Day 2: Rethinking regression as a predictive tool

Kenneth Benoit

Data Mining and Statistical Learning

February 25, 2015

Classification and prediction as goals

- Machine learning focuses on identifying classes (classification), while social science is typically interested"
  - estimating marginal effects
  - measuring things (latent trait scaling)
- Regression analysis is the workhorse of social science statistical analysis, but can also be used to predict out of sample
- "Statistical learning" view is that regression is a "supervised" machine learning method for continuously-valued outcomes

Supervised v. unsupervised methods compared

- Two different approaches:
  - *Supervised methods* require a training set that exmplify constrasting classes, identified by the researcher
  - *Unsupervised methods* scale differences and identify patterns, without requiring a training step
- Relative advantage of supervised methods: You set the input dimensions
- Relative disadvantage of supervised methods:
  You need to “know” in advance the dimensions being scaled, in order to train or fit the model

Supervised v. unsupervised methods: Examples

- General examples:
  - Supervised: Regression, logistic regression, Naive Bayes, k-Nearest Neighbor, Support Vector Machines (SVM)
  - Unsupervised: correspondence analysis, IRT models, factor analytic approaches
- Lots of applications in text analysis
  - Supervised: Wordscores (LBG 2003); SVMs (Yu, Kaufman and Diermeier 2008); Naive Bayes (Evans et al 2007)
  - Unsupervised "Wordfish" (Slapin and Proksch 2008); Correspondence analysis (Schonhardt-Bailey 2008); two-dimensional IRT (Monroe and Maeda 2004)

How do we get "true" condition?

- For regression examples: We have a sample with a continuously-valued dependent variable
- In some domains: through more expensive or extensive tests
- May also be through expert annotation or coding
  - A scheme should be tested and reported for its reliability

Generalization and overfitting

- Generalization: A classifier or a regression algorithm learns to correctly predict output from given inputs not only in previously seen samples but also in previously unseen samples
- Overfitting: A classifier or a regression algorithm learns to correctly predict output from given inputs in previously seen samples but fails to do so in previously unseen samples. This causes poor prediction/generalization

How model fit is evaluated

- For discretely-valued outcomes (class prediction): Goal is to maximize the frontier of precise identification of true condition with accurate recall, defined in terms of false positives and false negatives
  - will define formally later
- For continuously-valued outcomes: minimize Root Mean Squared Error (RMSE)

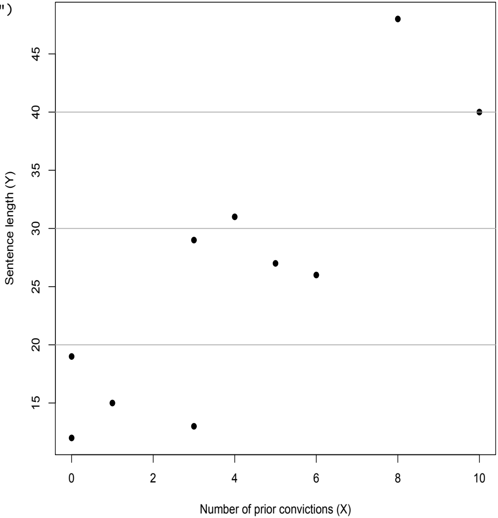
Regression as a prediction method

- Training step: fitting a model to data for which the outcome variable Y_i is known
- Test step: predicting out of sample Y_i for a new configuration of data input values $\mathbf{X}_i$
- Evaluation: based on RMSE, or average

$$\sqrt{\sum (Y_i - \hat{Y}_i)}$$

Example

```
par(mar=c(4,4,1,1))
x <- c(0,3,1,0,6,5,3,4,10,8)
y <- c(12,13,15,19,26,27,29,31,40,48)
plot(x, y, xlab="Number of prior convictions (X)",
     ylab="Sentence length (Y)", pch=19)
abline(h=c(10,20,30,40), col="grey70")
```



Least squares formulas

For the three parameters (simple regression):

- the regression coefficient:

$$\hat{\beta}_1 = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

- the intercept:

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

- and the residual variance σ^2:

$$\hat{\sigma}^2 = \frac{1}{n-2} \sum [y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)]^2$$

Least squares formulas continued

Things to note:

- the prediction line is $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$
- the value $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ is the predicted value for x_i
- the residual is $e_i = y_i - \hat{y}_i$
- The residual sum of squares (RSS) $= \sum_i e_i^2$
- The estimate for σ^2 is the same as

$$\hat{\sigma}^2 = \mathrm{RSS}/(n-2)$$

Example to show fomulas in R

```
> x <- c(0,3,1,0,6,5,3,4,10,8)
> y <- c(12,13,15,19,26,27,29,31,40,48)
> (data <- data.frame(x, y, xdev=(x-mean(x)), ydev=(y-mean(y)),
+                     xdevydev=((x-mean(x))*(y-mean(y))),
+                     xdev2=(x-mean(x))^2,
+                     ydev2=(y-mean(y))^2))
    x  y xdev ydev xdevydev xdev2 ydev2
1   0 12   -4  -14       56    16   196
2   3 13   -1  -13       13     1   169
3   1 15   -3  -11       33     9   121
4   0 19   -4   -7       28    16    49
5   6 26    2    0        0     4     0
6   5 27    1    1        1     1     1
7   3 29   -1    3       -3     1     9
8   4 31    0    5        0     0    25
9  10 40    6   14       84    36   196
10  8 48    4   22       88    16   484
> (SP <- sum(data$xdevydev))
[1] 300
> (SSx <- sum(data$xdev2))
[1] 100
> (SSy <- sum(data$ydev2))
[1] 1250
> (b1 <- SP / SSx)
[1] 3
> (b0 <- mean(y) - b1*mean(x))
```

From observed to "predicted" relationship

- In the above example, $\hat{\beta}_0 = 14$, $\hat{\beta}_1 = 3$
- This linear equation forms the regression line
- The regression line always passes through two points:
  - the point $(x = 0, y = \hat{\beta}_0)$
  - the point $(\bar{x}, \bar{y})$ (the average X predicts the average Y)
- The residual sum of squares (RSS) $= \sum_i e_i^2$
- The regression line is that which minimizes the RSS

Ordinary Least Squares (OLS)

- Objective: minimize $\sum e_i^2 = \sum (Y_i - \hat{Y}_i)^2$, where
  - $\hat{Y}_i = b_0 + b_1 X_i$
  - error $e_i = (Y_i - \hat{Y}_i)$

$$
\begin{aligned}
b &= \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sum (X_i - \bar{X})} \\
&= \frac{\sum X_i Y_i}{\sum X_i^2}
\end{aligned}
$$

- The intercept is: $b_0 = \bar{Y} - b_1 \bar{X}$

OLS rationale

- Formulas are very simple
- Closely related to ANOVA (sums of squares decomposition)
- Predicted Y is sample mean when $\Pr(Y|X) = \Pr(Y)$
  - In the special case where Y has no relation to X, $b_1 = 0$, then OLS fit is simply $\hat{Y} = b_0$
  - Why? Because $b_0 = \bar{Y} - b_1\bar{X}$, so $\hat{Y} = \bar{Y}$
  - Prediction is then sample mean when X is unrelated to Y
- Since OLS is then an extension of the sample mean, it has the same attractice properties (efficiency and lack of bias)
- Alternatives exist but OLS has generally the best properties when assumptions are met

OLS in matrix notation

- Formula for coefficient β:

$$\begin{aligned} Y &= X\beta + \epsilon \\ X'Y &= X'X\beta + X'\epsilon \\ X'Y &= X'X\beta + 0 \\ (X'X)^{-1}X'Y &= \beta + 0 \\ \beta &= (X'X)^{-1}X'Y \end{aligned}$$

- Formula for variance-covariance matrix: $\sigma^2(X'X)^{-1}$
  - In simple case where $y = \beta_0 + \beta_1 * x$, this gives $\sigma^2 / \sum(x_i - \bar{x})^2$ for the variance of β_1
  - Note how increasing the variation in X will reduce the variance of β_1

The "hat" matrix

- The hat matrix H is defined as:

$$\begin{aligned} \hat{\beta} &= (X'X)^{-1}X'y \\ X\hat{\beta} &= X(X'X)^{-1}X'y \\ \hat{y} &= Hy \end{aligned}$$

- $H = X(X'X)^{-1}X'$ is called the *hat-matrix*
- Other important quantities, such as $\hat{y}$, $\sum e_i^2$ (RSS) can be expressed as functions of H
- Corrections for heteroskedastic errors ("robust" standard errors) involve manipulating H

Some important OLS properties to understand

Applies to $y = \alpha + \beta x + \epsilon$

- If $\beta = 0$ and the only regressor is the intercept, then this is the same as regressing y on a column of ones, and hence $\alpha = \bar{y}$ — the mean of the observations
- If $\alpha = 0$ so that there is no intercept and one explanatory variable x, then $\beta = \frac{\sum xy}{\sum x^2}$
- If there is an intercept and one explanatory variable, then

$$\begin{aligned} \beta &= \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} \\ &= \frac{\sum_i (x_i - \bar{x}) y_i}{\sum (x_i - \bar{x})^2} \end{aligned}$$

Some important OLS properties (cont.)

- If the observations are expressed as deviations from their means, $y* = y - \bar{y}$ and $x^* = x - \bar{x}$, then $\beta = \sum x^* y^* / \sum x^{*2}$
- The intercept can be estimated as $\bar{y} - \beta \bar{x}$. This implies that the intercept is estimated by the value that causes the sum of the OLS residuals to equal zero.
- The mean of the $\hat{y}$ values equals the mean y values – together with previous properties, implies that the OLS regression line passes through the overall mean of the data points

Normally distributed errors

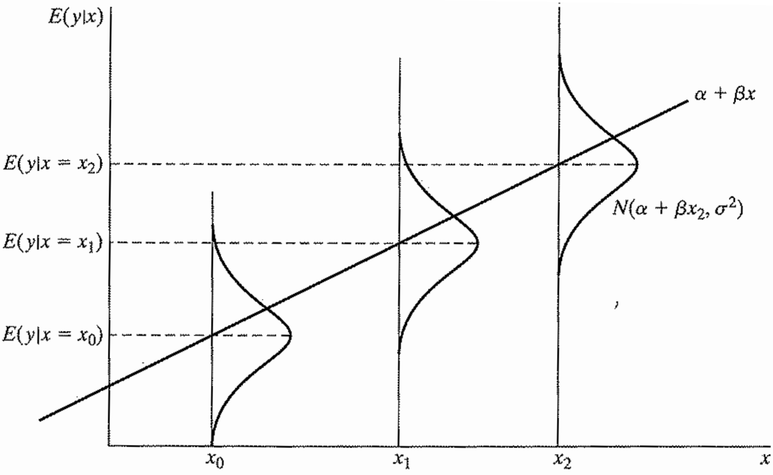


FIGURE 2.2 The Classical Regression Model.

OLS in R

```
> dail <- read.dta("dail2002.dta")
> mdl <- lm(votes1st ~ spend_total*incumb + minister, data=dail)
> summary(mdl)

Call:
lm(formula = votes1st ~ spend_total * incumb + minister, data = dail)

Residuals:
    Min      1Q  Median      3Q     Max
-5555.8  -979.2  -262.4   877.2  6816.5

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         469.37438  161.54635   2.906  0.00384 **
spend_total           0.20336    0.01148  17.713  < 2e-16 ***
incumb             5150.75818  536.36856   9.603  < 2e-16 ***
minister           1260.00137  474.96610   2.653  0.00826 **
spend_total:incumb   -0.14904    0.02746  -5.428 9.28e-08 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 1796 on 457 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared: 0.6672,    Adjusted R-squared: 0.6643
F-statistic:   229 on 4 and 457 DF,  p-value: < 2.2e-16
```

OLS in Stata

```
. use dail2002
(Ireland 2002 Dail Election - Candidate Spending Data)

. gen spendXinc = spend_total * incumb
(2 missing values generated)

. reg votes1st spend_total incumb minister spendXinc

      Source |       SS       df       MS              Number of obs =     462
-------------+------------------------------           F(  4,   457) =  229.05
       Model |  2.9549e+09     4   738728297           Prob > F      =  0.0000
    Residual |  1.4739e+09   457  3225201.58           R-squared     =  0.6672
-------------+------------------------------           Adj R-squared =  0.6643
       Total |  4.4288e+09   461  9607007.17           Root MSE      =  1795.9

------------------------------------------------------------------------------
    votes1st |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 spend_total |   .2033637   .0114807    17.71   0.000     .1808021    .2259252
      incumb |   5150.758   536.3686     9.60   0.000     4096.704    6204.813
    minister |   1260.001   474.9661     2.65   0.008      326.613     2193.39
   spendXinc |  -.1490399   .0274584    -5.43   0.000    -.2030003   -.0950794
       _cons |   469.3744   161.5464     2.91   0.004     151.9086    786.8402
------------------------------------------------------------------------------
```

Sums of squares (ANOVA)

TSS Total sum of squares $\sum(y_i - \bar{y})^2$

ESS Estimation or Regression sum of squares $\sum(\hat{y}_i - \bar{y})^2$

RSS Residual sum of squares $\sum e_i^2 = \sum(\hat{y}_i - y_i)^2$

The key to remember is that TSS = ESS + RSS

Examining the sums of squares

```
> yhat <- mdl$fitted.values   # uses the lm object mdl from previous
> ybar <- mean(mdl$model[,1])
> y <- mdl$model[,1]          # can't use dail$votes1st since diff N
> TSS <- sum((y-ybar)^2)
> ESS <- sum((yhat-ybar)^2)
> RSS <- sum((yhat-y)^2)
> RSS
[1] 1473917120
> sum(mdl$residuals^2)
[1] 1473917120
> (r2 <- ESS/TSS)
[1] 0.6671995
> (adjr2 <- (1 - (1-r2)*(462-1)/(462-4-1)))
[1] 0.6642865
> summary(mdl)$r.squared      # note the call to summary()
[1] 0.6671995
> RSS/457
[1] 3225202
> sqrt(RSS/457)
[1] 1795.885
> summary(mdl)$sigma
[1] 1795.885
```

Regression model return values

Here we will talk about the quantities returned with the `lm()` command and `lm` class objects.

Table 10.1 Extractor functions for the result of `lm()`

`summary()`	returns summary information about the regression
`plot()`	makes diagnostic plots
`coef()`	returns the coefficients
`residuals()`	returns the residuals (can be abbreviated `resid()`)
`fitted()`	returns fitted values, $\widehat{y}_i$
`deviance()`	returns RSS
`predict()`	performs predictions
`anova()`	finds various sums of squares
`AIC()`	is used for model selection

Uncertainty in regression models: the linear case revisited

- Suppose we regress y on X to produce $b = (X'X)^{-1}X'y$
- Then we set explanatory variables to new values X^p to predict Y^p
- The prediction Y^p will have two forms of uncertainty:
  1. estimation uncertainty that can be reduced by increasing the sample size. Estimated a $\hat{y}^p = X^p b$ and depends on sample size through b
  2. fundamental variability comes from variability in the dependent variable around the expected value $\mathrm{E}(Y^p) = \mu = X^p\beta$ – even if we knew the true β

Estimation uncertainty and fundamental variability

- We can decompose this as follows:

$$\begin{aligned}
Y^p &= X^p b + \epsilon^p \\
\mathrm{Var}(Y^p) &= \mathrm{Var}(X^p b) + \mathrm{Var}(\epsilon^p) \\
&= X^p \mathrm{Var}(b)(X^p)' + \sigma^2 I \\
&= \sigma^2 X^p ((X^p)' X^p)^{-1} + \sigma^2 I \\
&= \text{estimation uncertainty} + \text{fundamental variability}
\end{aligned}$$

- It can be shown that the distribution of $\hat{Y}^p$ is:

$$\hat{Y}^p \sim N(X^p\beta, X^p \mathrm{Var}(b)(X^p)')$$

- and that the unconditional distribution of Y^p is:

$$Y^p \sim N(X^p\beta,\ X^p\,\mathrm{Var}(b)(X^p)' + \sigma^2 I)$$

Confidence intervals for predictions

- For any set of explanatory variables x_0, the predicted response is $\hat{y}_0 = x_0'\hat{\beta}$
- But this prediction also comes with uncertainty, and by extension, with a confidence interval
- Two types:
  - *predictions of future observations*: based on the prediction plus the variance of ϵ (Note: this is what we usually want)

$$\hat{y}_0 \pm t_{n-k-1}^{\alpha/2} \hat{\sigma} \sqrt{1 + x_o'(X'X)^{-1} x_0}$$

  - *prediction of mean response*: the average value of a y_0 with the characteristics x_0 – only takes into account the variance of $\hat{\beta}$

$$\hat{y}_0 \pm t_{n-k-1}^{\alpha/2} \hat{\sigma} \sqrt{x_0'(X'X)^{-1} x_0}$$

Confidence intervals for predictions in R

```
> summary(m1)$coeff
                       Estimate   Std. Error   t value     Pr(>|t|)
(Intercept)         464.5955332 162.59752848  2.857335 4.466694e-03
spend_total           0.2041449   0.01155236 17.671273 1.154515e-53
incumb             4493.3251289 478.80828470  9.384393 2.962201e-19
spend_total:incumb   -0.1068943   0.02254283 -4.741832 2.832798e-06
> fivenum(dail$spend_total)    # what is typical spending profile
[1]     0.00  5927.32 14699.12 20812.66 51971.28
> x0 <- c(1, 75000, 1, 75000)  # set some predictor values
> (y0 <- sum(x0*coef(m1)))     # compute predicted response
[1] 12251.71
> fivenum(dail$votes1st)       # how typical is this response?
[1]    19.0  1151.5  3732.0  6432.0 14742.0
> quantile(dail$votes1st, .99, na.rm=T)  # versus 99th percentile
     99%
11138.44
> x0.df <- data.frame(incumb=1, spend_total=75000)
> predict(m1, x0.df)
       1
12251.71
> predict(m1, x0.df, interval="confidence")
       fit      lwr      upr
1 12251.71 10207.33 14296.09
> predict(m1, x0.df, interval="prediction")
       fit      lwr      upr
1 12251.71 8153.068 16350.36
```

Fundamental and estimation variability for non-linear forms

- For well-known cases, we known both the expectation and the fundamental variability, e.g.
  - *Poisson* $E(Y) = e^{X\beta}$, $\text{Var}(Y) = \lambda$
  - *logistic* $E(Y) = \frac{1}{1+e^{-X\beta}}$, $\text{Var}(Y) = \pi(1-\pi)$
- Calculating the estimation variability is harder, but can be done using a linear approximation from the Taylor series. The Taylor series approximation of $\hat{y}^p = g(b)$ is:

  $$\hat{y}^p = g(b) = g(\beta) + g'(\beta)(b - \beta) + \cdots$$

  where $g'(\beta)$ is the first derivative of the functional form $g(\beta)$ with respect to β
- If we drop all but the first two terms, then

  $$\begin{aligned} \text{Var}(\hat{Y}^p) &\approx \text{Var}[g(\beta)] + \text{Var}[g'(\beta)(b-\beta)] \\ &= g'(\beta)\text{Var}(b)g'(\beta)' \end{aligned}$$

- This is known as the Delta method for calculating standard errors of predictions

Precision and recall

- Illustration framework

		True condition	
		Positive	Negative
Prediction	Positive	True Positive	False Positive (Type I error)
	Negative	False Negative (Type II error)	True Negative

Precision and recall and related statistics

- Precision: $\frac{\text{true positives}}{\text{true positives} + \text{false positives}}$
- Recall: $\frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$
- Accuracy: $\frac{\text{Correctly classified}}{\text{Total number of cases}}$
- $F1 = 2\,\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
  (the harmonic mean of precision and recall)

Example: Computing precision/recall

Assume:

- We have a sample in which 80 outcomes are really positive (as opposed to negative, as in sentiment)
- Our method declares that 60 are positive
- Of the 60 declared positive, 45 are actually positive

Solution:

$$\text{Precision} = (45/(45+15)) = 45/60 = 0.75$$
$$\text{Recall} = (45/(45+35)) = 45/80 = 0.56$$

Accuracy?

		True condition		
		Positive	Negative	
Prediction	Positive	45		60
	Negative			
		80		

add in the cells we can compute

		True condition		
		Positive	Negative	
Prediction	Positive	45	*15*	60
	Negative	*35*		
		80		

but need True Negatives and N to compute accuracy

		True condition		
		Positive	Negative	
Prediction	Positive	45	*15*	60
	Negative	*35*	**???**	
		80		

assume 10 True Negatives:

		True condition	
		Positive	Negative	
Prediction	Positive	45	*15*	60
	Negative	*35*	***10***	***45***
		80	***25***	***105***

$$\text{Accuracy} = (45 + 10)/105 \qquad = 0.52$$

$$\text{F1} = 2 * (0.75 * 0.56)/(0.75 + 0.56) \qquad = 0.64$$

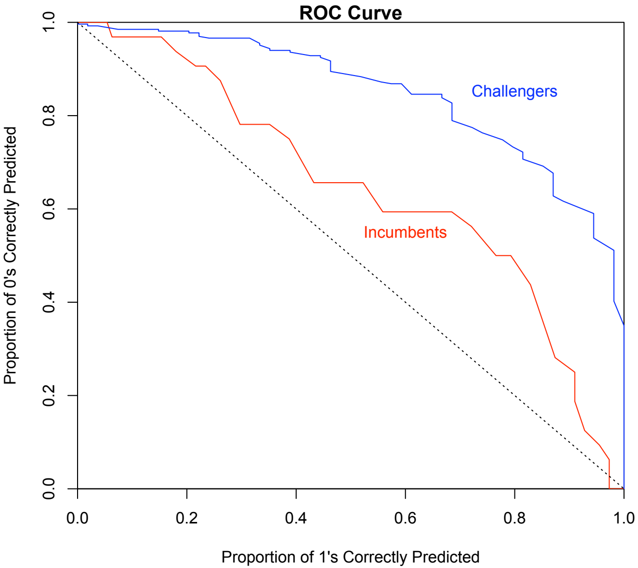
now assume 100 True Negatives:

		True condition		
		Positive	Negative	
Prediction	Positive	45	*15*	60
	Negative	*35*	***100***	***135***
		80	***115***	***195***

$$\text{Accuracy} = (45 + 100)/195 \qquad = 0.74$$

$$\text{F1} = 2 * (0.75 * 0.56)/(0.75 + 0.56) \qquad = 0.64$$

Receiver Operating Characteristic (ROC) plot



Estimating uncertainty through simulation

- King, Timz, and Wittenberg (2000) propose using statistical simulation to estimate uncertainty
- Notation:

  stochastic component $Y_i \sim f(\theta_i, \alpha)$
  systmatic component $\theta_i = g(X_i, \beta)$
  For example in a linear-normal model,
  $Y_i = N(\mu_i, \sigma^2)$ and $\mu_i = X_i\beta$
  simulated parameter vector $\hat{\gamma} = \text{vec}(\hat{\beta}, \hat{\alpha})$

  The central limit theorem tells us we can simulate γ as

$$\tilde{\gamma} \sim \mathrm{N}(\hat{\gamma}, \hat{V}(\hat{\gamma}))$$

Simulating predicted values

1. Using the algorithm in the previous subsection, draw one value of the vector $\tilde{\gamma} = \mathrm{vec}(\tilde{\beta}, \tilde{\alpha})$.
2. Decide which kind of predicted value you wish to compute, and on that basis choose one value for each explanatory variable. Denote the vector of such values X_c.
3. Taking the simulated effect coefficients from the top portion of $\tilde{\gamma}$, compute $\tilde{\theta}_c = g(X_c, \tilde{\beta})$, where $g(\cdot,\cdot)$ is the systematic component of the statistical model.
4. Simulate the outcome variable $\tilde{Y}_c$ by taking a random draw from $f(\tilde{\theta}_c, \tilde{\alpha})$, the stochastic component of the statistical model.

Repeat this $M = 1000$ times to approximate the entire probability distribution of Y_c. Using this estimated distribution we can compute mean and SDs which will approximate the predicted values and their error.

Simulating expected values

1. Following the procedure for simulating the parameters, draw one value of the vector $\tilde{\gamma} = \text{vec}(\tilde{\beta}, \tilde{\alpha})$.
2. Choose one value for each explanatory variable and denote the vector of values as X_c.
3. Taking the simulated effect coefficients from the top portion of $\tilde{\gamma}$, compute $\tilde{\theta}_c = g(X_c, \tilde{\beta})$, where $g(\cdot,\cdot)$ is the systematic component of the statistical model.
4. Draw m values of the outcome variable $\tilde{Y}_c^{(k)}$ ($k = 1, \ldots, m$) from the stochastic component $f(\tilde{\theta}_c, \tilde{\alpha})$. This step simulates fundamental uncertainty.
5. Average over the fundamental uncertainty by calculating the the mean of the m simulations to yield one simulated expected value $\tilde{E}(Y_c) = \sum_{k=1}^{m} \tilde{Y}_c^{(k)} \Big/ m$.

Note: It is m that approximates the fundamental variability but Step 5 averages it away. A large enough m will purge the simulated result of any fundamental uncertainty.

Repeat the entire process $M = 1000$ times to estimate the full probability distribution of $E(Y_c)$.

Calculating standard errors in Zelig

```
## Examples from titanic data
titanic <- read.dta("titanic.dta")
levels(titanic$class) <- c("first","second","third","crew")
z.out <- zelig(survived ~ age+sex+class, model="logit", data=titanic)
summary(z.out)
x.kate <- setx(z.out, ageadults=1, sexman=1,
               classsecond=0, classthird=0, classcrew=0)
x.kate[1,] <- c(1,1,0,0,0,0)
x.leo <- setx(z.out, ageadults=1, sexman=1,
              classsecond=0, classthird=1, classcrew=0)
x.leo[1,] <- c(1,1,1,0,1,0)
summary(s.out <- sim(z.out, x=x.leo, x1=x.kate))
```

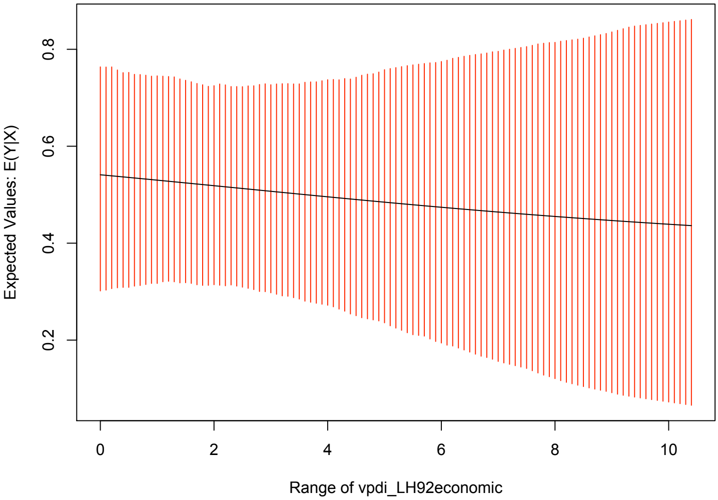
Calculating standard errors in Zelig

```
> summary(s.out <- sim(z.out, x=x.leo, x1=x.kate))

Values of X
  (Intercept) ageadults sexman classsecond classthird classcrew
1           1         1      1           0          1         0

Values of X1
  (Intercept) ageadults sexman classsecond classthird classcrew
1           1         1      0           0          0         0

Expected Values: E(Y|X)
   mean      sd    2.5%  97.5%
1 0.105 0.01205 0.08251 0.1290

Predicted Values: Y|X
      0     1
1 0.888 0.112

First Differences in Expected Values: E(Y|X1)-E(Y|X)
    mean      sd   2.5%  97.5%
1 0.7791 0.02423 0.7291 0.8227

Risk Ratios: P(Y=1|X1)/P(Y=1|X)
   mean    sd  2.5% 97.5%
1 8.538 1.062 6.723 10.89
```

More standard errors in Zelig

```
## economic_bills data
ecbills <- read.dta("economic_bills.dta")
z.out <- zelig(status ~ cabinet + vpdi_LH92economic + xland,
               model="logit", data=ecbills)
x.out <- setx(z.out)
x.out[1,] <- c(1,0,0,0,0,1)
summary(sim(z.out, x.out))
# for comparison:
predict(log2,new=data.frame(cabinet=0,vpdi_LH92economic=0,xland="UK"),
        type="response", se=T)

## economic_bills data qn4c
x.out[1,] <- c(1,1,5,1,0,0)
summary(sim(z.out, x.out))
# for comparison:
predict(log2,new=data.frame(cabinet=1,vpdi_LH92economic=5,xland="FRA"),
        type="response",se=T)

## economic_bills data qn4d
(x.out <- setx(z.out, vpdi_LH92economic=seq(0,10.4,.1)))
x.out[,2] <- 0
x.out[,5] <- 1
s.out <- sim(z.out, x.out)
plot.ci(s.out)
lines(seq(0,10.4,.1), apply(s.out$qi$ev,2,mean))
```

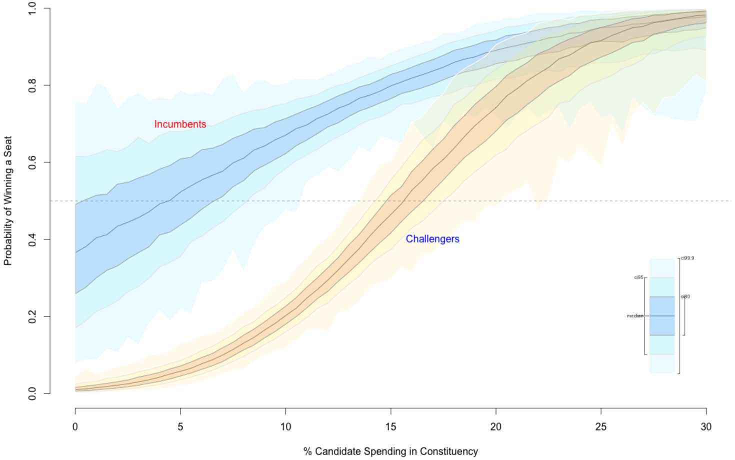
Plot from Economic bills data



Replicate Benoit and Marsh (PRQ, 2009) Figure 2

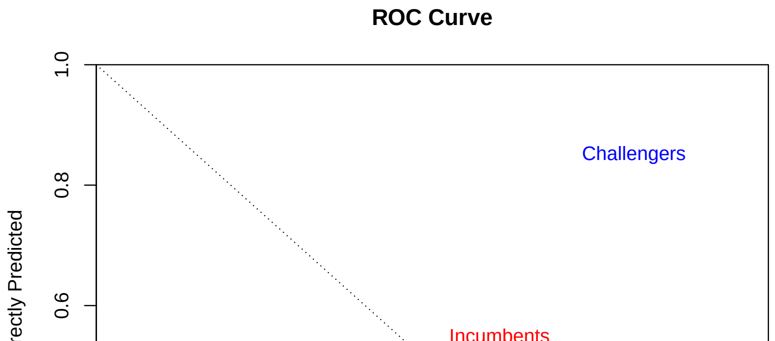
```
## replicate Figure 2 Benoit and Marsh (2009) PRQ
require(foreign)

## Loading required package:  foreign

suppressPackageStartupMessages(require(Zelig))
dail <- read.dta("http://www.kenbenoit.net/files/dailprobit.dta", convert.facto
z.out <- zelig(wonseat ~ pspend_total*incumb+m, model="probit", data=dail, cite
x.incumb <- setx(z.out, pspend_total=seq(0,30,.5), incumb=1, m=4)
x.chall <- setx(z.out, pspend_total=seq(0,30,.5), incumb=0, m=4)
# x.chall[1,5] <- .0001
s.out <- sim(z.out, x=x.incumb, x1=x.chall)
plot.ci(s.out, xlab="% Candidate Spending in Constituency",
        ylab="Probability of Winning a Seat")
text(5,.7,"Incumbents", col="red")
text(17,.4,"Challengers", col="blue")
abline(h=.5, lty="dashed", col="grey60")
```

Replicate Benoit and Marsh (PRQ, 2009) Figure 2



Compare models fits using a Receiver Operating Characteristic (ROC) plot

```
dail.incumb <- subset(dail, incumb==1, select=c(wonseat,pspend_total,incumb,m))
dail.chall <- subset(dail, incumb==0, select=c(wonseat,pspend_total,incumb,m))
z.out.i <- zelig(wonseat ~ pspend_total+m, model="probit", data=dail.incumb, ci
z.out.c <- zelig(wonseat ~ pspend_total+m, model="probit", data=dail.chall, cit
rocplot(z.out.i$y, z.out.c$y, fitted(z.out.i), fitted(z.out.c),
        lty1="solid", lty2="solid", col2="blue", col1="red")
text(.6,.55,"Incumbents",col="red")
text(.8,.85,"Challengers",col="blue")
```



Replicate Benoit and Marsh (PRQ, 2009) Figure 2