Data Analysis and Visualization:

Supervised learning - regression (2/2)

Erik-Jan van Kesteren
Peter van der Heijden

Department of Methodology and Statistics

Universiteit Utrecht

Utrecht Applied Data Science

Important concepts last time

- Prediction function
- k-nearest neighbors (KNN)
- Metrics for model evaluation
- Bias and variance (tradeoff)
- Training-validation-test set paradigm (or “Train/dev/test”)

Bias-variance tradeoff in training-test error

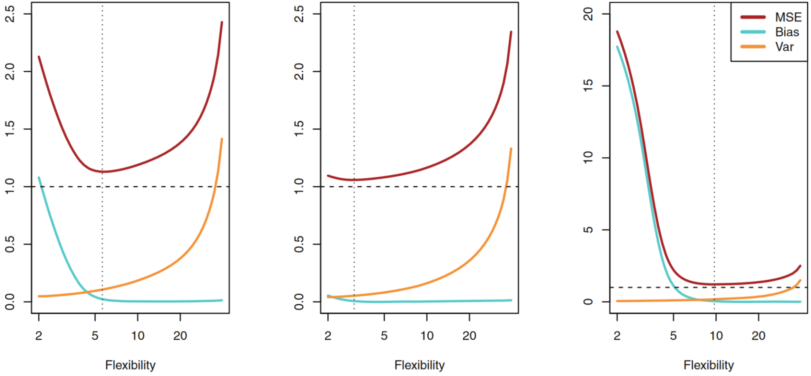
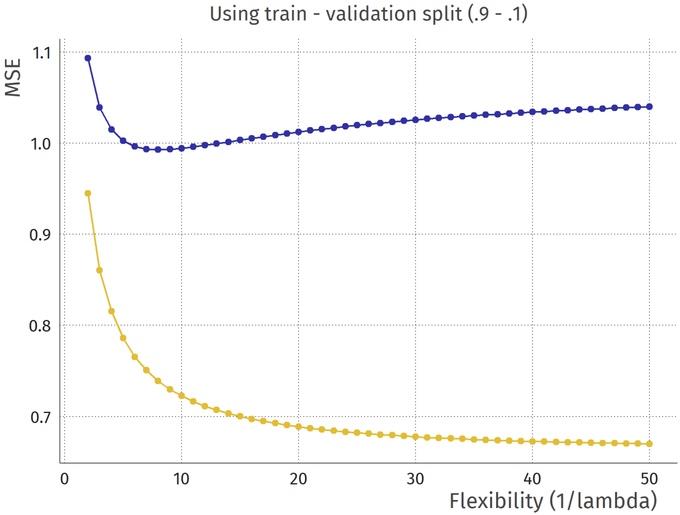


FIGURE 2.12. *Squared bias (blue curve), variance (orange curve), Var(ϵ) (dashed line), and test MSE (red curve) for the three data sets in Figures 2.9–2.11. The vertical dotted line indicates the flexibility level corresponding to the smallest test MSE.*

Ridge regression bias-variance tradeoff



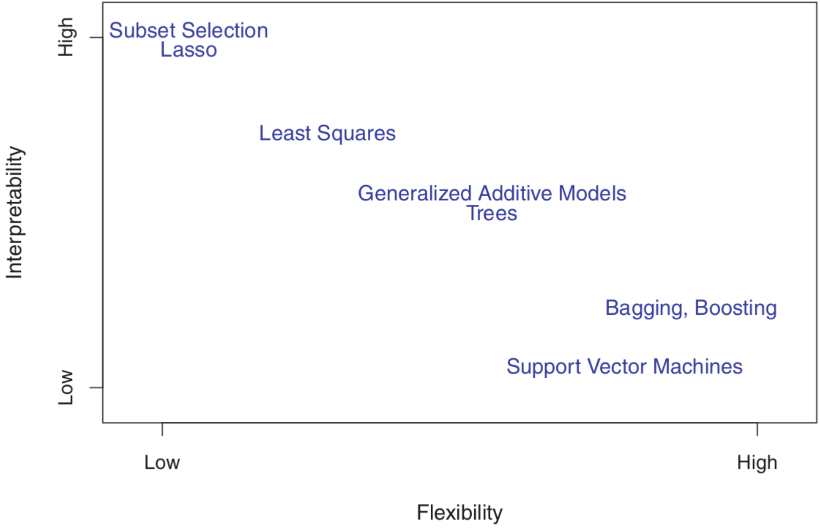
Questions about last week

1. In which situations is a parametric model, such as linear regression, better than KNN?
2. Many cities nowadays have a bike share system. Suppose you were asked to predict how many bikes are rented on a given day. It is more expensive to disappoint a customer than it is to have bikes left over at the end of the day. What would be an appropriate error measure and why?
3. In winter, typically between 0 and 17% of bee colonies in a hive die. A regression model predicting this percentage mortality gave training MAE $= 29.5$ and test MAE $= 30.2$. Is this model: (A) High variance; (B) High bias; (C) Both; (D) Neither.
4. A different model on the same data gave training MAE $= 1.3$ and test MAE $= 19.2$. Is this model: (A) High variance; (B) High bias; (C) Both; (D) Neither.

Important concepts today

- Feature selection
- Regularization
- Model flexibility
- Bias-variance tradeoff

Flexibility – interpretability tradeoff

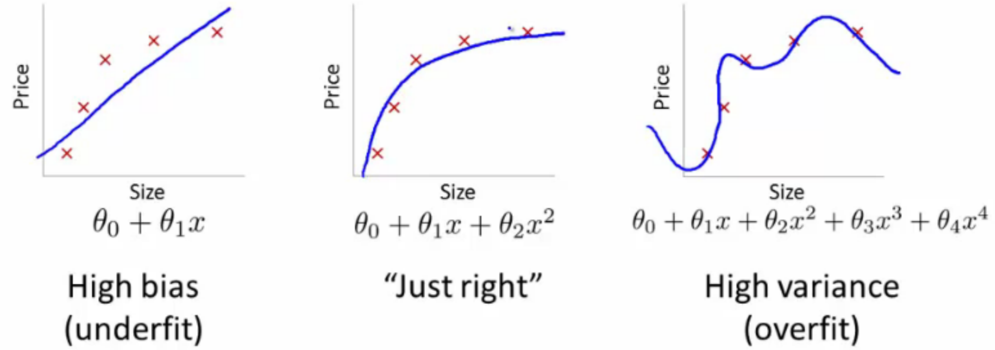


Feature selection/penalization

The bias-variance tradeoff again

- More flexibility is good: lower bias
- More flexibility is bad: higher variance

What if we made a very flexible model, but told it not to go overboard with the complexity (judging that by validation data)?



- A very flexible model is like a kid in candystore, with a platinum credit card:
- It goes around buying all the coefficients it wants and never stops.

Three ways to educate your child

- **Subset selection**
  - "You can buy at most *p* things"
  - $\rightarrow$ Pick the *p* best predictors of the model (*wrapper*)
  - "You can buy only the things you like more than *r*"
  - $\rightarrow$ Only pick predictors that correlate more than *r* with *y* (*filter*)
- **Shrinkage ("penalization", "regularization")**
  - "You can buy what you want, but don't spend more than €*s*".
  - $\rightarrow$ Keep the sum of squared (L2) or absolute (L1) coefficients below some budget *s*, for example $\sum_j \beta_j^2 \leq s$ ("ridge") (*embedded*)
- **Dimension reduction ($\rightarrow$ unsupervised learning)**
  - "We'll make *p* cookies out of a little bit of all the things and you can buy those."
  - $\rightarrow$ Run an unsupervised model first, then predict *y* from the resulting *p* scores.

Wrapper

Three common algorithms for subset selection *ISLR, p. 205-209*

- Best subset selection
- Forward stepwise
- Backward stepwise

Each of these fits several models and chooses the “best” among these.

Best subset selection

1. Fit all possible models with at most p predictors
2. Choose the "best" one (using your metric of choice)

How many models?

$$\sum_{k=0}^{k=p} \binom{p}{k} = 2^p \text{ possible models}$$

with $p = 20$ predictors, that's more than a million models!

Forward stepwise

ISLR, page 207

1. Let $\mathcal{M}_0$ denote the null model, which contains no predictors.
2. For $k = 0, ..., p-1$:
   (a) Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor
   (b) Choose the best among these $p-k$ models, and call it $\mathcal{M}_{k+1}$.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$.

with $p = 20$ predictors, that's 211 models to estimate. Much more reasonable.

Backward stepwise

ISLR, page 209

1. Let $\mathcal{M}_p$ denote the full model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain $k-1$ predictors in $\mathcal{M}_k$.
   (b) Choose the best among these k models, and call it $\mathcal{M}_{k-1}$.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$.

with $p = 20$ predictors, that's also 211 models to estimate.

Wrapper feature selection, pros and cons

- **Best subset**

\+ *Exhaustive search*: Finds the best subset, as advertised – when there is enough data to find it

– Need to fit 2^p models, e.g. with 20 predictors that is 1,048,576 regressions to run and evaluate. Not even mentioning squares, cubes, products, etc. You'll run out of validation data quickly too.

- **Forward/backward**

\+ Much more efficient, $O(p^2)$ instead of $O(2^p)$, e.g. 211 models for 20 predictors

– *Greedy search*: Not guaranteed to find the best subset (*why not?*).

Forward vs. backward

- Forward usually more efficient
- Backward sometimes not even possible (e.g. $p > n$)
- Forward can be fooled, especially when two variables work together but do nothing alone:
- Backward considers performance of variables together with others.

- Both backward and forward are well-known to be **bad** at finding "true" subset of predictors

$\rightarrow$ Reviled in several fields (e.g. social science);

- For prediction goal, we do not care about the "true" subset.

Filter

Univariate filters

- Highest p correlations with y
- All predictors with correlation above threshold r
- (other measures: p-value, MDL, mutual information, ...)

mtcars example: filter using absolute correlations

```
abs(cor(mtcars)[-1,1]) %>% sort(decreasing = TRUE)

#> wt        cyl       disp      hp        drat
#> 0.8676594 0.8521620 0.8475514 0.7761684 0.6811719

#> vs        am        carb      gear      qsec
#> 0.6640389 0.5998324 0.5509251 0.4802848 0.4186840
```

Embedded

Regularization: buying coefficients on a budget

- The algorithm wants to fit the training data, by buying coefficients at the cost of variance
- Make the child behave "regular"ly by penalizing the purchase of "too many" coefficients
- Extremely efficient way to approximately solve the best subset problem
- Often yields very good results

Ordinary least squares (OLS) regression

Find the β_j that minimizes

$$\begin{aligned} \text{MSE} &= n^{-1} \sum_i (y_i - \hat{y}_i)^2 \\ &= n^{-1} \sum_i (y_i - (\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i}))^2 \end{aligned}$$

Penalized (regularized) regression

Find the β_j that minimizes

$$\begin{aligned}\text{MSE} &= n^{-1}\sum_i (y_i - \hat{y}_i)^2 + \lambda \cdot \text{Penalty} \\ &= n^{-1}\sum_i (y_i - (\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i}))^2 + \lambda \sum_{j>0} \beta_j^2\end{aligned}$$

or

$$= n^{-1}\sum_i (y_i - (\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i}))^2 + \lambda \sum_{j>0} |\beta_j|$$

Penalties as a "budget" of coefficients

Equivalently, we can see the regularized regression as: find the β_j that minimizes

$$\text{MSE} = n^{-1} \sum_i (y_i - \hat{y}_i)^2$$

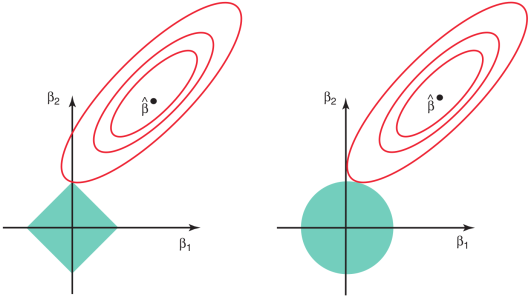
Subject to (LASSO):

$$\sum_{j>0} |\beta_j| \leq s$$

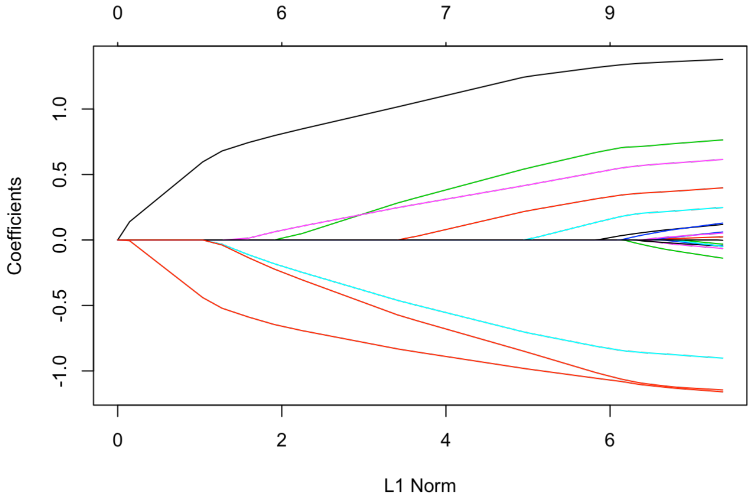
So "don't spend more than s on coefficients"

Different penalties

- "LASSO", L1: Penalize $\sum_{j>0} |\beta_j|$
- "Ridge", L2: Penalize $\sum_{j>0} \beta_j^2$



Penalization as “shrinkage” to zero



LASSO:

```
fit <- glmnet(x, y, alpha = 1, lambda = 1.5)
```

Ridge:

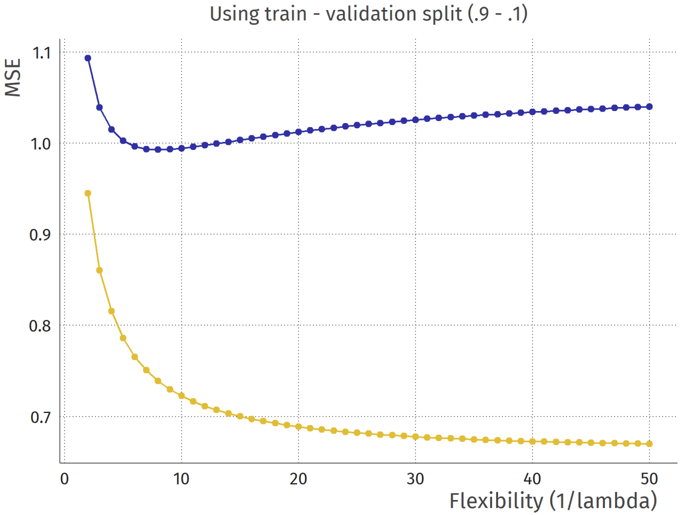
```
fit <- glmnet(x, y, alpha = 0, lambda = 0.01)
```

- LASSO and ridge have a tuning parameter λ ;
- The usual least squares is $\lambda = 0$;
- Higher $\lambda \rightarrow$ stricter penalty $\rightarrow$ smaller budget *s*;
- Higher λ “shrinks” coefficients to 0

mtcars example: penalization using `glmnet`

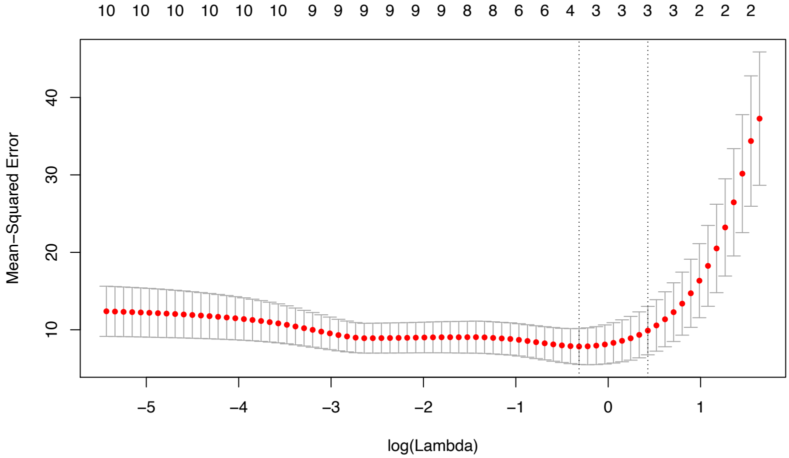
	Least squares	LASSO	Ridge
(Intercept)	12.303	33.593	21.198
cyl	-0.111	-0.836	-0.342
disp	0.013	.	-0.005
hp	-0.021	-0.006	-0.012
drat	0.787	.	1.034
wt	-3.715	-2.308	-1.438
qsec	0.821	.	0.189
vs	0.318	.	0.662
am	2.520	.	1.821
gear	0.655	.	0.565
carb	-0.199	.	-0.619

Ridge regression bias-variance tradeoff



Selecting λ with cross-validation

LASSO: `cvfit ← cv.glmnet(x, y, alpha = 1)`



Important concepts today

- Feature selection
- Regularization
- Model flexibility
- Bias-variance tradeoff

Conclusion

- There is a tradeoff between model complexity and interpretability
- Feature selection makes a model simpler
- Feature selection categories: filter, wrapper, embedded
- Regularization/penalization as an embedded form of feature selection: shrinkage
- Model complexity tuned using model accuracy estimate, e.g., k-fold cv
- Tuning model complexity $\rightarrow$ optimizing bias-variance tradeoff