INDR 450/550

Spring 2022

Lecture 14: Validation and Model Shrinkage

March 30, 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

A convention for the classification table

		True class		
		− or Null	+ or Non-null	Total
Predicted class	− or Null	True Neg. (TN)	False Neg. (FN)	N^*
	+ or Non-null	False Pos. (FP)	True Pos. (TP)	P^*
	Total	N	P	

TABLE 4.6. *Possible results when applying a classifier or diagnostic test to a population.*

Name	Definition	Synonyms
False Pos. rate	FP/N	Type I error, 1−Specificity
True Pos. rate	TP/P	1−Type II error, power, sensitivity, recall
Pos. Pred. value	TP/P^*	Precision, 1−false discovery proportion
Neg. Pred. value	TN/N^*	

TABLE 4.7. *Important measures for classification and diagnostic testing, derived from quantities in Table 4.6.*

The null hypothesis (more likely outcome) is placed in the top row.

Classification: KNN Classifier

- Same simple principle to KNN regression: find the most similar points (values of predictors) in the training data. The corresponding response is the prediction.
- Let's make a prediction for month 101 using the first three lag differences (-23, 975, 7087) (using Euclidean distances)

95	2401	1	-296	356	395
96	3291	1	986	-429	337
97	334	0	890	1282	-785
98	617	1	-2957	-96	1711
99	1875	1	283	-3847	-1378
100	1852	0	1258	3240	-3751
101	1938	1	-23	975	7087

Classification: KNN Classifier

- If K = 1, the nearest neighbour is month 77

101	1938	1	-23	975	7087

77	1403	1	-130	422	5967

- Our prediction for month 101 is 1 (‘up’)

Classification: KNN Classifier

- If K = 5, the nearest neighbours are months 77, 29, 17, 65, 89

101	1	-23	975	7087

77	1	-130	422	5967
29	1	-185	331	5090
17	1	209	2	5113
65	0	292	146	9467
89	1	-263	-411	4037

- We can now have richer predictions, in 4 of the 5 similar months, the market went up 4 times. We can then write $\mathrm{P}(\mathrm{Up}=4/5)$.
- If we assign weights to the distances we can compute a weighted estimate for the probability:

$$(P(\mathrm{up}) = 1\ (w_1 + w_2 + w_3 + w_5) + 0(w_4)),$$

where w_i is the weight for the i'th nearest month.

Classification Example: KNN classifier results

Train set: first 100 months, Test set: last 55 months

For K= 3, we obtain

$$\begin{bmatrix} 10 & 6 \\ 11 & 28 \end{bmatrix} \rightarrow \begin{bmatrix} 10 & 11 \\ 6 & 28 \end{bmatrix}$$

```
[[10  6]
 [11 28]]
              precision    recall  f1-score   support

         0.0      0.625     0.476     0.541        21
         1.0      0.718     0.824     0.767        34

    accuracy                          0.691        55
   macro avg      0.671     0.650     0.654        55
weighted avg      0.682     0.691     0.681        55
```

For K= 5, we obtain

Sklearn:

ISL book:

$$\begin{bmatrix} 9 & 5 \\ 12 & 29 \end{bmatrix} \rightarrow \begin{bmatrix} 9 & 12 \\ 5 & 29 \end{bmatrix}$$

```
[[ 9  5]
 [12 29]]
              precision    recall  f1-score   support

         0.0      0.643     0.429     0.514        21
         1.0      0.707     0.853     0.773        34

    accuracy                          0.691        55
   macro avg      0.675     0.641     0.644        55
weighted avg      0.683     0.691     0.674        55
```

These appear to be comparable to logistic regression results!

Validation Issues: validation set

- We already talked about separating the training set where the model is fit from the test set where the model is tested and the MSE is calculated.
- Since we have been looking at time series data, the natural split is $(1,t)$ vs $(t+1, T)$.
- But note that if the sequence were not important we could take a completely random sample as the train sample.
- This is known as the simple validation set approach for cross validation.
- The downside is that MSE of the test set may have high variance depending on the split.

This is material from Chapter 5 of James, Witten, Hastie, Thibshirani

Validation Issues: validation set

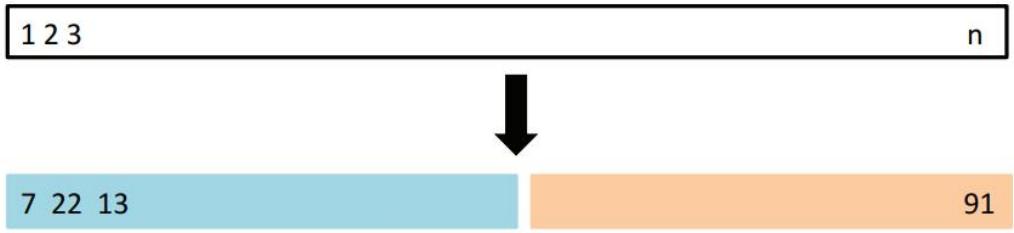


FIGURE 5.1. *A schematic display of the validation set approach. A set of n observations are randomly split into a training set (shown in blue, containing observations 7, 22, and 13, among others) and a validation set (shown in beige, and containing observation 91, among others). The statistical learning method is fit on the training set, and its performance is evaluated on the validation set.*

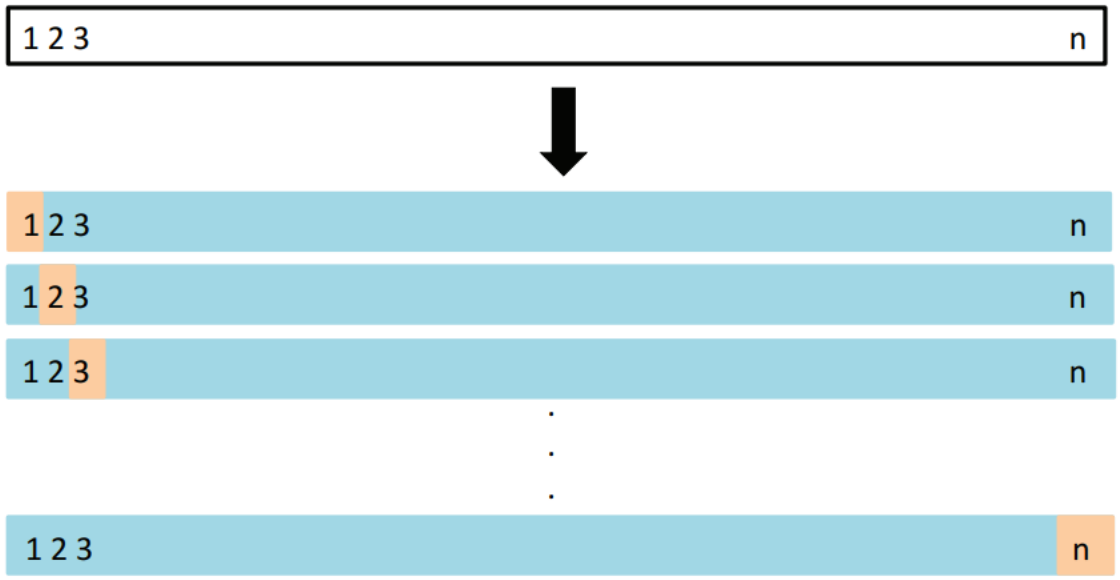
This is material from Chapter 5 of James, Witten, Hastie, Thibshirani

Validation Issues: leave one out cross-validation

- Similar in principle to the validation set approach but improves the robustness.
- In the validation test approach we take part of the data as the training set, and the rest as the test set.
- Here, we separate a single observation which becomes the mini test set, all other observations are the training set.
  - We fit a model on n-1 observations and test it on the single observation for MSE.
  - But then we do this for all individual observations.
  - We obtain n estimators of the error measure
  - Our ultimate estimator is the average of all individual errors.

This is material from Chapter 5 of James, Witten, Hastie, Thibshirani

Validation Issues: LOOCV



$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{MSE}_i.$$

FIGURE 5.3. *A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the n resulting MSE's. The first training set contains all but observation 1, the second training set contains all but observation 2, and so forth.*

This is material from Chapter 5 of James, Witten, Hastie, Thibshirani

Validation Issues: LOOCV

- LOOCV reduces the variance of the MSE estimator but it's relatively expensive (for a sample of size n, we have to fit n models).
- For a linear regression, it turns out there is a very efficient way to do this (we don't have a fit a regression each time but we fit an initial regression and update it with a simple formula for each different observation).
  - Unfortunately, there is no such easy way around for ARIMA, or logistic regressions.

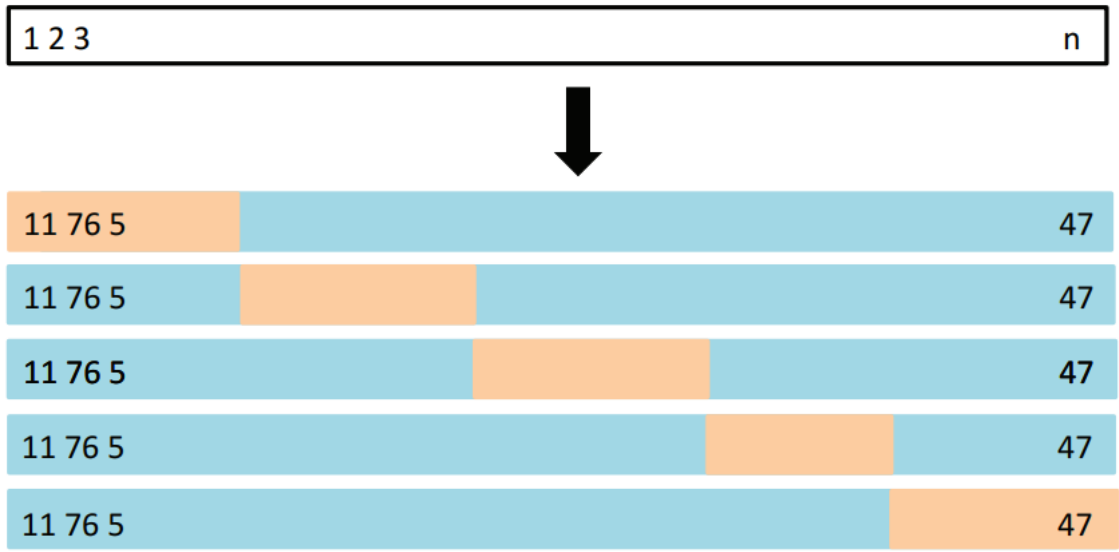
This is material from Chapter 5 of James, Witten, Hastie, Thibshirani

Validation Issues: k-fold cross validation

- LOOCV reduces the variance of the MSE estimator but it's relatively expensive. K-fold cross validation reduces the computational burden.
- We seperate the data into k groups (folds) of approximately equal size.
  - We fit the model on one of the k-folds and test its error on the other k-1.
  - We repeat this for all of the folds.
  - We then end up with k estimators for the MSE
  - Our ultimate estimator is the average of the k estimators.

This is material from Chapter 5 of James, Witten, Hastie, Thibshirani

Validation Issues: k-fold cross validation



$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\mathrm{MSE}_i.$$

FIGURE 5.5. *A schematic display of 5-fold CV. A set of n observations is randomly split into five non-overlapping groups. Each of these fifths acts as a validation set (shown in beige), and the remainder as a training set (shown in blue). The test error is estimated by averaging the five resulting MSE estimates.*

This is material from Chapter 5 of James, Witten, Hastie, Thibshirani

Validation Issues: k-fold cross validation

- This gives us control over the computational burden.
- If we choose to perform 10-fold cross validation, we fit the model only 10 times.
- This is important when model fitting is computationally hard.
- We still reduce the variance of the MSE estimator due to smoothing (i.e. Averaging multiple independent estimators).

This is material from Chapter 5 of James, Witten, Hastie, Thibshirani

Validation Issues: classification problems

- Recall that the MSE is not the criterion in classification problems.
- But we can apply similar ideas on the classification error rates.

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{Err}_i,$$

This is material from Chapter 5 of James, Witten, Hastie, Thibshirani

Model selection and regularization

- Linear regression is fairly robust when the sample size is n is much larger than the number of predictors p.
- But things get more complicated if p is large with respect to n.
  - The variance of the estimators is very high, most ordinary regressions perform poorly on test data
  - in particular, if $p > n$, then the system is overdetermined. There is not a unique fit.
- But this is the natural framework for applications in data science where we are willing to experiment with a larger number of predictors.
- It turns out that we need ways of shrinking (regularizing) the regression model in such cases.

This is material from Chapter 6 of James, Witten, Hastie, Thibshirani

Model selection and regularization: brute force best subset selection

- If the number of predictors is small, we can evaluate all possible combinations and compare them in terms of error performance.

Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

This is material from Chapter 6 of James, Witten, Hastie, Thibshirani

Model selection and regularization: brute force best subset selection example

- We try to predict the quarterly US consumption using four candidate predictors: income, unemployment, production and savings

```
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)   0.26729    0.03721    7.18  1.7e-11 ***
#> Income        0.71448    0.04219   16.93  < 2e-16 ***
#> Production    0.04589    0.02588    1.77    0.078 .
#> Unemployment -0.20477    0.10550   -1.94    0.054 .
#> Savings      -0.04527    0.00278  -16.29  < 2e-16 ***
#> ---
#> Signif. codes:
#> 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.329 on 182 degrees of freedom
#> Multiple R-squared:  0.754,  Adjusted R-squared:  0.749
#> F-statistic:  139 on 4 and 182 DF,  p-value: <2e-16
```

Model selection and regularization: brute force best subset selection example

Table 5.1: All 16 possible models for forecasting US consumption with 4 predictors.

Income	Production	Savings	Unemployment	CV	AIC	AICc	BIC	AdjR2
1	1	1	1	0.116	-409.3	-408.8	-389.9	0.749
1	0	1	1	0.116	-408.1	-407.8	-391.9	0.746
1	1	1	0	0.118	-407.5	-407.1	-391.3	0.745
1	0	1	0	0.129	-388.7	-388.5	-375.8	0.716
1	1	0	1	0.278	-243.2	-242.8	-227.0	0.386
1	0	0	1	0.283	-237.9	-237.7	-225.0	0.365
1	1	0	0	0.289	-236.1	-235.9	-223.2	0.359
0	1	1	1	0.293	-234.4	-234.0	-218.2	0.356
0	1	1	0	0.300	-228.9	-228.7	-216.0	0.334
0	1	0	1	0.303	-226.3	-226.1	-213.4	0.324
0	0	1	1	0.306	-224.6	-224.4	-211.7	0.318
0	1	0	0	0.314	-219.6	-219.5	-209.9	0.296
0	0	0	1	0.314	-217.7	-217.5	-208.0	0.288
1	0	0	0	0.372	-185.4	-185.3	-175.7	0.154
0	0	1	0	0.414	-164.1	-164.0	-154.4	0.052
0	0	0	0	0.432	-155.1	-155.0	-148.6	0.000

Model selection and regularization: brute force best subset selection example

- The best model contains all four predictors but all models in the top 4 rows have similar performance much better than the rest.
- It appears that income and savings are the most important variables to predict consumption. For prediction purposes we might want to go with one of the models in rows 2, 3 or 4.
- But to determine this we checked all possible models that have up to 4 variables (2^4=16 possible cases). We cannot possibly do this with 40 variables (2^{40}possible cases).

Model selection and regularization: stepwise selection

- **Forward selection:** choose one predictor first, the one that results in the highest R^2.
  - The next predictor to be added is the one that gives the greatest additional improvement.
  - Stop if the additional improvement is small or complete the list and compare.
- **Backward selection:** Start with full model (all p predictors).
  - Remove the least useful one (the one that results in the lowest reduction in R^2).
  - Repeat until a threshold reduction level is reached or the number of desired predictors is reached.

Model selection and regularization: stepwise selection

- Stepwise selection is effective but it is a greedy approach.
- As usual, we can do better by formal global optimization that somehow constraints the number of predictors to be used
  - Seeking a trade-off between the number of predictors and the MSE.

Regularization

- Recall the ordinary least squares objective function:

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

- The problem with the above formulation is that all coefficients are non-zero at optimality (i.e. optimization suggests using all predictors).
- But we would like to limit the number of predictors that are used.
  - impose a constraint on the number of predictors

Regularization: Constrained Optimization formulation

- Here ‘s a natural way to control the number of predictors from an Operations Research perspective.
- We use binary variables to distinguish those predictors that will be used at optimality.
- This results in an integer optimization problems.
- Recent advances have made integer optimization extremely powerful.

Regularization: Constrained Optimization formulation

- The optimization formulation is an integer program with a quadratic objective function.
- Today, such problems can be solved to optimality easily.

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

subject to

$$-M z_j \leq \beta_j \leq M z_j \quad \forall j$$

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

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

Regularization: Constrained Optimization formulation

- To my knowledge, this formulation is not yet implemented in usual libraries.
- It has some other attractive modeling properties as we will see later
- But let’s first look at the statistical learning tools.

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.