Evaluating binary alignment methods in microsimulation models

#2012-003

Evaluating binary alignment methods in microsimulation models

By Jinjing Li and Cathal O'Donoghue

Maastricht Economic and social Research institute on Innovation and Technology (UNU-MERIT)
email: info@merit.unu.edu | website: http://www.merit.unu.edu

Maastricht Graduate School of Governance (MGSoG)
email: info-governance@maastrichtuniversity.nl | website: http://mgsog.merit.unu.edu

Keizer Karelplein 19, 6211 TC Maastricht, The Netherlands
Tel: (31) (43) 388 4400, Fax: (31) (43) 388 4499

UNU-MERIT Working Papers

ISSN 1871-9872

Maastricht Economic and social Research Institute on Innovation and Technology, UNU-MERIT

Maastricht Graduate School of Governance
MGSoG

Evaluating Binary Alignment Methods in Microsimulation Models[1]

Jinjing Li[2]

The National Centre for Social and Economic Modelling, University of Canberra

Maastricht University

Cathal O'Donoghue[3]

Rural Economy and Development Programme, Teagasc

Abstract: Alignment is a widely adopted technique in the field of microsimulation for social and economic policy research. However, limited research has been devoted to the understanding of their simulation properties. This paper discusses and evaluates six common alignment algorithms used in the dynamic microsimulation through a set of theoretical and statistical criteria proposed in the earlier literature (e.g. Morrison 2006; O'Donoghue 2010). This paper presents and compares the alignment processes, probability transformations, and the statistical properties of alignment outputs in transparent and controlled setups with both synthetic and real life dataset (LII). The result suggests that there is no single best method for all simulation scenarios. Instead, the choice of alignment method might need to be adapted to the assumptions and requirements in a specific project.

Key words: alignment, microsimulation, algorithm evaluation

[1] The authors are grateful to Rick Morrison, Howard Redway and Steven Caldwell for helpful discussions over time in relation to alignment in microsimulation models. We are grateful to the Luxembourg AFR for supporting this research.

[2] Email address: Jinjing.Li@maastrichtuniversity.nl

[3] Email address: Cathal.odonoghue@teagasc.ie

Evaluating Alignment Methods in Dynamic Microsimulation Models

I. INTRODUCTION

Microsimulation is a technique used to model complex real life events by simulating the actions and the impact of policy change on the individual micro unit. (Harding, 2007) Microsimulation models are usually categorised into “static” or “dynamic”. Static models, e.g. EUROMOD (Mantovani et al., 2007), are often arithmetic models that evaluate the immediate distributional impact upon individuals/households of possible policy changes. Dynamic models, e.g. DESTINIE, PENSIM, SESIM (Bardaji et al., 2003, Curry, 1996, Flood, 2007), extend the static model by allowing the individuals to change their characteristics as a result of endogenous factors within the model (O’Donoghue, 2001). Using this method, it is possible to generate new simulated populations that can be used for policy and scenario analysis.

Dynamic microsimulation models typically simulate behavioural processes such as demographic (e.g. marriage), labour market (e.g. unemployment) and income characteristics (e.g. wage). The method uses statistical estimates of these systems of equations and then applies Monte Carlo simulation techniques to generate the new populations, typically over time, both into the future and when creating histories with partial data, into the past.

As statistical models are typically estimated on historical datasets with specific characteristics and period effects, projections of the future may therefore contain error or may not correspond to exogenous expectations of future events. In addition, the complexity of micro behaviour may mean that simulation models may over or under predict the occurrence of a certain event, even in a well-specified model (Duncan and Weeks, 1998). Because of these issues, methods of calibration known as alignment have been developed within the microsimulation literature to correct for issues related to the adequacy of micro projections.

Scott (2001) defines alignment as “a process of constraining model output to conform more closely to externally derived macro-data ('targets').” There are both arguments for and against alignment procedures (Baekgaard, H., 2002). Concerns directed towards alignment mainly focus on the consistency issue within the estimates and the level of disaggregation at which this should occur. It is suggested that equations should be reformulated rather than constrained ex post. Clearly, in an ideal world, one would try to estimate a system of equations that could replicate reality and have effective future projections without the need for alignment. However, as Winder (2000) stated, “microsimulation models usually fail to simulate known time-series data. By aligning the model, goodness of fit to an observed time series can be guaranteed.” Some modellers suggest that alignment is an effective pragmatic solution for highly complex models. (O’Donoghue, 2010)

Over the past decade, aligning the output of a microsimulation model to exogenous assumptions has become standard despite this controversy. In order to meet the need of alignment, various methods, e.g. multiplicative scaling, sidewalk, sorting based algorithm etc., have been experimented along with the development of microsimulation (See Morrison, 2006). Microsimulation models using historical datasets, e.g. CORSIM, align the output to historical data to create a more credible profile (SOA, 1997). Models that work prospectively, e.g. APPSIM, also utilise the technique to align their simulation with external projections (Kelly and Percival, 2009).

Nonetheless, the understanding of the simulation properties of alignment in microsimulation models is very limited. Literature on this topic are scarce, with a few exceptions such as Anderson (1990), Caldwell et al. (1998), Neufeld (2000), Chénard (2000a, 2000b), Johnson (2001), Baekgaard (2002), Morrison (2006), Kelly and Percival (2009) and O'Donoghue (2010). Although some new alignment methods were developed in an attempt to address some theoretical and empirical deficiencies of earlier methods, discussions on empirical simulation properties of different alignment algorithms are almost non-existent.

This paper aims to fill this gap and better understand the simulation properties of alignment algorithms in microsimulation. It evaluates all major binary alignment methods using a simple microsimulation model with a set of synthetic datasets and a real life dataset. It compares the alignment processes, probability transformations, and the statistical properties of alignment outputs in transparent and controlled setups. In addition, a real life panel dataset, Living in Ireland (LII), is used together with a simplified microsimulation model to evaluate the alignment performances in typical microsimulation project setup. Alignment performances are tested using various evaluation criteria, including the ones outlined in Morrison (2006).

The present paper is divided into 6 sections. In the next section, we will review the background to the alignment methodology used in microsimulation and summarizes the existing algorithms used in various models. Section 3 discusses the objectives of alignment and the method of algorithm evaluation. Section 4 describes the detail of the datasets used in the evaluation process and some key statistics. We will present the results of the evaluation in section 5, and conclude in the last section.

II. ALIGNMENT IN MICROSIMULATION

This section discusses the purpose of alignment in a microsimulation model and the common practise of their statistical implementation. Baekgaard (2000) suggests two broad categories for alignment: parameter alignment,

- whereby the distribution function is changed by adjustment of its parameters; and *ex post* alignment,
- whereby alignment is performed on the basis of unadjusted predictions or interim output from a simulation.

This paper primarily focuses on the *ex post* alignment methods, as they are the most common form of alignments in microsimulation.

Models of continuous events such as the level of earnings or investment income utilise statistical regressions with continuous dependent variables and produce a distribution of continuous values. However, the prediction of the statistical model may deviate from the expectation for example due to an expected change in the distribution or productivity or may need to be adjusted for scenario analysis. This raises the need for alignment, which is often may be an adjustment of multiplicative applied continuous variables or via adjusting the error distribution (Chénard, 2000a).

For binary variables however, one cannot not apply the same method, as binary variable simulation uses discrete choice models such as logit, probit or multinomial logit models and the outputs cannot be adjusted in this way like continuous variables. As the majority of processes, e.g. in-work, employment, health, retirement, etc., in dynamic microsimulation models are binary choice in nature, this paper focus its attention on the alignment of binary choice models.

Models of discrete events such as in-work, employment status, disability status etc. are typically produce probabilities of the event occurring as output. These models can be expressed in the following generic form:

$$f\left(p_i\right) = \alpha + \beta X_i + \varepsilon_i \qquad (1)$$

As seen, equation 1 can be divided into a deterministic component $\alpha + \beta X_i$ and a stochastic component ε_i. In a simple Monte Carlo simulation, we generate the random number ε_i^*, adjust the model for endogenous changes in the explanatory variables to produce a new deterministic component $\alpha + \beta X_i^*$ and simulate a new dependent variable.

In the case of a binary choice we produce[4]:

$$f\left(p_i^*\right) = \alpha + \beta X_i^* + \varepsilon_i^* \qquad (2)$$

The dependent variable is predicted to have a value 1 if $f\left(p_i^*\right) \geq 0$ and 0 otherwise[5]. In most cases, a microsimulation model applies this prediction process to all observations individually without interaction. However, this may lead to a potential side effect: The output of the predication, although it may look reasonable at each individual level, may not meet the modeller's expectation at the aggregate level. For instance, the simulated average earning might be higher or lower than the assumption, or the in-work rate is beyond the expectation. Therefore, alignment is introduced as the step after the initial prediction in order to correct this "error".

Although the theoretical debate of alignment is not over, alignment is *de facto* widely adopted in the models built or updated within last decade, e.g. DYANACAN (Neufeld, 2000), CORSIM (SOA, 1997), APPSIM (Bacon, 2009). Many papers, e.g. Baekgaard (2002), Bacon (2009) and O'Donoghue (2010), have discussed the main reasons for alignment, and summarise them as follows:

- Alignment may be used to 'repair' the unfortunate consequences of insufficient estimation data by incorporating additional information in the simulations. Since no country has an ideal dataset for estimating all the parameters needed for microsimulation, modellers often make compromises, which adversely affects the output quality. Alignment can be used to fix some of these errors.

[4] Note $f\left(p_i^*\right)$ in the case of a logit model is defined as $f\left(p_i^*\right) = \ln\left(\frac{p_i^*}{1-p_i^*}\right)$

[5] A more detailed description of logit based discrete model in microsimulation can be found in O'Donoghue (2010)

- Alignment can be used to adjust for poor predictive performance of the micro model or its misspecification. Even with perfect data, relationships between dependent variables and explanatory variables may change considerably in countries where substantial structural changes are taking place. Alignment allows one to correct for these issues and make the simulation consistent with holistic projection assumptions.
- Alignment provides an opportunity for producing scenarios based on different assumptions. Examples include the simulation of alternative recession scenarios on employment with different impacts on different social groups (e.g. sex, education or occupation)
- Alignment is instrumental in establishing links between microsimulation models of the household sector and the macro models. It is a crucial step to reach a consistent Micro-Macro simulation model (see Davies 2004).
- Alignment can be used to reduce Monte Carlo variability though its deterministic calculation (Neufeld, 2000). This is particularly useful for small samples to confine the variability of aggregate statistics.

Alignment Methods

In order to calibrate a simulation of a binary variable, we need a method that can adjust the outcome of a logit or probit model to produce outcomes that are consistent with the external total. At the moment, there is no standardised method for implementing alignment in microsimulation. Given that different modellers may have different views or needs, it is not surprising that various binary alignment methods have appeared.

Papers by Neufeld (2000), Morrison (2006) and O'Donoghue (2010) provide descriptions on some popular options for alignment used in the literature. Existing documented alignment methods include

- Multiplicative Scaling
- Sidewalk Shuffle, Sidewalk Hybrid and their derivatives
- Central Limit Theorem Approach
- Alignment by Sorting (with different sorting variables)

Multiplicative scaling, which was described in Neufeld (2000), involves undertaking an unaligned simulation using Monte Carlo techniques and then comparing the proportion of transitions with the external control total. The ratio between the desired transition rate and the actual transition is calculated and applied in a second pass to the simulated probabilities. The method, however, is criticized by Morrison (2006) as probabilities are not limited to the range 0-1, although the problem is rare in practice as the multiplicative ratio tends to be small. Neufeld (2000) suggests solutions to this may include using nonlinear adjustment.

The sidewalk method was first introduced in Neufeld (2000) as a variance reduction technique, which was also used as an alternative to pure Monte Carlo simulation. It reduces the possibility of unlikely simulated outcomes because of the use of random numbers. The original method, however, does not align the simulated data to an external control. It simply involves accumulating a running total of predicted probabilities. Once the accumulation exceeds 1, a transition occurs. Therefore, it eliminates the use of random numbers as a variance reduction technique. Nevertheless, the method has some difficulties in output replications when the order observations changes. The order of the observations may be altered due to the deletion of an observation (e.g. deaths) or other changes. Serial correlation within families (or other clustering unit) is also an issue as people within the cluster are simulated in order. It is therefore unlikely for two people within a family to be simulated to make a transition in one year if the transitional probabilities are low.

Neufeld (2000) further developed an alignment method that he characterized as a hybrid of independent Monte Carlo simulation and the sidewalk method. DYNACAN adopted this method with non-linear adjustment to the equation-generated probabilities, combined with a minor tweaking of the resulting probabilities depending on whether the simulated rate is ahead of or behind the target rate for the pool during the progress and some randomisations. (Morrison, 2006). The method calibrates the probabilities through the logit transformation instead using probabilities directly in order to assure the values are bounded between 0 and 1. (SOA, 1998) Sidewalk Hybrid method requires two key parameters, which decides how similar the output is to standard Monte Carlo or standard sidewalk method.

The Central Limit Theorem approach is described in Morrison (2006). It utilises the assumption that the mean simulated probability is close to the expected mean when N is large. It manipulates the probabilities of each individual observation on the fly so that the simulated mean matches the expectation. A more detailed description of the method can be found in Morrison (2006). As all the methods we have discussed so far, this method does not need any sorting routine.

Alignment by sorting was first documented by O’Donoghue (2001) and Johnson (2001). It involves sorting of the predicted probability adjusted with a stochastic component, and selects desired number of events according to the sorting order. It is seen as a more “transparent” method (O’Donoghue, 2010) although computationally more intensive due to the sorting procedure. Many variations of the methods have been used in the past years and we will discuss the mostly used three algorithms in this paper:

- Sort by predicted probability (SBP),
- Sort by the difference between predicted probability and random number (SBD), and
- Sort by the difference between logistic adjusted predicted probability and random number (SBDL).

Sort by predicted probability (SBP)

Assuming that the predicted probability from a logit model can be defined as:

$$p_i^* = \frac{\exp\left(\alpha + \beta X_i^*\right)}{1 + \exp\left(\alpha + \beta X_i^*\right)} \qquad (3)$$

p_i^* is the predicted probability, both α and β are estimated coefficients. This method essential picks up the observations with highest p_i^* in each alignment pool. One consequence, however, is that those with the highest risk are always being selected for transition. In the example of in-work, the higher educated, all other things being equal would be selected to have a job. In reality those with the highest risk will on average be selected more than those with lower risk, but not always be selected. As a result some variability needs to be introduced. Kelly and Percival (2009) propose a variant of this method, where a proportion (typically 10% of the desired number) are selected when the sorting order is inverted, so as to allow low risk units to make a transition.

Sort by the difference between predicted probability and random number (SBD)

Given the shortcoming of the simple probability sorting, Baekgaard (2002) uses another method, which sorts by differences between predicted probability and a random number. Instead of sorting the probability p_i^* directly, it sorts r_i, which equals to the difference between p_i^* and a random number u_i, a number that is uniformly distributed between 0 and 1. Mathematically, this sorting variable can be defined as follows:

$$r_i = \text{logit}^{-1}\left(\alpha + \beta X_i\right) - u_i = \frac{\exp\left(\alpha + \beta X_i\right)}{1 + \exp\left(\alpha + \beta X_i\right)} - u_i \qquad (4)$$

A concern about this method is that the range of possible sorting values is not the same for each point. In other words, because the random number $\mathrm{u_i} \in [0,1]$ is subtracted from the deterministically predicted $\mathrm{p*_i}$, and the sorting value takes the range $\mathrm{r_i} \in [-1,1]$. For each individual, r will only take a possible range $\mathrm{r_i} \in [\mathrm{u_i} - 1, \mathrm{u_i}]$. As a result, when $\mathrm{p*_i}$ is small say 0.1, the range of possible sorting values is [-0.9, 0.1]. At the other extreme if $\mathrm{p*_i}$ is large say = 0.9, then the range of possible sorting values is [-0.1, 0.9]. Thus because there is only a small overlap for these extreme points, an individual with a small $\mathrm{p*_i}$ will have a very low chance of being selected even if a low value random number is paired with the observation. Ideally the range of possible sorting values should be the same, so that for each individual, $\mathrm{r_i} \in [a, \mathrm{b}]$, with individuals with a low $\mathrm{p*_i}$ being clustered towards the bottom and those with a high $\mathrm{p*_i}$ being clustered towards the top.

Sort by the difference between logistic adjusted predicted probability and random number (SBDL)

An alternative method described in Flood et al. (2005), Morrison (2006) and O'Donoghue et al. (2008) mitigates the range problem of SBD by using logistic transformation. This method takes a predicted logistic variable from a logit model, $\text{logit}(p_i) = \alpha + \beta X_i$ combined with a random number ε_i that is drawn from a logistic distribution to produce a randomised variable:

$$p_i = \text{logit}^{-1}\left(\alpha + \beta X_i + \varepsilon_i\right) \qquad (5)$$

p_i is then used to sort individuals and similarly the top n_j of households are selected. The sorting variable can therefore be described as follows:

$$r_i = \text{logit}^{-1}(\alpha + \beta X_i + \varepsilon_i) = \frac{\exp(\alpha + \beta X_i + \varepsilon_i)}{1 + \exp(\alpha + \beta X_i + \varepsilon_i)} \qquad (6)$$

ε_i is a logistically distributed random number with mean value 0 and a standard error of $\pi / \sqrt{3}$ .Since the random number is not uniformly distributed as u_i in the previous method, it produces a different sorting order.

III. Methods of Evaluating Alignment Algorithm

In order to evaluate the simulation properties of all alignment algorithms, it is important to define what we need to compare, and what the criteria are. Although different alignment methods have been briefly documented in a few papers, there is little discussion on the actual performance differences among these methods. Implementations vary from model to model, but no paper so far validates the alignment methods. This paper tries to evaluate different algorithms and compares how they perform under different scenarios.

Objectives of Alignment

The objectives of alignment, discussed in Morrison (2006) and O'Donoghue (2010) serve as the basis of our evaluation criteria. From a practical point of view, a "good" alignment algorithm should be able to

a) Replicate as close as possible the external control totals for the alignment totals. This is one of the main reasons why alignment is implemented in microsimulation and the common goal of all alignment methods as discussed virtually all alignment papers, e.g. Neufeld (2000), Morrison (2006)

b) Retain the relationship between the deterministic and explanatory variables in the deterministic component of the model (O'Donoghue 2010). In achieving the external totals, the alignment process should not bias the underlying relationship between the dependent and explanatory variables.

c) Retain the shape of distributions in different subgroup and inter-relations unless there is a reason not to do it. Morrison (2006) suggests that alignment is about implementing the right numbers of events in the right proportions for a pool's prospective events, as opposed to simply getting the right expected numbers of events. Although alignment processes focus on the aggregated output, it should not significantly distort the relative distribution within different sub-groups. For instance, if we want to align the number of people in work, we not only want to get the numbers right at the aggregate level, but also at the micro/meso level, e.g. the labour participation rate for 30 years old should be higher than the rate for the 80 years old. This relative distribution should not be changed, at least substantially, by the alignment method. A highly distorted alignment process would adversely affect the distributional analysis, a typical usage of microsimulation models.

d) Compute efficiently. There is no doubt that today's computing resources have been more much more abundant that ever. However, when handling large dataset, e.g. full population dataset, computational constraint is still an important issue. Some projects, e.g. LIAM2/MiDaL (Liegeois, 2010), redesign the entire framework in order to achieve faster speed and accommodate larger datasets.

Indicators of alignment performance

In order to assess the alignment algorithms with very different designs, the paper uses a set of quantitative indicators that can measure the simulation properties according to the criteria discussed earlier. The indicators include

- A general fit measure: a false positive rate $\Pr(Y=1|0)$ and a false negative rate $\Pr(Y=0|1)$, which reflect how well the prediction fit the actual data in general.
- A target deviation index (TDI), which measures the difference between the external control and the simulation outcome. This indicator is directly linked to the first criterion.
- A distribution deviation index (DDI), which measures the distortion of the relationship between different variables and inter-relations, as discussed in criteria two and three.
- And a computational efficiency measurement: The number of seconds it takes to execute one round of alignment as outlined in criterion four.

Target Deviation Index (TDI)

Assuming among *N* observations, the ideal number of events is *T* and the actual simulated number of events after alignment is *S*. Target Deviation Index (TDI) is defined as

$$TDI = \frac{T-S}{N} \qquad (7)$$

It is a percentage number ranged 0 to 1, and shows how the alignment replicates the external control. Higher values imply the outcome is further away from the external control. It is a straightforward indicator to evaluate the first criterion.

Distribution deviation index (DDI)

In order to evaluate the second and the third criteria, it is necessary to find an indicator that can reflect how well the relationships are preserved and how different the new distribution is from the old one.

A first method could be to compare the original coefficients with re-estimated coefficients from aligned data. Statistically identical coefficients indicate that the relationship remains the same, at least mathematically. However, this might not be applied to alignment tests as alignment itself, by definition, distorts the original probabilities. The coefficients, as a result, are bound to change even under an optimal alignment, and in most cases, the “correct” aligned coefficients are not available.

A second method to compare the relationships is to see whether the distribution of key variables have changed after alignment, e.g. whether the proportion of male workers and females workers have changed substantially. A Chi-square test could be useful for this scenario, as it is frequently used to test whether the observed distribution follows the theoretical distribution. It is defined as

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \quad (8)$$

Nevertheless, the test itself is not designed for binary values and requires "no more than 20% of the expected counts to be less than 5 and all individual expected counts are 1 or greater" (Yates, Moore & McCabe, 1999). This requirement might not be always fulfilled in microsimulation depending on the scenario assumptions and the way groups are defined. As a result, an adaptation is required in order to best measure the deviation between two distributions for the purpose of binary variables and possibly low or zero expected counts.

This paper proposes a self-defined distribution deviation index (DDI) to evaluate the second and third criteria in choosing an alignment method. Assuming we are going to evaluate the distribution distortion in a single alignment pool via a grouping variable X. X could be anything like age, gender, or age gender interaction etc. N Observations are divided among $n(X)$ cells. S_i is the mean value of events occurrence after alignment in group i, and O_i is the observed value in the base dataset. If we define R as the alignment ratio used in the aligning process, $O_i R$ would represent the expected value after alignment. A distribution deviation index (DDI), therefore, can be defined as

$$DDI = \sum_{i=1}^{n(X)} \frac{N_i}{N} \left(\left(S_i - O_i R \right)^2 \right) \quad (9)$$

This indicator describes how well the micro-simulated data retain the relationships between dependent variable and variable X. It is a minimum distance estimation tailored for binary variable outcome in a simulation.

Essentially, DDI calculates the sum of squares of differences weighted by the number of observations. It measures the differences between distributions before and after alignment in multiple dimensions, depending on the vector X. When X is an independent variable, it measures the distortion introduced between the independent variable and the dependent by alignment. When X is the dependent variable, DDI reports the degree of nonlinearity in the probability distortion of alignment. When X is a variable outside of the equation, DDI assesses the level of distortion in an implicit relationship. In short, X could be a vector consisting of any variable and interaction terms.

The indicator is positively correlated with the alignment deviation, it increases when the aligned distribution departs from the original and decreases when the distributions are getting alike. The scale of the indicator is independent to the choice of variable X and the number of groups that X may produce. Since S_i and O_i are both probabilities between 0 and 1. DDI has a range of 0 to 1. When the dataset preserves the shape of distribution perfectly, the index has a value of 0. It increases when the difference of two redistributions grows, with a maximum value of 1.

Computation efficiency

The most intuitive indicator for the computational efficiency of an alignment algorithm is the execution time: the length of time an alignment method takes to execute one round of alignment with input in randomised order. In order to have comparable inputs and outputs, all methods are required to retain the initial order of inputs. This makes the algorithm ready as a module in the microsimulation model. However, this extra requirement penalizes the speed of the methods that require randomly shuffling, as the observations need to be re-sorted before the end of the execution.

The evaluation of the computational efficiency is performed in Stata because of its easy integration of estimation and simulation. Given that the computer speed varies much, the results presented in this paper may change dramatically on a different platform although we would expect the relative ranking to remain stable in most cases.

Alignment algorithms evaluated

This paper evaluates all alignment algorithms discussed earlier, which includes,

- Multiplicative scaling
- Sidewalk Hybrid with Nonlinear Adjustment
- Central Limit Theorem Approach
- Sort by predicted probability (SBP)
- Sort by the difference between predicted probability and random number (SBD)
- Sort by the difference between logistic adjusted predicted probability and random number. (SBDL)

When implementing *Sidewalk Hybrid with Nonlinear Adjustment*, there are two important parameters required, η and λ. η is the maximum allowed difference between the actual number of events and the expected number of events before λ is added or subtracted from predicted probability. In this paper, η is set to 0.5 and λ is set to 0.03, which are the same values that DYANCAN model used. (Neufeld, 2000) The order of initial input is shuffled in order to get rid of undesired serial correlation.

IV. DATASETS AND SCENARIOS IN ALIGNMENT ALGORITHM EVALUATION

In order to understand the simulation properties of alignment algorithms, this paper evaluates the performances of various methods under two settings, a “lab setting”, where synthetic dataset is used, and a “real-world setting”, where the algorithms are applied to a real world dataset. This setup makes it possible to examine the performances of the alignment methods under different scenarios.

This paper starts the evaluation by using synthetic datasets in a controlled setting. Alignments are used to correct some artificial “errors” in the outcome of the statistical model. Since it is possible to control the exact source of the error in a synthetic dataset, we could analyse the simulation properties of different alignment algorithms and the probabilities transformation in a fully transparent setup.

Synthetic dataset based evaluation tests the alignment performances of different models in four different scenarios. Each scenario represents a potential statistical error that alignment methods try to address or compensate in a microsimulation model. The quality of the alignment is measured by the target deviation index (TDI), and the distribution deviation index (DDI), where the grouping variable X is the percentile of the correct probabilities. Computation cost is measured by the number of seconds the algorithm takes to execute one run.

Baseline scenario

Assuming there is a binary model expressed as following

$$y_i = \text{logit}^{-1}(\alpha + \beta x_i + \varepsilon) \qquad (10)$$

α, β are the parameters in the equation, and ε is an error term which follows a logistic distribution with zero mean and a variance of $\pi / \sqrt{3}$ .To simplify the calculation in the evaluation, we assign $\alpha = 0$, $\beta = 1$. x is randomly drawn from a standard normal distribution $N(0,1)$. The number of observation in the synthetic dataset is 100,000. Table 1 lists all the key statistics in the baseline scenario and Figure 1 illustrates the distribution of the baseline probabilities.

First scenario: Sample bias

In the first synthetic test scenario, we try to replicate an error that commonly exists in survey datasets: sample bias. Sample bias exists widely among survey datasets and it is most commonly corrected by the implementation of observation weights. Unbiased estimations of behaviour equations depend on accurate weights. Nonetheless, despite all efforts, survey datasets may still suffer from various sample bias, particularly the selection bias and the attrition bias in panel dataset, such as ECHP (Vandecasteele and Debels, 2007). Sample bias leads to a non-representative dataset, which affects the quality of simulation output. Alignment is sometimes used to compensate to the error of sample bias.

In our test, a simple sample bias is recreated. We remove 50% of the observations with positive response ($y^* > 0$) randomly from the baseline dataset. This produces a non-representative sample with the size equivalent to 75% of the original one. In other words, the observations with negative response ($y^* \leq 0$) weigh twice as much as they should in the dataset. In addition, the error structure (ε_i) have a different distribution than the baseline scenario as a consequence of the bias introduced.

Second scenario: Biased alpha (intercept)

The second synthetic scenario aims to replicate a monotonic shift of the probabilities. This is commonly used in scenario analysis, where a certain ratio, e.g. unemployment rate, is required to be increased or decreased to meet the scenario assumptions.

By manipulating the intercept of the equations, it is possible to shift the probabilities across all observations. In this scenario, α is changed to -1 while everything else is constant. The result is a monotonic, but non-uniform change in the probabilities. A non-uniform transformation is required to make sure the probabilities are still bounded within the range of [0,1]. Figure 2 demonstrates the transformation graphically. As seen, the probabilities transformation curve for the second scenario stays below 45-degree line and has a varying slope. This indicates that the transformation is monotonic but non-uniform. Contrary to the previous scenario, the error structure and the number of observations stay the same in this setup. Table 1 highlights the statistical differences between this scenario and the other ones.

Third scenario: Biased beta

The third synthetic test scenario introduces a biased slope β in the equation. This represents a change in the behaviour pattern which could not be captured at the time of estimation (e.g. the evolution of fertility pattern). In this scenario, one may assume that the behaviour pattern shifts over time. This particular setup tests on how alignment works as a correction mechanism for behaviour pattern correction.

The simulated dataset in this scenario is generated with $\beta = 0.5$, half of its value in the baseline, and therefore creates a different distribution of probability. Since x has a mean value of 0, the change does not affect the total sample mean of y at the aggregate level. The transformation would yield a different distribution but with an unchanged sample mean. Figure 1 graphically illustrates the difference in probability distribution. As seen, the standard deviation of probabilities in scenario 3 is much lower than the baseline scenario while the mean value remains the same.

Unlike the first and second scenarios, the transformation in this scenario causes a non-monotonic change in probabilities. Observations with low probability ($p < 0.5$) in baseline scenario have increased probability since their x have negative values, while the observations with high probability ($p > 0.5$) have a lower probabilities compared with the baseline scenario.

Forth scenario: Biased intercept and beta

The last synthetic test scenario combines both the change in intercept and the shift in slope. The new transformed dataset has a $\alpha = -1$ and $\beta = 0.5$. This scenario represents a relatively complex change. The change results in a lowered aggregate mean of y and a non- monotonic change in the individual probabilities.

Table 1 Overview of the Synthetic Data Scenarios

	Scenario				
Synthetic Scenario	*Baseline*	*1*	*2*	*3*	*4*
Number of observations in estimation	100,000	75,000	100,000	100,000	100,000
Number of observation in simulation	100,000	100,000	100,000	100,000	100,000
Mean value of outcome variable	0.500	0.330	0.303	0.500	0.277
α	0.000	-0.695 (0.008)	-1.000	0.000	-1.000
β	1.000	0.998 (0.010)	1.000	0.500	0.500
Target Ratio for Alignment		0.5	0.5	0.5	0.5

N.B.: Coefficients in the first scenario are estimated using logit model. Standard errors are included in the brackets.

As an overview, table 1 summarise the changes of alpha and beta in different scenario and compares the key statistics. As seen, all scenarios have the same number of observation except the first one. The mean value of outcome variable ranges from 0.277 to 0.5, and the target for alignment (external value) is 0.5 across all scenarios. Figure 1 gives a visualised picture of probability distributions in the different scenarios. We see that all probability distributions, with the exception of baseline and third scenario, exhibit a right skewed pattern. Figure 2 further compares the difference between "correct" probability and the transformed probabilities in the above scenarios.

Figure 1 Overview of Probability Distribution in Different Scenarios

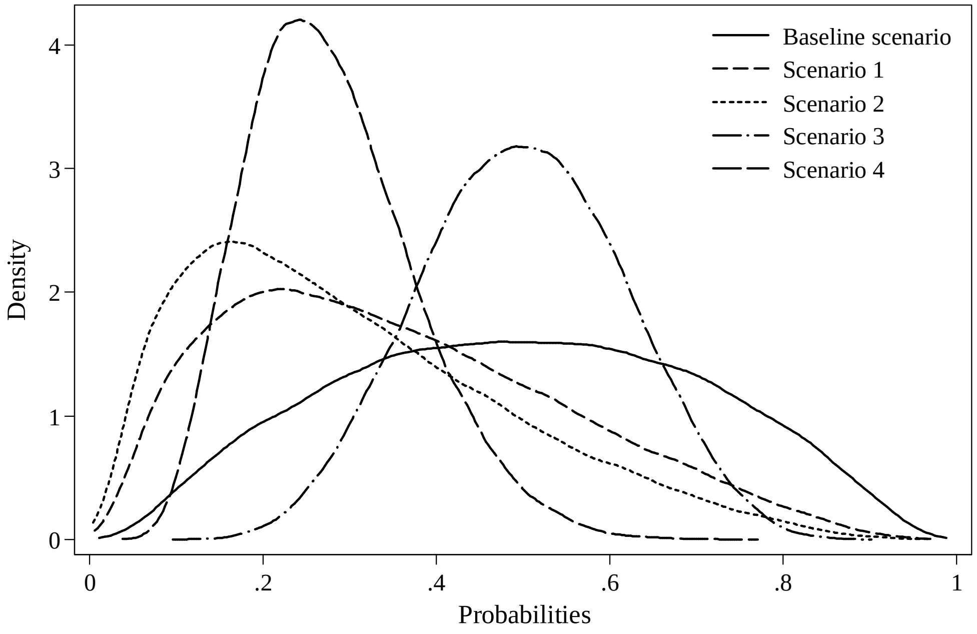
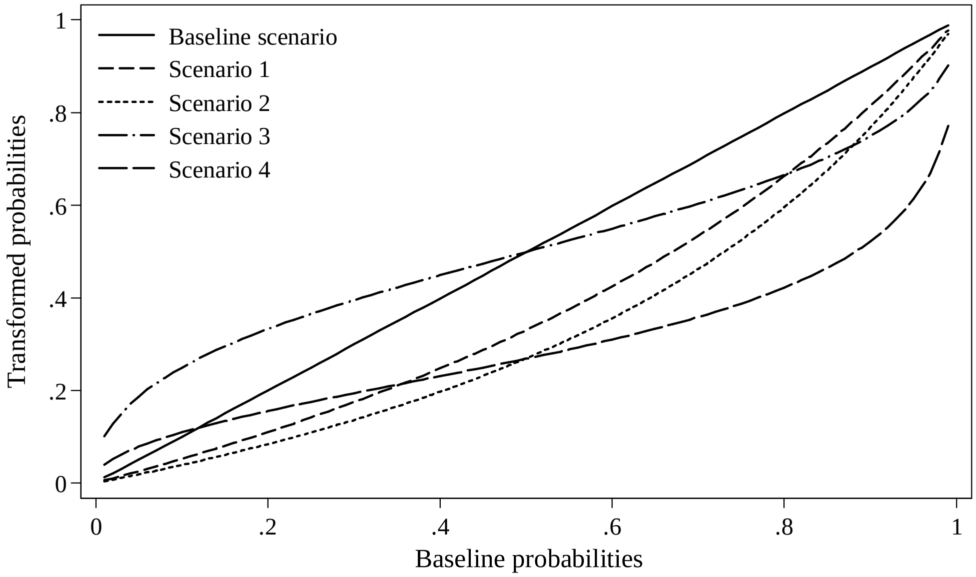


Figure 2 Overview of Probability Transformation in Different Scenarios



NB. Probability transformation curve records how probabilities change due to the artificial errors introduced in the scenario.

Evaluation using a real world dataset

There is no doubt that synthetic evaluation contributes to the understanding of alignment methods thanks to its complete transparency. An alignment algorithm, however, is only useful when applied to a real-world dataset. Therefore, this paper also analyses the performance of different alignment algorithms using a real dataset.

In this real-world evaluation, we use the 1994-2001 Living in Ireland Survey (ECHP-LII) dataset for a simple exercise of labour participation simulation. The LII survey constitutes the Irish component of the European Community Household Panel (ECHP). It is a representative household panel survey conducted on the Irish population annually for eight waves until 2001. The data contains information on demographic, employment, and other social economic characteristics of around 3500 households in each wave. In 2000, additional 1500 households were brought into the dataset to compensate for the attrition since 1994. The dataset has been cleaned and adjusted to ensure the consistency as described in Li and O’Donoghue (2010).

Labour participation simulation is selected because it is one of the popular components in dynamic microsimulation models. The simulation uses a reduced form equation for labour participation. Assuming the in-work status y_i^* is derived from following specification

$$y_i^* = \text{logit}^{-1}(\alpha + \beta X_i) \qquad (11)$$

Whereas X is a vector that covers lagged in-work status, education, gender, age, age squared, interaction term between gender and having a new-born, interaction term between marriage and gender. In the estimation, we include individuals age 15-69 with known previous working status. Table 2 provides some basic summary statistics of the variables included and estimation results are reported in appendix I.

Table 2 Overview of variables included in in-work estimation

Variable (Mean value)	*In-work*		*Out-work*	
	Mean	*Standard Deviation*	*Mean*	*Standard Deviation*
Lagged inwork status	0.86	0.32	0.14	0.31
Gender (female=1)	0.38	0.49	0.62	0.49
Age	37.18	13.18	37.60	17.79
Age squared	1555.98	1053.01	1730.53	1447.02
Having a new-born	0.03	0.17	0.02	0.12
Marriage	0.54	0.50	0.44	0.50
Secondary education	0.24	0.43	0.19	0.39
University education	0.31	0.46	0.15	0.35
Interaction term: new-born and gender	0.01	0.10	0.01	0.11
Interaction term: marriage and gender	0.18	0.39	0.33	0.47
Number of observations in the category	31784		29448	
Total number of observations	61232			

In the previous literature of microsimulation validation, Caldwell and Morrison (2000) suggest using in-sample validation, out-of-sample validation and multiple-module validation to evaluate simulation output. This paper follows a similar approach for algorithm evaluation except that there is no multi-module evaluation since alignment is usually an integrated part of a more complex model.

In-sample evaluation assesses the predictive power of the model in describing the data on which it was estimated. In this scenario, we test how well the model replicates the labour participation rate in year 1998 with known external control (observed number of workers) using different alignment methods. 1998 is selected because it is in the middle of period data covers. Equation coefficients are estimated from whole panel with the exception of first wave where lagged in-work status is not available. Alignment performance indicators are calculated in the same way as in the synthetic dataset evaluation.

An in-sample evaluation test is useful but it is different than the real microsimulation exercise where the values are predicted out of sample. An out-of-sample evaluation attempts to measure the predictive power of the model in explaining data of a similar type which were not used in the estimation of the model (Caldwell, 1996). In this particular test, we use year 1995-1998 data to predict the period 1999-2001 with the known external control (the observed number of workers) and analyse the differences in alignment methods performances. The benchmark distribution for DDI is the actual observed distribution in year 1999-2001.

V. EVALUATION RESULTS

This section reports the evaluation results of six different alignment algorithms and compares their performances under different scenarios through false positive/negative rate, two self-defined indices (TDI, DDI) and computational time.

Evaluation Results using Synthetic Datasets

Table 3 lists four key indicators obtained when evaluating using synthetic datasets,

- Target deviation index (TDI),
- False positive rate,
- False negative rate and,
- Distribution deviation index (DDI). The DDI in this synthetic dataset based test uses the percentile of dependent variable as grouping variable X.

Table 3 Properties of Different Alignment Methods in Synthetic Dataset Test

Method	*TDI*	*False Positive*	*False Negative*	*DDI*
Scenario 1: Selection Bias				
Multiplicative scaling	-0.43%	19.33%	19.76%	0.40%
Sidewalk hybrid with nonlinear adjustment	0.00%	20.63%	20.63%	0.03%
Central limit theorem approach	0.00%	19.65%	19.65%	0.43%
Sort by predicted probability (SBP)	0.00%	16.31%	16.31%	11.50%
Sort by the difference between predicted probability and random number (SBD)	0.00%	21.09%	21.09%	0.15%
Sort by the difference between logistic adjusted predicted probability and random number (SBDL)	0.00%	20.69%	20.69%	0.03%
Scenario 2: Biased Alpha (Iintercept)				
Multiplicative scaling	-1.41%	18.74%	20.15%	0.61%
Sidewalk hybrid with nonlinear adjustment	0.00%	20.69%	20.69%	0.03%
Central limit theorem approach	0.00%	19.29%	19.29%	0.65%
Sort by predicted probability (SBP)	0.00%	16.31%	16.31%	11.50%
Sort by the difference between predicted probability and random number (SBD)	0.00%	21.31%	21.31%	0.30%
Sort by the difference between logistic adjusted predicted probability and random number (SBDL)	0.00%	20.70%	20.70%	0.03%
Scenario 3: Biased beta coefficients				
Multiplicative scaling	-0.18%	22.58%	22.76%	0.90%
Sidewalk hybrid with nonlinear adjustment	-0.01%	22.59%	22.60%	0.84%
Central limit theorem approach	0.00%	22.69%	22.69%	0.91%
Sort by predicted probability (SBP)	0.00%	16.31%	16.31%	11.50%
Sort by the difference between predicted probability and random number (SBD)	0.00%	22.54%	22.54%	0.87%
Sort by the difference between logistic adjusted predicted probability and random number (SBDL)	0.00%	22.56%	22.56%	0.88%
Scenario 4: Biased alpha and beta (all coefficients)				
Multiplicative scaling	0.18%	21.57%	21.39%	0.26%
Sidewalk hybrid with nonlinear adjustment	0.00%	22.45%	22.44%	0.85%
Central limit theorem approach	0.00%	21.54%	21.54%	0.28%

Sort by predicted probability (SBP)	0.00%	16.31%	16.31%	11.50%
Sort by the difference between predicted probability and random number (SBD)	0.00%	22.97%	22.97%	1.33%
Sort by the difference between logistic adjusted predicted probability and random number (SBDL)	0.00%	22.67%	22.67%	0.92%
Average Performances				
Multiplicative scaling	-0.46%	20.55%	21.02%	0.54%
Sidewalk hybrid with nonlinear adjustment	0.00%	21.59%	21.59%	0.44%
Central limit theorem approach	0.00%	20.79%	20.79%	0.57%
Sort by predicted probability (SBP)	0.00%	16.31%	16.31%	11.50%
Sort by the difference between predicted probability and random number (SBD)	0.00%	21.98%	21.98%	0.66%
Sort by the difference between logistic adjusted predicted probability and random number (SBDL)	0.00%	21.66%	21.66%	0.46%

As seen in table 3, all alignment methods except *multiplicative scaling*, in all scenarios, have less than 0.01% deviation from the target number of event occurrence while *multiplicative scaling* shows a deviation up to 1.41% from the target during the evaluation. The result is largely driven by the design of the algorithm, as *multiplicative scaling* cannot guarantee a perfect alignment ratio although the expected deviation is zero. *Sidewalk hybrid* sometimes has a slight deviation (less than 0.01%), as the non-linear transformation may not be always perfect under existing implementation[6]. *Central limit theorem* methods have built-in counters that prevent the events from manifesting when the target is met. Sorting based algorithms only pick the exact number of observations required, which is why their target deviation index (TDI) is always zero.

In terms of false positive and false negative rates when compared with the "correct" values, alignment method *SBP* yields the best result, which is on average 4 to 6 percentage points lower than other algorithms, as shown in the tables. *Sidewalk Hybrid*, together with *SBD*, *SBDL,* have the highest false positive/ false negative rates on average. It seems that the false positive and false negative rates are closely related to the complexity of the algorithms. The "nonlinear transformation" in *Sidewalk Hybrid* and "differencing" operations in *SBD* and *SBDL* are both more computationally complicated than the other methods. This pattern is consistent across all scenarios, though absolute numbers fluctuate across different scenarios.

Whilst false positive and false negative is a useful indicator when the correct value is known, it is a less critical indicator for simulation as microsimulation exercises tend to focus more on the distributions. Therefore, the distribution deviation index (DDI) is particularly important in judging how well the relative relations between variables are preserved after alignment. Appendix 2 visualises the difference between actual probabilities and aligned probabilities in all synthetic tests.

[6] The process usually requires several iterations and it is computationally expensive (Neufeld, 2000). Our test model used in this paper stops its calibration when the iteration only improves the average probability by no more than 10^{-8}. This increases the calculation speed but sometimes results in imperfectly aligned probabilities. Details of the calibration steps can be found in the book published by Society of Actuaries (SOA, 1998).

The results show that *SBP* method heavily distorts the original distribution of the probabilities across all scenarios using percentile grouping. This is also reflected by distributional deviation index (DDI), which is effectively calculating a weighted size of the gap in this case. It seems that there is no method consistently outperforming across all scenarios. In the first two scenarios, s*idewalk hybrid* and *SBDL* method gives the best result; In the third scenario, where the synthetic dataset modifies the slope of x_i, all methods have similar DDI values except SBP; In the last scenario, *multiplicative scaling* and *central limit* methods generally perform much better than the rest. Compared with other methods, methods which involves "differencing" and "logistic transformation" (incl. sidewalk hybrid with non-linear transformation, SBD and SBDL) seem to be more sensitive to the change in the beta coefficient. Their performances are much better when beta remains stable, e.g. scenario 1 and 2. This may be due to the nature of these algorithms as the "differencing" and "logit transformation" operations assume monotonic changes in the probabilities.

Evaluation Results using a Real-world Dataset

The synthetic dataset based evaluation offers an overview of the performances of different algorithms under particular source of noise, but the performance with real-world dataset is more interesting for empirical modellers. Table 4 reports all the key indicators calculated when applying alignment in a real life dataset with the example of estimating in-work population. DDI is calculated based on independent variables, including sex, education, marriage status with childbirth interaction, and external variable, nationalities. It reflects an overall shift of the distribution in multi-dimensions.

Table 4 Properties of Different Alignment Methods with a Real World Dataset (LII)

Method	*TDI*	*False Positive*	*False Negative*	*DDI*
In-Sample Evaluation				
Multiplicative scaling	0.24%	10.00%	9.76%	0.62%
Sidewalk hybrid with nonlinear adjustment	0.01%	9.47%	9.45%	0.64%
Central limit theorem approach	0.00%	9.57%	9.57%	0.62%
Sort by predicted probability (SBP)	0.00%	5.86%	5.86%	0.62%
Sort by the difference between predicted probability and random number (SBD)	0.00%	9.64%	9.64%	0.62%
Sort by the difference between logistic adjusted predicted probability and random number (SBDL)	0.00%	9.60%	9.60%	0.67%
Out-of-Sample Evaluation				
Multiplicative scaling	0.10%	11.24%	11.14%	0.75%
Sidewalk hybrid with nonlinear adjustment	0.00%	11.04%	11.04%	0.68%
Central limit theorem approach	0.00%	11.12%	11.12%	0.74%
Sort by predicted probability (SBP)	0.00%	7.63%	7.63%	1.47%
Sort by the difference between predicted probability and random number (SBD)	0.00%	11.14%	11.14%	0.66%
Sort by the difference between logistic adjusted predicted probability and random number (SBDL)	0.00%	11.03%	11.03%	0.76%

N.B.: In-sample evaluation predicts 1998 in-work using 1995-2001 data
Out-of-Sample evaluation predicts 1999-2001 in-work using 1995-1998 data

Similar to the results from synthetic dataset, *multiplicative scaling* is the only method with a TDI greater than 0.01% and the *SBP* method outperforms all other methods in terms of false positive and false negative rates at a significant margin. All other evaluated methods have similar false positive and negative rates.

As to the DDI, there is no dramatic difference between different methods in in-sample evaluation. We notice that the *SBP* method has a much more comparable DDI performance in the real life dataset than in the synthetic dataset. In fact, *SBP* has one of the best results in in-sample evaluation. In the out-of-sample exercise, we find that the *SBD*, a method with average performance with synthetic datasets, has the lowest DDI value, while *SBP* has the worst result. Besides the algorithm design, the change of grouping variables also affects the observed DDI pattern in this evaluation. With the synthetic datasets, groups are divided based on the percentile value of the dependent variable while in the real-world dataset, observations were grouped using a realistic setting, using different characteristics variables, like age, gender etc.

Computing Performance and Scalability

Computational efficiency is another main criterion for evaluating alignment algorithm. Given the increasing availability of large-scale datasets in microsimulation and the model complexity, alignment may consume considerable resources in the computation processes. Nonetheless, the study of the computational efficiency is rather scarce in the field of microsimulation and there is no paper so far analysing how the number of observations affect the algorithms' performance. This section compares different alignment algorithms in terms of computation efficiency and discusses the issue of scalability of the algorithms.

Table 5 shows an overview of the computation time required during the synthetic scenario test and real-world data test. The computational premium is timed on an Intel i5-520m processor when only single core is used. As indicated, the method that takes least computation resources is *multiplicative scaling* method. This is not surprising, as *multiplicative scaling* involves only a single calculation for each observation. Sorting-based alignment methods seem to be in the next tier, which consume up to 5 times more resources compared with *multiplicative scaling*. The variations in sorting method does not change the execution time much although the last sorting variation, *SBDL*, consumes around 10% more resources than the other sorting based algorithms due to its higher computation complexity.

Sidewalk Hybrid with nonlinear transformation seems to be on the bottom list in terms of the efficiency. It takes about 80 times more CPU time than what the fastest method, *multiplicative scaling*, requires, and 15-20 more CPU time than the sorting based algorithms. There are three reasons for its relatively poor performances. Firstly, the nonlinear transformation may take many iterations and it is computational expensive (Neufeld, 2000). Secondly, the method itself suffers from serial correlation in the original design, as the calculation is dependent on the result of the last observation. In order to mitigate this effect, an extra randomisation via sorting is implemented. This is accompanied by a reverse process, which restores the original order of the input at the end of the alignment. Thirdly, the Sidewalk method requires iterating through observations. Stata, which is the platform of our evaluation, is not particular efficient at individual observation iteration compared with the batch processing for which Stata optimises[7]. This is also the primary reason why *Central limit theorem approach* has a relatively long running time. We speculate from a theoretical point of view, that the performances of *the Sidewalk method* and *the Central limit theorem approach* could be significantly improved when implemented correctly as native code in C/C++ as compiled code does not re-interpret the syntax over the iterations. Nonetheless, *sidewalk method* may still be slower than the other algorithms when nonlinear probability transformation is applied.

Table 5 Computational Costs for Different Alignment Methods

	Synthetic Dataset Scenario				Real-world Dataset	
Method	1	2	3	4	In-Sample	Out-Sample
Multiplicative scaling	0.07	0.07	0.07	0.07	0.04	0.13
Sidewalk hybrid with nonlinear adjustment	5.71	5.88	5.49	5.78	1.30	4.22
Central limit theorem approach	3.34	3.40	3.50	3.55	0.63	2.12
Sort by predicted probability (SBP)	0.32	0.33	0.33	0.35	0.17	0.58
Sort by the difference between predicted probability and random number (SBD)	0.34	0.34	0.34	0.34	0.18	0.61
Sort by the difference between logistic adjusted predicted probability and random number (SBDL)	0.36	0.36	0.36	0.38	0.18	0.63

When increasing the number of observations, i.e. size of input, all algorithms exhibit a mostly linear growth rate of the execution time in Stata (See figure 3 to figure 5) for a dataset under 15 million observations. The run-time seems to be directly proportional to its input size. All alignments are using the same input dataset, which is a randomly generated pool of uniformly distributed probabilities. The linear growth rate indicates that Stata might use a non-comparison sorting algorithm, e.g. Radix sort, in its default implementation.

[7] Observation iteration, a necessary step for these two algorithms, tends to be very slow in Stata because loops are reinterpreted at each iteration. Stata recommends using compiled plug-in for the best performance for this type of scenarios (Stata, 2008). However, algorithm specific optimization using compiled code is beyond the scope of this paper and it would make the comparison difficult.

Figure 3 Computational Time Curve of Multiplicative Scaling Alignment

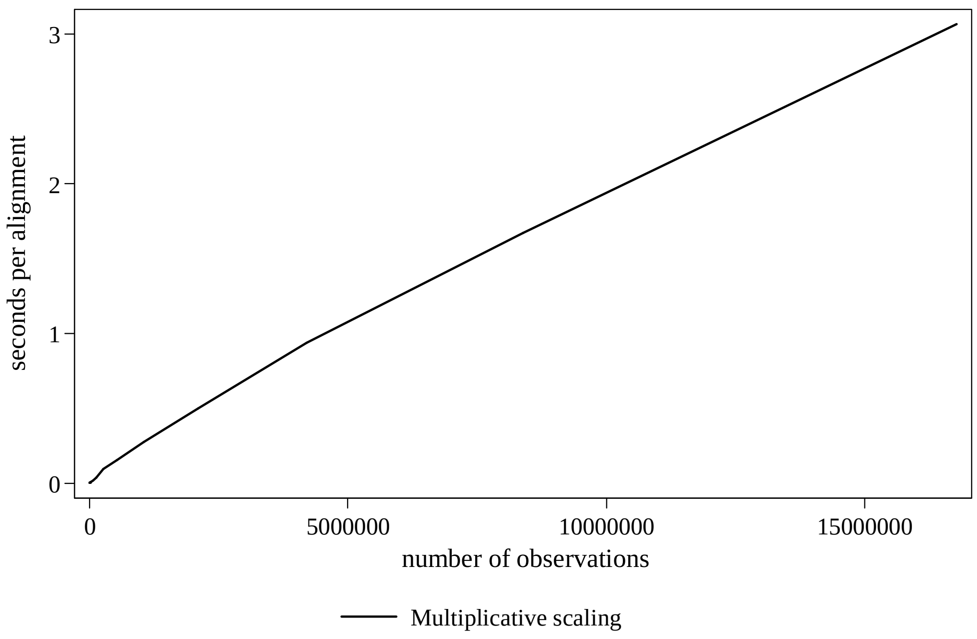


Figure 4 Computational Time Curve of Sidewalk and Central limit theorem approach

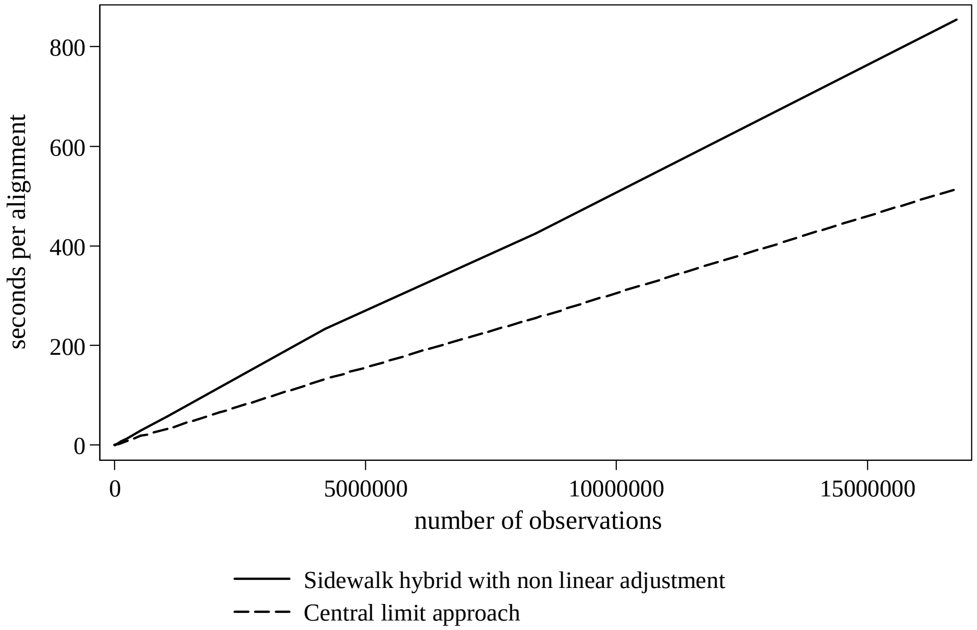
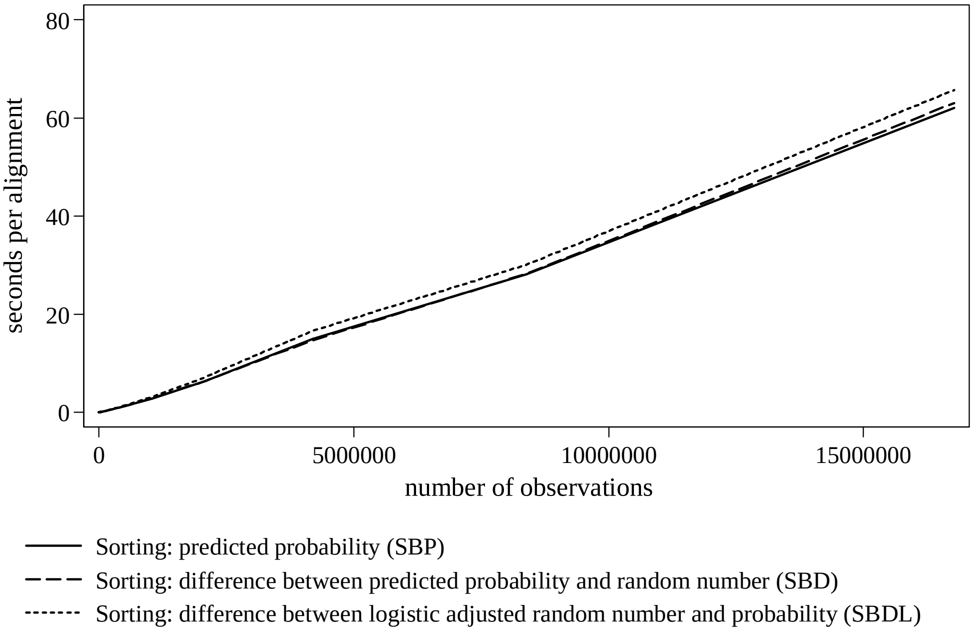


Figure 5 Computational Time Curve of Sorting Based Alignment Algorithms



Due to the actual implementation in different environment may vary, the results do not reflect the performance in real projects on a different platform, but do provide a reference to illustrate the potential computation cost. It is important to note that since the sorting algorithm and most calculations are encapsulated in Stata, the actual performance is the mixed result of Stata performance, algorithm design quality, and implementation quality. The actual performance may be very different in other implementation settings (e.g. C/C++). Results are timed with the internal timer from Stata on a windows box.

VI. CONCLUSION

Calibrating results of a statistical forecasting model, which is known as alignment, is de facto widely adopted over the past decade in the field of microsimulation despite its controversy. Microsimulation models uses alignment for various purposes, e.g. historical data alignment in CORSIM, forecasting alignment in APPSIM etc. Although alignment cannot be used as a replacement of a well-specified statistical model, it is an effective pragmatic solution to undertaking analyses of complex phenomena such as the performance of pension systems within a highly complex context of evolving social and economic change. Many alignment methods have appeared in the literature as the development of dynamic microsimulation progressed.

This paper fills a gap in the literature in relation to the evaluation of different alignment algorithms. Although pervious literatures, e.g. Johnson (2001), Morrison (2006), and O’Donoghue (2010) have listed a few criteria that a “good” alignment method should meet, and analysed some theoretical expectation of the alignment simulation properties and their performances, e.g. Morrison (2006), there was no direct or quantitative comparison of various methods.

In this paper, we have reviewed and evaluated most binary model alignment techniques, including multiplicative scaling, hybrid sidewalk method, central limit theorem approach and sorting based algorithms (including its variations). The paper compares different algorithms through a set of indicators including false positive rate, false negative rate, self-defined target deviation index (TDI), distribution deviation index (DDI), and computation time. Target deviation index (TDI), gives a scale independent view on how well an alignment method replicates the external control. The false positive, false negative rate, give an overview on the general quality of the output after alignment. The preservation of inter-correlations is measured by the distribution deviation index (DDI), an indicator ranged 0 to 1. It calculates the distance between the ideal distribution and the actual distributions after the alignment.

The evaluations report a mixed result of alignment performances. It shows that the selecting the "best" alignment method is not only about the algorithm design, but also the requirements and reasoning in a particular scenario.

Overall speaking, *multiplicative scaling* is the easiest to implement, and fastest to compute method for alignment. It could align more than 3 million observations in less than 1 second on a laptop computer in 2010. Nonetheless, it cannot perfectly align to external control as the events are calculated purely based on the calculated probabilities. Moreover, due to lack of restrictions in the algorithm design, the outcome produced by the multiplicative scaling method is subject to higher fluctuations than by other methods.

Sidewalk hybrid with nonlinear adjustment is a very computationally expensive method due to its nonlinear adjustment. However, the method has an above average performance in all scenarios. It exhibits a similar pattern with one sorting based method, *sort by the difference between logistic adjusted predicted probability and random number (SBDL)*. Because of the logistic transformation applied in both algorithms, both methods are good at handling the error of intercept in logit model.

Central limit theorem approach tends to have similar statistical patterns with *multiplicative scaling* method except it can match the alignment target more precisely. The method exhibits an above average performance in the evaluations with the real world dataset, although it performs poorly in the first scenario with synthetic data, where the intercept in the equation is shifted. Nonetheless, the algorithm is very slow when implemented in Stata due to the need of observation iteration.

As to the sorting based algorithms, the *sort by probabilities (SBP)* method yields the best result in terms of false positive and false negative whilst it distorts the internal distributions heavily in most cases. This is due to the nature of the algorithm, which over-predicts the observations with higher probabilities and under-predicts the observations with lower probabilities. However, the method is easy to implement and does not involve random number sorting. Its simulation properties suggest that *SBP* is a good method in imputation, but not ideal for forward or backward simulation.

Sort by the difference between predicted probability and random number (SBD) and *Sort by the difference between logistic adjusted predicted probability and random number (SBDL)* are similar in terms of computation steps, but they produce very different distributions of probability. *SBDL* works better with logit model, especially when the intercept is used for alignment calibration. *SBD* seems to have below average performances when looking at all indicators and scenarios.

As the results show, the selection of alignment methods is a more complicated than previously thought. Each algorithm has its own advantages and disadvantages. For a microsimulation project that is speed oriented, *multiplicative scaling* seems to be a good choice. *Central limit theorem approach* could also be considered when implemented in a compiled language, like C/C++. In a project where speed is not the major concern, the choice might depend on the reason for alignment. For instance, if alignment is used to create a shift in intercept, *SBDL* or *sidewalk hybrid with nonlinear transformation* may be the best choice. In addition, for microsimulation analysis with the focus on distributional analysis, *SBP* may not be the ideal because of its distortion of distributions.

Understanding the simulation properties is not an easy job as there are many implicit and explicit assumptions in every simulation project. The evaluation method used in this paper also has its own limits. In the synthetic dataset based tests, the evaluations only cover the most common scenarios. However, the sources of errors in a real simulation are more complex than what has been illustrated and the distribution of independent variables, e.g. normal distribution, may not be always true. Further work is required to understand the simulation properties of different methods under different assumptions and more complicated error structures. In addition, algorithms should also be evaluated on more panel datasets with stripped-down microsimulation models in order to understand the impact of alignments in real-life projects.

RERFERENCE

Anderson, J.M., (1990) Micro-Macro Linkages in Economic Models, in Lewis, G.H., Michel, R.C. (eds.) Microsimulation techniques for tax and transfer analysis. Washington DC: Urban Institute.

Bacon, P. B. (2009) Microsimulation, Macrosimulation : model validation, linkage and alignment, NATSEM Working paper

Baekgaard, H. (2002) Micro-macro linkage and the alignment of transition processes : some issues, techniques and examples, National Centre for Social and Economic Modelling Technical paper

Bardaji, J., B. Sédillot and E. Walraet. (2003) Un outil de prospective des retraites: le modèle de microsimulation Destinie, Économie et prévision pp. 193-214.

Caldwell S. and R. Morrison, (2000) Validation of longitudinal microsimulation models: experience with CORSIM and DYNACAN, in Mitton et al. (eds.) Microsimulation in the New Millennium, Cambridge: Cambridge University Press.

Caldwell S., Favreault M., Gantman A., Gokhale J., Johnson T. and Kotlikoff L.J. (1998) Social Security's Treatment of Postwar Americans, NBER Working Paper No. W6603

Chénard, D. (2000a) Earnings in DYNACAN: distribution alignment methodology, Paper Presented to the 6th. Nordic Workshop on Microsimulation, Copenhagen, June.

Chénard, D. (2000b) Individual alignment and group processing: an application to migration processes in DYNACAN D. in Mitton, L., Sutherland, H. and Weeks, M. Microsimulation Modelling for Policy Analysis: Challenges and Innovations. Cambridge: Cambridge University Press.

Curry, C. (1996) PENSIM: A Dynamic Simulation Model of Pensioners' Income, in Government Economic Service Working Paper No. 129, London: Analytical Services Division , Department of Social Security.

Davies, J.B., (2004) Microsimulation, CGE and Macro Modelling for Transition and Developing Economies. UNU/WIDER research paper

Duncan, A., & Weeks, M. (1998). Simulating transitions using discrete choice models, In Proceedings of the American Statistical Association (Vol. 106, p. 151–156)

Flood, Lennart, Fredrik Jansson, Thomas Pettersson, Olle Sundberg, and Anna Westerberg. (2005) SESIM III – A Swedish dynamic micro simulation model.

Flood, L. (2007) Can we Afford the Future? An evaluation of the new Swedish pension system, Modelling our future: population ageing, social security and taxation pp. 33.

Harding, A. (2007) Challenges and Opportunities of Dynamic Microsimulation Modelling": Citeseer.

Johnson, T., (2001) Nonlinear Alignment by Sorting, CORSIM Working Paper

Kelly, S. and R. Percival (2009) Longitudinal benchmarking and alignment of a dynamic microsimulation model, IMA Conference Paper.

Li, J. & O'Donoghue, C. (2010) Simulating Histories using Household Survey Dataset, Working paper

Liegeois, P. (2010) MiDaL Project Meeting Minutes, MiDaL Project Paper

Mantovani, D., F. Papadopoulos, H. Sutherland and P. Tsakloglou. (2007) Pension incomes in the European Union: policy reform strategies in comparative perspective, Micro-simulation in action: policy analysis in Europe using EUROMOD pp. 27.

Morrison, R. (2006) Make it so: Event alignment in dynamic microsimulation. DYNACAN paper.

Neufeld, C., (2000) Alignment and Variance Reduction in DYNACAN" in Anil Gupta and Vishnu Kapur (eds) Microsimulation in Government Policy and Forecasting. North-Holland.

O'Donoghue, C. (2010) Alignment and calibration in LIAM, LIAM working paper

O'Donoghue, C., Stephen Hynes and John Lennon (2008) The Life-Cycle Income Analysis Model (LIAM): A Study of a Flexible Dynamic Microsimulation Modelling Computing Framework", International Journal of Microsimulation.

O'Donoghue, C. (2001) Redistribution in the Irish Tax-Benefit System, PhD Thesis

Scott, A., & A. (2001) computing strategy for SAGE: 1. Model options and constraints. Technical Note 2. London, ESRC-Sage Research Group.

SOA (1997) Chapter 5 on CORSIM, Society of Actuaries, [http://www.soa.org/files/pdf/Chapter_5.pdf] accessed June 10th 2010.

SOA (1998) Chapter 6 on DYNACAN, Society of Actuaries, [http://www.soa.org/files/pdf/Chapter_6.pdf] accessed June 10th 2010.

Stata (2008) Creating and using Stata plugins, [http://www.stata.com/plugins/] accessed June 10th 2010.

Vandecasteele, L., & Debels, a. (2007) Attrition in Panel Data: The Effectiveness of Weighting. European Sociological Review, 23(1), 81-97.

Winder, N. (2000) Modelling within a thermodynamic framework: a footnote to Sanders. Cybergeo : European Journal of Geography Systèmes, Modélisation, Géostatistiques, Vol. 138, No. 05.

APPENDIX

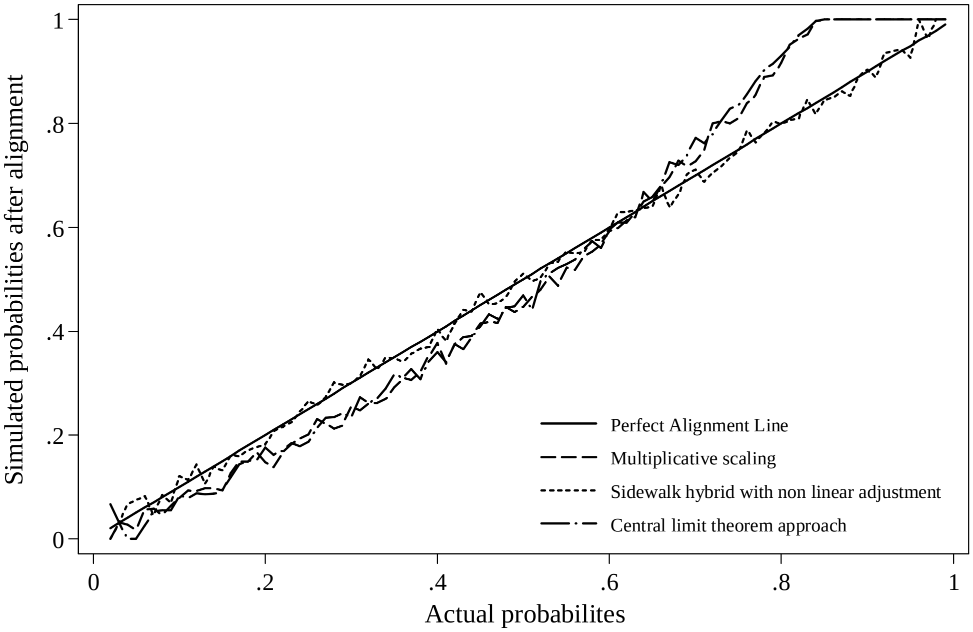
1. ESTIMATION RESULTS FOR IN-WORK VARIABLE IN LII

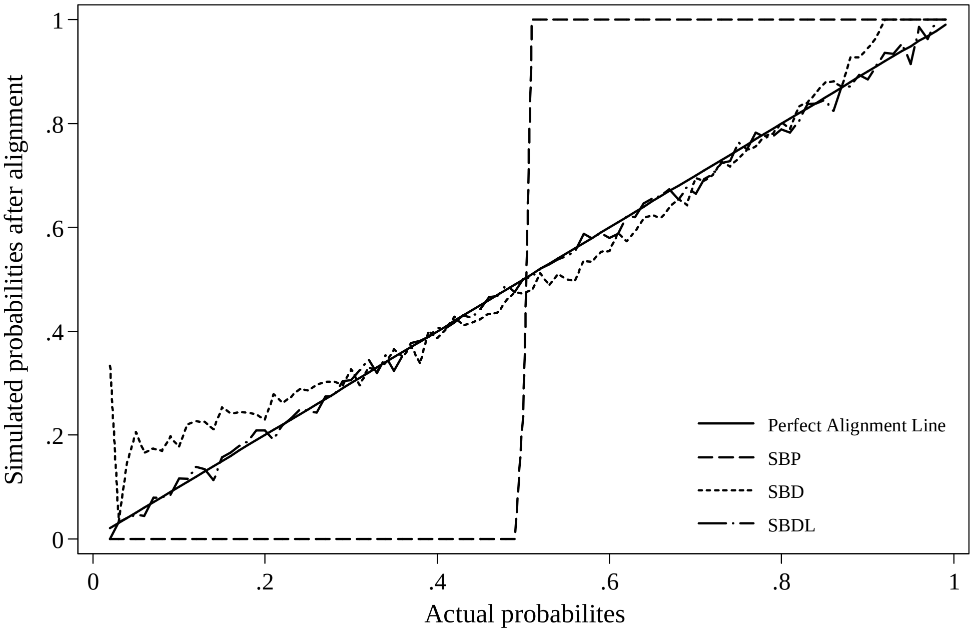
Variables	*Estimation using 1995-2001 (for in-sample evaluation)*	*Estimation using 1995-1998 (for out-of-sample evaluation)*
	Coefficients (Standard Error)	Coefficients (standard error)
Lagged inwork status	3.86 (0.03)	4.00 (0.04)
Gender (female=1)	-0.36 (0.03)	-0.46 (0.04)
Age	0.15 (0.01)	0.19 (0.01)
Age squared	0.002 (0.00)	0.002 (0.00)
Secondary education	0.96 (0.03)	1.01 (0.05)
University education	1.20 (0.03)	1.24 (0.05)
Interaction term: new-born and gender	-0.33 (0.12)	-0.25 (0.15)
Interaction term: marriage and gender	-0.44 (0.04)	-0.51 (0.06)
Constant	-4.58 (0.1)	-5.26 (0.14)
Number of Observations	61232	36053

N.B. Models were estimated using standard Logit.

2. ACTUAL VS. ALIGNED PROBABILITIES WITH SYNTHETIC DATASETS

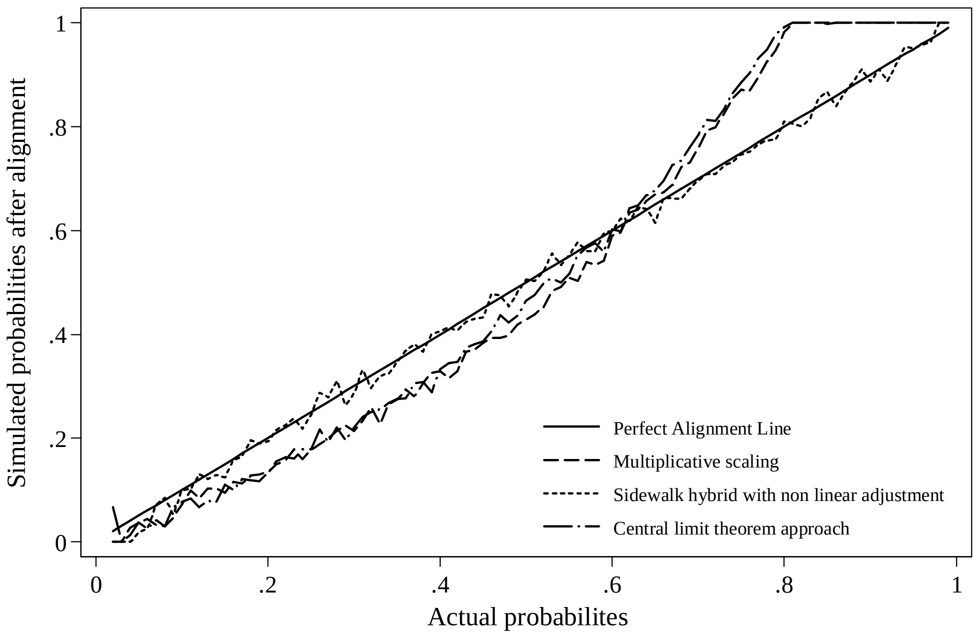
Synthetic Dataset Scenario 1: Sample bias

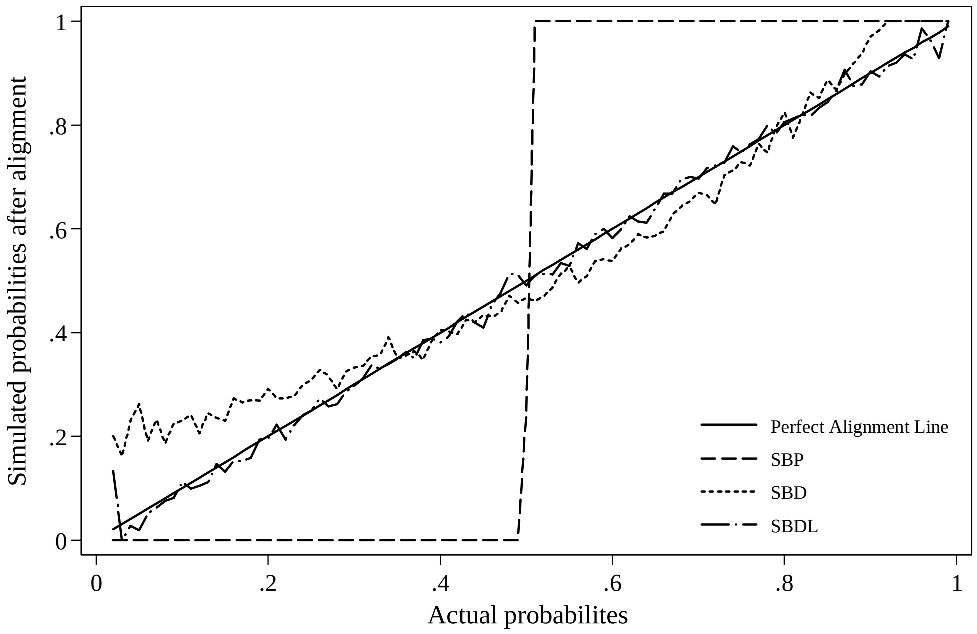




Abbreviations used in the Figure
SBP: Sort by predicted probability
SBD: Sort by the difference between predicted probability and random number
SBDL: Sort by the difference between logistic adjusted predicted probability and random number

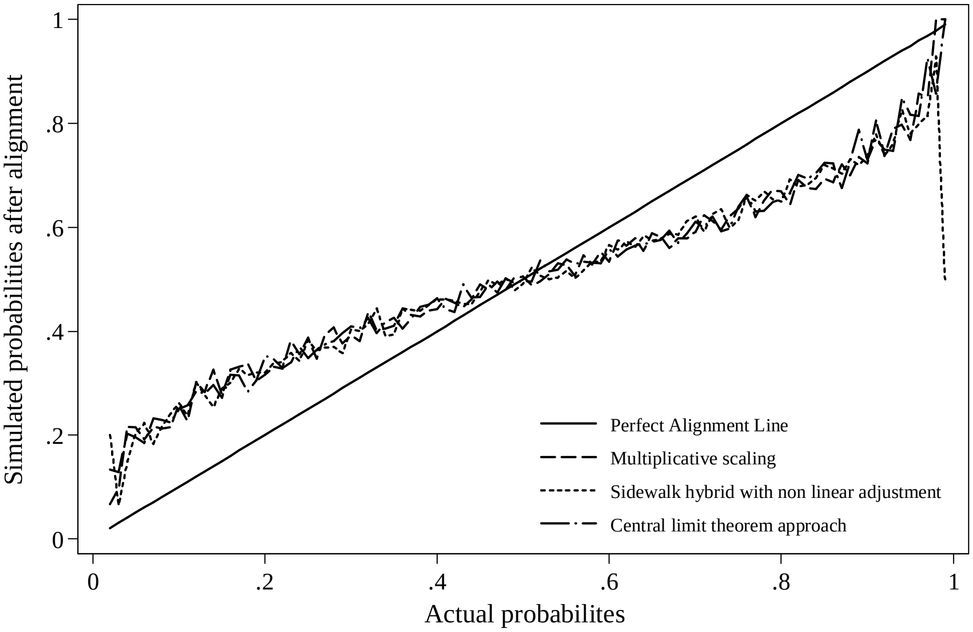
Synthetic Dataset Scenario 2: Biased Alpha (Intercept)

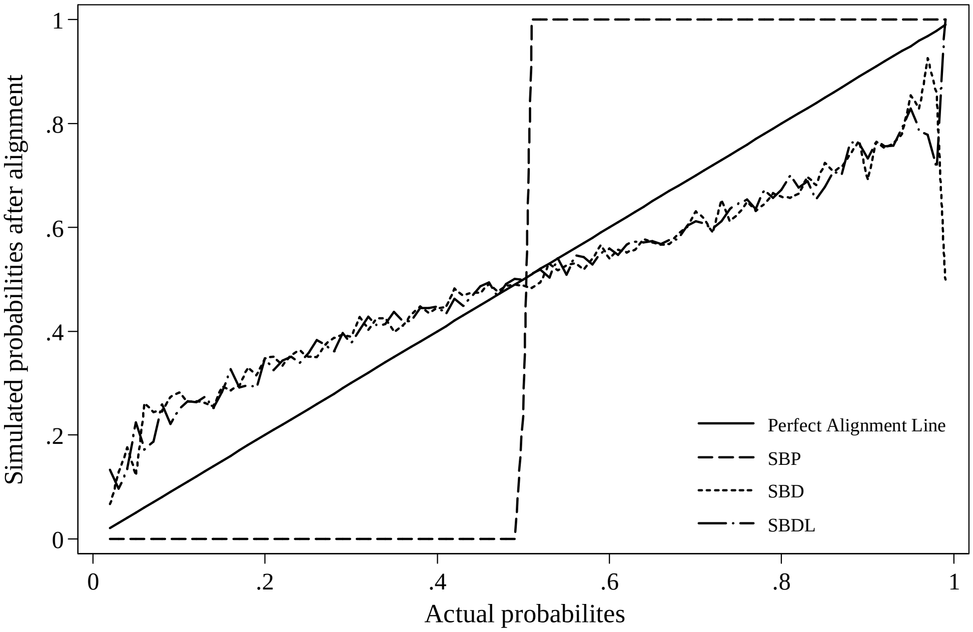




Abbreviations used in the Figure
SBP: Sort by predicted probability
SBD: Sort by the difference between predicted probability and random number
SBDL: Sort by the difference between logistic adjusted predicted probability and random number

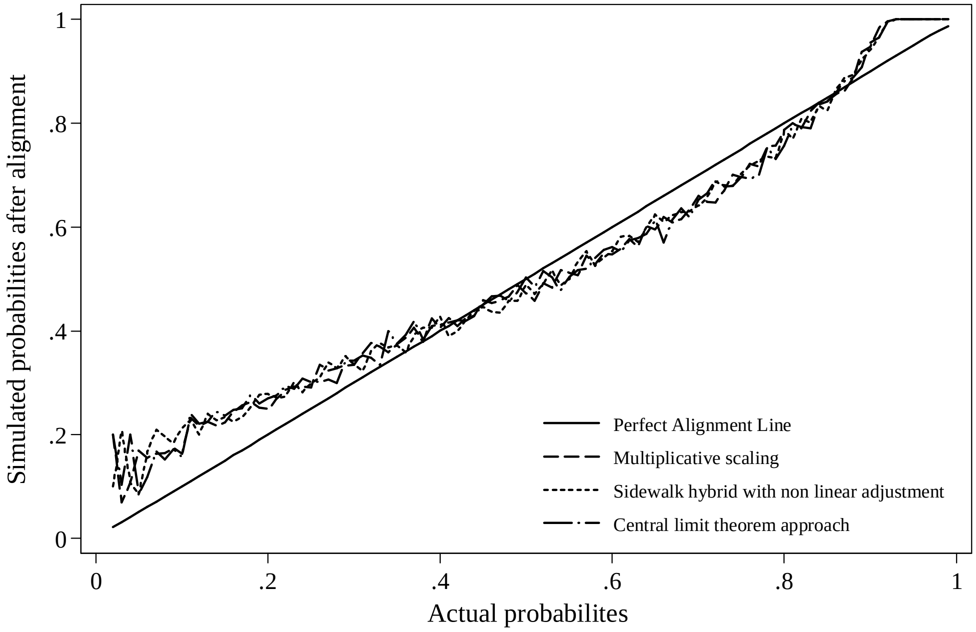
Synthetic Dataset Scenario 3: Biased Slope (Beta)

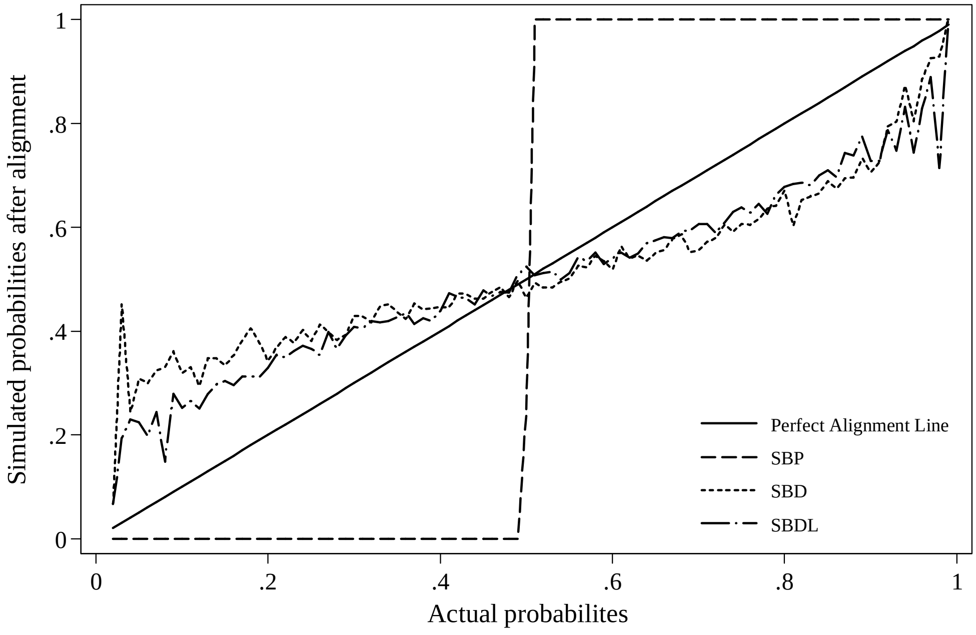




Abbreviations used in the Figure
SBP: Sort by predicted probability
SBD: Sort by the difference between predicted probability and random number
SBDL: Sort by the difference between logistic adjusted predicted probability and random number

Synthetic Dataset Scenario 4: All coefficients biased





Abbreviations used in the Figure
SBP: Sort by predicted probability
SBD: Sort by the difference between predicted probability and random number
SBDL: Sort by the difference between logistic adjusted predicted probability and random number

3. COMPUTATION EFFICIENCY AND SCALABILITY

	Computation time for N observations*			
Method	N= 65536 (2^{16})	N= 524288 (2^{19})	N= 4194304 (2^{22})	Average Computational Time per 1 million observations (seconds)
Multiplicative scaling	0.02	0.15	1.19	0.28
Sidewalk hybrid with nonlinear adjustment	3.64	29.20	233.83	55.61
Central limit theorem approach	2.03	18.21	132.27	31.55
Sort by predicted probability (SBP)	0.14	1.29	14.94	3.18
Sort by the difference between predicted probability and random number (SBD)	0.13	1.28	14.66	3.15
Sort by the difference between logistic adjusted predicted probability and random number (SBDL)	0.15	1.42	16.60	3.55

** Results obtained using Stata 11 SE on a Windows 7 box with Intel i5-520M CPU*

4. NATURAL DISTRIBUTION DEVIATION IN LII DATASET

Year	*DDI using last year distribution as benchmark value*
1995	0.60%
1996	0.46%
1997	0.76%
1998	0.69%
1999	0.92%
2000	1.11%
2001	0.65%

The UNU-MERIT WORKING Paper Series

2012-01 *Maastricht reflections on innovation* by Luc Soete

2012-02 *A methodological survey of dynamic microsimulation models* by Jinjing Li and Cathal O'Donoghue

2012-03 *Evaluating binary alignment methods in microsimulation models* by Jinjing Li and Cathal O'Donoghue