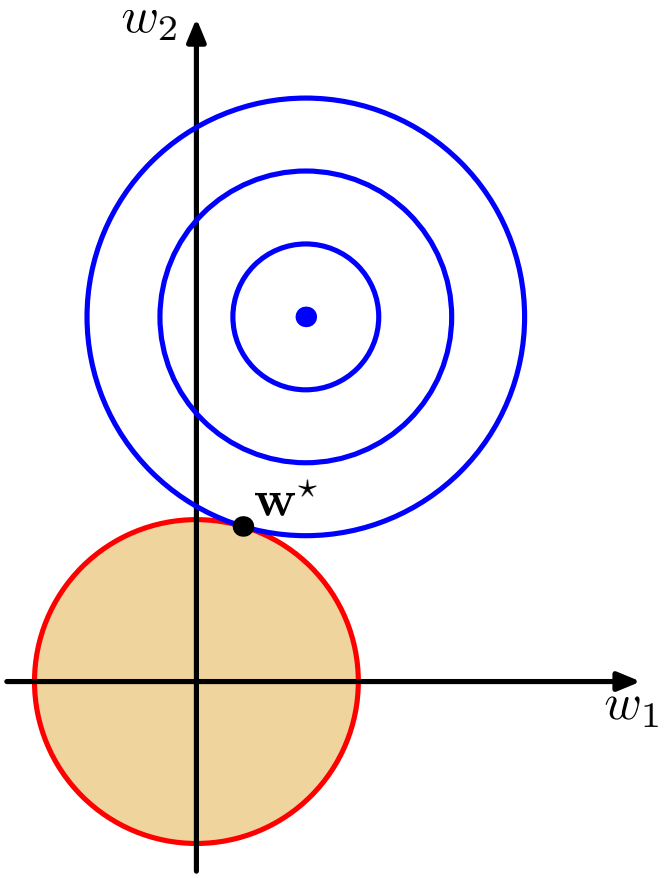


Figure 1: Geometric interpretation of Ridge Regression. Blue lines indicating the level sets of the MSE cost function.

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

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

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 for Lasso, 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 least-squares can be interpreted as the maximum likelihood estimator.

$$\mathbf{w}_{\text{lse}} = \arg\max_{\mathbf{w}} \quad \log\left[\prod_{n=1}^{N}\mathcal{N}(y_n \,|\, \mathbf{x}_n^\top \mathbf{w}, \sigma^2)\right]$$

Ridge regression has a very similar interpretation:

$$\mathbf{w}_{\text{ridge}} = \arg\max_{\mathbf{w}} \quad \log\left[\prod_{n=1}^{N}\mathcal{N}(y_n \,|\, \mathbf{x}_n^\top \mathbf{w}, \sigma^2)\cdot\mathcal{N}(\mathbf{w} \,|\, 0, \tfrac{1}{\lambda}\mathbf{I})\right]$$

This is called a Maximum-a-posteriori (MAP) estimate.

Plug in the definition of the Gaussian distribution, and take the log, to see that you obtain the Ridge Regression optimization problem.

MAP Estimate and Bayes' Rule

In general, a MAP estimate maximizes the product of the likelihood *and* the prior (on the model), as opposed to a maximum likelihood estimate

which maximizes only the former.

$$\mathbf{w}_{\text{lik}} = \arg\max_{\mathbf{w}} \; p(\mathbf{y} \,|\, \mathbf{w}, \mathbf{X}, \boldsymbol{\lambda}) \tag{1}$$

$$\mathbf{w}_{\text{MAP}} = \arg\max_{\mathbf{w}} \; p(\mathbf{y} \,|\, \mathbf{w}, \mathbf{X}, \boldsymbol{\lambda}) \cdot p(\mathbf{w} \,|\, \boldsymbol{\theta}) \tag{2}$$

Here $p(\mathbf{y} \,|\, \mathbf{w}, \mathbf{X}, \boldsymbol{\lambda})$ defines the *likelihood* of observing the data $\mathbf{y}$ given $\mathbf{X}, \mathbf{w}$ and some likelihood parameters $\boldsymbol{\lambda}$.
Similarly, $p(\mathbf{w} \,|\, \boldsymbol{\lambda})$ is the *prior* distribution. This incorporates our prior knowledge about $\mathbf{w}$. Using the Bayes rule,

$$p(A, B) = p(A|B)\, p(B) = p(B|A)\, p(A) \tag{3}$$

we can rewrite the MAP estimate, as follows:

$$\mathbf{w}_{\text{MAP}} = \arg\max_{\mathbf{w}} \; p(\mathbf{w} \,|\, \mathbf{y}, \mathbf{X}, \boldsymbol{\lambda}, \boldsymbol{\theta}) \tag{4}$$

Therefore, the MAP estimate is the maximum of the posterior distribution, which explains the name.