CEE 616: Probabilistic Machine Learning

M2 Linear Methods: L2d Ridge and Lasso Regression

Jimi Oke

UMass Amherst

College of Engineering

October 9, 2025

Outline

Model selection objectives and approaches

Recall the standard linear model:

$$Y = w_0 + w_1 X_1 + \cdots + w_D X_D + \epsilon \tag{1}$$

In selecting a linear model for fitting a dataset, our objectives are:

- Prediction accuracy
- Interpretability

When tasked with finding the best set of predictors, three major methods can be applied:

- Subset selection
- Shrinkage (e.g. ℓ_1 regularization, ℓ_2 regularization)
- Dimensionality reduction: projecting the D-dimensional space of predictors to M-dimensional subspace, $M < D$

Model selection criteria

n: number of observations; p: number of features

1. Adjusted R^2:

$$R_a^2 = 1 - \frac{RSS/(N-D-1)}{TSS/(N-1)} \tag{2}$$

2. C_p statistic:

$$C_p = \frac{1}{N}\left(RSS + 2D\hat{\sigma}^2\right) \tag{3}$$

3. AIC (Gaussian case/least squares):

$$\text{AIC} = \frac{1}{N\hat{\sigma}^2}(RSS + 2D\hat{\sigma}^2) \tag{4}$$

4. BIC (Gaussian case):

$$\text{BIC} = \frac{1}{N\hat{\sigma}^2}\left(RSS + D\hat{\sigma}^2 \cdot \log N\right) \tag{5}$$

The overfitting problem

Claim

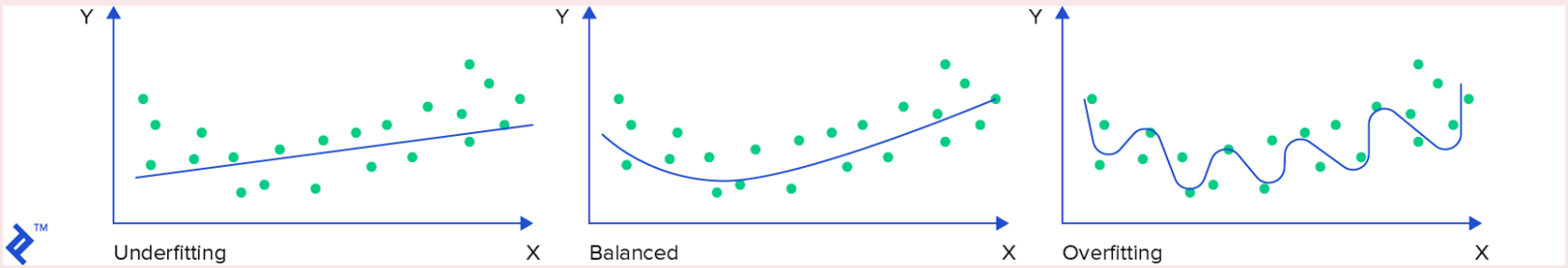
All machine/statistical learning problems are optimization problems, **e.g.**

- Minimize loss function (e.g. linear regression)
- Maximize likelihood (e.g. discriminant analysis)

The overfitting problem in regression

- Myopically increasing the number of **predictors/features** (or model complexity) may decrease the training error of a model:

Reduce training loss $\rightarrow$ **Lower bias** $\rightarrow$ **High variance** $\rightarrow$ **Poor performance**



Correcting for overfitting

Overfitting can be avoided by direct estimation of test error for model selection:

- Cross-validation
- Bootstrapping

We can also **indirectly** estimate the test error using:

- C_p
- AIC
- BIC

Subset selection (wrapper) methods can also be applied to decide on relevant features.

Regularization can also be used to shrink or eliminate irrelevant features.

Mitigation of overfitting via regularization

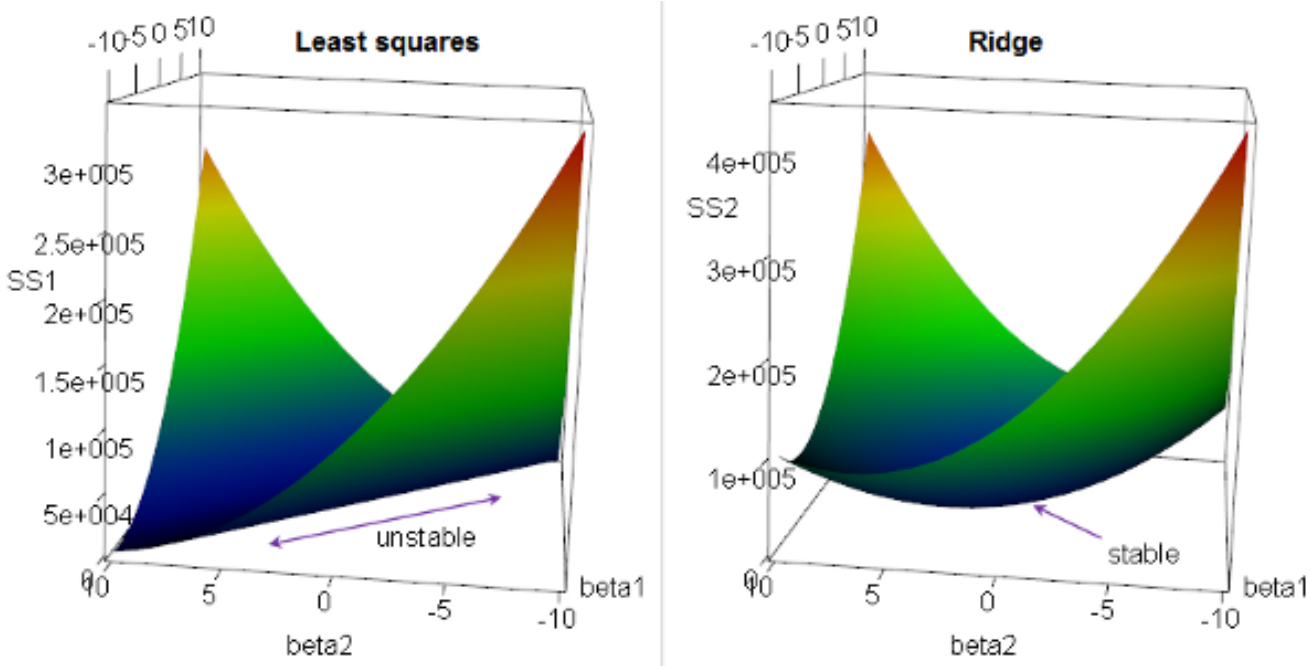
We can control for overfitting by introducing a **regularization term** in the loss function to be optimized:

$$\mathcal{L} = \sum_{n=1}^{N}(y_n - \hat{y}_n)^2 + \lambda \sum_{d=1}^{D} f(w_d) \tag{6}$$

- The regularization term penalizes the model coefficients (shrinks them to 0)
- Tuning parameter λ modulates the impact of the penalty
- As $\lambda \to \infty$, $w \to 0$
- The best value of λ can also be learned (e.g. via cross-validation)
- Regularization ensures that the coefficients of irrelevant predictors are sufficiently reduced and can also be used for feature selection.

Further perspective on ridge regression

Regularization helps us solve ill-posed problems. In this case, the OLS *RSS* is unstable.



With the addition of the ℓ_2 regularization, we can reduce uncertainty about the estimate.

Ridge regression

As an alternative to MLE, which seeks to maximize the likelihood $\prod_n p(y_n|\boldsymbol{x}_n, \boldsymbol{w})$, we can perform a **MAP** estimation to find the posterior mode:

$$\hat{\boldsymbol{w}}_{map} = \arg\max_{\boldsymbol{w}} p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X}) \tag{7}$$

- MAP estimation mitigates overfitting
- MAP estimation with a Gaussian prior on $\boldsymbol{w}$ is known as **ridge regression**
- Yields an ℓ_2 regularization

Maximum a posteriori (MAP) estimation

The OLS model is given by:

$$p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{X}\boldsymbol{w},\sigma^2\boldsymbol{I}) \tag{8}$$

In the Bayesian view, $\boldsymbol{w}$ is random with some **prior** distribution (assume **Gaussian**):

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0},\tau^2\boldsymbol{I}) = \frac{1}{\sqrt{2\pi\tau^2}}\exp\left[-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{0})^\top\frac{1}{\tau^2}\boldsymbol{I}(\boldsymbol{w}-\boldsymbol{0})\right] \tag{9}$$

According to Bayes theorem, the posterior is proportional to:

$$\begin{aligned} p(\boldsymbol{w}|\boldsymbol{y},\boldsymbol{X}) &\propto p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{w})p(\boldsymbol{w}) && (10)\\ &\propto e^{\left[-\frac{1}{2}\left(\boldsymbol{w}-\boldsymbol{0}\right)^\top\frac{1}{\tau^2}\boldsymbol{I}\left(\boldsymbol{w}-\boldsymbol{0}\right)\right]}\cdot e^{\left[-\frac{1}{2}\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\right)^\top\frac{1}{\sigma^2}\boldsymbol{I}\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\right)\right]} && (11)\\ &= \exp\left[-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^\top(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}) - \frac{1}{2\tau^2}||\boldsymbol{w}||_2^2\right] && (12)\end{aligned}$$

MAP estimation of linear model with Gaussian prior

We can express $\hat{\boldsymbol{w}}_{map}$ as a minimization of the negative log-posterior:

$$\hat{\boldsymbol{w}}_{map} = \arg\max_{\boldsymbol{w}} \log \left(\exp \left[-\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) - \frac{1}{2\tau^2}||\boldsymbol{w}||_2^2 \right] \right) \quad (13)$$

$$= \arg\min_{\boldsymbol{w}} \frac{1}{\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) + \frac{1}{\tau^2}||\boldsymbol{w}||_2^2 \quad (14)$$

$$= \arg\min_{\boldsymbol{w}} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) + \frac{\sigma^2}{\tau^2}||\boldsymbol{w}||_2^2 \quad (15)$$

$$= \arg\min_{\boldsymbol{w}} RSS(\boldsymbol{w}) + \lambda||\boldsymbol{w}||_2^2 \quad (16)$$

- $\hat{\boldsymbol{w}}_{map} = \hat{\boldsymbol{w}}^{ridge}$ if $\boldsymbol{w}$ has a Gaussian prior (as above)
- $\lambda = \frac{\sigma^2}{\tau^2}$ indicates the strength of the prior on $\boldsymbol{w}$
- The bias w_0 is not included in $||\boldsymbol{w}||_2$ as it does not contribute to the variance:

$$||\boldsymbol{w}||_2^2 := \sum_{d=1}^{D} w_d^2 \quad (17)$$

Ridge estimator

The ridge regression objective (negative log-posterior) can be written also as:

$$RSS(\boldsymbol{w}, \lambda) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) + \lambda \boldsymbol{w}^\top \boldsymbol{w} \tag{18}$$

Taking the derivative and equating to 0, we obtain:

$$\boxed{\hat{\boldsymbol{w}}^{ridge} = (\boldsymbol{X}^T \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^T \boldsymbol{y}} \tag{19}$$

where $\boldsymbol{I}$ is the $D \times D$ identity matrix.

- The matrix $\boldsymbol{X}^T \boldsymbol{X} + \lambda \boldsymbol{I}$ is *nonsingular*, i.e. it is *invertible* even when $\boldsymbol{X}^T \boldsymbol{X}$ is not full rank
- Recall that a matrix is **full rank** if *all* its columns are *linearly independent*
- The ridge regression coefficients $\hat{\boldsymbol{w}}^{ridge}$ are not scale invariant: thus we **standardize** the inputs before estimation
- Also, the intercept w_0 is not regularized (variance of a column of 1's is 0)
- Thus the input matrix is specified as $N \times D$ instead of $N \times (D + 1)$, while $\boldsymbol{w}$ does not include the intercept

Centering and scaling (normal standardization) of inputs

Note that because of the penalty term $\lambda \sum_{d=1}^{D} w_d^2$, ridge regression is sensitive to the scaling of the parameters (unlike in least squares).

- Thus, we *scale* (standardize) the inputs, i.e.

$$x'_{nd} \leftarrow \frac{x_{nd}}{\hat{\sigma}_d} \tag{20}$$

where $\hat{\sigma}_d = \sqrt{\frac{1}{n-1} \sum_i (x_{nd} - \overline{x}_d)^2}$; all the predictors will have an SD of 1.

- We also **center** the inputs for convenience:

$$x'_{nd} \leftarrow x_{nd} - \overline{x}_d \tag{21}$$

in which case the vector w does not include the intercept which can easily be recovered by $\hat{w}_0 = \frac{1}{N} \sum_i y_i$

- Thus, when we scale and center the inputs, we perform a **normal (z) standardization**:

$$x'_{nd} \leftarrow \frac{x_{nd} - \overline{x}_d}{\hat{\sigma}_d} \tag{22}$$

Ridge regression with standardized inputs

We replace the inputs with their standardized versions, take the derivative of the ridge regression objective and equate to 0:

$$RSS(\lambda) = \sum_{n=1}^{N} \left(y_i - w_0 - \sum_{d=1}^{D} \frac{x_{nd} - \overline{x}_d}{\sigma_d} w_d \right)^2 + \lambda \sum_{d=1}^{D} w_d^2$$

$$RSS'(\lambda)_{w_0} = -\sum_{n=1}^{N} \left(y_i - w_0 - \sum_{d=1}^{D} \frac{x_{nd} - \overline{x}_d}{\sigma_d} w_d \right) = 0$$

$$\sum_{n=1}^{N} (y_i - w_0) - \sum_{n=1}^{N} \sum_{d=1}^{D} \frac{x_{nd} - \overline{x}_d}{\sigma_d} w_d = 0 \quad (\to 0)$$

$$\sum_{n=1}^{N} (y_i - w_0) = 0 \quad \Longrightarrow \quad \boxed{\hat{w}_0 = \overline{y} = \frac{1}{N} \sum_{n=1}^{N} y_i}$$

The estimate of the intercept is thus the mean response if inputs are centered

Ridge regression (alternate motivation)

The OLS procedures find $\hat{w}$ which minimizes the *RSS*:

$$RSS = \sum_{n=1}^{N} \left(y_i - w_0 - \sum_{d=1}^{D} x_{nd} w_d \right)^2 \tag{23}$$

Thus:

$$\hat{w} = \arg\min_{w} \left\{ \sum_{n=1}^{N} \left(y_i - w_0 - \sum_{d=1}^{D} x_{nd} w_d \right)^2 \right\} \tag{24}$$

In ridge regression, we estimate the coefficients $\hat{w}^{\text{ridge}}$ to minimize the *RSS* augmented with an ℓ_2 penalty term:

$$\hat{w}^R = \arg\min_{w} \left\{ \sum_{n=1}^{N} \left(y_i - w_0 - \sum_{d=1}^{D} x_{nd} w_d \right)^2 + \lambda \sum_{d=1}^{D} w_d^2 \right\} \tag{25}$$

Ridge regression (cont.)

Alternatively, we can express the ridge coefficient as an optimization problem with constraints:

$$\hat{w}^R = \arg\min_w \left\{ \sum_{n=1}^{N} \left(y_i - w_0 - \sum_{d=1}^{D} x_{nd} w_d \right)^2 \right\} \quad \text{subject to} \quad \sum_{d=1}^{D} w_d^2 \leq B \tag{26}$$

- The constraint is a **ball** (hypersphere) constraining the size of w.

In both formulations, λ and B are considered tuning parameters (*hyperparameters*) whose optimal values can be learned (e.g. via cross-validation).

Ridge regression vs. ordinary least squares

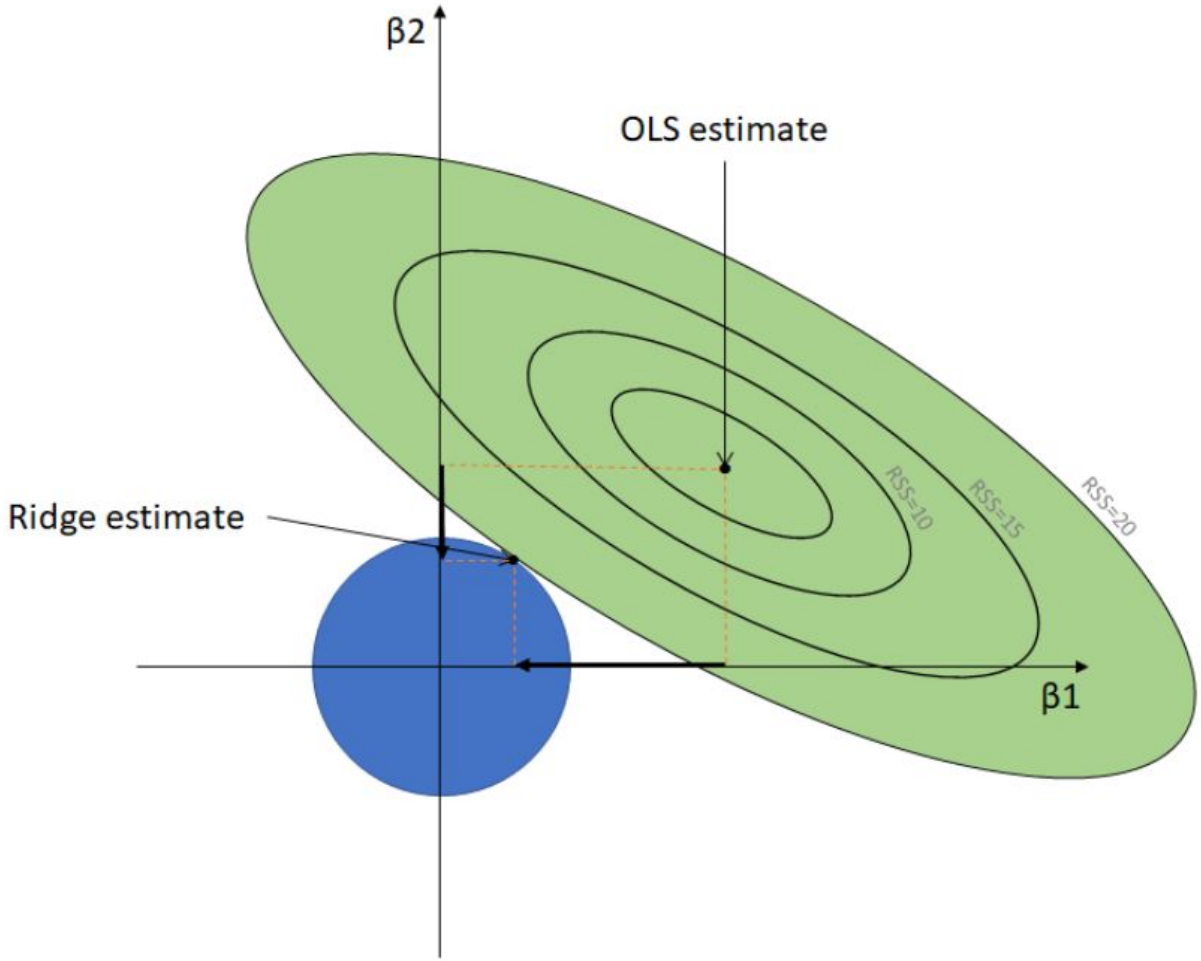


Figure: Ridge regression estimate compared to that of OLS

Example: Ridge regression for mpg

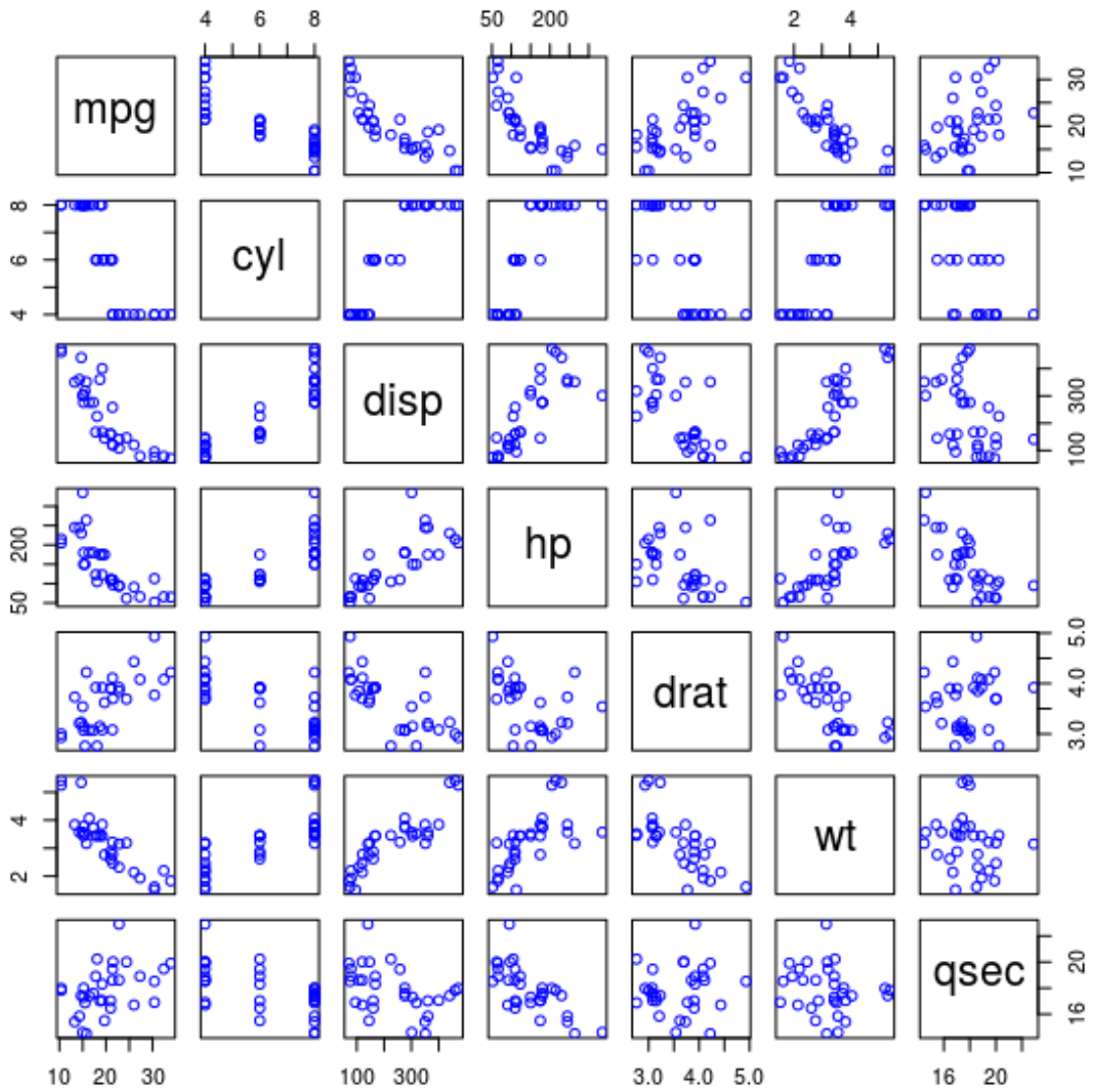
We consider the `mtcars` dataset, with observations taken from the 1974 *Motor Trend* US magazine, comprising fuel consumption (mpg) and 10 features of automobile design and performance. Sample size: 32.

Data description

```
A data frame with 32 observations on 11 (numeric) variables.

[, 1] mpg Miles/(US) gallon
[, 2] cyl Number of cylinders
[, 3] disp Displacement (cu.in.)
[, 4] hp Gross horsepower
[, 5] drat Rear axle ratio
[, 6] wt Weight (1000 lbs)
[, 7] qsec 1/4 mile time
[, 8] vs Engine (0 = V-shaped, 1 = straight)
[, 9] am Transmission (0 = automatic, 1 = manual)
[,10] gear Number of forward gears
[,11] carb Number of carburetors
```

Example: Ridge regression for mpg (cont.)



Example: Ridge regression for mpg (cont.)

The optimal tuning parameter, λ^*, is learned via cross-validation (for test error estimation in each case).

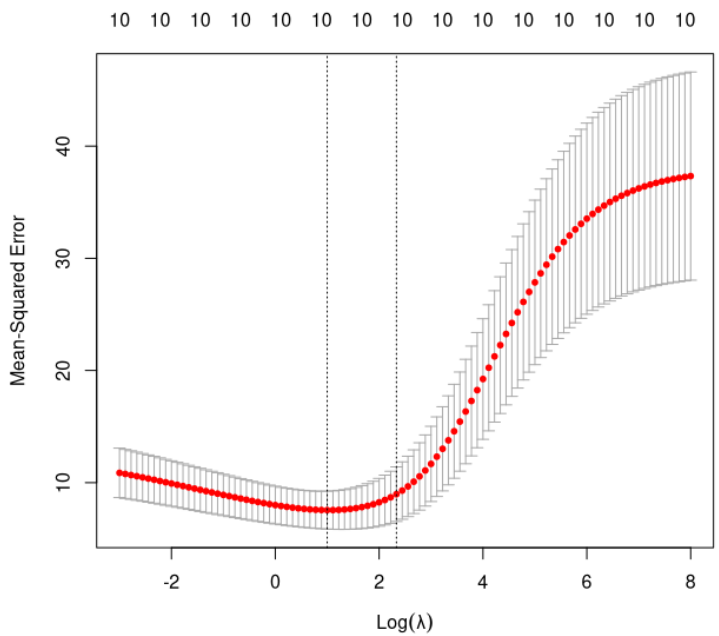


Figure: Ridge regression regularization path based on MSE

Here $\lambda^* = 2.72$.

Example: Ridge regression for mpg (cont.)

As λ increases, the coefficients shrink to 0

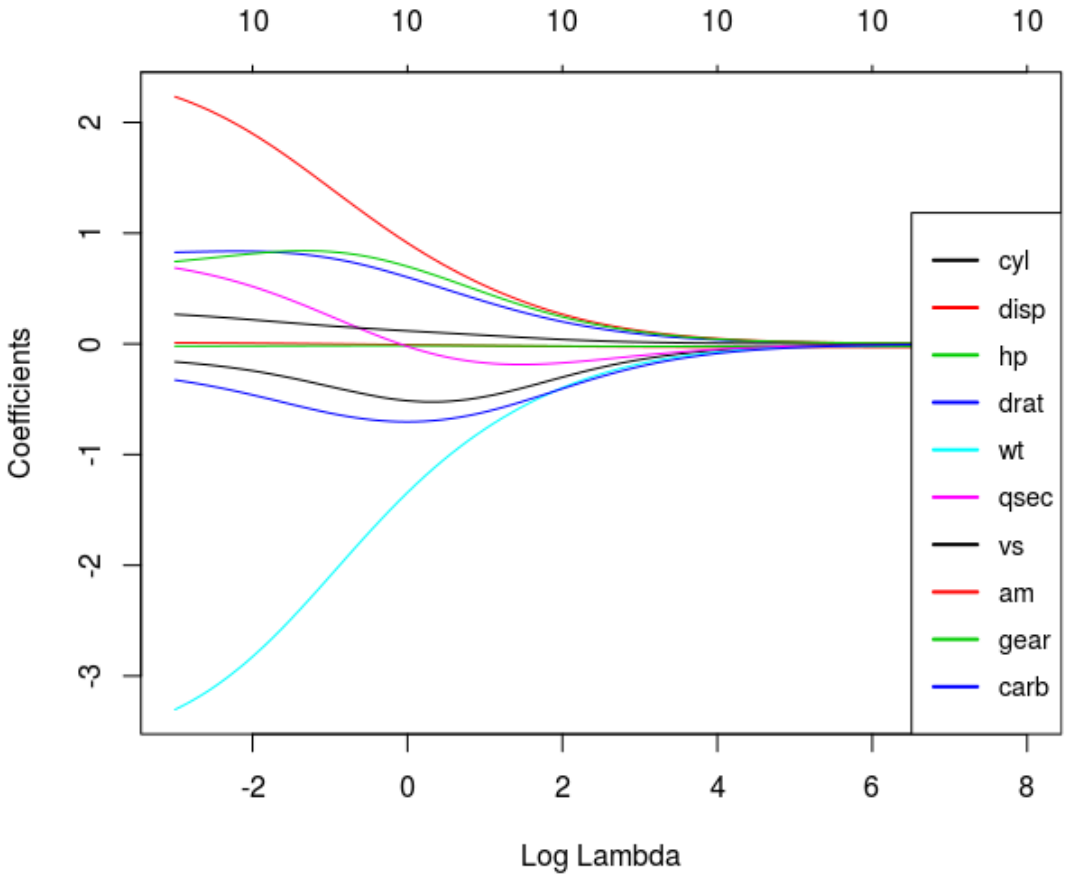


Figure: Impact of λ on ridge regression coefficient estimates

Example: Ridge regression for mpg (cont.)

The AIC and BIC can also be used to find λ^*:

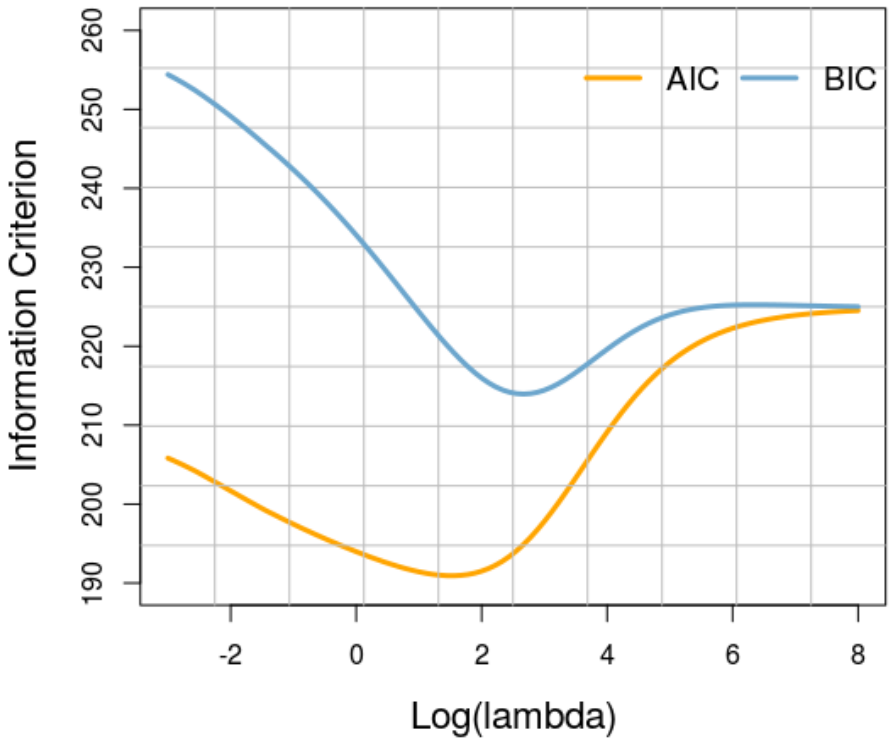


Figure: Impact of λ on ridge regression coefficient estimates

Here: $\lambda^*_{\text{AIC}} = 4.7$, $\lambda^*_{\text{BIC}} = 14.4$

Lasso regression

In lasso (**l**east **a**bsolute **s**hrinkage **s**election **p**rior) regression, we assume a **Laplace** prior on the parameters:

$$p(\boldsymbol{w}|\lambda) = \text{Lap}(\boldsymbol{w}|\mathbf{0}, b) = \frac{1}{2b}\exp\left(-\frac{|\boldsymbol{w} - \mathbf{0}|}{b}\right) \tag{27}$$

- Following the MAP estimation steps, this results in the following estimator:

$$\hat{\boldsymbol{w}}_{map}^{\text{lasso}} = \arg\min_{\boldsymbol{w}} RSS(\boldsymbol{w}) + \lambda||\boldsymbol{w}||_1 \tag{28}$$

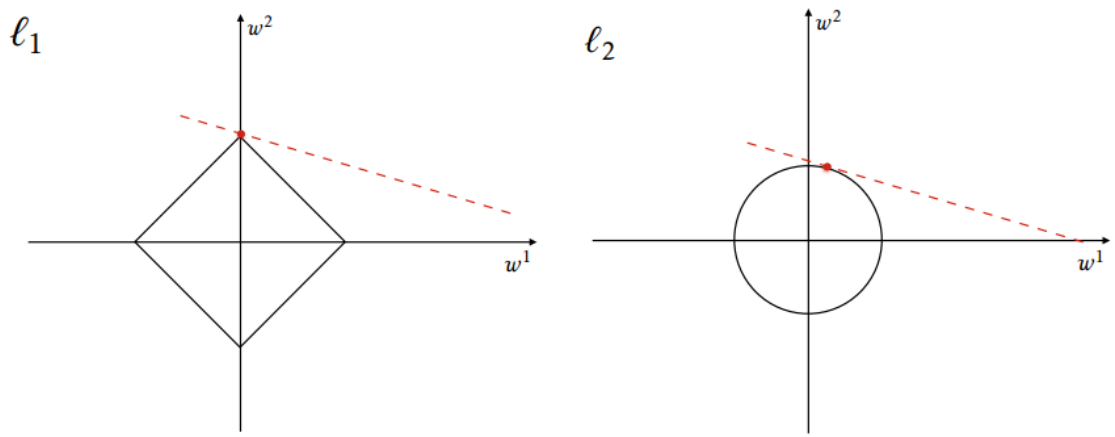
- $||\boldsymbol{w}||_1 = \sum_{d=1}^{D} |w_d|$ is the ℓ_1 norm of $\boldsymbol{w}$
- MAP estimation with a Laplace prior is known as ℓ_1-regularization

Lasso as feature selector

In Lagrangian form, the **lasso** estimate is given by:

$$\hat{w}^{\text{lasso}} = \arg\min_{w} \left\{ \sum_{n=1}^{N} \left(y_i - w_0 - \sum_{d=1}^{D} x_{nd} w_d \right)^2 + \lambda \sum_{d=1}^{D} |w_d| \right\} \tag{29}$$

- The lasso differs from ridge regression in that it uses ℓ_1 instead of ℓ_2 regularization (*sparsity constraint* $\rightarrow$ *greater interpretability*)



- Thus, it forces irrelevant coefficients to zero instead of shrinking them
- And therefore acts as a **feature selector**

Solving the lasso

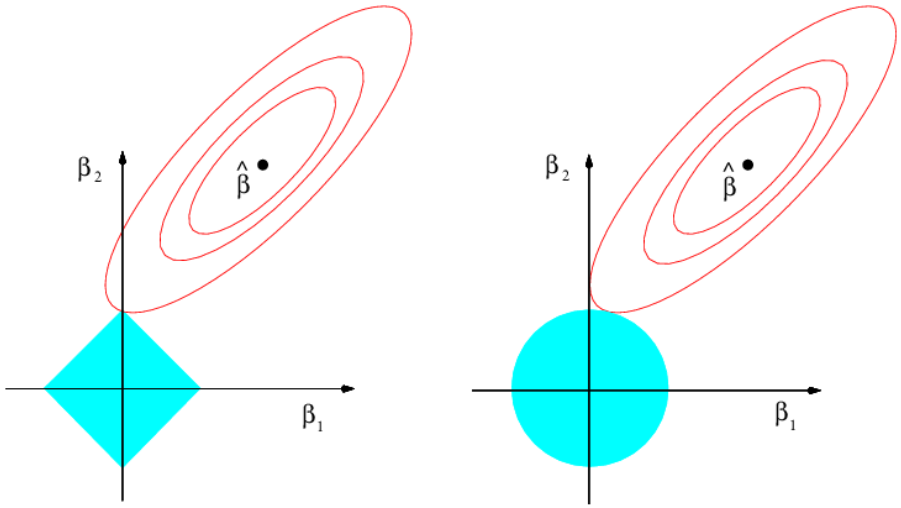


Figure: Comparing the lasso and ridge regression approaches

- $\hat{\boldsymbol{w}}^{\text{lasso}}$ has no closed form (unlike ridge regression)
- The optimization problem can be solved via quadratic programming (convex optimization)
- An algorithm (Efron et al.) for computing the lasso path: **least angle regression (LARS)**
- The lasso can be implemented in R using the `lars` package or in Python using `LassoCV`, `LassoLarsCV` or `LassoLarsIC` modules in `sklearn`.

Best subset selection

Given a dataset with D possible predictors, how many possible subsets can be selected in a linear model?

Number of possible linear models for a given dataset

In deciding on the subset, we have 2 possibilities for each predictor: include/exclude.
Since there are D predictors, the total number of possibilities is:

$$\prod_{d=1}^{D} 2 = 2^D \tag{30}$$

Thus, to select the best subset of predictors, we could examine all 2^D possibilities (exhaustive enumeration), but this could be computationally expensive.

Can we reduce the problem from 2^D possibilities to $D + 1$?

Best subset selection algorithm

1. Let $\mathcal{M}_0$ be the null model containing no predictors.
2. For $k = 1, 2, \ldots, D$:
   a. Fit all $\binom{D}{k}$ models containing exactly k predictors
   b. Pick the model with the smallest RSS. Let this be $\mathcal{M}_k$
3. Select a single best model from $\mathcal{M}_0, \ldots, \mathcal{M}_D$ using the CV prediction error (AIC), BIC or R_a^2

Notes on best subset selection

- Can quickly get intractable if $D > 40$
- Can be generalized for other types of regression (e.g. logistic regression) and models
- Ultimately, 2^D models still have to be estimated, but only $\binom{D}{k}$ need to be evaluated in each step

Alternatives

1. Forward stepwise selection
2. Backward stepwise selection

Forward stepwise selection

As earlier discussed, this reduces the size of the best subset selection problem by evaluating a maximum of $D - k$ models in each step, with variables incrementally added to improve the goodness-of-fit measure in each iteration:

1. Let $\mathcal{M}_0$ denote the null model
2. For $k = 0, \ldots, D - 1$:
   a. Evaluate all $D - k$ models that augment the predictors in $\mathcal{M}_k$ by 1 additional predictor
   b. Choose the best of these models (based on R^2, RSS, etc), and call it $\mathcal{M}_{k+1}$
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_D$ using the CV prediction error (AIC), BIC or R_a^2

Notes

- Forward stepwise selection can fail to find the best subset if it does not contain the predictor chosen in $\mathcal{M}_{k+1}$

Backward stepwise selection

In this case, we begin with the full model containing all p predictors and then iteratively remove the least relevant predictor at each step:

1. Let $\mathcal{M}_D$ denote the full model containing all p predictors
2. For $k = D, D-1, \ldots, 1$:
   a. Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$
   b. Choose the best (based on R^2, RSS, etc) among these k models. Let it be $\mathcal{M}_{k-1}$
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_D$ using the CV prediction error (AIC), BIC or R_a^2

Comparing forward and backward stepwise selection

- Both require the estimation of the same number of models:

$$1 + \sum_{k=0}^{D-1}(D-k) = 1 + \frac{D(D+1)}{2} \tag{31}$$

- Backward selection requires $N > D$ but forward selection does not have this restriction. Why?
- Thus, when D is large, forward selection is the viable approach
- Hybrid approaches combining both forward and backward steps can be implemented for flexibility and computational feasibility

Example: Predicting monthly temperature

In this example, we consider monthly average temperature for Boston, MA (1978 to 2019).

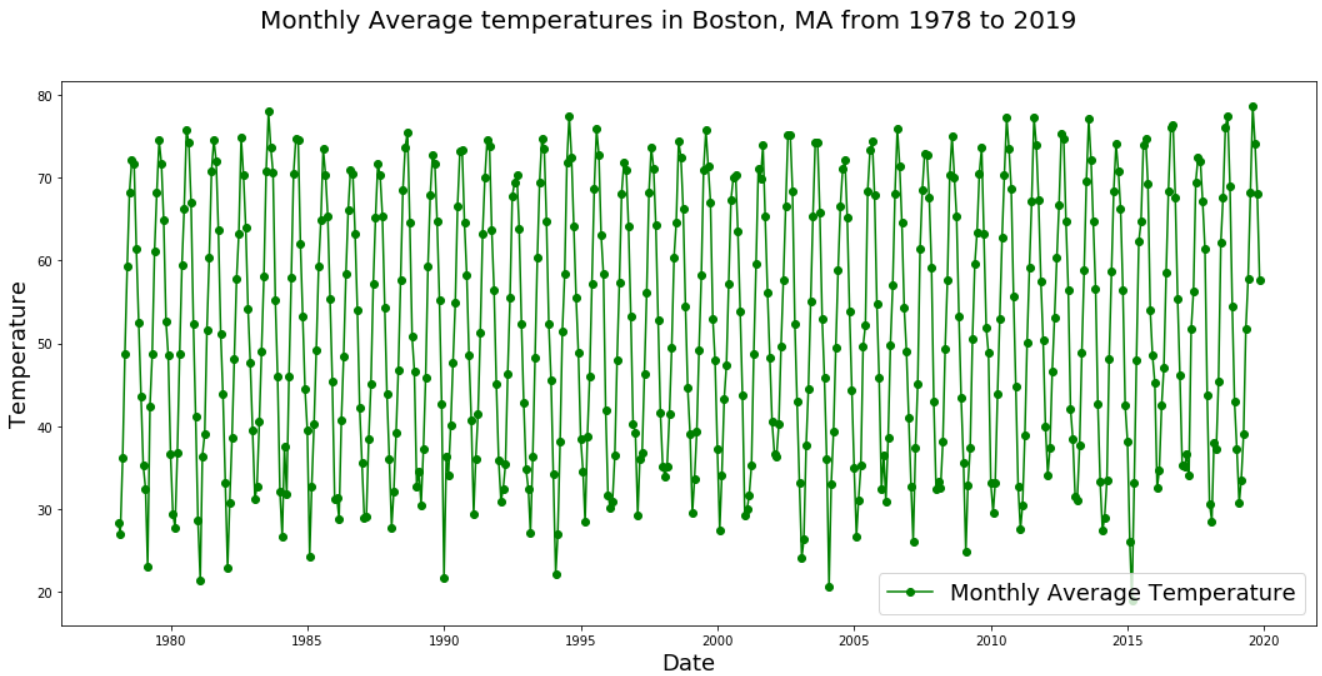


Figure: Monthly average temperature in Boston from 1978 to 2019

Example: Predicting monthly temperature (cont.)

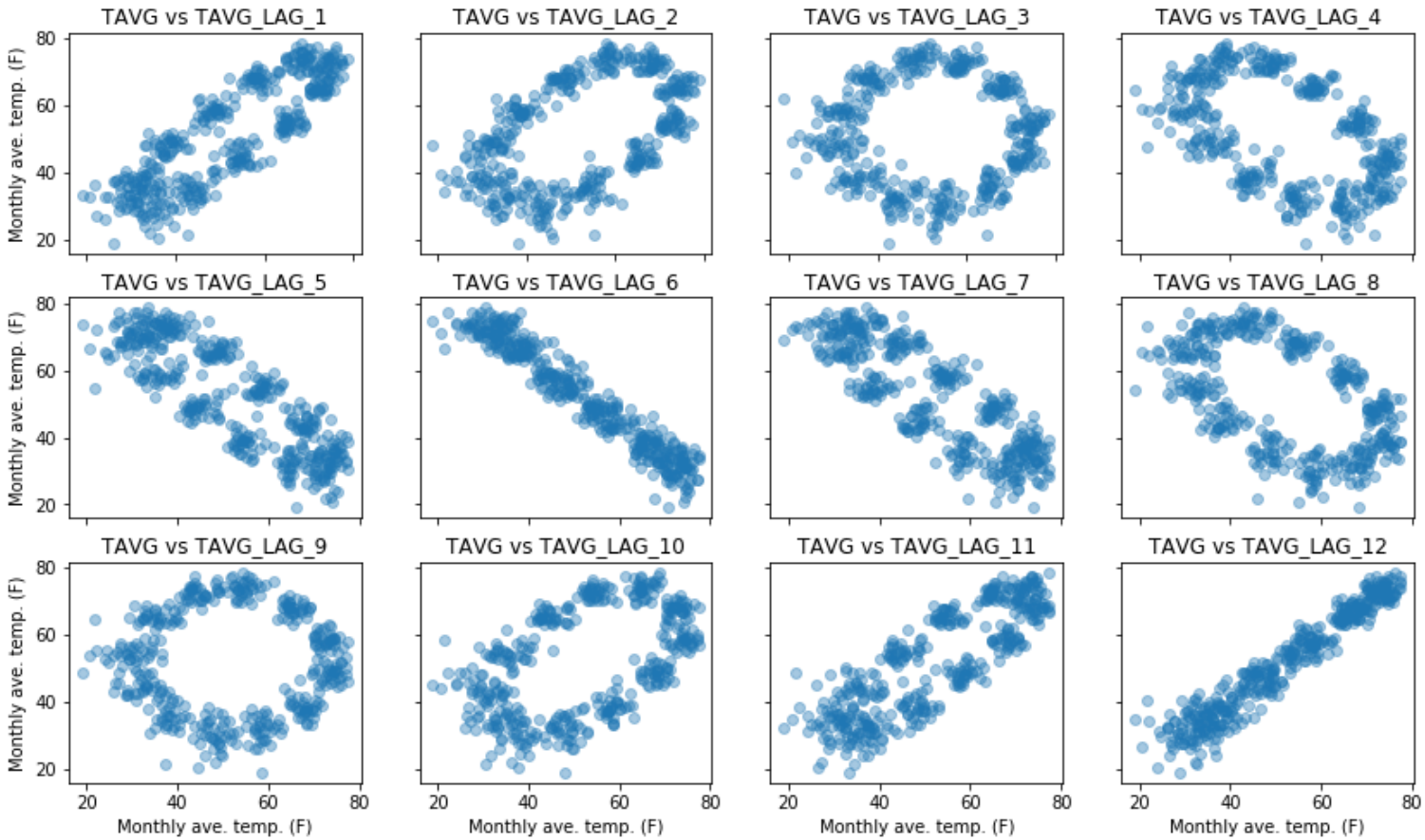
In time series regression, we might want to include the effects historical observations on the current response using **lag variables**:

$$
\begin{aligned}
X_t &\quad \text{(observation in period t)} \\
X_{\text{lag}_1} &= X_{t-1} \\
X_{\text{lag}_2} &= X_{t-2} \\
&\vdots \quad \vdots \\
X_{\text{lag}_p} &= X_{t-p}
\end{aligned}
$$

where k is the number of historical time periods.

Example: Predicting monthly temperature (cont.)

We create 12 lag variables: `T_AVG_LAG_k`, $\texttt{k} = 1, \ldots, 12$.



Which lag variables are strongly correlated with monthly temperature?

Example 1: Predicting monthly temperature (cont.)

In the Jupyter notebook, we illustrate an exhaustive enumeration to select the best linear regression model using AIC:

```
Best expr=TAVG ~ TAVG_LAG_1 + TAVG_LAG_2 + TAVG_LAG_5 + TAVG_LAG_6 + TAVG_LAG_10 + TAVG_LAG_11 + TAVG_LAG_12
min AIC=1989.7279346561804
                            OLS Regression Results
==============================================================================
Dep. Variable:                   TAVG   R-squared:                       0.964
Model:                            OLS   Adj. R-squared:                  0.963
Method:                 Least Squares   F-statistic:                     1458.
Date:                Fri, 21 Feb 2020   Prob (F-statistic):          1.15e-272
Time:                        02:51:46   Log-Likelihood:                -986.86
No. Observations:                 393   AIC:                             1990.
Df Residuals:                     385   BIC:                             2022.
Df Model:                           7
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      31.2917      6.126      5.108      0.000      19.247      43.337
TAVG_LAG_1      0.2809      0.047      5.915      0.000       0.188       0.374
TAVG_LAG_2      0.1038      0.042      2.461      0.014       0.021       0.187
TAVG_LAG_5     -0.1530      0.048     -3.156      0.002      -0.248      -0.058
TAVG_LAG_6     -0.2334      0.049     -4.730      0.000      -0.330      -0.136
TAVG_LAG_10     0.0780      0.044      1.786      0.075      -0.008       0.164
TAVG_LAG_11     0.1663      0.051      3.263      0.001       0.066       0.267
TAVG_LAG_12     0.1517      0.050      3.047      0.002       0.054       0.250
==============================================================================
Omnibus:                       14.731   Durbin-Watson:                   2.014
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               25.147
Skew:                          -0.232   Prob(JB):                     3.46e-06
Kurtosis:                       4.149   Cond. No.                     5.53e+03
==============================================================================
```

Example: Predicting monthly temperature (cont.)

We compare the fitted values to the observations in the test set using this "best" model.

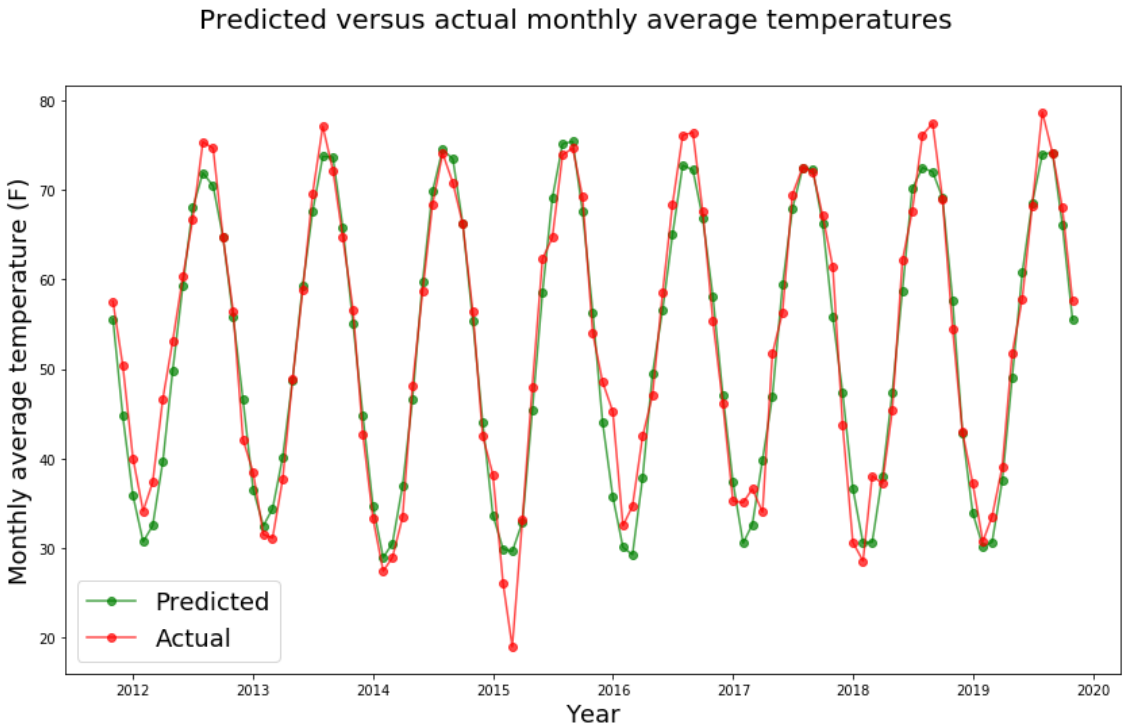


Figure: Performance of "best" model

Examine the code in the Jupyter notebook. Identify at least **3 weaknesses** of the approach used and how you would address them.

Summary

Model performance

We consider model fitness by:

- Inference (statistical tests on parameter significance)
- Error estimation (MSE, R^2, AIC, BIC, etc)

Subset selection

Algorithms for selecting best number of features

- Forward stepwise
- Backward stepwise

Regularization

Another approach to mitigating overfitting

- Ridge regression is closed form and shrinks the coefficients to zero
- The lasso produces a sparse solution (effective for feature selection) as it employs ℓ_1 regularization

Reading assignments

- **PMLI** 11.3-4
- **ESL** 3.3-7

OLS model

The ordinary least squares (OLS) model is given by

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\epsilon} \tag{32}$$

where the errors are independent and identically distributed (iid) as follows:

- $\mathbb{E}(\boldsymbol{\epsilon}) = \mathbf{0}$ (zero mean)
- $Cov(\boldsymbol{\epsilon}) = \sigma^2 \boldsymbol{I}$ (constant variance)

Also:

- $\boldsymbol{y}$ and $\boldsymbol{\epsilon}$ are normally distributed random variables (r.v.'s)
- $\mathbb{E}(\boldsymbol{y}) = \boldsymbol{X}\boldsymbol{w} + \mathbb{E}(\boldsymbol{\epsilon}) = \boldsymbol{X}\boldsymbol{w}$

Recall the OLS estimator:

$$\hat{\boldsymbol{w}}_{OLS} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{33}$$

which is an **unbiased** estimator of w, since

$$\mathbb{E}(\hat{\boldsymbol{w}}_{OLS}) = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\mathbb{E}(\boldsymbol{y}) = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = \boldsymbol{w} \tag{34}$$

Generalized least squares (GLS) model

The GLS model is given by

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\epsilon} \tag{35}$$

where

- $\mathbb{E}(\boldsymbol{\epsilon}) = \mathbf{0}$ (zero mean)
- $Cov(\boldsymbol{\epsilon}) = \sigma^2\boldsymbol{\Omega}$
- $\boldsymbol{\Omega}$ is a known $n \times n$ positive definite (PD) matrix (i.e. not necessarily constant variance)

The GLS estimator is then given by:

$$\hat{\boldsymbol{w}}_{GLS} = (\boldsymbol{X}^T\boldsymbol{\Omega}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Omega}^{-1}\boldsymbol{y} \tag{36}$$

Fitted values are given as

$$\hat{\boldsymbol{y}}_{GLS} = \boldsymbol{X}\hat{\boldsymbol{w}}_{GLS} \tag{37}$$

Weighted least squares (WLS)

Special case of GLS where $\boldsymbol{\Omega}$ is diagonal:

$$\boldsymbol{\Omega} = \boldsymbol{W}^{-1} = \begin{bmatrix} w_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_n \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{w_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{w_n} \end{bmatrix} \tag{38}$$

where $w_1, \ldots, w_n$ are weights.

Thus, the WLS model is specified as:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\epsilon} \tag{39}$$

where $\mathbb{E}(\boldsymbol{\epsilon}) = \mathbf{0}$ and $Cov(\boldsymbol{\epsilon}) = \sigma^2 \text{diag}\,(w_1, \ldots, w_n)$
And the estimator is:

$$\hat{\boldsymbol{w}}_{WLS} = (\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{y} \tag{40}$$

WLS in practice

So, how do we estimate the weights to find the WLS estimator?

- It has been shown that $\hat{\boldsymbol{w}}_{WLS}$ is BLUE when:

$$w_i = \frac{1}{\sigma_i^2} \tag{41}$$

- In many cases, the i-th squared residual e_i is a good estimate of σ_i^2
- If the relationship between the residuals and fitted values indicates a megaphone shape, regress $|e_i|$ against the fitted values to find $\hat{\sigma}_i$
- If the squared residuals indicate an upward trend w.r.t predictors or fitted values, regress against to find $\hat{\sigma}_i$

Feasible generalized least squares (FGLS)

When a diagonal covariance matrix cannot be assumed, the FGLS provides a more general approach.

In FGLS, we assume that $\boldsymbol{\Omega} = \boldsymbol{\Omega}(\boldsymbol{\theta})$, i.e. it is a function of an unknown vector of parameters $\boldsymbol{\theta}$.

The estimator is given by:

$$\hat{\boldsymbol{w}}_{FGLS} = (\boldsymbol{X}^T \hat{\boldsymbol{\Omega}}^{-1} \boldsymbol{X})^{-1} \boldsymbol{X}^T \hat{\boldsymbol{\Omega}}^{-1} \boldsymbol{y} \tag{42}$$