Statistical Learning

Contents

Introduction

This is a set of notes based on the Stanford Statistical Learning course by Trevor Hastie and Rob Tibshirani.

Videos for the course are listed here.

The ISLR book is available for download here.

Chapter 2: Statistical Learning

Chapter 3: Linear Regression

Linear Regression

- Allows to easily explain dependencies between predictors and target, but can miss non-linear relationships.
- Least squares to estimate parameters - the least squares approach chooses params to minimize RSS:
  - Assume a model: $Y = \beta_0 + \beta_1 X + \epsilon$
  - Predictions: $\hat{y_i} = \hat{\beta_0} + \hat{\beta_1} x_i$
  - Residuals: $e_i = y_i - \hat{y_i}$
  - $RSS = e_1^2 + ... + e_n^2$ - Residual Sum of Squares, total squared discrepancy
- Use Standard Error to assess accuracy of coefficient estimates
  - SE reflects how estimator varies under repeated sampling.
  - SE for slope: $SE(\hat{B_1})^2 = \sigma^2 / (\sum_{i=1}^{n} (x_i - \bar{x})^2)$
    - ratio of the variance (noise) to the spread of the x around their means
    - bigger when there is more noise, and smaller when the x's are more spread out.
  - Confidence Intervals: $param \pm 2 * SE(param)$. A 95% CI is the range of values such that with 95% probability, the range will contain the true unknown value of the parameter.

Hypothesis testing in linear regression

- A hypothesis test for a parameter, testing the null hypothesis:
  - H_0: There is no relationship between X and Y
  - H_A: Alternative hypothesis, there is some relationship between X and Y.
  - For regression parameters: $H_0 : \beta_1 = 0$, $H_A : \beta_1 \neq 0$

- Compute a `t-statistic` to test the null hypothesis. This is a normal random variable?
  - $t = (\hat{\beta_1} - 0)/SE(\hat{\beta_1})$ is the t-distribution with n-2 DOF (degrees of freedom), assuming $\beta_1 = 0$
  - then compute p-value based on t-statistic. A p-value is probability of observing any value equal to |t| or larger.
- If the H-test fails (we reject H_0), the CI for that parameter will not contain 0.

Assessing overal accuracy of the model

- Residual Standard Error: $RSE = \sqrt{RSS/(n-2)}$
  - RSE is an estimate of the standard deviation of ϵ, the residual.
- Total Sum of Squares can be thought of as assessing the accuracy of the "no-predictor" model
  - $TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$
- Residual Sum of Squares (RSS) can be thought of as accuracy for the model that uses all predictors
- R-squared measures how much the TSS is reduced relative to itself. It is the fraction of variance explained.
  - $R^2 = (TSS - RSS)/TSS = 1 - RSS/TSS$
- Example in R:

Multiple Linear Regression

- Assume a model: $Y = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p + \epsilon$
- Predictions: $\hat{y_i} = \hat{\beta_0} + \hat{\beta_1} x_1 + ... + \hat{\beta_p} x_p$
- Correlations amongst predictors cause problems. Ideally, the predictors are uncorrelated, and thus can be estimated and tested separately. When X are correlated, the variance of all coefficients tends to increase, and interpretations become hazardous.
- Use R-squared and p-values for looking at relationships. In the presence of certain features, other features might become insignificant.

Additional topics on regression

- F-ratio (or F-statistic): can be used to determine if predictors are useful: Large F means that predictors have a large effect on the target
  - F-Ratio = (Drop in training error divided by number of params) divided by (mean squared residual divided by (sample size - num of params - 1))

$$F = \frac{(TSS - RSS)/p}{RSS/(n-p-1)}$$

- Deciding on important variables
  - Forward selection: beginning with the `null model` (intercept only), add variables one at a time until some stopping rule is satisfied, for example when all remaining vrs have a significant p-value.
  - Backward selection: start with all variables, remove variables with least significance. Look at the t-statistics to determine significance.

Extensions of the Linear model

- Interaction terms: multiply two terms and multiply that by a coefficient
  - this can be rewritten as the original terms with the coefficients now including terms from the interaction
  - when including interaction terms, also include the original components of the interaction term separately

  - to compute the additional variance explained when adding interaction terms, assuming new var is higher:
    - (var2 - var1)/(100 - var1), where var1 is the variance explained by the non-interaction model

- non-linear effects: add polynomial terms like x^2 and x_1*x_2 to capture non-linearities
- Example in R:

```
#multiple terms, interaction terms, polynomial terms
```

Chapter 4: Classification

Classification

- $p(X) = Pr(Y = 1|X)$ for a 2-class problem. Then, logistic regression uses the form:

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

- This is the `log odds` or `logit` transformation of p(X): $log(p(X)/(1 - p(X)) = \beta_0 + \beta_1 X$. So, the linear model is modeling probabilities on a nonlinear scale. The probabilities lie on a scale between 0 and 1.
- Max Likelihood: used to estimate the regression parameters. The likelihood gives the probability of observed target in the data. Pick params to maximize the likelihood:

$$l(\beta_0, beta) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} 1 - p(x_i)$$

  Use Z-statistic and p-value to determine significance of the coefficients.

Case-control sampling and logistic regression

- When classes are skewed, use case-control sampling. Build a model with more equal classes, and then correct the estimated intercept by a transformation.

Lindear Discriminant Analysis and Bayes Theorem - TODO

Multivariate Linear Discriminant Analysis and ROC curves - TODO

Quadratic Discriminant Analysis and Naive Bayes - TODO

- Example in R:

```
# TODO: logistic model
```

Chapter 5: Resampling methods

Cross-validation and bootstrap are ways of resampling from the training set, in order to obtain additional info about the fitted model. E.g., estimates of test-set prediction error (cv), and std dev and bias of param estimates (bootstrap).

Training vs test set performance (bias-variance tradeoff)

- Bias: how far on average the model is from the truth
- Variance: how much the estimate varies around its average
- A plot of test and training error vs increasing model complexity will show that training error generally decreaases, but test error will decrease and at some point start increasing with higher complexity. This plot goes from a high-bias, low-variance state to low-bias, high-variance state with higher model complexity.
  - At low complexity, bias is high because the model is not accurate, and variance is low because there are few parameters being fit.
  - With higher complexity, bias decreases because the model adapts to the subtleties in the data, but variance increases because there are more parameters to estimate.

Solutions to testing model accuracy

1. Large designated test set, often not available.
2. Mathematical adjustment to the training error rate to estimate the test error rate: Cp statistic, AIC, BIC. These are available for some models.
3. Validation and cross-validation: estimate the test error by applying model to a held out subset of the training data.

Validation set

- Procedure
  - Divide samples into training and validation (hold-out) sets
  - Fit model to train set, check error on validation set
  - Validation error is an estimate of the test error
- Drawbacks
  - Validation estimate of test error can be highly variable, depending on precisely which observations are included in which set
  - Only a subset of the observations are used to fit the model (e.g., 50% or 70% that is used for training)
  - this suggests that the validation set error may tend to overestimate the test error for the model fit on the entire dataset (since the model is fit on less data, it will be less accurate)

K-fold cross-validation

- Procedure
  - randomly divide data into K equal-sized parts
  - Fit model on K-1 parts (combined)
  - Predict on kth part
  - Do this in turn for each part k=1,2,…,K, then combine results
- Estimates can be used to select best model, and to give an idea of the test error of the final model.
- Details
  - n_k = number of observations in kth part; $n_k = n/K$ if N is a multiple of K
  - $CV_{(K)} = \sum_{k=1}^{K} (n_k/n) MSE_k$, where $MSE_k = \sum_{iinC_k} (y_i - \hat{y_i})^2/n_k$
- LOOCV (leave one out cross validation): special case with K=n; also called n-fold cross-validation
  - For least-squares linear or polynomial regression, the cost of LOOCV is the same as that of a single model fit; can be achieved using leverage
  - for LOOCV, estimates from each fold are highly correlated (since the training sets are different by only 1 observation), and hence their average can have high variance (but low bias)
  - better choice is k=5 or 10, but there is a bias-variance tradeoff

- Other issues with cross-validation
  - Since each training set is only (K-1)/K as the original training set, the estimates of prediction error will be biased upward
  - This bias is minimized when K=n (LOOCV), but the estimate has high variance.
- Estimated standard deviation of CV_k (standard error of CV_k):
  - put formula here
  - gives an idea of how variable the CV error is
  - not quite valide since the errors are correlated, but can be used in practice

The Bootstrap

Flexible and powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical method. E.g., can provide estimate of the standard error of a coefficient, or a CI for that coefficient.

Idea: We cannot sample repeated independent data sets from the population, because the population is usually not available. Instead, sample from the data set *with replacement*: the data set (sample) acts as a population.

Uses of the bootstrap:

- Primarily used to obtain standard errors of an estimate
- Also provides *approximate* confidence intervals for a population parameter ("Bootstrap percentile"). Can use a histogram to get 5% and 95% quantiles, for example.
- Bootstrap cannot estimate prediction error.
  - In CV, each of the K folds is distinct from the other K-1 folds used for training - there is no overlap.
  - Each bootstrap sample has significant overlap with the original data (2/3 of original data appear in each bootstrap sample). Bootstrapping severely underestimates prediction error.

Chapter 6: Linear Model Selection and Regularization - TODO

- There are several ways by which the simple linear model can be improved, by replacing ordinary least squares fitting with some alternative fitting procedures.
- Reasons to consider alternatives to least squares
  - Prediction accuracy: especially when p > n (num features > num samples), to control the variance
  - Model interpretability: by removing irrelevant features (setting the corresponding coefficient estimates to 0), obtain a model that is more easily interpreted.
- Three classes of methods
  - Subset selection: find a subset of the p predictors (forward, backward, subsets)
  - Shrinkage (Regularization): use all p predictors, but shrink coef estimates towards 0. this reduces variance and also performs variable selection.
  - Dimension reduction: project p predictors into a M-dimensional subspace, M < p.

Best Subset Selection

- Consider all combinations of predictors
- Procedure
  - Let M_0 denote the *null model*, which contains no features and predicts the sample mean.
  - For k = 1,2, …, p:
    - fit all (p choose k) models that contain exactly k predictors; p!/(2! * (p-2)!)

        - pick the best (having smallest RSS or largest R^2) and call it M_k.
    - Select a single best model from among $M_0, \ldots\ M_p$ using CV prediction error, Cp (AIC), BIC, or adjusted R^2. This error is on a different data than in the previous step.
- p can get large, and best subset selection does not scale well. This can also overfit the data.
- Extensions to other models (logistic regression, etc.)
    - The *deviance* (-2 * max log-likelihood) plays the role of RSS for a broader class of models.

Forward Stepwise Selection

- Procedure
    - Begin with null model, and add predictors one at a time, until all predictors are in the model.
        - At each step, the variable that gives the greatest *additional* improvement to the fit is added to the model.
    - Select a single best model among $M_0, \ldots\ M_p$ using CV or other error.
- FSS considers p^2 models instead of 2^p as in best subset selection
- FSS is not guaranteed to find the best model.
- Can be used even when n < p, and is the only viable subset method when p is very large.

Backward Stepwise Selection

- Procedure

    - Start with all predictors, and remove predictors one at a time until the null model.
        - At each step, remove the least useful predictor (one that causes the least reduction in RSS)
    - Select a single best model as before.

- Backward stepwise selection searches through 1 + p(p+1)/2 models

- Not guaranteed to yield the best model.

- Requires that n > p (otherwise cannot fit a least squares?)

- Choosing the Optimal Model

    - the model with all predictors will always have smallest RSS and largest R^2, since these quantities are related to the training error. Since we want a model with low test error, RSS and R^2 are not suitable for selecting the best model among a collection of models with different numbers of predictors.

Estimating Test Error Using Mallow's Cp, AIC, BIC, and Adjusted R^2

- These techniques adjust the training error for the model size, and can be used to select among a set of models with different numbers of variables.

Estimating Test Error Using Cross-Validation

Chapter 7: Moving Beyond Linearity - TODO

Chapter 8: Tree-based methods

Regression trees: top-down, greedy approach

- Procedure

    - select the predictor X_j and the cutpoint s such that splitting the predictor space into two regions leads to the greatest possible reduction in RSS.
    - Repeat the process, looking for the best predictor and best cutpoint to minimize RSS within each of the resulting regions.
    - Stop when some criteria is reached; e.g., each region contains no more than 5 observations.
- Predictions on new data: pass a test case down the tree, and use the mean of the training results in that region to make a prediction
- Tree pruning
    - if a tree is so large that each observation has its own terminal node, the tree is overfitting the data.
    - better strategy is to grow a large tree and then prune it to obtain a subtree (AKA cost complexity pruning, weakest link pruning).
    - Use a regularization parameter (α) to avoid overfitting. Tune α and then return the subtree with the best α.

Classification trees

- Predict using the most common class in region
- Alternatives to RSS for classification:
    - Classification error rate, but this is too noisy (?)
    - Gini index: measure of total variance across the K classes, takes on small value if all $\hat{p}_{mk}$ are close to 0 or 1.
    - Referred to as a measure of node purity: a small value indicates that node contains mostly one class
    - Cross-entropy: alternative to Gini index

Bagging

- Procedure
    - Take repeated samples from the single training set
    - Grow a tree on each sample
    - Take an average of the predictions (or majority vote)
- Averaging a set of observations reduces variance
- OOB error estimation: each bagged tree only uses 2/3 of the observations. The remaining 1/3 not used to fit a tree are the OOB observations. Using the trees in which an observation was OOB, predict the observation, and average the result. This is essentially LOO (leave one out) cross-validation.

Random Forests

- Procedure
    - Bagging, but only consider a random selection m of predictors when making a split point. Typically, $m = sqrt(p)$, where p is the full set of predictors used in bagging. This reduces bias toward the most influential factors and allows secondary factors to play a role in the model.
- Build trees in such a way as to make the correlation between the trees smaller (even smaller than from bootstrapping). This reduces variance when averaging the trees.

Boosting

- Idea:
    - Trees are grown sequentially; each tree is grown using information from previously grown trees (residuals)
    - Can be applied to other methods, not just decision trees

- Procedure for regression:
    - Set $\hat{f}(x) = 0$, an average of trees, and $r_i = y_i$ for all i in training set (r=residuals)
    - For b = 1,2,..,B, repeat:
        - Fit tree $\hat{f}^b$ with d splits (d+1 terminal nodes) to the training data (X, r), where r is current residuals
        - Update $\hat{f}$ by adding in a shrunken version of the new tree (shrunk by factor λ): $\hat{f}(x) < -\hat{f}(x) + \lambda \hat{f}^b$
        - Update residuals: $r_i < -r_i - \lambda \hat{f}^b(x_i)$
    - At the end, output the boosted model, which is a sum of shrunken trees: $\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$
- This approach learns slowly, by picking up a small piece of the signal with each new tree. Trees here can be small.
- Tuning parameters for boosting
    - number of trees B: unlike in RF, boosting can overfit if B is too large.
    - shrinkage param λ: 0.01 to 0.001. Very small λ may require very large B.
    - boosting depth: depth=1 means that each tree only has 1 split ("stump"). Larger d allows more predictor interactions.

Variable importance measure

- There is no single coefficient with SE to use, since variables are used in multiple places.
- For bagged/RF trees, record the total amt that RSS (or Gini index) is decreased due to splits over a given predictor, averaged over B trees.

Chapter 9: Support Vector Machines - TODO

Chapter 10: Unsupervised Learning - TODO