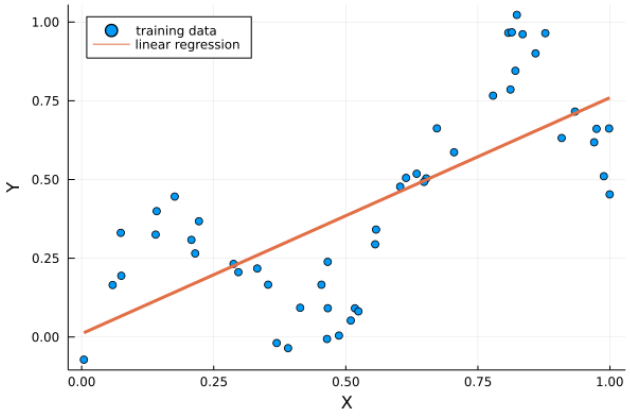
The Flexibility of Models and the Bias-Variance Trade-Off

Johanni Brea

Introduction to Machine Learning

When Linear Methods are Not Flexible Enough



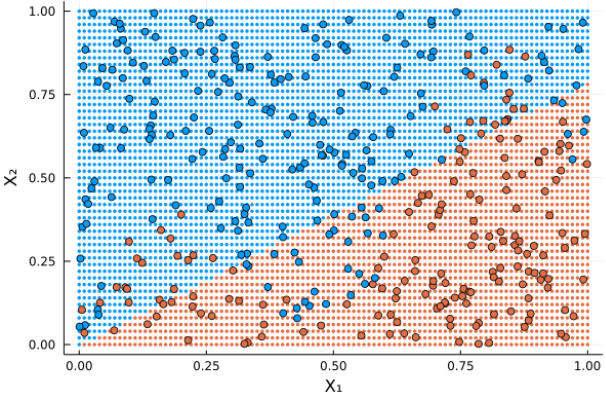
Data Generating Process

$$Y = 0.3\sin(10X) + 0.7X + \epsilon$$

There is structure in the data that linear regression fails to capture.

Linear regression has a high **reducible error**.

When Linear Methods are Not Flexible Enough



large dots: training data
small dots: classification with logistic regression at decision threshold 0.5

Data Generating Process

$Y = \text{red}$ if

$\sigma\big(20(0.3\sin(10X_1) + 0.7X_1 - X_2)\big) > \epsilon$

There is structure in the data that logistic regression fails to capture.

Logistic regression has a high **reducible error**.

Table of Contents

Error Decomposition for Regression

Assume the true conditional data generating process has the form

$$p(y|x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-f(x))^2}{2\sigma^2}}$$

This is also sometimes written as $Y = f(X) + \epsilon$.

We call f the **systematic** information that X provides about Y and ϵ the **noise**.

Let us assume with some supervised machine learning method we find

$$\hat{p}(y|x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-\hat{f}(x))^2}{2\sigma^2}}$$

Let us define the **predicted average response** $\hat{Y} = \hat{f}(X)$.

We would like to know how much the predicted average response $\hat{Y}$ differs in expectation from test data.

Blackboard: Error Decomposition for Regression

$$\mathrm{E}_{Y|X=x}(Y-\hat{Y})^2 = \underbrace{\left(f(x)-\hat{f}(x)\right)^2}_{\text{Reducible}} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{Irreducible}}$$

Quiz

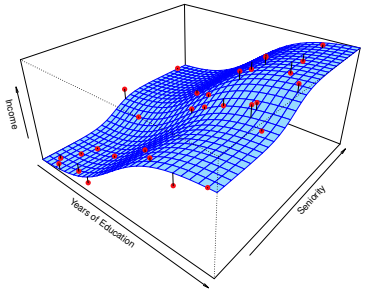
- Let us assume the red data points in this figure were generated with the help of a function whose graph is shown in blue. What is correct?

  A. A linear fit has a non-zero reducible error.
  B. The irreducible error is zero.
  C. A method that perfectly fits the red data points (zero training error) has a reducible error of zero.

- Assume we perform linear regression on the red data points and compute the residuals $\hat{\epsilon}_i = y_i - \hat{y}_i$ and the empirical variance $\frac{1}{n-1}\sum_{i=1}^{n}\hat{\epsilon}^2$. The irreducible error is

  A. larger B. smaller

  than this empirical variance.



- What is the irreducible error for the following data generating process?

$$p(y|x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(y-2x)^2}{2}}$$

A. 1 B. 2 C. 4

Summary

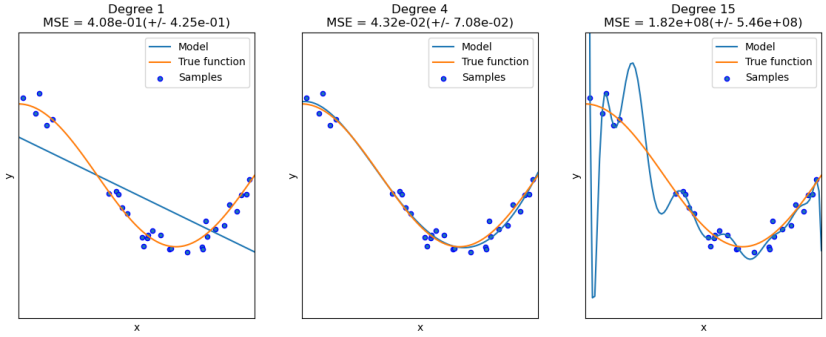
- The true systematic information f that X provides about Y is usually unknown.
- Our goal: find the function $\hat{f}$ that minimizes the reducible error.
- The test loss of $\hat{f}$ for a given data generating process is never lower than its irreducible error, i.e. $\mathrm{E}_{Y|X}(Y - \hat{f}(x))^2 \geq \mathrm{Var}(\epsilon)$.
  Note that the estimate of the true test loss with a test set (often just called "test loss") can by smaller than the irreducible error, by chance.

Table of Contents

Polynomial Regression

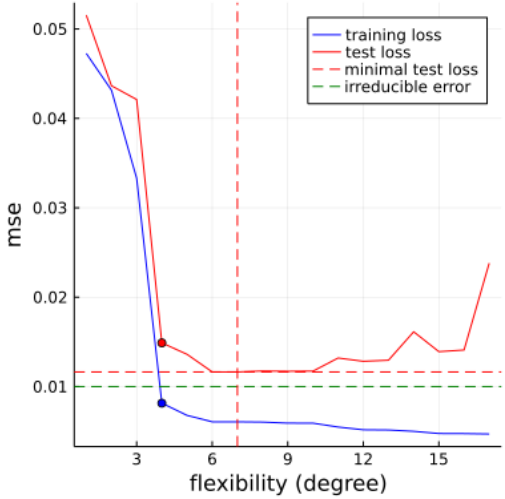
Finding the parameters of a polynomial $f(X) = \theta_0 + \theta_1 X + \theta_2 X^2 + \cdots \theta_d X^d$
is equivalent to finding the parameters in linear regression with polynomial features

$$X_1 = X, X_2 = X^2, \ldots, X_d = X^d$$



https://scikit-learn.org/1.3/auto_examples/model_selection/plot_underfitting_overfitting.html

Polynomial Regression

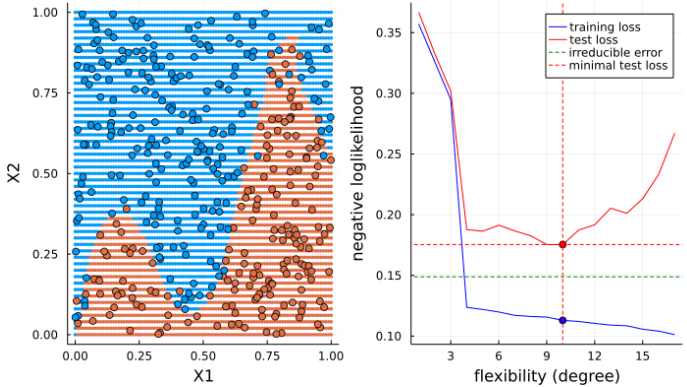


- Polynomial regression with degree 1 is linear regression.
- For small degrees (<6) training loss and test loss are high.
- Minimal test loss is reached for intermediate degrees; it is only slightly higher than the irreducible error.
- For high degrees (>10) the method is too flexible; it can fit peculiarities of the given training set. Therefore the training loss is lowest there, but the test loss is higher.

Multiple Polynomial Classification

$$f(X) = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \theta_3 X_1 X_2 + \theta_4 X_1^2 + \theta_5 X_2^2 + \cdots + \theta_{(d+1)(d+2)/2} X_2^d$$

(all powers up to d'th order)



Summary

- Polynomial regression is like multiple regression with the additional features being higher orders of the original feature(s).
- For p predictors and d degrees there are $\binom{p+d}{p}$ parameters [1]. This number grows quickly in p and d, making polynomial regression unpractical for high dimensional input data. For example a fourth order polynomial in 20 dimensions has $\binom{24}{20} = 10625$ parameters.
- The degree is a **hyper-parameter** that controls the flexibility of the method.
- The training loss decreases with increasing flexibility (degree).
- The test loss is U-shaped as a function of the flexibility.
- The lowest test loss is not lower than the irreducible error.

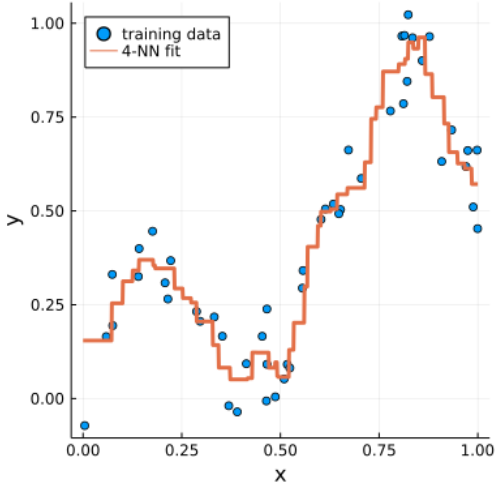
[1] Distribute d balls into $p+1$ urns: all terms of the form $1^{d_0} x_1^{d_1} \cdots x_p^{d_p}$ s.t. $\sum_{i=0}^{p} d_i = d$

Table of Contents

kNN Regression

$$\hat{y} = \frac{1}{K} \sum_{i \in \mathcal{N}_0} y_i$$

where $\mathcal{N}_0$ is the set of k nearest neighbours of x_0.

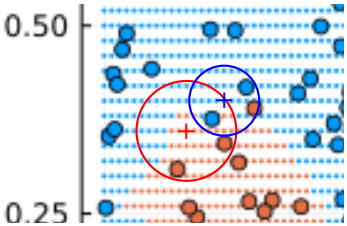


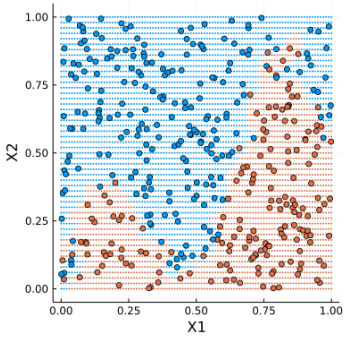
kNN Classification

$$\widehat{\Pr}(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j)$$

where $\mathcal{N}_0$ is the set of k nearest neighbours of x_0.

$k = 3$





Properties of kNN

- KNN is a non-linear method.
- kNN is an example of a **non-parametric method**, where the raw training data is used to make predictions.
  This is in contrast to **parametric methods** (like linear regression), where all information about the training data is stored in the parameters θ.
- The number of neighbors k is a **hyper-parameter**; it controls the assumed smoothness of the function family and does not depend on the training data.
- If the hyper-parameter k is small, the function family is **flexible**, which allows to follow the training data accurately. If k is large the function family is **inflexible** and follows only the main trend of the data.
- kNN is very fast to fit (no parameters are learnt).
- kNN is slow to make predictions, because the neighbors of the test point need to be searched in the training set.

kNN Classification on MNIST

First nearest neighbors classification on MNIST reaches a misclassification rate of approximately 3%.

This is better than linear classification.

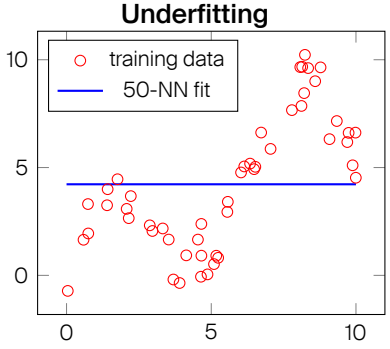
Table of Contents

Curse of Dimensionality

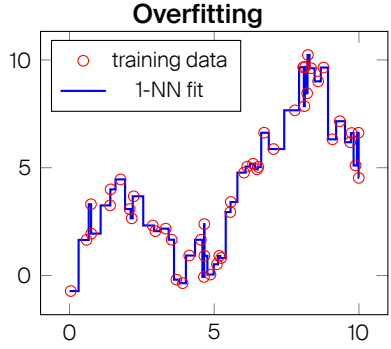
Things that work in low-dimensional spaces, may require A LOT OF data (or parameters) in high-dimensional spaces. Example: if we want to have 10 data points per dimension for accurate 3-NN regression, we need 10^d data points in d dimensions.

Table of Contents

Underfitting and Overfitting



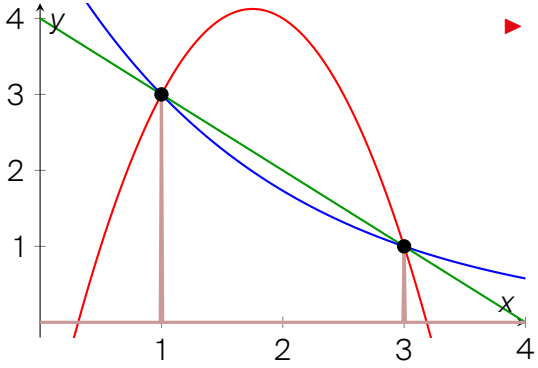
the function family is not flexible enough.
training and test error are high.



the function family is too flexible.
training error is low, test error is high.

Hyper-parameters control the flexibility (k for kNN, degree for polynomial regression).

Inductive Bias



- Inductive bias: set of assumptions used to predict outputs of unseen inputs
- Generalization to the test set will be good
  - for the parabola, if the data comes from ballistic motion (x: position, y: position)
  - for the exponential, if the data comes from radioactive decay (x: time, y: activity)
  - for the line, if the data comes from measuring the temperature (y) as a function of the altitude (x).
  - for the "spike curve", if the data comes from neural recordings (x: time, y: number of action potentials in 20ms bins)

Inductive Bias

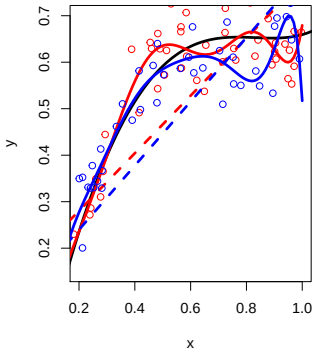
Given a finite amount of training data, we can be

- extremely lucky, and find the true data generator (zero reducible error), if we chose the right inductive bias
- very unlucky, if the true data generator is far away from anything reachable with a given inductive bias (e.g. inductive bias: smooth functions, true data generator: x^2 if x is a rational number, –100 otherwise).

Current wisdom: generalization on real-world data works well, if we use a lot of data and a flexible method that favors simplicity.

Table of Contents

Fitting Multiple Training Sets



- Red and blue data points generated with
  $Y = f(X) + \epsilon \quad$ with $\text{Var}(\epsilon) = 0.06^2$
  and $f(X) = \sin(2X) + 2(X - 0.5)^3 - 0.5X$ (black line)
- Red/blue lines: fits on red/blue training set
- The linear fits (dashed lines) are close to each other (small variance) but far away from f (large bias).
- The polynomial fits (with $d = 10$) are far from each other (large variance) but close to f (small bias).

The Bias-Variance Trade-Off

Expected test MSE

$$E_{Y,\hat{f}|X}(Y - \hat{f}(X))^2 = [\text{Bias}(\hat{f}(X))]^2 + \text{Var}(\hat{f}(X)) + \text{Var}(\epsilon)$$

- Here, $\text{Var}(\hat{f}(X))$ refers to the variance of the estimator, if it would be fitted multiple times to each time another training set.
- Generally: flexible methods have higher variance but lower bias than less flexible methods.

Blackboard: Bias-Variance Decomposition

$$E_{Y|X,\hat{f}}\left(Y-\hat{f}(X)\right)^2 = E_{\varepsilon,\hat{f}}\left(f(X)+\varepsilon-\hat{f}(X)\right)^2$$

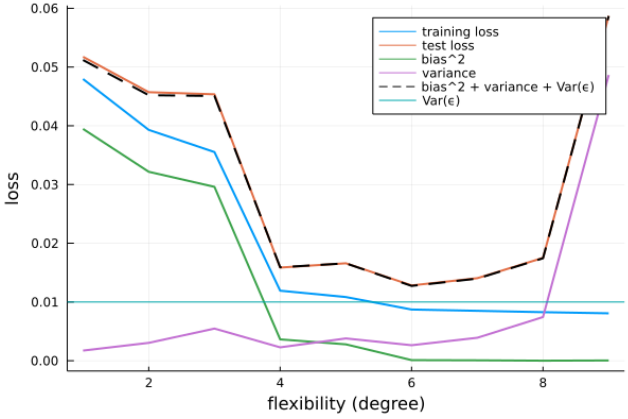
$$= \underbrace{E_{\hat{f}}\left(f(X)-\hat{f}(X)\right)^2}_{\text{expected reducible error}} + \underbrace{Var(\varepsilon)}_{\text{irreducible error}}$$

$$E_{\hat{f}}\left(f(X)-\hat{f}(X)\right)^2 = E_{\hat{f}}\left(f(X)-E_{\hat{f}}(\hat{f}(X))+E_{\hat{f}}(\hat{f}(X))-\hat{f}(X)\right)^2$$

$$= \underbrace{\left(f(X)-E_{\hat{f}}(\hat{f}(X))\right)^2}_{\text{squared bias}} + \underbrace{Var\,\hat{f}(X)}_{\text{variance}}$$

$$2\left(f(x)-E_{\hat{f}}(\hat{f}(x))\right)\left(E_{\hat{f}}(\hat{f}(x))-E_{\hat{f}}(\hat{f}(x))\right)=0$$

The Bias-Variance Trade-Off



For 1000 different training sets of size $n = 50$, polynomial fits of degrees 1 to 9 where performed. Each fit was evaluated on a large test set.

The plot shows the average training and test loss (over all 1000 training sets), and the average squared bias and variance (over all test points).

Note that the squared bias, the variance and the irreducible error sum up to the test loss.

Summary

- Rigid (inflexible) methods
  - may underfit the data.
  - may have a large bias.
  - tend to have little variance when fitted on different training sets.
- Flexible methods
  - may overfit the data.
  - can achieve a low bias.
  - tend to have high variance when fitted on different training sets.

In expectation over training sets and responses, the test MSE of a method is given by the sum of the method's variance, its squared bias and the irreducible error.

Side remark: Thanks to their simplicity, polynomial regression and KNN are great for educational purposes, but they are rarely used in modern applications of machine learning. For successful choices of more advanced machine learning methods it is, however, crucial to understand the flexibility of a method, the bias-variance decomposition and overfitting and underfitting.

Table of Contents

Making Linear Models Less Flexible

Idea 1: Fix some parameters at zero

$$\hat{y} = f(x) = f(x_1, x_2, \ldots, x_p) = \beta_0 + \underbrace{\beta_1}_{=0} x_1 + \underbrace{\beta_2}_{=0} x_2 + \beta_3 x_3 + \cdots + \underbrace{\beta_{p-1}}_{=0} x_{p-1} + \beta_p x_p$$

Problem: Many different models to fit; $\binom{p+1}{m}$ combinations of m non-fixed parameters.

Idea 2: constrain the parameters

Minimize the original loss $L(\beta)$ under the constraint $\|\beta\|_2^2 = \sum_{i=1}^{p} \beta_i^2 \leq S$.

This is equivalent to replacing the original loss $L(\beta)$ by

$$L_{\text{L2}}(\beta) = L(\beta) + \lambda \|\beta\|_2^2$$

Note: idea 2 does not put the parameters to zero. Instead, it decreases the flexibility by restricting the space of possible parameter values.

Ridge Regression (L2 Regularization)

$$L_{L2}(\theta) = L(\theta) + \lambda \|\theta\|_2^2$$

with **regularization constant** λ and (squared) **L2 norm** $\|\theta\|_2^2 = \sum_{i=1}^{p} \theta_i^2$.

1. The regularization constant λ is a hyper-parameter.
2. Often the intercept θ_0 is not regularized.
3. If $\lambda = 0$: original loss (no penalty)
4. The larger λ, the stronger the impact of the penalty on the result.
5. With increasing λ the model becomes less flexible.
6. With increasing λ all parameters tend to zero; it happens rarely that one is exactly zero.

Lasso (L1 Regularization)

$$L_{\text{L1}}(\theta) = L(\theta) + \lambda \|\theta\|_1$$

with **regularization constant** λ and **L1 norm** $\|\theta\|_1 = \sum_{i=1}^{p} |\theta_i|$.

Points 1-5 from ridge regression are also valid for the Lasso. However:

6. With large λ some parameters are exactly zero (in contrast to ridge regression).

Analytical Solutions for Simple Linear Regression

Notation: $\langle x \rangle = \frac{1}{n} \sum_{i=1}^{n} x_i$

Ridge Regression

$$L(\theta, \lambda) = \langle (y - \theta_0 - \theta_1 x)^2 \rangle + \lambda \theta_1^2$$

$$\theta_1 = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\langle x^2 \rangle - \langle x \rangle^2 + \lambda}, \qquad \theta_0 = \langle y \rangle - \theta_1 \langle x \rangle$$

Lasso

$$L(\theta, \lambda) = \frac{1}{2} \langle (y - \theta_0 - \theta_1 x)^2 \rangle + \lambda |\theta_1|$$

$$\theta_1 = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle - \text{sign}(\theta_1)\lambda}{\langle x^2 \rangle - \langle x \rangle^2} \text{ or 0 if } |\langle xy \rangle - \langle x \rangle \langle y \rangle| < \lambda$$

An Alternative Formulation of Regularization

Thanks to a result from constraint optimization (see Karush-Kuhn-Tucker conditions, a generalization of Lagrange multipliers) the above formulations of Ridge Regression and the Lasso are equivalent to a constraint optimization problem:

Ridge Regression

minimize $L(\theta)$ under the constraint that $\|\theta\|_2^2 \leq S$.

The parameters are confined to a p-ball of radius S with center at the origin.

Lasso

minimize $L(\theta)$ under the constraint that $\|\theta\|_1 \leq S$.

The parameters are confined to a hypercube with edge length S, center at the origin and corners on the axes.

S is a (complicated) function of λ and the original loss $L(\theta)$.
With increasing S the model becomes more flexible.

Regularization Is Not Invariant Under Rescaling

Linear Regression is invariant

original			scaled		
X_1	X_2	Y	X_1	X_2	Y
1	2	4	10	2	4
2	1	3	20	1	3
4	0	7	40	0	7

solution for original data:
$\hat{\theta}_0 = -13, \hat{\theta}_1 = 5, \hat{\theta}_2 = 6$
prediction for $X_1 = 2, X_2 = 3$: 15

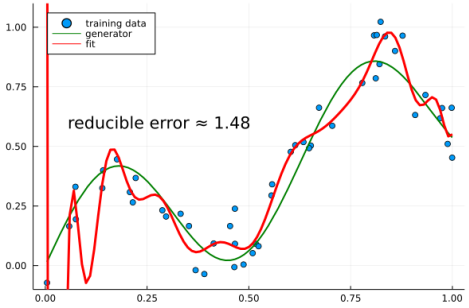
solution for scaled data:
$\hat{\theta}_0 = -13, \hat{\theta}_1 = 0.5, \hat{\theta}_2 = 6$
prediction for $X_1 = 20, X_2 = 3$: 15

Regularization is not invariant
solution for original data with $\lambda = 10^{-3}$:
$\hat{\theta}_0 = -11.76, \hat{\theta}_1 = 4.65, \hat{\theta}_2 = 5.47$
prediction for $X_1 = 2, X_2 = 3$: 14.06

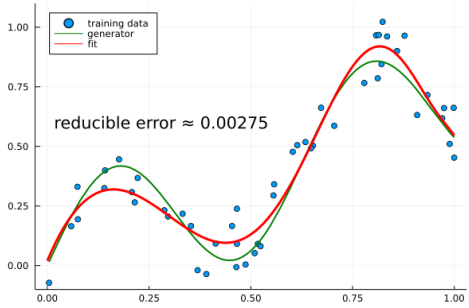
solution for scaled data with $\lambda = 10^{-3}$:
$\hat{\theta}_0 = -11.76, \hat{\theta}_1 = 0.47, \hat{\theta}_2 = 5.5$
prediction for $X_1 = 20, X_2 = 3$: 14.13

Polynomial Ridge Regression

$d = 20, \lambda = 0$



$d = 20, \lambda = 10^{-4}$



With a little bit of L2 regularization ($\lambda = 10^{-4}$)
one can prevent overfitting of polynomials with high degrees.

Multiple Logistic Ridge Regression on the Spam Data

$n = 2000$ emails, $p = 801$ features (size of the lexicon)

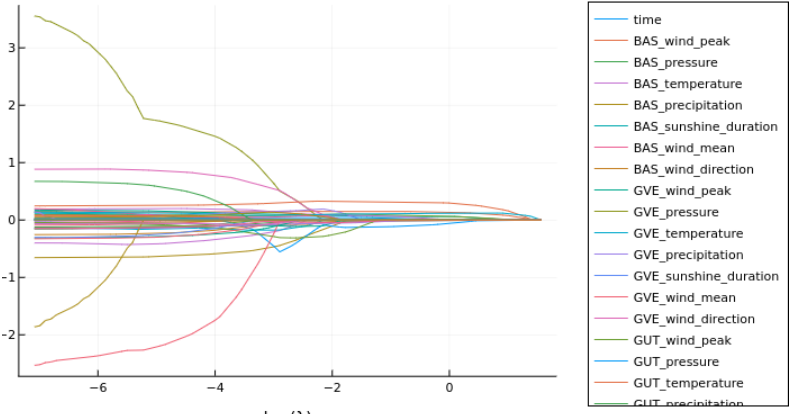
Without regularization

training misclassification rate: 0.0015
test misclassification rate: 0.048

With L2 regularization

training misclassification rate: 0.013
test misclassification rate: 0.041

The Lasso Path for the Weather Data



The Lasso path is useful to identify the most important predictors.

There are specialized efficient methods to compute the Lasso path.

As we lower λ (from right to left), `BER_wind_peak` is the first non-zero factor, `BAS_wind_peak` the second and `LUZ_wind_mean` the third.

Summary

- Regularization allows to lower the flexibility of a model by restricting the parameters to certain areas of the parameter space.
- L1 regularization leads to sparse models with some parameters exactly zero $\Rightarrow$ great for interpretability.

Quiz

- The Lasso tends to have larger variance (when fitted on different training sets from the same data generator) but smaller bias (relative to the true data generator) than linear regression.
- Indicate which is correct: as we increase S from 0 to a large value in L2 regularized linear regression the training error will be

  A) inverted U shape. B) U shape.
  C) steadily increasing. D) steadily decreasing. E) constant.
- Indicate which is correct: as we increase S from 0 to a large value in L2 regularized linear regression the test error will be

  A) inverted U shape. B) U shape.
  C) steadily increasing. D) steadily decreasing. E) constant.

Suggested Reading

- 2.1 What is Statistical Learning
- 2.2 Assessing Model Accuracy
- 7.1 Polynomial Regression
- 3.5 Comparison of Linear Regression with K-Nearest Neighbors
- 6.2 Shrinkage Methods
- 6.4 Considerations in High Dimensions