ECON 557 – Advanced Data Analysis

Michael T. Sandfort

Department of Economics
Masters in Applied Economics Program
Georgetown University

January 27, 2023

Estimator Properties

Consistency

Consistency is the lodestar for us. More data should get us closer to the truth. With modest regularity assumptions, the QML estimator is consistent for the pseudo-true value.

Asymptotic Normality

Asymptotic normality helps us to understand estimator precision. With modest regularity assumptions, the QML estimator is asymptotically normal (more next lecture).

True Models

If everyone agrees that the model is the true model, then the QML estimator is the same as the ML estimator, the pseudo-true value is the true value of θ, and things will probably go well for us.

What Do We Talk About When We Talk About R^2?

- [1] How much of the observed variation does the (true) model capture? This is a feature of the population, and can be useful to know because it impacts how well we can expect to predict when using the best predictor. When y is inherently highly variable, many actual outcomes may fall far from the best predictor.

- [2] "My high-R^2 model is better than your low-R^2 model." In this case, the parties evidently don't agree about the true model, so the discussion of R^2 is standing in for a discussion of model assessment and selection. Can we do better?

Assessing Model Performance and Selecting Among Models

- Finding a model with good fit to a given data set $\mathcal{D}$ is actually fairly easy to do.
- For example, in a regression setting, inserting a RHS variable for every observation will do it. This is just regressing using $\mathbf{X} = \mathbf{I}_n$, and we always have an identity matrix lying around.
- So we must have something in mind other than just memorizing the data when we talk about a model that "fits well."

  The existence of a problem in knowledge depends on the future being different from the past, while the possibility of a solution of the problem depends on the future being like the past. – Frank Knight

Generalization

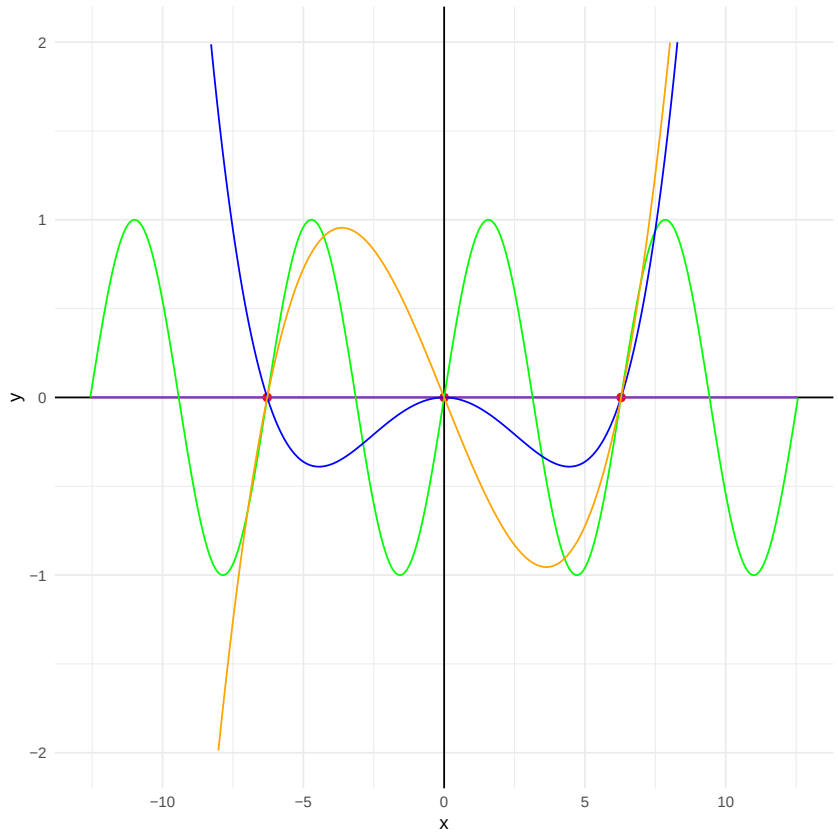
- Generalization captures the tradeoff between:
  - Providing predictions that rely on the training data in a meaningful way, adding sharpness through the use of covariates.
  - Providing predictions that don't rely on the data so much that they just regurgitate the memorized covariates present in the training data. Such predictions could be expected to fail to agree well with new (test) observations.
- This is a very hard problem and affects even attempts to learn an unknown function $f(\mathbf{x})$ when outputs are observed without error (e.g., $y = f(\mathbf{x})$ rather than $y = f(\mathbf{x}) + e$ with e random).
- To better understand the problem, the proposed solution, and why it works, we're going to "turn off" the randomness on y conditional on $\mathbf{x}$ for the remainder of this lecture.

No Free Lunch

- Can't expect to learn an infinite amount from a finite collection of observations (even if large), so we should start with low expectations.
- Can we ever expect to learn anything at all?
- Initial attempts are not promising...

No Free Lunch (Continuum Outcomes Edition)

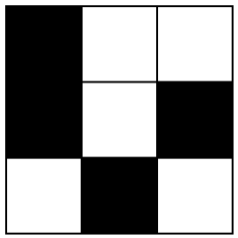
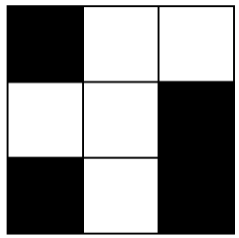
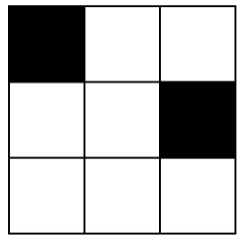
The function below was sampled at the points $\mathcal{D} = \{-2\pi, 0, 2\pi\}$. The function values at those points are shown in red.



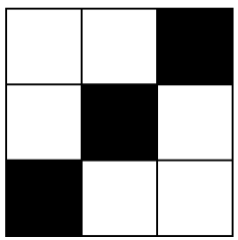
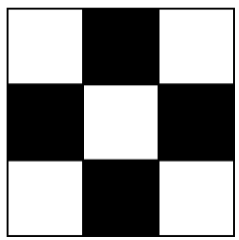
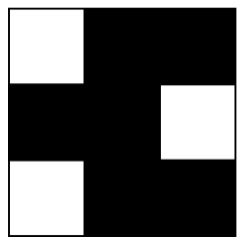
In case you think $y \in \mathbb{R}$ is the source of our problem, consider the following...

No Free Lunch (Discrete Outcomes Edition)

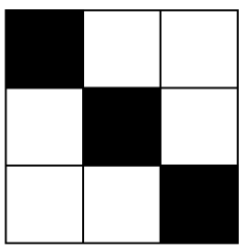
Suppose you are told that the following patterns all have $y = 0$

while these patterns all have $y = 1$

Predict y for



So Is This Going To Be A Really Short Class? (no)

- Learning from $\mathcal{D}$ is doomed if any unknown f can happen.
- But if we are willing to dial back our ambition, we already have a framework for learning (inference) from data.
- Consider a large population "bin" with many orange and green marbles.
- We don't know the true proportion (μ) of marbles which are orange.
- But suppose you get a sample of the marbles:

$$\mathcal{D} = \{ \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \}$$

  - Can you infer the true proportion of orange marbles in the population?
  - Can you bound the distance between your inference and the truth?
  - Will your inference always fall in that bound?

Probability Bounds

- You may have already seen a few of the basic bounds from probability theory.
- For non-negative random variables with finite mean, the Markov inequality:

$$\mathbb{P}[y \geq t] \leq \frac{\mathbb{E}[y]}{t} \qquad t > 0$$

- For those also with finite variance, the Chebychev inequality:

$$\mathbb{P}[|y - \mu| \geq t] \leq \frac{\mathbb{V}[y]}{t^2} \qquad t > 0$$

- This latter is a first concentration inequality, guaranteeing that large deviations from the mean (approximation statement) are very unlikely (probability statement).
- Concentration inequalities are available for most moments, and (importantly) for something called the moment generating function, where it is called the Chernoff bound.

Probability Bounds

- Draws from our population bin have only two possible outcomes: orange and green.
- Let μ be the true proportion of orange balls.
- Let ν be the fraction of orange balls in our sample $\mathcal{D}$.
- In this environment, a concentration inequality known as the Hoeffding bound applies:

$$\mathbb{P}[|\nu - \mu| > \varepsilon] \leq 2e^{(-2N\varepsilon^2)}$$

- The statement "$\nu = \mu$" is within ε of being correct $1 - 2\exp^{-2N\varepsilon^2}$ of the time.
- The statement is probably approximately correct, or PAC.

Probability Bounds

$$\mathbb{P}[|\nu - \mu| > \varepsilon] \leq 2e^{(-2N\varepsilon^2)}$$

- The statement is valid for all N and ε.
- If we can live with a worse approximation (larger ε) we get a higher probability that "$\nu = \mu$".
- If we can get a larger sample (larger N), then we get a higher probability that "$\nu = \mu$".
- The bound does not depend on the true parameter μ.

How Does This Relate to Generalization?

- For our bin
  - Bin full of marbles ($\bullet$).
  - Some marbles are $\bullet$ ($y = 1$).
  - Some marbles are $\bullet$ ($y = 0$).
  - Unknown proportion $\mathbb{E}[y] = \mu$ of $\bullet$.
  - Sample $\mathcal{D}$ of size N from bin, taken i.i.d.
- For our learning problem
  - Feature space full of covariate combinations $\mathbf{x} \in \mathcal{X}$.
  - Some $\mathbf{x}$ have $h(\mathbf{x}) \neq f(\mathbf{x})$. ($h$ is wrong.)
  - Some $\mathbf{x}$ have $h(\mathbf{x}) = f(\mathbf{x})$. ($h$ is right.)
  - For hypothesis $h(\cdot)$, unknown proportion of $\mathbf{x}$ have $h(\mathbf{x}) \neq f(\mathbf{x})$.
  - Sample $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ is available, i.i.d, with $y_i = f(\mathbf{x}_i)$.
- If N is large, we can probably approximately infer the unobserved proportion of $h(\mathbf{x}) \neq f(\mathbf{x})$ by computing the observed proportion of $h(\mathbf{x}_i) \neq y_i$.

Have We "Learned"?

- In the language of Lecture 2, and for a fixed $h \in \mathcal{H}$, we have

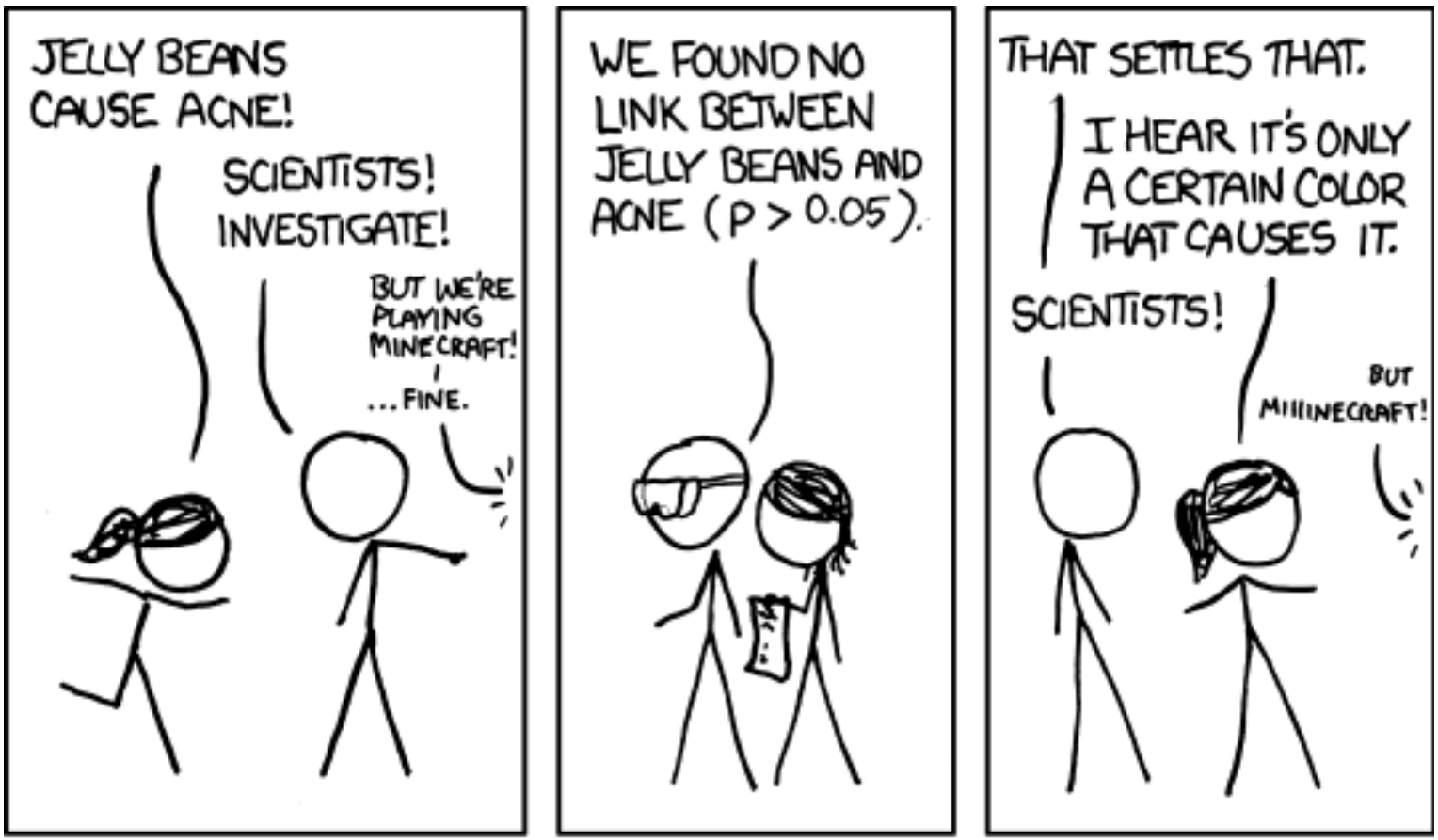
$$\mathbb{P}[|\widehat{R}_N(h) - R(h)| \geq \varepsilon] \leq 2e^{-2N\varepsilon^2}$$

- That is, the empirical risk for this h is probably close to the actual risk.
- We did this for a fixed h. Does that actually reflect the problem posed in Lecture 2? (Looks more like verification.)
- If it turns out that empirical risk $\widehat{R}_N(h)$ is small for this h, then $h = f$ is PAC.
- We don't currently even have a guarantee that h is small among the candidates in $\mathcal{H}$!
- We can't claim to have learned the oracle in any meaningful sense unless our algorithm is choosing among the $h \in \mathcal{H}$.

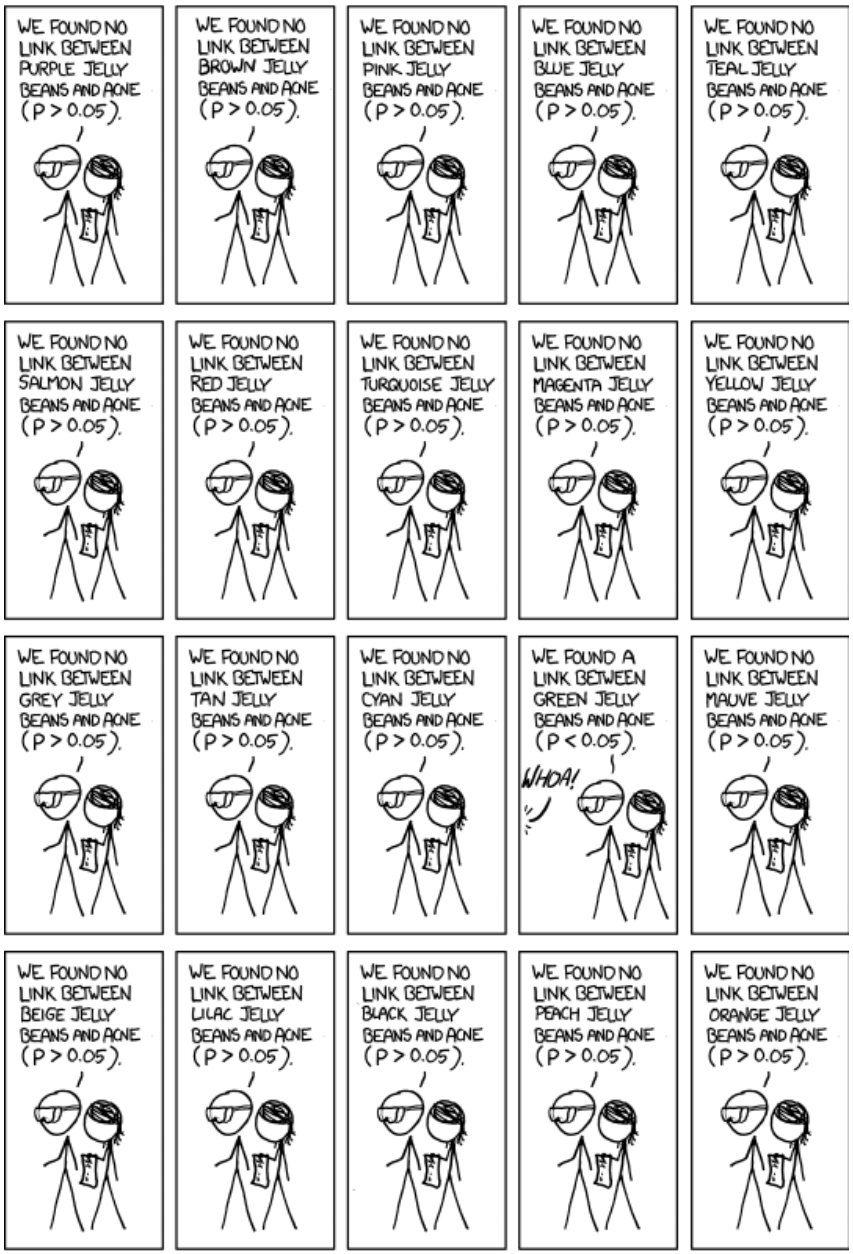
A Peril of Multiple h: False Discovery

- Suppose we now compute the empirical risk for every $h \in \mathcal{H}$.
- Every $h \in \mathcal{H}$ gets the "Hoeffding guarantee," so when we find an h^* with the smallest empirical risk, do we shout "hooray!"?
- Suppose I hand out a box of 100 fair coins and we flip each coin $(h_1, \ldots, h_{100})$ 5 times.
- One coin gets 5 heads in a row. Is it magical?

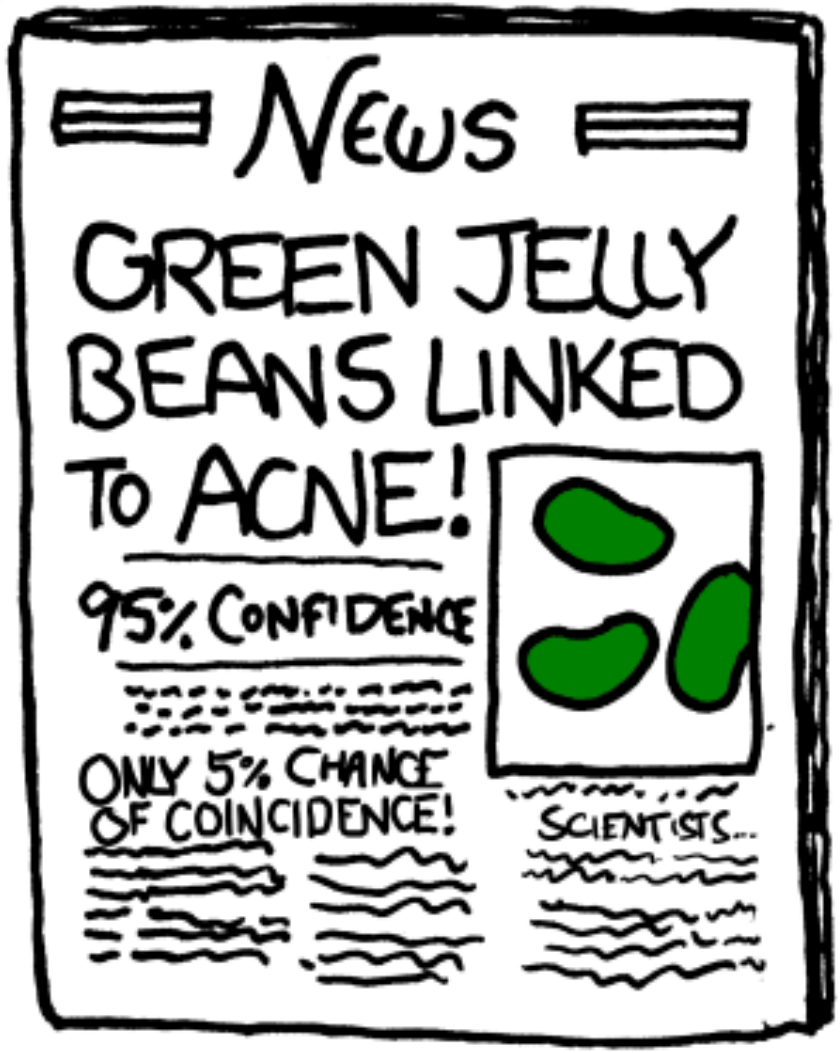
Wait, I Think I've Seen This Question Before...



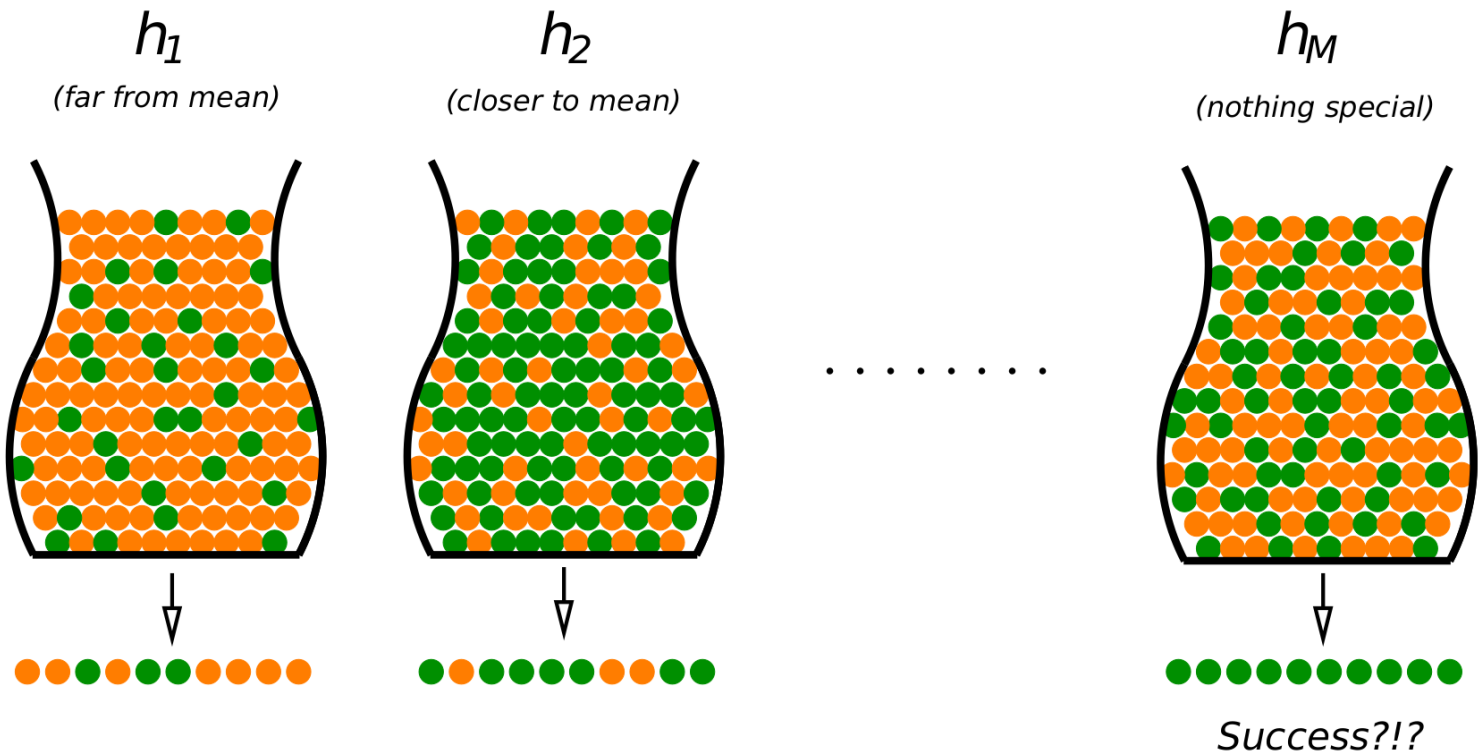
Wait, I Think I've Seen This Question Before...



Wait, I Think I've Seen This Question Before...



Multiple h: False Discovery



Multiple h: False Discovery

- Work it out!

$$\begin{aligned}\mathbb{P}[\text{At least one coin gets 5 heads}] &= 1 - \mathbb{P}[\text{No coin gets 5 heads}] \\ &= 1 - \left[\frac{2^{\text{tries}} - 1}{2^{\text{tries}}}\right]^{\text{coins}} \\ &= 1 - \left[\frac{31}{32}\right]^{100} > 0.95(!)\end{aligned}$$

- We're selecting for "lucky" (actually BAD) samples rather than for good agreement between empirical and true risk!
- In the case of the coins, the true risk is always $\frac{1}{2}$, but we chose the coin with empirical risk 0!

Multiple h: The Union Bound

- We want a Hoeffding-like bound that accounts for the fact that we are testing hypotheses $h_1, h_2, \ldots, h_M$ simultaneously.
- That is, we want

$$\mathbb{P}\left[\exists h \in \mathcal{H} : |\widehat{R}_N(h) - R(h)| \geq \varepsilon\right] = \mathbb{P}\left[\max_{h \in \mathcal{H}} |\widehat{R}_N(h) - R(h)| \geq \varepsilon\right]$$

- But note that the event

$$\max_{h \in \mathcal{H}} |\widehat{R}_N(h) - R(h)| \geq \varepsilon$$

is equivalent to the union of the individual hypothesis "verification" events

$$\bigcup_{h \in \mathcal{H}} \left\{|\widehat{R}_N(h) - R(h)| \geq \varepsilon\right\}.$$

Multiple h: The Union Bound

- Therefore, we can use Bonferroni's bound (also known as the "union bound") to get

$$\begin{aligned}
\mathbb{P}\left[\max_{h\in\mathcal{H}} |\widehat{R}_N(h) - R(h)| \geq \varepsilon\right] &= \mathbb{P}\left(\bigcup_{h\in\mathcal{H}} \left\{|\widehat{R}_N(h) - R(h)| \geq \varepsilon\right\}\right) \\
&\leq \sum_{h\in\mathcal{H}} \mathbb{P}[|\widehat{R}_N(h) - R(h)| \geq \varepsilon] \\
&\leq \sum_{h\in\mathcal{H}} 2e^{-2N\varepsilon^2} \\
&= 2|\mathcal{H}|e^{-2N\varepsilon^2}
\end{aligned}$$

- Now we can take the empirical risk minimizer over $h_1, \ldots, h_M$, but with less surety (a looser bound) on success.
- This will work for all (finite) $|\mathcal{H}|$, N and ε.
- It will work for our discrete example at the start of lecture, but won't for the continuum ($|M| = \infty$).

Our First Generalization Result

- There is a one-sided version of the above inequality which has

$$\mathbb{P}\left[\max_{h\in\mathcal{H}} R(h) - \widehat{R}_N(h) \geq \varepsilon\right] \leq |\mathcal{H}|e^{-2N\varepsilon^2}$$

- Setting $\delta = |\mathcal{H}|e^{-2N\varepsilon^2}$ and solving for ε shows that for all $h \in \mathcal{H}$ and all $\delta > 0$,

$$R(h) \leq \widehat{R}_N(h) + \sqrt{\frac{\log|\mathcal{H}| + \log(1/\delta)}{2N}}$$

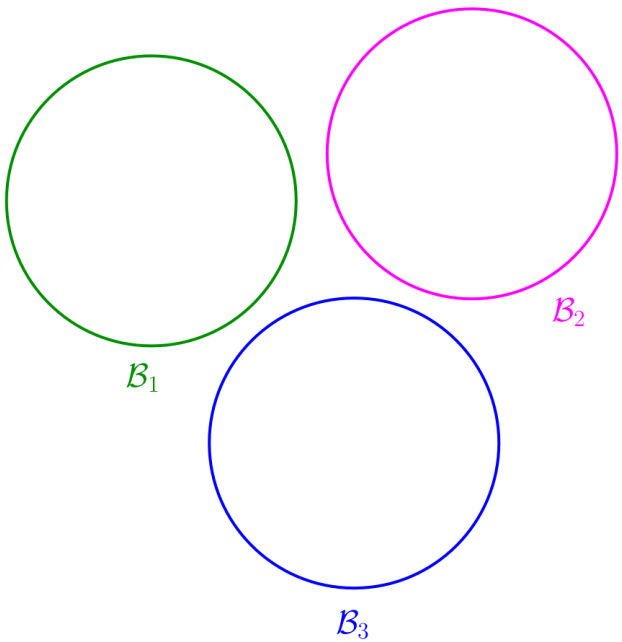
  with probability at least $1 - \delta$.

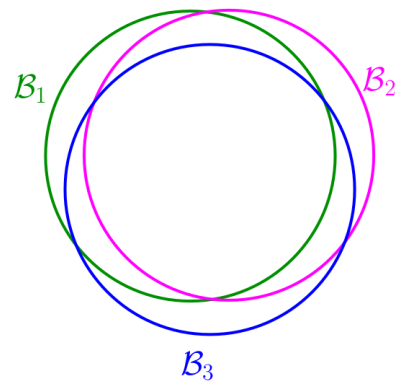
- This expression makes concrete the sense in which minimizing only the empirical risk $\widehat{R}_N(h)$ leads to estimates of true risk that are wildly optimistic – they fail to account for "BAD" samples and for false discovery.

- But if $|\mathcal{H}| = \infty$, this bound is no bound at all.

Can We Do Better?

- In many cases, we can do better than the bound on the previous slide.
- The union bound does not account for overlap in the "BAD" events (e.g., $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$) – that a "BAD" event under one hypothesis may be "BAD" under other hypotheses as well.





Looking at $\mathcal{H}$ Through the Lens of $\mathcal{D}$

- Suppose our hypothesis space $\mathcal{H}$ =half-planes in $\mathbb{R}^2$.
- A one-point sample $\mathcal{D} = \mathbf{x}_1$ is classified $y = 1$ if it falls in the half-plane, and $y = 0$ otherwise.
- How many half planes are there? $|\mathcal{H}| = \infty$.
- But how many kinds of half-planes are there from the standpoint of their classification of $\mathbf{x}_1 \in \mathcal{D}$? Only 2!
- "Type-h_1" half-planes say $h(\mathbf{x}_1) = 0$. "Type-h_2" half-planes say $h(\mathbf{x}_1) = 1$.
- Moreover, all h in one of these equivalence classes will be equally BAD with respect to the data $\mathcal{D}$.

Looking at $\mathcal{H}$ Through the Lens of $\mathcal{D}$

- Suppose our hypothesis space $\mathcal{H}$ =half-planes in $\mathbb{R}^2$.
- But now we have a two-point sample $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2\}$.
- Now how many equivalence classes of half-planes are there?
- You should be able to convince yourself that there are just four.
- What about $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$?
- For most $\mathcal{D}$, there are eight equivalence classes.
- But when the elements of $\mathcal{D}$ are collinear, there may be fewer (two of the eight possible labellings cannot be achieved).

Looking at $\mathcal{H}$ Through the Lens of $\mathcal{D}$

- Suppose our hypothesis space $\mathcal{H}$ =half-planes in $\mathbb{R}^2$.
- What about a four-point sample $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$?
- This case is different. Regardless of the arrangement of points, there always exists a labelling which cannot be achieved with half-planes.
- The equivalence classes of $\mathcal{H}$ capture at most 14 of the total $2^4 = 16$ labellings.

The Growth Function

- For any hypothesis space $\mathcal{H}$, we let $\mathcal{H}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ represent the set of equivalence classes on $\mathcal{H}$ induced by the data $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$.
- We then take out the dependence on any particular data set $\mathcal{D}$ by maximizing over all possible $\mathcal{D}$.
- This procedure defines the growth function:

$$m_{\mathcal{H}}(N) = \max_{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \in \mathcal{X}} |\mathcal{H}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)|$$

  the largest number of equivalence classes induced on $\mathcal{H}$ by a data set $\mathcal{D}$ of size N.

- It turns out that for half-lines in $\mathbb{R}^1$, $m_{\mathcal{H}}(N) = N + 1$.
- For intervals in $\mathbb{R}^1$, $m_{\mathcal{H}}(N) = \frac{1}{2}N^2 + \frac{1}{2}N + 1$.
- For half-planes in $\mathbb{R}^2$, $m_{\mathcal{H}}(N) < 2^N$ if $N < 4$.

The Vapnik-Chervonenkis (VC) Dimension

- Using the growth function, we can define the VC dimension of a hypothesis space $\mathcal{H}$.
- The VC dimension of $\mathcal{H}$ is the largest N for which $m_{\mathcal{H}}(N) = 2^N$.
- For example, we saw for half-planes that the growth function $m_{\mathcal{H}}(N)$ was 2^N for $N = 1, 2, 3$, but that at $N = 4$ there was a "break" in the growth function: $m_{\mathcal{H}}(4) = 14 < 16 = 2^4$.
- Therefore, the VC dimension of the hypothesis space of half-planes is $d_{\text{VC}} = 3$.

The VC Bound

- Using the VC dimension, one can prove a probability bound that works even for some infinite hypothesis spaces.
- It has the same form that we saw earlier:

$$\mathbb{P}\left[|R(h) - \widehat{R}_N(h)| \geq \varepsilon\right] \leq 4(2N)^{d_{\text{VC}}} e^{-\frac{1}{8}N\varepsilon^2}$$

- As before, there is a one-sided version of the bound that allows us to say

$$R(h) \leq \widehat{R}_N(h) + \sqrt{\frac{8}{N} + \log\left(\frac{4(2N)^{d_{\text{VC}}}}{\delta}\right)}$$

  or just

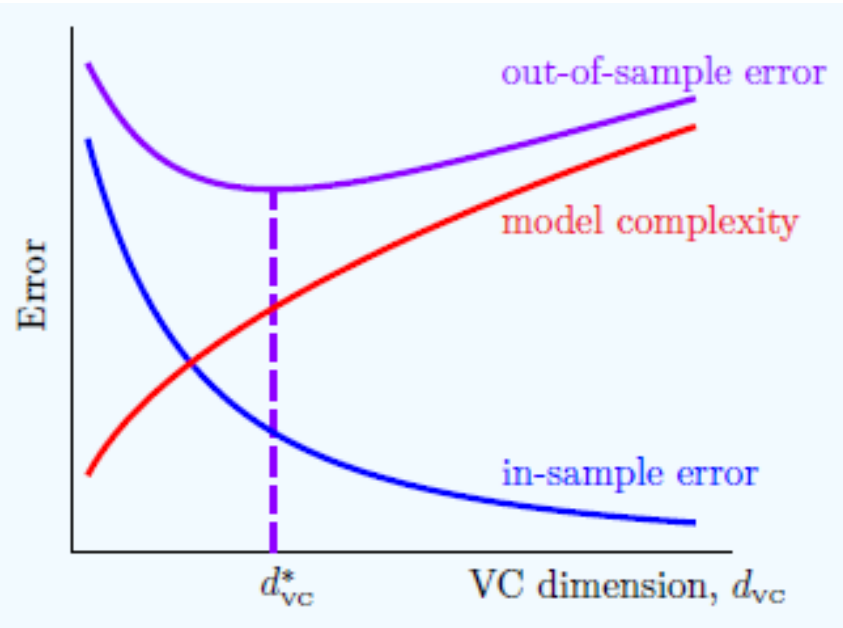
$$R(h) \leq \widehat{R}_N(h) + \Omega(N, \mathcal{H}, \delta)$$

- This suggests minimizing the empirical risk, but with a penalty $\Omega(N, \mathcal{H}, \delta)$ for model complexity.

The Key Takeaway

$$R(h) \leq \widehat{R}_N(h) + \sqrt{\frac{8}{N} + \log\left(\frac{4(2N)^{d_{\text{VC}}}}{\delta}\right)}$$

- We saw earlier that we can always reduce $\widehat{R}_N(h)$ by working from a very complex $\mathcal{H}$. The empirical risk declines monotonically in the complexity of the model d_{VC}.
- But the term $\Omega(N, \mathcal{H}, \delta)$ increases monotonically in the complexity of the model.
- At last, this is our pushback against overfitting – a very flexible $\mathcal{H}$ is not always good!



Missing Details

- This covers the basics of generalization theory.
- Missing from the discussion are the effect of making the labels imperfect (i.e., y is random, conditional on $\mathbf{x}$) and how to deal with hypothesis spaces of functions.
- The mathematics gets pretty advanced, but the structure of the bounding equations is very similar (so hopefully this intuition helps).
- In particular, the intuition helps explain why, as we move forward with new estimators, the estimator will often be composed of two additive terms: an empirical loss and a "complexity penalty."
- The "adjusted R^2", or $\overline{R}^2$, is one ad-hoc way to penalize complexity.
- There are other, information-based criteria, that we will also talk about briefly.
- The remainder of today we will devote to a nonparametric means of assessing the "true risk" of a model of given complexity, cross-validation.

Model Assessment and Model Selection

- Model assessment: Evaluating a model's performance
- Model selection: Choosing the model with the best performance
- Both model assessment and model selection require an estimate of the model's out-of-sample error.

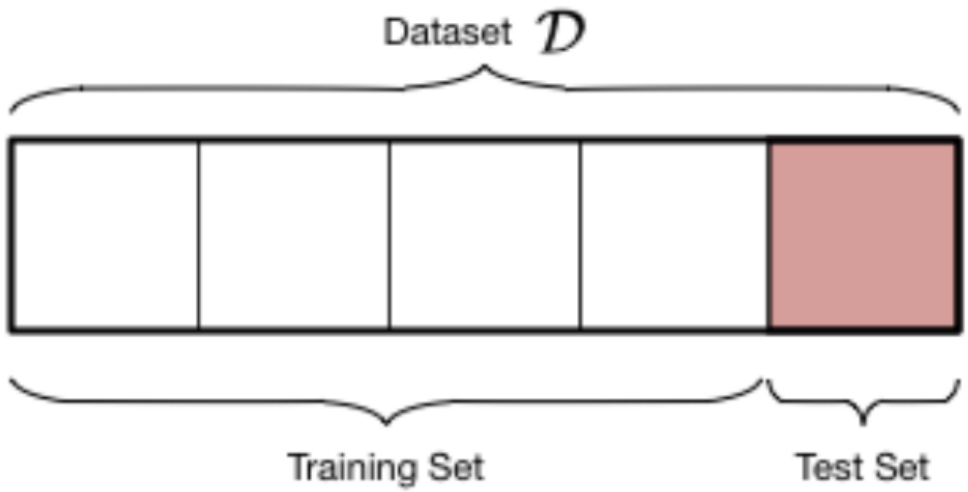
Model Assessment

To evaluate the performance of a model $h \in \mathcal{H}$, we can randomly split our data into two parts:

- Training set: This set is used to fit/estimate the model (selecting h^* from $\mathcal{H}$)
- Test set: This set is used to evaluate the model fit (how good is h^*)

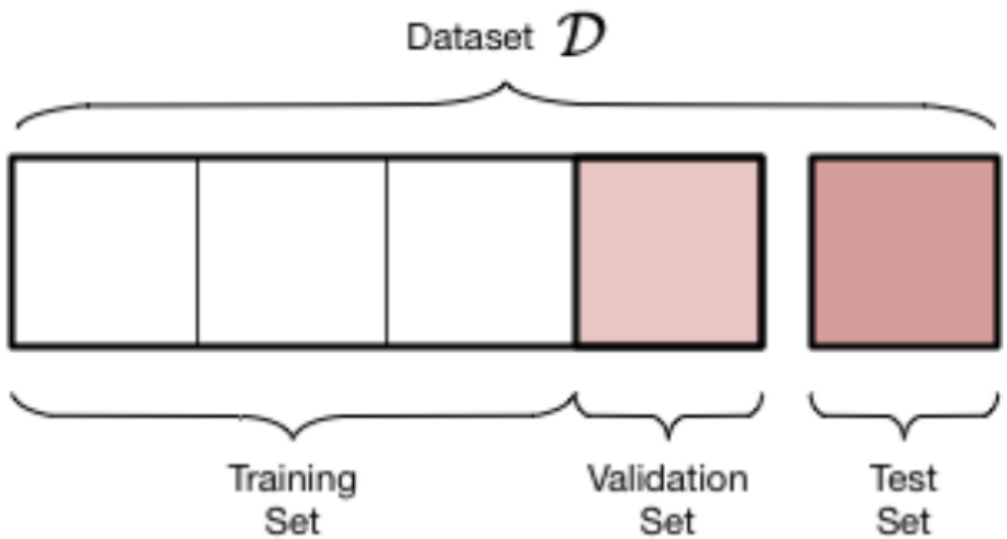
Training versus Testing

- The generalization bound for test error $\widehat{R}_N(h^*)$ is much tighter than the bound for general h because we are testing a (single) specific hypothesis.
- The test set is not biased, whereas the training set has an optimistic bias, since it is used to choose a hypothesis that looks good on the data.
- The price of having a test set is that those data are not available for use in training.



Model Selection

- To choose the best model $\mathcal{H}_*$ among a set of models $\mathbb{H}$, we can randomly split our data into three parts:
  - A training set
  - A test set
  - A validation (hold-out) set



Model Selection

Then follow the steps below:

1. Fit each model $\mathcal{H}_i \in \mathbb{H}$ on the training set and obtain h_i for $i \in \{0, 1, \ldots, n\}$.

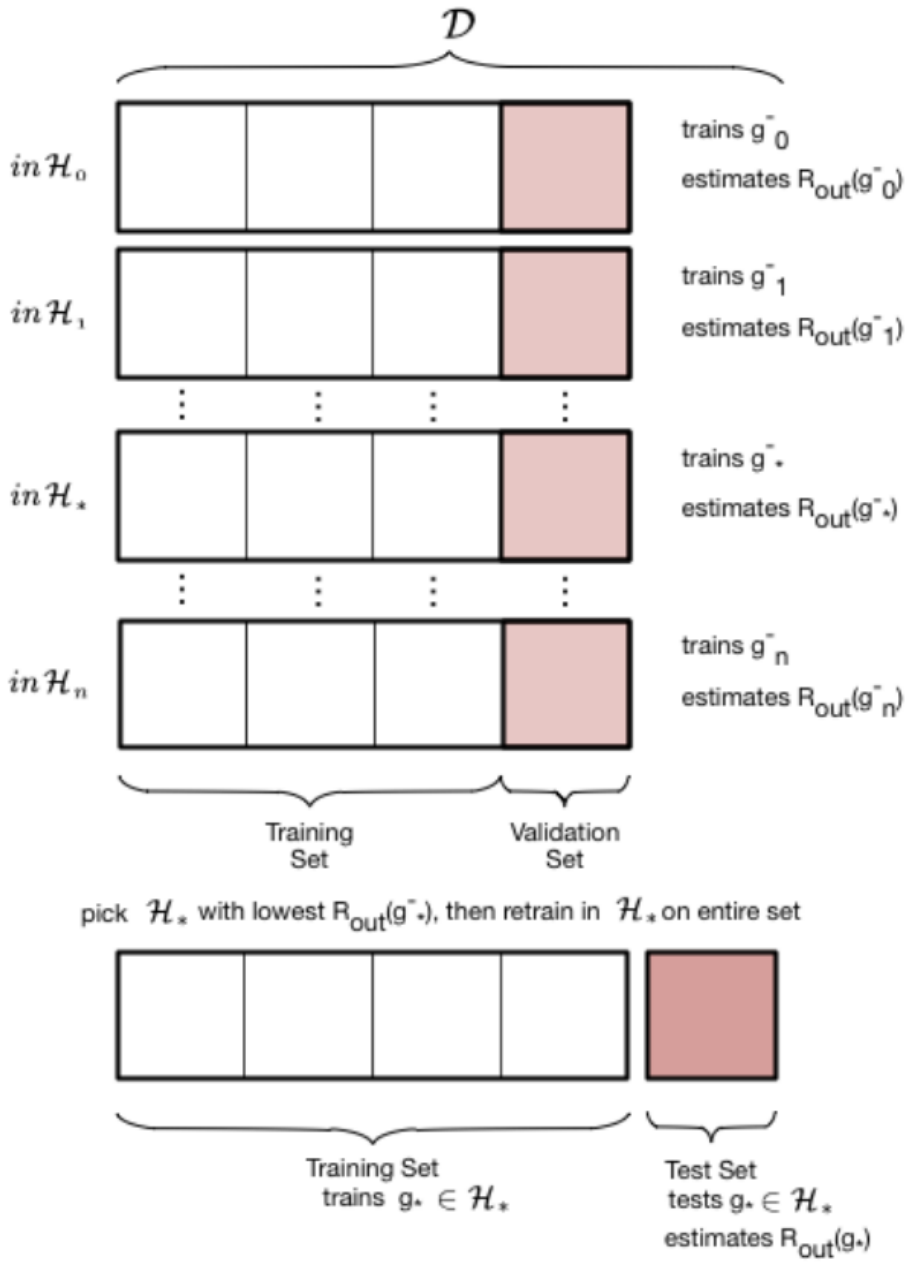
2. Use the validation set to evaluate the performance of each h_i and select $h^* \in \mathcal{H}_*$ with the smallest error on the validation set. $\mathcal{H}_*$ is our selected model.

3. Combine the training set and the validation set. Refit model $\mathcal{H}_*$ on the training-plus-validation sets to obtain h^{**}. This is the selected hypothesis.

4. Assess the performance of h^{**} on the test set.

This is called the validation set approach.

Model Selection



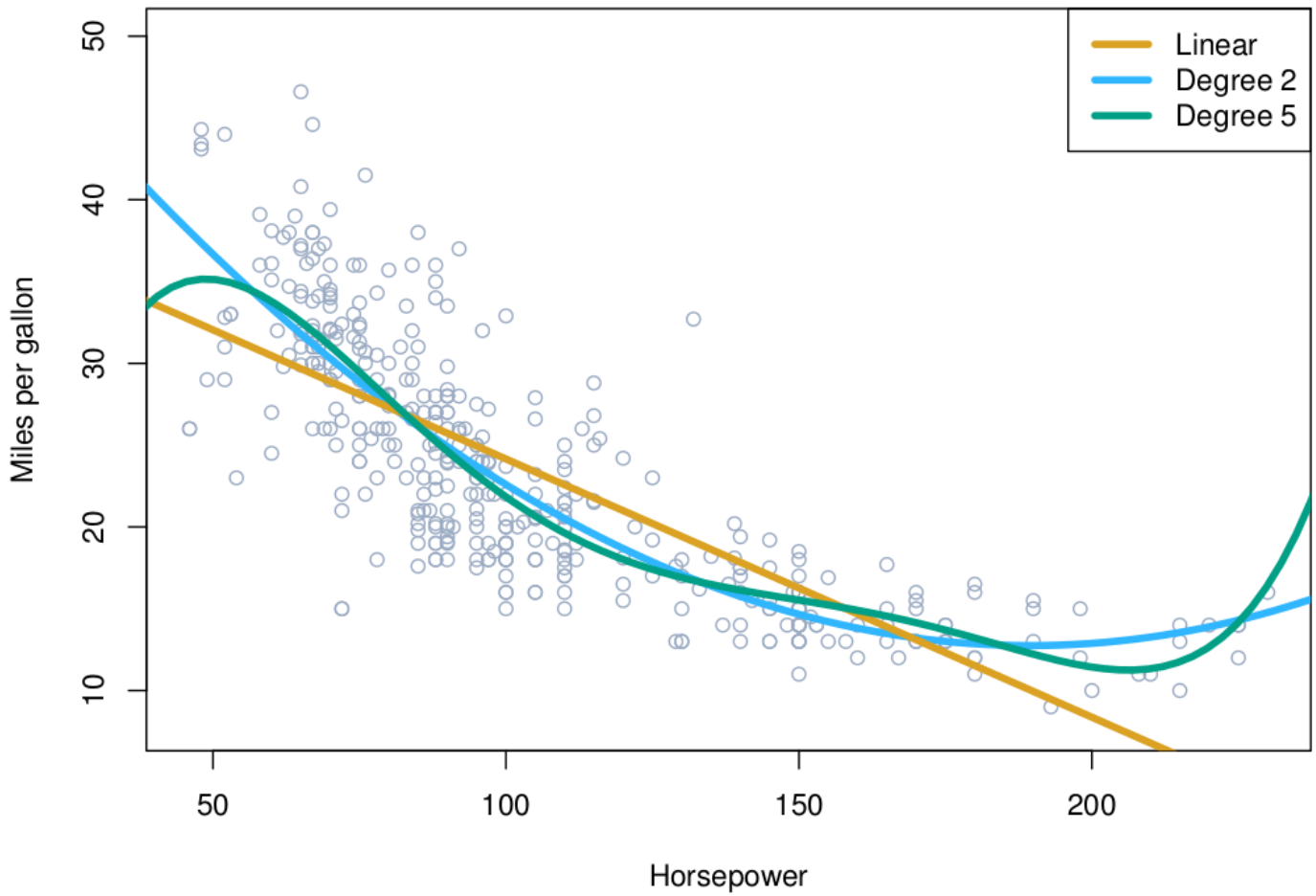
Model Selection

- A validation set is needed because we cannot use the test set to both help select the best model and assess the performance of the final hypothesis.
- Once a data set has been used in the learning process, it is "contaminated" – evaluations of performance based on such data will have optimistic bias, and the error calculated on the data set will not have a sufficiently conservative bound.

Model Selection

- Often, the different models in $\mathbb{H}$ can be indexed by a complexity parameter λ. Choosing the best model amounts to finding the best value of λ.
- For example, when choosing among polynomial models of varying degree, λ is often the order of the polynomial.
- As in the case of polynomials, this form of indexing typically nests/orders the models within $\mathbb{H}$ in a natural way, an approach suggested very early in the literature, e.g., Grenander's (1981) "method of sieves."
- Using the validation set approach, the training set is used to fit each model with a given λ, the validation set is the set on which λ itself is fit, while the test set is used to estimate the true out-of-sample performance of the final hypothesis.

MPG and Horsepower



MPG and Horsepower

- We want to compare linear and higher-order polynomial terms in a linear regression of miles per gallon (MPG) on horsepower using information on 392 vehicles.

- Suppose there exists an independent test data set somewhere else, so we only need this data set for model selection (without assessment). Then we can randomly split the 392 observations into two sets, a training set containing 196 of the data points, and a validation set containing the remaining 196 observations.

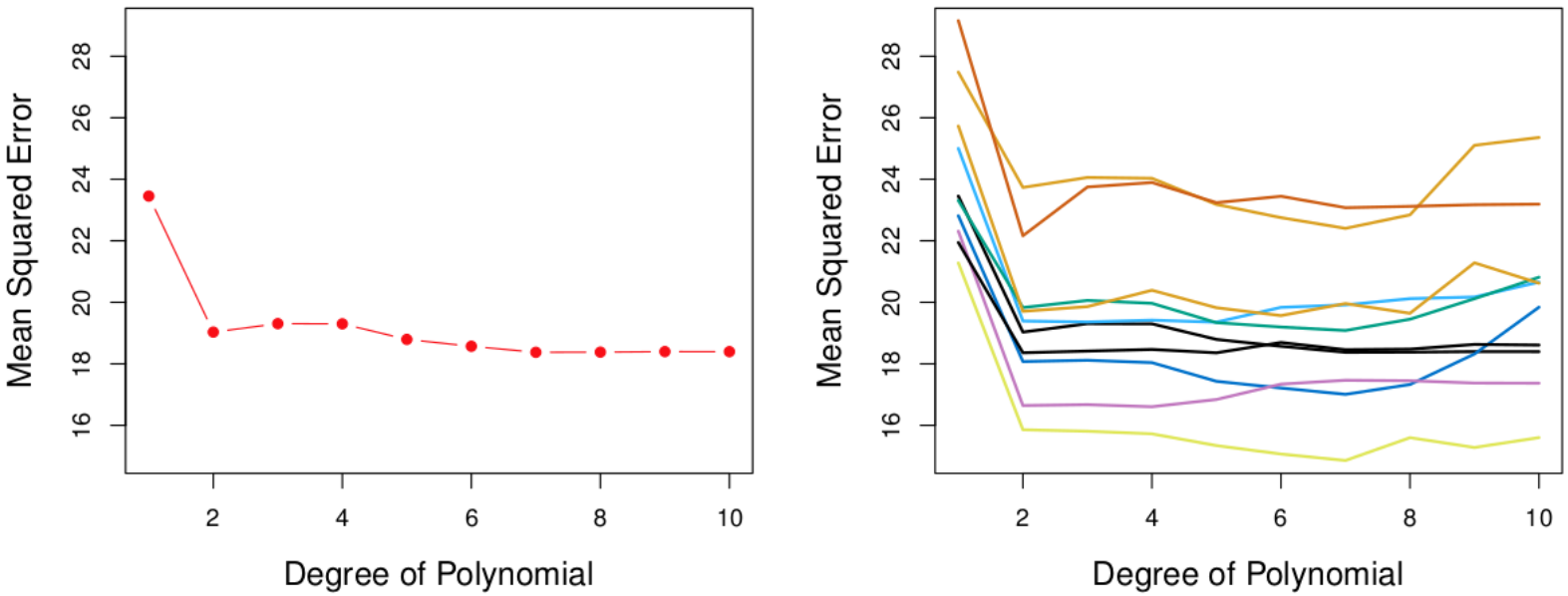
MPG and Horsepower

```
> require(ISLR)
Loading required package:  ISLR
> attach(Auto)
The following object is masked from package:ggplot2:
   mpg
> train <- sample(392,196) # draw a random subset from the sample
> fit <- lm(mpg ~ horsepower, subset=train)
> yhat <- predict(fit,Auto)
> V.e <- (mpg - yhat)[-train] # validation set error
> V.MSE <- mean(V.e^2) # validation set MSE
> V.MSE
[1] 22.65726
```

MPG and Horsepower

```
> V.MSE <- rep(0,5)
> for (i in 1:5){
+ fit <- lm(mpg ~ poly(horsepower,i), subset=train)
+ V.e <- (mpg - predict(fit,Auto))[-train]
+ V.MSE[i] <- mean(V.e^2)
+ }
> V.MSE
[1] 22.65726 18.03085 18.10659 18.02565 17.53585
```

MPG and Horsepower



Left: Validation-set MSE for a single split into training and validation data sets. Right: The validation method was repeated ten times, each time using a different random split.

The Validation Set Approach

Problems:

- Results can be highly variable, depending on which observations are included in the training set and which are in the validation set.
- The larger the validation set is, the smaller the training set has to be, hence the better the validation set errors are as estimates of true out-of-sample errors, but the poorer the model fits produced by the training set are.
- Conversely, the smaller the validation set, the better the model fits on the training set, but the poorer the validation set errors are as estimates of true out-of-sample errors.

Cross Validation

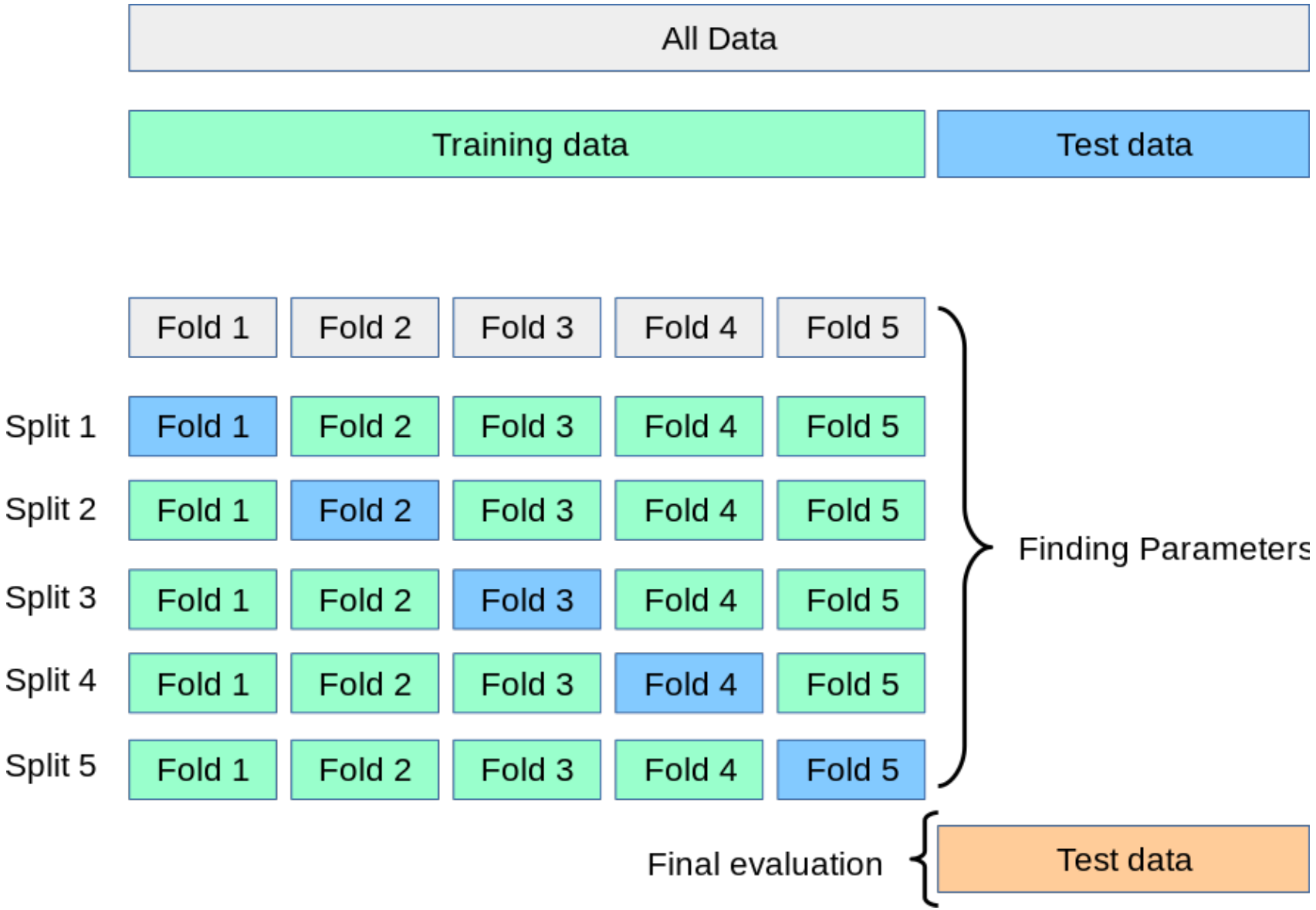
- The results of the validation set approach are variable due to the whims of a single random split. Thus, it might be better to randomly split the data multiple times and average the results – this is what cross-validation does.

- Like the bootstrap, cross-validation is a resampling method: these methods involve repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain additional information about the fitted model.

Cross Validation

- The cross-validation approach splits the original data into two parts: a training set and a test set, with no separate validation set.
- The training set is then randomly divided into K equal-sized folds.
- In turn, training is performed on $K-1$ folds (combined), and the remaining fold is used for evaluation. The K evaluation results are then averaged to produce an estimate of a model's out-of-sample error.

Cross Validation



Leave-one-out CV

- When $K = n$, this method is equivalent to holding-out one observation at a time, and then using the results to predict the held-out case. Because of this, n-fold CV is also called leave-one-out CV (LOOCV).
- The LOOCV criterion is

$$\text{CV}_n = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_{(i)})^2,$$

  where $\widehat{y}_{(i)}$ is a predictor of y_i from the sample with the ith case held out. The LOOCV procedure selects the model for which CV_n is smallest.
- For a model estimated by OLS,

$$y_i - \widehat{y}_{(i)} = y_i - \mathbf{x}_i\widehat{\boldsymbol{\beta}}_{(i)} = \frac{\widehat{u}_i}{1 - h_{ii}},$$

  where $\widehat{\boldsymbol{\beta}}_{(i)}$ is the OLS estimate from the sample with the ith case held out and h_ii is the ith diagonal element of the "hat" matrix. Hence,

$$\text{CV}_n = \sum_{i=1}^{n}\left(\frac{\widehat{u}_i}{1 - h_{ii}}\right)^2.$$

Properties of LOOCV

- Under the classical linear model, $\mathbb{E}[\widehat{u}_i^2] = (1 - h_{ii})\sigma^2$ and so

$$\mathbb{E}[\mathsf{CV}_n] = \frac{\sigma^2}{n} \sum_{i=1}^{n} \frac{1}{1 - h_{ii}}$$

- If n is large and there is no high-leverage point (i.e., $h_{ii} << 1$ for all i), a first order Taylor series expansion of $(1 - h_{ii})^{-1}$ about $h_{ii} = 0$ gives

$$\mathbb{E}[\mathsf{CV}_n] \approx \frac{\sigma^2}{n} \sum_{i=1}^{n} (1 + h_{ii}) = \frac{\sigma^2}{n}(n + k)$$

- This implies

$$\mathbb{E}\left[\sum_{i=1}^{n} (y_i - \widehat{y}_{(i)})^2\right] = \mathbb{E}[n\mathsf{CV}_n] = (n + k)\sigma^2$$

so the LOOCV criterion is approximately unbiased for the MSPE of $\widehat{y}^*$.

K-fold CV

- Given K folds, with $2 \leq K \leq n$, let $\widehat{\text{MSPE}}_1$ denote the average squared prediction error from the first held-out fold. This procedure is repeated for each fold, resulting in K estimates $\widehat{\text{MSPE}}_1, \ldots, \widehat{\text{MSPE}}_K$. The K-fold CV criterion is computed by averaging these values,

$$\text{CV}_K = \frac{1}{K} \sum_{j=1}^{K} \widehat{\text{MSPE}}_j.$$

- Relative to LOOCV, K-fold CV is less expensive computationally, except in the case of OLS regressions, for which the computational burden is the same.
- Another advantage is that K-fold CV, while nearly unbiased for the MSPE, often gives more accurate estimates than LOOCV.
- The intuition for this is that LOOCV averages the output from n fitted models, each trained on an almost identical set of observations, so these outputs are highly positively correlated. In contrast, K-fold CV with $K << n$ (typically, $K = 5$ or $K = 10$) averages the output from K fitted models that are somewhat less correlated, as the overlap between the training sets in each model is smaller.

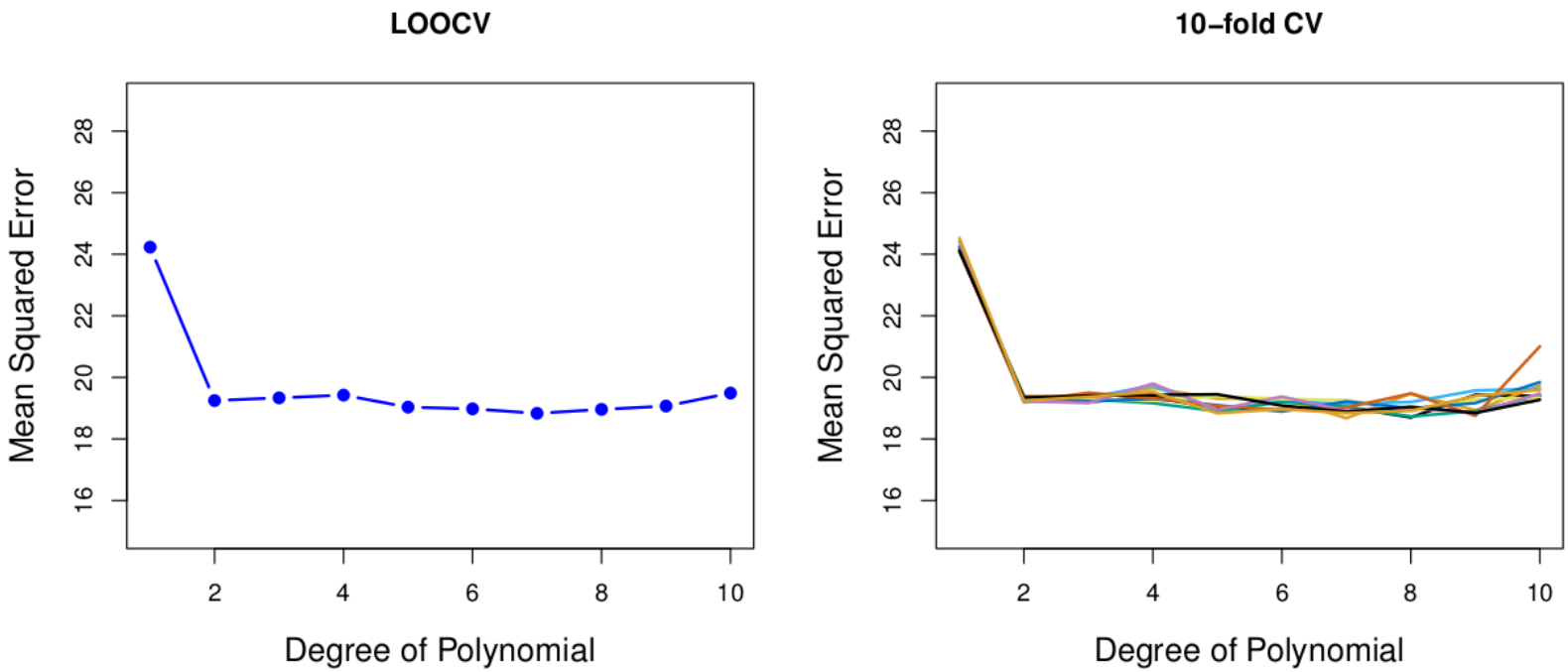
MPG and Horsepower

```
> ## LOOCV
> require(boot)
Loading required package:  boot
> LOOCV.MSE <- rep(0,5)
> for (i in 1:5){
+ fit <- glm(mpg ~ poly(horsepower,i))
+ LOOCV.MSE[i] <- cv.glm(Auto,fit)$delta[1]
+ }
> LOOCV.MSE
[1] 24.23151 19.24821 19.33498 19.42443 19.03321
```

MPG and Horsepower

```
> ## 10-fold CV
> CV.MSE <- rep(0,5)
> for (i in 1:5){
+ fit <- glm(mpg ~ poly(horsepower,i))
+ CV.MSE[i] <- cv.glm(Auto,fit,K=10)$delta[1]
+ }
> CV.MSE
[1] 24.38393 19.18221 19.26307 19.24297 19.32044
```

Cross Validation



Left: The LOOCV error curve. Right: 10-fold CV was run nine separate times, each with a different random split of the data into ten folds. The figure shows the nine slightly different CV error curves.