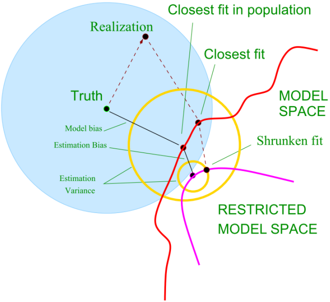
Einführung in das Statistische Lernen

Introduction to Regularization



Learning goals

- Understand why overfitting happens
- Know how overfitting can be avoided
- Know regularized empirical risk minimization

Motivation for Regularization

EXAMPLE: OVERFITTING

- Assume we want to predict the daily maximum **ozone level** in LA given a data set containing 50 observations.
- The data set contains 12 features describing time conditions (e.g., weekday, month), the weather (e.g., temperature at different weather stations, humidity, wind speed) or geographic variables (e.g., the pressure gradient).
- We fit a linear regression model using **all** of the features

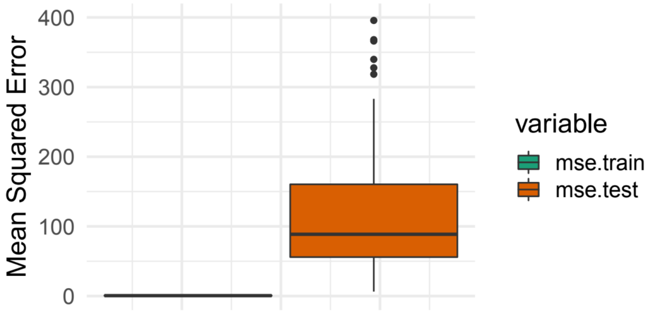
  $$f(\mathbf{x} \mid \boldsymbol{\theta}) = \boldsymbol{\theta}^T \mathbf{x} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_{12} x_{12}$$

  with the *L*2 loss.
- We evaluate the performance with 10 times 10-fold CV.

We use (a subset of) the `Ozone` data set from the `mlbench` package. This way, we artificially create a “high-dimensional” dataset by reducing the number of observations drastically while keeping the number of features fixed.

EXAMPLE: OVERFITTING

While our model fits the training data almost perfectly (left), it generalizes poorly to new test data (right). We overfitted.



AVOID OVERFITTING

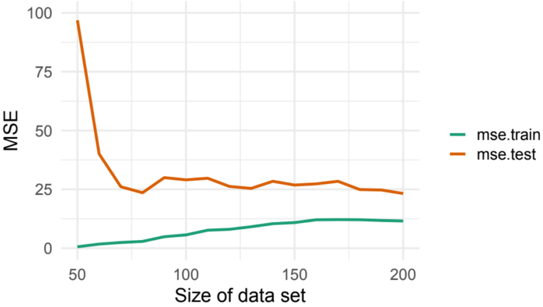
Why can **overfitting** happen? And how to avoid it?

1. Not enough data
   $\rightarrow$ collect **more data**
2. Data is noisy
   $\rightarrow$ collect **better data** (reduce noise)
3. Models are too complex
   $\rightarrow$ use **less complex models**
4. Aggressive loss optimization
   $\rightarrow$ **optimize less**

AVOID OVERFITTING

Approach 1: Collect more data

We explore our results for increased dataset size by 10 times 10-fold CV. The fit worsens slightly, but the test error decreases.



Good idea, but often not feasible in practice.

AVOID OVERFITTING

Approach 3: Reduce complexity

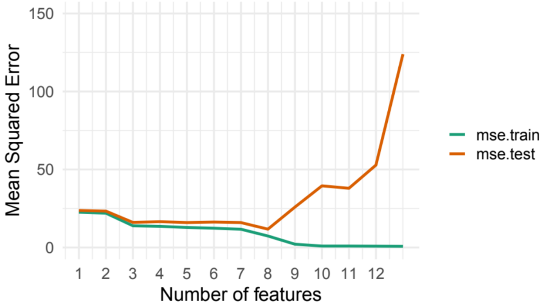
We try the simplest model we can think of: the constant model. For the *L*2 loss, the optimal constant model is

$$f(\mathbf{x} \mid \theta) = \frac{1}{n} \sum_{i=1}^{n} y^{(i)}$$

We then increase the complexity of the model step-by-step by adding one feature at a time.

AVOID OVERFITTING

We can control the complexity of the model by including/excluding features. We can try out all feature combinations and investigate the model fit.



Note: For simplicity, we added the features in one specific (clever) order, so we cheated a bit. Also note there are $2^{12} = 4096$ potential feature combinations.

AVOID OVERFITTING

Approach 4: Optimize less

Now we use polynomial regression with temperature as the only feature to predict the ozone level, i.e.,

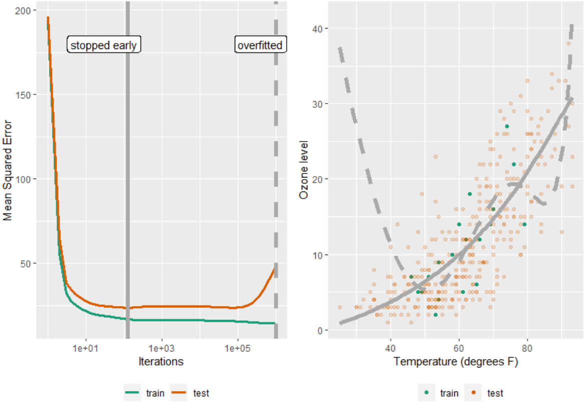
$$f(\mathbf{x} \mid \boldsymbol{\theta}) = \sum_{i=0}^{d} \theta_i (x_T)^i.$$

We choose $d = 15$, for which we get a very flexible model, which can be prone to overfitting for small data sets.

In this example, we don't solve for $\hat{\theta}$ directly, but instead, we use the gradient descent algorithm to find $\hat{\theta}$ stepwise.

AVOID OVERFITTING

We want to stop the optimization early when the generalization error starts to degrade.



Note: For polynomial regression, gradient descent usually needs many iterations before it starts to overfit. Hence a very small training set was chosen to accelerate this effect.

AVOID OVERFITTING

We have contradictory goals

- **maximizing the fit** (minimizing the train loss)
- **minimizing the complexity** of the model.

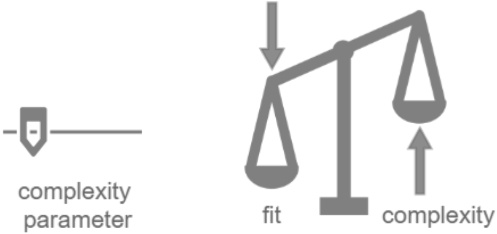
We need to find the “sweet spot”.

fit
complexity

AVOID OVERFITTING

Until now, we can either add a feature completely or not at all.

Instead of controlling the complexity in a discrete way by specifying the number of features, we might prefer to control the complexity **on a continuum** from simple to complex.



Regularized Empirical Risk Minimization

REGULARIZED EMPIRICAL RISK MINIMIZATION

Recall, empirical risk minimization with a complex hypothesis set tends to overfit. A major tool to handle overfitting is **regularization**.

In the broadest sense, regularization refers to any modification made to a learning algorithm that is intended to reduce its generalization error but not its training error.

Explicitly or implicitly, such modifications represent the preferences we have regarding the elements of the hypothesis set.

REGULARIZED EMPIRICAL RISK MINIMIZATION

Commonly, regularization takes the following form:

$$\mathcal{R}_{\text{reg}}(f) = \mathcal{R}_{\text{emp}}(f) + \lambda \cdot J(f) = \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right) + \lambda \cdot J(f)$$

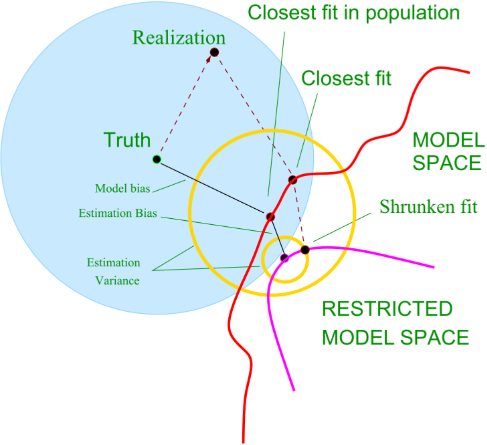
- $J(f)$ is called **complexity penalty**, **roughness penalty** or **regularizer**.
- $\lambda > 0$ is called **complexity control** parameter.
- It measures the “complexity” of a model and penalizes it in the fit.
- As for $\mathcal{R}_{\text{emp}}$, often $\mathcal{R}_{\text{reg}}$ and J are defined on $\boldsymbol{\theta}$ instead of f, so $\mathcal{R}_{\text{reg}}(\boldsymbol{\theta}) = \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) + \lambda \cdot J(\boldsymbol{\theta})$.

REGULARIZED EMPIRICAL RISK MINIMIZATION

Remarks:

- Note that we now face an optimization problem with two criteria:
  1. models should fit well (low empirical risk),
  2. but not be too complex (low $J(f)$).
- We decide to combine the two in a weighted sum and to control the trade-off via the complexity control parameter λ.
- λ is hard to set manually and is usually selected via cross-validation (see later).
- $\lambda = 0$: The regularized risk $\mathcal{R}_{\text{reg}}(f)$ reduces to the simple empirical $\mathcal{R}_{\text{emp}}(f)$.
- If λ goes to infinity, we stop caring about the loss/fit and models become as “simple” as possible.

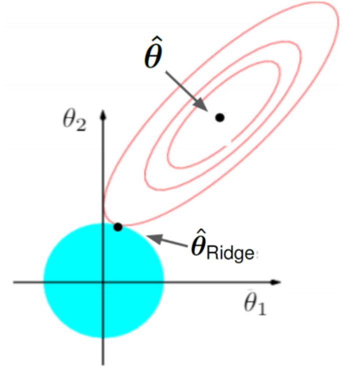
REGULARIZED EMPIRICAL RISK MINIMIZATION



Hastie, The Elements of Statistical Learning, 2009 (p. 225)

Einführung in das Statistische Lernen

Lasso and Ridge Regression



Learning goals

- Know the regularized linear model
- Know Ridge regression ($L2$ penalty)
- Know Lasso regression ($L1$ penalty)

REGULARIZATION IN THE LINEAR MODEL

- Linear models can also overfit if we operate in a high-dimensional space with not that many observations.
- OLS usually require a full-rank design matrix.
- When features are highly correlated, the least-squares estimate becomes highly sensitive to random errors in the observed response, producing a large variance in the fit.
- We now add a complexity penalty to the loss:

$$\mathcal{R}_{\text{reg}}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \left(y^{(i)} - \boldsymbol{\theta}^\top \mathbf{x}^{(i)}\right)^2 + \lambda \cdot J(\boldsymbol{\theta}).$$

- Intuitive to measure model complexity as deviation from the 0-origin, as the 0-model is empty and contains no effects. Models close to this either have few active features or only weak effects.
- So we measure $J(\boldsymbol{\theta})$ through a vector norm. This shrinks coefficients closer 0, hence the term **shrinkage methods**.

RIDGE REGRESSION

Ridge regression uses a simple *L*2 penalty:

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{\text{Ridge}} &= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \left(y^{(i)} - \boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)^2 + \lambda \|\boldsymbol{\theta}\|_2^2 \\ &= \arg\min_{\boldsymbol{\theta}} \left(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\right)^\top \left(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\right) + \lambda \boldsymbol{\theta}^\top \boldsymbol{\theta}.\end{aligned}$$

Optimization is possible (as in the normal LM) in analytical form:

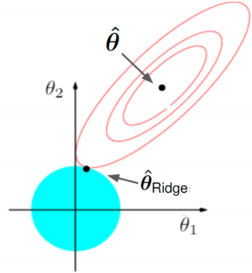
$$\hat{\boldsymbol{\theta}}_{\text{Ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda \boldsymbol{I})^{-1}\mathbf{X}^T\mathbf{y}$$

Name comes from the fact that we add positive entries along the diagonal "ridge" $\mathbf{X}^T\mathbf{X}$.

RIDGE REGRESSION

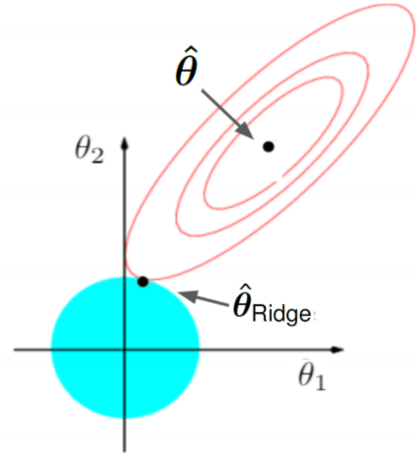
We understand the geometry of these 2 mixed components in our regularized risk objective much better, if we formulate the optimization as a constrained problem (see this a Lagrange multipliers in reverse).

$$
\begin{aligned}
\min_{\boldsymbol{\theta}} \quad & \sum_{i=1}^{n} \left(y^{(i)} - f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta} \right) \right)^2 \\
\text{s.t.} \quad & \|\boldsymbol{\theta}\|_2^2 \leq t
\end{aligned}
$$


$\hat{\boldsymbol{\theta}}$
θ_2
$\hat{\boldsymbol{\theta}}_{\text{Ridge}}$
θ_1

NB: Relationship between λ and t will be explained later.

RIDGE REGRESSION



- We still optimize the $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$, but cannot leave a ball around the origin.
- $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ grows monotonically if we move away from $\hat{\boldsymbol{\theta}}$.
- Inside constraints perspective: From origin, jump from contour line to contour line (better) until you become infeasible, stop before.
- Outside constraints perspective: From $\hat{\boldsymbol{\theta}}$, jump from contour line to contour line (worse) until you become feasible, stop then.
- So our new optimum will lie on the boundary of that ball.

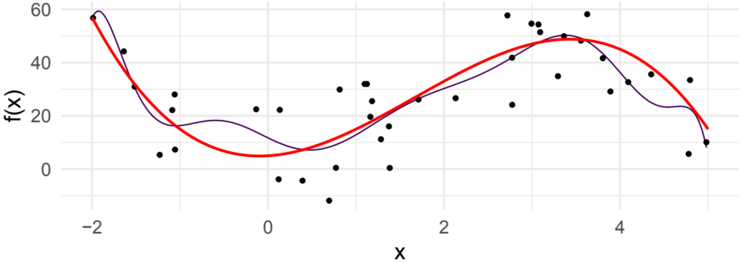
EXAMPLE: POLYNOMIAL RIDGE REGRESSION

True (unknown) function is $f(x) = 5 + 2x + 10x^2 - 2x^3 + \epsilon$ (in red).

Let us consider a dth-order polynomial

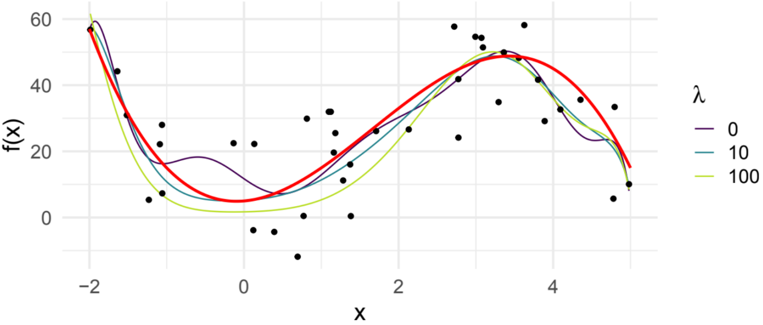
$$f(x) = \theta_0 + \theta_1 x + \cdots + \theta_d x^d = \sum_{j=0}^{d} \theta_j x^j.$$

Using model complexity $d = 10$ overfits:



EXAMPLE: POLYNOMIAL RIDGE REGRESSION

With an *L*2 penalty we can now select *d* "too large" but regularize our model by shrinking its coefficients. Otherwise we have to optimize over the discrete *d*.



λ	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}
0.00	12.00	-16.00	4.80	23.00	-5.40	-9.30	4.20	0.53	-0.63	0.13	-0.01
10.00	5.20	1.30	3.70	0.69	1.90	-2.00	0.47	0.20	-0.14	0.03	-0.00
100.00	1.70	0.46	1.80	0.25	1.80	-0.94	0.34	-0.01	-0.06	0.02	-0.00

LASSO REGRESSION

Another shrinkage method is the so-called **Lasso regression**, which uses an $L1$ penalty on $\boldsymbol{\theta}$:

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{\text{Lasso}} &= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \left(y^{(i)} - \boldsymbol{\theta}^T \mathbf{x}^{(i)} \right)^2 + \lambda \|\boldsymbol{\theta}\|_1 \\ &= \arg\min_{\boldsymbol{\theta}} \left(\mathbf{y} - \mathbf{X}\boldsymbol{\theta} \right)^\top \left(\mathbf{y} - \mathbf{X}\boldsymbol{\theta} \right) + \lambda \|\boldsymbol{\theta}\|_1 .\end{aligned}$$

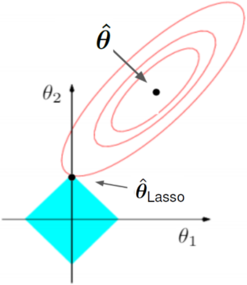
Note that optimization now becomes much harder. $\mathcal{R}_{\text{reg}}(\boldsymbol{\theta})$ is still convex, but we have moved from an optimization problem with an analytical solution towards a non-differentiable problem.

Name: least absolute shrinkage and selection operator.

LASSO REGRESSION

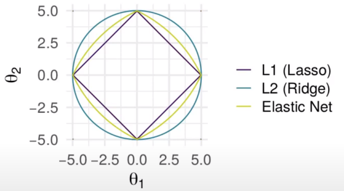
We can also rewrite this as a constrained optimization problem. The penalty results in the constrained region to look like a diamond shape.

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \sum_{i=1}^{n} \left(y^{(i)} - f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right)^2 \\ \text{subject to:} \quad & \|\boldsymbol{\theta}\|_1 \leq t \end{aligned}$$


$\hat{\boldsymbol{\theta}}$
θ_2
$\hat{\boldsymbol{\theta}}_{\text{Lasso}}$
θ_1

Einführung in das Statistische Lernen

Elastic Net and Regularization for GLMs



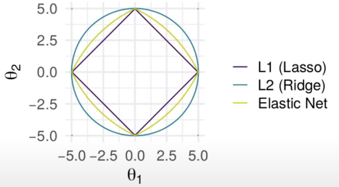
Learning goals

- Know the elastic net as compromise between Ridge and Lasso regression
- Know regularized logistic regression

ELASTIC NET

Elastic Net combines the L_1 and L_2 penalties:

$$\mathcal{R}_{\text{elnet}}(\boldsymbol{\theta}) = \sum_{i=1}^{n}(y^{(i)} - \boldsymbol{\theta}^\top \mathbf{x}^{(i)})^2 + \lambda_1 \|\boldsymbol{\theta}\|_1 + \lambda_2 \|\boldsymbol{\theta}\|_2^2.$$



- Correlated predictors tend to be either selected or zeroed out together.
- Selection of more than n features possible for $p > n$.

ELASTIC NET

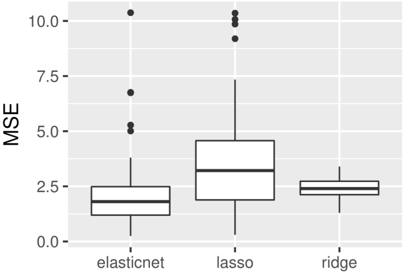
Simulating two examples with each 50 data sets and 100 observations each:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \sigma\epsilon, \quad \epsilon \sim N(0,1), \quad \sigma = 1$$

Ridge performs better for:

$$\boldsymbol{\beta} = (\underbrace{2,\ldots,2}_{5}, \underbrace{0,\ldots,0}_{5})$$

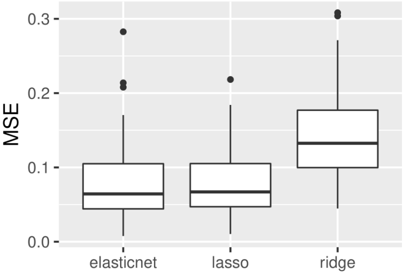
$$\text{corr}(\mathbf{X}_i, \mathbf{X}_j) = 0.8^{|i-j|} \text{ for all } i \text{ and } j$$



Lasso performs better for:

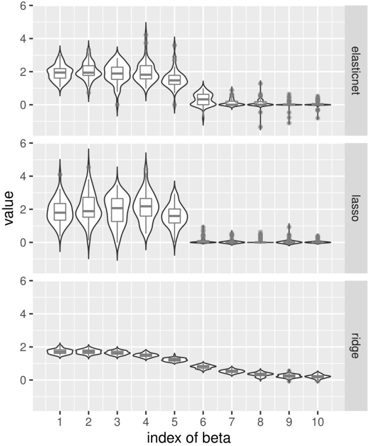
$$\boldsymbol{\beta} = (2,2,2,\underbrace{0,\ldots,0}_{7})$$

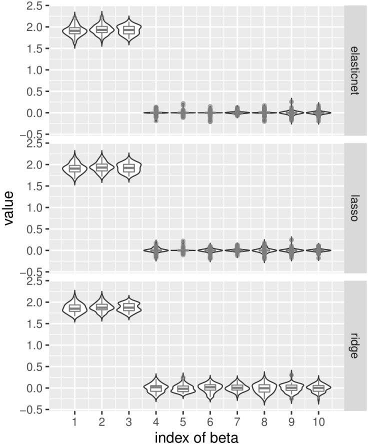
$$\text{corr}(\mathbf{X}_i, \mathbf{X}_j) = 0 \text{ for all } i \neq j\text{, otherwise } 1$$



©

ELASTIC NET





Since Elastic Net offers a compromise between Ridge and Lasso, it is suitable for both data situations.

REGULARIZED LOGISTIC REGRESSION

Regularizers can be added very flexibly to basically any model which is based on ERM.

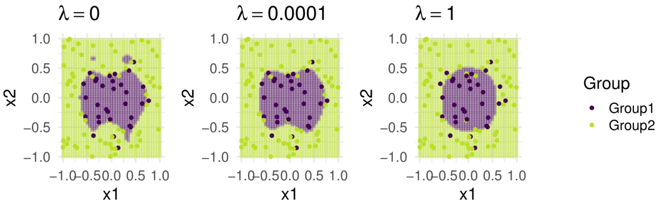
Hence, we can, e.g., construct L_1- or L_2-penalized logistic regression to enable coefficient shrinkage and variable selection in this model.

$$\begin{aligned}\mathcal{R}_{\text{reg}}(\boldsymbol{\theta}) &= \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) + \lambda \cdot J(\boldsymbol{\theta}) \\ &= \sum_{i=1}^{n} \log\left[1 + \exp\left(-2y^{(i)} f\left(\mathbf{x}^{(i)} \;\middle|\; \boldsymbol{\theta}\right)\right)\right] + \lambda \cdot J(\boldsymbol{\theta})\end{aligned}$$

REGULARIZED LOGISTIC REGRESSION

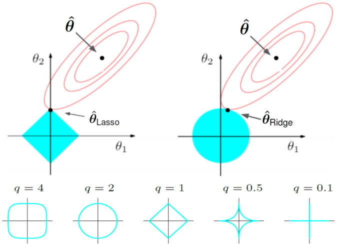
We fit a logistic regression model using polynomial features for x_1 and x_2 with maximum degree of 7. We add an L_2 penalty. We see for

- $\lambda = 0$: The unregularized model seems to overfit.
- $\lambda = 0.0001$: Regularization helps to learn the underlying mechanism.
- $\lambda = 1$: The real data-generating process is captured very well.



Einführung in das Statistische Lernen

L0 Regularization



Learning goals

- Know LQ norm regularization
- Understand that L0 norm realization simply counts the number of non-zero parameters

LQ NORM REGULARIZATION

Besides L_1 and L_2 norm we could use any L_q norm for regularization.

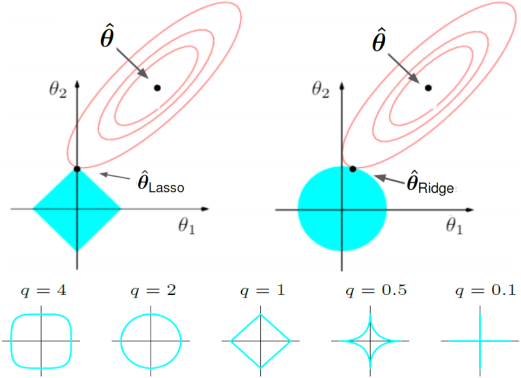


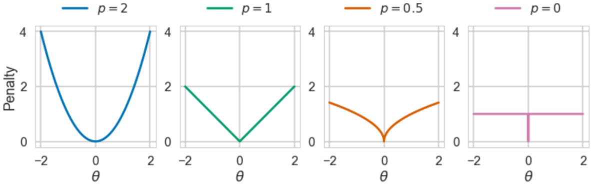
Figure: *Top:* Ridge and Lasso loss contours and feasible regions. *Bottom:* Different feasible region shapes for L_q norms $\sum_j |\theta_j|^q$.

L0 REGULARIZATION

- Consider the L_0-regularized risk of a model $f(\mathbf{x} \mid \boldsymbol{\theta})$

$$\mathcal{R}_{\text{reg}}(\boldsymbol{\theta}) = \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_0 := \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) + \lambda \sum_j |\theta_j|^0.$$

- Unlike the L_1 and L_2 norms, the L_0 "norm" simply counts the number of non-zero parameters in the model.



Credit: Christos Louizos

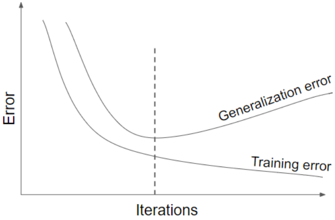
Figure: L_p norm penalties for a parameter $\boldsymbol{\theta}$ according to different values of p.

L0 REGULARIZATION

- For any parameter $\boldsymbol{\theta}$, the L_0 penalty is zero for $\boldsymbol{\theta} = 0$ (defining $0^0 := 0$) and is constant for any $\boldsymbol{\theta} \neq 0$, no matter how large or small it is.
- L_0 regularization induces sparsity in the parameter vector more aggressively than L_1 regularization, but does not shrink concrete parameter values as L1 and L2 does.
- Model selection criteria such as Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) are special cases of L_0 regularization (corresponding to specific values of λ).
- The L_0-regularized risk is neither continuous, differentiable or convex.
- It is computationally hard to optimize (NP-hard) and likely intractable. For smaller n and p we might be able to solve this nowadays directly, for larger scenarios efficient approximations of the L_0 are still topic of current research.

Einführung in das Statistische Lernen

Early Stopping



Learning goals

- Know how early stopping works
- Understand how early stopping acts as a regularizer

EARLY STOPPING

- When training with an iterative optimizer such as SGD, it is commonly the case that, after a certain number of iterations, generalization error begins to increase even though training error continues to decrease.
- **Early stopping** refers to stopping the algorithm early before the generalization error increases.

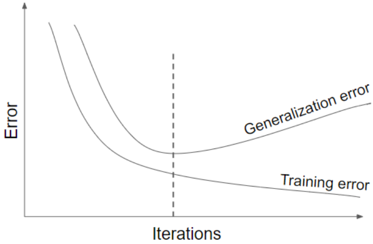


Figure: After a certain number of iterations, the algorithm begins to overfit.

EARLY STOPPING

How early stopping works:

1. Split training data $\mathcal{D}_{\text{train}}$ into $\mathcal{D}_{\text{subtrain}}$ and $\mathcal{D}_{\text{val}}$ (e.g. with a ratio of 2:1).
2. Train on $\mathcal{D}_{\text{subtrain}}$ and evaluate model using the validation set $\mathcal{D}_{\text{val}}$.
3. Stop training when validation error stops decreasing (after a range of "patience" steps).
4. Use parameters of the previous step for the actual model.

More sophisticated forms also apply cross-validation.

EARLY STOPPING

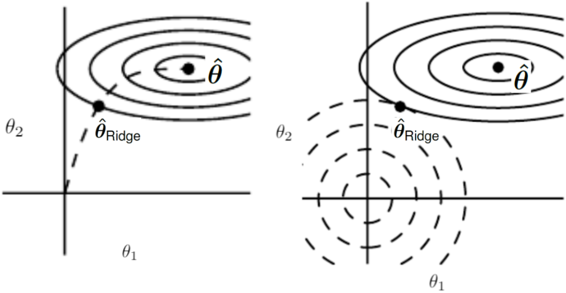
Strengths	Weaknesses
Effective and simple	Periodical evaluation of validation error
Applicable to almost any model without adjustment	Temporary copy of $\boldsymbol{\theta}$ (we have to save the whole model each time validation error improves)
Combinable with other regularization methods	Less data for training $\rightarrow$ include $\mathcal{D}_{\text{val}}$ afterwards

- Relation between optimal early-stopping iteration T_{stop} and weight-decay penalization parameter λ for step-size α (see Goodfellow et al. (2016) page 251-252 for proof):

$$T_{\text{stop}} \approx \frac{1}{\alpha\lambda} \Leftrightarrow \lambda \approx \frac{1}{T_{\text{stop}}\alpha}$$

- Small λ (low penalization) $\Rightarrow$ high T_{stop} (complex model / lots of updates).

EARLY STOPPING



Credit: Goodfellow et al. (2016)

Figure: An illustration of the effect of early stopping. *Left:* The solid contour lines indicate the contours of the negative log-likelihood. The dashed line indicates the trajectory taken by SGD beginning from the origin. Rather than stopping at the point $\hat{\boldsymbol{\theta}}$ that minimizes the risk, early stopping results in the trajectory stopping at an earlier point $\hat{\boldsymbol{\theta}}_{\text{Ridge}}$. *Right:* An illustration of the effect of L_2 regularization for comparison. The dashed circles indicate the contours of the L_2 penalty which causes the minimum of the total cost to lie closer to the origin than the minimum of the unregularized cost.