FRAUD DETECTION BY A MULTINOMIAL MODEL: SEPARATING HONESTY FROM UNOBSERVED FRAUD

ANDREAS OLDEN*

JOINT WITH
JONAS ANDERSSON*
AND AIJA POLAKOVA*

*NORWEGIAN SCHOOL OF ECONOMICS

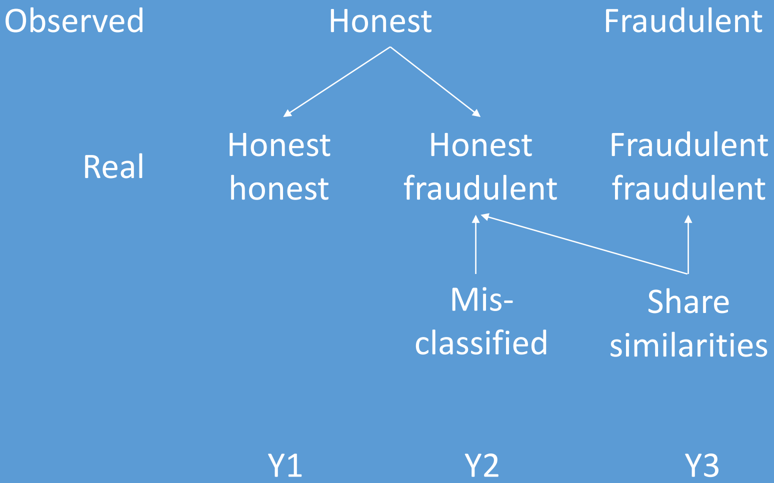
Overview

Fraud is often detected through audits. We then use this information to predict which new claims, tax returns etc. are fraudulent.

However, an audit process (almost) never has a perfect detection rate. This implies that a group of cheaters are (mis)classified as honest.

Artís, Ayuso and Guillén (2002) estimates that 5 percent of reviewed insurance claims are undetected fraudulent claims

Hausman, Abrevaya, and Scott-Morton (1998): Misclassification of only 2 percent can bias coefficients by 15-25 percent (probit)



Observed

Honest

Fraudulent

Real

Honest honest

Honest fraudulent

Fraudulent fraudulent

Mis-classified

Share similarities

Y1

Y2

Y3

Literature

Much work on fraud detection, but not that much on misclassification. Some computer science and marketing papers.

Hausman, Abrevaya and Scott-Morton (1998) runs a simulation study on misclassification in surveys

Artís, Ayuso and Guillén (2002) shows that HAS-M can be used in an insurance fraud setting, but no evaluation of performance

Caudill, Ayuso and Guillén (2005) introduces a new model based on the EM-algorithm, again without evaluation

The two latter articles shows that these methods can be used in setting close to tax fraud, but has nothing to say on whether we should do so.

The Trinomial Logit

If we observe all three categories, the estimation is straight forward

$$p_k = \frac{e^{\alpha_k + \beta_k x}}{1 + e^{\alpha_2 + \beta_2 x} + e^{\alpha_3 + \beta_3 x}},$$

for $k = 1, 2, 3$ and $\alpha_1 = \beta_1 = 0$.

And we max the log-likelihood

$$\ln L(\alpha_2, \alpha_3, \beta_2, \beta_3) = \sum_{i=1}^{n} (Y_{1i} \ln p_1 + Y_{2i} \ln p_2 + Y_{3i} \ln p_3)$$

However, since we do not observe Y_1 and Y_2, only $(Y_1 + Y_2)$, we have to consider them as latent variables. We now call

$$\ln L(\alpha_2, \alpha_3, \beta_2, \beta_3) = \sum_{i=1}^{n} (Y_{1i} \ln p_1 + Y_{2i} \ln p_2 + Y_{3i} \ln p_3)$$

the log-likelihood function for the *full data* (which is not completely observed).

The identifying assumption for the model is $\beta_2 = \beta_3$, meaning that the "HF" and the "FF" have similar characteristics.

But we do not get any further with the standard multinomial model without full data.

The EM-algorithm

1. Select starting values for $\alpha_2, \alpha_3, \beta_2$
2. E-step: Compute the expectation of $\ln L(\alpha_2, \alpha_3, \beta_2)$ given the *observed data*. $Q(\alpha_2, \alpha_3, \beta_2) = E(\ln L(\alpha_2, \alpha_3, \beta_2 | Y, X)$
3. M-step: Maximize Q to obtain new parameters
4. Use new parameters as new starting values, repeat until convergence.

In step 2 of the algorithm above we need to compute the following conditional expectations and use instead of Y_1 and Y_2

$$Y_1^* = E(Y_1 | Y_3 = 0) = \frac{1}{1 + e^{\alpha_2 + \beta_2 x}}$$

and

$$Y_2^* = E(Y_2 | Y_3 = 0) = \frac{e^{\alpha_2 + \beta_2 x}}{1 + e^{\alpha_2 + \beta_2 x}}$$

What we do

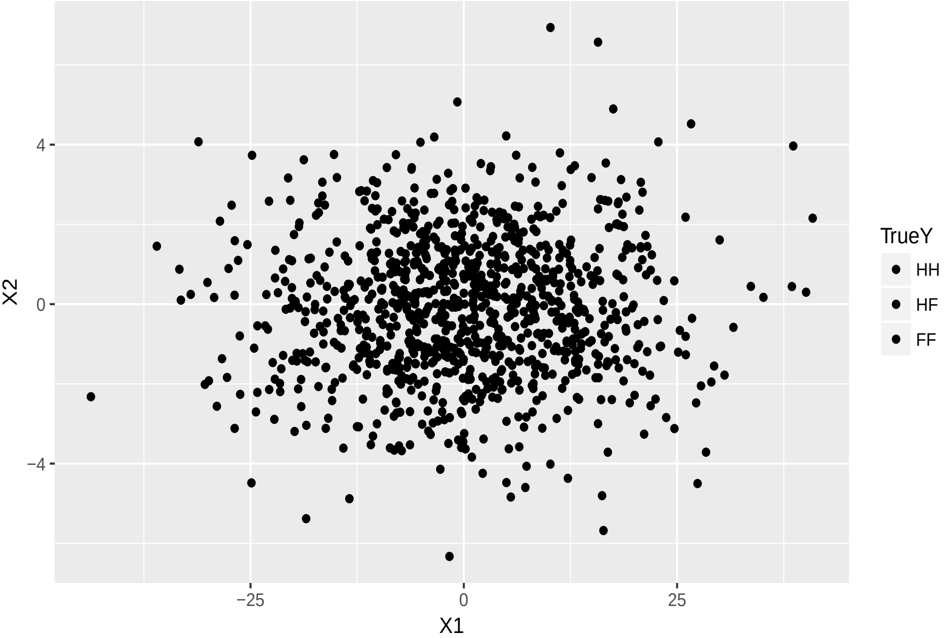
We do a **Monte Carlo simulation study**: 1000 random draws (N) of 2 x-variables with standard deviations from CAG (2005). We do this 1000 times (nrepl).

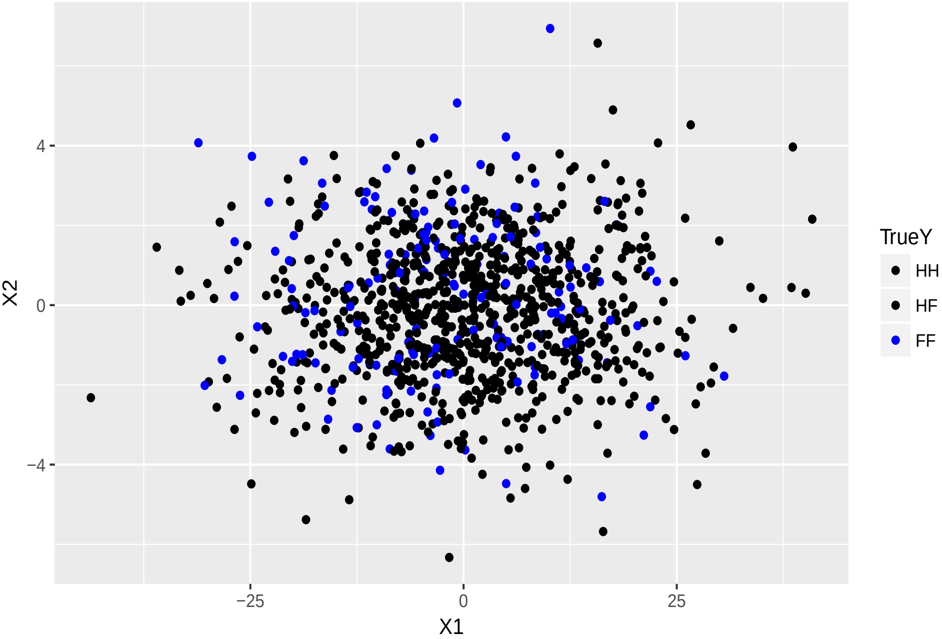
We then apply the parameters of CAG (2005) and apply a trinomial model to create outcomes, Y-variables, for the three categories HH, HF and FF

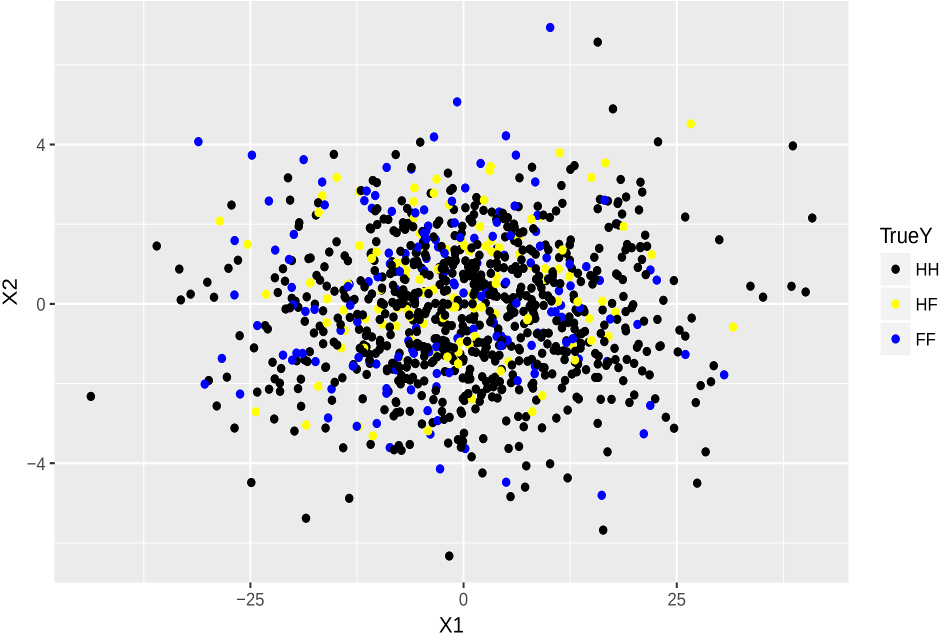
We pretend not to observe all three categories and use the EM-algorithm for missing data to see how close to the true values we come

We manipulate several conditions and see how the EM-algorithm performs and compare it to a naive binomial model which assumes no misclassification

Variable	Mean	Std dev	Coeff
CONSTANT	-	-	-1.440
AGE	38.02	12.32	-0.021
LICENSE	14.23	9.09	0.003
RECORDS	1.42	1.80	0.177
COVERAGE	0.91	0.29	0.795
DEDUCTIBLE	0.03	0.16	-0.303
ACCESORI	0.07	0.25	-0.350
VEHUSE	0.88	0.32	-0.507
VEHAGE	6.17	4.48	0.012
FAULT	0.32	0.47	1.388
NONURBAN	0.07	0.26	0.559
NIGHT	0.13	0.34	1.488
WEEKEND	0.27	0.44	0.274
WITNESS	0.01	0.08	1.140
POLICE	0.11	0.31	-1.805
ZONE1	0.14	0.34	0.320
ZONE3	0.49	0.50	0.642
REPORT	0.59	0.49	0.562
NAMES	0.06	0.24	1.172
PROXIM	0.02	0.13	1.716
DELAY	0.24	0.43	1.212


4
0
−4
X2
−25
0
25
X1
TrueY
● HH
● HF
● FF

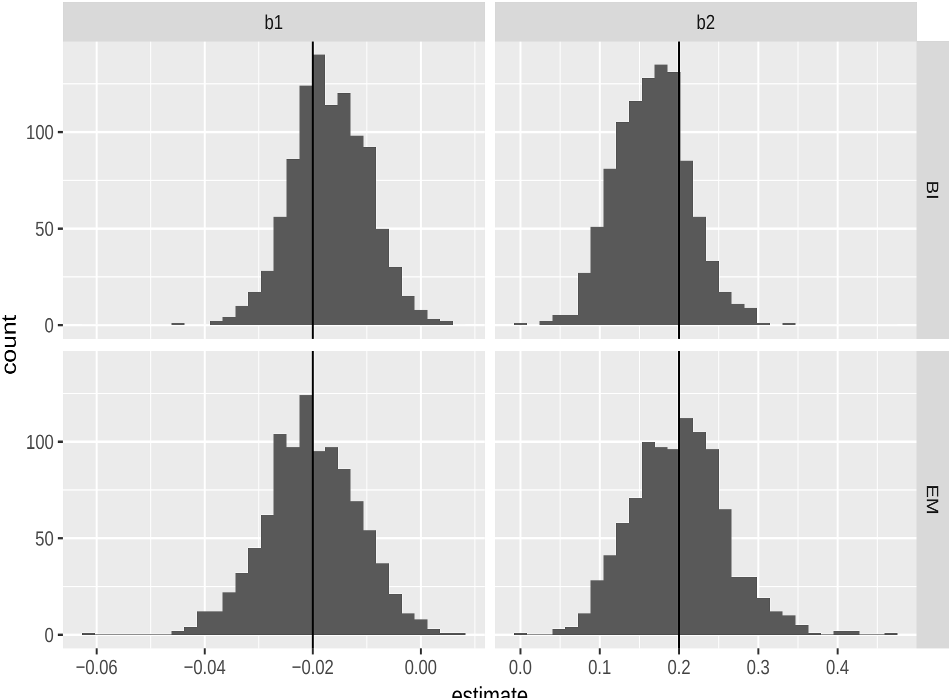

X2
4
0
−4
−25
0
25
X1
TrueY
● HH
● HF
● FF


X2
4
0
−4
−25
0
25
X1
TrueY
● HH
● HF
● FF

1000 coefficient estimates

Statistic	N	Mean	St. Dev.	Min	Max
iter	1,000	13.389	25.028	3	100
ad	1,000	0.167	0.000	0.167	0.167
y1	1,000	711.679	14.839	663	757
y2	1,000	122.698	10.495	89	159
y3	1,000	165.623	12.397	129	202
tb1	1,000	−0.020	0.000	−0.020	−0.020
tb2	1,000	0.200	0.000	0.200	0.200
EMb1	1,000	−0.020	0.009	−0.062	0.006
EMb2	1,000	0.200	0.059	0.007	0.474
BIb1	1,000	−0.017	0.007	−0.044	0.006
BIb2	1,000	0.166	0.047	0.006	0.337

Table: Estimations with 2 variables, 1000 draws (N), 1000 times (nrepl). tb are true betas, EM is the EM-algorithm, BI is the naive binomial


b1
b2
BI
EM
count
100
50
0
100
50
0
−0.06
−0.04
−0.02
0.00
0.0
0.1
0.2
0.3
0.4
estimate

Some more results

	BI	BI	EM	EM			
2 variables	$b1/\beta1$	$b2/\beta2$	$b1/\beta1$	$b2/\beta2$	y1	y2	y3
0 Corr	0.8	0.86	1	1.035	713	122	165
0.5 corr	0.85	0.85	1	1.015	715	121	164
0.9 corr	0.9	0.87	1.05	1.025	719	120	162
0 corr x-sd*10	0.35	0.36	1	1.02	579	179	241
0.9 corr x-sd*10	0.65	0.65	1	1.015	651	149	201
1 variable							
Base	0.85		1		717	121	163
a-dev 200% y2 small	0.95		1.1		783	40	178

Two variables with true values: $\beta1 = -0.02$, $\beta2 = 0.2$

Summary

EM does better than a naive binomial model in most cases

Particularly in cases with large misclassification and lots of variation in the data

with low misclassification the naive binomial case is slightly better

The EM-algorithm has higher standard deviation and more uncertainty (partly because more parameters are estimated).

Starting-values matter, but our suggested solution of using the numbers from the binomial logit works well in most cases

The EM-algorithm does not always converge to our criteria

The EM-algorithm is slow

Next steps

Add variables

Move to predictions- here the additional uncertainty with the EM-algorithm will matter

Try other estimation techniques- omission errors that do not explicitly model the FF (Hausman)

Other: Clustering

Possible real-life testing

The ideal would be an RCT with tax audits

One alternative would be to look at data for known evaders

If we for instance have data on foreign evaded income in Denmark and Sweden, we could pretend not to observe both and use the one to estimate the other.

References

Artís, M., Ayuso, M., & Guillén, M. (2002). Detection of automobile insurance fraud with discrete choice models and misclassified claims. Journal of Risk and Insurance, 69(3), 325-340.

Caudill, S. B., Ayuso, M., & Guillén, M. (2005). Fraud detection using a multinomial logit model with missing information. Journal of Risk and Insurance, 72(4), 539-550.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the royal statistical society. Series B (methodological), 1-38.

Hausman, J. A., Abrevaya, J., & Scott-Morton, F. M. (1998). Misclassification of the dependent variable in a discrete-response setting. Journal of Econometrics, 87(2), 239-269.