Model selection and validation

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 if there are lots of features.
  - 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 minutes 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)

There are methods we can use for this:
- Holdout method
- Random Subsampling
- K-fold cross-validation
- Leave-one-out Cross Validation
- Bootstrapping/Boosting

Holdout method issues

- It is volatile due 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)

Machine learning overview

A whirlwind tour of some common algorithms
(not instruction, just exposure... come back Spring '18 for instruction)
Deciding which algorithm to use

- First consideration: (supervised or unsupervised)
  Is there an underlying truth? Do you know exactly what values Y should be taking?
- Second consideration: (classification vs regression)
  Are you predicting a probability, class/cluster or value
- Third consideration:
  Are you interested in predictive accuracy or understanding influence of predictors

Unsupervised vs supervised

- Mostly we’ll work in supervised....
- Considering both regression and classification
- With parametric models

Regression/Classification/Clustering

- Values vs labels
- All regression problems can be turned into a classification problem (e.g. Does the house cost more than 100k)
- Classification problems are either predicting labels or predicting probabilities of events
- Clustering is detecting communities/groups within a dataset

Linear regression

![Linear regression graph]

Logistic Regression

![Logistic Regression graph]

K nearest neighbors

![K nearest neighbors graph]
K means clustering

Naïve Bayes Classifier
- Assume everything is independent given class.

Support Vector Machines

Support Vector Machines: Kernels

Artificial Neural Networks