# Linear Regression

## Simple Linear Regression (P61):

It is a very straightforward simple linear approach for predicting a quantitative response Y on the basis of a single predictor variable X. It assumes that there is approximately a linear relationship between X and Y . Mathematically, we can write this linear relationship as:

Y = β0+β1X

β0 and β1 are known as model coefficients or parameters.

Let ˆyi = ˆ β0 + ˆ β1xi be the prediction for Y based on the ith value of X. Then ei = yi−ˆyi represents the ith residual—this is the difference between the ith observed response value and the ith response value that is predicted by our linear model. We define the residual sum of squares (RSS) as:

RSS = e1^2 + e2^2 + … + en^2

equivalently

RSS = (y1− ˆβ0− ˆβ1x1)^2+(y2− ˆβ0− ˆβ1x2)^2+. . .+(yn− ˆβ0− ˆβ1xn)^2

Least squares: <https://www.datalearner.com/blog/1051539222770649>

如何找β0和β1:

The least squares approach chooses ˆ β0 and ˆ β1 to minimize the RSS. Using some calculus, one can show that the minimizers are: (P62) where y- and x- are the sample means, in other words, this defines the least squares coefficient estimates for simple linear regression.

The true relationship between X and Y takes the form Y = f(X) + for some unknown function f, where epsilon(E) is a mean-zero random error term. If f is to be approximated by a linear

function, then we can write this relationship as:

Y = β0 + β1X + E

The true relationship is generally not known for real data, but the least squares line can always be computed using the coefficient estimates given in (3.4). In other words, in real applications, we have access to a set of observations from which we can compute the least squares line; however, the population regression line is unobserved. 真实的population regression无法被观测，只有estimate出来的least squares line。(P65)

standard error解释: <https://www.zhihu.com/question/264813135/answer/286455446>

Residual Standard Error: is standard deviation of epsilon(E)

RSE = √[(1/(n-2)) RSS]

RSE RSS R2区别: <https://www.cnblogs.com/HuZihu/p/9692814.html>

R2 Statistic: provides an alternative measure of fit. It takes the form of a proportion - the proportion of variance explained, and so it always takes on a value between 0 and 1, and is independent of the scale of Y. The R2 statistic is a measure of the linear relationship between X and Y.

R^2 = (TSS - RSS)/TSS = 1 - (RSS/TSS)

TSS: sum(yi - y`)^2, is the Total Sum of Squares. TSS measures the total variance in the response Y. y` = (1/n) sum(yi), TSS就是每个yi减去n个y的平均值的平方的总和

RSS: sum(yi - yi^)^2. y^ = β0 + β1X, RSS就是每个yi减去预测yi值的平方的总和

An R2 statistic that is close to 1 indicates that a large proportion of the variability in the response has been explained by the regression. A number near 0 indicates that the regression did not explain much of the variability in the response.

RSE 描述模式适合度，预测值和真实值差距不大RSE就小，反之则大

R2 描述可用X来解析Y的比例，接近1则表示绝大比例的数据都可以用预测解释，接近0表示大比例数据无法解析，有可能是数据完全不适合linear，或者inherent error(variance)太大

Standard deviation 和 variance: <https://www.investopedia.com/ask/answers/021215/what-difference-between-standard-deviation-and-variance.asp>

<https://www.shuxuele.com/data/standard-deviation.html>

## Multiple Linear Regression(P71)

The multiple linear regression model takes the form:

Y = β0 + β1X1 + β2X2 + · · · + βpXp + epsilon

RSS will be:

RSS = sum(yi - y`i)^2

RSS = sum(yi - β0 - β1Xi1 - β2Xi2 + · · · + βpXip)^2

Response and Predictors:

<https://support.minitab.com/en-us/minitab/18/help-and-how-to/modeling-statistics/regression/supporting-topics/basics/what-are-response-and-predictor-variables/>

Important Questions(P75):

**One: Is there a relationship between the response and predictors?**

In order to determine whether there is a relationship between the response and the predictor we can simply check whether β1 = 0. In the multiple regression setting with p predictors, we need to ask whether all of the regression coefficients are zero,

i.e. whether β1 = β2 = · · · = βp = 0.

H0 : β1 = β2 = · · · = βp = 0

Ha : at least one βj is non-zero.

Provided H0 is true,

E{(TSS − RSS)/p} = Variance σ^2

If Ha is true,

E{(TSS − RSS)/p} > Variance σ^2

F-statistic is:

F = [(TSS - RSS)/p] / [RSS/(n-p-1)]

When n is large, an F-statistic that is just a little larger than 1 might still provide evidence against H0. In contrast, a larger F-statistic is needed to reject H0 if n is small. When H0 is true and the errors have a normal distribution, the F-statistic follows an F-distribution.

**Two: Deciding on Important Variables**

If p = 2, then we can consider four models: (1) a model containing no variables, (2) a model containing X1 only, (3) a model containing X2 only, and (4) a model containing both X1 and X2. We can then select the best model out of all of the models that we have considered. How do we determine which model is best? Various statistics can be used to judge the quality of a model. These include Mallow’s Cp, Akaike informa-Mallow’s Cp tion criterion (AIC), Bayesian information criterion (BIC), and adjusted R2. But if p = 30, then we must consider 230 = 1,073,741,824 models! This is not practical. Therefore, unless p is very small, we cannot consider all 2p models.

Forward selection: We begin with the null model—a model that contains an intercept but no predictors. We then fit p simple linear regressions and add to the null model the variable that results in the lowest RSS. We then add to that model the variable that results in the lowest RSS for the new two-variable model. This approach is continued until some stopping rule is satisfied.

Backward selection. We start with all variables in the model, and remove the variable with the largest p-value—that is, the variable that is the least statistically significant. The new (p − 1)-variable model is fit, and the variable with the largest p-value is removed. This procedure continues until a stopping rule is reached. For instance, we may stop when all remaining variables have a p-value below some threshold. (Backward selection cannot be used if p > n, while forward selection can always be used.)

Mixed selection. This is a combination of forward and backward selection. We start with no variables in the model, and as with forward selection, we add the variable that provides the best fit. We continue to add variables one-by-one. Of course, as we noted with the Advertising example, the p-values for variables can become larger as new predictors are added to the model. Hence, if at any point the p-value for one of the variables in the model rises above a certain threshold, then we remove that variable from the model. We continue to perform these forward and backward steps until all variables in the model have a sufficiently low p-value, and all variables outside the model would have a large p-value if added to the model.

**Three: Model Fit**

R2 value close to 1 indicates that the model explains a large portion of the variance in the response variable. R2 will always increase when more variables are added to the model even if those variables are only weakly associated with the response.

RSE: 和R2一起看判断模型是否合适

RSE = √[(1/(n-p-1)) RSS]

**Four: Predictions**

1. The coefficient estimates ˆβ0, ˆβ1, …, ˆβp are estimates for β0, β1, …, βp, that is, the least squares plane

ˆY = ˆβ0 + ˆβ1X1 + … + ˆβpXp

is only an estimate for the true population regression plane

f(x) = β0 + β1X1 + … + βpXp

1. Of course, in practice assuming a linear model for f(X) is almost always an approximation of reality, so there is an additional source of potentially reducible error which we call model bias. So when we use a linear model, we are in fact estimating the best linear approximation to the true surface. However, here we will ignore this discrepancy, and operate as if the linear model were correct.
2. Even if we knew f(x) cannot be predicted perfectly because of the random error epsilon in the model. We use prediction intervals to measure how much will Y very from ˆY.

## Other Considerations in the Regression Model(P82)

Qualitative Predictors

*Quantitative* data are data about numeric variables (e.g. how many; how much; or how often). *Qualitative* data are measures of 'types' and may be represented by a name, symbol, or a number code.

Support qualitative predictor only has two levels, for example “gender” variable, we can create a new variable that takes the form

xi = 1 if ith person is female, 0 if ith person is male

use this variable as a predictor in the regression equation:

yi = β0 + β1xi + epsilon

= β0 + β1 + epsilon, if ith person is female

= β0 + epsilon, if ith person is male

Instead of a 0/1 coding scheme, we could create a dummy variable：

xi = 1 if ith person is female, -1 if ith person is male

use this variable as a predictor in the regression equation:

yi = β0 + β1xi + epsilon

= β0 + β1 + epsilon, if ith person is female

= β0 - β1 + epsilon, if ith person is male

When a qualitative predictor has more than two levels, we can create additional dummy variables. For example, “ethnicity” variable.

xi1 = 1 if ith person is Asian, 0 if ith person is not Asian

xi2 = 1 if ith person is Caucasian, 0 if ith person is not Caucasian

use two variables as a predictor in the regression equation:

yi = β0 + β1xi1 + β2xi2 + epsilon

= β0 + β1 + epsilon, if ith person is Asian

= β0 + β2 + epsilon, if ith person is Caucasian

= β0 + epsilon, if ith person is African American

There will always be one fewer dummy variable than the number of levels.

There are many different ways of coding qualitative variables besides the dummy variable approach taken here. All of these approaches lead to equivalent model fits, but the coefficients are different and have different interpretations, and are designed to measure particular contrasts.

Extensions of the Linear Model

Additive assumption: the effect of changes in a predictor X1 on the response Y is independent of the values of the other predictors. Meaning the change in the response Y due to a one-unit change in X1 is constant, regardless of the value of X1.

Removing the Additive Assumption:

Include a third predictor, called an interaction term, which is constructed by computing the product of X1 and X2.

Non-linear Relationship:

Extending the linear model to accommodate non-linear relationships is known as *polynomial regression*, which includes polynomial functions of the predictors in the regression model. (e.g. X1 = horsepower and X2 = horsepower^2)

Potential Problems

**Non-linearity of the Data**

Residual plots are a useful graphical tool for identifying non-linearity. If the residual plot indicates that there are non-linear associations in the data, then a simple approach is to use non-linear transformations of the predictors such as logX, X^2 in the regression model.

**Correlation of Error Terms**

For instance, if the errors are uncorrelated, then the fact that Epsilon i is positive provides little or no information about the sign of Epsilon i+1. The standard errors that are computed for the estimated regression coefficients or the fitted values are based on the assumption of uncorrelated error terms. If in fact there is correlation among the error terms, then the estimated standard errors will tend to underestimate the true standard errors. As a result, confidence and prediction intervals will be narrower than they should be.

For instance, consider a study in which individuals’ heights are predicted from their weights. The assumption of uncorrelated errors could be violated if some of the individuals in the study are members of the same family, or eat the same diet, or have been exposed to the same environmental factors. In general, the assumption of uncorrelated errors is extremely important for linear regression as well as for other statistical methods, and good experimental design is crucial in order to mitigate the risk of such correlations.

**Non-constant Variance of Error Terms**

Another important assumption of the linear regression model is that the error terms have a constant variance, Var(Ei) = σ2. The standard errors, confidence intervals, and hypothesis tests associated with the linear model rely upon this assumption.

One can identify non-constant variances in the errors, or heteroscedasticity, from the presence of a funnel shape in the residual plot.

One solution is to transform the response Y using a concave function such as logY or √Y.

**Outliers**

An outlier is a point for which yi is far from the value predicted by the model. If we believe that an outlier has occurred due to an error in data collection or recording, then one solution is to simply remove the observation.

**High Leverage Points**

Observations with high leverage have an unusual value for xi. In order to quantify an observation’s leverage, we compute the *leverage statisti*c. A large value of this statistic indicates an observation with high leverage. For a simple linear regression,

hi = (1/n) + (xi - x`)^2 / sum(xi[from 1 to n] - x`)^2

x` = (1/n) sum(xi)

The leverage statistic hi is always between 1/n and 1, and the average leverage for all the observations is always equal to (p+1)/n. So if a given observation has a leverage statistic that greatly exceeds (p+1)/n, then we may suspect that the corresponding point has high leverage.

**Collinearity**

Collinearity refers to the situation in which two or more predictor variables are closely related to one another. Collinearity reduces the accuracy of the estimates of the regression coefficients, it causes the standard error for βj^ to grow.

The VIF for each variable can be computed using the formula:

VIF(βj^) = 1/(1-R^2xj|x-j)

where R^2xj|x-j is the R^2 from a regression of Xj onto all of the other predictors. If it is close to one, then collinearity is present, and so the VIF will be large.

When faced with the problem of collinearity, there are two simple solutions. The first is to drop one of the problematic variables from the regression. The second solution is to combine the collinear variables together into a single predictor.

## The Marketing Plan(P102)

1. Is there a relationship between advertising sales and budget?

This question can be answered by fitting a multiple regression model of *sales* onto *TV, radio, and newspaper*, as in (3.20), and testing the hypothesis:

H0 : βTV = βradio = βnewspaper = 0

In Section 3.2.2, we showed that the *F-statistic* can be used to determine whether or not we should reject this null hypothesis. In this case the p-value corresponding to the F-statistic in Table 3.6 is very low, indicating clear evidence of a relationship between advertising and sales.

1. How strong is the relationship?

We discussed two measures of model accuracy. First, the RSE estimates the standard deviation of the response from the population regression line. Second, the R^2 statistic records the percentage of variability in the response that is explained by the predictors.

1. Which media contribute to sales?

We can examine the p-values associated with each predictor’s t-statistic. In Table 3.4, the p-values for TV and radio are low, but the p-value for newspaper is not. This suggests that only TV and radio are related to sales.

1. How large is the effect of each medium on sales?

We saw in Section 3.1.2 that the standard error of βj^ can be used to construct confidence intervals for βj.

1. How accurately can we predict future sales?

The response can be predicted using (3.21). The accuracy associated with this estimate depends on whether we wish to predict an individual response, Y = f(X) + e, or the average response, f(X). If the former, we use a prediction interval, and if the latter, we use a confidence interval. Prediction intervals will always be wider than confidence intervals because they account for the uncertainty associated with epsilon, the irreducible error.

1. Is the relationship linear?

In Section 3.3.3, we saw that residual plots can be used in order to identify non-linearity. If the relationships are linear, then the residual plots should display no pattern.

1. Is there synergy among the advertising media?

The standard linear regression model assumes an additive relationship between the predictors and the response. An additive model is easy to interpret because the effect of each predictor on the response is unrelated to the values of the other predictors. However, the additive assumption may be unrealistic for certain data sets. In Section 3.3.2, we showed how to include an interaction term in the regression model in order to accommodate non-additive relationships. A small p-value associated with the interaction term indicates the presence of such relationships.

## Comparison of Linear Regression with K-Nearest Neighbors(P104)

Linear regression is an example of a *parametric* approach. Parametric methods have advantages: easy to fit, coefficients have simple interpretations, tests of statistical significance can be easily performed. But parametric methods do have a disadvantage: by construction, they make strong assumptions about the form of f(x), if the specified functional form is far from the truth, and prediction accuracy is our goal, then the parametric method will perform poorly. In contrast, *non-parametric* methods do not explicitly assume a parametric form for f(X), and thereby provide an alternative and more flexible approach for performing regression.

K-nearest neighbors regression (KNN regression) is one of the simplest and best-known non-parametric methods.

Given a value for K and a prediction point x0, KNN regression first identifies the K training observations that are closest to x0, represented by N0. It then estimates f(x0) using the average of all the training responses in N0. In other words,

f(x0) = 1/K (sum[xi belongs to N0] yi)

<https://www.youtube.com/watch?v=HVXime0nQeI>

In general, the optimal value for K will depend on the bias-variance tradeoff. A small value for K provides the most flexible fit, which will have low bias but high variance.

The parametric approach will outperform the nonparametric approach if the parametric form that has been selected is close to the true form of f.

P=1 (only one predictor):

In linear relationship, linear regression is superior to that of KNN for low values of K, but for K >= 4, KNN out-performs linear regression, for KNN regression, best results occur with a very large value of K. KNN performs slightly worse than linear regression when the relationship is linear, but much better than linear regression for non-linear situations.

p>1 (higher dimensions):

KNN often performs worse than linear regression. However, spreading 100 observations over p = 20 dimensions results in a phenomenon in which a given observation has no

nearby neighbors—this is the so-called curse of dimensionality. That is, the K observations that are nearest to a given test observation x0 may be very far away from x0 in p-dimensional space when p is large, leading to a very poor prediction of f(x0) and hence a poor KNN fit. As a general rule, parametric methods will tend to outperform non-parametric approaches when there is a small number of observations per predictor.

Even in problems in which the dimension is small, we might prefer linear regression to KNN from an interpretability standpoint. If the test MSE of KNN is only slightly lower than that of linear regression, we might be willing to forego a little bit of prediction accuracy for the sake of a simple model that can be described in terms of just a few coefficients, and for which p-values are available.

KNN和K-means区别：<https://zhuanlan.zhihu.com/p/31580379>

# Classification

## An Overview of Classification(P128):

## Why Not Linear Regression?(P129):

Linear regression is not appropriate in the case of a qualitative response. Because in general there is no natural way to convert a qualitative response variable with more than two

levels into a quantitative response that is ready for linear regression.

## Logistic Regression(P130):

Logistic regression models the probability that Y belongs to a particular category.

The Logistic Model

Linear regression model to represent probabilities:

p(X) = β0 + β1X

But it has problems e.g. to get values bigger than 1.

To avoid this problem we must model p(X) using a function that gives outputs between 0 and 1 for all values of X, we use Logistic Function:

p(X) = [e^(β0+β1X)] / [1+e^(β0+β1X)]

After a bit of manipulation, we find that:

p(X) / (1-p(X)) = e^(β0+β1X)

p(X) / (1-p(X)) is called *odds*, can take on any value between 0 and inf

By taking the logarithm of both sides, we arrive at:

log (p(X) / (1-p(X))) = β0+β1X

log (p(X) / (1-p(X))) is called *log-odds* or *logit*.

Estimating the Regression Coefficients

We try to find β0ˆ and β1ˆ such that plugging these estimates into the model for p(X), yields a number close to one for all individuals who defaulted, and a number close to zero for all individuals who did not.

Likelihood function:

l(β0, β1) = ∏[i:yi=1] p(xi) ∏[j:yj=0] (1-p(xj))

The estimates β0ˆ and β1ˆ are chosen to maximize this likelihood function.

Making Predictions

The probability for an individual with a balance of $1000 is:

p(X) = [e^(β0+β1X)] / [1+e^(β0+β1X)] = e^(-10.6513+0.0055x1000) / (1+e^(-10.6513+0.0055x1000)) = 0.00576

The probability for student:

Student[yes] = 1, student[no] = 0

Pr(default=Yes | student=Yes) = e^(-3.5041+0.4049x1) / (1+e^(-3.5041+0.4049x1)) = 0.0431

Pr(default=Yes | student=No) = e^(-3.5041+0.4049x0) / (1+e^(-3.5041+0.4049x0)) = 0.0292

Multiple Logistic Regression  
log(p(x) / (1-p(x))) = β0 + β1X1 + β2X2 + …. + βpXp

rewritten as:

p(x) = e^(β0 + β1X1 + … + βpXp) / (1+e^(β0 + β1X1 + … + βpXp))

Logistic Regression for > 2 Response Classes

Discriminant analysis is popular for multiple-class classification.

## Linear Discriminant Analysis(P137)

Using Bayes’ Theorem for Classification

Let fk(X) = Pr(X=x | Y=k), then Bayes’s theorem states that:

Pr(Y = k | X = x) = πk fk(x) / (sum(l=1 to k) πl fl(x))

Linear Discriminant Analysis for p=1

fk(x) = (1 / √2πσk) exp (- (1/2σk^2) (x - μk)^2)

where muk and σk^2 are the mean and variance parameters for the kth class.

Assume σ1^2 = σ2^2 = … = σk^2, we find that:

pk(x) = Pr(Y = k | X = x) = πk (1 / √2πσk) exp (- (1/2σk^2) (x - μk)^2) / (sum(l=1 to k) πl ((1 / √2πσl) exp (- (1/2σl^2) (x - μl)^2)))

!! Note: πk denotes the prior probability that an observation belongs to the kth class, not 3.1415926 !!

^μk = 1/nk sum[i:yi = k] xi

^σ^2 = (1/(n-K)) sum[k=1 to K] sum[i:yi = k] (xi - ^μk)^2

where n is the total number of training observations, and nk is the number of training observations in the kth class. Then:

^πk = nk / n

Discriminant function δk(x) is:

δk(x) = x \* (μk / σ^2) - (μk^2 / 2σ^2) + log(πk)

Then the estimate ^δk(x) is:

^δk(x) = x \* (^μk / ^σ^2) - (^μk^2 / 2^σ^2) + log(^πk)

Linear Discriminant Analysis for p>1

Assume X = (X1, X2,..., Xp) is drawn from a multivariate Gaussian distribution, with a class-specific mean vector and a common covariance matrix.

To indicate that a p-dimensional random variable X has a multivariate Gaussian distribution, we write X ~ N (μ, Σ), here E(X) = μ is the mean of X and Cov(X) = Σ is the p x p covariance matrix of X. The multivariate Gaussian density function:

f(x) = [1 / ((2π)^(p/2) |Σ|^(1/2))] exp (-½ (x-μ)^T Σ^-1 (x-μ))

Plugging the density function for the kth class, fk(X = x) into Bayes’s theorem states that:

δk(x) = x^T Σ^-1 μk - ½ μk^T Σ^-1 μk + log πk

How can we decide which threshold value is best?

The ROC curve is a popular graphic for simultaneously displaying the two types of errors for all possible thresholds. The name ROC is historic, and comes from communications theory. It is an acronym for receiver operating characteristics. The overall performance of a classifier, summarized over all possible thresholds, is given by the area under the ROC curve (AUC). An ideal ROC curve will hug the top left corner, so the larger the AUC the better the classifier.

ROC拓展：<https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=zh_cn>

通过ROC确定Best Threshold：<https://blog.csdn.net/qq_39917365/article/details/108273866>

Quadratic Discriminant Analysis

Like LDA, the QDA classifier results from assuming that the observations from each class are drawn from a Gaussian distribution, and plugging estimates for the parameters into Bayes’ theorem in order to perform prediction. However, unlike LDA, QDA assumes that each class has its own covariance matrix. That is, it assumes that an observation from the kth class is of the form X ~ N(μk,Σk), where Σk is a covariance matrix for the kth class.

The Bayes classifier assigns an observation X = x to the class for which the following is largest.

δk(x) = -½ (x - μk)^T Σk^-1 (x - μk) - ½ log|Σk| + logπk

= -½ x^T Σk^-1 x + x^T Σk^-1 \* μk - ½ μk^T Σk^-1 μk -½ log|Σk| + logπk

When there are p predictors, then estimating a covariance matrix requires estimating p(p+1)/2 parameters. QDA estimates a separate covariance matrix for each class, for a total of Kp(p+1) / 2 parameters. Roughly speaking, LDA tends to be a better bet than QDA if there are relatively few training observations and so reducing variance is crucial. In contrast, QDA is recommended if the training set is very large, so that the variance of the classifier is not a major concern, or if the assumption of a common covariance matrix for the K classes is clearly untenable.

## A Comparison of Classification Methods(P151)

LDA vs Logistic Regression:

In the LDA framework, we can see that the log odds is given by

log (p1(x) / (1-p1(x))) = log (p1(x) / p2(x)) = c0 + c1x

where c0 and c1 are functions of μ1, μ2, and σ^2. We know that in logistic regression

log (p1 / (1-p1)) = β0 + β1x

Hence, both logistic regression and LDA produce linear decision boundaries. The only difference between the two approaches lies in the fact that β0 and β1 are estimated using maximum likelihood, whereas c0 and c1 are computed using the estimated mean and variance from a normal distribution.

LDA assumes that the observations are drawn from a Gaussian distribution with a common covariance matrix in each class, and so can provide some improvements over logistic regression when this assumption approximately holds. Conversely, logistic regression can outperform LDA if these Gaussian assumptions are not met.

KNN:

KNN takes a completely different approach from the classifiers seen in this chapter. Therefore, we can expect this approach to dominate LDA and logistic regression when the decision boundary is highly non-linear. On the other hand, KNN does not tell us which predictors are important.

QDA:

QDA serves as a compromise between the non-parametric KNN method and the linear LDA and logistic regression approaches. Since QDA assumes a quadratic decision boundary, it can accurately model a wider range of problems than can the linear methods. Though not as flexible as KNN, QDA can perform better in the presence of a limited number of training observations because it does make some assumptions about the form of the decision boundary.

4 Methods Scenarios:

See P153

Variance Bias解释：<https://www.cnblogs.com/sddai/p/9240293.html>

方差偏差更深入解释：<https://zhuanlan.zhihu.com/p/38853908>

KNN: Take an extreme example: I can model you as equalling your twin brother or a person that is the most similar to you in the whole world (k=1). This is highly flexible (low bias), but relying on a single data point is very risky (high variance). Or I can model you as an average (in regression) or mode (in classification) of all the people on the planet (k=N). This is highly inflexible (high bias) but very robust (low variance).

Conclusion:

When the true decision boundaries are linear, then the LDA and logistic regression approaches will tend to perform well.When the boundaries are moderately non-linear, QDA may give better results. Finally, for much more complicated decision boundaries, a non-parametric approach such as KNN can be superior. But the level of smoothness for a non-parametric approach must be chosen carefully.

# Resampling Methods

## Cross-Validation(P176)

The Validation Set Approach

It involves randomly dividing the available set of observations into two parts, a *training set* and a *validation set* or *hold-out set*, and the fitted model is used to predict the responses for the observations in the validation set. The resulting validation set error rate – typically assessed using MSE in the case of a quantitative response – provides an estimate of the test error rate.

Leave-One-Out Cross-Validation

Like the validation set approach, LOOCV involves splitting the set of observations into two parts. However, instead of creating two subsets of comparable size, a single observation (x1, y1) is used for the validation set, and the remaining observations {(x2, y2), …, (xn, yn)} make up the training set. Such that MSE1 = (y1 - ^y1)^2, and repeat the procedure by selecting (x2, y2) for the validation data, repeating this approach n times produces n squared errors: MSE1, MSE2, …, MSEn. The LOOCV estimate for the test MSE is the average of these n test error estimates:

CV(n) = 1/n sum(i=1 to n) MSEi

LOOCV advantage:

First, it has far less bias. Second, in contrast to the validation approach which will yield different results when applied repeatedly due to randomness in the training/validation set splits, performing LOOCV multiple times will always yield the same results: there is no randomness in the training/validation set splits.

With least squares linear or polynomial regression, an amazing shortcut makes the cost of LOOCV the same as that of a single model fit! The following formula holds:

CV(n) = 1/n sum(i=1 to n) [(yi - ^yi) / (1 - hi)^2]

k-Fold Cross-Validation

An alternative to LOOCV is k-fold CV. This approach involves randomly dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining k-1 folds. Then this procedure is repeated k times, each time a different group of observations is treated as a validation set. The k-fold CV estimate is computed by averaging there values:

CV(k) = 1/k sum(i=1 to k) MSEi

Bias-Variance Trade-Off for k-fold Cross-Validation

LOOCV has higher variance than does k-fold CV with k<n. Typically, given these considerations, one performs k-fold cross-validation using k = 5 or k = 10, as these values have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance.

Cross-Validation on Classification Problems

For instance, in the classification setting, the LOOCV error rate takes the form:

CV(n) = 1/n sum(i=1 to n) Erri

where Erri = I(yi != ^yi).

## The Bootstrap(P187)

1. Simulated 100 pairs of returns for X and Y.
2. Estimates for alpha.
3. Recycle the procedure 1000 times.
4. Calculated estimates alpha = 1/1000 (all alphas)

详解：<https://zhuanlan.zhihu.com/p/24851814>

# Linear Model Selection and Regularization

## Subset Selection(P205)

Best Subset Selection

Best subset selection algorithm:

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:
3. Fit all (p select k) models that contain exactly k predictors
4. Pick the best among these (p select k) models, and call it Mk. Here best is defined as having the smallest RSS, or equivalently largest R^2.
5. Select a single best model from among M0, …, Mp using cross-validated prediction error, Cp (AIC), BIC, or adjusted R^2.

The BSS algorithm is simple but computationally infeasible for large p values.

Stepwise Selection

Forward Stepwise Selection algorithm:

1. Let M0 denote the null model, which contains no predictors.
2. For k = 0, …, p-1:
3. Consider all p - k models that augment the predictors in Mk with one additional predictor.
4. Choose the best among these p - k models, and call it Mk+1. Here best is defined as having the smallest RSS or highest R^2.
5. Select a single best model from among M0, …, Mp using cross-validated prediction error, Cp (AIC), BIC, or adjusted R^2.

Forward stepwise’s computational advantage over best subset selection is clear.

Backward Stepwise Selection algorithm:

1. Let Mp denote the full model, which contains all p predictors.
2. For k = p, p-1, …, 1:
3. Consider all k models that contain all best one of the predictors in Mk, for a total of k - 1 predictors.
4. Choose the best among these k models, and call it Mk-1. Here best is defined as having the smallest RSS or highest R^2.
5. Select a single best model from among M0, …, Mp using cross-validated prediction error, Cp (AIC), BIC, or adjusted R^2.

Backward selection requires that the number of samples n is larger than the number of variables p. In contrast, forward stepwise can be used even when n < p, and so is the only viable subset method when p is very large.

Forward and Backward Stepwise详解：<https://quantifyinghealth.com/stepwise-selection/>

Choosing the Optimal Model

For a fitted least squares model containing d predictors, the Cp estimate of test MSE is computed using the equation:

Cp = 1/n (RSS + 2dˆσ^2)

The AIC criterion is defined for a large class of models fit by maximum likelihood:

AIC = 1/(nˆσ^2) (RSS + 2dˆσ^2)

BIC is derived from a Bayesian point of view, but ends up looking similar to Cp (and AIC) as well:

BIC = 1/(nˆσ^2) (RSS + log(n)dˆσ^2)

The adjusted R^2 statistic is another popular approach for selecting among a set of models that contain different numbers of variables:

Adjusted R^2 = 1 - (RSS/(n-d-1)) / (TSS/(n-1))

Unlike Cp, AIC, BIC, for which a small value indicates a model with a low test error, a large value of adjusted R^2 indicates a model with a small test error.

We can also use Cross-Validation to determine the best model, in the past performing cross-validation was computationally prohibitive for many problems with large p, so AIC BIC Cp and adjusted R^2 were more attractive approaches for choosing among a set of models. But nowadays with fast computers, Cross-Validation is the best way.

## Shrinkage Methods(P214)

Shrinking the coefficient estimates can significantly reduce their variance. The two best-known techniques for shrinking the regression coefficients towards zeros are ridge regression and the lasso.

Ridge Regression

Ridge regression is very similar to least squares, except that the coefficients are estimated by minimizing a slightly different quantity, the ridge regression coefficient estimates ˆβ^R are the values that minimize:

sum(i=1 to n) (yi - β0 - sum(j=1 to p)βj xij)^2 + λ sum(j=1 to p)βj^2

= RSS + λ sum(j=1 to p)βj^2

where λ >= 0 is a tuning parameter, to be determined separately.

Ridge Regression详解：<https://www.youtube.com/watch?v=Q81RR3yKn30>

Ridge regression’s advantage over least squares is rooted in the bias-variance trade-off. As λ increases, the flexibility of the ridge regression fit decreases, leading to decreased variance but increased bias.

The Lasso

Ridge regression does have one obvious disadvantage, unlike best subset, forward/backward stepwise selection, which will generally select models that involve just a subset of the variables, ridge regression will include all p predictors in the final model, it can create a challenge in model interpretation in settings in which the number of variables p is quite large.

The *lasso* is a relatively recent alternative to ridge regression that overcomes this disadvantage. The lasso coefficients, ˆβλL, minimize the quantity:

sum(i=1 to n) (yi - β0 - sum(j=1 to p) (βj xij)^2 + λ sum(j=1 to p) |βj|

= RSS + λ sum(j=1 to p) |βj|

对比Ridge Regression，Lasso使用L1 norm |βj|，会使得一些coefficient estimates强制归0达到variable selection的效果。

为什么Lasso能把coefficient estimates压缩为0：<https://zhuanlan.zhihu.com/p/116869931>

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.

Consider a simple special case with n=p, and X a diagonal matrix with 1’s on the diagonal and 0’s in all off-diagonal elements, also that we are performing regression without an intercept. In this case, the least squares solution is given by

ˆβj = yj

And in this setting, ridge regression amounts to finding β1, …, βj such that

sum(j=1 to p) (yj - βj)^2 + λ sum(j=1 to p) βj^2

is minimized, and the lasso amounts to finding the coefficients such that

sum(j=1 to p) (yj - βj)^2 + λ sum(j=1 to p) |βj|

is minimized. Once we can show that in this setting, the ridge regression estimates take the form

ˆβRj = yj / (1+λ)

and the lasso estimates take the form

ˆβLj = yj - λ/2 if yj > λ/2;

= yj + λ/2 if yj < -λ/2;

= 0 if |yj| <= λ/2.

The main ideas still hold approximately: ridge regression more or less shrinks every dimension of the data by the same proportion, whereas the lasso more or less shrinks all coefficients toward zero by a similar amount, and sufficiently small coefficients are shrunken all the way to zero.

From a Bayesian viewpoint, ridge regression and the lasso follow directly from assuming the usual linear model with normal errors, together with a simple prior distribution for β.

Selecting the Tuning Parameter

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. Finally, the model is re-fit using all of the available observations and the selected value of the tuning parameter.

## Dimension Reduction Methods(P228)

Let Z1, Z2, …, Zm represent M < p linear combinations of our original p predictors. That is:

Zm = sum(j=1 to p) φjm Xj

for some constants φ1m, φ2m, …, φpm, m = 1, …, M. We can then fit the linear regression model:

yi = θ0 + sum(m=1 to M) φm zim + epsolion i, i = 1, …, n

The term dimension reduction comes from the fact that this approach reduces the problem of estimating the p+1 coefficients β0, β1, . . . , βp to the simpler problem of estimating the M + 1 coefficients θ0, θ1, . . . , θM, where M < p. In other words, the dimension of the problem has been reduced from p + 1 to M + 1.

All dimension reduction methods work in two steps. First, the transformed predictors Z1, Z2, …, Zm are obtained. Second, the model is fit using these M predictors.

Principal Components Regression

PCA is a technique for reducing the dimension of a n x p data matrix X. The *first principal component* direction of the data is that along which the observations *vary the most*. In general, one can construct up to p distinct principal components. The second principal component Z2 is a linear combination of the variables that is uncorrelated with Z1, and has largest variance subject to this constraint. It turns out that the zero correlation condition of Z1 with Z2 is equivalent to the condition that the direction must be perpendicular, or orthogonal, to the first principal component direction. PC1是数据的最佳拟合线，PC2是穿过数据平均点做PC1垂线，两条PC就能组成Model Plane，接着让数据投射在plane上达到降维。

With two-dimensional data, such as in our advertising example, we can construct at most two principal components. However, if we had other predictors, such as population age, income level, education, and so forth, then additional components could be constructed. They would successively maximize variance, subject to the constraint of being uncorrelated with the preceding components. 处理2维数据只能生成两条principal components，但如果维度更多则会有PC3。

PCA详解：<https://zhuanlan.zhihu.com/p/77151308>

PCA通俗版：<https://blog.csdn.net/HLBoy_happy/article/details/77146012>

PCA详解2：<https://www.sartorius.com/en/knowledge/science-snippets/what-is-principal-component-analysis-pca-and-how-it-is-used-507186>

PCR approach involves constructing 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.

Partial Least Squares

The PCR approach that we just described involves identifying linear combinations, or directions, that best represent the predictors X1, . . .,Xp. These directions are identified in an unsupervised way, since the response Y is not used to help determine the principal component directions. That is, the response does not supervise the identification of the principal components.

We now present partial least squares (PLS), a supervised alternative to PCR. Like PCR, PLS is a dimension reduction method, which first identifies a new set of features Z1, . . . , Zm that are linear combinations of the original features, and then fits a linear model via least squares using these M new features. But unlike PCR, 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.

## Considerations in High Dimensions(P238)

High-Dimensional Data

It is now commonplace to collect an almost unlimited number of feature measurements (p very large). Data sets containing more features than observations are often referred to as high-dimensional.

What Goes Wrong in High Dimensions?

When the number of features p is as large as, or larger than, the number of observations n, least squares cannot (or rather, should not) be performed. The reason is simple: regardless of whether or not there truly is a relationship between the features and the response, least squares will yield a set of coefficient estimates that result in a perfect fit to the data, such that the residuals are zero.

Regression in High Dimensions

We see that new technologies that allow for the collection of measurements for thousands or millions of features are a double-edged sword: they can lead to improved predictive models if these features are in fact relevant to the problem at hand, but will lead to worse results if the features are not relevant.

Interpreting Results in High Dimensions

It is important to instead report results on an independent test set, or cross-validation errors. For instance, the MSE or R2 on an independent test set is a valid measure of model fit, but the MSE on the training set certainly is not.

# Moving Beyond Linearity

## Polynomial Regression(P266)

Polynomial Regression:

yi = β0 + β1xi + β2xi^2 + β3xi^3 + … + βdxi^d + Ei

## Step Functions(P268)

We create cutpoints c1, c2, …, ck in the range of X, and then construct K + 1 new variables:

C0(X) = I (x < c1)

C1(X) = I (c1 <= X < c2)

…

CK-1(X) = I (cK-1 <= X < cK)

CK(X) = I (cK <= X)

where I is an indicator function that returns a 1 if the condition is true, and returns a 0 otherwise. For example, I (cK <= X) equals 1 if cK <= X, and 0 otherwise. Notice that for any value of X, C0(X) + C1(X) + … + CK(X) = 1, since X must be in exactly one of the K+1 intervals. We then use least squares to fit a linear model using C1(X), … ,CK(X) as predictors:

yi = β0 + β1C1(xi) + β2C2(xi) + … + βKCK(xi) + Ei

For a given value of X, at most one of C1, C2, …, CK can be non-zero.

## Basis Functions(P270)

The idea is to have at hand a family of functions or transformations that can be applied to a variable X: b1(X), b2(X), b3(X), …, bK(X). Fit the model:

yi = β0 + β1b1(xi) + β2b2(xi) + β3b3(xi) + … + βKbK(xi) + Ei

## Regression Splines(P271)

Piecewise Polynomials

A piecewise cubic polynomial works by fitting a cubic regression model of the form:

yi = β0 + β1xi + β2xi^2 + β3xi^3 + Ei

where the coefficients β0 β1 β2 β3 differ in different parts of the range of X. The points where the coefficients change are called *knots*.

i.e. A piecewise cubic with no knots is just a standard cubic polynomial, a piecewise cubic polynomial with a single knot at a point c takes the form:

yi = β01 + β11xi + β21xi^2 + β31xi^3 + Ei if xi < c;

yi = β02 + β12xi + β22xi^2 + β32xi^3 + Ei if xi >= c;

using more knots leads to a more flexible piecewise polynomial. If we place K different knots throughout the range of X, then we will end up fitting K+1 different cubic polynomials.

Constraints and Splines

The Spline Basis Representation

A cubic spline with K knots can be modeled as:

yi = β0 + β1b1(xi) + β2b2(xi) + β3b3(xi) + … + β(K+3)b(K+3)(xi) + Ei

b1xi = x, b2xi = x^2, b3xi = x^3, …

To make regression smooth, we do a derivative to yi.

Piecewise Regression and Splines详解：<https://www.youtube.com/watch?v=790G152GYz4>

Choosing the Number and Locations of the Knots

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.

How many knots should we use? One option is to try out different numbers of knots and see which produces the best looking curve.

Comparison to Polynomial Regression

Regression splines often give superior results to polynomial regression. This is because 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.

## Smoothing Splines(P277)

An Overview of Smoothing Splines

In fitting a smooth curve to a set of data, what we really want to do is find some function, say g(x), that fits the observed data well: that is, we want RSS = sum(i=1 to n)(yi − g(xi))^2 to be small. A natural approach is to find the function g that minimizes:

sum(i=1 to n) (yi − g(xi))^2 + λ**∫**g’’(t)^2 dt

where λ is a nonnegative *tuning parameter*. The function g that minimizes is known as a *smoothing spline*.

When λ = 0, then the penalty term in the above function 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.

Choosing the Smoothing Parameter λ

dfλ is a measure of the flexibility of the smoothing spline—the higher it is, the more flexible (and the lower-bias but higher-variance) the smoothing spline. The definition of effective degrees of freedom is somewhat technical. We can write:

gλ = Sλy

The effective degrees of freedom is defined to be:

dfλ = sum(i=1 to n) {Sλ}ii

the sum of the diagonal elements of the matrix Sλ.

## Local Regression(P280)

Local Regression at X = x0

1. Gather the fraction s = k/n of training points whose xi are closest to x0.
2. Assign a weight Ki0 = K(xi, x0) to each point in this neighborhood, so that the point furthest from x0 has weight zero, and the closest has the highest weight. All but these k nearest neighbors get weight zero.
3. Fit a weighted least squares regression of the yi on the xi using the aforementioned weights, by finding β0 and β1 that minimize

sum(i=1 to n) Ki0 (yi - β0 - β1xi)^2

1. The fitted value at x0 is given by f(x0) = β0 + β1x0

In order to perform local regression, there are a number of choices to be made, such as how to define the weighting function K, and whether to fit a linear, constant, or quadratic regression in Step 3 above. While all of these choices make some difference, the most important choice is the *span s*, defined in Step 1 above. The span plays a role like that of the tuning parameter λ in smoothing splines: it controls the flexibility of the non-linear fit. 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. We can again use cross-validation to choose s, or we can specify it directly.

local regression详解：<https://www.youtube.com/watch?v=Vf7oJ6z2LCc&t=1s>

## Generalized Additive Models(P282)

Generalized additive models (GAMs) provide a general framework for extending a standard linear model by allowing non-linear functions of each of the variables, while maintaining additivity.

GAMs for Regression Problems

A natural way to extend the multiple linear regression model

yi = β0 + β1xi1 + β2xi2 + … + βpxip + Ei

In order to allow for non-linear relationships between each feature and the response is to replace each linear component βjxij with a (smooth) non-linear function fj(xij). We would then write the model as

yi = β0 + f1(xi1) + f2(xi2) + … + fp(xip) + Ei

Pros of GAMs:

1. 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. This means that we do not need to manually try out many different transformations on each variable individually.
2. The non-linear fits can potentially make more accurate predictions for the response Y.
3. Because the model is additive, we can still examine the effect of each Xj on Y individually while holding all of the other variables fixed. Hence if we are interested in inference, GAMs provide a useful representation.
4. The smoothness of the function fj for the variable Xj can be summarized via degrees of freedom.

Con of GAMs:

The main limitation of GAMs is that the model is restricted to be additive. With many variables, important interactions can be missed. However, as with linear regression, we can manually add interaction terms to the GAM model by including additional predictors of the form Xj × Xk. In addition we can add low-dimensional interaction functions of the form fjk(Xj,Xk) into the model; such terms can be fit using two-dimensional smoothers such as local regression, or two-dimensional splines (not covered here).

GAMs for Classification Problems

A natural way to extend logistic regression to allow for non-linear relationships is to use the model:

log(p(X)/(1-p(X)) = β0 + f1(X1) + f2(X2) + … + fp(Xp)

Local Regression和GAM详解：<https://www.youtube.com/watch?v=0PQKHlwh49A>

非线性总览：<https://zhuanlan.zhihu.com/p/265768367>

# Tree-Based Methods

## The Basics of Decision Trees(P303)

Regression Trees

In keeping with the tree analogy, the regions R1, R2, R3 are known as terminal nodes or leaves of the tree.

Building a regression 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.
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.

We take a top-down, greedy approach that is known as recursive binary splitting.

A better strategy is to grow a very large tree T0, and then prune it back in order to obtain a subtree.

Algorithm: Building a Regression Tree:

1. Use recursive binary splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations.
2. Apply cost complexity pruning to the large tree in order to obtain a sequence of best subtrees, as a function of alpha.
3. Use K-fold cross-validation to choose alpha. That is, divide the training observations into K folds. For each k=1,...,K:
4. Repeat Steps 1 and 2 on all but the kth fold of the training data.
5. Evaluate the mean squared prediction error on the data in the left-out kth fold, as a function of alpha.
6. Average the results for each value of alpha, and pick alpha to minimize the average error.

4. Return the subtree from Step 2 that corresponds to the chosen value of alpha.

Regression Tree详解：<https://www.youtube.com/watch?v=g9c66TUylZ4>

Classification Trees

A classification tree is very similar to a regression tree, except that it is used to predict a qualitative response rather than a quantitative one.

Trees Versus Linear Models

Linear regression assumes a model of the form:

f(X) = β0 + sum(j=1 to p) Xjβj

whereas regression trees assume a model of the form:

f(X) = sum(m=1 to M) cm \* 1(x belongs to Rm)

where R1,....,Rm represents a partition of feature space.

Which model is better? It depends on the problem at hand. If the relationship between the features and the response is well approximated by a linear model, then an approach such as linear regression will likely work well, and will outperform a method such as a regression tree that does not exploit this linear structure. If instead there is a highly non-linear and complex relationship between the features and the response, then decision trees may outperform classical approaches.

Advantages and Disadvantages of Trees

Advantages:

1. Trees are very easy to explain to people. In fact, they are even easier to explain than linear regression!
2. Some people believe that decision trees more closely mirror human decision-making than do the regression and classification approaches seen in previous chapters.
3. Trees can be displayed graphically, and are easily interpreted even by a non-expert.
4. Trees can easily handle qualitative predictors without the need to create dummy variables.

Disadvantages:

1. Unfortunately, trees generally do not have the same level of predictive accuracy as some of the other regression and classification approaches seen in this book.
2. 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.

## Bagging, Random Forests, Boosting(P316)

Bagging

The decision trees suffer from high variance. This means that if we split the training data into two parts at random, and fit a decision tree to both halves, the results that we get could be quite different. In contrast, a procedure with low variance will yield similar results if applied repeatedly to distinct data sets; linear regression tends to have low variance, if the ratio of n to p is moderately large. Bootstrap aggregation, or bagging, is a general-purpose procedure for reducing the variance of a statistical learning method; we introduce it here because it is particularly useful and frequently used in the context of decision trees.

To apply bagging to regression trees, we simply construct B regression trees using B bootstrapped training sets, and average the resulting predictions. These trees are grown deep, and are not pruned. Hence each individual tree has high variance, but low bias. Averaging these B trees reduces the variance.

Out-of-Bag Error Estimation

An OOB prediction can be obtained in this way for each of the n observations, from which the overall OOB MSE (for a regression problem) or classification error (for a classification problem) can be computed. The resulting OOB error is a valid estimate of the test error for the bagged model, since the response for each observation is predicted using only the trees that were not fit using that observation.

Random Forests

Random forests provide an improvement over bagged trees by way of a small tweak that decorrelates the trees.

The main difference between bagging and random forests is the choice of predictor subset size m. For instance, if a random forest is built using m = p, then this amounts simply to bagging.

Random Forests详解：<https://www.youtube.com/watch?v=J4Wdy0Wc_xQ>

优缺点：<https://easyai.tech/ai-definition/random-forest/>

Boosting

Recall that bagging involves creating multiple copies of the original training data set using the bootstrap, fitting a separate decision tree to each copy, and then combining all of the trees in order to create a single predictive model. Notably, each tree is built on a bootstrap data set, independent of the other 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.

Boosting has three tuning parameters:

1. The number of trees B. Unlike bagging and random forests, boosting can overfit if B is too large, although this overfitting tends to occur slowly if at all. We use cross-validation to select B.
2. The shrinkage parameter λ, a small positive number. This controls the rate at which boosting learns. Typical values are 0.01 or 0.001, and the right choice can depend on the problem. Very small λ can require using a very large value of B in order to achieve good performance.
3. The number d of splits in each tree, which controls the complexity of the boosted ensemble. Often d = 1 works well, in which case each tree is a stump, consisting of a single split. In this case, the boosted ensemble is fitting an additive model, since each term involves only a single variable. More generally d is the interaction depth, and controls the interaction order of the boosted model, since d splits can involve at most d variables.

Algorithm: Boosting for Regression Trees:

1. Set f(x) = 0 and ri = yi for all i in the training set.
2. For b = 1,2, …, B, repeat:
   1. Fit a tree fb with d splits (d+1 terminal nodes) to the training data (X, r).
   2. Update f by adding in a shrunken version of the new tree:

f(x) <- f(x) + λfb(x)

* 1. Update the residuals,

ri <- ri - λfb(xi)

1. Output the boosted model,

f(x) = sum(b=1 to B) λfb(x)

# Support Vector Machines

## Maximal Margin Classifier(P337)

What Is a Hyperplane?

In a p-dimensional space, a hyperplane is a flat affine subspace of dimension p - 1. For instance, in two dimensions, a hyperplane is a flat one-dimensional subspace – in other words, a line. In three dimensions, a hyperplane is a flat two-dimensional subspace – that is , a plane. In p>3 dimensions, it can be hard to visualize a hyperplane, but a notion of a (p-1)-dimensional flat subspace still applies.

In two dimensions:

β0 + β1X1 + β2X2 = 0

In p-dimensions:

β0 + β1X1 + β2X2 + … + βpXp = 0

Classification Using a Separating Hyperplane

If a separating hyperplane exists, we can use it to construct a very natural classifier: a test observation is assigned a class depending on which side of the hyperplane it is located.

The Maximal Margin Classifier

In order to construct a classifier based upon a separating hyperplane, we must have a reasonable way to decide which of the infinite possible separating hyperplanes to use.

A natural choice is the maximal margin hyperplane, which is the separating hyperplane that is farthest from the training observations.

Maximal Margin概念：<https://www.youtube.com/watch?v=pq88UFYJ2PA>

Construction of the Maximal Margin Classifier

This optimization problem (maximize M, subject to sum(j=1 to p) βj^2 = 1, yi(β0 +... +βpxip)>= M) is actually simpler than it looks, first of all, the constraint:

yi(β0 + β1xi1 + β2xi2 + … + βpxip) >= M

guarantees that each observation will be on the correct side of the hyperplane, provided that M is positive. Second, note that (subject …) is not really a constraint on the hyperplane, since if β0 + β1xi1 + β2xi2 + … + βpxip = 0 defines a hyperplane, then so does k(β0 + β1xi1 + β2xi2 + … + βpxip) = 0for any k != 0.

Hence, M represents the margin of our hyperplane, and the optimization problem chooses β0, β1, . . . , βp to maximize M.

The Non-separable Case

However, as we have hinted, in many cases no separating hyperplane exists, and so there is no maximal margin classifier. The generalization of the maximal margin classifier to the non-separable case is known as the support vector classifier.

## Support Vector Classifiers(P344)

Overview of the Support Vector Classifier

In fact, even if a separating hyperplane does exist, then there are instances in which a classifier based on a separating hyperplane might not be desirable. A classifier based on a separating hyperplane will necessarily perfectly classify all of the training observations; this can lead to sensitivity to individual observations.

In this case, we might be willing to consider a classifier based on a hyperplane that does not perfectly separate the two classes, in the interest of

1. Greater robustness to individual observations, and
2. Better classification of most of the training observations

The support vector classifier, sometimes called a soft margin classifier, does exactly this.

Details of the Support Vector Classifier

ei >= 0, sum(i=1 to n)ei <= C

where C is a nonnegative tuning parameter. C bounds the sum of the ei’s, and so it determines the number and severity of the violations to the margin (and to the hyperplane) that we will tolerate. We can think of C as a budget for the amount that the margin can be violated by the n observations. If C = 0 then there is no budget for violations to the margin, and it must be the case that e1 = . . . = en = 0. For C > 0 no more than C observations can be on the wrong side of the hyperplane, because if an observation is on the wrong side of the hyperplane then ei > 1, and requires that sum(i=1 to n) ei <= C. As the budget C increases, we become more tolerant of violations to the margin, and so the margin will widen. Conversely, as C decreases, we become less tolerant of violations to the margin and so the margin narrows.

## Support Vector Machines(P349)

Classification with Non-linear Decision Boundaries

The support vector classifier is a natural approach for classification in the two-class setting, if the boundary between the two classes is linear. However, in practice we are sometimes faced with non-linear class boundaries.

For instance, rather than fitting a support vector classifier using p features

X1, X2, …, Xp

we could instead fit a support vector classifier using 2p features

X1, X12, X2, X22, …, Xp, Xp2

Then the hyperplane rules would become:

1. maximize M
2. subject to yi(β0 + sum(j=1 to p)βj1xij + sum(j=1 to p)βj2xij2) >= M(1-ei)
3. sum(i=1 to n) ei <= C, ei >= 0, sum(j=1 to p) sum(k=1 to 2) βjk2 = 1

Why does this lead to a non-linear decision boundary? In the enlarged feature space, the decision boundary that results from above is in fact linear. But in the original feature space, the decision boundary is of the form q(x) = 0, where q is a quadratic polynomial, and its solutions are generally non-linear.

The Support Vector Machine

SVM is an extension of the support vector classifier that results from enlarging the feature space in a specific way, using Kernels. A Kernel is a function that quantifies the similarity of two observations. For instance, we could simply take:

K(xi, xi’) = sum(j=1 to p) xij xi’j

This is known as a linear kernel because the support vector classifier is linear in the features.

For instance, one could replace every instance of sum(j=1 to p) xij xi’j with the quantity:

K(xi, xi’) = (1 + sum(j=1 to p) xij xi’j)^d

This is known as a polynomial kernel of degree d, where d is a positive integer.

Another popular choice is the radial kernel, which takes the form:

K(xi, xi’) = exp (-γ sum(j=1 to p) (xij - xi’j)^2)

What is the advantage of using a kernel rather than simply enlarging the feature space using functions of the original features?

One advantage is computational, and it amounts to the fact that using kernels, one need only compute K(xi, xi’) for all (n select 2) distinct pairs i, i’.

## SVMs with More than Two Classes(P355)

One-Versus-One Classification

Suppose that we would like to perform classification using SVMs, and there

are K > 2 classes. A one-versus-One or all-pairs approach constructs (K select 2) SVMs, each of which compares a pair of classes. We classify a test observation using each of the (K select 2) classifiers, and we tally the number of times that the test observation is assigned to each of the K classes. The final classification is performed by assigning the test observation to the class to which it was most frequently assigned in these (K select 2) pairwise classifications.

One-Versus-All Classification

The one-versus-all approach is an alternative procedure for applying SVMs in the case of K>2 classes. We fit K SVMs, each time comparing one of the K classes to the remaining K-1 classes. Let β0k, β1k, …, βpk denote the parameters that result from fitting an SVM comparing the kth class (coded as +1) to the others (coded as -1). Let x denote a test observation. We assign the observation to the class for which β0k + β1kx1 + β2kx2 + … + βpkxp is largest, as this amounts to a high level of confidence that the test observation belongs to the kth class rather than to any of the other classes.

SVM详解：<https://www.youtube.com/watch?v=efR1C6CvhmE>

SVM数学概念：<https://zhuanlan.zhihu.com/p/77750026>

Kernel概念解释得不错：<https://tangshusen.me/2018/10/27/SVM/>

One Vs All和One Vs One区别：<https://towardsdatascience.com/multi-class-classification-one-vs-all-one-vs-one-94daed32a87b>

## Relationship to Logistic Regression(P356)

Due to the similarities between their loss functions, logistic regression and the support vector classifier often give very similar results. When the classes are well separated, SVMs tend to behave better than logistic regression; in more overlapping regimes, logistic regression is often preferred.

Is the SVM unique in its use of kernels to enlarge the feature space to accommodate non-linear class boundaries? The answer to this question is “no”. We could just as well perform logistic regression or many of the other classification methods seen in this book using non-linear kernels; this is closely related to some of the non-linear approaches. However, for historical reasons, the use of non-linear kernels is much more widespread in the context of SVMs than in the context of logistic regression or other methods.

# Unsupervised Learning

## The Challenge of Unsupervised Learning(P373)

Unsupervised learning is often much more challenging. The exercise tends to be more subjective, and there is no simple goal for the analysis, such as prediction of a response. Unsupervised learning is often performed as part of an exploratory data analysis. In unsupervised learning, there is no way to check our work because we don’t know the true answer—the problem is unsupervised.

## Principal Components Analysis(P374)

Unsupervised machine learning is the machine learning task of inferring a function to describe hidden structure from "unlabeled" data.

大意是说对没有标签的数据进行发掘探索就是非监督式学习。

用PCA对数据进行降维时是不需要数据的标签的（也就是在做PCA时，我们会剔除y那列），所以这也符合了非监督学习的定义

What Are Principal Components?

Suppose that we wish to visualize n observations with measurements on a set of p features, X1,X2, . . .,Xp, as part of an exploratory data analysis. We could do this by examining two-dimensional scatterplots of the data, each of which contains the n observations’ measurements on two of the features.

PCA provides a tool to do just this. It finds a low-dimensional representation of a data set that contains as much as possible of the variation. The idea is that each of the n observations lives in p-dimensional space, but not all of these dimensions are equally interesting. 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. Each of the dimensions found by PCA is a linear combination of the p features.

PCA和K-means: <https://f7ed.com/2020/10/31/unsupervised-learning-pca/>

Another Interpretation of Principal Components

The first principal component loading vector has a very special property: it is the line in p-dimensional space that is closest to the n observations. The appeal of this interpretation is clear: we seek a single dimension of the data that lies as close as possible to all of the data points, since such a line will likely provide a good summary of the data. The notion of principal components as the dimensions that are closest to the n observations extends beyond just the first principal component. The first three principal components of a data set span the three-dimensional hyperplane that is closest to the n observations, and so forth. Using this interpretation, together the first M principal component score vectors and the first M principal component loading vectors provide the best M-dimensional approximation (in terms of Euclidean distance) to the ith observation xij .

More on PCA

Scaling the Variables

We typically scale each variable to have standard deviation one before we perform PCA. In certain settings, however, the variables may be measured in the same units. In this case, we might not wish to scale the variables to have standard deviation one before performing PCA. For instance, suppose that the variables in a given data set correspond to expression levels for p genes. Then since expression is measured in the same “units” for each gene, we might choose not to scale the genes to each have standard deviation one.

Uniqueness of the Principal Components

Each principal component loading vector is unique, up to a sign flip. This means that two different software packages will yield the same principal component loading vectors, although the signs of those loading vectors may differ.

The Proportion of Variance Explained

The PVE of the mth principal component is given by:

[sum(i=1 to n) (sum(j=1 to p)φjm xij)^2] / [sum(j=1 to p) sum(i=1 to n) xij^2]

Deciding How Many Principal Components to Use

In general, a n × p data matrix X has min(n − 1, p) distinct principal components. However, we usually are not interested in all of them; rather, we would like to use just the first few principal components in order to visualize or interpret the data. In fact, we would like to use the smallest number of principal components required to get a good understanding of the data.

We typically decide on the number of principal components required to visualize the data by examining a scree plot. We choose the smallest number of principal components that are required in order to explain a sizable amount of the variation in the data. Unfortunately, there is no well-accepted objective way to decide how many principal components are enough. In fact, the question of how many principal components are enough is inherently ill-defined, and will depend on the specific area of application and the specific data set.

Other Uses for Principal Components

We can perform regression using the principal component score vectors as features. In fact, many statistical techniques, such as regression, classification, and clustering, can be easily adapted to use the n ×M matrix whose columns are the first M << p principal component score vectors, rather than using the full n × p data matrix.

## Clustering Methods(P385)

Clustering refers to a very broad set of techniques for finding subgroups, or clusters, in a data set.

Both clustering and PCA seek to simplify the data via a small number of summaries, but their mechanisms are different:

PCA looks to find a low-dimensional representation of the observations that explain a good fraction of the variance;

Clustering looks to find homogeneous subgroups among the observations.

K-Means Clustering

The K-means clustering procedure results from a simple and intuitive mathematical problem.We begin by defining some notation. Let C1, . . ., CK denote sets containing the indices of the observations in each cluster. These sets satisfy two properties:

1. C1 ∪ C2 ∪ … ∪ Ck = {1, …, n}. In other words, each observation belongs to at least one of the K clusters.
2. Ck ∩ Ck’ = ∅ for all k != k’. In other words, the clusters are non-overlapping: no observation belongs to more than one cluster.

K-Means Clustering Algorithm:

1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.
2. Iterate until the cluster assignments stop changing:
3. For each of the K clusters, compute the cluster centroid. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.
4. Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).

It is important to run the algorithm multiple times from different random initial configurations. Then one selects the best solution, i.e. that for which the objective is smallest.

K-means简单演示（step1和上述有些许区别）：<https://www.youtube.com/watch?v=_aWzGGNrcic>

Hierarchical Clustering

One potential disadvantage of K-means clustering is that it requires us to pre-specify the number of clusters K. Hierarchical clustering is an alternative approach which does not require that we commit to a particular choice of K.

Hierarchical Clustering Algorithm:

1. Begin with n observations and a measure (such as Euclidean distance) of all the (n select 2) = n(n-1)/2 pairwise dissimilarities. Treat each observation as its own cluster.
2. For i = n, n-1, …, 2:
3. Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.
4. Compute the new pairwise inter-cluster dissimilarities among the i − 1 remaining clusters.

详解：<https://www.youtube.com/watch?v=7xHsRkOdVwo>

Choice of Dissimilarity Measure

For instance, consider an online retailer interested in clustering shoppers based on their past shopping histories. The goal is to identify subgroups of similar shoppers, so that shoppers within each subgroup can be shown items and advertisements that are particularly likely to interest them. Therefore, for this application, correlation-based distance may be a better choice. In addition to carefully selecting the dissimilarity measure used, one must also consider whether or not the variables should be scaled to have standard deviation one before the dissimilarity between the observations is computed.

Practical Issues in Clustering

Small Decisions with Big Consequences

In order to perform clustering, some decisions must be made.

1. Should the observations or features first be standardized in some way? For instance, maybe the variables should be centered to have mean zero and scaled to have standard deviation one.
2. In the case of hierarchical clustering,
3. What dissimilarity measure should be used?
4. What type of linkage should be used?
5. Where should we cut the dendrogram in order to obtain clusters?
6. In the case of K-means clustering, how many clusters should we look for in the data?

Validating the Clusters Obtained

Any time clustering is performed on a data set we will find clusters. But we really want to know whether the clusters that have been found represent true subgroups in the data, or whether they are simply a result of clustering the noise.

Other Considerations in Clustering

Clustering methods generally are not very robust to perturbations to the data.

A Tempered Approach to Interpreting the Results of Clustering

We have described some of the issues associated with clustering. However, clustering can be a very useful and valid statistical tool if used properly.We mentioned that small decisions in how clustering is performed, such as how the data are standardized and what type of linkage is used, can have a large effect on the results. Therefore, we recommend performing clustering with different choices of these parameters, and looking at the full set of results in order to see what patterns consistently emerge.