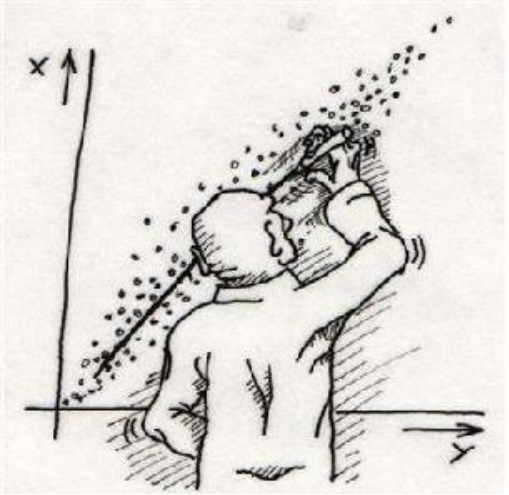
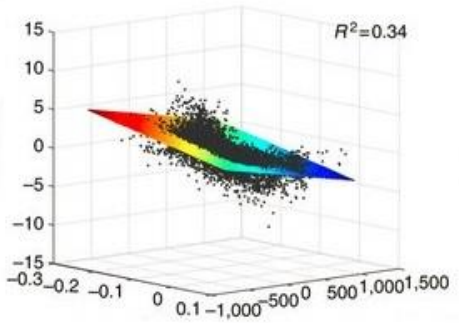
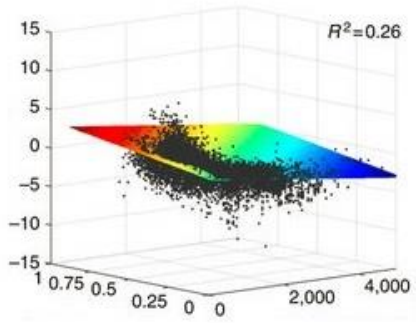


DATA 621 - Business Analytics and Data Mining





Meetup #2

Department of Data Analytics and IS
CUNY School of Professional Studies
The City University of New York

Dr. Nathan Bastian

Variable Selection and Shrinkage Methods

R Lab Demo:

Ordinary Least Squares Regression

Improving the Linear Model

- We may want to improve the simple linear model by replacing OLS estimation with some alternative fitting procedure.

- Why use an alternative fitting procedure?
  - Prediction Accuracy
  - Model Interpretability

Prediction Accuracy

- The OLS estimates have relatively low bias and low variability especially when the relationship between the response and predictors is linear and $n >> p$.

- If n is not much larger than p, then the OLS fit can have high variance and may result in over fitting and poor estimates on unseen observations.

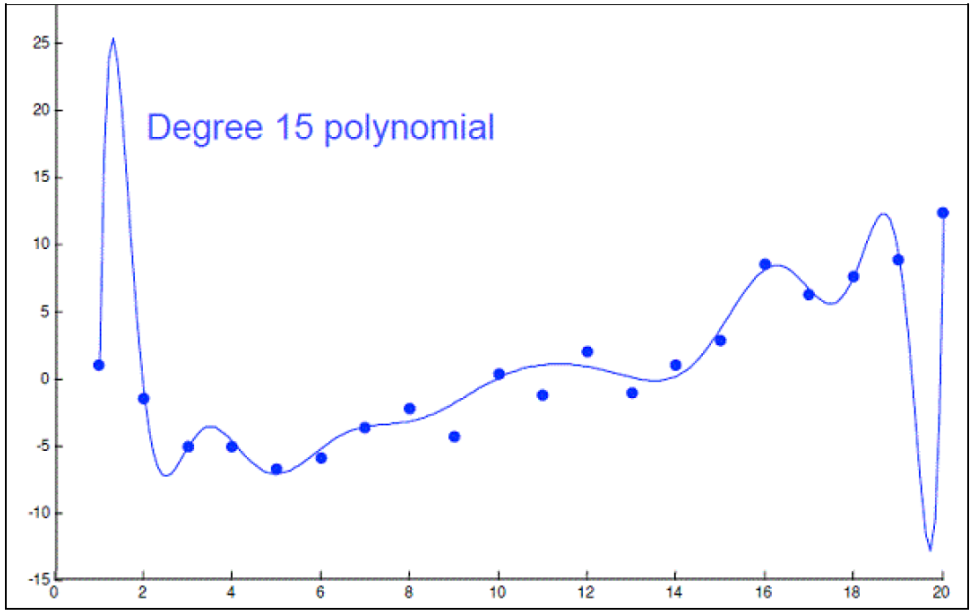
- If $p > n$, then the variability of the OLS fit increases dramatically, and the variance of these estimates in infinite.

Model Interpretability

- When we have a large number of predictors in the model, there will generally be many that have little or no effect on the response.
- Including such irrelevant variable leads to unnecessary complexity.
- Leaving these variables in the model makes it harder to see the effect of the important variables.
- The model would be easier to interpret by removing (i.e., setting the coefficients to zero) the unimportant variables.

Feature/Variable Selection

- Carefully selected features can improve model accuracy, but adding too many can lead to overfitting.
  - Overfitted models describe random error or noise instead of any underlying relationship.
  - They generally have poor predictive performance on test data.



- For instance, we can use a 15-degree polynomial function to fit the following data so that the fitted curve goes nicely through the data points.
- However, a brand new dataset collected from the same population may not fit this particular curve well at all.

Feature/Variable Selection (cont.)

- Subset Selection
  - Identify a subset of the *p* predictors that we believe to be related to the response; then, fit a model using OLS on the reduced set.
  - Methods: best subset selection, stepwise selection

- Shrinkage (Regularization)
  - Involves shrinking the estimated coefficients toward zero relative to the OLS estimates; has the effect of reducing variance and performs variable selection.
  - Methods: ridge regression, lasso

- Dimension Reduction
  - Involves projecting the *p* predictors into a *M*-dimensional subspace, where $M < p$, and fit the linear regression model using the *M* projections as predictors.
  - Methods: principal components regression, partial least squares

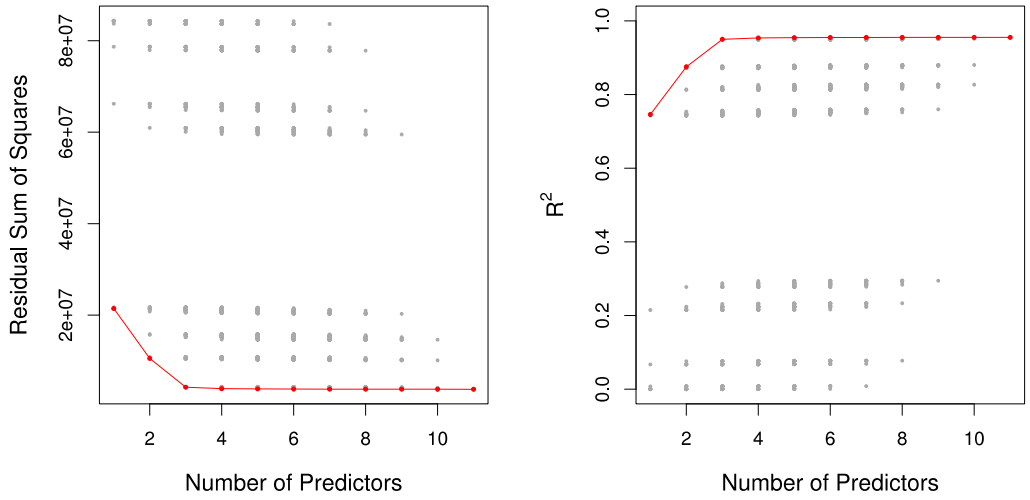
Best Subset Selection

- We fit a separate OLS regression for each possible combination of the *p* predictors:

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Best Subset Selection (cont.)

- The RSS (R^2) will always decline (increase) as the number of predictors included in the model increases, so they are not very useful statistics for selecting the *best* model.
- The red line tracks the best model for a given number of predictors, according to RSS and R^2



Best Subset Selection (cont.)

- While best subset selection is a simple and conceptually appealing approach, it suffers from computational limitations.
- The number of possible models that must be considered grows rapidly as p increases.
- Best subset selection becomes computationally *infeasible* for value of p greater than around 40.

Stepwise Selection

- For computational reasons, best subset selection cannot be applied with very large *p*.
- 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.
- An enormous search space can lead to overfitting and high variance of the coefficient estimates.

Stepwise Selection (cont.)

More attractive methods include:

- Forward Stepwise Selection
  - Begins with a null OLS model containing no predictors, and then adds one predictor at a time that improves the model the most until no further improvement is possible.

- Backward Stepwise Selection
  - Begins with a full OLS model containing all predictors, and then deletes one predictor at a time that improves the model the most until no further improvement is possible.

Forward Stepwise Selection

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   - 2.1 Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   - 2.2 Choose the *best* among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Backward Stepwise Selection

1. Let $\mathcal{M}_p$ denote the *full* model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   - 2.1 Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   - 2.2 Choose the *best* among these k models, and call it $\mathcal{M}_{k-1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Stepwise Selection (cont.)

- Both forward and backward stepwise selection approaches search through only $1 + p(p + 1)/2$ models, so they can be applied in settings where p is too large to apply best subset selection.

- Both of these stepwise selection methods are *not* guaranteed to yield the best model containing a subset of the p predictors.

- Forward stepwise selection can be used even when $n < p$, while backward stepwise selection requires that $n > p$.

- There is a *hybrid* version of these two stepwise selection methods.

Choosing the Optimal Model

- The model containing all the predictors will always have the smallest RSS and the largest R^2, since these quantities are related to the training error.

- We wish to choose a model with low test error, not a model with low training error. Note that training error is usually a poor estimate of test error.

- Thus, RSS and R^2 are not suitable for selecting the *best* model among a collection of models with different numbers of predictors.

Estimating Test Error

1. We can indirectly estimate test error by making an *adjustment* to the training error to account for the bias due to overfitting.

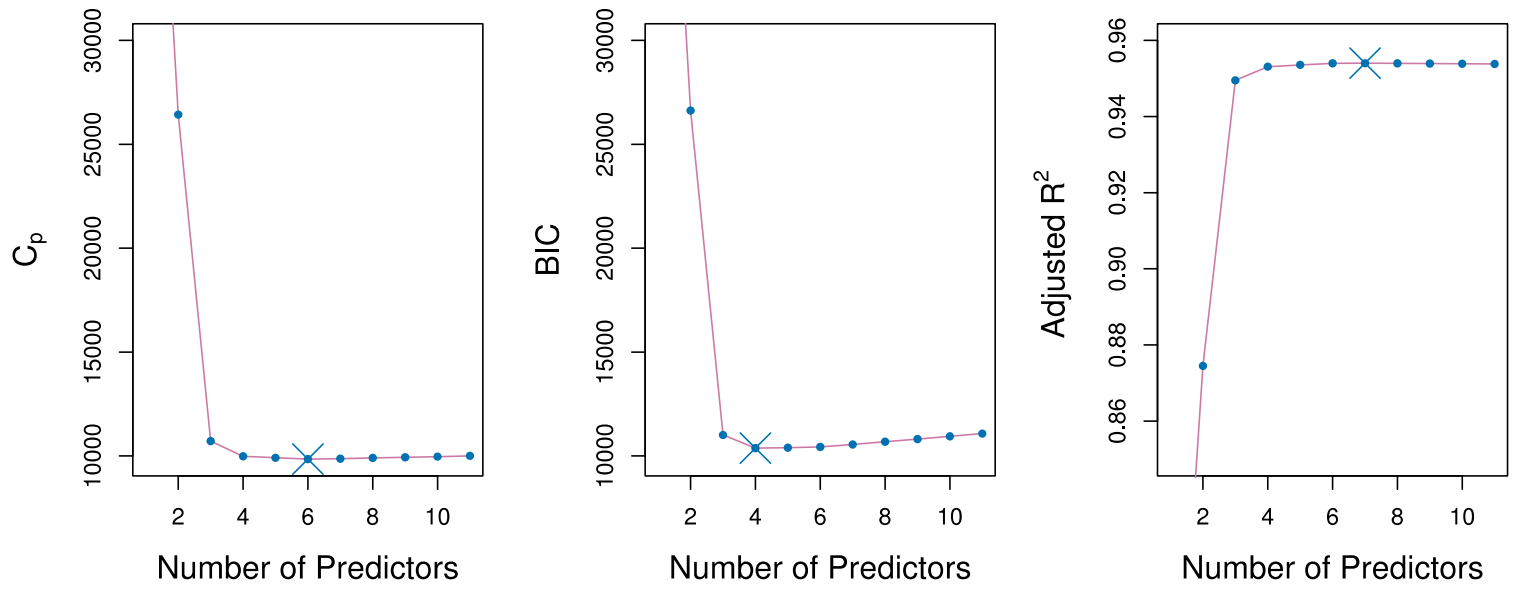
2. We can *directly* estimate the test error, using either a validation set approach or a cross-validation approach.

Other Measures of Comparison

- To compare different models, we can use other approaches:
  - Adjusted R^2
  - AIC (Akaike information criterion)
  - BIC (Bayesian information criterion)
  - Mallow's C_p (equivalent to AIC for linear regression)
- These techniques adjust the training error for the model size, and can be used to select among a set of models with different numbers of variables.
- These methods add penalty to RSS for the number of predictors in the model.

Credit Data: C_p, BIC, and Adjusted R^2

- A small value of C_p and BIC indicates a low error, and thus a better model.
- A large value for the Adjusted R^2 indicates a better model.



Mallow's C_p

- For a fitted OLS model containing *d* predictors, the C_p estimate of test MSE:

$$C_p = \frac{1}{n}\left(\text{RSS} + 2d\hat{\sigma}^2\right)$$

where $\hat{\sigma}^2$ is an estimate of the variance of the error ε associated with each response measurement.

- Here, a penalty is added to the training RSS in order to adjust for the fact that the training error tends to underestimate the test error.

Akaike Information Criterion (AIC)

- Defined for a large class of models fit by maximum likelihood.

$$\mathrm{AIC} = -2 \log L + 2 \cdot d$$

where *L* is the maximized value of the likelihood function for the estimated model.

- In the case of the linear model with Gaussian errors, MLE and OLS are the same thing; thus, C_p and AIC are equivalent.

Bayesian Information Criterion (BIC)

- BIC will tend to take on a small value for a model with a low test error, and so generally we select the model that has the lowest BIC value.

$$\mathrm{BIC} = \frac{1}{n}\left(\mathrm{RSS} + \log(n) d \hat{\sigma}^2\right)$$

- Since log *n* > 2 for an *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 C_p.

- Notice that BIC replaces the $2d\hat{\sigma}^2$ used by C_p with a $\log(n)d\hat{\sigma}^2$ term, where *n* is the number of observations.

Adjusted R^2

- For an OLS model with *d* variables, the adjusted R^2 is calculated:

$$\text{Adjusted } R^2 = 1 - \frac{\text{RSS}/(n-d-1)}{\text{TSS}/(n-1)}$$

where TSS is the total sum of squares.

- Unlike the other statistics, a large value of adjusted R^2 indicates a model with a small test error.

- The adjusted R^2 statistics *pays a price* for the inclusion of unnecessary variables in the model.

Validation and Cross-Validation

- Each of the procedures returns a sequence of models indexed by model size k = 0, 1, 2, … Our job here is to select $\hat{k}$.

- We compute the validation set error or the CV error for each model under consideration, and then select the k for which the resulting estimated test error is smallest.

- This procedure provides a direct estimate of the test error, and it can also be used in a wider range of model selection tasks.

R Lab Demo:

Best Subset Selection using BIC and 10-fold CV

Shrinkage (Regularization) Methods

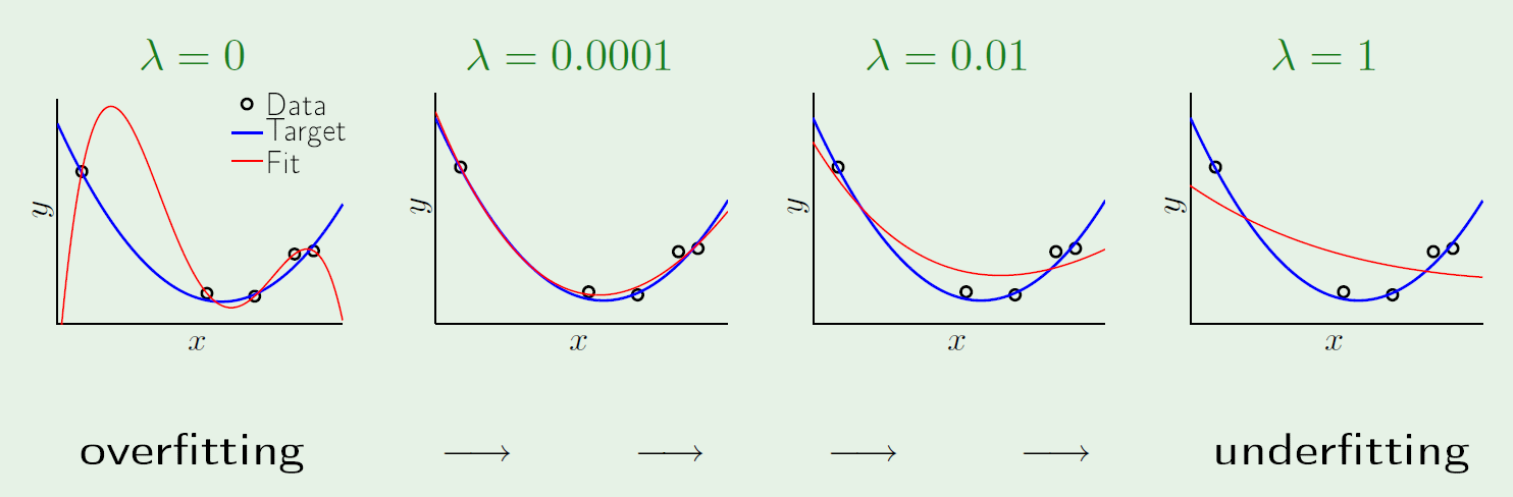
- The subset selection methods use OLS to fit a linear model that contains a subset of the predictors.

- As an alternative, we can fit a model containing all *p* predictors using a technique that constrains or *regularizes* the coefficient estimates (i.e., *shrinks* the coefficient estimates towards zero).

- It may not be immediately obvious why such a constraint should improve the fit, but it turns out that *shrinking* the coefficient estimates can significantly reduce their variance.

Shrinkage (Regularization) Methods (cont.)

- Regularization is our first weapon to combat overfitting.
- It constrains the prediction algorithm to improve out-of-sample error (i.e., test error), especially when noise is present.
- Look at what a little regularization can do:



Ridge Regression

- Recall that the OLS fitting procedure estimates the beta coefficients using the values that minimize:

$$\text{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

- Ridge regression is similar to OLS, except that the coefficients are estimated by minimizing a slightly different quantity:

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \text{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2$$

where $\lambda \geq 0$ is a *tuning parameter*, to be determined separately.

Ridge Regression (cont.)

- Note that $\lambda \geq 0$ is a complexity parameter that controls the amount of shrinkage.
- The idea of penalizing by the sum-of-squares of the parameters is also used in neural networks, where it is known as *weight decay*.
- An equivalent way to write the ridge problem is:

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\text{argmin}} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 \leq t,$$

Ridge Regression (cont.)

- The effect of this equation is to add a shrinkage penalty of the form

$$\lambda \sum_{j=1}^{p} \beta_j^2,$$

where the tuning parameter λ is a positive value.

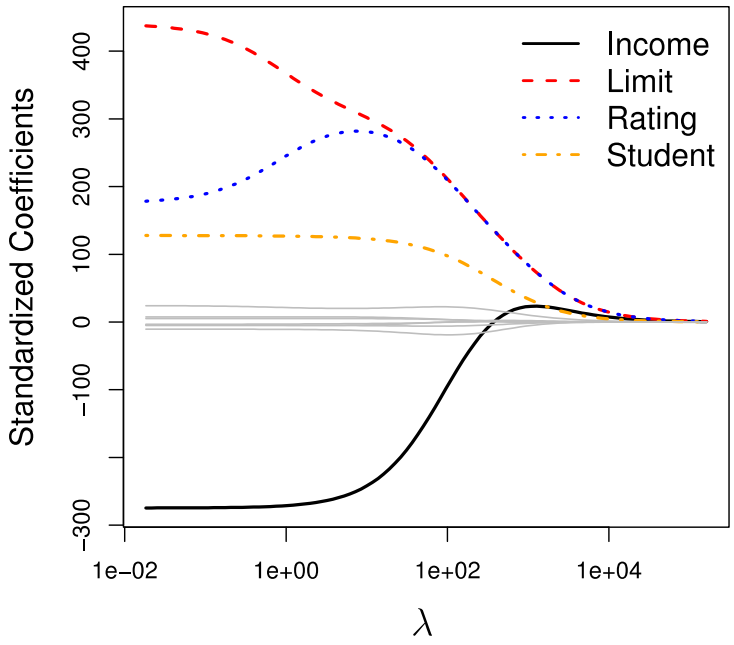
- This has the effect of shrinking the estimated beta coefficients towards zero. It turns out that such a constraint should improve the fit, because shrinking the coefficients can significantly reduce their variance.

- Note that when λ = 0, the penalty term has no effect, and ridge regression will produce the OLS estimates. Thus, selecting a good value for λ is critical (can use cross-validation for this).

Ridge Regression (cont.)

- As λ increases, the standardized ridge regression coefficients shrinks towards zero.

- Thus, when λ is extremely large, then all of the ridge coefficient estimates are basically zero; this corresponds to the *null model* that contains no predictors.



Ridge Regression (cont.)

- The standard OLS coefficient estimates are *scale equivariant*.

- However, the ridge regression coefficient estimates can change *substantially* when multiplying a given predictor by a constant, due to the sum of squared coefficients term in the penalty part of the ridge regression objective function.

- Thus, it is best to apply ridge regression after *standardizing the predictors*:

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2}}$$

Ridge Regression (cont.)

- It turns out that the OLS estimates generally have low bias but can be highly variable.
- In particular when n and p are of similar size or when $n < p$, then the OLS estimates will be extremely variable
- The penalty term makes the ridge regression estimates *biased* but can also substantially reduce variance.
- As a result, there is a bias/variance trade-off.

Ridge Regression (cont.)

Computational Advantages of Ridge Regression

- If p is large, then using the best subset selection approach requires searching through enormous numbers of possible models.

- With ridge regression, for any given λ we only need to fit one model and the computations turn out to be very simple.

- Ridge regression can even be used when $p > n$, a situation where OLS fails completely (i.e., OLS estimates do not even have a unique solution).

The Lasso

- One significant problem of ridge regression is that the penalty term will never force any of the coefficients to be exactly zero.
- Thus, the final model will include all *p* predictors, which creates a challenge in model interpretation
- A more modern alternative is the *lasso*.
- The lasso works in a similar way to ridge regression, except it uses a different penalty term that shrinks some of the coefficients exactly to zero.

The Lasso (cont.)

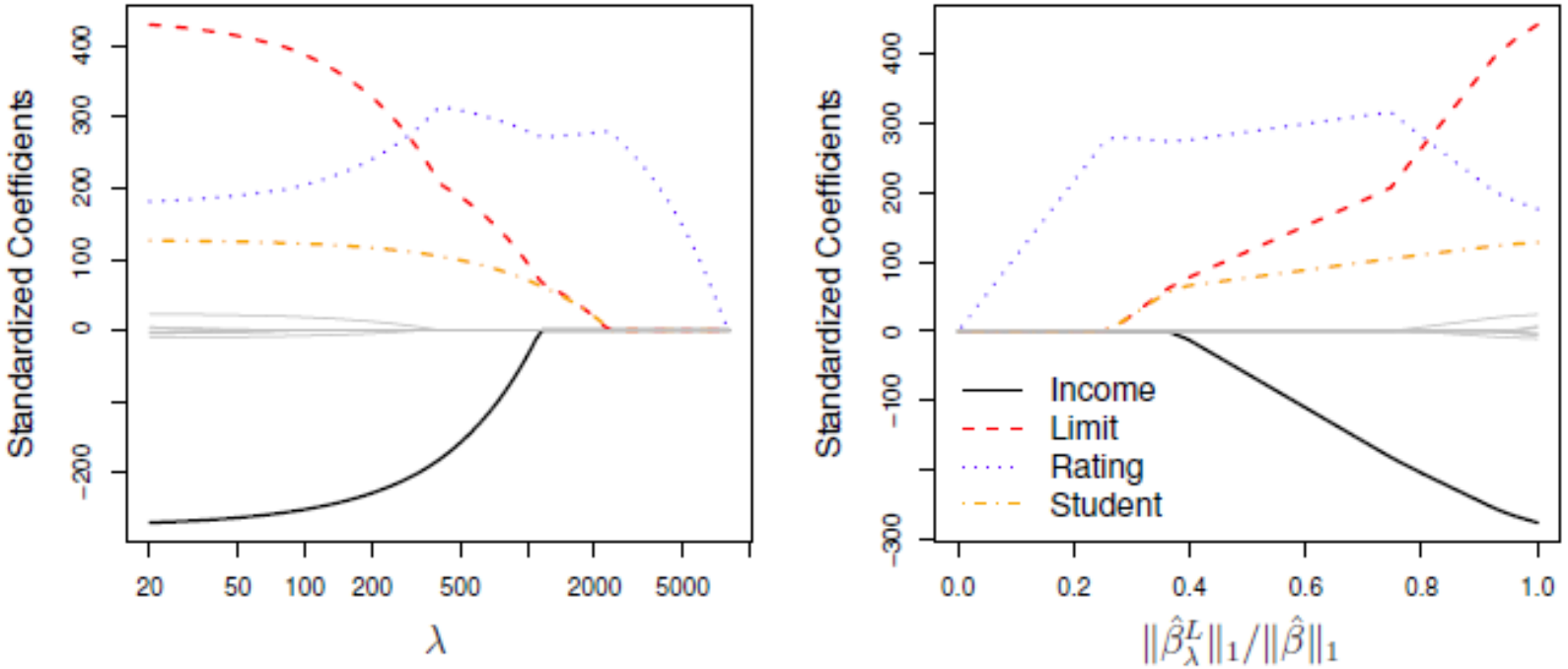
- The lasso coefficients minimize the quantity:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \text{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|$$

- The key difference from ridge regression is that the lasso uses an ℓ_1 penalty instead of an ℓ_2, which has the effect of forcing some of the coefficients to be exactly equal to zero when the tuning parameter λ is sufficiently large.

- Thus, the lasso performs variable/feature selection.

The Lasso (cont.)



- When λ = 0, then the lasso simply gives the OLS fit.
- When λ becomes sufficiently large, the lasso gives the null model in which all coefficient estimates equal zero.

The Lasso (cont.)

- One can show that the lasso and ridge regression coefficient estimates solves the problems:

$$\underset{\beta}{\text{minimize}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq s$$

and

$$\underset{\beta}{\text{minimize}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq s,$$

Lasso vs. Ridge Regression

- The lasso has a major advantage over ridge regression, in that it produces simpler and more interpretable models that involved only a subset of predictors.
- The lasso leads to qualitatively similar behavior to ridge regression, in that as λ increases, the variance decreases and the bias increases.
- The lasso can generate more accurate predictions compared to ridge regression.
- Cross-validation can be used in order to determine which approach is better on a particular data set.

Selecting the Tuning Parameter λ

- As for subset selection, for ridge regression and lasso we require a method to determine which of the models under consideration in best; thus, we required a method selecting a value for the tuning parameter λ or equivalently, the value of the constraint *s*.

- Select a grid of potential values; use cross-validation to estimate the error rate on test data (for each value of λ) and select the value that gives the smallest error rate.

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

R Lab Demo:

Ridge Regression / Lasso Regression with 10-fold CV

Dimension Reduction

- The methods we have discussed so far have involved fitting linear regression models, via OLS or a shrunken approach, using the original predictors.

- We now explore a class of approaches that *transform* the predictors and then fit an OLS model using the transformed variables.

- We refer to these techniques as *dimension reduction* methods.

Dimension Reduction (cont.)

- Let Z_1, Z_2,…,Z_M represent $M < p$ *linear combinations* of our original p predictors. That is,

$$Z_m = \sum_{j=1}^{p} \phi_{mj} X_j$$

for some constants $\phi_{m1}, \ldots, \phi_{mp}$.

- We can then fit an OLS linear regression model,

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \epsilon_i, \quad i = 1, \ldots, n,$$

Dimension Reduction (cont.)

- If the constants $\phi_{m1}, \ldots, \phi_{mp}$ are chosen wisely, then such dimension reduction approaches can outperform OLS regression.

- The term *dimension reduction* comes from the fact that this approach reduces the problem of estimating the $p + 1$ coefficients $\beta_0, \ldots, \beta_p$ to the simpler problem of estimating the $M + 1$ coefficients $\theta_0, \ldots, \theta_M$, where $M < p$.

$$\sum_{m=1}^{M} \theta_m z_{im} = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} \phi_{mj} x_{ij} = \sum_{j=1}^{p} \sum_{m=1}^{M} \theta_m \phi_{mj} x_{ij} = \sum_{j=1}^{p} \beta_j x_{ij}, \qquad \beta_j = \sum_{m=1}^{M} \theta_m \phi_{mj}$$

- This method serves to constrain the estimated β_j coefficients.

Principal Components Regression

- Here, we apply principal components analysis (PCA) to define the linear combinations of the predictors, for use in the regression.
- The *first principal component* is that (normalized) linear combination of the variables with the largest variances.
- The *second principal component* has largest variance, subject to being uncorrelated with the first….etc.
- Thus, with many correlated variables, we replace them with a small set of principal components that capture their joint variation.

Principal Components Regression (cont.)

- The *principal components regression* (PCR) approach involves constructing the first *M* principal components, and then using these components as the predictors in an OLS linear regression model.
- The key idea is that often a small number of principal components suffice to *explain* most of the variability in the data, as well as the relationship with the response.
- We assume that the directions in which $X_1,\ldots,X_p$ show the most variation are the directions that are associated with Y.
- When performing PCR, predictors should be *standardized* prior to generating the principal components.

Principal Components Regression (cont.)

- PCR forms the derived input columns $\mathbf{z}_m = \mathbf{X}v_m$, and then regresses $\mathbf{y}$ on $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_M$ for some $M \leq p$. Since the $\mathbf{z}_m$ are orthogonal, this regression is just a sum of univariate regressions:

$$\hat{\mathbf{y}}^{\text{pcr}}_{(M)} = \bar{y}\mathbf{1} + \sum_{m=1}^{M} \hat{\theta}_m \mathbf{z}_m$$

where $\hat{\theta}_m = \langle \boldsymbol{z}_m, \boldsymbol{y} \rangle / \langle \boldsymbol{z}_m, \boldsymbol{z}_m \rangle$. Since $\mathbf{z}_m$ are each linear combinations of the original $\mathbf{x}_j$, we can express the solution in terms of coefficients of the $\mathbf{x}_j$:

$$\hat{\beta}^{\text{pcr}}(M) = \sum_{m=1}^{M} \hat{\theta}_m v_m$$

- PCR discards the $p - M$ smallest eigenvalue components.

Principal Components Regression (cont.)

- By manually setting the projection onto the principal component directions with small eigenvalues set to 0 (i.e., only keeping the large ones), dimension reduction is achieved.

- PCR is very similar to ridge regression in a certain sense.

- Ridge regression can be viewed conceptually as projecting the *y* vector onto the principal component directions and then shrinking the projection on each principal component direction.

Principal Components Regression (cont.)

- The amount of shrinkage depends on the variance of that principal component.
- Ridge regression shrinks everything, but it never shrinks anything to zero.
- By contrast, PCR either does not shrink a component at all or shrinks it to zero.

Principal Components Regression (cont.)

- As more principal components are used in the regression model, the bias decreases but the variance increases.
- 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.
- We note that even though PCR provides a simple way to perform regression using $M < p$ predictors, it *is not* a feature selection method.
- In PCR, the number of principal components is typically chosen by cross-validation.

R Lab Demo:

Principal Component Regression

Partial Least Squares

- PCR identifies linear combinations, or *directions*, that best represents the predictors.
- These directions are identified is 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.
- PCR suffers from a potentially serious 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 (cont.)

- Like PCR, *partial least squares* (PLS) is a dimension reduction method, which first identifies a new set of features $Z_1,\ldots,Z_M$ that are linear combinations of the original features.
- Then PLS fits an OLS linear model using these *M* new features.
- Unlike PCR, PLS identifies these new features in a *supervised* way; PLS 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*.
- The PLS approach attempts to find directions that help explain both the response and the predictors.

Partial Least Squares (cont.)

- After standardizing the p predictors, PLS computes the first partial least squares direction Z_1 by setting each ϕ_{1j} in

$$Z_m = \sum_{j=1}^{p} \phi_{mj} X_j$$

equal to the coefficient from the simple linear regression of Y onto X_j.

- One can show that this coefficient is proportional to the correlation between Y and X_j.

Partial Least Squares (cont.)

- Hence, in computing $Z_1 = \sum_{j=1}^{p} \phi_{1j} X_j$, PLS places the highest weight on the variables that are most strongly related to the response.
- Subsequent directions are found by taking residuals and then repeating the above prescription.
- As with PCR, the number *M* of PLS directions used in PLS is a tuning parameters that is typically chosen by cross-validation.
- While the supervised dimension reduction of PLS can reduce bias, it also has the potential to increase variance.

R Lab Demo:

Partial Least Squares Regression

Considerations in High Dimensions

- While *p* can be extremely large, the number of observations *n* is often limited due to cost, sample availability, etc.
- Data sets containing more features than observations are often referred to a *high-dimensional*.
- When the number of features *p* is as large as, or larger than, the number of observations *n*, OLS should not be performed.
  - It is too *flexible* and hence overfits the data.
- Forward stepwise selection, ridge regression, lasso, and PCR are particularly useful for performing regression in the high-dimensional setting.

Considerations in High Dimensions (cont.)

- Regularization or shrinkage plays a key role in high-dimensional problems.
- Appropriate tuning parameter selection is crucial for good predictive performance.
- The test error tends to increase as the dimensionality of the problem (i.e., the number of predictors) increases, unless the additional features are truly associated with the response.
  - Known as the *curse of dimensionality*

Considerations in High Dimensions (cont.)

- *Curse of dimensionality*
  - Adding additional *signal* features that are truly associated with the response will improve the fitted model, in the sense of leading to a reduction in test set error.
  - Adding *noise* features that are not truly associated with the response will lead to a deterioration in the fitted model, and consequently an increased test set error.

- Noise features increase the dimensionality of the problem, exacerbating the risk of overfitting without any potential upside in terms of improved test set error.

Considerations in High Dimensions (cont.)

- In the high-dimensional setting, the multicollinearity problem is extreme: any variable in the model can be written as a linear combination of all of the other variables in the models.
- It is also important to be particularly careful in reporting errors and measures of model fit in the high-dimensional setting.
- One should *never* use sum of squared errors, p-values, R^2 statistics, or other traditional measures of model fit on the *training data* as evidence of good model fit in the high-dimensional setting.
- It is important to report results on an independent test set, or cross-validation errors.

Binary Logistic Regression

Classification

- Predicting a *qualitative* response for an observation can be referred to as *classifying* that observation, since it involves assigning the observation to a category, or class.

- Thus, *classification models* are supervised learning methods for which the true class labels for the data points are given in the training data.

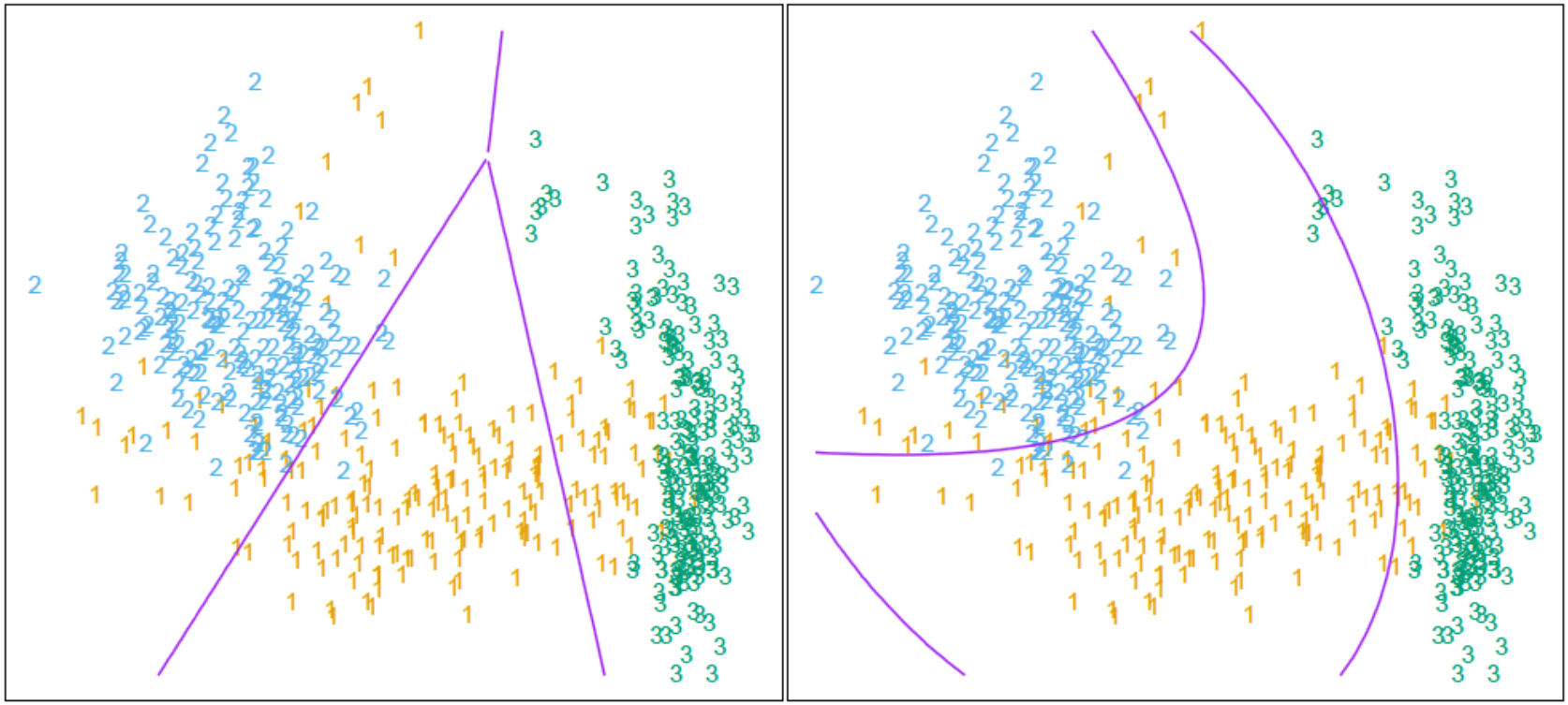
- The methods used for classification often predict the probability of each of the categories of a qualitative variable as the basis for making the classification decision.

Classification Setting

- Training data: $\{(x_1,g_1), (x_2,g_2),\ldots,(x_N,g_N)\}$
- The feature vector $X = (X_1, X_2,\ldots,X_p)$, where each X_j is quantitative.
- The response variable G is categorical s.t. $G \in G = \{1,2,\ldots,K\}$
- Form a predictor $G(x)$ to predict G based on X.
- Note that $G(x)$ divides the input space (feature vector space) into a collection of regions, each labeled by one class.

Classification Setting (cont.)

- For each plot, the feature vector space is divided into three pieces, each assigned with a particular class.



Classification Error Rate

- The *classification error rate* is the number of observations that are misclassified over the sample size:

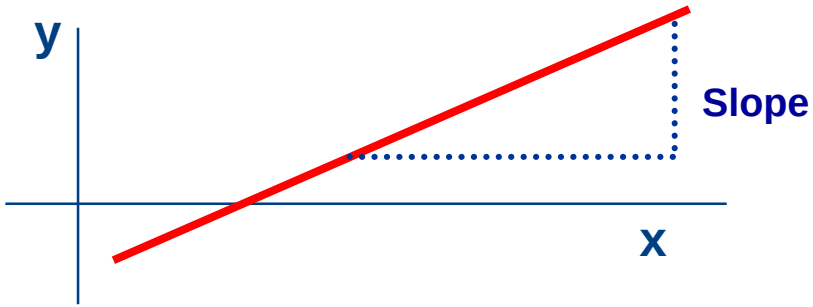
$$\frac{1}{n}\sum_{i=1}^{n} I(\hat{Y}_i \neq Y_i)$$

where $I(\hat{Y}_i \neq Y_i) = 1$ if $\hat{Y}_i \neq Y_i$ and 0 otherwise.

- For binary classification let $\hat{Y}$ be a 0-1 vector of the predicted class labels and Y be a 0-1 vector of the observed class labels.

Review: Simple Linear Regression

- Relation between 2 continuous variables (systolic blood pressure [SBP] and age)



$$y = \alpha + \beta_1 x_1$$

- Regression coefficient b1
  - Measures association between y and x
  - Amount by which y changes on average when x changes by one unit
  - Ordinary least squares method (equivalent to MLE)

Review: Multiple Linear Regression

- Relation between a continuous variable and a set of *i* continuous variables

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i$$

- Partial regression coefficients b_i
  - Amount by which y changes on average when x_i changes by one unit and all the other x_is remain constant
  - Measures association between x_i and y adjusted for all other x_i
- Example
  - SBP *versus* age, weight, height, etc.

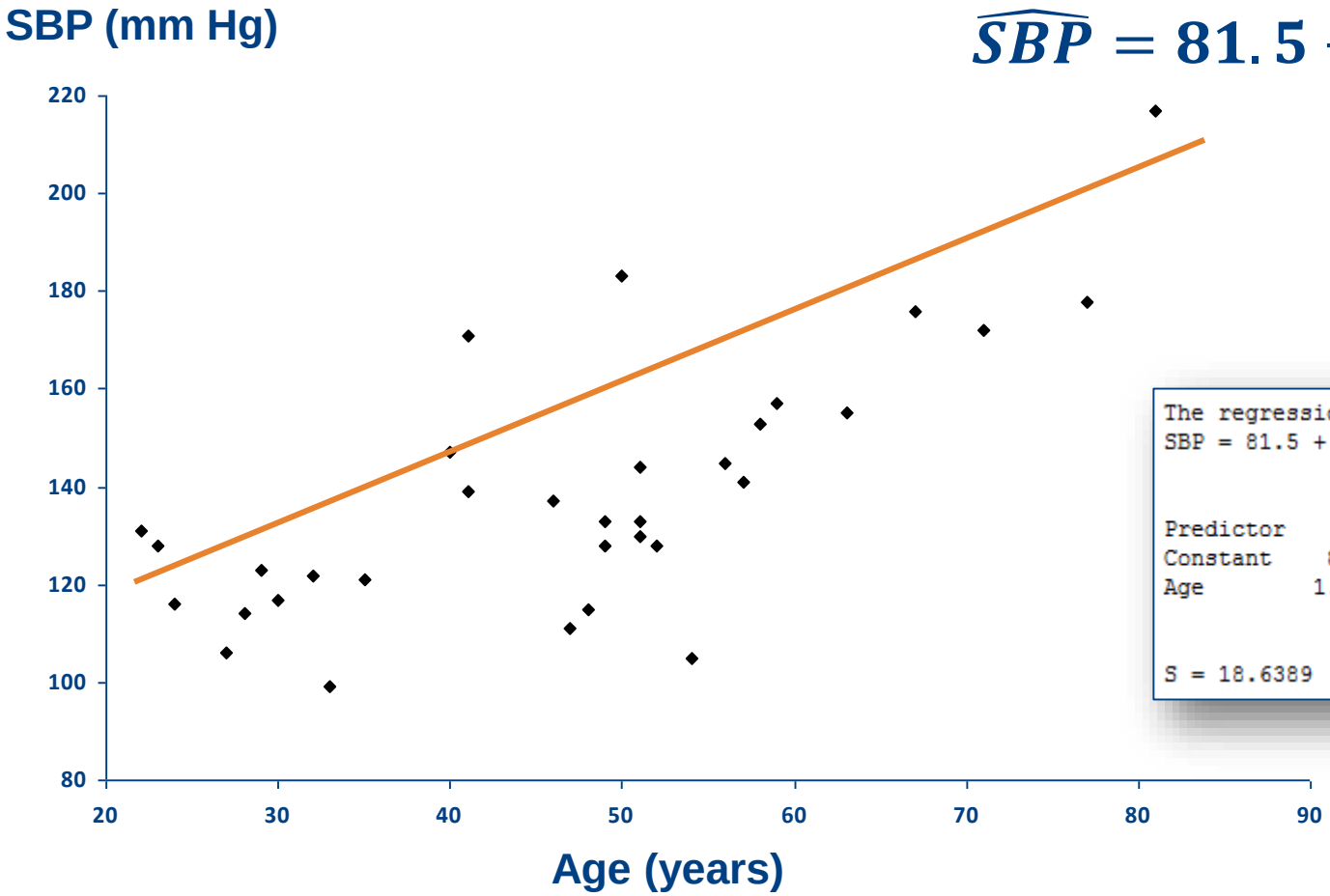
Healthcare Example

Age and systolic blood pressure (SBP) among 33 adult women

Age	SBP	Age	SBP	Age	SBP
22	131	41	139	52	128
23	128	41	171	54	105
24	116	46	137	56	145
27	106	47	111	57	141
28	114	48	115	58	153
29	123	49	133	59	157
30	117	49	128	63	155
32	122	50	183	67	176
33	99	51	130	71	172
35	121	51	133	77	178
40	147	51	144	81	217

Simple Linear Regression

$$\widehat{SBP} = 81.5 + 1.22 * Age$$



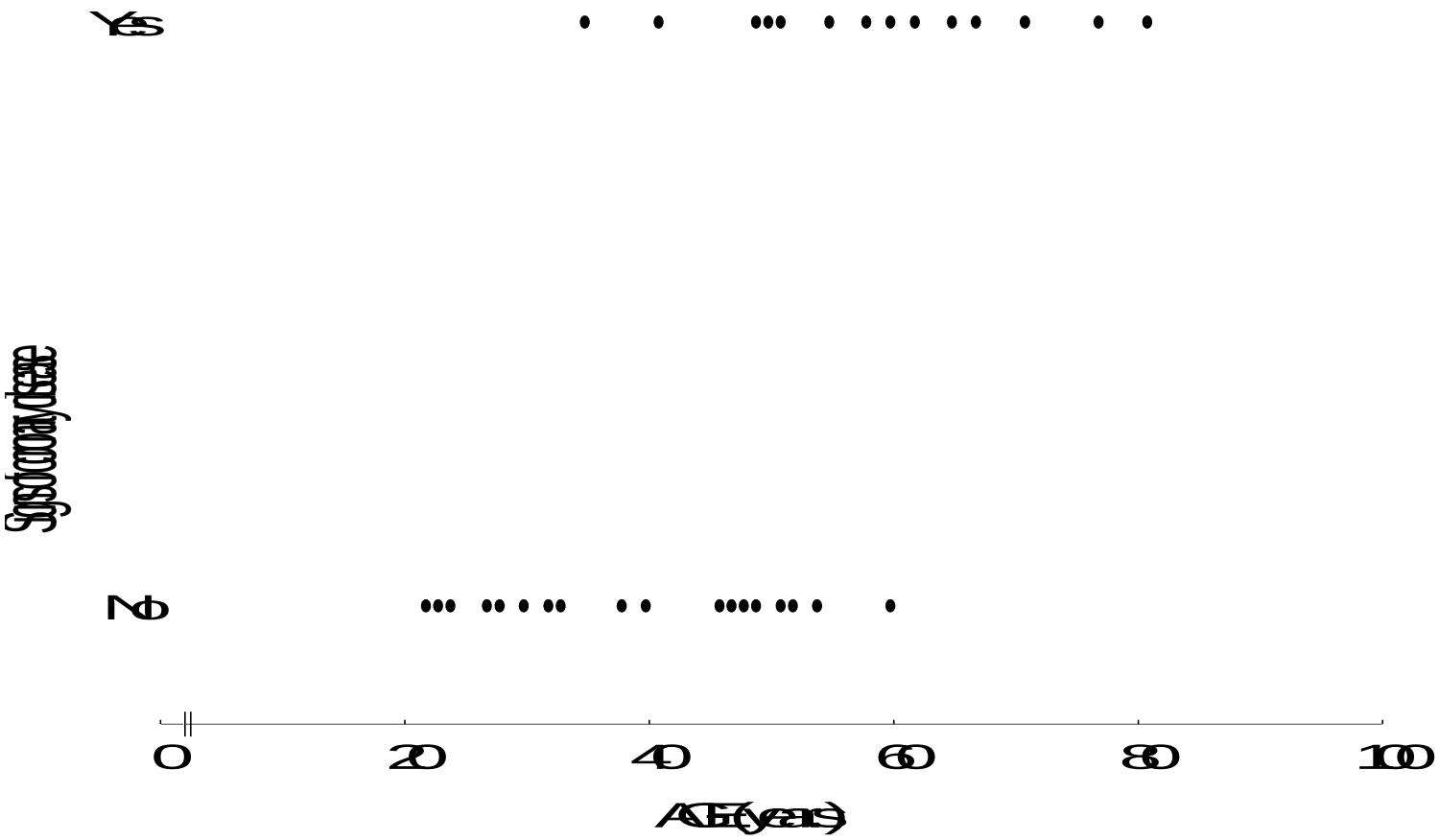

```
The regression equation is
SBP = 81.5 + 1.22 Age


Predictor     Coef  SE Coef      T      P
Constant     81.52    10.47   7.79  0.000
Age         1.2224   0.2129   5.74  0.000


S = 18.6389   R-Sq = 51.5%   R-Sq(adj) = 50.0%
```

Healthcare Example (cont.)

Age and signs of coronary heart disease (CD)

Age	CD	Age	CD	Age	CD
22	0	40	0	54	0
23	0	41	1	55	1
24	0	46	0	58	1
27	0	47	0	60	1
28	0	48	0	60	0
30	0	49	1	62	1
30	0	49	0	65	1
32	0	50	1	67	1
33	0	51	0	71	1
35	1	51	1	77	1
38	0	52	0	81	1

Dot-plot: Healthcare Data



Linear Methods for Classification

- *Decision boundaries* are linear.

- Two class problem:
  - The decision boundary between the two classes is a *hyperplane* in the feature vector space.

  - A hyperplane in the *p* dimensional input space is the set:

$$\{x : \alpha_o + \sum_{j=1}^{p} \alpha_j x_j = 0\}$$

Linear Methods for Classification (cont.)

- The two regions separated by a hyperplane:

$$\{x : \alpha_o + \sum_{j=1}^{p} \alpha_j x_j > 0\} \qquad \{x : \alpha_o + \sum_{j=1}^{p} \alpha_j x_j < 0\}$$

- For more than two classes, the decision boundary between any pair of classes *k* and *l* is a hyperplane.

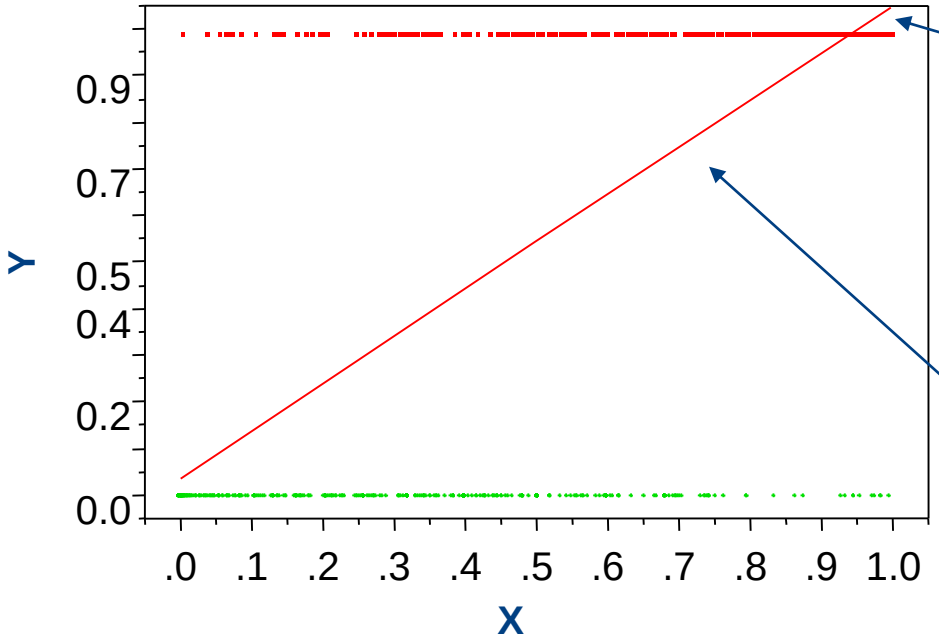
- Example method for deciding the hyperplane:
  - Logistic regression

Can we use Linear Regression?

- One rather formal justification is to view the linear regression as an estimate of conditional expectation.
- However, how good an approximation to conditional expectation is the rather rigid linear regression model?
- The key issue is that the fitted values can be negative or greater than 1.
- This is a consequence of the rigid nature of linear regression, especially if we make predictions outside the domain of the training data.

Can we use Linear Regression?

- When *Y* only takes on values of 0 and 1, we can see why is OLS linear regression is often not appropriate.



How do we interpret values greater than 1?

How do we interpret values of *Y* between 0 and 1?

Linear vs. Logistic Regression

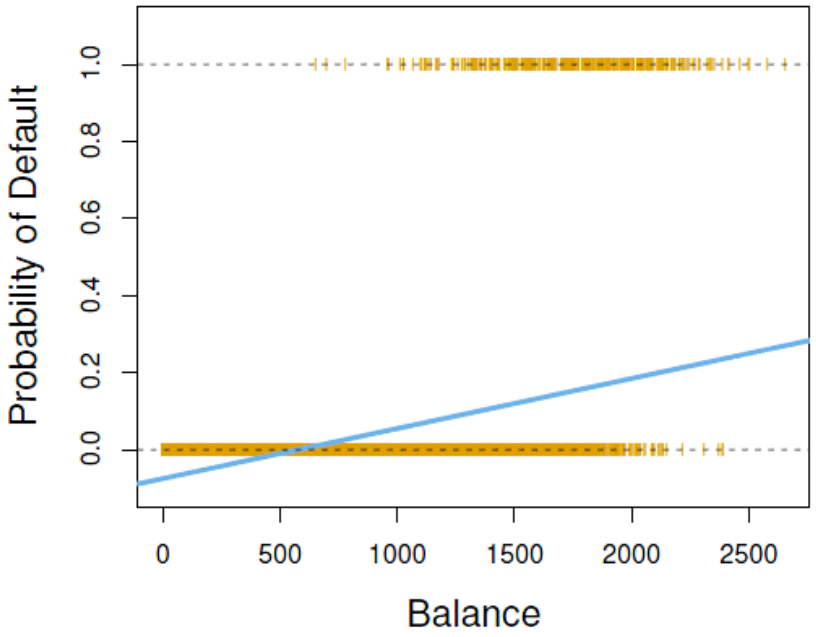
- Linear regression might produce probabilities less than zero or bigger than one because the regression line can take on any value between negative and positive infinity.

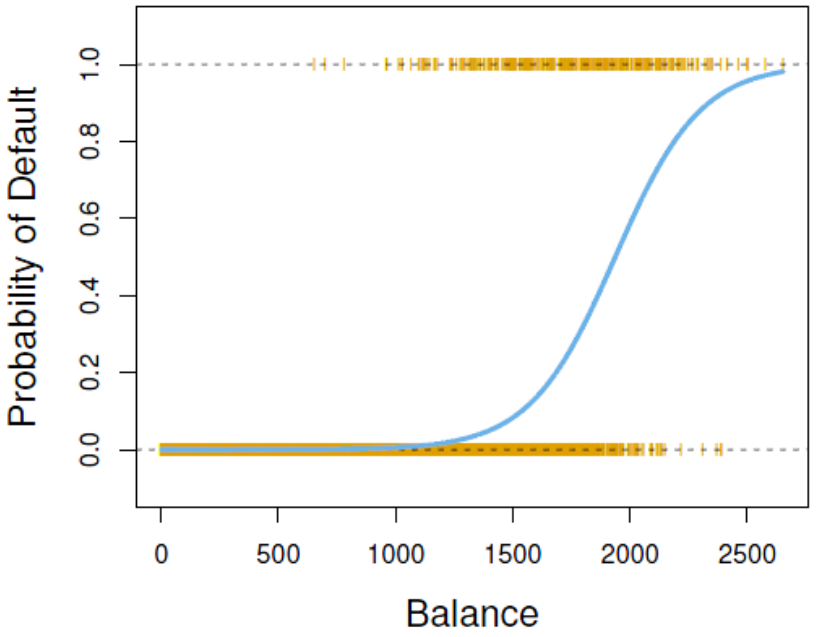
- Thus, the regression line almost always predicts the wrong value for *Y* in classification problems.

- *Logistic regression* is more appropriate, especially because we are interested in estimating the *probabilities* that *X* belong to each category, or class.

Linear vs. Logistic Regression (cont.)

- Below, the orange marks indicate the response Y, either 0 or 1. Linear regression does not estimate $\Pr(Y = 1|X)$ well, so logistic regression seems well suited to the task.





Logistic Regression

- There are two big branches of methods for classification. One is called *generative* modeling, and the other is called *discriminative* modeling.
  - A **generative** model learns the joint probability distribution.
  - A **discriminative** model learns the conditional probability distribution.

- *Logistic regression* for classification is a discriminative modeling approach, where we estimate the *posterior probabilities* of classes given X directly without assuming the marginal distribution on X.

- As a result, this method preserves linear classification boundaries.

Logistic Regression (cont.)

- A review of Bayes rule shows us that when we use 0-1 loss, we pick the class *k* that has the maximum posterior probability:

$$\hat{G}(x) = \arg\max_k Pr(G = k|X = x)$$

- The decision boundary between classes *k* and *l* is determined by:

$$Pr(G = k|X = x) = Pr(G = l|X = x)$$

- That is, the *x*'s at which the two posterior probabilities of *k* and *l* are equal.

Logistic Regression (cont.)

- If we divide both sides by $Pr(G = l \mid X = x)$ and take the log of this ratio, the previous equation is equivalent to:

$$\log \frac{Pr(G = k|X = x)}{Pr(G = l|X = x)} = 0$$

- Since we want to enforce a linear classification boundary, we assume the function above is linear:

$$\log \frac{Pr(G = k|X = x)}{Pr(G = l|X = x)} = a_0^{(k,l)} + \sum_{j=1}^{p} a_j^{(k,l)} x_j$$

- The is the basic assumption of logistic regression.

Logistic Regression (cont.)

- Models the relationship between a set of variables x_i
  - dichotomous (eat : yes/no)
  - categorical (social class, ...)
  - continuous (age, ...)

✖ And dichotomous variable Y

- Dichotomous (binary) outcome most common situation in biology and epidemiology

Logistic Regression (cont.)

- We use the superscript (k, l) on the coefficients of the linear function because for every pair of k and l, the decision boundary would be different, determined by the different coefficients.

- For logistic regression, there are restrictive relations between $a^{(k,l)}$ for different pairs of (k, l).

- We do not really need to specify this equation for every pair of k and l, but instead we only need to specify it for K – 1 such pairs.

Logistic Regression (cont.)

- If we take class *K* as the base class, the assumed equations are:

$$\log \frac{Pr(G=1|X=x)}{Pr(G=K|X=x)} = \beta_{10} + \beta_1^T x$$
$$\log \frac{Pr(G=2|X=x)}{Pr(G=K|X=x)} = \beta_{20} + \beta_2^T x$$
$$\vdots$$
$$\log \frac{Pr(G=K-1|X=x)}{Pr(G=K|X=x)} = \beta_{(K-1)0} + \beta_{K-1}^T x$$

- This indicates that we do not have to specify the decision boundary for every pair of classes. We only need to specify the decision boundary between class *j* and the base class *K*.

Logistic Regression (cont.)

- Once we have specified the parameters for these K – 1 log ratios, then for any pair of classes (k, l), we can derive the log ratios without introducing new parameters:

$$\log \frac{Pr(G = k|X = x)}{Pr(G = l|X = x)} = \beta_{k0} + \beta_{l0} + (\beta_k - \beta_l)^T x$$

- Number of parameters: (K – 1)(p + 1)

- We denote the entire parameter set by θ and arrange them as:

$$\theta = \{\beta_{10}, \beta_1^T, \beta_{20}, \beta_2^T, \ldots, , \beta_{(K-1)0}, \beta_{K-1}^T\}$$

Logistic Regression (cont.)

- The log ratios of posterior probabilities are called *log-odds* or *logit* transformations.

- Under the previously stated assumptions, the posterior probabilities are given by the following two equations:

$$Pr(G = k|X = x) = \frac{\exp\left(\beta_{k0} + \beta_k^T x\right)}{1 + \sum_{l=1}^{K-1} \exp\left(\beta_{l0} + \beta_l^T x\right)} \text{ for } k = 1, \ldots, K-1$$

$$Pr(G = K|X = x) = \frac{1}{1 + \sum_{l=1}^{K-1} \exp\left(\beta_{l0} + \beta_l^T x\right)}$$

Logistic Regression (cont.)

- For Pr($G = k \mid X = x$) given previously:
  - These must sum to 1:

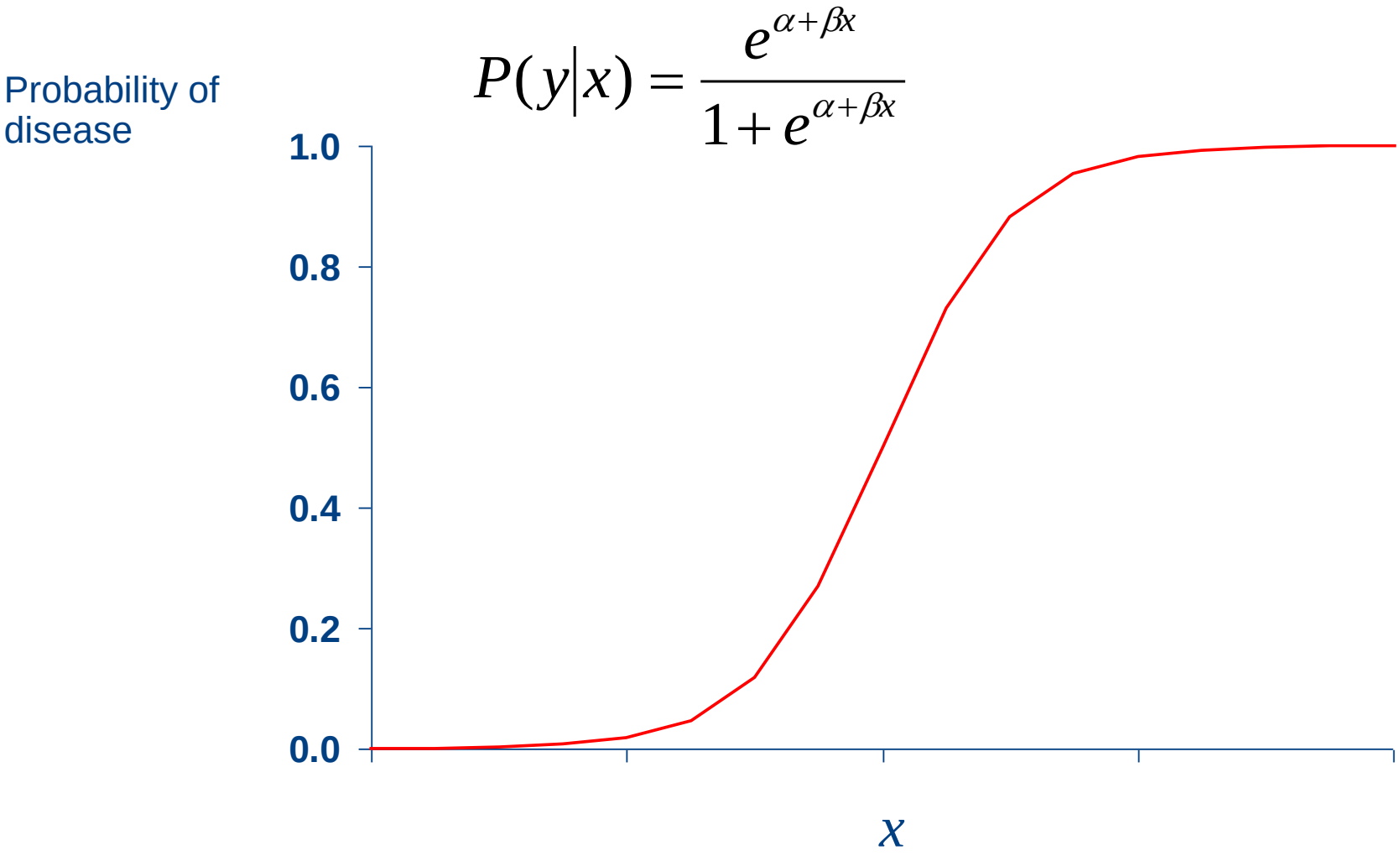
$$\sum_{k=1}^{K} Pr(G = k|X = x) = 1$$

- Similarities with linear regression on indicators:
  - Both attempt to estimate Pr($G = k \mid X = x$), both have linear classification boundaries, and the posterior probabilities sum to 1 across classes

- Differences with linear regression on indicators:
  - For linear regression, approximate Pr($G = k \mid X = x$) by a linear function of x; it is not guaranteed to fall between 0 and 1.
  - For logistic regression, Pr($G = k \mid X = x$) is a nonlinear (sigmoid) function of x; it is guaranteed to range from 0 to 1.

Logistic Function

$$P(y|x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$$



Logistic Function (cont.)

$$P(y|x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$$

Log-odds (logit) of $P(y|x)$

$$\ln\left[\frac{P(y|x)}{1-P(y|x)}\right] = \alpha + \beta x$$

- Properties of the logit
  - Simple transformation of P(y|x)
  - Linear relationship with x
  - Can be continuous (Logit between - ∞ to + ∞)
  - Known binomial distribution (P between 0 and 1)
  - Directly related to the notion of odds of disease

Interpretation of b: Binary case

	Exposure (x)	
Disease (y)	**Yes**	**No**
Yes	$P(y\|x=1)$	$P(y\|x=0)$
No	$1-P(y\|x=1)$	$1-P(y\|x=0)$

$$\ln\left[\frac{P(y|x)}{1-P(y|x)}\right]=\alpha+\beta x$$

$$odds\ (Disease \mid Exposure) = e^{\alpha+\beta}$$

$$odds\ (Disease \mid No\ Exposure) = e^{\alpha}$$

$$OR=\frac{e^{\alpha+\beta}}{e^{\alpha}}$$

$$\ln(OR)=\beta$$

Logistic Regression (cont.)

- How do we estimate the parameters and how do we fit a logistic regression model?

- What we want to do is find parameters that *maximize* the conditional *likelihood* of class labels *G* given *X* using the training data.

- We are not interested in the distribution of *X*, but our focus is on the conditional probabilities of the class labels given *X*.

Logistic Regression (cont.)

- Given point x_i, the posterior probability for the class to be k is:

$$p_k(x_i;\theta) = Pr(G = k|X = x_i;\theta)$$

- Given the first input x_1, the posterior probability of its class, denoted as g_1, is computed by: $Pr(G = g_1|X = x_1)$

- Since samples in the training data are assumed independent, the posterior probability for the N sample points each have class g_i, $I = 1,2,\ldots,N$, given their inputs $x_1,x_2,\ldots,x_N$ is:

$$\prod_{i=1}^{N} Pr(G = g_i|X = x_i)$$

Logistic Regression (cont.)

- The joint conditional likelihood is the product of the conditional probabilities of the classes given every data point.

- Thus, the conditional *log-likelihood* of the class labels in the training data set becomes a summation:

$$l(\theta) = \sum_{i=1}^{N} \log Pr(G = g_i | X = x_i)$$

$$= \sum_{i=1}^{N} \log p_{g_i}(x_i; \theta)$$

Logistic Regression (cont.)

- We discuss in detail the two-class case (i.e., binary logistic regression), since the algorithms simplify considerably.

- It is convenient to code the two-class g_i via a 0/1 response y_i, where $y_i = 1$ when g_i = class 1, and $y_i = 0$ when g_i = class 2.

- Let $p_1(x; \theta) = p(x; \theta)$ and let $p_2(x; \theta) = 1 - p_1(x; \theta) = 1 - p(x; \theta)$ because the posterior probabilities of the two classes must sum up to one.

- Since $K = 2$, there is only one decision boundary between the two classes.

Logistic Regression (cont.)

- Thus, the log-likelihood function can be written as:

$$
\begin{aligned}
l(\beta) &= \sum_{i=1}^{N} \log p_{g_i}(x_i;\beta) \\
&= \sum_{i=1}^{N} [y_i \log p(x_i;\beta) + (1-y_i)\log\,(1-p(x_i;\beta))] \\
&= \sum_{i=1}^{N} \left[y_i \beta^T x_i - \log\,(1+e^{\beta^T x_i}) \right]
\end{aligned}
$$

- There are $p + 1$ parameters in β, and we assume that the vector of inputs x_i includes the constant term 1 to accommodate the intercept.

Logistic Regression (cont.)

- If we want to maximize the log-likelihood function, we set the first order partial derivatives of the function $l(\beta)$ with respect to β_{1j} to zero for all $j = 0, 1, \ldots, p$:

$$\frac{\partial l(\beta)}{\beta_{1j}} = \sum_{i=1}^{N} y_i x_{ij} - \sum_{i=1}^{N} \frac{x_{ij} e^{\beta^T x_i}}{1 + e^{\beta^T x_i}}$$

$$= \sum_{i=1}^{N} y_i x_{ij} - \sum_{i=1}^{N} p(x; \beta) x_{ij}$$

$$= \sum_{i=1}^{N} x_{ij}(y_i - p(x_i; \beta))$$

- In matrix form:

$$\frac{\partial l(\beta)}{\beta_{1j}} = \sum_{i=1}^{N} x_i(y_i - p(x_i; \beta))$$

Logistic Regression (cont.)

- To solve the set of $p + 1$ nonlinear equations, we use the Newton-Raphson algorithm.

- This algorithm requires the second-order derivatives (i.e., Hessian matrix), given as follows:

$$\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T} = -\sum_{i=1}^{N} x_i x_i^T p(x_i; \beta)(1 - p(x_i; \beta))$$

Logistic Regression (cont.)

- We now derive the Hessian matrix, where the element on the *j*th row and *n*th column is (counting from 0):

$$\frac{\partial l(\beta)}{\partial \beta_{1j} \partial \beta_{1n}}$$

$$= -\sum_{i=1}^{N} \frac{(1 + e^{\beta^T x_i}) e^{\beta^T x_i} x_{ij} x_{in} - (e^{\beta^T x_i})^2 x_{ij} x_{in}}{(1 + e^{\beta^T x_i})^2}$$

$$= -\sum_{i=1}^{N} x_{ij} x_{in} p(x_i; \beta) - x_{ij} x_{in} p(x_i; \beta)^2$$

$$= -\sum_{i=1}^{N} x_{ij} x_{in} p(x_i; \beta)(1 - p(x_i; \beta))$$

Logistic Regression (cont.)

- Starting with β^{old}, a single Newton-Raphson update is given by this matrix formula:

$$\beta^{new} = \beta^{old} - \left(\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T} \right)^{-1} \frac{\partial l(\beta)}{\partial \beta}$$

where the derivatives are evaluated at β^{old}.

- If given an old set of parameters, we update the new set of parameters by taking β^{old} minus the inverse of the Hessian matrix times the first-order derivative vector.

Healthcare Example (cont.): Relationship of Age and SBP w/ CD

Logistic Regression Table

Predictor	Coef	SE Coef	Z	P	Odds Ratio	95% CI Lower	95% CI Upper
Constant	-10.8551	4.06665	-2.67	0.008			
Age	0.112926	0.0584885	1.93	0.054	1.12	1.00	1.26
SBP	0.0373089	0.0267671	1.39	0.163	1.04	0.98	1.09

Log-Likelihood = -13.232
Test that all slopes are zero: G = 18.524, DF = 2, P-Value = 0.000

Interpretation:

- A unit increase in Age (with SBP held constant) increases the log-odds of CD by 0.113.
- In other words, a unit increase in Age increases the odds of getting CD by a factor of 1.12 (i.e., 12%)

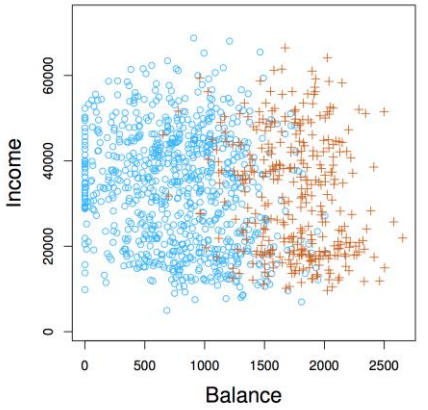
Regression Equation

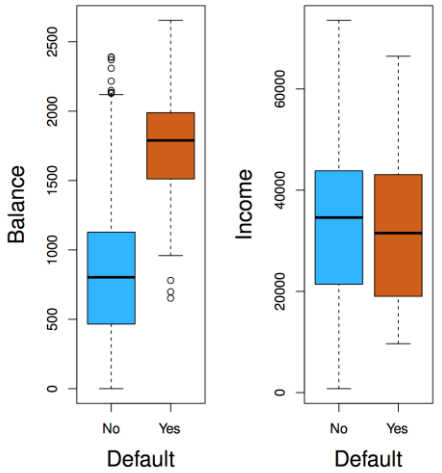
$$P(1) = \exp(Y')/(1 + \exp(Y'))$$

$$Y' = -10.86 + 0.1129\ Age + 0.0373\ SBP$$

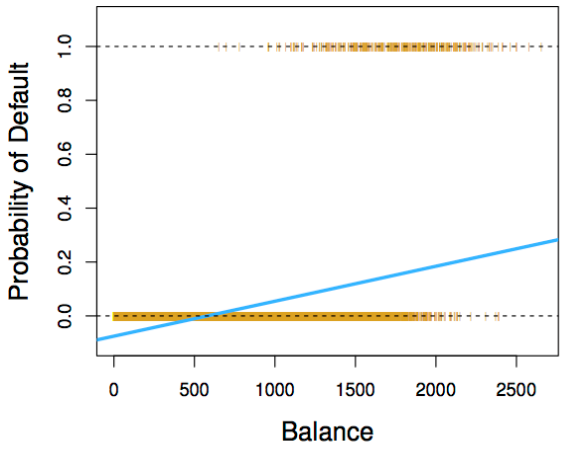
Logistic Regression: Credit Card Default Example

- We would like to be able to predict customers that are likely to default on their credit card.
- Possible *X* variables:
  - Annual Income
  - Monthly Credit Card Balance
- The *Y* variable (Default) is categorical: Yes or No

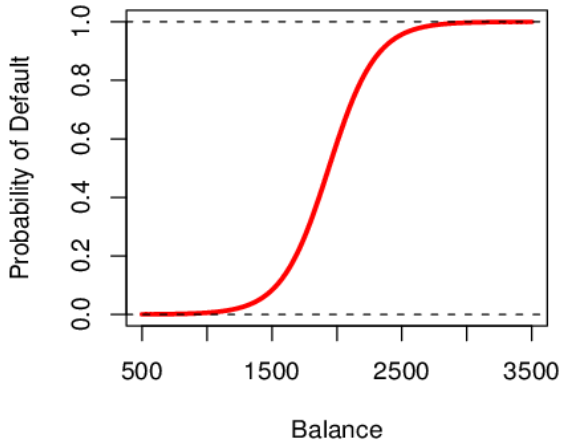




Logistic Regression: Credit Card Default Example (cont.)



- If we fit a linear regression model to the Default data, then:
  - For very low balances we predict a negative probability!
  - For high balances we predict a probability above 1!



- If we fit a logistic regression model, then the probability of default is between 0 and 1 for all balances.

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} \quad \log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$$

Logistic Regression: Credit Card Default Example (cont.)

- Interpreting what β_1 means is not very easy with logistic regression, simply because we are predicting Pr(Y) and not Y.

- In a logistic regression model, increasing X by one unit changes the *log odds* by β_1, or equivalently it multiplies the odds by e^{β_1}.

- If $\beta_1 = 0$, then there is no relationship between Y and X. If $\beta_1 > 0$, then when X gets larger so does the probability that $Y = 1$. If $\beta_1 < 0$, then when X gets larger the probability that $Y = 1$ gets smaller.

- How much increase or decrease in probability depends on the slope.

Logistic Regression: Credit Card Default Example (cont.)

- We still want to perform a hypothesis test to see whether we can be sure that β_0 and β_1 are significantly different from zero.
- We use a Z test and interpret the p-value as usual.
- In this example, the p-value for balance is very small and $\hat{\beta}_1$ is positive. This means that if the balance increases, then the probability of default will increase as well.

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.6513	0.3612	-29.5	< 0.0001
balance	0.0055	0.0002	24.9	< 0.0001

Logistic Regression: Credit Card Default Example (cont.)

- What is the estimated probability of default for someone with a balance of $1000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1000}}{1+e^{-10.6513+0.0055\times 1000}} = 0.006$$

- What is the estimated probability of default for someone with a balance of $2000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 2000}}{1+e^{-10.6513+0.0055\times 2000}} = 0.586$$

Logistic Regression: Credit Card Default Example (cont.)

- We can also use student (0,1) as a predictor to estimate the probability that an individual defaults:

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-3.5041	0.0707	-49.55	< 0.0001
student[Yes]	0.4049	0.1150	3.52	0.0004

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=Yes}) = \frac{e^{-3.5041+0.4049\times 1}}{1+e^{-3.5041+0.4049\times 1}} = 0.0431,$$

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=No}) = \frac{e^{-3.5041+0.4049\times 0}}{1+e^{-3.5041+0.4049\times 0}} = 0.0292.$$

Multiple Logistic Regression

- More than one independent variable
  - Dichotomous, ordinal, nominal, continuous …

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

- Interpretation of β_i
  - Increase in log-odds for a one unit increase in x_i with all the other x_is constant
  - Measures association between x_i and log-odds adjusted for all other x_i

L_1 Regularized Logistic Regression

- For logistic regression, we would maximize a penalized version of the log-likelihood function:

$$\max_{\beta_0,\beta} \left\{ \sum_{i=1}^{N} \left[y_i(\beta_0 + \beta^T x_i) - \log(1 + e^{\beta_0+\beta^T x_i}) \right] - \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

- As with the lasso method, we typically do not penalize the intercept term, and we standardize the predictors for the penalty to be meaningful.

- Because this optimization criterion is concave, a solution can be found using nonlinear programming methods or via quadratic approximations as used in the Newton-Raphson algorithm.

Weighted Least Squares Regression / Generalized Least Squares Regression

What is Heteroskedasticity

- Recall the assumption of homoskedasticity implied that conditional on the explanatory variables, the variance of the unobserved error, *u*, was constant.

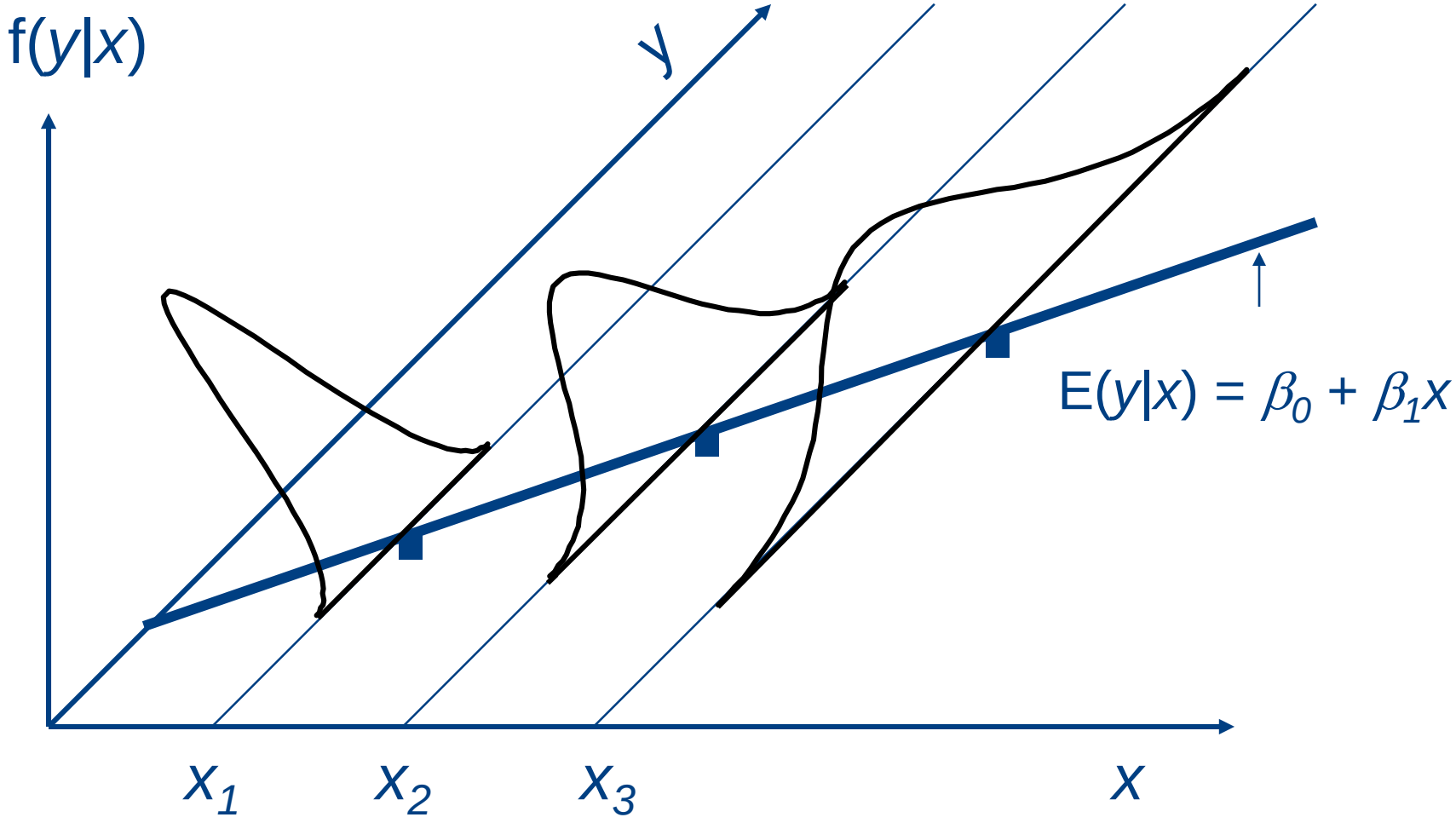
- If this is not true, that is if the variance of *u* is different for different values of the *x*'s, then the errors are **heteroskedastic**.

- For example, consider a regression of housing expenditures (i.e., rent) on income.

$$Rent_i = \beta_0 + \beta_1 Income_i + \varepsilon_i$$

- Consumers with low values of income have little scope for varying their rent expenditures. $Var(\varepsilon_i)$ is low. Wealthy consumers can choose to spend a lot of money on rent, or to spend less, depending on tastes. $Var(\varepsilon_i)$ is high.

Example of Heteroskedasticity



Why Worry About Heteroskedasticity?

- OLS is still unbiased and consistent, even if we do not assume homoscedasticity.
- However, OLS is no longer efficient; some other linear estimator will have a lower variance.
- The standard errors of the estimates are biased if we have heteroscedasticity.
- If the standard errors are biased, we can not use the usual *t* statistics or *F* statistics for drawing inferences (i.e., the CIs and hypothesis tests will be incorrect).

Variance with Heteroskedasticity

- If the errors contain heteroscedasticity, then $Var(u_i|x_i) = \sigma_i^2 \neq \sigma^2$

For simple linear regression, $\hat{\beta}_1 = \beta_1 + \dfrac{\sum (x_i - \bar{x}) u_i}{\sum (x_i - \bar{x})^2}$, so

$$Var(\hat{\beta}_1) = \frac{\sum (x_i - \bar{x})^2 \sigma_i^2}{SST_x^2}, \text{ where } SST_x = \sum (x_i - \bar{x})^2$$

A valid estimator for this when $\sigma_i^2 \neq \sigma^2$ is

$$\frac{\sum (x_i - \bar{x})^2 \hat{u}_i^2}{SST_x^2}, \text{ where } \hat{u}_i \text{ are are the OLS residuals}$$

Variance with Heteroskedasticity

For the general multiple regression model, a valid estimator of $Var\left(\hat{\beta}_j\right)$ with heteroskedasticity is

$\hat{Var}\left(\hat{\beta}_j\right) = \frac{\sum \hat{r}_{ij}\hat{u}_i^2}{SST_j^2}$, where $\hat{r}_{ij}$ is the i^{th} residual from regressing x_j on all other independent variables, and SST_j is the sum of squared residuals from this regression

Robust Standard Errors

- Now that we have a consistent estimate of the variance, the square root can be used as a standard error for inference.
- Typically call these **robust standard errors**.
- Sometimes the estimated variance is corrected for degrees of freedom by multiplying by $n/(n-k-1)$.
- As $n \to \infty$ it's all the same, though.

Robust Standard Errors (cont.)

- It's important to remember that these robust standard errors only have asymptotic justification – with small sample sizes *t* statistics formed with robust standard errors will not have a distribution close to the *t*, and inferences will not be correct.

- For calculating robust standard errors in R, look at the sandwich package.

Testing for Heteroskedasticity

- We want to test the null hypothesis of homoscedasticity:
  - H_0: $\text{Var}(u|x_1, x_2, \ldots, x_k) = \sigma^2$, which is equivalent to H_0: $E(u^2|x_1, x_2, \ldots, x_k) = E(u^2) = \sigma^2$

- If we assume the relationship between u^2 and x_j is linear, we can test as a linear restriction.

- So, for $u^2 = \delta_0 + \delta_1 x_1 + \ldots + \delta_k x_k + v$ this means testing H_0: $\delta_1 = \delta_2 = \ldots = \delta_k = 0$.

The Breusch-Pagan Test

- Regress Y against your predictors using OLS.
- Compute the OLS residuals.
- Regress the residuals squared on all predictors.
- Compute the R^2 from this auxiliary regression to form an F test.
- The F statistic is just the reported F statistic for overall significance of the regression, $F = [R^2/k]/[(1 - R^2)/(n - k - 1)]$, which is distributed $F_{k,\, n-k-1}$
- If the F statistic > F critical value (or p-value < 0.05), reject H_0 (i.e., there is heteroscedasticity).

The White Test

- Regress Y against your predictors using OLS.
- Compute the OLS residuals.
- Regress the residuals squared on all predictors, the squares of all predictors, and all possible interactions b/t predictors.
- Compute the R^2 from this auxiliary regression to form an F test.
- The F statistic is just the reported F statistic for overall significance of the regression, $F = [R^2/k]/[(1 - R^2)/(n - k - 1)]$, which is distributed $F_{k,\, n-k-1}$
- If the F statistic > F critical value (or p-value < 0.05), reject H_0 (i.e., there is heteroscedasticity).

Alternate Form of the White Test

- Consider that the fitted values from OLS, $\hat{y}$, are a function of all the predictors.
- Thus, $\hat{y}^2$ will be a function of the squares and interaction terms, and $\hat{y}$ and $\hat{y}^2$ can proxy for all of the x_j, x_j^2, and $x_j x_h$.
- Regress the residuals squared on $\hat{y}$ and $\hat{y}^2$ and use the R^2 to form an F statistic.

Weighted Least Squares

- While it's always possible to estimate robust standard errors for OLS estimates, if we know something about the specific form of the heteroskedasticity, we can obtain more efficient estimates than OLS.

- The basic idea is to transform the model into one that has homoskedastic errors – called **weighted least squares**.

Weighted Least Squares (cont.)

- Suppose the heteroskedasticity can be modeled as Var($u|\boldsymbol{x}$) = $\sigma^2 h(\boldsymbol{x})$, where $h(\boldsymbol{x})$ is some function of the predictors that determines the heteroscedasticity.

- The trick is to figure out what $h(\boldsymbol{x}) \equiv h_i$ looks like, where $Var(u_i|\boldsymbol{x}_i) = \sigma_i^2 = \sigma^2 h(\boldsymbol{x}_i) = \sigma^2 h_i$.

- Since h_i is just a function of $\boldsymbol{x}_i$, then $E\left(u_i/\sqrt{h_i}\middle|\boldsymbol{x}_i\right) = 0$. Since $Var(u_i|\boldsymbol{x}_i) = E\left(u_i^2\middle|\boldsymbol{x}_i\right) = \sigma^2 h_i$, then $Var\left(u_i/\sqrt{h_i}\middle|\boldsymbol{x}_i\right) = \sigma^2$

- So, if we divide the whole equation by $\sqrt{h_i}$ we would have a model where the error is homoscedastic.

Generalized Least Squares

- Estimating the transformed equation by OLS is an example of **generalized least squares (GLS)**.
- GLS will be the best linear unbiased estimator (BLUE) in this case of accounting for heteroskedastic errors (i.e., GLS estimates are more efficient than OLS).
- GLS is a weighted least squares (WLS) procedure where each squared residual is weighted by the inverse of $\text{Var}(u_i|\boldsymbol{x}_i)$.
- The idea is that less weight is given to observations with a higher error variance.

Weighted Least Squares (cont.)

- While it is intuitive to see why performing OLS on a transformed equation is appropriate, it can be tedious to do the transformation.
- Weighted least squares is a way of getting the same thing, without the transformation.
- Idea is to minimize the weighted sum of squared residuals (weighted by $1/h_i$). That is, we specify weights proportional to the inverse of the variance.
- WLS is great if we know what $\text{Var}(u_i|\boldsymbol{x}_i)$ looks like. In most cases, however, we won't know the form of heteroscedasticity.

Feasible GLS

- In this case, we need to estimate $h(\boldsymbol{x}_i)$. Using $\widehat{h_i}$ instead of h_i in the GLS transformation yields the **feasible GLS (FGLS) estimator**.

- The FGLS estimator is not unbiased, but it is consistent and asymptotically more efficient than OLS.

- We start with the assumption of a fairly flexible model, such as:
  - $\text{Var}(u|\boldsymbol{x}) = \sigma^2 \exp(\delta_0 + \delta_1 x_1 + \ldots + \delta_k x_k)$
  - Thus, $h(\boldsymbol{x}) = \exp(\delta_0 + \delta_1 x_1 + \ldots + \delta_k x_k)$

- Since we don't know δ, we estimate using the data. Then, we use these estimates to construct weights.

Feasible GLS (cont.)

- Our assumption implies that $u^2 = \sigma^2 \exp(\delta_0 + \delta_1 x_1 + \ldots + \delta_k x_k) v$, where $E(v|\boldsymbol{x}) = 1$.

- If $E(v) = 1$, then $\ln(u^2) = \alpha_0 + \delta_1 x_1 + \ldots + \delta_k x_k + e$, where $E(e) = 0$ and e is independent of $\boldsymbol{x}$

- Now, we know that $\hat{u}$ is an estimate of u, so we can estimate this by OLS (i.e., regress $\ln(\hat{u}^2)$ on the predictors).

- An estimate of h_i is obtained as $\widehat{h_i} = \exp(\widehat{g_i})$, where $\widehat{g_i}$ are the fitted values from regressing $\ln(\hat{u}^2)$ on all of the predictors. We now use WLS with weights $1/\widehat{h_i}$.

Feasible GLS (cont.)

Steps: FGLS to Correct for Heteroskedasticity

1. Regress *y* on the predictors and obtain the residuals, $\hat{u}$
2. Create $\ln(\hat{u}^2)$ by first squaring the $\hat{u}$ and taking the ln
3. Regress $\ln(\hat{u}^2)$ on the predictors and obtain the fitted values, $\hat{g}_i$
4. Exponentiate the fitted values: $\hat{h}_i = \exp(\hat{g}_i)$
5. Estimate the original equation by WLS using weights $1/\hat{h}_i$
   - Note: the squared residual for observation *i* gets weighted by $1/\hat{h}_i$. If instead we first transform all variables and run OLS, then each variable gets multiplied by $1/\sqrt{\hat{h}_i}$