LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Introduction to Machine Learning

Regularization

Department of Statistics – LMU Munich

Motivation

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

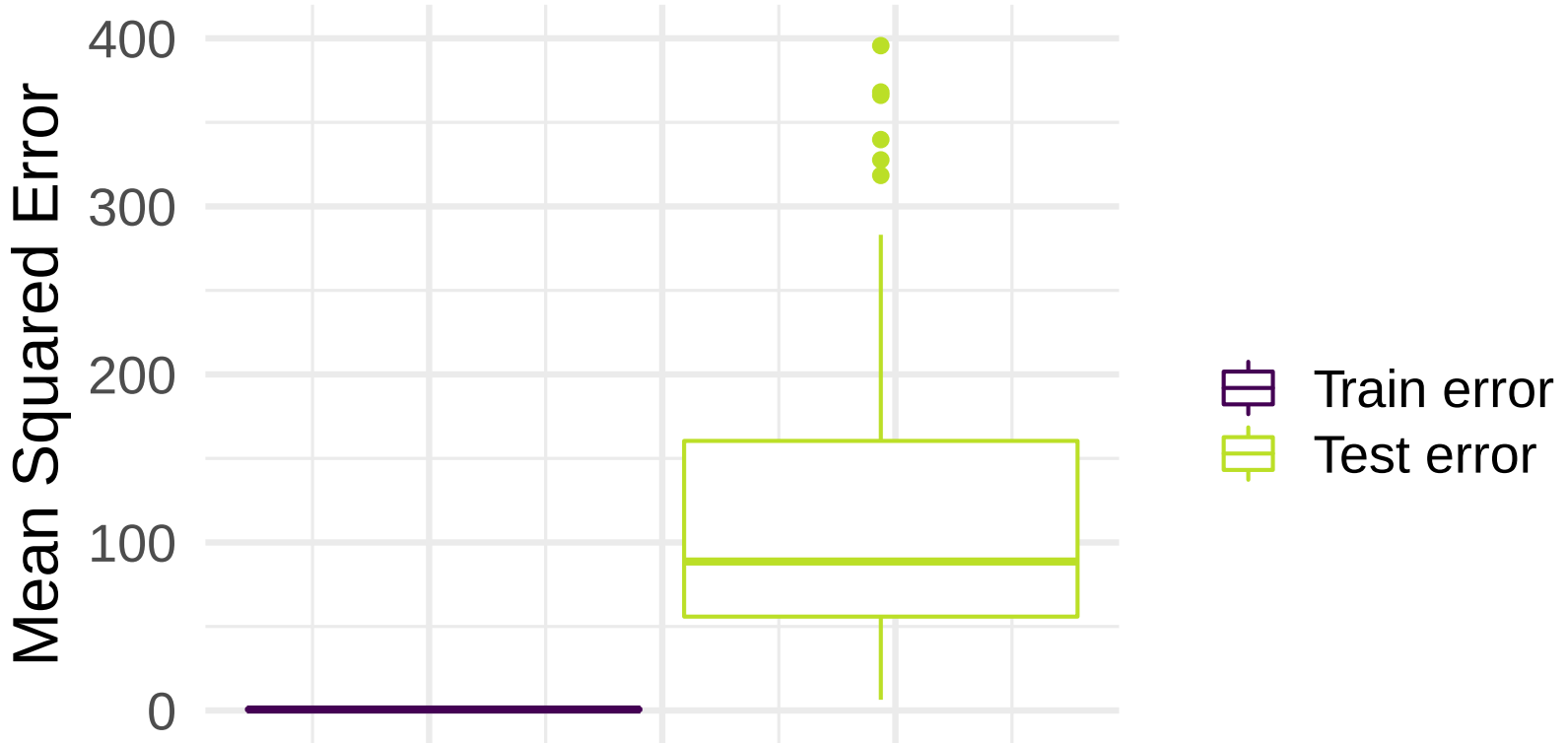
$$f(\mathbf{x} \mid \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{x} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_{12} x_{12}$$

  with the $L2$-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

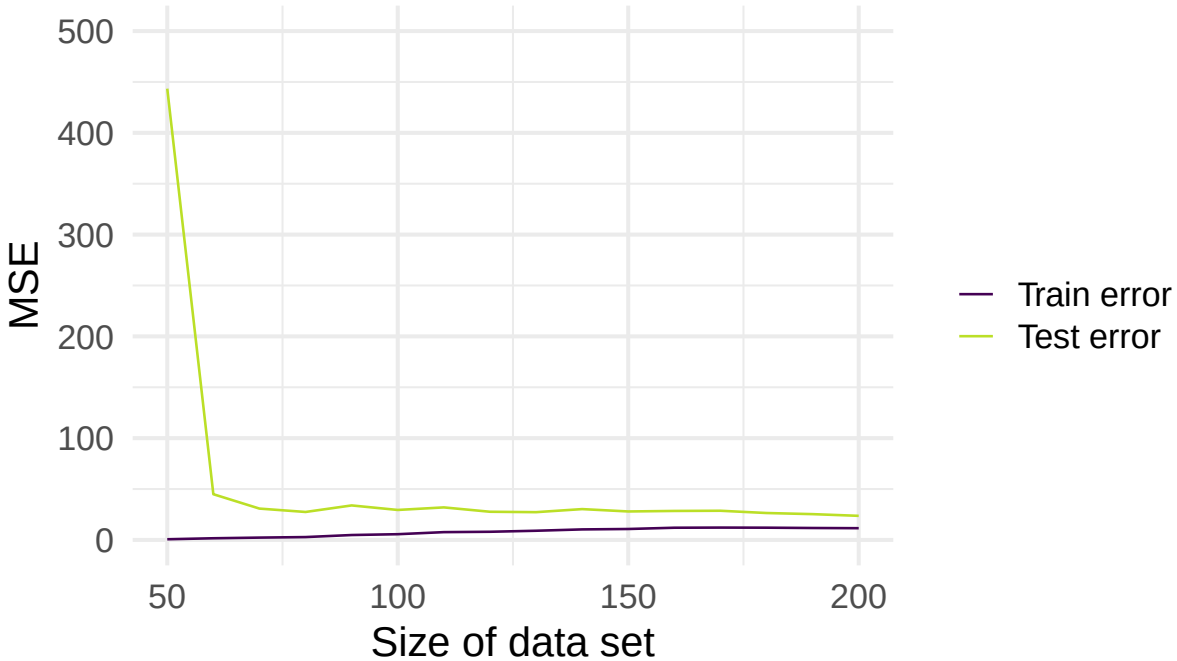
Why can **overfitting** happen? And how to avoid it?

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 the model's performance for increased data set size by 10 times 10-fold CV. The fit worsens slightly, but the test error decreases.



Good idea, but often not feasible in practice.

Note: We initially only used 50 out of 200 train observations to simulate the collection of more data.

AVOID OVERFITTING

Approach 2: Reduce **complexity**

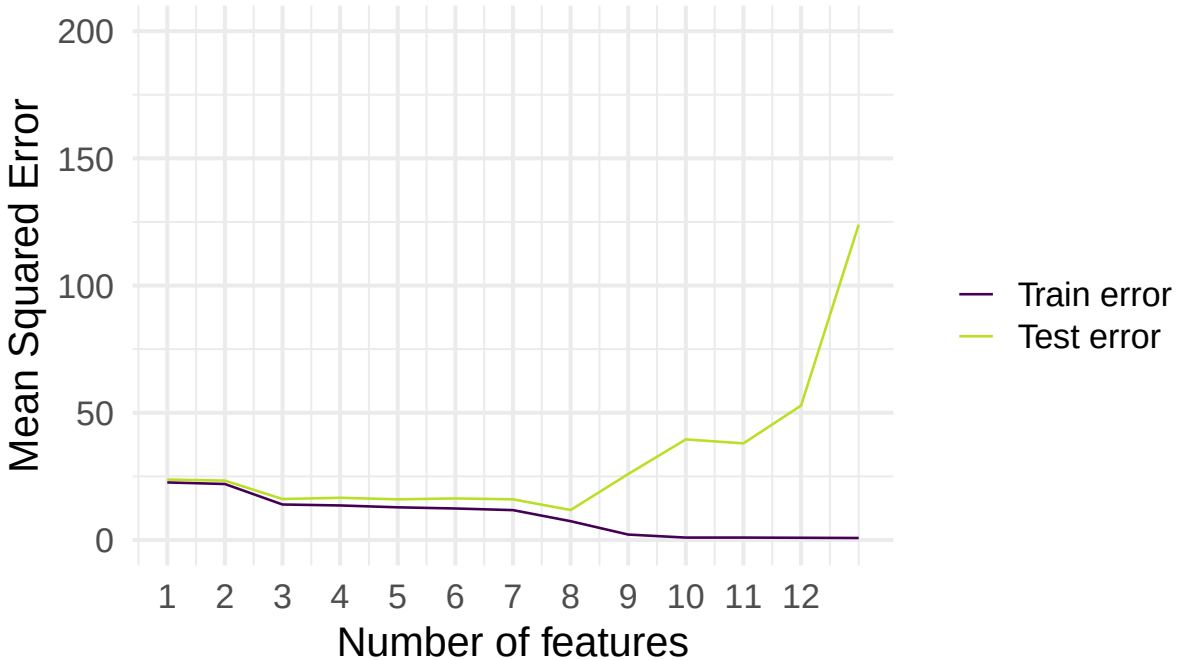
We try the simplest model we can think of: the constant model. For the *L2*-loss, the optimal constant model is mean over all $y^{(i)}$:

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

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 order - but there are $2^{12} = 4096$ potential feature combinations.

AVOID OVERFITTING

We have contradictory goals

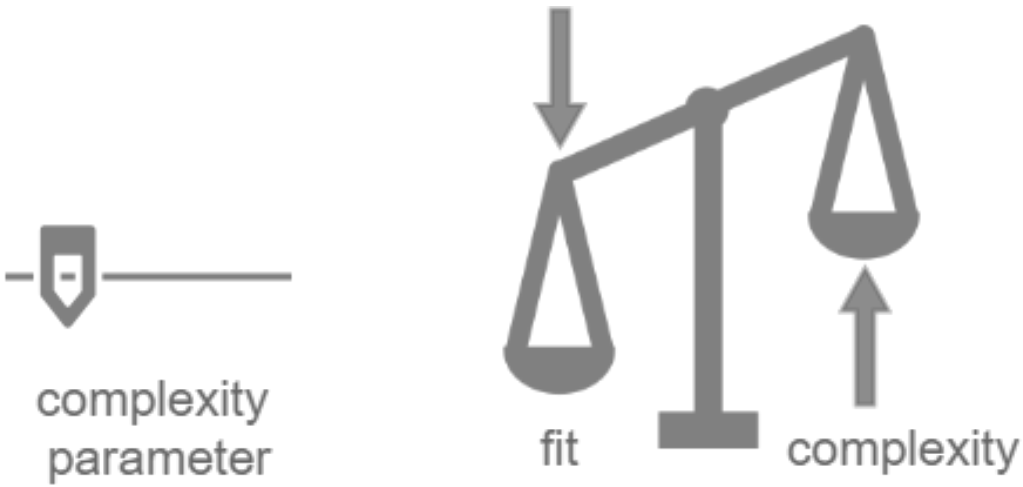
- **maximizing the fit** (minimize the empirical risk $\mathcal{R}_{\text{emp}}(f)$)
- **minimizing the complexity** of the model (e.g. min #features)

We need to find the “sweet spot”.

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:

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

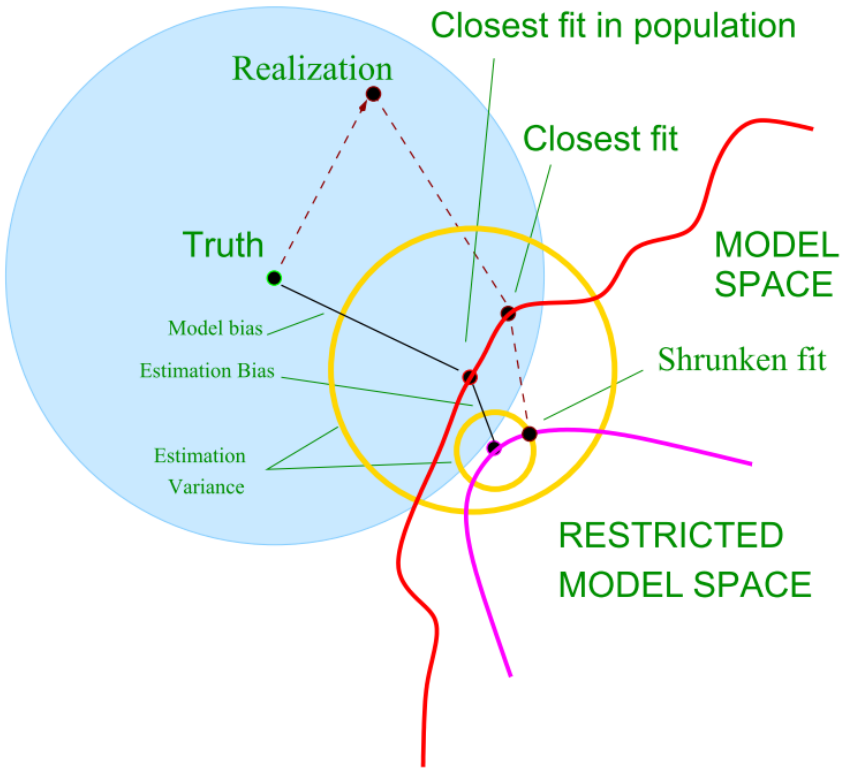
- $J(f)$ is called **complexity penalty**, **roughness penalty** or **regularizer**.
- It measures the "complexity" of a model and penalizes it in the fit.
- As for $\mathcal{R}_{\text{emp}}$, often $\mathcal{R}_{\text{reg}}$ and J are defined on $\boldsymbol{\theta}$ instead of f, so $\mathcal{R}_{\text{reg}}(\boldsymbol{\theta})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

REGULARIZED EMPIRICAL RISK MINIMIZATION

Remark:

- 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

Regularization in Linear Modeling

REGULARIZATION IN THE LINEAR MODEL

Recall, for the (unregularized) linear model we optimize

$$\mathcal{R}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \left(y^{(i)} - \boldsymbol{\theta}^\top \mathbf{x} \right)^2 .$$

Recall that the problem has a closed form solution:

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X} \mathbf{y}.$$

- Coefficient estimates for ordinary least squares with $f(\mathbf{x} \mid \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{x}$ usually require a full rank design matrix $\mathbf{X}$, otherwise $\mathbf{X}^\top \mathbf{X}$ is not invertible.
- When features are highly correlated (i.e. $\mathbf{X}$ is close to "being not a full rank matrix"), the least-squares estimate becomes highly sensitive to random errors in the observed response, producing a large variance in the fit.

RIDGE REGRESSION

Ridge regression is a **shrinkage method**, as it shrinks the regression coefficients θ towards 0, by putting an L_2-penalty$^{(*)}$ on it:

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

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

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

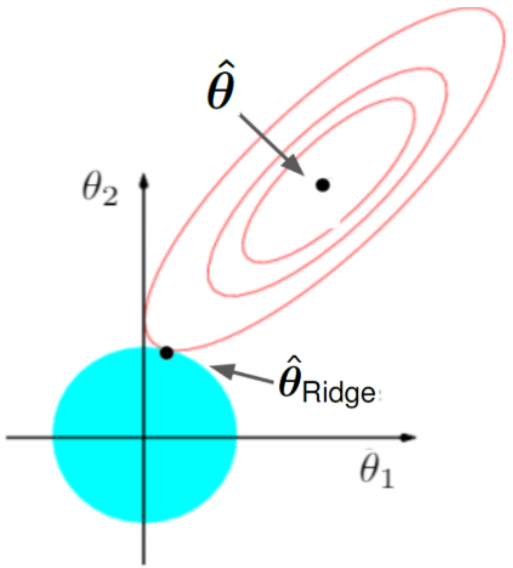
$^{(*)}$ $J(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_2^2 = \theta_1^2 + \theta_2^2 + ... + \theta_p^2$

RIDGE REGRESSION

The problem can be formulated equivalently as a constrained optimization problem (for an appropriate choice of t):

$$\min_{\boldsymbol{\theta}} \quad \sum_{i=1}^{n} \left(y^{(i)} - \boldsymbol{\theta}^{\top} \right)^2$$

$$\text{subject to:} \quad \|\boldsymbol{\theta}\|_2^2 \leq t$$

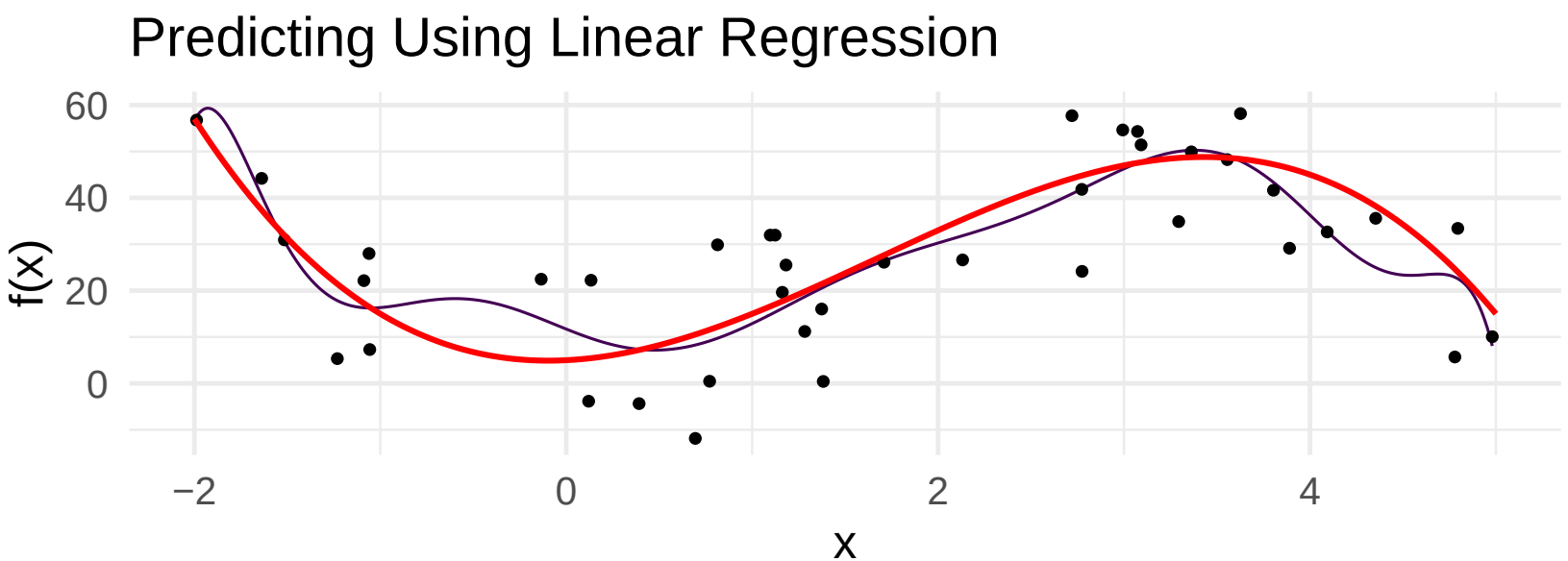


EXAMPLE: POLYNOMIAL RIDGE REGRESSION

Again, let us consider a dth-order polynomial

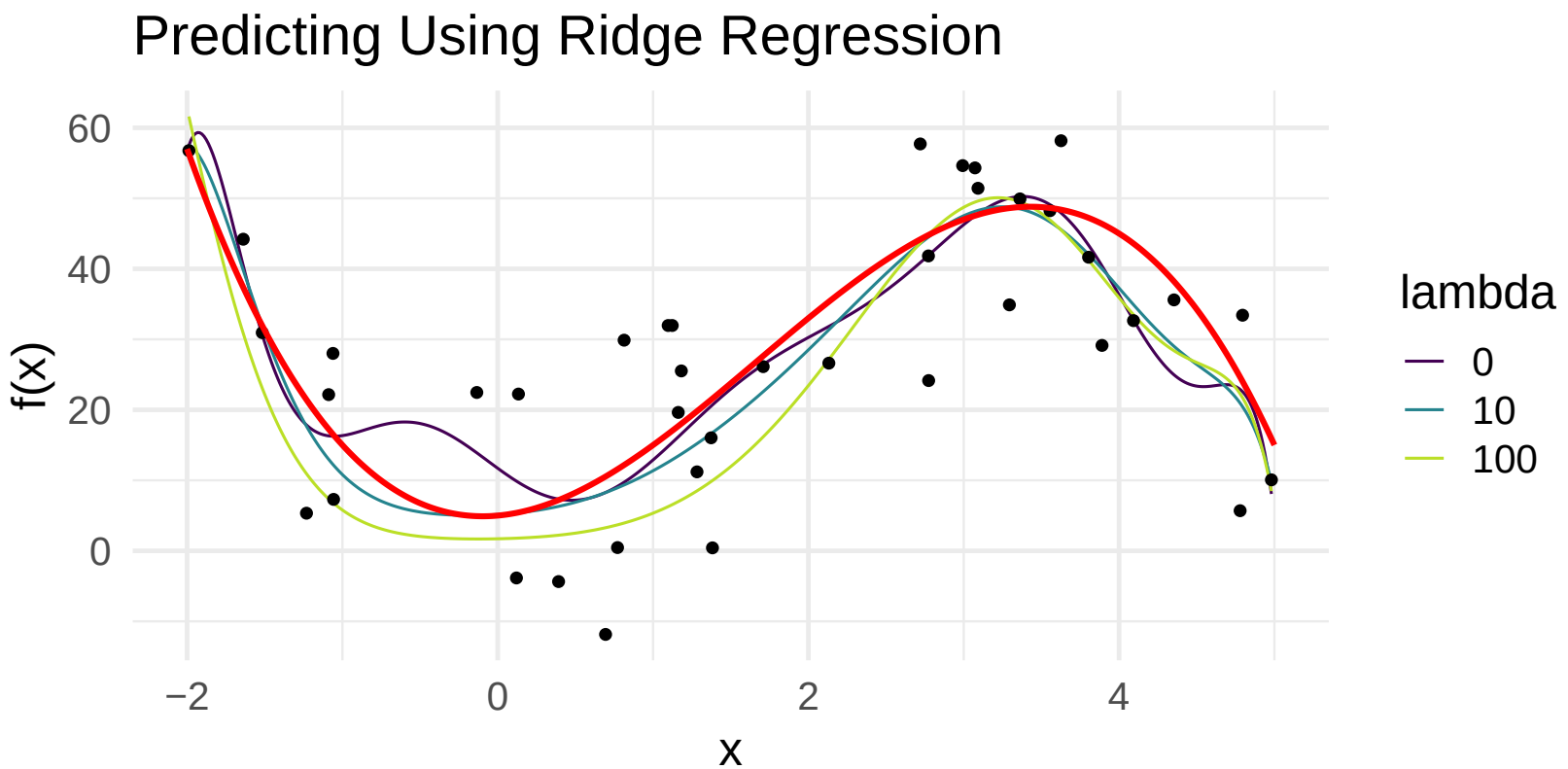
$$f(x) = \theta_0 + \theta_1 x + \cdots + \theta_d x^d = \sum_{j=0}^{d} \theta_j x^j.$$

True relationship is $f(x) = 5 + 2x + 10x^2 - 2x^3 + \epsilon$
Making predictions for $d = 10$ with linear regression tends to overfit:



EXAMPLE: POLYNOMIAL RIDGE REGRESSION

Using Ridge Regression with different penalty terms approximates the real data generating process better:



LASSO REGRESSION

Another shrinkage method is the so-called **Lasso regression**, which uses a L_1 penalty on $\boldsymbol{\theta}$ $^{(*)}$:

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

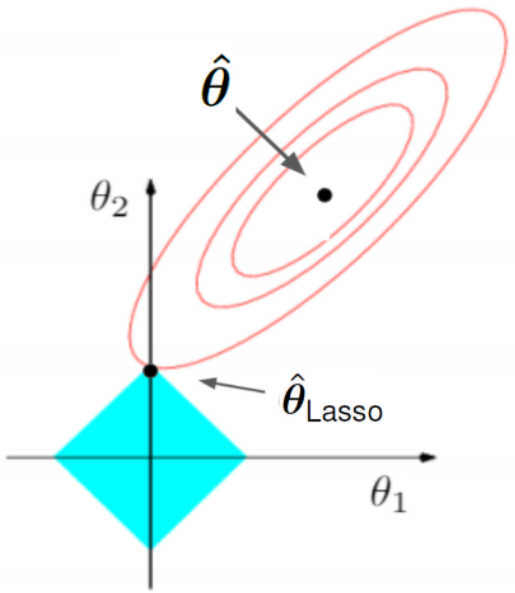
Note that optimization now becomes much harder. $\mathcal{R}_{\text{reg}}(\boldsymbol{\theta})$ is still convex, but we have moved from an optimization problem with an analytical solution towards an undifferentiable problem.

$^{(*)}$ $J(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_1 = |\theta_1| + |\theta_2| + ... + |\theta_p|$

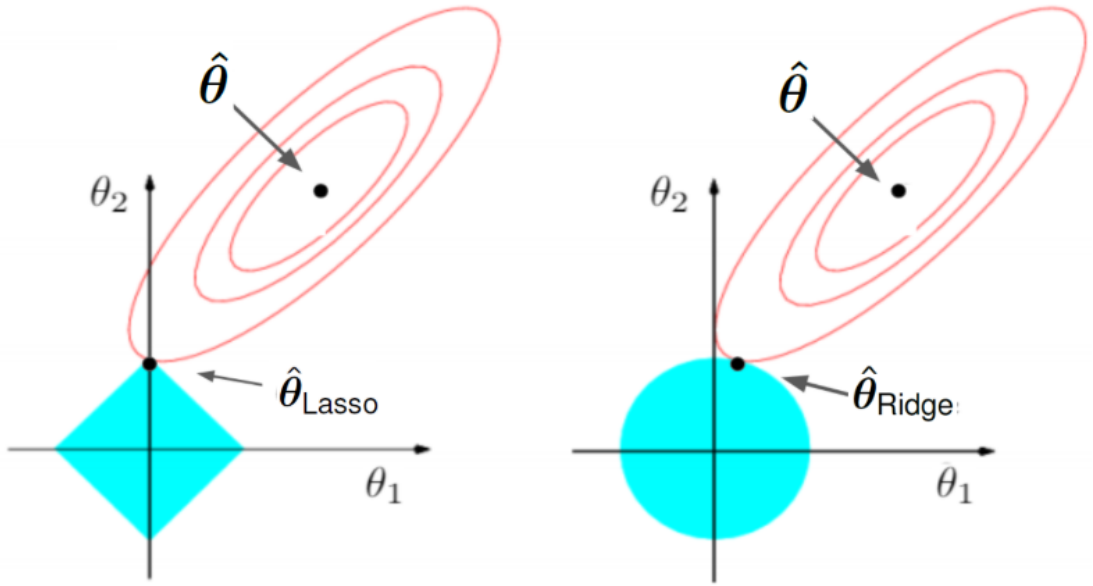
LASSO REGRESSION

Analogously, the problem can be formulated equivalently as a constrained optimization problem (for an appropriate choice of t):

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

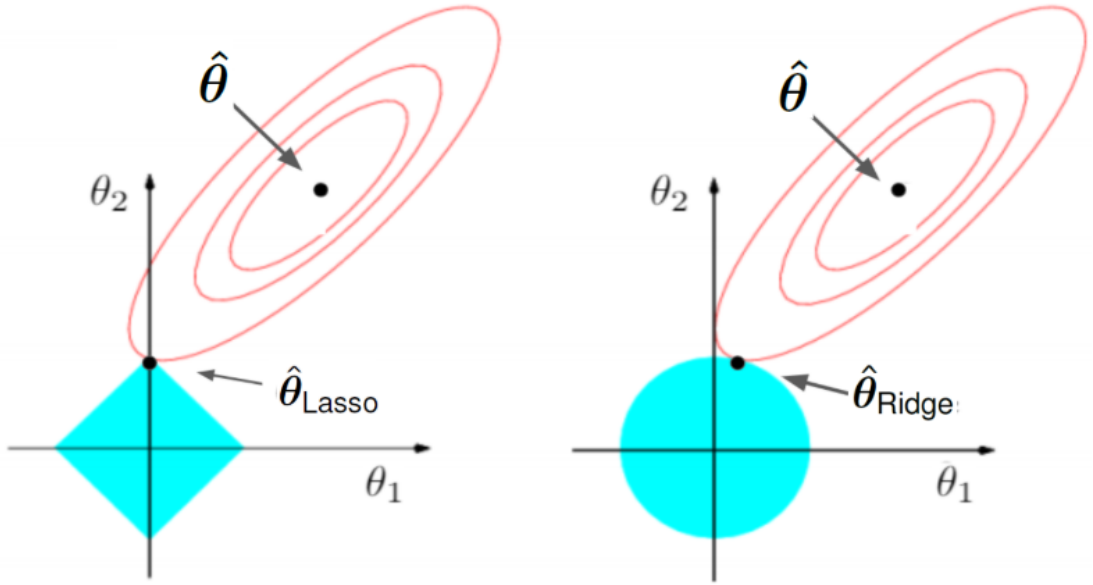


LASSO VS. RIDGE REGRESSION



- The ellipses are the level curves of the unregularized loss $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ and $\hat{\boldsymbol{\theta}}$ is the location of the minimum.
- The regions in cyan represent the values of $\boldsymbol{\theta}$ that satisfy the inequality constraint.
- As λ increases, the area of the region shrinks.

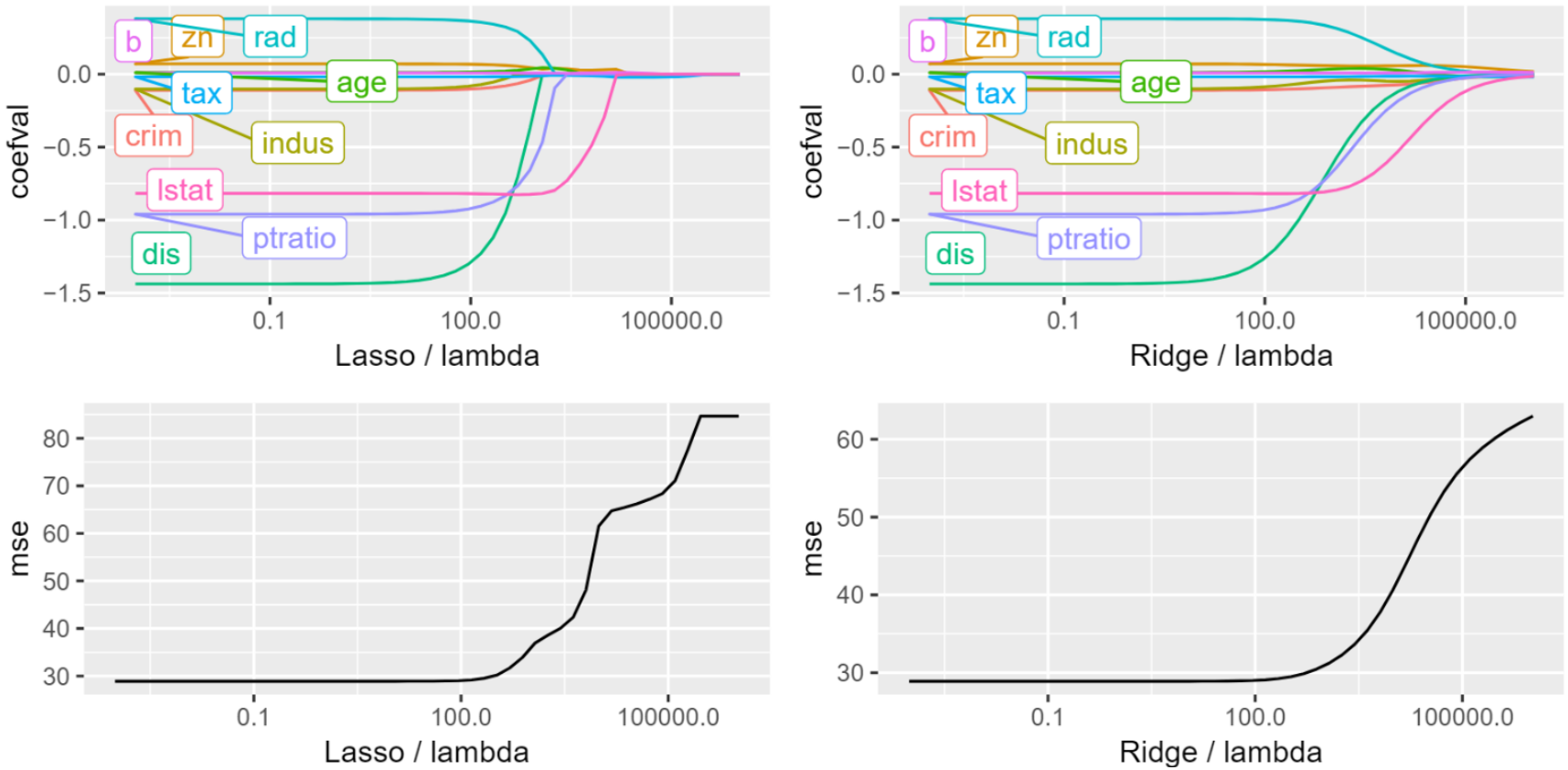
LASSO VS. RIDGE REGRESSION



- In both cases, the solution which minimizes $\mathcal{R}_{\text{reg}}(\boldsymbol{\theta})$ is always a point on the boundary of the feasible region (for sufficiently large λ).
- In the case of Lasso, the solution is more likely to be one of the vertices of the constraint region. This induces sparsity and is a form of variable selection.
- As expected, $\hat{\boldsymbol{\theta}}_{\text{Lasso}}$ and $\hat{\boldsymbol{\theta}}_{\text{Ridge}}$ have smaller parameter norms than $\hat{\boldsymbol{\theta}}$.

LASSO VS. RIDGE REGRESSION

Coefficient paths for different λ values for Ridge and Lasso, on Boston Housing (few features removed for readability) along with the cross-validated MSE. We cannot really overfit here with an unregularized linear model as the task is so low-dimensional.



LASSO VS. RIDGE REGRESSION

Comments regarding Ridge / Lasso implementations:

- Note that very often we do not include θ_0 in the penalty term $J(\boldsymbol{\theta})$ (but this can be implementation-dependent).
- These methods are typically not equivariant under scaling of the inputs, so one usually standardizes the features.

Comments on **Ridge vs. Lasso:**

- Often neither one is overall better.
- Lasso can set some coefficients to zero, thus performing variable selection, while ridge regression usually leads to smaller estimated coefficients, but still dense $\boldsymbol{\theta}$ vectors.

LASSO VS. RIDGE REGRESSION

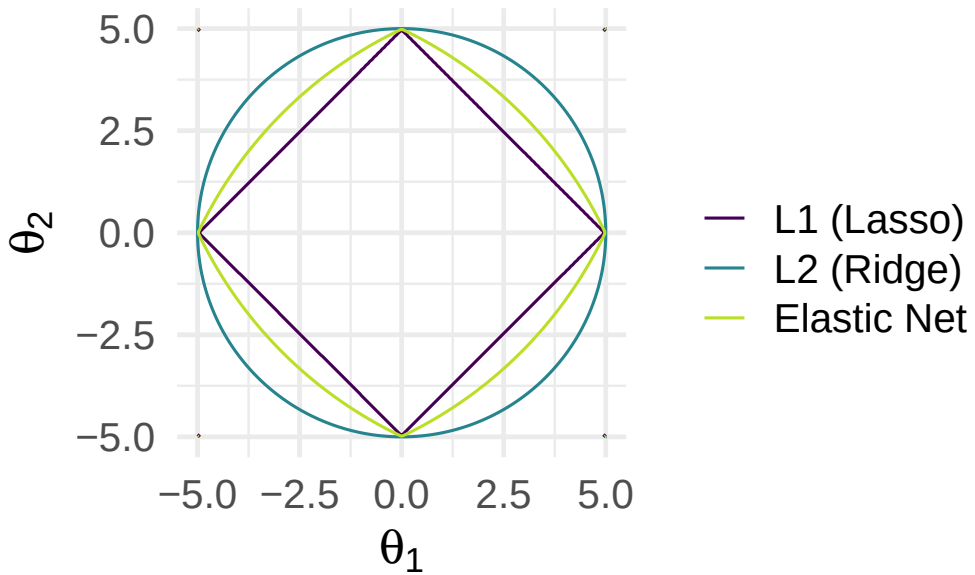
- Both methods can handle correlated predictors, but they solve the multicollinearity issue differently:
  - In ridge regression, the coefficients of correlated predictors are similar
  - In Lasso, one of the correlated predictors has a larger coefficient, while the rest are (nearly) zeroed.
- Lasso tends to do well if there are a small number of significant parameters and the others are close to zero (ergo: when only a few predictors actually influence the response)
- Ridge works well if there are many large parameters of about the same value (ergo: when most predictors impact the response).

Idea: Combine the two!

ELASTIC NET

Elastic Net combines the L_1 and L_2 penalty:

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



REGULARIZED LOGISTIC REGRESSION

We described regularization for the special case of the linear model $f(\mathbf{x} \mid \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{x}$. Of course, regularized risk minimization can be performed for any model $f(\mathbf{x})$.

We can, for example, also construct L_1 or L_2 penalized **logistic regression** to enable coefficient shrinkage and variable selection in this model.

We can add a regularizer to the risk of logistic regression

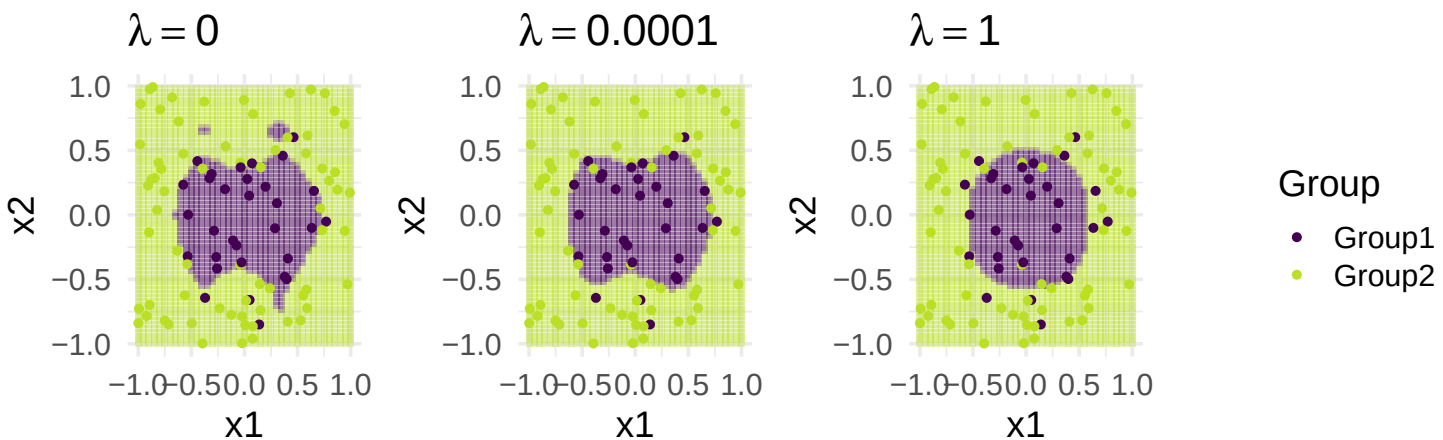
$$\begin{aligned}\mathcal{R}_{\text{reg}}(\boldsymbol{\theta}) &= \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) + \lambda \cdot J(\boldsymbol{\theta}) \\ &= \sum_{i=1}^{n} \log\left[1 + \exp\left(-2y^{(i)} f\left(\mathbf{x}^{(i)} \,\middle|\, \boldsymbol{\theta}\right)\right)\right] + \lambda \cdot J(\boldsymbol{\theta})\end{aligned}$$

Instead of $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ we minimize $\mathcal{R}_{\text{reg}}(\boldsymbol{\theta})$ (both have no closed-form solution, numerical optimization methods are necessary).

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 a L2 penalty. We see

- $\lambda = 0$: the unregularized model seems to overfit
- $\lambda = 0.0001$: regularization helps to learn the underlying mechanism
- $\lambda = 1$ displays the real data generating process very well



SUMMARY: REGULARIZED RISK MINIMIZATION

The principle of regularization can be used **independent** of the model $f(\mathbf{x})$ being used. In $\mathcal{R}_{\text{reg}}$ one has extreme flexibility to make appropriate choices

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

for a given ML problem:

- the **representation** of f, which determines how features can influence the predicted y
- the **loss** function, which measures how errors should be treated
- the **regularization** $J(f)$, which encodes our inductive bias and preference for certain simpler models

By varying these choices one can construct a huge number of different ML models. Many ML models follow this construction principle or can be interpreted through the lens of regularized risk minimization.