|  |  |
| --- | --- |
| **Subset Selection** | |
| What do best subset, forward stepwise, and backward stepwise do in general? | The subset selection methods involve using least squares to fit a linear model that contains a subset of the predictors |
| Best Subset Selection | * To perform best subset selection, we fit a separate least squares regression best subset for each possible combination of the p predictors * We then look at all of the resulting models, with the goal of identifying the one that is best. |
| Permutation formula for Best subset selection | * p(p-1)/n where p is the total number predictors and n is the amount contained in the model in question (i.e. all p(p-1)/2 models that contain exactly two predictors) * For total combinations, it’s just 2^p |
| Stages for selecting the best model from among 2^p possibilities with best subset | 1. Let M0 denote the null model, which contains no predictors. This model simply predicts the sample mean for each observation. 2. For k = 1, 2,...p:    1. Fit all (p k) models that contain exactly k predictors.    2. Pick the best among these (p k) models, and call it Mk. Here best is defined as having the smallest RSS, or equivalently largest R2. 3. Select a single best model from among M0,...,Mp using crossvalidated prediction error, Cp (AIC), BIC, or adjusted R2. [Must be done due to bias-variance trade off]    1. With logistic regressions, we use deviance: The deviance is negative two times the maximized log-likelihood; the smaller the deviance, the better the fit |
| Trajectory of errors in best subset selection as number of features included in the model increases | * RSS of these p + 1 models decreases monotonically * R2 increases monotonically, as the number of features included in the models increases. |
| Problem with Best subset selection | * Suffers from computational limits and often becomes computationally impossible for p greater than around 40 * Statistical problems when p is large: The larger the search space, the higher the chance of finding models that look good on the training data, even though they might not have any predictive power on future data (i.e. can lead to overfitting) |
| Are the predictors in the k-variable model identified by best subset a subset of the predictors in the (k + 1)-variable model identified by best subset selection? | No. Because of the different combinatorial possibilities in the k-variable model versus the (k+1) variable model, they are not guaranteed to select the same best model. |
| **Stepwise Selection** | |
| Forward Stepwise Selection | * Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model. * At each step the variable that gives the greatest additional improvement to the fit is added to the model. |
| Algorithm for Forward Stepwise (Greedy search) | 1. Let M0 denote the null model, which contains no predictors. 2. For k = 0,...,p − 1:    * Consider all p − k models that augment the predictors in Mk with one additional predictor.    * Choose the best among these p − k models, and call it Mk+1. Here best is defined as having smallest RSS or highest R2. 3. Select a single best model from among M0,...,Mp using crossvalidated prediction error, Cp (AIC), BIC, or adjusted R2. |
| Permutation calculation for forward and backward stepwise | 1+p(p+ 1)/2 models where p is total predictors (so if it’s 20, we get 211) |
| Are the predictors in the k-variable model identified by forward stepwise a subset of the predictors in the (k+1)-variable model identified by forward stepwise selection? | Yes. Because we are moving forward in adding predictors, the k+1 variable model will include all k features chosen plus additional features that decrease error. Same for backward |
| Problem with forward stepwise and backward stepwise) | It is not guaranteed to find the best possible model out of all 2p models containing subsets of the p predictors (X1 may have been best in one var model, but X2 and X4 may have been better in the two var model instead of X1 and X4, but X1 has to be included in the two predictor model since it was the best in the one predictor model) (just think in reverse here for backward) |
| Advantage of forward stepwise over backward | * Can be applied even in the high dimensional setting where n < p (not less than or equal to though because it can only do submodels here) * Backward selection requires that the number of samples n is larger than the number of variables p (so that the full model can be fit). |
| Backward Stepwise Selection | Begins with the full least squares model containing all p predictors, and then iteratively removes the least useful predictor, one-at-a-time |
| Algorithm for backward stepwise selection | 1. Let Mp denote the full model, which contains all p predictors. 2. For k = p, p − 1,..., 1:    1. (a) Consider all k models that contain all but one of the predictors in Mk, for a total of k − 1 predictors.    2. (b) Choose the best among these k models, and call it Mk−1. Here best is defined as having smallest RSS or highest R2. 3. Select a single best model from among M0,...,Mp using crossvalidated prediction error, Cp (AIC), BIC, or adjusted R2. |
| How to select the best model for best, forward, and backward using test error | 1. We can indirectly estimate test error by making an adjustment to the training error to account for the bias due to overfitting. 2. We can directly estimate the test error, using either a validation set approach or a cross-validation approach |
| OLS Stepwise |  |
| Cp | * where is an estimate of the variance of the error associated with each response measuremen * Typically is estimated using the full model containing all predictors * Adds a penalty of to the training RSS in order to adjust for the fact that the training error tends to underestimate the test error * We choose the model with the lowest Cp |
| Akaike information criterion (AIC) | * Defined for a large class of models fit by maximum likelihood * We choose the model with the lowest AIC |
| Bayesian information criterion (BIC) | * We choose the model with the lowest BIC * Since log n > 2 for any n > 7, the BIC statistic generally places a heavier penalty on models with many variables, and hence results in the selection of smaller models than Cp |
| Adjusted R2 | * A large value of adjusted R2 indicates a model with a small test error * The intuition behind the adjusted R2 is that once all of the correct variables have been included in the model, adding additional noise variables will lead to only a very small decrease in RSS |
| What advantage does validation and cross validation offer over BIC, AIC, Cp, and Adjusted R2 | * Provides a direct estimate of the test error, and makes fewer assumptions about the true underlying model * It can also be used in a wider range of model selection tasks, even in cases where it is hard to pinpoint the model degrees of freedom (e.g. the number of predictors in the model) or hard to estimate the error variance |
| One standard error rule | * We first calculate the standard error of the estimated test MSE for each model size, and then select the smallest model for which the estimated test error is within one standard error of the lowest point on the curve * The rationale here is that if a set of models appear to be more or less equally good, then we might as well choose the simplest model—that is, the model with the smallest number of predictors |
| R code for subset selection | * regsubsets (leaps) (default is best subset, otherwise specify in method option for forward and backward |
| **Shrinkage and Regularized Regression** | |
| What do the shrinkage methods do in general? | * Fit a model containing all p predictors using a technique that constrains or regularizes the coefficient estimates, or equivalently, that shrinks the coefficient estimates towards zero * Good to use if there are a large number of predictors, some of which might be uncorrelated with Y, and if we might have more predictors than observations |
| How does regularized regression deal with large coefficients? | Large values of beta tend to be penalized, so that Betahat will tend to reveal smaller overall magnitudes of the parameters. |
| Ridge Regression | * Ridge regression coefficient estimates minimize * The second term, called a shrinkage penalty, is small when β1,...,βp are close to zero, and so it has the effect of shrinking the estimates of βj towards zero. * When λ = 0, the penalty term has no effect, and ridge regression will produce the least squares estimates.   + However, as λ → ∞, the impact of the shrinkage penalty grows, and the ridge regression coefficient estimates will approach zero |
| L2 Norm (Euclidean magnitude) (Ridge) | * Measures the distance of β from zero * As λ increases, the 2 norm of βˆR λ will always decrease |
| What must be done before applying ridge regression | Standardize the predictors to make sure they are on the same scale |
| Strength of ridge regression over OLS | * As λ increases, the flexibility of the ridge regression fit decreases, leading to decreased variance but increased bias   + For values of λ up to about 10, the variance decreases rapidly, with very little increase in bias, and the MSE drops considerably during that increase   + Test mean squared error (MSE) is a function of the variance plus the squared bias * At the least squares coefficient estimates, which correspond to ridge regression with λ = 0, the variance is high but there is no bias. * Beyond this point, the decrease in variance due to increasing λ slows, and the shrinkage on the coefficients causes them to be significantly underestimated, resulting in a large increase in the bias. * Ridge regression works best in situations where the least squares estimates have high variance |
| How does ridge regression handle situations where p > n | Ridge regression can still perform well by trading off a small increase in bias for a large decrease in variance |
| Strength of ridge over subset selection | Computational advantages: only fits a single model |
| Disadvantage of ridge regression | Cannot do feature selection because it can’t reach 0 but only approach it |
| Lasso |  |
| L1 Norm (Lasso) | the L1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large (So Lasso can do feature selection) |
| Lasso vs. OLS in terms of flexibility, bias-variance trade-off, and prediction accuracy | * Lasso will either be the same as least squares for a λ of 0 or less flexible than least squares for a λ of greater than 0 * As λ increases from 0 up to about 10, the increase in bias will be less than the decrease in variance. Because mean squared error is a function of the variance plus the squared bias, the MSE drops considerably * While a high λ may make the model less flexible, if the variance is rising significantly, then it means that a small change in the training data can cause a large change in the least squares coefficient estimate, thereby decreasing prediction accuracy |
| When might lasso perform better than ridge and vice versa | * In general, one might expect the lasso to perform better in a setting where a relatively small number of predictors have substantial coefficients, and the remaining predictors have coefficients that are very small or that equal zero. * Ridge regression will perform better when the response is a function of many predictors, all with coefficients of roughly equal size |
| How to choose lambda | * We choose a grid of λ values, and compute the cross-validation error for each value of λ * We then select the tuning parameter value for which the cross-validation error is smallest (We can see that at some point, they are all just about the same, so we'll calculate how far left or right until I get an SD away from the best one; We want least complex will still approximately same performance as the best one.) * Finally, the model is re-fit using all of the available observations and the selected value of the tuning parameter. |
| Relationship between lambda and MSE | * As λ increases, the mean squared error increases as well due to a larger bias (since the coefficients are forced to be small) |
| **Dimension Reduction** | |
| What does dimension reduction entail? | * Transforming the predictors and then fit a least squares model using the transformed variables * The principal components regression (PCR) approach involves constructing principal components regression the first M principal components, Z1,...,ZM, and then using these components as the predictors in a linear regression model that is fit using least squares. * It is not a feature selection method because each of the M principal components used in the regression is a linear combination of all p of the original features |
| How does dimension reduction handle situations where p is large relative to n? | In situations where p is large relative to n, selecting a value of M << p can significantly reduce the variance of the fitted coefficients |
| Two steps for dimension reduction | First, the transformed predictors Z1, Z2,...,ZM are obtained.  Second, the model is fit using these M predictors. |
| Principal Components Analysis | The first principal component direction of the data is that along which the observations vary the most (i.e. var that explains most variation in Y) |
| What is the first PC mathematically (i.e. how to get the PC score) | * where the are the pc loadings that indicate the direction and is the mean of all that var in the dataset * The idea is that out of every possible linear combination of pop and ad such that + = 1, this particular linear combination yields the highest variance |
| Bias-Variance Trade Off for PCA | As more PCs are used in the regression model, the bias decreases, but the variance increases |
| When might PCA do better than lasso or ridge? | PCR will tend to do well in cases when the first few principal components are sufficient to capture most of the variation in the predictors as well as the relationship with the response |
| How to determine number of PCs? | Cross-validation |
| What needs to be done before PCA? | * Standardization * In the absence of standardization, the high-variance variables will tend to play a larger role in the principal components obtained, and the scale on which the variables are measured will ultimately have an effect on the final PCR model. |
| Weakness of PCR | Consequently, PCR suffers from a drawback: there is no guarantee that the directions that best explain the predictors will also be the best directions to use for predicting the response |
| Partial Least Squares | * PLS identifies these new features in a supervised way—that is, it makes use of the response Y in order to identify new features that not only approximate the old features well, but also that are related to the response. * While the supervised dimension reduction of PLS can reduce bias, it also has the potential to increase variance, so that the overall benefit of PLS relative to PCR is a wash |
| **Non-Linearity** | |
| Step-functions | * Cut the range of a variable into K distinct regions in order to produce a qualitative variable. This has the effect of fitting a piecewise constant function * We break the range of X into bins, and fit a different constant in each bin * This amounts to converting a continuous variable into an ordered categorical variable |
| Weakness of step functions | Unless there are natural breakpoints in the predictors, piecewise-constant functions can miss the action (Hard to determine where the right break is) |
| Piecewise Polynomial | * Instead of fitting a high-degree polynomial over the entire range of X, piecewise polynomial regression involves fitting separate low-degree polynomials piecewise polynomial regression over different regions of X * In general, if we place K different knots throughout the range of X, then we will end up fitting K + 1 different cubic polynomials |
| Regression Splines | * More flexible than polynomials and step functions, and in fact are an extension of the two * They involve dividing the range of X into K distinct regions. * Within each region, a polynomial function is fit to the data * However, these polynomials are constrained so that they join smoothly at the region boundaries, or knots * The most direct way to represent a cubic spline using is to start off with a basis for a cubic polynomial—namely, x, x2, x3—and then add one truncated power basis function per knot |
| Determining where to place the knot in splines | * One option is to place more knots in places where we feel the function might vary most rapidly, and to place fewer knots where it seems more stable * Another option is to place knots in a uniform fashion * Cross-validation: With this method, we remove a portion of the data (say 10 %), fit a spline with a certain number of knots to the remaining data, and then use the spline to make predictions for the held-out portion. We repeat this process multiple times until each observation has been left out once, and then compute the overall cross-validated RSS. We loop this over different values of K and then choose the k that produced smallest average RSS |
| Strength of regression splines over polynomial | * Unlike polynomials, which must use a high degree (exponent in the highest monomial term, e.g. X15) to produce flexible fits, splines introduce flexibility by increasing the number of knots but keeping the degree fixed * Splines also allow us to place more knots, and hence flexibility, over regions where the function f seems to be changing rapidly, and fewer knots where f appears more stable |
| Weakness of splines | Splines can have high variance at the outer range of the predictors—that is, when X takes on either a very small or very large value |
| Smoothing Splines | Smoothing splines result from minimizing a residual sum of squares criterion subject to a smoothness penalty |
| Penalizing with smoothing splines | * The larger the value of λ, the smoother g will be * When λ = 0, then the penalty term in (7.11) has no effect, and so the function g will be very jumpy and will exactly interpolate the training observations * When λ → ∞, g will be perfectly smooth—it will just be a straight line that passes as closely as possible to the training points. (Will be at OLS) |
| Local regression | * Like splines, but the regions are allowed to overlap, and indeed they do so in a very smooth way. * In order to obtain the local regression fit at a new point, we need to fit a new weighted least squares regression model by minimizing for a new set of weights |
| Tuning parameter in local regression | The smaller the value of s, the more local and wiggly will be our fit; alternatively, a very large value of s will lead to a global fit to the data using all of the training observations. |
| Generalized Additive Models | Allow us to extend the methods above to deal with multiple predictors.  Provide a general framework for extending a standard linear model by allowing non-linear functions of each of the variables, while maintaining additivity |
| Pros and Cons of GAM | Pros   * GAMs allow us to fit a non-linear fj to each Xj , so that we can automatically model non-linear relationships that standard linear regression will miss (Don’t need to try out manual transformations for each var individually) * The non-linear fits can potentially make more accurate predictions for the response Y * We can still examine the effect of each Xj on Y individually while holding all of the other variables fixed (still good for inference) * The smoothness of the function fj for the variable Xj can be summarized via degrees of freedom   Cons   * The model is restricted to be additive, so interactions can be missed |
| **Tree-Based Methods** | |
| Process of building a tree | 1. We divide the predictor space—that is, the set of possible values for X1, X2,...,Xp—into J distinct and non-overlapping regions, R1, R2,...,RJ (Goal is to find R that minimize RSS) 2. For every observation that falls into the region Rj , we make the same prediction, which is simply the mean of the response values for the training observations in Rj |
| Recursive binary splitting | * Begins at the top of the tree (at which point all observations belong to a single region) and then successively splits the predictor space; each split is indicated via two new branches further down on the tree * We consider all predictors X1,...,Xp, and all possible values of the cutpoint s for each of the predictors, and then choose the predictor and cutpoint such that the resulting tree has the lowest RSS * Once the regions R1,...,RJ have been created, we predict the response for a given test observation using the mean of the training observations in the region to which that test observation belongs |
| Why do we need pruning? | * The process described above may produce good predictions on the training set, but is likely to overfit the data, leading to poor test set performance |
| Best way to prune | * Cost complexity pruning: Consider a sequence of trees indexed by a nonnegative tuning parameter α * Use K-fold cross-validation to choose α * Average the results for each value of α, and pick α to minimize the average error. * Return the subtree that corresponds to the chosen value of α |
| Difference between classification and regression tree | * With classification, RSS cannot be used as the criterion to make splits * When building a classification tree, either the Gini index or the entropy are typically used to evaluate the quality of a particular split, since these two approaches are more sensitive to node purity than is the classification error rate. * Classification error rate is preferable if prediction accuracy of the final pruned tree is the goal |
| Pros and Cons of Trees | Pros   * Trees are very easy to explain to people. * Some people believe that decision trees more closely mirror human decision-making than do the regression and classification approaches * Trees can be displayed graphically, and are easily interpreted even by a non-expert (especially if they are small). * Trees can easily handle qualitative predictors without the need to create dummy variables.   Cons   * Unfortunately, trees generally do not have the same level of predictive accuracy as some of the other regression and classification approaches * Additionally, trees can be very non-robust. In other words, a small change in the data can cause a large change in the final estimated tree (Can fix though) |
| Bagging and Bootstrapping | * Natural way to reduce the variance and hence increase the prediction accuracy of a statistical learning method is to take many training sets from the population, build a separate prediction model using each training set, and average the resulting predictions * Because we generally do not have access to multiple training sets, we can bootstrap, by taking repeated samples from the (single) training data set * We then train our method on the bth bootstrapped training set in order to get ˆf ∗b(x), and finally average all the predictions |
| How to apply bagging to decision trees | * We simply construct B regression trees using B bootstrapped training sets, and average the resulting predictions |
| Out-of-bag estimation error | We can predict the response for the ith observation using each of the trees in which that observation was OOB. |
| Random Forests | * Random forests provide an improvement over bagged trees by way of a random small tweak that decorrelates the trees * But when building these decision trees, each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors * The number of predictors considered at each split is approximately equal to the square root of the total number of predictors (vs. bagging which is just m = p) |
| Why RF is better than bagging | Averaging many highly correlated quantities does not lead to as large of a reduction in variance as averaging many uncorrelated quantities. In particular, this means that bagging will not lead to a substantial reduction in variance over a single tree in this setting |
| Boosting | * With trees, boosting works in a similar way, except that the trees are grown sequentially: each tree is grown using information from previously grown trees * Boosting does not involve bootstrap sampling; instead each tree is fit on a modified version of the original data set * We fit a tree using the current residuals, rather than the outcome Y , as the response * We then add this new decision tree into the fitted function in order to update the residuals |
| Relationship between number of trees and bias-variance trade off | More trees can mean less bias and more variance, so there is a risk of overfitting |
| **SVM** | |
| Maximal margin hyperplane (optimal separating hyperplane) | Separating hyperplane farthest from the training observations (i.e. perpendicular distance from each training observation to a given separating hyperplane  We hope that a classifier that has a large margin on the training data will also have a large margin on the test data, and hence will classify the test observations correctly |
| Support vectors | Vectors in p-dimensional spaces that support the maximal marginal hyperplane in the sense vector that if these points were moved slightly then the maximal margin hyperplane would move as well |
| Support vector classifier | * We can still find a separating hyperplane that does a "good" job, and just incurs some penalty for not perfectly separating the classes * Parameterized by a tunable "Cost" parameter |
| Support vector machine | The support vector machine (SVM) is an extension of the support vector support vector machine classifier that results from enlarging the feature space in a specific way, using kernels (kernel function that quantifies the similarity of two observations |
| What does the kernel do? | * Expands the feature space in order to accommodate non-linear class boundaries * Instead of manually creating new non-linear features with quadratics or log or cos, we take advantage of the "kernel trick" which implicitly compute these transformations and feature expansions for us |
| Linear kernel | Support vector classifier is linear in the features; the linear kernel essentially quantifies the similarity of a pair of observations using Pearson (standard) correlation |
| Polynomial kernel | Using such a kernel with d > 1, instead of the standard linear kernel kernel (9.21), in the support vector classifier algorithm leads to a much more flexible decision boundary |
| Radial Kernel | Radial kernel has very local behavior, in the sense that only nearby training observations have an effect on the class label of a test observation |
| Unsupervised Learning | |
| Principal Components Analysis | * When faced with a large set of correlated variables, principal components allow us to summarize this set with a smaller number of representative variables that collectively explain most of the variability in the original set * Purpose is to find a low-dimensional representation of the data that captures as much of the information as possible * PCA seeks a small number of dimensions that are as interesting as possible, where the concept of interesting is measured by the amount that the observations vary along each dimension |
| Loading vectors and projections | The principal component loading vectors as the directions in feature space along which the data vary the most, and the principal component scores as projections along these directions |
| Why does it matter that we scale the variables before PCA? | If we perform PCA on the unscaled variables, then the first principal component loading vector will have a very large loading for the biggest variable, which might only be big because of the unit |
| K-means clustering | * To perform K-means clustering, we must first specify the desired number of clusters K; then the K-means algorithm will assign each observation to exactly one of the K clusters * The idea behind K-means clustering is that a good clustering is one for which the within-cluster variation is as small as possible |
| Squared Euclidean Distance | The within-cluster variation for the kth cluster is the sum of all of the pairwise squared Euclidean distances between the observations in the kth cluster, divided by the total number of observations in the kth cluster |
| K-means clustering algorithm | 1. Randomly assign a number, from 1 to K, to each of the observations.  2. Iterate until the cluster assignments stop changing:  (a) For each of the K clusters, compute the cluster centroid.  (b) Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance). |
| Hierarchical Clustering | Hierarchical clustering is an alternative approach which does not require that we commit to a particular choice of K (main disadvantage of k-means) |
| How many possible renderings of the hierarchical clustering dendogram exist? | * 2^n-1 where n is the number of leaves * This is because at each of the n − 1 points where fusions occur, the positions of the two fused branches could be swapped without affecting the meaning of the dendrogram. |
| Hierarchical clustering algorithm | * We begin by defining some sort of dissimilarity measure between each pair of observations * The two clusters that are most similar to each other are then fused so that there now are n−1 clusters. * Next the two clusters that are most similar to each other are fused again, so that there now are n − 2 * The algorithm proceeds in this fashion until all of the observations belong to one single cluster, and the dendrogram is complete |
| Complete linkage | Maximal intercluster dissimilarity (largest pairwise similarities) |
| Single linkage | * Minimal intercluster dissimilarity (smallest pairwise similarities) * clusters formed via single linkage clustering may be forced together due to Chaining (pro and con): single elements being close to each other, even though many of the elements in each cluster may be very distant to each other |
| Average linkage | Mean intercluster dissimilarity (mean pairwise similarities) |
| Centroid linkage | Dissimilarities between cluster centroids |
| Practical issues with clustering | K-means   * Should observations or features first be standardized? * How many clusters?   Hierarchical   * What type of dissimilarity and linkage should be used? * Where should we cut the dendrogram? |
| Latent Dirichlet Allocation |  |
| Dbscan clustering | * Uses the assumption that within a cluster density of datapoints should be high and fall off between clusters * DBSCAN handles the half moon by saying that what really matters is that there is a continuous region of high density data points * Epsilon: What is the usual distance between things in the same cluster (“typical” distance between points) |