FIT2086 Lecture 8
Model Selection and Penalized Regression

Daniel F. Schmidt

Faculty of Information Technology, Monash University

September 10, 2018

Outline

Revision from last week (1)

- We also have p predictor variables $X_1, \ldots, X_p$
- We can build a classifier for Y using our predictors, i.e., we want to find
$$\mathbb{P}(Y = y \,|\, X_1 = x_1, X_2 = x_2, \ldots, X_p = x_p)$$
- Naïve Bayes assumes predictors are conditionally independent, given the value of the target
$$P(x_1, \ldots, x_p \,|\, y) = \prod_{j=1}^{p} P(x_j \,|\, y)$$
- Naïve Bayes classifier:
$$P(y \,|\, x_1, \ldots, x_p) = \frac{\prod_{j=1}^{p} P(x_j \,|\, y) P(y)}{P(x_1, \ldots, x_p)}$$

Revision from last week (2)

- A logistic regression models the conditional log-odds as

$$\log\left(\frac{\mathbb{P}(Y_i = 1 \,|\, x_{i,1}, \ldots, x_{i,p})}{\mathbb{P}(Y_i = 0 \,|\, x_{i,1}, \ldots, x_{i,p})}\right) = \beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j} \equiv \eta_i$$

- Logistic regression model of conditional probability

$$\mathbb{P}(Y_i = 1 \,|\, x_{i,1}, \ldots, x_{i,p}) = \frac{1}{1 + \exp(-\eta_i)}$$

- Estimated using maximum likelihood
- Performance measures for classification
  - Classification error
  - Sensitivity and specificity
  - Area-under-the-curve (AUC)
  - Logarithmic loss

Outline

Supervised Learning – recap

- Imagine we have measured $p+1$ variables on n individuals (people, objects, things)
- One variable is our target
- We would like to predict our target using the remaining p variables (predictors)
- If the variable we are predicting is categorical, we are performing classification
- If the variable we are predicting is numerical, we are performing regression

Underfitting/Overfitting Example (1)

- We often have many measured predictors
  - Should we use them all, and if not, why not?
- Omitting important predictors
  - Called underfitting
  - Leads to systematic error ("bias") in predicting the target
- Including spurious predictors
  - Called overfitting
  - Leads our model to "learn" noise and random variation
  - Poorer ability to predict to new, unseen data from our population

Underfitting/Overfitting Example (2)

- Example: we observe x and y data and want to build a prediction model for y using x
  - Data looks nonlinear so we use polynomial regression
  - We take $x, x^2, x^3, \ldots, x^{20} \Rightarrow$ very flexible model
  - How many terms to include?
- For example, do we use

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon$$

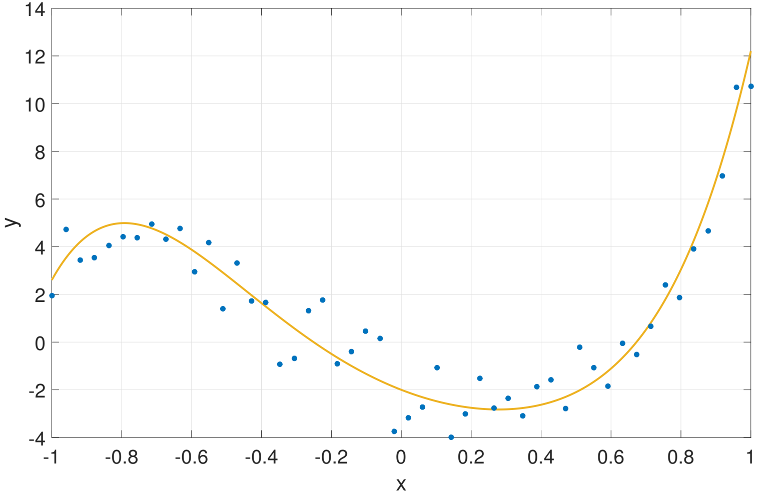
or

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4 + \beta_5 x^5 + \varepsilon$$

or another model with some other number of polynomial terms.

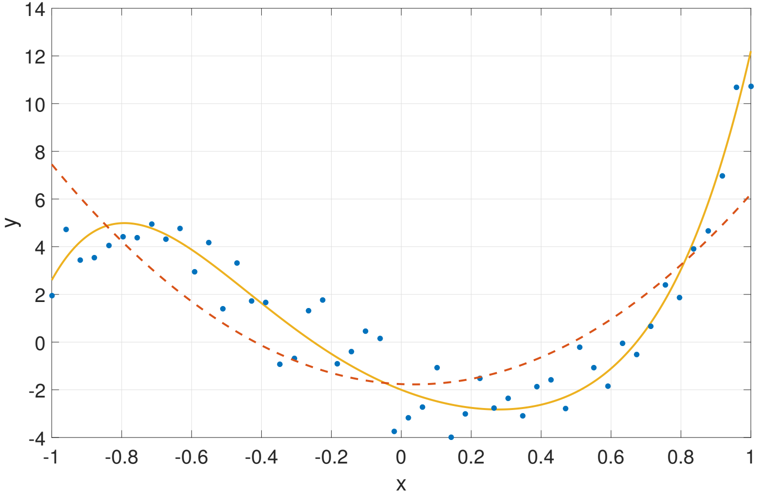
Underfitting/Overfitting Example (3)

- Example dataset of 50 samples



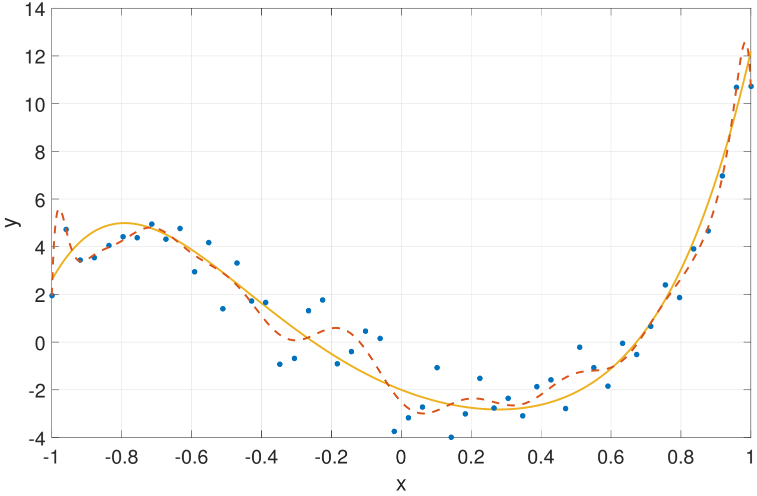
Underfitting/Overfitting Example (4)

- Use (x, x^2), too simple – underfitting



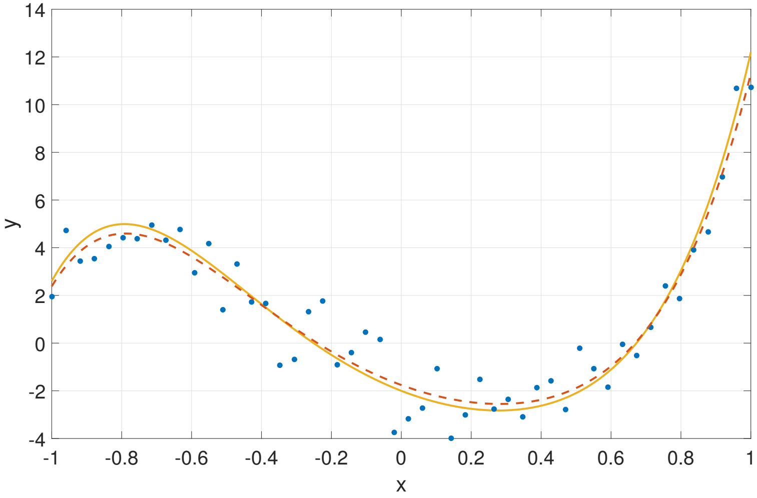
Underfitting/Overfitting Example (5)

- Use $(x, x^2, \ldots, x^{20})$, too complex – overfitting



Underfitting/Overfitting Example (6)

- $(x, x^2, \ldots, x^5)$ seems "just right". But how to find this model?



Mean-Squared Prediction Error (1)

- Introduce the idea of a true, underlying (population) model $\implies$ this is the model that "generated" the data
- For regression:

$$\mathbb{E}\left[Y \,|\, x_1, \ldots, x_p\right] = f(x_1, \ldots, x_p)$$

- For binary classification:

$$\log\left(\frac{\mathbb{P}(Y = 1 \,|\, x_1, \ldots, x_p)}{1 - \mathbb{P}(Y = 1 \,|\, x_1, \ldots, x_p)}\right) = f(x_1, \ldots, x_p)$$

- In real life, $f(\cdot)$ could be any function
- For simplicity, let's focus on regression

Mean-Squared Prediction Error (2)

- We receive a sample of data from our population $\mathbf{y} = (y_1, \ldots, y_n)$
  - This is our training data
- We use this to build a model $\hat{\mathcal{M}}(\mathbf{y}) \equiv \hat{\mathcal{M}}$ to estimate $f(\cdot)$
- In linear regression, a model $\mathcal{M}$ is a particular linear combination of
  - predictors $x_1, \ldots, x_p$, and
  - non-linear transformations of the predictors
- We would like our model to predict well onto new data
  - We call this the model's ability to generalise
- How can we measure this?

Mean-Squared Prediction Error (3)

- Imagine we receive new data $\mathbf{y}' = (y'_1, \ldots, y'_m)$ from our population
  - This is called testing data
- Once we have a model, $\hat{\mathcal{M}}(\mathbf{y})$, we can make predictions

$$\hat{y}_i(\hat{\mathcal{M}}(\mathbf{y})) \equiv \hat{y}(x'_{i,1}, \ldots, x'_{i,p}, \hat{\mathcal{M}}(\mathbf{y}))$$

for the value of y'_i, given predictors $x'_{i,1}, \ldots, x'_{i,p}$:

- We can measure how well our model predicts the new data using

$$\text{MSPE}(\hat{\mathcal{M}}(\mathbf{y})) = \frac{1}{m} \sum_{i=1}^{m} (y'_i - \hat{y}_i(\hat{\mathcal{M}}(\mathbf{y})))^2$$

which is the mean-squared prediction error on our testing data

Mean-Squared Prediction Error (4)

- Remember that our predictions $\hat{y}(x_1, \ldots, x_p, \hat{\mathcal{M}}(\mathbf{y}))$ are produced by a model that we learned using the training data $\mathbf{y}$
  $\Rightarrow$ they depend on the training data that we have observed
- If we sampled different training data from our population, we would learn a different model, $\hat{\mathcal{M}}(\mathbf{y})$, and get different predictions $\hat{y}_i(\hat{\mathcal{M}}(\mathbf{y}))$
- To remove this dependency, we can *average* the $\mathrm{MSPE}(\hat{\mathcal{M}}(\mathbf{y}))$ over all the different models we could learn from all the different possible training samples we could draw from our population
- We can write this *expected* mean-squared prediction error as

$$\mathrm{EMSPE} = \mathbb{E}\left[\mathrm{MSPE}(\hat{\mathcal{M}}(\mathbf{y}))\right] = \mathrm{bias}^2 + \mathrm{variance}$$

  where
  - bias measures systematic error of our predictions;
  - variance measures the error incurred through random variation

Mean-Squared Prediction Error (4)

- Remember that our predictions $\hat{y}(x_1, \ldots, x_p, \hat{\mathcal{M}}(\mathbf{y}))$ are produced by a model that we learned using the training data $\mathbf{y}$
  $\Rightarrow$ they depend on the training data that we have observed
- If we sampled different training data from our population, we would learn a different model, $\hat{\mathcal{M}}(\mathbf{y})$, and get different predictions $\hat{y}_i(\hat{\mathcal{M}}(\mathbf{y}))$
- To remove this dependency, we can *average* the $\mathrm{MSPE}(\hat{\mathcal{M}}(\mathbf{y}))$ over all the different models we could learn from all the different possible training samples we could draw from our population
- We can write this *expected* mean-squared prediction error as

$$\mathrm{EMSPE} = \mathbb{E}\left[\mathrm{MSPE}(\hat{\mathcal{M}}(\mathbf{y}))\right] = \mathrm{bias}^2 + \mathrm{variance}$$

  where
  - bias measures systematic error of our predictions;
  - $\mathrm{variance}$ measures the error incurred through random variation

Underfitting/Overfitting (1)

- We can demonstrate these ideas on our example
- The true model for this example was

$$\mathbb{E}\left[y \,|\, x\right] = f(x) = 9.7\,x^5 + 0.8\,x^3 + 9.4\,x^2 - 5.7\,x - 2$$

- To generate a sample from this true model:
  - I random sampled $\mathbf{x} = (x_1, \ldots, x_{50})$ uniformly from $(-1, 1)$
  - For each x_i value, I generated the samples y_i using

$$y_i = f(x_i) + \varepsilon$$

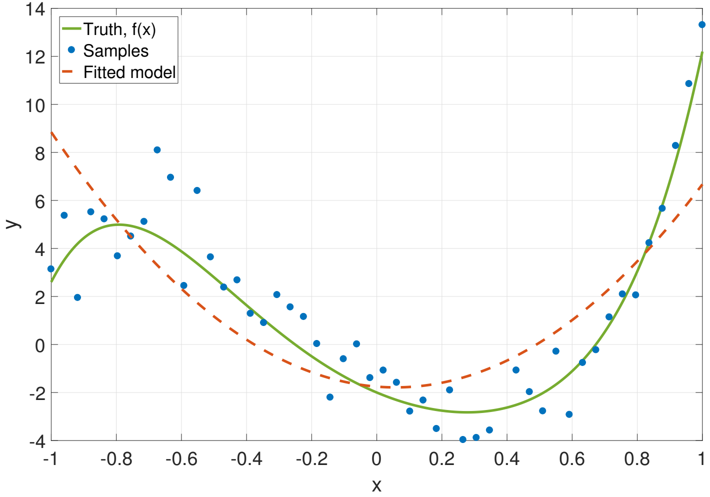
  where $\varepsilon_i \sim N(0, 1)$.

Underfitting and Bias (1)

- When underfitting, our model is too simple to capture the truth
- In linear regression, this means we are omitting important predictors
- For our example, imagine we fit a quadratic polynomial to our data
    - The "truth" is a fith-order polynomial
- Regardless of how much training data from the population we see, we cannot make a quadratic look like a fifth order polynomial
- The inability of our model to fit the truth well leads to systematic error in predicting $f(\cdot)$
- The increase in EMSPE introduced by underfitting is therefore primarily due to bias

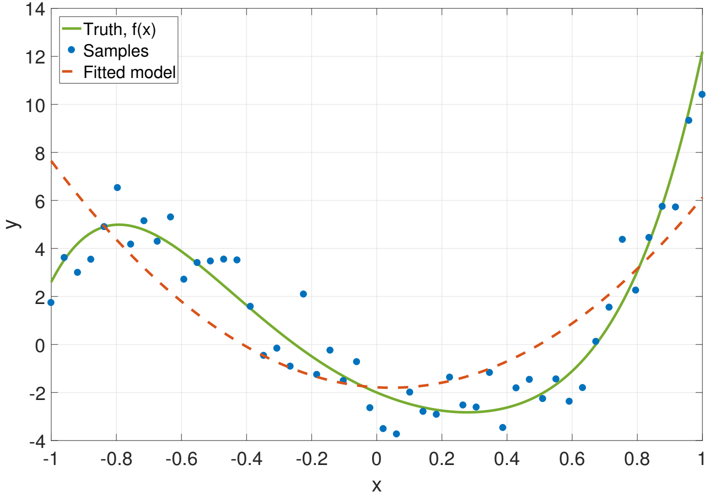
Underfitting and Bias (2)

- Using $\mathcal{M} = \{x, x^2\}$ on a new sample – too simple



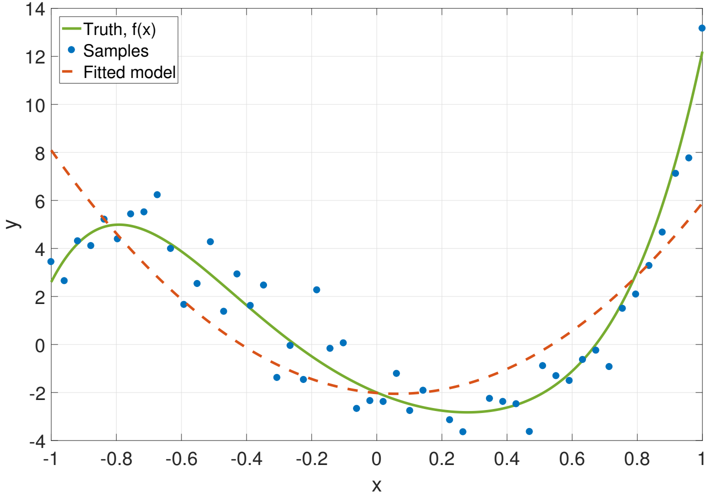
Underfitting and Bias (3)

- Again, using $\mathcal{M} = \{x, x^2\}$ – again, too simple



Underfitting and Bias (4)

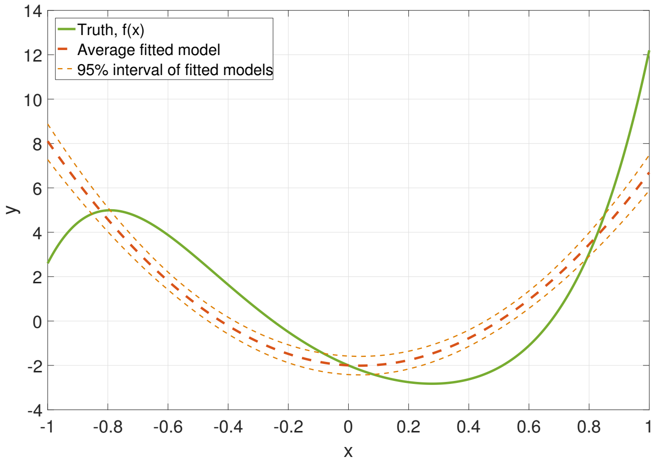
- Again, using $\mathcal{M} = \{x, x^2\}$ – once more, too simple



Underfitting and Bias (5)

- Bias and variance were determined by simulation
- Generated $10,000$ samples of size $n = 50$ from our true model (our training data)
- For each sample, fitted a 2nd order polynomial, i.e., x, x^2 terms
- Produced predictions for $5,000$ equally spaced x' values in $(-1, 1)$ (our testing data) for each of the $10,000$ different models
  - Calculated the average prediction at each x' value
  - Calculated the variance of predictions at each x' value
  - Used these to get bias^2 and $\mathrm{variance}$
- For visualisation:
  - Calculated the average prediction at each x' value
  - Calculated 2.75 and 97.5 percentiles of predictions at each x'

Underfitting and Bias (6)



Fitting 2nd order polynomial to data generated from a 5th order polynomial; average fit is poor (high bias) but does not vary greatly (low variance).

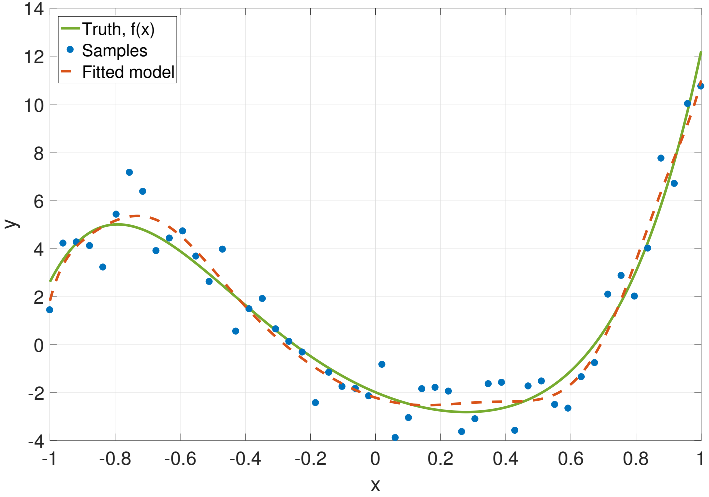
$\text{EMSPE} = 3.33$, $\text{bias}^2 = 3.27$, $\text{variance} = 0.058$

Overfitting and Variance (1)

- When overfitting, our model is more complex than the truth
- In linear regression, this means we are including unassociated predictors
- The extra predictor introduces a relationship between predictor and target that is not really there
  $\Rightarrow$ we learn random "noise" from our training data
- In our example, imagine we fitted a 7th order polynomial to our data
  - The "truth" is a 5th order polynomial, so the coefficients for the 6th and 7th order polynomial terms should be zero
  - But when we learn them from our training data, due to random chance, they will never be exactly equal to zero
- The increase in EMSPE introduced by overfitting is therefore primarily due to variance

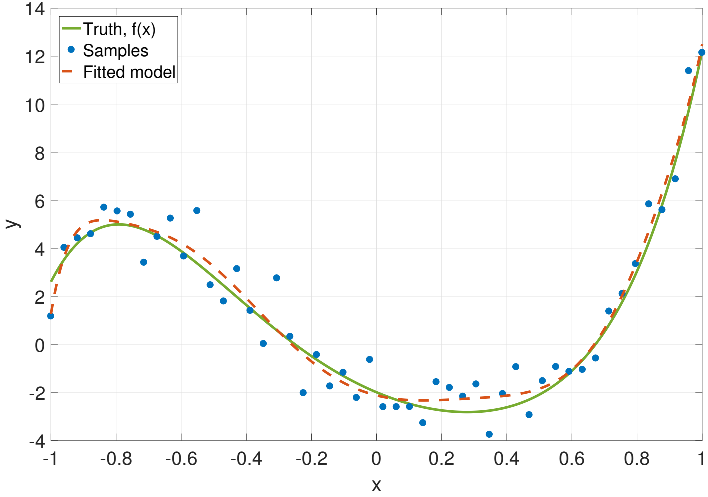
Overfitting and Variance (2)

- Using $(x, x^2, \ldots, x^7)$ on a new sample – too complex



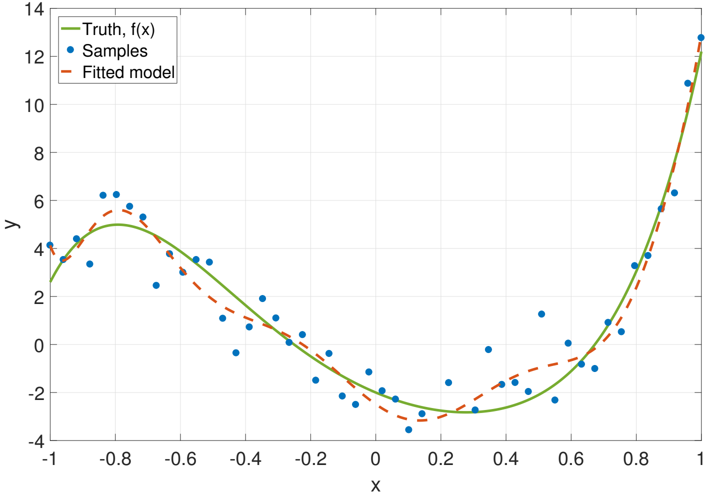
Overfitting and Variance (3)

- Again, using $(x, x^2, \ldots, x^7)$ – too complex

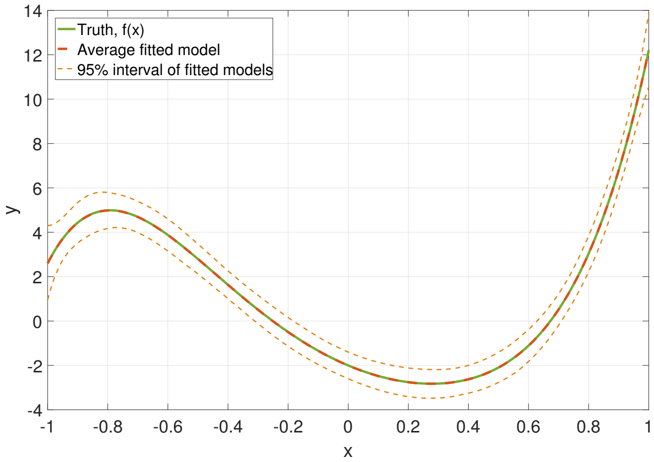


Overfitting and Variance (4)

- Again, using $(x, x^2, \dots, x^7)$ – too complex

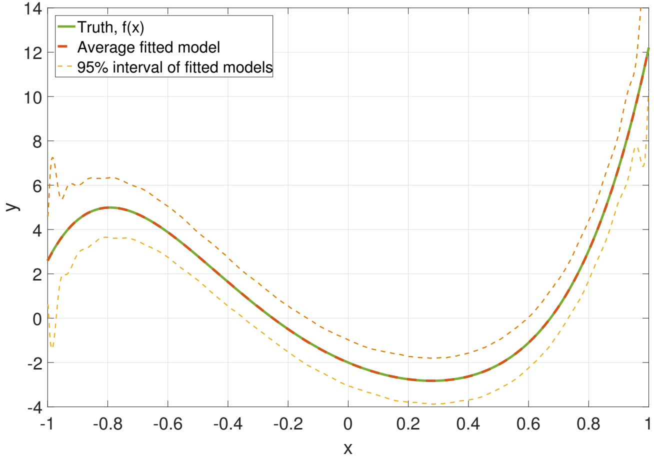


Overfitting and Variance (5)



Fitting 7th order polynomial to data generated from a 5th order polynomial; average fit is good (no bias) but does varies more than 2nd order polynomial.
$\mathrm{EMSPE} = 0.1470, \mathrm{bias}^2 = 0, \mathrm{variance} = 0.1470$

Overfitting and Variance (6)



Fitting 20th order polynomial to data generated from a 5th order polynomial; average fit is good (no bias) but variation is higher, especially near boundaries.
$\text{EMSPE} = 0.4860, \text{bias}^2 = 0, \text{variance} = 0.4860$

Underfitting/Overfitting (1)

- In general:
  - Simpler models have higher bias, lower variance
  - More complex models have lower bias, higher variance
  - Variance decreases with increasing n, bias does not
  - Variance increases with increasing number of predictors
- Recall EMSPE can be written as:

$$\text{EMSPE} = \text{bias}^2 + \text{variance}$$

- Model selection for prediction is about trading these off
  $\Rightarrow$ select the right level of complexity for the data
- In linear/logistic regression, “complexity” is increased by including more predictors, or transformations of predictors
  - Omitting a predictor will reduce variance, but may lead to bias if predictor is important

Underfitting/Overfitting (2)

- Model selection is quite subtle
- In linear models, including all predictors that are associated with target may not be enough to remove bias
  - If relationship between target and predictor is nonlinear, also need to include the right transformations ...
- You can overfit and underfit at the same time
  - Omit important predictor, include unassociated predictor ...
- Sometimes can do better by omitting associated predictors!
  - If association is weak and noise is large, may be too hard to get a good estimate of the coefficient β_j
  - Ergo, may be better to omit it
- Combining tools with real world knowledge/thinking can often improve predictions

Hypothesis Testing to Select Predictors (1)

- Recap: can select predictors using hypothesis testing
  $\Longrightarrow$ we know that a predictor j is unimportant if $\beta_j = 0$
- So we can test the hypothesis:

$$\begin{aligned} H_0 &: \quad \beta_j = 0 \\ &\text{vs} \\ H_A &: \quad \beta_j \neq 0 \end{aligned}$$

  which, in this setting is a variant of the t-test (see Ross, Chapter 9 and Studio 6)
- Imagine we tried fitting each predictor by itself to the data
  - Could choose to include predictor if p-value was less than 0.05

Hypothesis Testing to Select Predictors (2)

- If we were testing one hypothesis, this would guarantee probability of false discovery to be 5%
  - If predictor is unassociated, we would incorrectly conclude it to be associated for 5% of possible samples from our population
- But if we have p predictors, we need to do p tests
- So, even if all predictors were unassociated, would expect $0.05p$ tests to have p-values < 0.05 just by chance
- If p is large, this can be big, i.e., if $p = 1000$, we would falsely believe 50 predictors were associated just by chance on average
- This can easily happen if we include lots of transformations; e.g., trying all squares, logarithms, interactions of predictors
- Sometimes called data dredging

Hypothesis Testing to Select Predictors (3)

- This is called the multiple testing problem
- We examine one way to resolve it: the Bonferroni procedure
- Let α be our significance level
- This says we should instead reject null hypothesis only if

$$p\text{-value} < \frac{\alpha}{p}$$

- So if $\alpha = 0.05$ and $p = 1000$, we would need to see p-values of $0.05/1000 = 5 \times 10^{-5}$ to believe that a predictor is associated with the target

Hypothesis Testing to Select Predictors (4)

- Bonferroni guarantees the probability of making at least one false discovery is 5%
- Very conversative approach, but can be confident that predictors we identify are very likely truly associated
- Related method called false discovery rate (FDR)
  $\Rightarrow$ more power to discover associated predictors

Likelihood

- The likelihood $p(\mathbf{y} \,|\, \hat{\mathcal{M}})$ is a general measure of goodness of fit
- The likelihood quantifies how much probability our model assigns to the data we have observed
  $\Longrightarrow$ more probability, the more compatible the data is with our model
- Given a probability distribution $p(y \,|\, \theta)$ and estimates $\hat{\theta}$ for the parameters, compute likelihood by:
  1. Plug our estimated parameters $\hat{\theta}$ into our probability distribution
  2. Evaluate the probability assigned to each $y_1, \ldots, y_n$ by this model, i.e.,

$$p(\mathbf{y} \,|\, \hat{\theta}) = \prod_{i=1}^{n} p(y_i \,|\, \hat{\theta})$$

- We have seen likelihoods of
  - Simple Gaussian, Poisson, Bernoulli models (Lecture 3, Studio 3)
  - Linear regression models (Lecture 6)
  - Logistic regression models (Lecture 7)

Model Selection Criteria (1)

- We can use adjusted likelihood to guide selection of models
  $\Longrightarrow$ can control complexity through model selection criteria
- A model selection criterion assigns a score to a model which
  - takes into account how well the model fits the data (likelihood)
  - takes into account how complex the model is
- Search for the model that minimises model selection criterion
  $\Rightarrow$ in linear/logistic regression, find good set of predictors
- Let
  - $\mathcal{M}$ denote a model (set of predictors to use);
  - $\hat{\mathcal{M}}$ denote the model fitted to data $\mathbf{y}$;
  - $L(\mathbf{y} \,|\, \hat{\mathcal{M}})$ denote the minimised negative log-likelihood for the model $\mathcal{M}$
- Neg-log-likelihood always decreases with more predictors

Model Selection Criteria (2)

- Penalize neg-log-likelihood of model by a complexity penalty
- The Akaike information criterion (AIC):

$$\mathrm{AIC}(\hat{\mathcal{M}}) = L(\mathbf{y} \,|\, \hat{\mathcal{M}}) + k_{\mathcal{M}}$$

  where $k_{\mathcal{M}}$ is number of predictors in model $\mathcal{M}$
- The Kullback–Leibler information criterion (KIC):

$$\mathrm{KIC}(\hat{\mathcal{M}}) = L(\mathbf{y} \,|\, \hat{\mathcal{M}}) + 3/2\, k_{\mathcal{M}}$$

- Both of these tend to overfit, even for large n
  - Moderate overfitting less damaging for large n;
  - Both rarely underfit
- In practice, KIC often performs better than AIC

Model Selection Criteria (2)

- Penalize neg-log-likelihood of model by a complexity penalty
- The Akaike information criterion (AIC):

$$\mathrm{AIC}(\hat{\mathcal{M}}) = L(\mathbf{y} \,|\, \hat{\mathcal{M}}) + k_{\mathcal{M}}$$

  where $k_{\mathcal{M}}$ is number of predictors in model $\mathcal{M}$
- The Kullback–Leibler information criterion (KIC):

$$\mathrm{KIC}(\hat{\mathcal{M}}) = L(\mathbf{y} \,|\, \hat{\mathcal{M}}) + 3/2\, k_{\mathcal{M}}$$

- Both of these tend to overfit, even for large n
  - Moderate overfitting less damaging for large n;
  - Both rarely underfit
- In practice, KIC often performs better than AIC

Model Selection Criteria (3)

- The Bayesian information criterion (BIC):

$$\mathrm{BIC}(\hat{\mathcal{M}}) = L(\mathbf{y} \,|\, \hat{\mathcal{M}}) + k_{\mathcal{M}}/2 \, \log n$$

  where n is the number of samples used to fit $\hat{\mathcal{M}}$
  - As n increases, probability of overfitting goes to zero
  - More conservative; for small n can sometimes underfit
- The Risk inflation information criterion (RIC):

$$\mathrm{RIC}(\hat{\mathcal{M}}) = L(\mathbf{y} \,|\, \hat{\mathcal{M}}) + k_{\mathcal{M}} \log p$$

  where p is total number of possible predictors
  - Similar to multiple testing idea; can be quite conservative
  - Recommended over AIC/KIC if p is very large
- Many refined criteria exist: MML, MDL, corrected AIC/KIC
  $\Rightarrow$ often better for small n

Model Selection Criteria (3)

- The Bayesian information criterion (BIC):

$$\mathrm{BIC}(\hat{\mathcal{M}}) = L(\mathbf{y} \,|\, \hat{\mathcal{M}}) + k_{\mathcal{M}}/2 \, \log n$$

  where n is the number of samples used to fit $\hat{\mathcal{M}}$
  - As n increases, probability of overfitting goes to zero
  - More conservative; for small n can sometimes underfit
- The Risk inflation information criterion (RIC):

$$\mathrm{RIC}(\hat{\mathcal{M}}) = L(\mathbf{y} \,|\, \hat{\mathcal{M}}) + k_{\mathcal{M}} \log p$$

  where p is total number of possible predictors
  - Similar to multiple testing idea; can be quite conservative
  - Recommended over AIC/KIC if p is very large
- Many refined criteria exist: MML, MDL, corrected AIC/KIC
  $\Rightarrow$ often better for small n

Cross validation (1)

- Cross-validation (CV) is a very general strategy
- Motivation: ideally, we would select model that minimises prediction error on future data $y'_1, \ldots, y'_m$

$$\text{MSPE}(\hat{\mathcal{M}}(\mathbf{y})) = \frac{1}{m} \sum_{i=1}^{m} (y'_i - \hat{y}_i(\hat{\mathcal{M}}(\mathbf{y})))^2$$

$\Rightarrow$ requires knowing our testing data ahead of time ...

- We don't have future testing data, but we know the data $\mathbf{y}$ we used to fit the model is generated by the same process
- So we can try and estimate the MSPE

Cross validation (2)

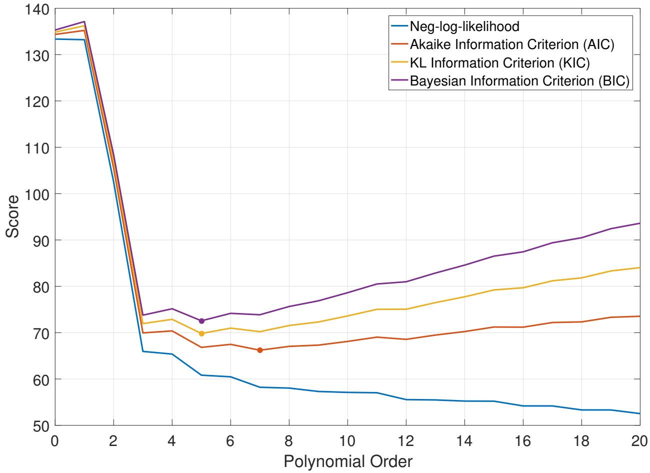
- To estimate MSPE using cross validation
  1. We randomly partition data $\mathbf{y}$ we have into two sets:
     - A training set $\mathbf{y}_{\text{train}}$,
     - and a testing set $\mathbf{y}_{\text{test}}$
  2. Fit a model $\mathcal{M}$ to the training data $\mathbf{y}_{\text{train}}$
  3. Compute the MSPE of this model on the testing data $\mathbf{y}_{\text{test}}$
  4. Repeat this procedure multiple times and average the MSPE
- This gives us an estimate of how well our model would predict future data from the same population
- Do this for all our different models, and choose the one that has the smallest CV error

Cross validation (3)

- K-fold CV
  - split into K equal sized random partitions;
  - train on $K-1$ partitions, test on the remaining partition
  - repeat this m times
- The bigger m, the more stable the estimate
- Leave-one-out CV (LOO CV):
  - train on $n-1$ of the samples, test on the remaining sample
  - do this for all n possible partitions
- Cross-validation is:
  - simple to implement, very flexible;
  - potential slow if model fitting is computationally expensive
- LOO CV is similar to AIC for large samples n

  $\Rightarrow$ can be very different for small samples

Example: fitting a polynomial



Model selection scores for fitting zero-order through twentieth-order polynomials to data generated by our example fifth-order polynomial. The likelihood always decreases with extra polynomial terms. The dots signify the models selected by each criterion.

Outline

Conventional Model Selection Methods (1)

- All-subsets selection:
  - Try all combination of predictors to model with smallest model selection criterion score
- Forward selection algorithm:
  1. Start with the empty model;
  2. Find the predictor that reduces info criterion by most
  3. If no predictor improves model, end.
  4. Add this predictor to the model
  5. Return to Step 2
- Backwards selection is related algorithm
  - Start with the full model and remove predictors

Conventional Model Selection Methods (2)

- Problems with traditional methods
- Statistically unstable
  - Small changes in data $\Rightarrow$ big changes in model
- Multiple hypothesis testing problem
  - Potential issues with false positives
- All-subsets selection
  - Infeasible for moderate to large number of predictors p
- Stepwise selection
  - Affected adversely by correlation in predictors
  - Slow for large p

Statistical Instability (1)

- Example dataset ($n = 354$, $p = 10$)
  - Target: `Y`, measure of diabetes progression
  - Predictors:
    - `AGE`, `SEX`, `BMI`
    - `BP`: blood pressure
    - `S1`,...,`S6`: blood serum measurements
- Experiment
  1. Removed one of the 354 samples at random
  2. Found "best" combination of predictors
     - Checked all possible (1024) combinations of predictors
     - Selected combination with smallest CV
- Repeated this four times

Statistical Instability (2)

Test	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6
1		✓	✓	✓	✓	✓	✓		✓	
2		✓	✓	✓	✓	✓	✓			
3	✓	✓	✓	✓	✓	✓	✓	✓		
4	✓	✓	✓	✓	✓	✓	✓		✓	

Predictors chosen by selecting model that minimises cross-validation score using all-subsets selection method. For each test, a single one of the $n = 354$ samples was removed at random.

- The “all-or-nothing” nature of this type of model selection (either exclude or include predictor) means that the “best” model can change substantially with only a small change to the data.

Penalized Regression (1)

- Alternative: apply a “penalty” to the coefficients
- For linear regression, we have penalized least-squares:

$$\left(\hat{\beta}_0, \hat{\boldsymbol{\beta}}_\lambda\right) = \underset{\beta_0, \boldsymbol{\beta}}{\arg\min} \left\{ \mathrm{RSS}(\beta_0, \boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} g(\beta_j) \right\}$$

- In comparison regular least squares only minimises goodness-of-fit (RSS)
- How to choose the penalty function $g(\cdot)$?
  - Chosen to increase with increasing magnitude of β_j
- Generalises easily to logistic regression, i.e., penalized maximum likelihood

Penalized Regression (2)

- Intuition:
  - Large coefficients imply strong effects
  - Coefficient of zero implies no association
- For this to make sense we need to standardise our predictors before fitting, i.e., ensure that

$$\sum_{i=1}^{n} x_{i,j} = 0 \text{ and } \sum_{i=1}^{n} x_{i,j}^2 = n$$

- This can be easily done by subtracting mean and dividing by standard deviation
- Each predictor is then on the same (arbitrary) scale
  $\Rightarrow$ implies that larger β_j means stronger association

Penalized Regression (3)

- The penalty includes a user-chosen hyperparameter λ
- λ controls how *strong* the penalty is
- Generally,
  - As λ goes to zero, the estimates of β are closer to least-squares
  - As λ gets larger, the estimates of β become smaller
- Choosing λ sets the "complexity" of the regression model
- By varying λ we can generate a "path" of regression models
  - The models become more complex as we move along the path

Penalized Regression (4)

- Advantages of penalized regression:
  - Increased statistical stability
  - Can be applied even when the sample size is smaller than the number of predictors (the $n < p$ setting)
  - Handles correlation between predictors
- Multi-collinearity occurs when at least two of the predictors are highly correlated
  - This means it is difficult to distinguish between their relationship with the target
  - In traditional least-squares this leads to increased variance
  - In stepwise methods, this dramatically increases instability

Ridge Regression (1)

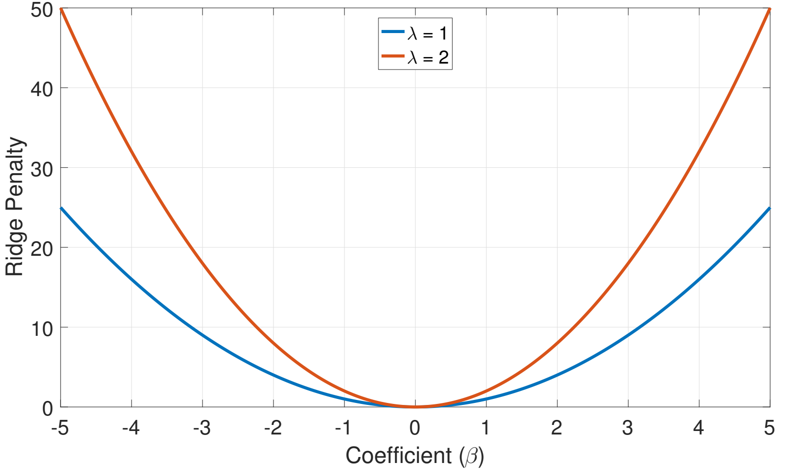
- The first penalized method introduced
- Ridge regression solves

$$\left(\hat{\beta}_0, \hat{\boldsymbol{\beta}}_\lambda\right) = \underset{\beta_0, \boldsymbol{\beta}}{\arg\min} \left\{ \mathrm{RSS}(\beta_0, \boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

- Strengths:
  - Very quick to run, even for large n and p
  - Produces very stable, low variance estimates
  - Very appropriate for multi-collinearity
- Weaknesses:
  - Cannot estimate coefficients to be exactly zero
    - i.e., no variable selection

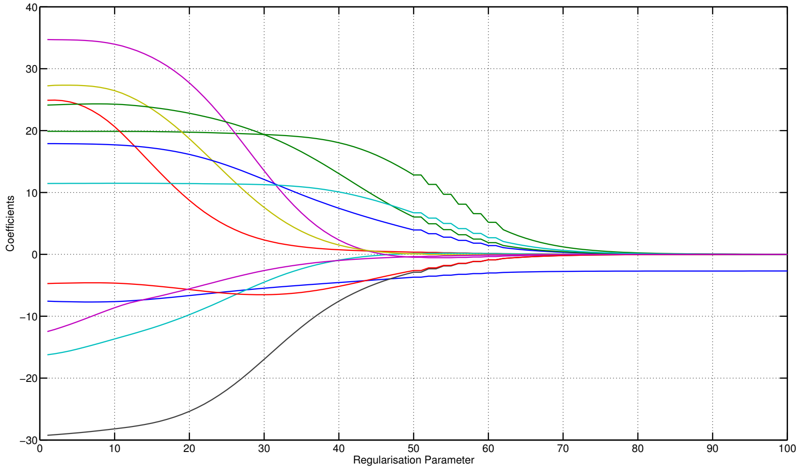
Ridge Regression (2)

- Penalty is the square of the coefficient (ℓ_2 penalty)



Ridge Regression (3)

- Example ridge regression coefficient path



Lasso Regression (1)

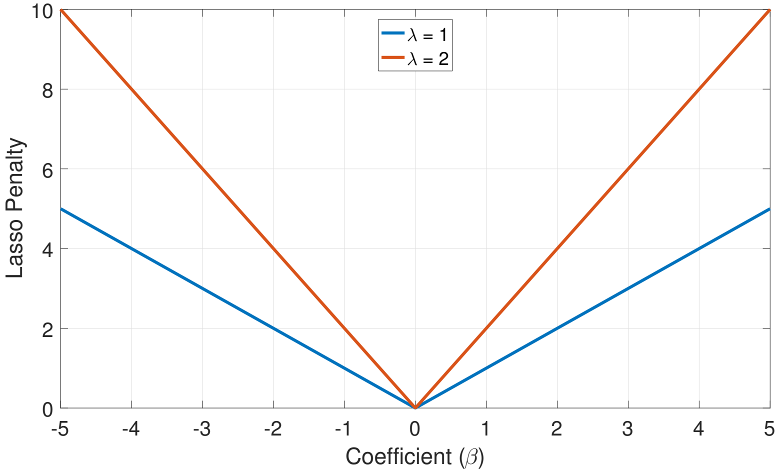
- A "sparse" estimator
- Lasso regression solves

$$\left(\hat{\beta}_0, \hat{\boldsymbol{\beta}}_\lambda\right) = \underset{\beta_0, \boldsymbol{\beta}}{\arg\min} \left\{ \mathrm{RSS}(\beta_0, \boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

- Strengths:
  - Efficient algorithms exist
  - Can estimate coefficients to be exactly zero
    - i.e., can perform variable selection
- Weaknesses:
  - Produces biased estimates for large coefficients
  - Correlated predictors can be problematic
  - Can overfit (include unassociated predictors)

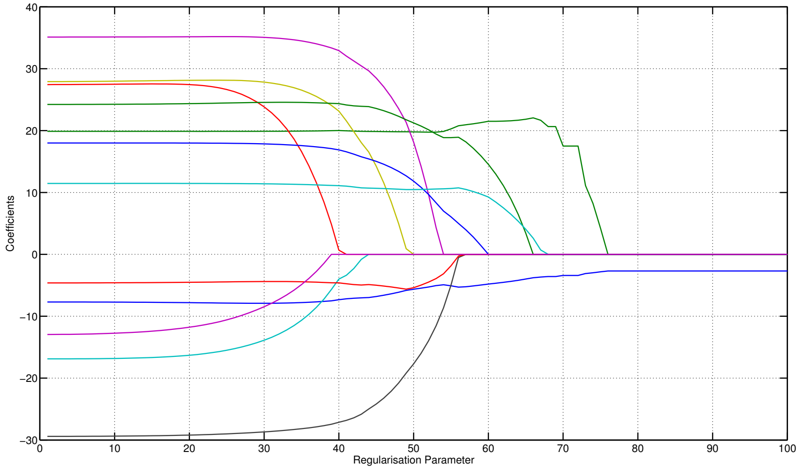
Lasso Regression (2)

- Penalty is the absolute value of the coefficient (ℓ_1 penalty)



Lasso Regression (3)

- Example lasso regression coefficient path



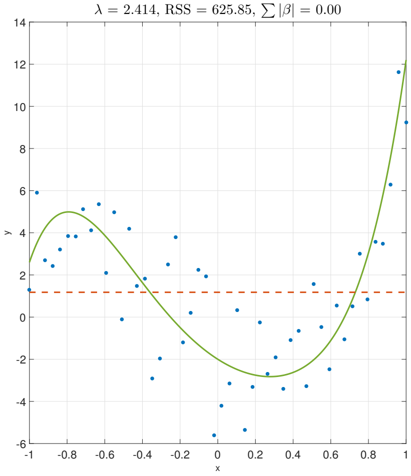
Selecting a Model Using Penalized Regression

- How to choose the hyperparameter λ
- Standard procedure:
  1. Vary λ over some grid of values
  2. For each λ, use cross-validation to estimate prediction error
  3. Select λ with smallest cross-validation error
  4. Use that λ to estimate our final model from all the data
- Can also use methods like AIC, BIC, etc.
- Implementations:
  - MATLAB `lasso()` and `lassoglm()` functions
  - R, use the `glmnet` package

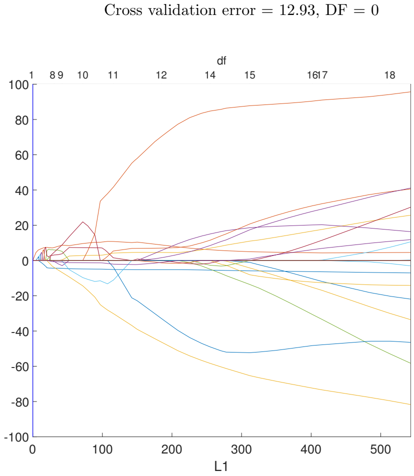
Example – Lasso regression

- Example: we observe x and y data and want to build a prediction model for y using x
  - Data looks nonlinear so we use polynomial regression
  - We take $x, x^2, x^3, \dots, x^{20} \Rightarrow$ very flexible model
  - Which terms to include?
- For example, our model could be:
  - (x, x^2, x^3); or
  - (x, x^4, x^{16}, x^{17}); or
  - $(x^2, x^3, x^5, x^6, x^9)$; and so on ...
- Let us use Lasso regression

Example – Lasso regression

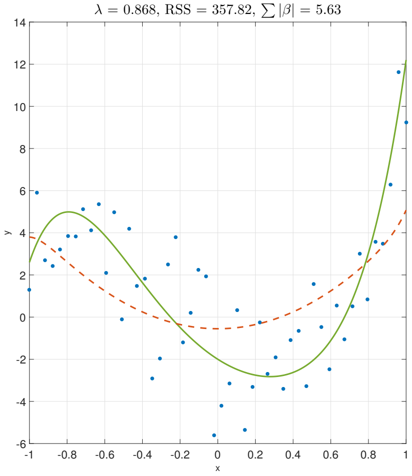


(a) Fitted model

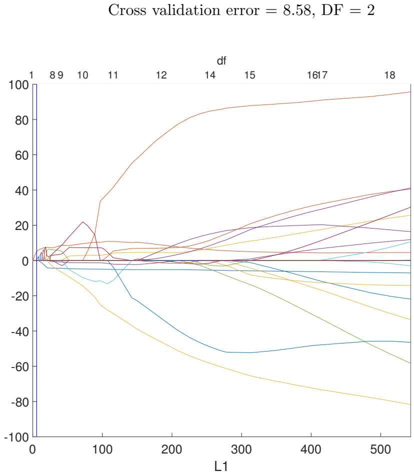


(b) Lasso path

Example – Lasso regression

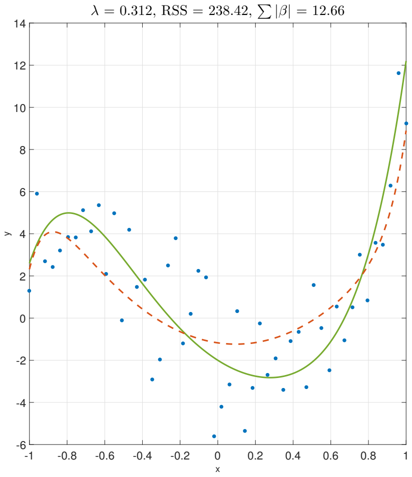


(a) Fitted model

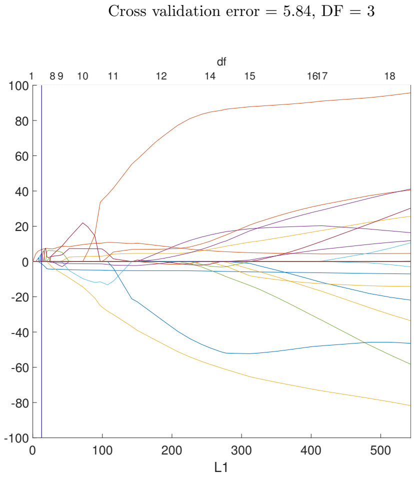


(b) Lasso path

Example – Lasso regression

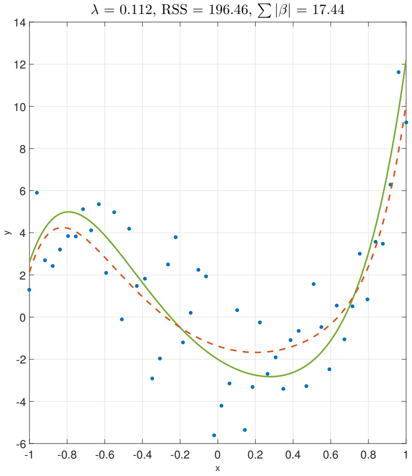


(a) Fitted model

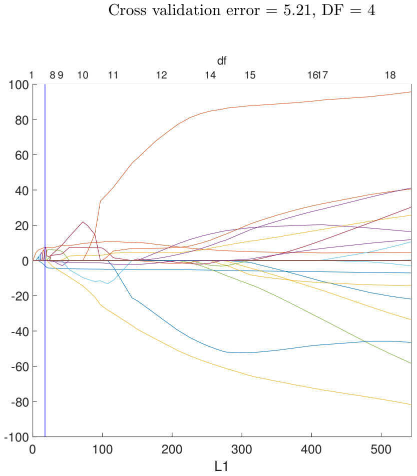


(b) Lasso path

Example – Lasso regression

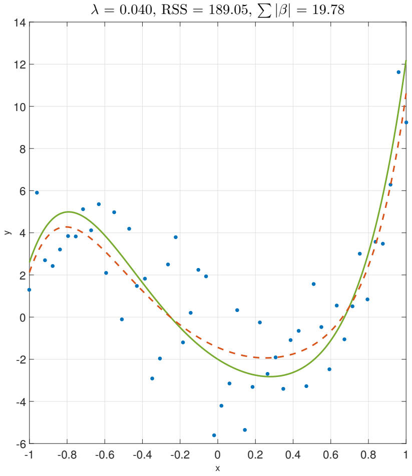


(a) Fitted model

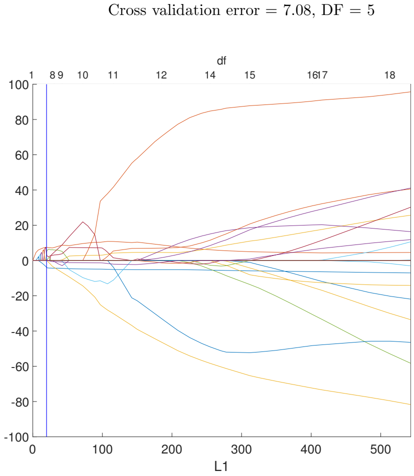


(b) Lasso path

Example – Lasso regression

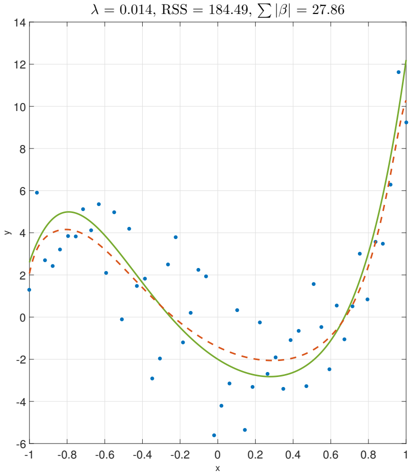


(a) Fitted model

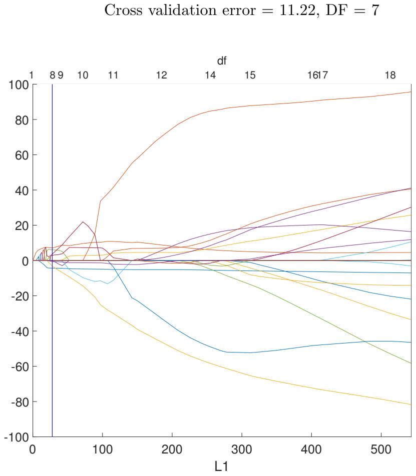


(b) Lasso path

Example – Lasso regression

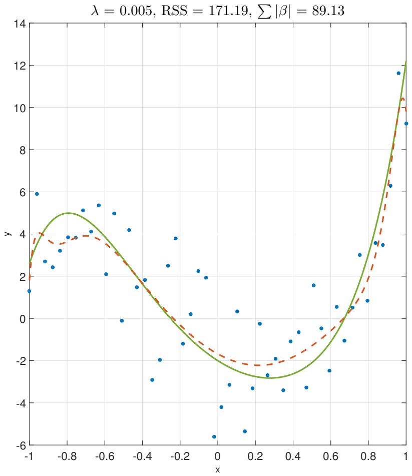


(a) Fitted model

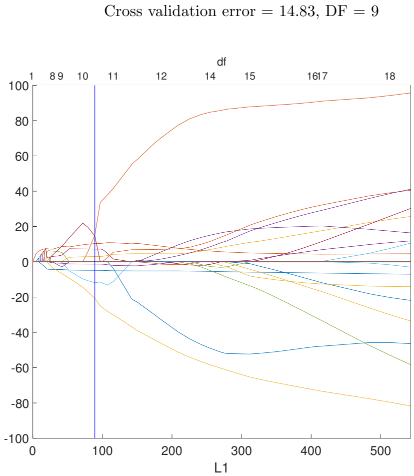


(b) Lasso path

Example – Lasso regression

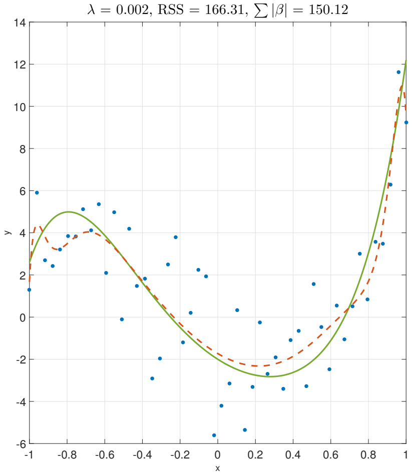


(a) Fitted model

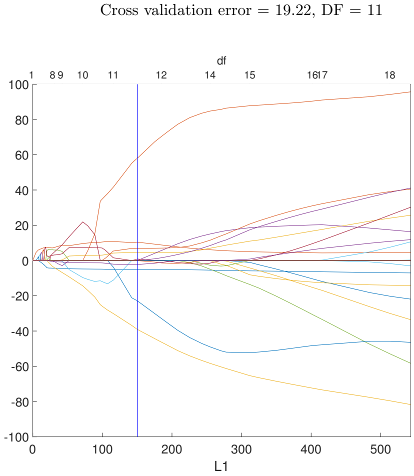


(b) Lasso path

Example – Lasso regression

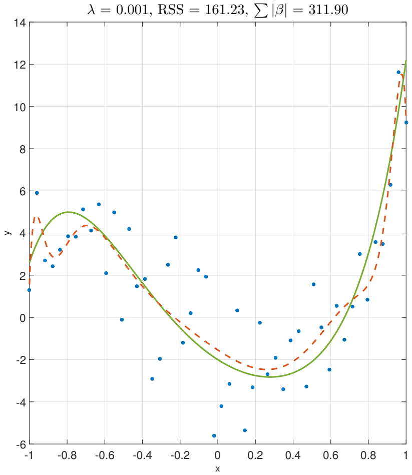


(a) Fitted model

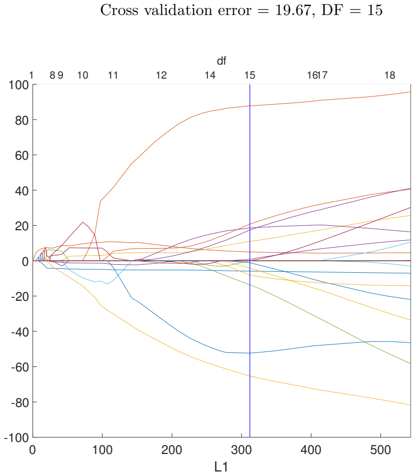


(b) Lasso path

Example – Lasso regression

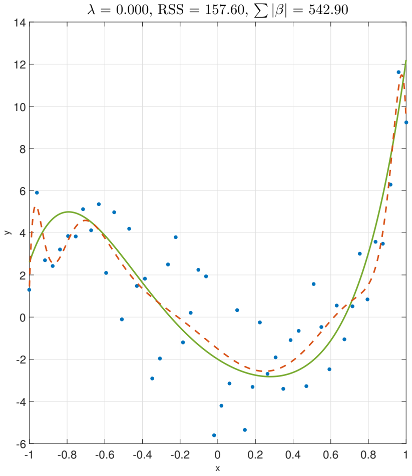


(a) Fitted model

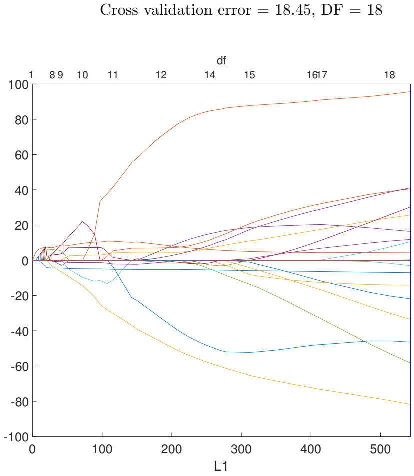


(b) Lasso path

Example – Lasso regression



(a) Fitted model



(b) Lasso path

Bias/Variance Trade-off

- Penalized regression has a bias/variance motivation
- Recall that the EMSPE of a model can be written as

$$\text{EMSPE} = \text{bias}^2 + \text{variance}$$

- If all relevant predictors included, least-squares is unbiased
  $\Rightarrow$ however, LS variance can be high
- Penalized regression introduce some bias but reduces variance
  - They shrink the coefficients towards zero
- Choosing a good λ means we can reduce the overall EMSPE

Reading/Terms to Revise

- Terms you should know:
  - Underfitting and bias
  - Overfitting and variance
  - Expected mean-squared prediction error
  - Multiple hypothesis testing, Bonferroni procedure
  - AIC, BIC
  - Cross-validation
  - Statistical instability
  - Penalized regression
  - Ridge/lasso regression
- Next week: non-linear machine learning algorithms for classification and regression (k nearest neighbours, decision trees, random forests).