Model fitting and validation

Reasons to estimate $f$:
- Understand the relationship between $X$ and $Y$.
- Test hypotheses about the strength and relationship of $X$ on $Y$.

Finding a good model:
- How do we pick a model or learning algorithm?
- In practice, this is conditional on:
  - Domain (what is state of the art in the field).
  - Required interpretability.
  - Amount of data.
- Rule of thumb:
  - The more complex model, the more data we need.
  - Closer (or more confident) we want in predictions, the more data we need.
  - Can't really use the same data for both.

Interpretability vs. Flexibility:

Estimating $f$:
- Data Science often revolves around estimating the true relationship between $Y$ and $X$.
  $$ Y = f(X) + \epsilon. $$
  $$ \hat{Y} = \hat{f}(X) $$
- Note on notation: I may sometimes use $f$ for the true $Y$ and $\hat{f}$ for our estimated $Y$. (normally, we'd use $f$ and $\hat{f}$ with a hat)...

Why estimate $f$?
- It allows us to understand the relationship between $X$ and $Y$.
- It tests our hypotheses about the strength, relationship, and possibly even interpretation of $X$ on $Y$.

Interpretability vs. Flexibility:

- Linear model: Strong assumptions about relationship between $X$ and $Y$.
- Neural networks: Few assumptions about relationship, once relationship is uncovered, difficult to interpret.
We are often interested in understanding the way that our prediction for well-being on that day changes as a function of $X, ..., X$. The relationship between the input and output variables is of interest to us. 

Given predictor may also depend on the values of the other predictors. The application of well-being on that day.

• If our model is too complex, our model is more capable of accounting for the observed training data.
• This includes capturing the noise and error (or variance) of these data.
• If our model is too simplistic (e.g., linear model for a quadratic system), we cannot account for the observed training data perfectly.
• In fact our estimates will be biased.
• So it’s important to find the right point on that line.

General model performance

• Can we know anything about model selection and learning before we start?
• E.g. Is learning feasible?
• There’s a whole literature on this (statistical/computational learning theory) that we won’t cover here.
• We cover this much more in ML course (spring 2018) as well.

Evaluating model accuracy

• How good of a model is our estimated model at actually approximating $f$?
• Two types of error
  ▪ reducible: we can build better and better models to close the gap between the true model and the model we are using to estimate
  ▪ irreducible (i.e.,)

$$E[(Y - Y')^2] = E[|f(X) + \epsilon - \hat{f}(X)|^2]$$

$$= \int |f(X) - \hat{f}(X)|^2 + \text{Var}(\epsilon) \cdot \text{Reducible, Irreducible}$$
Figure 2.10 provides another example in which the true relationship is given by the black curve. The orange, blue and green curves illustrate three possible estimates for the relationship, varying between different levels of smoothness. It is evident that as the level of flexibility increases, the curves fit the observed data more closely. The green curve is the most flexible and matches the observed data best, while the orange curve is the least flexible and deviates the most from the observed data.

Though the mathematical proof is beyond the scope of this book, it is clear that as the level of flexibility increases, the curves fit the observed data better. However, because the training MSE and the test MSE appear to be closely related, there is no guarantee that the method with the lowest training MSE will also have the lowest test MSE. Unfortunately, there is a fundamental problem with this strategy: there is no test data available. As the previous three examples illustrate, the training and test MSE curves still exhibit the same general patterns, but the test MSE is often much larger than the training MSE, even though the squares fit is substantially better than the highly flexible green curve. This suggests that simply selecting a statistical learning method that minimizes the training MSE is not sufficient for estimating the test MSE.

**Intuition: Variance of \( f \)**

- How much would our estimated \( f \) change if we estimated it using different training data?
- If we removed a point, how much would our estimate change?
- If we reran our data collection, we’d get different observations. Would that change the fit of the green line? of the yellow line?

**Intuition: Bias of \( f \)**

- Introduced by the assumption of our estimated \( h(x) \)
- \( E[\epsilon] = 0 \)
- Expected difference between true \( f \) and our hypothesis.
- For example, linear regression assumes a linear relationship. Probably not entirely true so our estimated outcomes are biased.
- Bias cannot be reduced by more training data but only by a more complex model.
Bias-Variance tradeoff

- We want simple interpretable models
- But with simple models comes biases (e.g., assumptions of linearity)
- As we increase flexibility, we decrease bias.
- The cost of flexibility though is increased variance.

The basics of model evaluation

- We train a model on observations where we know both X and Y.
- We test our model by asking, given X, what do we expect Y to be.
- To understand how our model performs we look at the difference between our expected Y and our observed Y on the test set.
- Error is computed as a cost function based on the difference between estimated Y and true Y.

\[ MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2, \]

Evaluating model accuracy

- How good of a model is our estimated model at actually approximating f?
- Two types of error
  - reducible: we can build better and better models to close the gap between the true model and the model we are using to estimate
  - irreducible (\(\epsilon\))

\[
E(Y - \hat{Y})^2 = E[f(X) + \epsilon - \hat{f}(X)]^2 \\
= [f(X) - \hat{f}(X)]^2 + \text{Var(\(\epsilon\))},
\]

Reducible Irreducible

Accuracy of a model

- We want to minimize the difference between the true outcomes Y and our estimates of Y.
- We may want to constrain the form of our model
  - interpretability
  - forces the model to consider specific types of relationships
  - assess change over time (e.g., longitudinal, time course modes)

Model selection and validation

- Used for Model selection
  - What’s the best parameterization of a model?
  - E.g. how many neighbors in kNN
  - Or the regularization parameter in Ridge Regression
- Performance estimation
  - How do we evaluate the performance of a particular model
  - Standard training and testing split overestimates model performance
Limited data…

- These questions are easy to answer with infinite data and a finite set of estimates
  - Just choose the model with the lowest error
  - Why? Because the sample converges almost surely to the population.
  - So the parameter estimates do as well.
- Solution?

Model selection: finite data

- Sol. 1: Use all the training data
  - The model usually overfits the training data
  - Especially in the case where there are lots of parameters
  - Error rate is optimistic
  - We can usually get training data down to 0%.

Malcolm Gladwell’s blink

- First chapter reports that the can predict 80% of couples who will get a divorce by watching 10 minute of an argument between couples.
- It’s actually not true… The researchers (John Gottman) did not use model validation.
- And the research hasn’t held up.

Better solution

- Split training data into disjoint subsets
  - Hold out some data to see how model generalizes to unseen data
- There are a few methods we can use for this
  - Holdout method
  - Random Subsampling
  - K-fold cross-validation
  - Leave-one-out Cross Validation
  - Bootstrapping/Boosting

Better solution

- Split training data into disjoint subsets
  - Hold out some data to see how model generalizes to unseen data
- There are a few methods we can use for this
  - Holdout method
  - Random Subsampling
  - K-fold cross-validation
  - Leave-one-out Cross Validation
  - Bootstrapping/Boosting

Holdout method

- Split dataset into training and testing
  - Training: used to estimate parameters
  - Testing: estimate error rate of trained classifier (prediction error)
- Choose parameter setting where testing error begins to increase
Holdout method issues

- It is volatile to ‘bad’ splits in the data.
- If the training data somehow is different than the testing data this model will perform poorly.
- It is an optimistic estimate of extension to unseen data.
  - Frankly, the test set is being used to select parameters.
  - The minimal error of the test data is the best estimate…

Random Subsampling

- Perform K data splits
- Randomly select a specific number of examples w.o. replacement
- Train a model on each of the data, evaluate performance on test data.
- Compute the expected value of the parameter by averaging parameter estimates across experiments.

K-fold Cross-validation

- Create a K-fold partition of the data
- Run K experiments, using K-1 folds for training, 1 fold for testing.

Random subsample vs K-fold CV

- Advantage of K-fold is that we know exactly how many experiments to run to have each example be in the testing set.
- Requirement of final project: use K-folds cross-validation.
- Both average estimates across multiple runs
  - More accurate than the holdout method.

Leave one out CV

- Run n experiments
  - Training data n-1 data points, testing 1 data point
- Common for time series analysis.
- Useful in sparse datasets or small data sets.
  - Cannot generalize to unseen examples.

How many folds are needed?

- Large numbers of folds
  - Bias of true error estimate will be low, the estimator is accurate
  - But the variance will be high. Why?
  - Computationally intensive
- Small number of folds
  - Variance decreases
  - What happens to bias?
  - The number of experiments is smaller
  - Computational time is less
- In practice, it depends on the complexity of the model, amount of data, sparsity of observations.
Better than Cross-Validation

- Split the data three ways.
  - One becomes testing set. This NEVER gets looked at while selecting features/parameters/etc. of model.
  - Validation and test set become initial folds in k-fold validation.

Three way data split

- **Training set**: set of examples used to train the model
- **Validation set**: used to optimize parameters based on extension to unseen data
- **Test set**: used to evaluate the model only after fully training and fixing parameters of models
- Kaggle: Validation set is public leaderboard, test set is private leaderboard.
- You get a good idea of how the model would perform if deployed in the real world. (evidence that prediction error is low)