INDR 450/550

Spring 2022

Lecture 15: Model Shrinkage,
Non-linear regressions

April 4, 2022

Fikri Karaesmen

Announcements

- Class Exercise at the end of lecture today. If you are participating online, please upload your document under Course Contents/Class Exercises
- HW 2 available with a deadline of April 4 (Labs 3 and 4).
- Lab 5 material (on KNN regression) and a short video are available
- Exam scheduled.

- The first five labs were uploaded. Please follow them.
  - Current HW based on lab2 and lab3

Predictive Analytics

- Remaining topics (to complete at the latest the week after the spring break)
  - Validation
  - Model selection / regularization
  - Non-linear regressions, generalized additive models
  - Tree-based methods

Regularization: Ridge regression

- We would like to limit the number of predictors that are used.
- Inspired by lagrangian relaxation formulations of constrained optimization problems here is a way to do that: we add a penalty term to the objective function that penalizes the magnitude of the coefficients and solve:

$$\min \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

- where λ plays the role of a lagrange multiplier (i.e. a penalty parameter)
- This formulation seeks a trade-off between the MSE and the size of the coefficients, thereby forcing some of them to zero as λ becomes larger.
- We can then solve this problem for different values of λ to obtain solutions that have different MSEs and a different number of predictors.

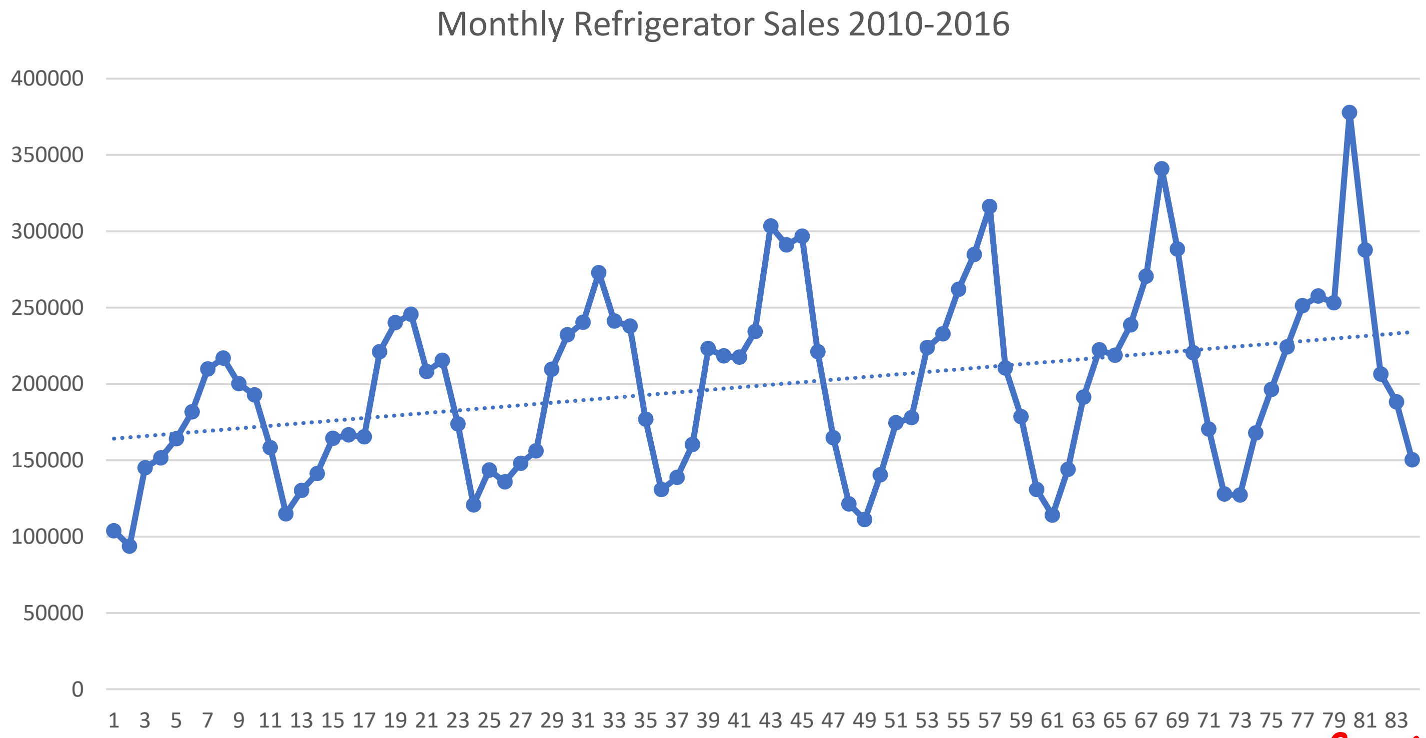
Regularization: Ridge regression

- Ridge regression works by finding the right tradeoff between the MSE (or RSS) and the penalty function.
  - The reduced model may perform better on the test set.
- Because of the penalty function, the predictors have to be normalized so that they are of the same scale.
- The typical normalization is standardizing by dividing by the standard deviation.

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2}},$$

Regularization: Ridge regression Example

- The monthly refrigerator sales data for Turkey (300 months starting from Jan. 2004)



trend and seasonality

Regularization: Ridge regression Example

- I'll experiment with monthly dummies and trend terms (t,t^2, etc.)

On a spreadsheet, part of the data table would look like:

trend
dummy January
dummy February

Jan-10	103862	1	1	0	0	0	0	0
Feb-10	93764	2	0	1	0	0	0	0
Mar-10	145047	3	0	0	1	0	0	0
Apr-10	151481	4	0	0	0	1	0	0
May-10	164173	5	0	0	0	0	1	0
Jun-10	181646	6	0	0	0	0	0	1
Jul-10	209652	7	0	0	0	0	0	0
Aug-10	216918	8	0	0	0	0	0	0
Sep-10	200085	9	0	0	0	0	0	0
Oct-10	192734	10	0	0	0	0	0	0
Nov-10	158260	11	0	0	0	0	0	0
Dec-10	114968	12	0	0	0	0	0	0

Regularization: Ridge regression Example

- This is what the dataframe looks like:

```
8]: dftrain.head()
```

8]:

	Sales	x1t	x2t	x3t	x4t	x5t	x6t	x7t	x8t	x9t	x10t	x11t	x12t	x13t	x14t	x15t
Jan-94	54640	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1.000000
Feb-94	76247	2	0	1	0	0	0	0	0	0	0	0	0	4	8	1.414214
Mar-94	77781	3	0	0	1	0	0	0	0	0	0	0	0	9	27	1.732051
Apr-94	61416	4	0	0	0	1	0	0	0	0	0	0	0	16	64	2.000000
May-94	47039	5	0	0	0	0	1	0	0	0	0	0	0	25	125	2.236068

Regularization: Ridge regression Example

- Here's the full regression result on the training set: (first 240 months training, last 60 months test)

Out[30]:

OLS Regression Results

Dep. Variable:	Sales	**R-squared:**	0.741
Model:	OLS	**Adj. R-squared:**	0.724
Method:	Least Squares	**F-statistic:**	42.78
Date:	Fri, 01 Apr 2022	**Prob (F-statistic):**	3.49e-57
Time:	11:13:52	**Log-Likelihood:**	-2800.9
No. Observations:	240	**AIC:**	5634.
Df Residuals:	224	**BIC:**	5689.
Df Model:	15		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	5.074e+04	2.65e+04	1.916	0.057	-1446.253	1.03e+05
x1t	2118.0871	1479.275	1.432	0.154	-796.989	5033.163
x2t	-7063.2097	9302.379	-0.759	0.448	-2.54e+04	1.13e+04
x3t	-63.4137	9290.754	-0.007	0.995	-1.84e+04	1.82e+04
x4t	1.8e+04	9284.229	1.938	0.054	-300.504	3.63e+04
x5t	2.485e+04	9279.968	2.678	0.008	6566.685	4.31e+04
x6t	3.992e+04	9276.956	4.304	0.000	2.16e+04	5.82e+04
x7t	5.782e+04	9274.724	6.235	0.000	3.95e+04	7.61e+04
x8t	7.761e+04	9273.027	8.369	0.000	5.93e+04	9.59e+04
x9t	6.849e+04	9271.731	7.387	0.000	5.02e+04	8.68e+04
x10t	5.407e+04	9270.760	5.832	0.000	3.58e+04	7.23e+04
x11t	3.39e+04	9270.078	3.657	0.000	1.56e+04	5.22e+04
x12t	2.272e+04	9269.668	2.451	0.015	4457.515	4.1e+04
x13t	-8.0399	6.519	-1.233	0.219	-20.886	4.806
x14t	0.0181	0.014	1.335	0.183	-0.009	0.045
x15t	-1.084e+04	1.2e+04	-0.903	0.368	-3.45e+04	1.28e+04

Regularization: Ridge regression Example

- All three trend terms appear to be not statistically significant
- But if we remove all of them, the R^2 drops sharply.

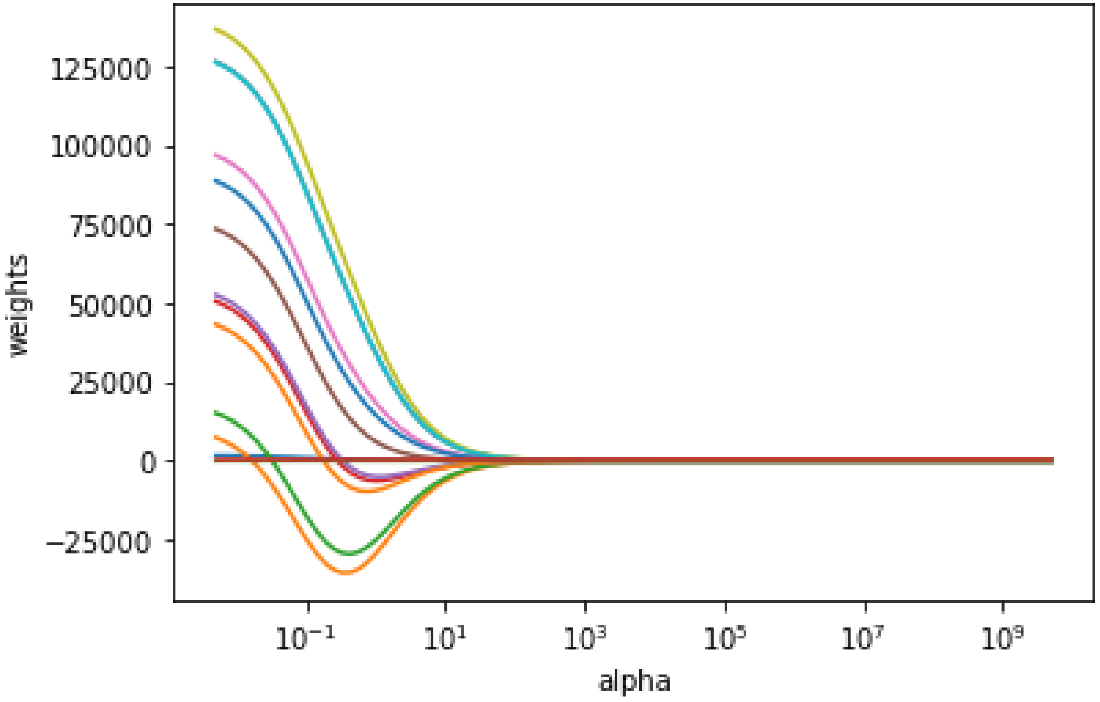
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	5.074e+04	2.65e+04	1.916	0.057	-1446.253	1.03e+05
x1t	2118.0871	1479.275	1.432	0.154	-796.989	5033.163
x2t	-7063.2097	9302.379	-0.759	0.448	-2.54e+04	1.13e+04
x3t	-63.4137	9290.754	-0.007	0.995	-1.84e+04	1.82e+04
x4t	1.8e+04	9284.229	1.938	0.054	-300.504	3.63e+04
x5t	2.485e+04	9279.968	2.678	0.008	6566.685	4.31e+04
x6t	3.992e+04	9276.956	4.304	0.000	2.16e+04	5.82e+04
x7t	5.782e+04	9274.724	6.235	0.000	3.95e+04	7.61e+04
x8t	7.761e+04	9273.027	8.369	0.000	5.93e+04	9.59e+04
x9t	6.849e+04	9271.731	7.387	0.000	5.02e+04	8.68e+04
x10t	5.407e+04	9270.760	5.832	0.000	3.58e+04	7.23e+04
x11t	3.39e+04	9270.078	3.657	0.000	1.56e+04	5.22e+04
x12t	2.272e+04	9269.668	2.451	0.015	4457.515	4.1e+04
x13t	-8.0399	6.519	-1.233	0.219	-20.886	4.806
x14t	0.0181	0.014	1.335	0.183	-0.009	0.045
x15t	-1.084e+04	1.2e+04	-0.903	0.368	-3.45e+04	1.28e+04

Regularization: Ridge regression Example

- The RMSE on the training set is: 28319
- And on the test set: 66097
- The MAPE on the test set is 31.2%.
- Let's see if we can improve the error on the test set by using a ridge regression
- We'll experiment with different values of the penalty parameter λ.

Regularization: Ridge regression Example

- Here are the coefficients as a function of λ. Note that they are shrinking in absolute value but not necessarily in a monotone manner.



Regularization: Ridge regression Example

- Let’s take a look at how the coefficient change as λ changes

λ=0.1

```
print('Test MSE for lambda = 0.1:' , mse1) #
print('Test RMSE for lambda = 0.1:' , np.sqrt(

0        170.164210
1     -21795.829318
2     -15543.655198
3        905.302844
4       7130.332296
5      20900.235750
6      37278.653663
7      55394.022556
8      47046.209427
9      33851.807014
10     15399.879186
11      5186.836906
12         0.467231
13         0.001495
14      3689.909969
dtype: float64
Test MSE for lambda = 0.1: 3834837257.449019
Test RMSE for lambda = 0.1: 61926.06282857824
```

λ=4

```
print('Test RMSE for lambda = 4:' , np.sqrt(

0        74.126524
1     -8626.608049
2     -7094.839435
3     -3180.251075
4     -1658.679388
5      1626.960635
6      5522.152043
7      9823.156901
8      7933.256209
9      4909.258811
10      655.087860
11    -1672.019191
12        0.282890
13        0.001162
14     1411.839997
dtype: float64
Test MSE for lambda = 4: 6458740289.35789
Test RMSE for lambda = 4: 80366.28826415894
```

λ=20

```
print('Test MSE for lambda = 20: ' , mse3)  # C
print('Test RMSE for lambda = 20: ' , np.sqrt(m

0        24.151528
1     -2167.628123
2     -1787.259801
3      -831.947356
4      -454.361846
5       348.977344
6      1299.418109
7      2347.797303
8      1901.813566
9      1182.083245
10      165.420937
11     -386.077172
12        0.093875
13        0.000390
14      454.300925
dtype: float64
Test MSE for lambda = 20:  10771749633.455107
Test RMSE for lambda = 20:  103787.03981449276
```

This has lower MSE than the initial regression.

Regularization: Lasso regression

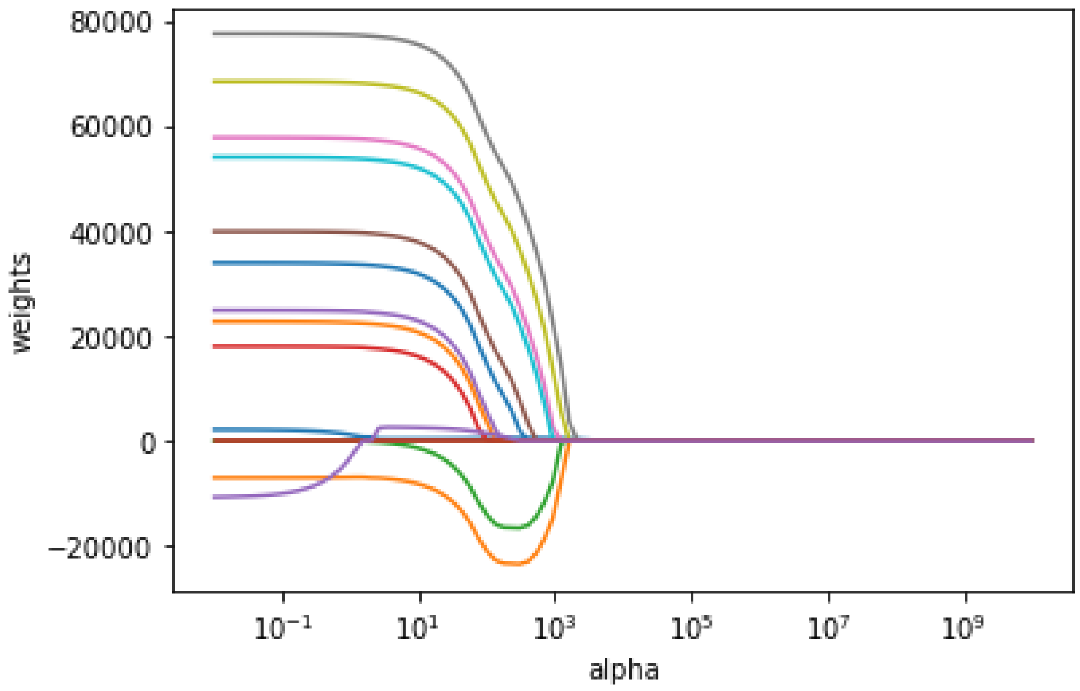
- Similar idea but an alternative formulation

$$\min \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p}|\beta_j|$$

- This is supposed to be an improvement over a ridge regression.
- Due to linearity of the penalty term, the coefficients are driven to zero earlier with respect to a ridge regression.
  - This is a desirable property for model reduction where the goal is to reduce the number of parameters.

Regularization: Lasso regression example

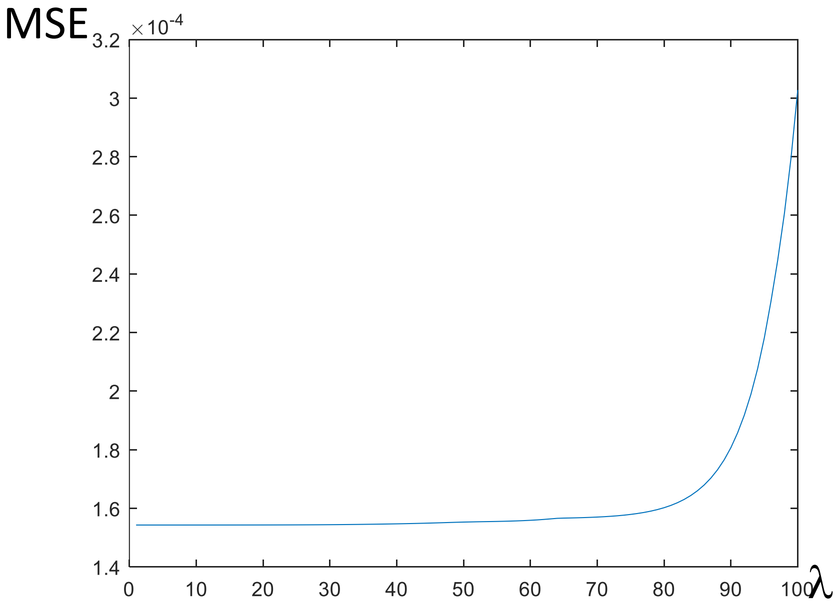
- For the refrigerator data, we have:



Note that some predictors vanish rapidly while others are still shrinking!

Lasso regression

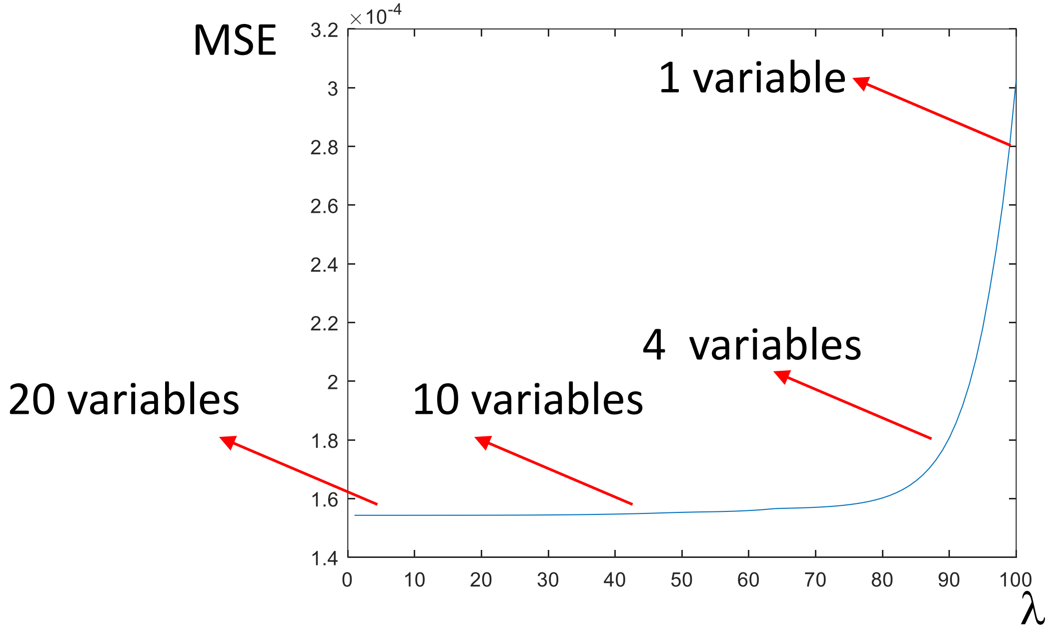
- Note that as λ increases, the model becomes smaller (i.e. throws away some of the variables by setting their coefficients to zero), as a result the MSE grows.



- But we observe that there is no significant loss of MSE until λ exceeds its value at λ_{80}. This is very typical of models with big data.

Lasso regression

- The analyst can then choose the trade-off between MSE and model simplicity (number of predictors).



Lasso regression vs. ridge regression

- Both implement the same idea.
- Equally computationally efficient.
- Lasso is supposed to work better in theory but in practice it’s not always clear.
- Most statistical software allows using combinations of the two.

Lasso regression vs. ridge regression: the elastic net regression

- Elastic net combines ridge and lasso in the following way:

$$\min \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \quad + \mu \sum_{j=1}^{p} |\beta_j|$$

- Lasso and ridge are then special cases:

Lasso regression: Example

- We'll estimate the Daily percentage change in the Istanbul Stock Exchange using lagged predictors corresponding to other financial indicators (SP, DAX, FTSE, NIKKEI).

- Let us use several predictors initially and see if we can filter them out using lasso regression.

Lasso regression Example:

	TL BASED	USD BASED				imkb_x			
date	ISE	ISE	SP	DAX	FTSE	NIKKEI	BOVESPA	EU	EM
5-Jan-09	0.0357537	0.03837619	-0.00468	0.002193	0.003894	0	0.03119	0.012698	0.028524
6-Jan-09	0.0254259	0.03181274	0.007787	0.008455	0.012866	0.004162	0.01892	0.011341	0.008773
7-Jan-09	-0.028862	-0.02635297	-0.03047	-0.01783	-0.02873	0.017293	-0.0359	-0.01707	-0.02002
8-Jan-09	-0.062208	-0.0847159	0.003391	-0.01173	-0.00047	-0.04006	0.028283	-0.00556	-0.01942
9-Jan-09	0.0098599	0.00965811	-0.02153	-0.01987	-0.01271	-0.00447	-0.00976	-0.01099	-0.0078
12-Jan-09	-0.029191	-0.04236116	-0.02282	-0.01353	-0.00503	-0.04904	-0.05385	-0.01245	-0.02263
13-Jan-09	0.0154453	-0.00027218	0.001757	-0.01767	-0.00614	0	0.003572	-0.01222	-0.00483
14-Jan-09	-0.041168	-0.03555248	-0.03403	-0.04738	-0.05095	0.002912	-0.0403	-0.04522	-0.00868
15-Jan-09	0.0006619	-0.01726784	0.001328	-0.01955	-0.01433	-0.05045	0.030314	-0.01207	-0.02343
16-Jan-09	0.0220373	0.03227803	0.007533	0.006791	0.006289	0.025453	0.004867	0.008561	0.010917
19-Jan-09	-0.022692	-0.04434878	-0.05426	-0.01155	-0.00935	0.003239	-0.01315	-0.01205	-0.00403
20-Jan-09	-0.013709	-0.02966137	0	-0.01783	-0.00417	-0.02341	-0.0409	-0.01509	-0.02411
21-Jan-09	0.0008647	0.00152943	0.042572	0.005011	-0.00773	-0.02056	0.033532	-0.00334	-0.00509
22-Jan-09	-0.003815	0.00504316	-0.01528	-0.00984	-0.0019	0.018818	-0.01698	-0.00655	-0.00323
23-Jan-09	0.0056613	-0.01000795	0.005363	-0.00964	7.40E-05	-0.03881	0.006261	-0.00362	-0.00808
26-Jan-09	0.0468313	0.06170818	0.005538	0.034787	0.037891	-0.00818	0.009838	0.0328	0.01032
27-Jan-09	-0.006635	0.01094866	0.010866	-0.0008	-0.00347	0.048148	0.004922	-0.00264	0.006344
28-Jan-09	0.034567	0.03587086	0.033007	0.044182	0.023748	0.005594	0.038725	0.029974	0.022104
29-Jan-09	-0.020528	-0.02027185	-0.03368	-0.02026	-0.02477	0.017723	-0.01475	-0.02311	0.000409
30-Jan-09	-0.008777	-0.02345826	-0.02305	-0.02048	-0.00971	-0.03167	-0.00854	-0.0072	0.002243
2-Feb-09	-0.025919	-0.03560749	-0.00053	-0.01564	-0.01745	-0.01513	-0.01629	-0.01974	-0.01909
3-Feb-09	0.0152795	0.02240282	0.01571	0.02404	0.021039	-0.00617	0.027574	0.017862	0.012719
4-Feb-09	0.0185778	0.02323077	-0.00752	0.026577	0.015275	0.026908	0.009565	0.01877	0.015166
5-Feb-09	-0.014133	-0.01457138	0.016233	0.003932	7.09E-05	-0.01117	0.024128	-0.00414	0.002073
6-Feb-09	0.036607	0.04275893	0.026541	0.029306	0.014788	0.015846	0.039282	0.019127	0.032338
9-Feb-09	0.0113532	0.02146764	0.001484	0.004766	0.003651	-0.01341	-0.01546	0.005627	0.007895
10-Feb-09	-0.040542	-0.04390687	-0.05037	-0.03517	-0.02218	-0.0029	-0.02144	-0.02439	-0.00214
11-Feb-09	-0.022106	-0.03389347	0.007923	0.005434	0.005019	-0.03074	-0.0088	0.001097	-0.00793
12-Feb-09	-0.014888	-0.02082504	0.001738	-0.02742	-0.00761	0	-0.00848	-0.01409	-0.01477
13-Feb-09	0.0070267	0.0097092	-0.01005	0.001322	-0.003	0.009563	0.028551	0.003032	0.017764
16-Feb-09	-0.011495	-0.02462909	-0.04663	-0.04561	-0.01317	-0.00376	0.003999	-0.01272	-0.00685
17-Feb-09	-0.041136	-0.0506875	0	0	-0.02466	-0.0136	-0.04883	-0.03093	-0.03832
18-Feb-09	-0.002631	-0.01210114	-0.00095	-0.00276	-0.00679	-0.01463	-0.00435	-0.0033	-0.00334
19-Feb-09	0.0246546	0.03292321	-0.0121	0.002435	0.002891	0.003076	0.001411	0.002569	0.006764
20-Feb-09	-0.035841	-0.04977568	-0.01148	-0.04875	-0.03271	-0.01887	-0.02588	-0.03792	-0.02729
23-Feb-09	0.0173032	0.02956625	-0.03532	-0.01967	-0.00992	-0.00544	-0.01255	-0.01216	0.009678
24-Feb-09	0.0017254	-0.00415166	0.03932	-0.01039	-0.00895	-0.01469	0	-0.00945	-0.01188
25-Feb-09	0.0049759	0.0101978	-0.01072	-0.0128	0.008506	0.026161	0	-0.00099	0.006099
26-Feb-09	0.0006718	-0.00081201	-0.01591	0.024757	0.017155	-0.00044	-0.00136	0.020903	-0.00233
27-Feb-09	-0.005892	-0.00951845	-0.02385	-0.0254	-0.02208	0.014706	7.86E-05	-0.01826	-0.00615
2-Mar-09	-0.013689	-0.03132395	-0.04774	-0.0354	-0.05482	-0.03883	-0.05236	-0.04882	-0.03536
3-Mar-09	0.002193	0.00412295	-0.00643	-0.00523	-0.03186	-0.00695	0.00641	-0.0203	0.00129
4-Mar-09	0.0079132	0.00272274	0.023475	0.052829	0.037389	0.008435	0.051674	0.036548	0.030312
5-Mar-09	-0.038522	-0.04338157	-0.04346	-0.05154	-0.03233	0.01936	-0.02727	-0.03747	-0.00867
6-Mar-09	0.0079588	-0.01923487	0.001215	-0.0079	0.000227	-0.03566	-0.00709	-0.01117	-0.0021

Daily percentage change in ISE and other major stock exchanges

Can we explain the change in ISE as a function of the changes in other major exchanges?

Lasso regression: Example

- We separate the data into a training set and a test set.
- We'll fit (lasso) regressions on the training set.
- And test their performance on the test set.

Lasso regression: Example

Regression output: using all variables:

Estimated Coefficients:

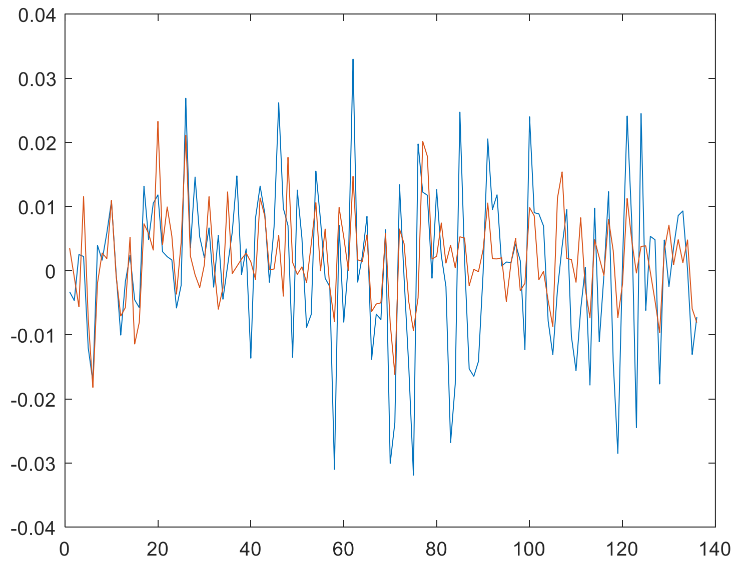
	Estimate	SE	tStat	pValue
(Intercept)	0.0013145	0.00063319	2.076	0.038543
x1	0.04298	0.067988	0.63218	0.52764
x2	-0.078836	0.11757	-0.67052	0.50292
x3	-0.18668	0.15067	-1.239	0.2161
x4	-0.040947	0.0522	-0.78443	0.43326
x5	-0.12772	0.066621	-1.9171	0.055951
x6	0.83532	0.20901	3.9966	7.6776e-05
x7	0.54117	0.11258	4.807	2.1867e-06

Number of observations: 400, Error degrees of freedom: 392
Root Mean Squared Error: 0.0125
R-squared: 0.49, Adjusted R-Squared 0.481
F-statistic vs. constant model: 53.9, p-value = 1.19e-53

Lasso regression: Example

And we evaluate the performance on the test data.

```
MSE1 = 9.9260e-05
```



Lasso regression: Example

- With only seven factors, we can eliminate some of the non-significant ones using the regression output but this requires manual work

```
Estimated Coefficients:
                   Estimate         SE         tStat       pValue
                  _________    __________    ________    __________

    (Intercept)   0.0013145    0.00063319       2.076      0.038543
    x1              0.04298      0.067988     0.63218       0.52764
    x2            -0.078836       0.11757    -0.67052       0.50292
    x3             -0.18668       0.15067      -1.239        0.2161
    x4            -0.040947        0.0522    -0.78443       0.43326
    x5             -0.12772      0.066621     -1.9171      0.055951
    x6              0.83532       0.20901      3.9966    7.6776e-05
    x7              0.54117       0.11258       4.807    2.1867e-06


Number of observations: 400, Error degrees of freedom: 392
Root Mean Squared Error: 0.0125
R-squared: 0.49,  Adjusted R-Squared 0.481
F-statistic vs. constant model: 53.9, p-value = 1.19e-53
```

Lasso regression: Example

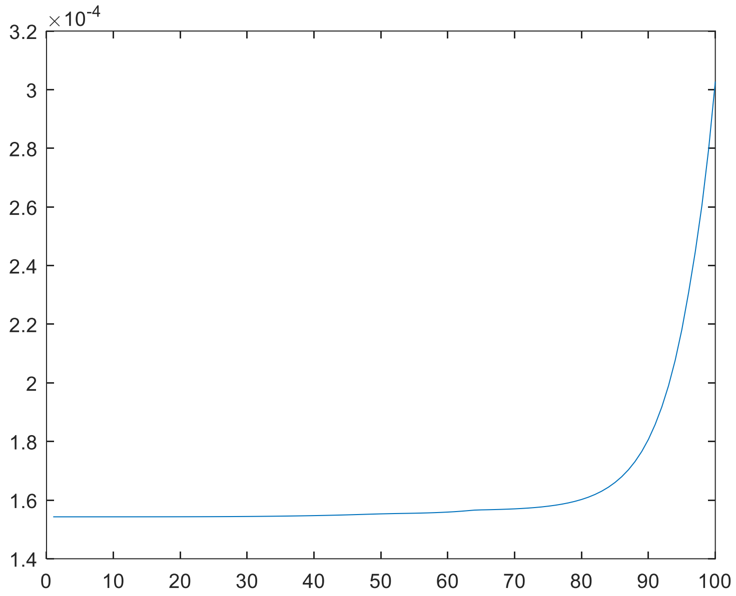
- We can automate model simplification (shrinkage) by using a lasso regression:

$$\min \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

- By default, statistical learning software automatically runs the lasso for varying value of the penalty parameter λ.
- We can then access the estimated coefficents for each value of λ and the corresponding MSE's.

Lasso regression: Example

- Let's take a look at the plot of MSE3 as function of the λ parameter.



- Note that as λ increases, the model becomes simpler (i.e. throws away some of the variables by setting their coefficients to zero), as a result the MSE grows.
- But we observe that there is no significant loss of MSE until λ exceeds its value at λ_{80}.

Lasso regression: Example

- Let's investigate the model for λ_{75}.

```
target=75;
coeff=B(:,target)

coeff =
         0
         0
         0
         0
         0
    0.5328
    0.3525
```

- Only x_6 and x_7 are significant for this model (for a penalty parameter λ_{75}).

Lasso regression

- Let's investigate the model for different penalty parameters λ_i.

λ_1	λ_{50}	λ_{75}	λ_{98}
`>> coeff=B(:,1)`	`>> coeff=B(:,50)`	`>> coeff=B(:,75)`	`>> coeff=B(:,98)`
`coeff =`	`coeff =`	`coeff =`	`coeff =`
0.0418	0.0003	0	0
-0.0765	0	0	0
-0.1818	-0.0016	0	0
-0.0402	-0.0068	0	0
-0.1262	-0.0672	0	0
0.8289	0.5992	0.5328	0.1396
0.5392	0.4573	0.3525	0

Lasso regression

- Let's now test the simplified model (λ_{75}) on the test data.

```
MSE_new = 9.8400e-05
```

- Recall that the MSE for the full model using seven predictors was:

```
MSE1 = 9.9260e-05
```

- Note that even though we are only using two predictors, the reduced model has a slightly lower MSE on the test set.
- It is likely to be much more stable.

Regularization: Constrained Optimization formulation

- Lasso and ridge have their constrained optimization counterparts:

Another Formulation for Ridge Regression and the Lasso

One can show that the lasso and ridge regression coefficient estimates solve the problems

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq s \tag{6.8}$$

and

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq s, \tag{6.9}$$

Regularization: Constrained Optimization formulation

- There are additional benefits of the constrained optimization formulation.

$$\min \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

subject to

$$-Mz_j \leq \beta_j \leq Mz_j \;\; \forall j$$

$$\sum_{j=1}^{p} z_j \leq k$$

$$z_j \in \{0,1\} \;\; \forall j, \quad \beta_j \in \Re, \forall j$$

Regularization: Constrained Optimization formulation

- For instance, one can easily eliminate correlated predictors using an additional constraint.
- Assume that predictor i and k have correlation (in absolute value) above a threshold. We can then add the constraint:

$$z_i + z_k \leq 1$$

- And we can repeat this for all pairwise correlated predictors.

This part is based on *Machine Learning under a Modern Optimization Lens* by Bertsimas and Dunn

Regularization: Constrained Optimization formulation

- One can easily control functional forms.
- For instance, we might be tempted to try t, $t^{4/3}$, $t^{5/3}$ and t^2 as predictors.
- But for robustness, we might prefer to use at most of the four predictors:

$$z_1 + z_2 + z_3 + z_4 \leq 1$$

This part is based on *Machine Learning under a Modern Optimization Lens* by Bertsimas and Dunn

Regularization: Constrained Optimization formulation

- Using more complicated and non-linear constraints:
  - Multi-collinearity can be handled
  - We can specify that only statistically significant predictors are used
- This constrained optimization framework is called holistic regression (Berstimas and Dunn, 2019).

This part is based on *Machine Learning under a Modern Optimization Lens* by Bertsimas and Dunn