A Novel Methodology for Improving Applications of Modern Predictive Modeling Techniques to Linked Data Sets Subject to Mismatch Error

Brady T. West, Institute for Social Research, University of Michigan

Emanuel Ben-David, U.S. Census Bureau

Martin Slawski, Department of Statistics, George Mason University

Acknowledgements

- Thanks to NSF-MMS for providing financial support for this work (NSF Grant #2120318, PI: Martin Slawski)
- Thank you to my co-authors, and also to Stas Kolenikov, who provided critical review of this work and stepped in to present the work at BigSurv23.

Motivation

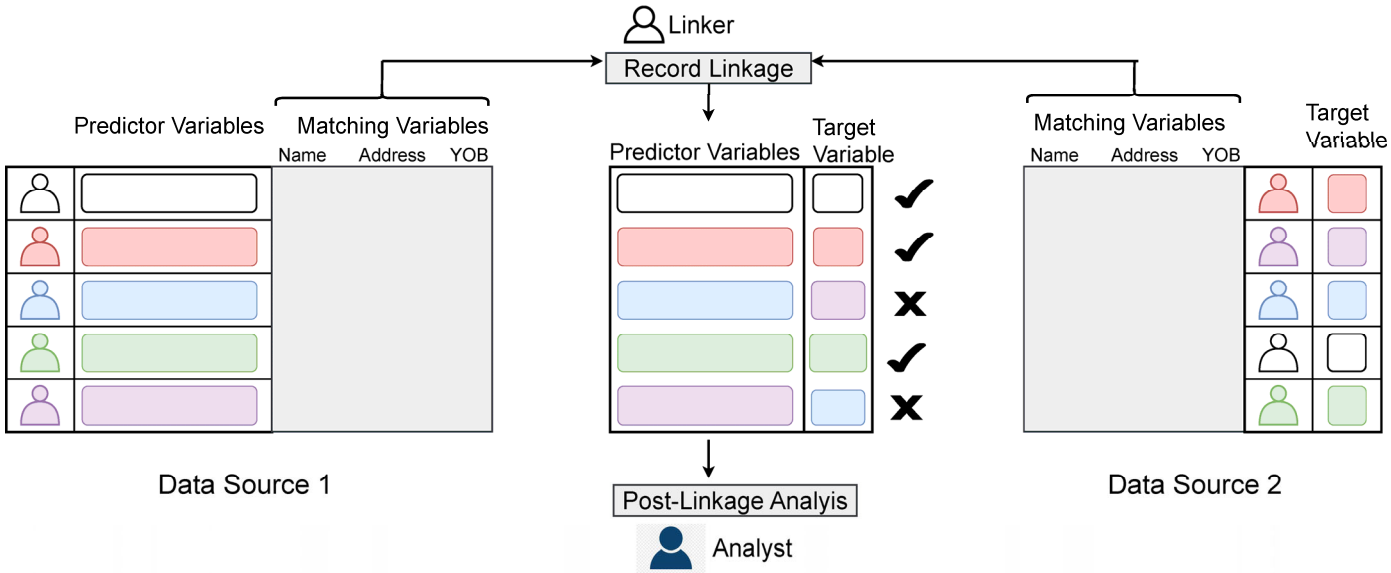
- Researchers frequently perform secondary analyses of linked data sets to answer novel research questions. Common linkages include:
  - Survey data and social media data
  - Survey data and administrative records
  - Electronic health records and patient questionnaires
- Machine learning techniques are often applied when analyzing such linked data sets, to explore patterns in the data and perform predictive modeling
  - **Example:** Predict reported wages (from survey data) with establishment size (from administrative data); see **Abowd et al., 2021**

Motivation, cont'd

- Probabilistic record linkage (RL) is often used when exact matches aren't possible, introducing error in the RL process
- **How can we adjust machine learning algorithms (specifically bagging and random forests) to account for these potential errors?**

Two Types of Errors in Record Linkage

- **Our focus today is on *mismatch error:***



Two Types of Errors in Record Linkage

- **Mismatch errors:** Records from two separate data files corresponding to two distinct entities are incorrectly matched
  - Several years of novel work on approaches to adjusting estimates in *regression models* in the presence of these errors (**Slawski et al., 2021**)
- **Missed-match errors:** The RL process fails to link a record from one data file with the correct match in a second data file, leading to only selected cases being matched
  - Similar to the problem of selection bias in surveys
- **Our focus is on the mismatch error problem, and how to account for it when using popular machine learning techniques**

A General Mixture Modeling Framework

We are interested in using an ensemble method to estimate some general regression function $\mu_{y|\mathbf{x}} = E[y \mid \mathbf{x}]$, where y corresponds to a dependent variable of interest and $\mathbf{x}$ represents a vector of values on predictor variables of interest.

After a record linkage process, we have values on these variables of interest available for each subject in a study denoted by i, with $i = 1, \ldots, n$.

In the *permuted* linked data file that arises as a result of a record linkage procedure subject to mismatch error, we observe $\tilde{y}_i$ instead of y_i, where some fraction of the cases in the linked data file have a mismatched value on the dependent variable y.

A General Mixture Modeling Framework

Following a mixture modeling approach (**Hof and Zwinderman, 2015**), the conditional density of $\tilde{y}_i \mid (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is equal to $(1-\alpha) f_{y_i|\mathbf{x}_i}(\tilde{y}_i \mid \mathbf{x}_i) + \alpha f_y(\tilde{y}_i)$, where $f_{y_i|\mathbf{x}_i}$ denotes the conditional density of $y_i \mid \mathbf{x}_i$, f_y denotes the marginal density of y, and α is the probability of a mismatch error.

More generally, we can assume a different α for each linked record i, that is,

$$f_{\tilde{y}_i|\mathbf{x}_1,\ldots,\mathbf{x}_n}(\tilde{y}_i \mid \mathbf{x}_1, \ldots, \mathbf{x}_n) = (1-\alpha_i) f_{y_i|\mathbf{x}_i}(\tilde{y}_i \mid \mathbf{x}_i) + \alpha_i f_y(\tilde{y}_i), \qquad (1)$$

where α_i is interpreted as the probability that $\tilde{y}_i \neq y_i$ (i.e., the probability of a mismatch error).

The conditional density in (1) implies that $\mu_{y_i|\mathbf{x}_i} = \frac{1}{1-\alpha_i}\mu_{\tilde{y}_i|\mathbf{x}_i} - \frac{\alpha_i}{1-\alpha_i}\mu_y$. (2)

Adjustment Approach 1: Optimal alpha

When analyzing real data in practice, we would first apply the analyst's favorite predictive modeling algorithm to the linked data (including the other algorithms to be introduced shortly), including mismatch errors.

Given the resulting estimates of $\hat{\mu}_{\tilde{y}_1|\mathbf{x}_1}, \ldots, \hat{\mu}_{\tilde{y}_n|\mathbf{x}_n}$, along with the sample mean of the observed $\tilde{y}_i$, we can then substitute these quantities in (2). Reminder:

$$\mu_{y_i|\mathbf{x}_i} = \frac{1}{1-\alpha_i}\mu_{\tilde{y}_i|\mathbf{x}_i} - \frac{\alpha_i}{1-\alpha_i}\mu_y. \qquad (2)$$

Adjustment Approach 1: Optimal alpha

This yields $f_{\tilde{y}_i|\mathbf{x}_1,\ldots,\mathbf{x}_n}(\tilde{y}_i \mid \mathbf{x}_1, \ldots, \mathbf{x}_n)$ as a function of α_i alone, which we can then maximize to find an optimal $\hat{\alpha}_i^{opt}$, which can then be used in (2) to obtain an improved estimate of $\mu_{y_i|\mathbf{x}_i}$.

We can also simply work with the mean of the $\hat{\alpha}_i^{opt}$, $\hat{\alpha}^{opt} = \sum_{i=1}^{n} \hat{\alpha}_i^{opt} / n$, in (2).

We refer to this as a **"Mean Optimal alpha"** approach. Mean Optimal alpha can be estimated from a sample of cases to save on computational time.

Adjustment Approach 1: Optimal alpha

The improvement in estimates of $\mu_{y_i|\mathbf{x}_i}$ based on this approach thus depends on:
(1) $\hat{\alpha}^{opt}$ being a good estimate of α,
(2) $\hat{\mu}_{\tilde{y}_i|\mathbf{x}_i}$ being a good estimate of $\mu_{\tilde{y}_i|\mathbf{x}_i}$, and
(3) the mixture density being a good fit to the conditional density of $\tilde{y}_i \mid (\mathbf{x}_1, \ldots, \mathbf{x}_n)$.

NOTE: This estimation of Optimal alpha values would typically be performed **after** applying a (possibly adjusted) predictive modeling algorithm to generate initial estimates of the regression function.

Adjustment Approach 2: Weighting-Reweighting

Extending this idea to the more general context of the ensemble methods (bagging and random forests) that are the focus of the current study, the α values described above can play the role of *weights* in the algorithms used to build the decision trees.

We distinguish between two different approaches to using weights in the construction of decision trees:

- ***Adj-trees***, where differential case weights are used at each step of the tree construction process to determine optimal splits, and
- ***Adj-rf***, where differential case weights are used when the bootstrap samples are selected for the ensemble method (and cases with a higher weight would have a higher probability of selection).

Adjustment Approach 2: Weighting-Reweighting

With no prior information about the mismatch probabilities, we would start by assigning a weight of 1 to each case and set $\alpha_i = 0.5$ for all cases. We then take some large number of bootstrap samples from the linked data.

For each sample, we first obtain $\hat{\mu}_{\tilde{y}_1|\mathbf{x}_1}, \ldots, \hat{\mu}_{\tilde{y}_n|\mathbf{x}_n}$ from a decision tree, or random forests, with our initial weights, and then we update α_i as the posterior probability of a mismatch, i.e., $\frac{\alpha_i f_y(\tilde{y}_i)}{(1-\alpha_i) f_{y|\mathbf{x}}(\tilde{y}_i|\mathbf{x}_i) + \alpha_i f_y(\tilde{y}_i)}$, and update the *weight* of each observation i as $1 - \alpha_i$.

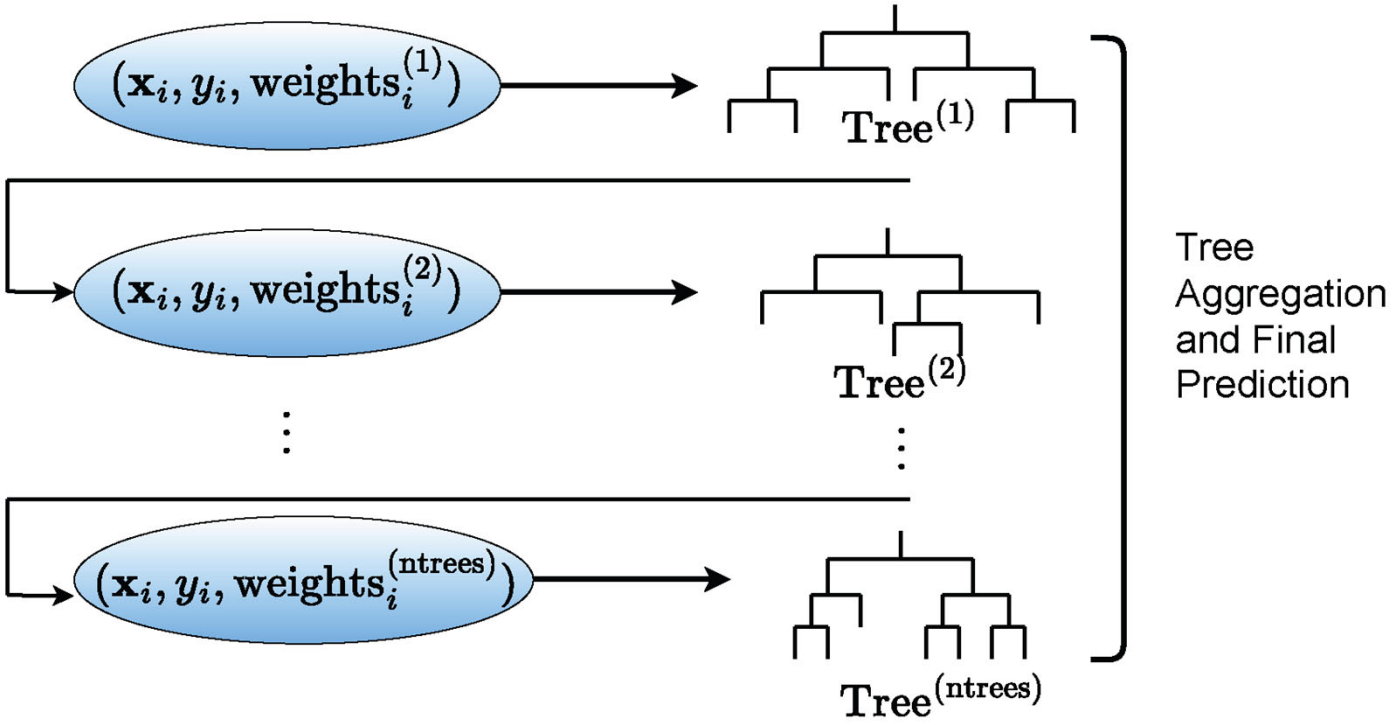
Adjustment Approach 2: Weighting-Reweighting

We then re-run the decision tree, or random forests, with these new weights to update the predictions $\hat{\mu}_{\tilde{y}_1|\mathbf{x}_1}, \ldots, \hat{\mu}_{\tilde{y}_n|\mathbf{x}_n}$. Note that the weights in the decision tree affect how the tree is split at each node.

We repeat this procedure, updating the weights and then updating $\hat{\mu}_{\tilde{y}_1|\mathbf{x}_1}, \ldots, \hat{\mu}_{\tilde{y}_n|\mathbf{x}_n}$ until the likelihood function indicates no significant improvement is obtained with the new weights.

In the end, we average over the $\hat{\mu}_{\tilde{y}_1|\mathbf{x}_1}, \ldots, \hat{\mu}_{\tilde{y}_n|\mathbf{x}_n}$ obtained from all the bootstrap samples, and report these as the adjusted predictions $\hat{\mu}_{\tilde{y}_1|\mathbf{x}_1}, \ldots, \hat{\mu}_{\tilde{y}_n|\mathbf{x}_n}$. See the visual of this approach on the next slide.

Adjustment Approach 2: Weighting-Reweighting



Possible Adjustment Approaches Evaluated

There are thus several alternative adjustment approaches that one could consider!

Adjustment Method	Description
Bagging (None)	A standard application of bootstrap aggregating (bagging) using the original linked data and no random selection of predictors at each step of the tree construction.
Random Forests (None)	A standard application of random forests, similar to bagging but including the random selection of possible predictors at each step of the tree construction.
Adj-trees	An application of the weighting-reweighting adjustment method that starts with default values of alpha (0.5) for all cases (and equal weights of 1), and then proceeds in an iterative fashion with applying weights to cases when splits are determined to construct individual trees. Improved estimates of the regression function are based on the mixture model.
Adj-rf	Like Adj-trees, but applying the weights in the selection of the bootstrap samples (rather than in the formation of splits).
Optimal-alpha-bagging	A modification of Bagging including a subsequent application of the Optimal alpha algorithm to improve adjusted estimates based on the mixture model. Given our results and the additional computational burden introduced by using a unique optimal alpha for each case (without apparent benefits of this approach), we focus on the mean optimal alpha value for all "Optimal alpha" approaches.
Optimal-alpha-rf	A modification of Random Forests including an application of the Optimal alpha algorithm to improve adjustment estimates based on the mixture model.
Optimal-alpha-Adj-trees	A modification of Adj-trees including the Optimal alpha algorithm.
Optimal-alpha-Adj-rf	A modification of Adj-rf including the Optimal alpha algorithm.

Simulation Design

We compare and evaluate the performance of the proposed adjustment methods with **two simulation studies**.

We use the **MSE** to measure the fit of a model and the **MSPE** to measure the quality of the model predictions:

- The **MSE** is calculated as the mean of squared residuals, i.e., $(\frac{1}{n})\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$, where n denotes the size of the data and $\hat{y}_i$ denoted the fitted y value for the i-th observation.
- The **MSPE** is calculated similarly but by averaging over the test data set, which is obtained by randomly splitting the data to 30% for the test data and the rest for the training data.

Simulation Design, cont'd

In the first simulation study, the simulated data set contains $n = 1000$ observations and two variables: a single predictor x and a response variable y generated via the equation $y_i = g(x_i) + 4N(0,1)$, where $g(\cdot)$ is a non-linear function of x, and $N(0,1)$ is Gaussian noise.

We first simulate a new data set resembling a linked data set, in which k percent of the pairs (y_i, x_i) are mismatched. The mismatches are created by randomly permuting k percent of the indices of y. The prediction and adjustment methods are trained by this data set.

To see the effects of different mismatch rates on these methods, the MSE and MSPE are computed for $k = 0, 10, 15, 20, 25, 30, 35, 40$. Henceforth, when no mismatches are in the data, i.e., $k = 0$, we refer to the data as the *exact* data.

Simulation Design, cont'd

Each run of the experiment thus results in two tables, one for MSE and another for MSPE, each with 8 columns representing the mismatch rate and 8 methods applied.

We run each experiment 50 times and report the averages for the MSE and the MSPE over the 50 replications.

In the second simulation study, the simulated data set also has $n = 1000$ observations with 10 predictors $x_1, \ldots, x_{10}$. The response variable y is generated via equation $y_i = h(\mathbf{x}_i) + \sqrt{2}/2N(0,1)$, where $h(\cdot)$ is a non-linear real-valued function of $\mathbf{x} = (x_1, \ldots, x_{10})^\top$.

Simulation Results: Single Predictor

MSE Results

Mismatch Rate	0%	10%	15%	20%	25%	30%	35%	40%
Bagging trees	22.01	28.72	42.72	50.37	62.46	79.12	92.82	108.09
Random forests	**18.42**	59.02	91.08	112.98	131.53	155.56	174.54	199.94
Adj-rf	18.47	25.36	**33.89**	40.28	56.6	77.64	89.71	109.99
Adj-trees	22.08	28.73	42.95	50.1	62.16	79.5	92.28	107.83
Optimal-α-bagging	21.9	**24.03**	34.64	36.97	46.12	**60.47**	70.62	78.1
Optimal-α-rf	19.05	60.45	93.53	115.45	134.77	161.27	181.68	208.68
Optimal-α-Adj-rf	19.09	27.88	37.73	46.17	63.76	86.13	101.31	127.07
Optimal-α-Adj-trees	21.98	24.06	34.9	**36.73**	**45.86**	60.92	**70.09**	**77.91**

Run Times (Seconds)

Optimal-α	**Mean-optimal α**	**Bagging trees**
15.455	1.911	0.614
Random forests	**Adj-rf**	**Adj-trees**
0.041	0.26	7.569

Summary:

- The best adjustment is achieved by combining the optimal-α adjustment with the Adj-trees method.
- Among the methods with a single adjustment (less computational time), Optimal-α-bagging, i.e., Optimal-α adjustment with $\hat{\mu}_{y|\mathbf{x}}$ estimated from bagging trees, has the best performance.

Simulation Results: Single Predictor

MSPE Results

Mismatch Rate	0%	10%	15%	20%	25%	30%	35%	40%
Bagging trees	34.8	40.74	50.42	60.08	71.78	89.67	104.16	127.06
Random forests	**18.49**	61.41	84.48	110.24	136.25	162.86	186.68	223.27
Adj-rf	18.51	27.72	34.12	43.91	56.47	70.66	89.6	120.45
Adj-trees	23.72	30.61	38.59	49.57	61.62	78.19	94.32	115.54
Optimal-α-bagging	35.48	37.7	43.66	48.44	55.37	67.33	77.83	95.29
Optimal-α-rf	19.2	62.81	86.42	112.03	139.59	167.62	192.49	233.51
Optimal-α-Adj-rf	19.23	30.52	39	50.79	65.6	84.35	105.84	143.96
Optimal-α-Adj-trees	23.82	**26.28**	**30.09**	**35.6**	**42.75**	**53**	**64.86**	**80.22**

Summary:

- Optimal-alpha-Adj-trees seems to do best in terms of MSPE
- Consistent support for this approach overall, but computational time is a trade-off

Simulation Results: Multiple Predictors

MSE Results

Mismatch Rate	0%	10%	15%	20%	25%	30%	35%	40%
Bagging	0.75	0.79	0.82	0.87	0.92	0.95	0.99	1.05
Random forests	0.84	0.91	0.94	1	1.03	1.07	1.11	1.15
Adj-rf	0.84	0.86	0.89	0.91	0.94	0.96	0.99	1.03
Adj-trees	0.75	0.79	0.82	0.87	0.92	0.95	0.99	1.05
Optimal-α-bagging	**0.7**	**0.72**	**0.75**	**0.79**	**0.84**	**0.87**	**0.9**	**0.96**
Optimal-α-rf	0.8	0.85	0.88	0.93	0.96	0.99	1.03	1.08
Optimal-α-Adj-rf	0.8	0.82	0.84	0.86	0.88	0.9	0.93	0.97
Optimal-α-Adj-trees	**0.7**	**0.72**	**0.75**	**0.79**	**0.84**	**0.87**	0.91	**0.96**

Run Times (Seconds)

Optimal-α	**Mean-optimal-α**	**bagging trees**
26.544	2.886	3.029
random forest	**Adj-rf**	**Adj-trees**
0.065	0.239	121.836

Summary:

- Optimal-alpha-bagging seems to do best when taking computational time into account
- Performance declines as mismatch rates increase

Simulation Results: Multiple Predictors

MSPE Results

Mismatch Rate	0%	10%	15%	20%	25%	30%	35%	40%
Bagging trees	1.23	1.27	1.29	1.32	1.37	1.39	1.4	1.45
Random forests	0.88	0.94	0.98	1.01	1.06	1.12	1.15	1.2
Adj-rf	0.88	0.91	0.94	0.95	1	1.04	1.06	1.09
Adj-bagging	0.97	1	1.03	1.04	1.08	1.12	1.14	1.19
Optimal-α-bagging	1.32	1.35	1.35	1.38	1.42	1.44	1.43	1.48
Optimal-α-rf	**0.83**	0.88	0.91	0.93	0.99	1.04	1.07	1.11
Optimal-α-Adj-rf	**0.83**	**0.86**	**0.88**	**0.89**	**0.93**	**0.97**	**0.99**	**1.02**
Optimal-α-Adj-trees	0.95	0.96	0.98	0.98	1.02	1.06	1.07	1.12

Summary:

- Optimal-alpha-Adj-rf seems to do best in terms of MSPE
- The random forests approach seems robust to increases in mismatch rate

Summary of Results

The simulation studies show that random forests and bagging trees can perform well on the exact data; however, their performance can rapidly deteriorate as mismatch rates increase.

In the presence of mismatches in the data, our proposed methods are generally effective in adjusting the outputs of random forests and bagging trees and improving their performance in terms of reduction in MSE, MSPE, or bias in the estimated regression function.

Combining the optimal-α method with Adj-rf or Adj-trees can be more effective than a single adjustment method.

Summary of Results, cont'd

Both Optimal-α-Adj-rf and Adj-trees are computationally expensive, however, and may not be scalable for large data sets.

In the case of the optimal-α method, the mean-optimal α method is a viable replacement. The mean-optimal α can be implemented much faster, and in both of our simulations, the mean-optimal α method performs almost identically to the optimal-α method.

However, this may be because the mismatches in the data are created by a permutation selected at random; thus, each observation has the same probability of being correctly matched. **This is a good direction for future work.**

Software

- R Software implementing these adjusted approaches is available on GitHub: https://github.com/ehb2126/Data-Analysis-after-Record-Linkage

Future Extensions

While we focused on bagging trees and random forests in this empirical evaluation, the methodology described in this paper can be extended to other popular machine learning approaches, such as gradient boosting and neural networks.

A recommendation for future research is a more thorough comparison between the optimal-α and mean-optimal-α methods by creating a more complex pattern for matches and mismatches, where the observations can have different probabilities of being correctly matched.

Future Extensions, cont'd

Finally, we also focused on *mismatches* in this study, as opposed to *missed matches*.

The latter type of error in record linkage is more likely to introduce selection bias in estimates of the relationships between variables based on the linked data file, depending on the extent to which the linked records differ from the missed matches in terms of the relationships of interest.

Future work needs to focus on methods for accounting for missed matches as well.

Thanks!

- Please direct any questions about this work to Brady West (bwest@umich.edu).

References

Abowd J. M., Abramowitz J., Levenstein M. C., McCue K., Patki D., Raghunathan T., Rodgers A. M., Shapiro M. D., Wasi N., Zinsser D. (2021), “Finding Needles in Haystacks: Multiple Imputation Record Linkage Using Machine Learning,” U.S. Census Bureau Working Paper No. CES-21-35.

Hof, M., & Zwinderman, A. (2015). A mixture model for the analysis of data derived from record linkage. *Statistics in Medicine*, 34, 74–92.

Slawski, M., Diao, G., & Ben-David, E. (2021). A pseudo-likelihood approach to linear regression with partially shuffled data. *Journal of Computational and Graphical Statistics*, 30(4), 991–1003.