A Simulation Study

We provide a simulation study based on the model in §2.1 and we simulate data from the NLTCS based on our model, with varying levels of distortion. The varying levels of distortion (0, 0.25%, 0.5%, 1%, 2%, 5%) associated with the simulated data are then run using our MCMC algorithm to assess how well we can match under "noisy data." Figure 3 illustrates an approximate linear relationship with FNR and the distortion level, while we see an near-exponential relationship between FPR and the distortion level. Figure 4 demonstrates that for moderate distortion levels (per field), we can estimate the true number of observed individuals extremely well via estimated posterior densities. However, once the distortion is too *noisy*, our model has trouble recovering this value.

In summary, as records become more noisy or distorted, our matching algorithm typically matches less than 80% of the individuals. Furthermore, once the distortion is around 5%, we can only hope to recover approximately 65% of the individuals. Nevertheless, this degree of accuracy is in fact quite encouraging given the noise inherent in the data and given the relative lack of identifying variables on which to base the matching.

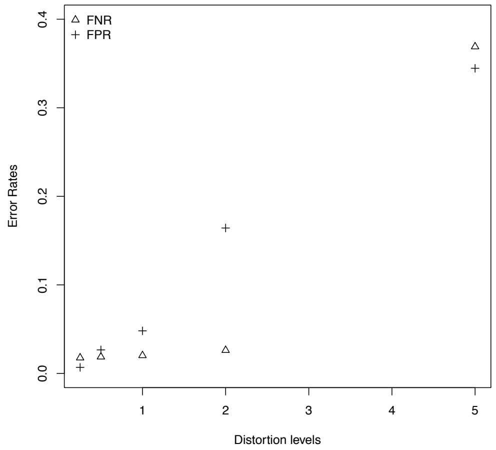

Figure 3: FNR and FPR plotted against 5 levels of distortion, where the former (plusses) shows near linear relationship and latter shows exponential one (triangles).

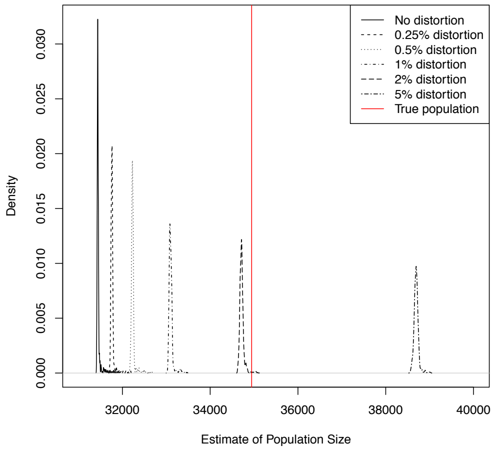

Figure 4: Posterior density estimates for 6 levels of distortion (none, 0.25%, 0.5%, 1%, 2%, and 5%) compared to ground truth (in red). As distortion increases (and approaches 2% per field), we undermatch N, however as distortion quickly increases to high levels (5% per field), the model overmatches. This behavior is expected to increase for higher levels of distortion. The simulated data illustrates that under our model, we are able to capture the idea of moderate distortion (per field) extremely well.

B Convergence Diagnostics and Hyperparameter Sensitivity

As for convergence diagnostics, for S_G, our standard for NLTCS when running SMERE was to set $S_G = S_M = 10^5$, after fixing on a burn-in of 1000 steps and a thinning the chain by 100 iterations from pilot runs. For SMERED, we used $S_G = 10^5$ and $S_M = 10000$. Moreover, our simulation study (Appendix A) varies a_ℓ and b_ℓ but we do not varying μ_l away from a uniform; if users have a priori knowledge regarding some idea about the expected distribution of categories, though, this could be incorporated fairly directly. For the NLTCS study itself, we set the parameters of β are $a_\ell = 5$ and $b_\ell = 10$ and took $\mu_\ell = 1$, corresponding to equivalent to a uniform distribution over the $M_\ell - 1$ simplex.

C Confusion Matrix for NLTCS

Est vs Truth	82	89	82,89	94	82, 94	89, 94	AY	RS
82	8051.9	0.0	385.1	0.0	162.9	0.0	338.6	8938.5
89	0.0	2768.4	291.1	0.0	0.0	240.6	131.7	341.8
94	0.0	0.0	0.0	7255.4	139.3	240.5	325.12	7960.32
82, 89	118.4	2.2	8071.7	0.0	4.4	0.4	803.2	9000.3
89, 94	0.0	186.8	6.1	190.6	1.5	7365.8	488.2	8239
82, 94	163.1	0.0	9.5	97.0	2662.2	0.09	331.5	3263.39
AY	62.5	1.6	164.4	28.9	51.7	10.6	15923.7	18342.02
NLTCS	8396	2959	4464	7572	1511	3929	6114	

Table 4: Confusion Matrix for NLTCS