

# Probit Regression

Jim Albert

June 29, 2007

To illustrate Bayesian fitting of a probit regression model, consider data on 30 college students. For the  $i$ th student, we collect her SAT score  $SAT_i$  and an indicator  $y_i$  (1 or 0) if she passed a statistics class. If  $p_i = P(y_i = 1)$  denotes the probability the  $i$ th student passes the class, the probit model is represented as

$$\Phi^{-1}(p_i) = \beta_0 + \beta_1 SAT_i,$$

where  $\Phi^{-1}(\cdot)$  is the inverse cdf of the standard normal. If we place a uniform prior on the regression vector  $(\beta_0, \beta_1)$ , then the posterior density is given by

$$g(\beta_0, \beta_1) \propto \prod_{i=1}^{30} p_i^{y_i} (1 - p_i)^{1 - y_i}.$$

We first place the grades and sat scores for the 30 students in the vectors `grade` and `sat`.

```
> grade = c(0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1,
+ 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1)
> sat = c(525, 533, 545, 582, 581, 576, 572, 609, 559, 543, 576,
+ 525, 574, 582, 574, 471, 595, 557, 557, 584, 599, 517, 649,
+ 584, 463, 591, 488, 563, 553, 549)
```

We perform the usual mle fit of this model by the `glm` command, indicating by the `family=binomial(link="probit")` option that we are assuming binomial sampling with a probit link.

```
> fit1 = glm(grade ~ sat, family = binomial(link = "probit"))
> summary(fit1)
```

Call:

```
glm(formula = grade ~ sat, family = binomial(link = "probit"))
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.2977	-0.1469	0.3599	0.5177	1.4870

Coefficients:

```

                Estimate Std. Error z value Pr(>|z|)
(Intercept) -17.96110    6.62603  -2.711  0.00671 **
sat          0.03338     0.01195   2.794  0.00521 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

(Dispersion parameter for binomial family taken to be 1)

```

Null deviance: 36.652 on 29 degrees of freedom
Residual deviance: 22.233 on 28 degrees of freedom
AIC: 26.233

```

Number of Fisher Scoring iterations: 6

One can simulate from the posterior distribution by means of the data augmentation/Gibbs sampling algorithm of Albert and Chib (1993). This fitting is done by the function `bayes.probit`. The inputs are the vector of binary responses, the covariate matrix, and the number of simulations required. The output is a matrix, where each row corresponds to a single draw of the regression vector  $(\beta_0, \beta_1)$ .

```
> sim.par = bayes.probit(grade, cbind(1, sat), 10000)
```

We can summarize the simulated draws by the computation of the 5th, 50th, 95th percentiles. The 50th percentile is a point estimate and the 5th and 95th percentiles form a 90% interval estimate for  $\beta_i$ .

```
> apply(sim.par, 2, quantile, c(0.05, 0.5, 0.95))
```

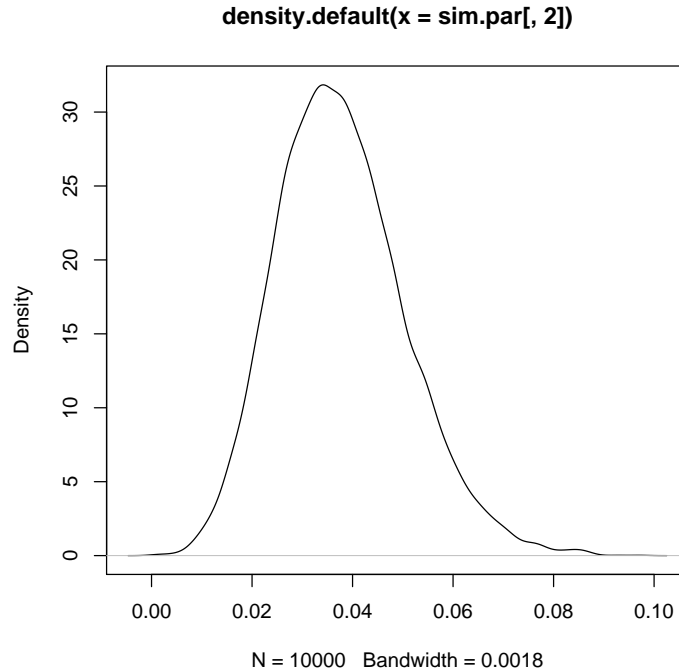
```

      [,1]      [,2]
5%  -32.829425  0.01877990
50%  -19.791962  0.03672554
95%   -9.893167  0.06015423

```

The following command constructs a density estimate of the simulated draws of the slope parameter  $\beta_1$ . Note that most of the mass is on positive values, indicating that there is a high probability that  $\beta_1$  is positive and there is an increasing relationship between SAT score and the probability of passing the class.

```
> plot(density(sim.par[, 2]))
```



To look further at the relationship between SAT score and course success, we can focus on the probability of passing  $p = \Phi(\beta_0 + \beta_1 SAT)$  for specific SAT scores. We can obtain simulated draws from the posterior of the fitted probability using the function `bprobit.probs`. First, we define three covariate vectors of interest corresponding to SAT scores 500, 530, and 560. We stack the vectors in the matrix `covariates`.

```
> cov1 = c(1, 500)
> cov2 = c(1, 530)
> cov3 = c(1, 560)
> covariates = rbind(cov1, cov2, cov3)
```

We then use the function `bprobit.probs`; the two inputs are the matrix of covariate and the matrix of simulated draws from `bayes.probit`. The output is a matrix of simulated draws `fitted.probs`, where each column corresponds to a particular covariate vector.

```
> fitted.probs = bprobit.probs(covariates, sim.par)
```

We summarize each posterior density of  $p = \Phi(\beta_0 + \beta_1 SAT)$  by a density estimate. We place all three density estimates on the same graph; this clearly shows how the passing probability increases as the SAT score increases.

```
> plot(density(fitted.probs[, 1]), xlim = c(0, 1), lwd = 2, xlab = "Probability of passing")
> lines(density(fitted.probs[, 2]), lwd = 2)
> lines(density(fitted.probs[, 3]), lwd = 2)
> text(0.18, 4, "SAT=500")
> text(0.38, 3, "SAT=530")
> text(0.8, 5, "SAT=560")
```

