Computational Statistics manuscript No.
(will be inserted by the editor)

Online Estimation of Individual-Level Effects using Streaming Shrinkage Factors

Lianne Ippel · Maurits C. Kaptein · Jeroen K. Vermunt

Received: date / Accepted: date

Abstract In the last few years, it has become increasingly easy to collect data from individuals over long periods of time. Examples include smart-phone applications used to track movements with GPS, web-log data tracking individuals' browsing behavior, and longitudinal (cohort) studies where many individuals are monitored over an extensive period of time. All these datasets cover a large number of individuals and collect data on the same individuals repeatedly, causing a nested structure in the data. Moreover, the data collection is never 'finished' as new data keep streaming in. It is well known that predictions that use the data of the individual whose individual-level effect is predicted in combination with the data of all the other individuals, are better in terms of squared error than those that just use the individual mean. However, when data are both nested and streaming, and the outcome variable is binary, computing these individual-level predictions can be computationally challenging. In this paper, we develop and evaluate four computationally-efficient estimation methods which do not revise "old" data but do account for the nested data structure. The methods that we develop are based on four existing shrinkage factors. A shrinkage factor is used to predict an individual-level effect (i.e., the probability to score a 1), by weighing the individual mean and the mean over all data points. In a simulation study, we compare the performance of existing and newly developed shrinkage factors. We find that the existing methods differ in their prediction accuracy, but the differences in accuracy between our novel shrinkage factors and the existing methods are small. Our novel methods are however computationally feasible in the context of streaming data.

Keywords Data streams · shrinkage factors · James-Stein estimator · online learning · nested data

L. Ippel
Tilburg University, Warandelaan 2, PObox 90153, 5000 LE Tilburg, the Netherlands Tel.: +31 13 466 2959
E-mail: G.J.E.Ippel@tilburguniversity.edu

M.C. Kaptein
Tilburg University,

J.K. Vermunt
Tilburg University,

1 Introduction

Researchers often encounter *grouped* data where the outcome variable of interest is *binary*. For example, Murnaghan, Sihvonen, Leatherdale, and Kekki (2007) compared the smoking behavior (smoking versus none smoking) of students that are grouped within different schools. Quintelier (2010) studied the effect of schools on students voting behavior (vote versus no vote), and Linares, Guizar, Amador, Garcia, Miranda, Perez, and Chapela (2010) monitored children over a long period of time (repeated measurements nested within children) to investigate the effect of air pollution on the presence (or absence) respiratory symptoms. Furthermore, Cheng and Cantú-Paz (2010) studied 'click' behavior in e-commerce, i.e., whether an individual clicks on an advertisement on a website. In this latter case, the repeatedly observed click-through behavior is nested within the individual. In each and every instance above researchers are interested in obtaining good estimates of the probability of an event occurring at the level of the individual, while respecting the nested structure in the data. In this paper, we examine efficient methods of obtaining such estimates in a situation where the collected data arrive continuously and datasets are rapidly augmented.

To settle for an unambiguous terminology throughout, we adopt the terms of the latter e-commerce example. Here a researcher could be interested in the individual-level effect μ_i, which is the estimated probability that an individual will click. Note that we use i to index the grouping factor which, in this particular case of multiple observations nested within the individual, denotes the individual customer whose click-through rate is being estimated. However, the methods discussed in this paper do not restrict themselves to the nesting of observations that are nested within individuals but could also be used for groupings such as individuals within schools or schools within districts. Our interest lies in estimating the individual-level effect μ_i, accurately and computationally efficiently.

In a now classical paper, Stein (1956) showed that predicting the individual-level effects of one individual (i.e., μ_i) using only the data of this particular individual, thus without taking the other individuals into account, results in a larger average squared prediction error than when these other individuals are taken into account. He demonstrated that combining the estimated mean of an individual, which we denote p_i, with the estimated sample mean over all data points, $\bar{p}$, results in better out-of-sample predictions (see, for instance, Efron and Morris, 1977). Following this result, James and Stein in 1961 introduced the idea of a *shrinkage factor*, a way to weigh the estimated mean of an individual and the mean over data points to obtain a prediction of μ_i. The resulting weighted combination can be denoted as follows:

$$\hat{\mu}_i = (1-\hat{\beta})p_i + \hat{\beta}\bar{p}, \tag{1}$$

where β is the so-called *shrinkage factor*. Because we focus on binary outcomes, the p_i in our case denotes the proportion of (for instance) clicks. In the remainder of this paper we refer to p_i as the *individual mean*, and $\bar{p}$ as the *group mean*.

The aim of this paper is to develop and evaluate different shrinkage factors which can be used to efficiently estimate the individual-level effect in a situation where new data present themselves over time. We refer to this situation as a *data stream*. In

a data stream, the data collection is never "finished", for instance in click-behavior data on a website. In the case of real-time prediction, where up-to-date predictions of the individual-level effects are required at each moment during the stream, methods that can *update* rather than *re-estimate* the individual-level effects, greatly improve the speed of the estimation process (Pébay, Terriberry, Kolla, and Bennett, 2016).

In general, various methods are available to deal with data streams. For instance one could subsample from the data stream (i.e., at random include some of the data points in the analysis while excluding others), and analyze the subsample in order to obtain predictions (Efraimidis and Spirakis, 2006). While this method solves the problem of a growing dataset, it inherently limits the information and risks not being able to include data of specific individuals who are of future interest. Another method that deals well with a data stream is a sliding-window approach. Effectively the sliding window is also a subsample of the data, existing of only the most recent data points. The advantages of this method are that memory burden is fixed and, in cases in which the data-generating process is not stationary over time, the most recent observations most heavily influence the resulting predictions. However, choosing the size of the window often requires domain knowledge: too small might not catch any events meaningful, too large a window might computationally be too expensive (see, Aggarwal, 2007, for an introduction on many more data-stream techniques, including sliding windows). In this paper, we focus on another method to deal with data streams: *online learning*, "computing estimates of model parameters on-the-fly, without storing the data and by continuously updating the estimates as more observations become available" (Cappé, 2011). Note that our current focus is solely on estimating the individual-level effects in the context nested data and hence accounting for the grouping present in the data. While the inclusion of additional explanatory variables (for instance to take into account when an individual was last seen in the data stream, previous purchases, etc.) in the prediction model is possible when estimating shrinkage factors (see, for instance Morris and Lysy (2012) or Ippel, Kaptein, and Vermunt (2016b)), we restrict our attention solely to random-intercept models with binary outcomes.

A possible solution to efficiently obtaining estimates in a situation where the data come streaming in, is to estimate the individual-level effects in real time using *online* estimated shrinkage factors. Online estimation (or online learning) implies that a parameter (e.g., a mean, or regression coefficient) is updated using a single (or small batch of) data point and some sufficient statistics (e.g., a summation of the previous data points, Bottou, 1998; Ippel, Kaptein, and Vermunt, 2016a). An illustrative example is the computation of the sample mean $\frac{1}{n}\sum_{t=1}^{n} x_t$. Estimating a sample mean in a data stream using online learning can be done as follows:

$$
\begin{aligned}
n^{(t+1)} &= n^{(t)} + 1 \\
\bar{p}^{(t+1)} &= \bar{p}^{(t)} + \frac{x^{(t+1)} - \bar{p}^{(t)}}{n^{(t+1)}},
\end{aligned}
$$

or equivalently,

$$\begin{aligned} n &:= n+1 \\ \bar{p} &:= \bar{p} + \frac{x-\bar{p}}{n}, \end{aligned} \tag{2}$$

where n is the total number of observations and ':=' is an assignment operator, meaning that the left-hand side is updated using the expression on the right-hand side. Throughout this paper we will use the notation presented in Equation 2 as opposed to using explicit superscripts.

Note that the *offline* estimation procedure stores all the observations and for each new estimate revisits the older data points. Updating the sample mean offline in a data stream thus takes increasingly more time because more and more data need to be processed. On the contrary, the *online* estimation procedure only stores n and $\bar{p}$ in memory, and, when a new data point enters, these are updated according to Equation 2. This results in a time-constant update. Attractively, using online estimation methods, there is no need to revisit previous data points, which can therefore be discarded from memory (Kaptein, 2014). However, not every offline estimation procedure can be used exactly for online estimation (see, e.g., Ippel, Kaptein, and Vermunt, 2016a; Neal and Hinton, 1998). Hence, we often have to resort to approximate solutions. In this paper, we evaluate the accuracy of online approximations of a number of shrinkage factors. Note that although we focus on data streams, extremely large static datasets can be analyzed using the same methods.

The paper is organized as follows. Section 2 describes four existing shrinkage factors and develops the online implementation of each of the shrinkage factors. In Section 3 we discuss when the individual-level effect should be estimated, an issue which arises due to the fact that new data present themselves over time. Section 4 presents a simulation study where we compare the online and offline implementations of the shrinkage factors in terms of the accuracy of the estimated individual-level effects. Here we explicitly explore different data-generating mechanisms. In Section 5 we apply the developed online shrinkage factors to analyze a real dataset. The dataset contains data coming from a large panel study. Because dropouts in panel data is a serious threat, we focus on predicting the probability of non-response per repeatedly observed individual. These predictions could facilitate the choice of which respondents to invite for the next wave, or personalize the response request to achieve higher response rates. Finally, in Section 6, we discuss the limitations of the shrinkage factors and their possible extensions to a broader setting.

2 Estimation of shrinkage factors

The intuition of a shrinkage model (Eq. 1) is as follows: there is information available both on the group level as on the individual level, so by shrinking the individual-level effect towards the group mean, the estimator "borrows strength from the neighbors", thereby reducing the average squared prediction error (Efron and Morris, 1977; James and Stein, 1961; Stein, 1956). In this section, we discuss four shrinkage factors and develop their online implementations:

- James-Stein estimator, (JS): Here, we use the formulation as introduced by Morris and Lysy (2012). This shrinkage factor assumes normally distributed individual-level effects. This assumption is clearly violated for binary data; using the data transformation, also suggested by Morris and Lysy (2012), the normal distribution is approximated. Furthermore, this shrinkage factor is equal across all individuals.
- Approximate Maximum Likelihood estimator, (ML): The ML is unlike the JS individual specific. The level of shrinkage is *influenced* by the number of observations of an individual. This shrinkage factor also assumes that the individual-level effects are normally distributed. Hence, here also we use the data transformation suggested by Morris and Lysy (2012).
- Beta-Binomial estimator, (BB): This shrinkage factor does not assume a normal distribution, instead the individual-level effects are assumed to have a Beta distribution. Similar to ML, the level of shrinkage is individual specific and the level of shrinkage is influenced by the number of observations of an individual. We estimate the BB using the method of moments estimator (see, for instance Young-Xu and Chan, 2008).
- Heuristic estimator, (HN): Unlike the previous three shrinkage factors, the HN does not rely on any distributional assumptions. This shrinkage factor is an ad-hoc estimator which solely depend on the number of observations of an individual.

2.1 The James Stein estimator

The JS is historically important since it is among one of the first shrinkage factors to be considered in the literature. This shrinkage factor assumes normally distributed individual-level effects. Thus, the assumed data-generating model is:

$$\begin{aligned} \mathbf{y}_i &\sim N(\mu_i, \sigma_i^2 \mathbf{I}) \\ \mu_i &\sim N(\mu, \tau^2), \end{aligned} \tag{3}$$

where $\mathbf{y}_i$ is the response vector of individual i with n_i observations, $\mathbf{I}$ is a $n_i \times n_i$ identity matrix, σ_i^2 the residual variance, μ is the population average, which below we estimated using $\bar{p}$, and τ^2 the variance of the individual-level effects. Since we focus on grouped binary data, the individual means (i.e., proportions) are bounded, and therefore, not nearly normally distributed. To address this Morris and Lysy (2012) suggested the following data transformation:

$$w_i = \sqrt{\bar{n}}(\arcsin(1-2p_i) - \arcsin(1-2\bar{p})), \tag{4}$$

where w_i is the transformed individual mean, $\bar{n} = n/N$, the total number of observations divided by the total number of individuals, p_i the individual mean, $\bar{p}$ the sample mean over all data points. The transformation stabilizes the within-variance to be approximately equal to $\hat{\sigma}_i^2 = \bar{n}/n_i$. Using this data transformation to estimate the individual-level effects results in the following shrinkage model:

$$\hat{w}_i = w_i(1-\hat{\beta}_{js}) + \bar{w}\hat{\beta}_{js}, \tag{5}$$

where $\bar{w}$ the average across the transformed individual means and $\hat{\beta}_{js}$, the JS shrinkage factor, is given by

$$\begin{aligned}\hat{\beta}_{js} &= \frac{N-2}{\sum_{i=1}^{N} \frac{(w_i-\bar{w})^2}{\bar{n}/n_i}}, \\ &= \frac{N-2}{SS_{js}},\end{aligned} \tag{6}$$

as formulated by Morris and Lysy (2012), where SS_{js} is the sum of squares between individuals. To obtain the estimated individual-level effect in terms of probabilities one computes

$$\hat{\mu}_i = (\sin((\hat{w}_i/\bar{n}) + \arcsin(1-2\bar{p})) - 1)/-2. \tag{7}$$

Thus, the quantities or parameters that are needed to estimate μ_i using the JS shrinkage factor are: w_i, p_i, n_i, n, N, $\bar{p}$, $\bar{n}$, $\bar{w}$ and SS_{js} (see, Eq. 4, 5, and 6). While the above formulas detail how to estimate β_{js} in an *offline* setting, we now turn to deriving an *online* formulation.

Parts of the online computation of $\hat{\beta}_{js}$ are straightforward, for instance the computations of n or n_i using, $n := n+1$, to merely count the observations. We do not detail these further. However, counting the number of unique individuals (N) requires some additional thought: before N is incremented when a new data point arrives, we need to check whether this new data point originates from an already observed individual or from a new individual. Only in the latter case we increment the counter:

$$N := \begin{cases} N & \text{if } i_t \in \mathbf{N}, \\ N+1 & \text{if } i_t \notin \mathbf{N}, \end{cases} \tag{8}$$

where i is the index of an individual and subscript t indicates that we only focus on the individual belonging to the most recent data point. Furthermore, $\mathbf{N}$ is set of unique identifiers of all known individuals observed up to now. Each individual is labeled with an identifier such that we can track the individual over time. If a new individual is observed a new element is added to the set $\mathbf{N}$. Thus, the vector of unique identifiers grows when *new* individuals arrive in the data stream, but does not grow when an *observed* individual arrives (again) in the data stream. To check whether the individual i_t is new or not, the set of unique identifiers of individuals $\mathbf{N}$ needs to be available.

The online update of the transformed individual means, $\bar{w}$, is less trivial than count observations or the online update of the sample mean (Eq. 2). The $\bar{w}$, is a sample mean averaged over individuals (N), not over data points (n). Similar to the count of individuals (N, Eq. 8), we check whether the individual belonging to the new data point is observed before. Different update functions are used depending on whether or not an individual is observed before. When the data point belongs to a known individual, there is already a contribution of this individual to $\bar{w}$. We, first, correct $\bar{w}$ by subtracting the old contribution (i.e, the previous w_i'), then, the new contribution (i.e., the updated w_i) is added to $\bar{w}$:

$$\bar{w} := \begin{cases} (N\bar{w} - w_i' + w_i)/N & \text{if } i_t \in \mathbf{N}, \\ (N\bar{w} + w_i)/N & \text{if } i_t \notin \mathbf{N}, \end{cases} \tag{9}$$

where w_i' is the previous transformed individual mean from the last time this individual (i_t) entered and w_i the current estimate of the transformed individual mean.

Note that, due to the influx of new data, group parameters ($n, \bar{p}$, and N) are constantly changing. The data transformation uses these group parameters. This implies that *all* transformed individual means change when new data enter, not only the individual that just entered (i_t). In order to obtain the *exact* same result using the offline and online estimation procedure, *all* the transformed individual means should be updated every time a data point enters. Updating all these transformed individual means is inefficient and becomes infeasible when the number of individuals grows rapidly. Hence, we approximate the offline version by updating only the current individual. We discuss this issue in more detail in Section 3.

The remaining parameter needed for the estimation of $\hat{\beta}_{js}$ is the between individuals sum of squares (SS_{js}), which is also a summation over individuals. For the estimation of SS_{js} we make use of a similar update regime as used for $\bar{w}$:

$$SS_{js} := \begin{cases} SS_{js} - SS_{js_{i'}} + \dfrac{(w_i - \bar{w})^2}{\bar{n}/n_i} & \text{if } i_t \in \mathbf{N}, \\ SS_{js} + \dfrac{(w_i - \bar{w})^2}{\bar{n}/n_i} & \text{if } i_t \notin \mathbf{N}, \end{cases} \tag{10}$$

where $SS_{js_{i'}}$ denotes the previous contribution to the SS_{js}. Using Eq. 10, β_{js} can be estimated, with which we can estimate $\hat{w}_i$ (Eq. 5). Lastly, to obtain $\hat{\mu}_i$, $\hat{w}_i$ is imputed in Eq. 7 to transform $\hat{w}_i$ to $\hat{\mu}_i$.

2.2 Approximate Maximum likelihood estimator

The ML is an often used shrinkage factor for multilevel models, where μ_i's are normally distributed and the outcome variable is continuous (among others, Goldstein, 1986). Because the means of binary observations are not normally distributed, we use the same data transformation (Eq. 4) as discussed previously in Section 2.1. Similar to the estimation of μ_i using the JS, the ML estimation of μ_i uses the alternative shrinkage model (Eq. 5) which includes the transformed individual means. However, unlike the previous shrinkage factor, ML estimator is tailored to an individual: the level of shrinkage is influenced both by the number of observations of an individual as well as by information of the other individuals:

$$\begin{aligned} \hat{\beta}_{ml\_i} &= \frac{\hat{\sigma}_i^2}{\hat{\tau}^2 + \hat{\sigma}_i^2}, \\ &= \frac{\bar{n}/n_i}{\hat{\tau}^2 + \bar{n}/n_i}, \end{aligned} \tag{11}$$

where more observations of an individual (n_i) result in less shrinkage, and $\hat{\tau}^2$ is the maximum-likelihood value of the variance of the individual-level effects. The most likely value of τ^2 is found by maximizing the following log-likelihood function (see,

Morris and Lysy, 2012, equation at the bottom of page 128):

$$\begin{aligned}\ell(\tau^2) &= \sum_{i=1}^{N}\Big[\frac{(w_i-\bar{w})^2}{\bar{n}/n_i}\frac{\bar{n}/n_i}{\bar{n}/n_i+\tau^2}+\log\Big(\frac{\bar{n}/n_i}{\bar{n}/n_i+\tau^2}\Big)\Big]/2,\\ &= \sum_{i=1}^{N}\Big[\frac{(w_i-\bar{w})^2}{\sigma_i^2}\frac{\sigma_i^2}{\sigma_i^2+\tau^2}+\log\Big(\frac{\sigma_i^2}{\sigma_i^2+\tau^2}\Big)\Big]/2,\\ &= \sum_{i=1}^{N}\Big[\frac{(w_i-\bar{w})^2}{\sigma_i^2}\beta_{ml\_i}+\log(\beta_{ml\_i})\Big]/2,\end{aligned} \tag{12}$$

In the case of offline estimation, Eq. 12 is maximized by iterating over the dataset, using a numerical optimization method, for instance Newton Raphson.

For the estimation of μ_i using ML, the following parameters are needed: p_i, n_i, w_i, $\hat{\sigma}_i^2$, n, N, $\bar{n}$, $\bar{p}$, $\bar{w}$, and $\hat{\tau}^2$. Most of these parameters have already been discussed in the previous section (see, Eq. 2, 8, and 9), therefore we focus only on the remaining parameter: the estimation of the variance of the individual-level effects, $\hat{\tau}^2$.

Estimating $\hat{\tau}^2$ is not straightforward during the data stream since using an iterative maximization procedure is not feasible. For this reason, we use Stochastic Gradient Descent (SGD, Bottou, 2010). SGD updates the estimate of τ^2 by evaluating the gradient (in this case, a one-dimensional gradient or derivative) of $\ell(\tau^2)$ one data point at a time.

Intuitively, SGD works as follows: The first-order derivative of the log-likelihood function is a summation over individuals. SGD evaluates this first-order derivative for a single data point and based on the value of the derivative SGD determines whether the current estimate of the parameter is above or below the maximum-likelihood value. Using a learn rate (γ), SGD steps towards the maximum of the likelihood function. When a new data point enters, SGD evaluates the derivative again and updates the parameter estimate accordingly. The first-order derivative of $\ell(\tau^2)$ is:

$$\nabla\ell(\tau^2) = \sum_{i=1}^{N}\frac{(w_i-\bar{w})^2-\hat{\sigma}_i^2-\hat{\tau}^2}{2(\hat{\sigma}_i^2+\hat{\tau}^2)^2}. \tag{13}$$

Because Eq. 13 is a summation over individuals, we apply a similar update regime as in Equation 10:

$$\hat{\tau}^2 := \begin{cases}\hat{\tau}^2-\gamma\nabla\ell_{i'}(\tau^2)+\gamma\nabla\ell_i(\tau^2) & \text{if } i_t\in\mathbf{N},\\ \hat{\tau}^2+\gamma\nabla\ell_i(\tau^2) & \text{if } i_t\notin\mathbf{N},\end{cases}$$

where $\nabla\ell_{i'}(\tau^2)$ is the previous contribution of individual i to the gradient of τ^2 and $\nabla\ell_i$ the current contribution to that gradient of individual i. When the learn rate, γ, is large, SGD can 'move' fast towards the maximum-likelihood value, however with the same pace it can also step over the maximum of the likelihood function. When the learn rate is small it will take many evaluations of the derivative (i.e., many data points have to enter) before the maximum likelihood is reached (see, e.g., Bottou, 2010; Xu, 2011; Schaul, Zhang, and LeCun, 2013, for a more extensive discussion on learn rates for SGD). After the estimation of β_{ml_i}, the individual-level effect is estimated using the shrinkage model for transformed individual means (Eq. 5) after which $\hat{w}_i$ is transformed to $\hat{\mu}_i$ using Eq. 7.

2.3 The Beta-Binomial estimator

When we assume that the data-generating model is a Beta Binomial distribution,

$$\begin{aligned} k_i &\sim Bin(n_i, \mu_i) \\ \mu_i &\sim Beta(\alpha, \beta) \end{aligned} \tag{14}$$

where $k_i = \sum_{j=1}^{n_i} y_{ij}$, the individual means do not have to be transformed to estimate BB, because the Beta distribution naturally falls within the $[0,1]$ range. Thus, in order to estimate μ_i we can make use of the shrinkage model as defined in Eq. 1. In this case, we choose the method-of-moments estimation method to estimate BB because this method has a closed-form solution to estimate the shrinkage factor. The closed-form expression of the estimation procedure of BB makes it easier to rewrite the formulation of BB to an online formulation.

The compound distribution of the Beta-Binomial distribution is:

$$\begin{aligned} f(k|n,\alpha,\beta) =& \frac{\Gamma(n+1)}{\Gamma(k+1)\Gamma(n-k+1)} \frac{\Gamma(k+\alpha)\Gamma(n-k+\beta)}{\Gamma(n+\alpha+\beta)} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \\ =& \frac{\Gamma(M)}{\Gamma(M\mu)\Gamma(M(1-\mu))} \binom{n}{k} \frac{\Gamma(k+M\mu)\Gamma(n-k+M(1-\mu))}{\Gamma(n+M)} \end{aligned} \tag{15}$$

where μ is estimated by $\bar{p}$, and $k = \sum_{i=1}^{N} \sum_{j=1}^{n_i} y_{ij}$, and where $\hat{M}$ is computed as follows:

$$\hat{M} = \frac{\bar{p}(1-\bar{p}) - \hat{s}^2}{\hat{s}^2 - \frac{\bar{p}(1-\bar{p})}{N} c},$$

where $c = \sum_{i=1}^{N} 1/n_i$ and $\hat{s}^2$ is defined as

$$\begin{aligned} \hat{s}^2 =& \frac{N \sum_{i=1}^{N} n_i (p_i - \bar{p})^2}{(N-1) \sum_{i=1}^{N} n_i}, \\ =& \frac{N SS_{bb}}{(N-1) \sum_{i=1}^{N} n_i}, \end{aligned}$$

where SS_{bb} is the between-individual sum of squares using the individual means. The shrinkage factor of BB is:

$$\hat{\beta}_{bb\_i} = \frac{\hat{M}}{\hat{M} + n_i}, \tag{16}$$

Similar to the ML, the BB is also individual specific where the number of observations per individual influences the level of shrinkage. The parameters for the estimation of μ_i using BB are: n_i, p_i, SS_{bb_i}, n, N, c, $\bar{p}$, $\hat{M}$, and $\hat{s}^2$.

The computation of $\hat{s}^2$ requires the following: N, n, and SS_{bb}. While the first two parameters are easy to update during the data stream and already discussed in Section 2.1, the latter is again a sum over individuals, which requires an update similar to SS_{js} (Eq. 10):

$$SS_{bb} := \begin{cases} SS_{bb} - SS_{bb_{i'}} + n_i (p_i - \bar{p})^2 & \text{if } i_t \in \mathbf{N}, \\ SS_{bb} + n_i (p_i - \bar{p})^2 & \text{if } i_t \notin \mathbf{N}, \end{cases}$$

where $SS_{bb_{i'}}$ denotes the previous contribution to the SS_{bb}. Similar to the $\hat{\beta}_{js}$, the $\hat{\beta}_{bb}$ estimated online is slightly different compared to the offline estimated $\hat{\beta}_{bb}$. The difference between the two estimation procedures is due to the fact that SS_{bb} is dependent on $\bar{p}$ which fluctuates throughout the data stream.

For the computation of $\hat{M}$ we need $\bar{p}, \hat{s}^2, N$, and c. Because all parameters except the last parameter are already discussed previously, we only present the computation of c:

$$c := \begin{cases} c - c_{i'} + 1/n_i & \text{if } i_t \in \mathbf{N}, \\ c + 1/n_i & \text{if } i_t \notin \mathbf{N}, \end{cases}$$

where $c_{i'}$ is the previous contribution to c (i.e. $\frac{1}{n_i - 1}$). The individual-level effect μ_i is then estimated using Eq. 1, using $\hat{\beta}_{bb\_i}$, p_i and $\bar{p}$.

2.4 The Heuristic estimator

The previous two shrinkage factors (ML, Eq. 11 and BB, Eq. 16) both have a similar type of intuition: individual-level effects are moved more towards the group mean when little is known about the individual (i.e., a small number of observations) compared to when there is more information about an individual. The last shrinkage factor has the same intuition, however, we do so without any distributional assumptions or sophisticated formulas. The last shrinkage factor

$$\hat{\beta}_{hn\_i} = \frac{1}{\sqrt{n_i}},$$

shrinks individual-level effects only based on the (square root of) number of observations of an individual. Like BB, the HN also shrinks the individual-level effects using Eq. 1: When an individual only has 1 observation, $\hat{\mu}_i = \bar{p}$, and the amount of shrinkage decreases as more observations of an individual enter. All the parameters used for the estimation of μ_i using HN (p_i, n_i, $\bar{p}$, and n), have been discussed in Section 2.1.

Table 1 gives an overview of the online shrinkage factors that are used in the simulation study. The characteristics of each of the shrinkage factors are presented. The last three lines of the table give an indication how many parameters should be updated to estimate the shrinkage factor and the individual-level effect when an additional data point enters the dataset. First of the three lines are the individual parameters, second line are group level count parameters, and last line are the parameters that require more computations to update.

3 Predicting individual-level effects: when is the right time?

When analyzing static data, the exact moment at which one should predict the individual-level effects, does not come to question. It naturally follows that the prediction is only made once: after the shrinkage factor is estimated. This is, however, not the case when

Table 1 Overview of the characteristics of the shrinkage factors and their complexity

	JS	ML	BB	HN
distribution μ_i	Normal	Normal	Beta	–
group or individual	group	individual	individual	individual
equal variance	yes	no	no	no
transformation	yes	yes	no	no
update $\hat{\mu}_i$	p_i, n_i, w_i, SS_{js_i}	$p_i, n_i, w_i, \hat{\tau}_i^2, \hat{\sigma}_i^2$	p_i, n_i, SS_{bb_i}	p_i, n_i
	$\bar{p}, \mathbf{N}, n, N$	$\bar{p}, \mathbf{N}, n, N$	$\bar{p}, \mathbf{N}, n, N$	$\bar{p}, \mathbf{N}, n$
	$\bar{n}, \bar{w}, SS_{js}$	$\bar{n}, \bar{w}, \hat{\tau}^2$	$SS_{bb}, \hat{s}^2, c, \hat{M}$	–

data are entering over time. In this section, we explain why the researcher is faced with a choice when to estimate the individual-level effect.

The individual-level effect is a combination of the individual mean, the group mean, and a shrinkage factor. Every time a new data point enters, the record of one person changes. However, due to this new data point, the estimates of all the individual-level effects at that moment in time change slightly. That is, *if* one would re-estimate all the individual-level effects every time a new data point enters, the estimates change with every additional data point. Such re-estimation is, however, infeasible for the full set of individuals at each time point, and in many applications one would only obtain an estimate only for the individual concerned. In any case, the shrinkage of the individual-level mean to the group-level mean to obtain a prediction for a specific individual can be done at two distinct moments:

1. one could obtain a shrinkage estimate the moment an individual's data is observed and store the resulting prediction,
2. or, one could obtain a prediction at the moment that the individual is about to re-enter the dataset; hence, the moment a prediction might be needed.

The first option leads to the following procedure: when a data point enters, the group-level parameters, the parameters of the individual (i_t) whose data point entered, and shrinkage factor are updated or computed. With these parameters, a prediction of the individual-level effect is made and stored in memory. This option has two downsides. The first downside is that besides p_i, a prediction ($\hat{\mu}_i$) needs to be stored, which is potentially never used if we do not observe this individual anymore. The other downside is that while we store the prediction, new data are entering. These new data points affect the shrinkage factor and global statistics. All these changes are not taken into account because the prediction is stored and considered fixed. Therefore, the stored prediction does not optimally make use of the most recent information.

For the second option, imagine an individual (i_t) intends to pay our website—as discussed in the introduction—a visit again. Her browser will send out a request to access the website. At that point, we know who is about to enter our website, so we can retrieve this individual's record. Now, we can predict this individual's $\hat{\mu}_i$ based on all the information we know so far. The data generated by this individual during the website visit allows us to update both the group and individual-level parameters. This second option thus deals with both downsides of the first option: it does not waste memory on storing predictions that we might end up not using at all and it incorporates the most recent changes to the group-level parameters.

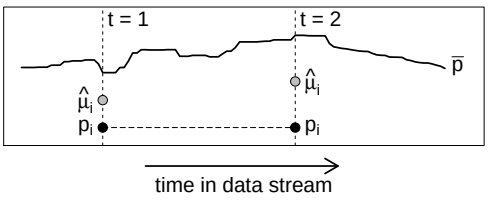


Fig. 1 An illustration of when to shrink the individual-level effect. Option 1 ($t = 1$) estimates μ_i right after the data point enters, option 2 estimates μ_i at $t = 2$. While p_i remains the same between the two time points, the group mean $\bar{p}$, does change over time.

The two options are illustrated in Figure 1. The black dot denotes an individual mean. One could choose to predict the individual-level effect right after this data point enters at $t = 1$, or one could wait until this individual returns ($t = 2$) and shrink towards the group mean at that point in time (which is $t = 2$ in this case). As can be seen from the plot, *when* the individual-level effect is estimated influences the estimate of μ_i. Because the group mean and the estimated shrinkage factor change over time, these two options (shrink at $t = 1$ vs. $t = 2$) do not result in the same $\hat{\mu}_i$. In the following simulation study, we have chosen this second implementation of predicting the individual-level effect.

4 Simulation Study

4.1 Design

To evaluate whether the online implementations of the shrinkage factors perform equally well as their original offline implementations, we conduct a simulation study. In this simulation study we compare the two estimation procedures in terms of the average squared prediction error ($\sum(\hat{\mu}_i - \mu_i)^2/N$). Since two of the four shrinkage factors assume a normal distribution, we specifically examine the case when this is violated in the simulation study. To do so, we generated three distributions of the individual-level effects, which increasingly depart from normality: the distribution underlying the *true* individual-level effects is centered $B(7,7)$, right skewed, $B(2,12)$, or a mixture of two Beta distributions: $B(1,6)$ and $B(6,1)$. We set the average[1] number of observations equal to 20. As a benchmark we use a multilevel model with logit link function, as implemented in the GLMR function from the 'lme4' package (in [R]) with 20 adaptive Gaussian Quadrature points. While this model is known to provide very good estimates of μ_i, it is computationally complex to fit (Agresti, Booth, Hobert, and Caffo, 2000; Bock and Aitkin, 1981; Breslow and Clayton, 1993; Moerbeek, Van Breukelen, and Berger, 2003; Rabe-Hesketh, Skrondal, and Pickles,

[1] Because we sample the individuals at random after which we generate an observation, the number of observations is not equal across individuals.

2002; Skrondal and Rabe-Hesketh, 2004), especially in a data stream. The generated data streams consist of $n = 10,000$ (which results in $N = 500$) and all conditions have 1,000 replications.

4.2 Results

The main results of the simulation study are presented in Figure 2 and Figure 3. Figure 2 presents the average of the estimated shrinkage factors over the simulation runs. Figure 3 presents the average squared prediction error over the simulation runs. Both figures consist of three subfigures: one for the centered ($B(7,7)$ distribution, one for the right skewed ($B(2,12)$) distribution, and one for the mixture distribution ($B(1,6)$ and $B(6,1)$). Table 2 further details the average squared prediction error at three points in the data stream and includes the standard deviation over the different simulation runs.

The x-axes of Figure 2 presents the length of the data stream and the y-axes the average shrinkage factor. The solid lines represent the online implementations of the shrinkage factors. The dashed lines represent the offline implementations of the shrinkage factors. The four gray lines indicate the offline (dashed) and the online (solid) shrinkage factors that do not require the data transformation. The BB carries triangle symbols (facing up) to differentiate the BB from HN which carries square symbols. The black lines are also marked with symbols: JS is denoted with circles and ML is marked with triangles (facing down). In all three subfigures, there is a small difference between the offline and online implementations of the shrinkage factors. In general, the online implementations tend to shrink somewhat more than the offline implementations.

In Figure 2a the centered distribution is presented. The BB (online and offline) shrinks the individual-level effects most, and the online implementation does so even more than the offline implementation. The BB (online and offline) needs many (over 2,000) data points before it can be estimated. This is an artifact of the method of moments estimator, which returns negative hyperparameters for the Beta distribution when the data does not (yet) comply to the beta distribution (under dispersion). Both the offline versions of the JS and the ML have a relatively stable level of shrinkage, while the online implementation of the JS quickly decreases during the data stream. The ML online implementation only changes very gradually. The chosen learn rate (γ =0.01) might have been slightly too small. Towards the end of the generated data streams three or the four shrinkage factors shrink approximately the same, only the heuristic shrinkage factor (online and offline) shrinks substantially less than the other factors.

The average estimated shrinkage factors in the right-skewed distribution of the individual-level effects are presented in Figure 2b. For the two shrinkage factors that do not use the data transformation the results are quite similar. However, the ML and JS show differences with the previous condition. The online implementation of the JS shrinks more over a longer time, also the offline implementation of the JS shrinks more in the beginning of the data stream. The offline ML shrinks on average some more than the offline JS but behaves qualitatively the same as the offline JS. Towards

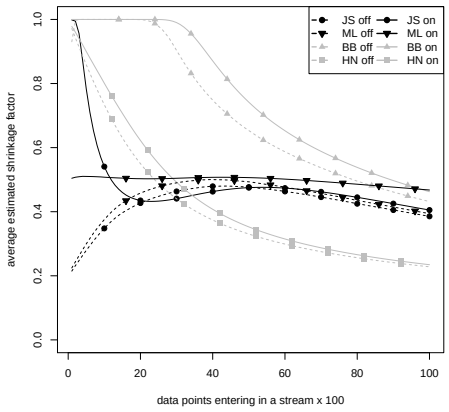
the end of the data stream, the different shrinkage factors are more spread out than in the previous condition, while the online and offline implementations of all four shrinkage factors have similar levels of shrinkage.

The last subfigure (Fig. 2c) presents a different pattern of shrinkage factors. Even at the end of the data stream, there are two distinct clusters of shrinkage factors. The cluster of shrinkage factors with the highest level of shrinkage consists of the online and offline implementations of the heuristic shrinkage factors, and the online implementation of ML. The remaining shrinkage factors (online and offline BB and JS, and the offline ML) hardly shrink at all. This is due to the fact that the data-generating distribution of the individual-level effects is bimodal. Because the heuristic estimator (online and offline) does not have any distributional assumptions, it cannot take into account that there are two modes. The online ML does decrease more in this condition than in the other two conditions. A larger learn rate or longer data stream would allow the online ML to decrease even more and reach a similar level of shrinkage as the offline ML. The offline ML, BB and JS do take into account that the distribution of individual level effects is not normal, and shrink very little accordingly.

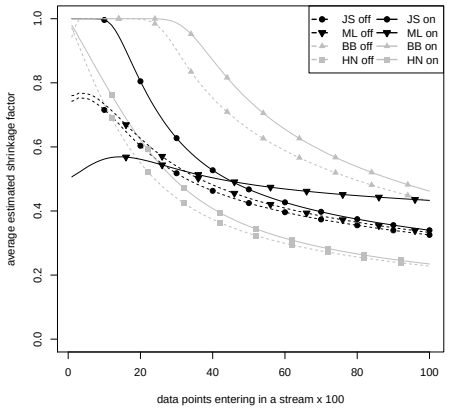
The subfigures of Figure 3 are organized as follows: The y-axes present the average squared prediction error: $\sum(\hat{\mu}_i - \mu_i)^2/N$ and the x-axes present the data stream. Note that the y-axes of the three subfigures of Figure 3 differ across the three scenarios. In addition to the already introduced lines, the dotted line represents the GLMR function. The results of the two unimodal distributions ($B(7,7)$ and $B(2,12)$) are comparable, however, the mixture distribution ($B(1,6), B(6,1)$) shows different results. Figure 3a and Figure 3b show that in the beginning of the data stream, the two shrinkage factors that make use of the data transformation have more error (JS, ML) than the two shrinkage factors (BB, HN) that do not rely on the transformation. The GLMR function performs 'best' in both scenarios. However, the difference between the shrinkage factors and the GLMR function is minimal later in the data stream. More importantly for our purpose, there is almost no difference between the *offline* and *online* implementations of the shrinkage factors.

Table 2 is organized as follows. In the rows are the three conditions (centered, right skewed and mixture), within each condition three points within the data stream are presented ($n = 1{,}000$, 5,000, and 10,000). In the columns are the different shrinkage factors with the offline and online implementations. Both the average squared prediction error of each of the shrinkage factors and the standard deviations are presented. In the centered scenario, the GLMR function outperforms the shrinkage factors (offline and online). However, as the data stream continues, the difference between the shrinkage factors and GLMR becomes smaller. The standard deviations across the shrinkage factors and during the stream are stable and small. The second scenario, the right-skewed distribution, has an even smaller average squared prediction error. This is due to the fact that the distribution of μ_i is narrowly distributed around the group mean making the mean over all data a good predictor of the individual-level effects. This results in a small average squared prediction error and even smaller standard deviations.

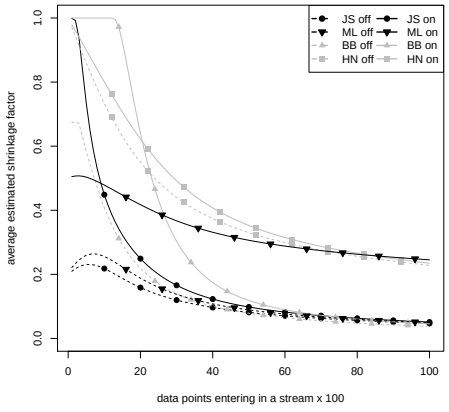
The mixture scenario provides quite different results. While the average squared prediction error decreases rapidly in the beginning of the data stream (see Fig. 3c), after about 2,000 data points the error *increases* for both JS and ML. For the other two



(a) $B(7,7)$

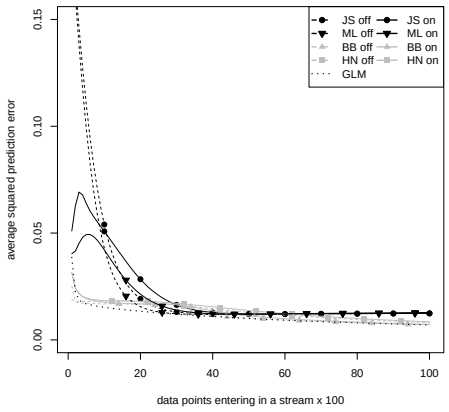


(b) $B(2,12)$

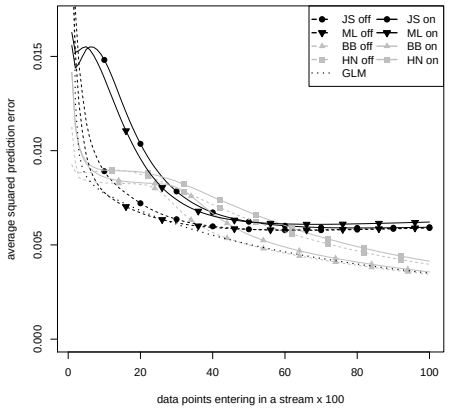


(c) $B(1,6)$ & $B(6,1)$

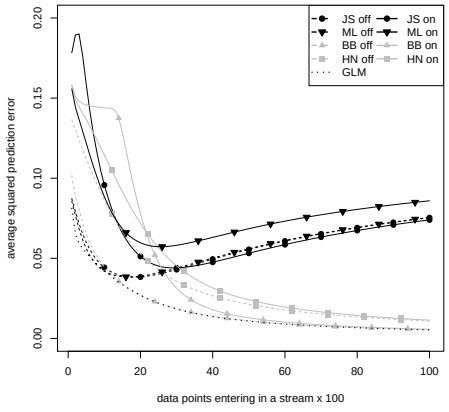
Fig. 2 The average estimated shrinkage factors, averaged over the 1,000 replications



(a) $B(7,7)$



(b) $B(2,12)$



(c) $B(1,6)$ & $B(6,1)$

Fig. 3 Average squared prediction error for the three scenarios, averaged over the 1,000 replications.

shrinkage factors and the benchmark GLMR, i.e., these estimation methods that do not use Morris and Lysy's (2012) data transformation, this is not the case. This pattern appears for both the online and offline estimated shrinkage factors. Due to the mixture of the two distributions the individual means are either clustered close to zero, or close to one. While the mean of these two distributions is 0.5, all the true individual-level effects are either close to zero or close to one. This makes the group mean a poor predictor of the individual-level effects. Because these individual means are far from the group mean, the transformed individual means have large absolute values. In an absolute sense, larger values are moved more towards the group mean than values that are closer to the group mean. Transforming the predicted individual-level effects to $\hat{\mu}_i$'s causes the individual-level effects to be moved even more towards the group mean. Thus, even though the shrinkage factors that use the data transformation are in fact small (see Figure 2c), the data transformation pushes the individual-level effects even closer to the group mean. This additional push towards the group mean causes the JS and ML to have larger prediction error than HN and BB.

From the simulation study, we can thus conclude that a) for a long enough data stream all online shrinkage factors perform as well as their offline counterparts, and b) the BB seems to have the most robust performance over the three conditions. Hence, for the analysis of large, nested, binary outcome data streams we would recommend using the our online version of the BB. In the following section, all the examined shrinkage factors are further evaluated in a real-data example. In this example we show that it is possible to accurately predict whether respondents of a long-running panel study will respond to a monthly questionnaire.

5 LISS Panel Study: Predicting Attrition

An application where data are entering over time and real-time prediction is relevant is a panel study, where new questionnaires are sent to the same respondents over a longer period of time. Panel studies are used to analyze ongoing trends. One of the major issues of a panel study is attrition (i.e., respondents who drop out) because it can affect the generalizability of the results to the population (Goodman and Blum, 1996). Much effort is spent on the prevention of non-response, for instance, reminders, rewards (Curtin, Singer, and Presser, 2007; Manzo and Burke, 2012), and multi-mode data collection (Leeuw, 2005). Knowing which respondents are likely to drop out of the panel, could facilitate the prevention of the dropout. For instance, when the probability for a given respondent to answer to the questionnaire drops below a threshold, an additional incentive (a letter of the importance of the panel, money etc.) could be sent when inviting the respondent to answer the questionnaire to increase the probability that the respondent will reply to the questionnaire. Knowing a respondent was unlikely to respond to the questionnaire, after the facts, is not very informative or helpful. Therefore, predicting non-response in a panel study is a good example of a case where real-time prediction is useful.

In this application, we predict whether a respondent of the panel study is going to participate in the next wave as well. Data are coming from the LISS (Longitudinal Internet Study for Social sciences) panel study, consisting of 50 monthly waves be-

Table 2 The average squared prediction error. In the parentheses are the sd's over 1,000 replications.

		JS				ML				BB				HN				GML	
		offline		online		offline		online		offline		online		offline		online		offline	
distribution	n	$\bar{e^2}$	(sd)	$\bar{e^2}$	(sd)	$\bar{e^2}$	(sd)	$\bar{e^2}$	(sd)	$\bar{e^2}$	(sd)	$\bar{e^2}$	(sd)	$\bar{e^2}$	(sd)	$\bar{e^2}$	(sd)	$\bar{e^2}$	(sd)
	1,000	.054	(.004)	.051	(.004)	.043	(.004)	.042	(.004)	.017	(.001)	.018	(.001)	.018	(.001)	.018	(.001)	.015	(.001)
$B(7,7)$	5,000	.012	(.001)	.012	(.001)	.012	(.001)	.012	(.001)	.010	(.001)	.011	(.001)	.013	(.001)	.014	(.001)	.010	(.001)
	10,000	.012	(.001)	.012	(.001)	.012	(.001)	.013	(.001)	.007	(.000)	.007	(.000)	.008	(.001)	.008	(.001)	.007	(.000)
	1,000	.009	(.001)	.015	(.001)	.008	(.001)	.014	(.001)	.008	(.001)	.009	(.001)	.009	(.001)	.009	(.001)	.008	(.001)
$B(2,12)$	5,000	.006	(.000)	.006	(.000)	.006	(.000)	.006	(.000)	.005	(.000)	.006	(.001)	.006	(.000)	.007	(.001)	.005	(.000)
	10,000	.006	(.000)	.006	(.000)	.006	(.000)	.006	(.000)	.003	(.000)	.004	(.000)	.004	(.000)	.004	(.000)	.004	(.000)
	1,000	.044	(.005)	.096	(.006)	.042	(.004)	.088	(.005)	.045	(.005)	.144	(.003)	.086	(.003)	.115	(.003)	.042	(.004)
$B(1,6), B(6,1)$	5,000	.056	(.001)	.053	(.002)	.056	(.001)	.068	(.002)	.011	(.001)	.013	(.001)	.021	(.001)	.024	(.001)	.011	(.001)
	10,000	.075	(.002)	.074	(.002)	.076	(.002)	.086	(.002)	.005	(.000)	.006	(.001)	.011	(.001)	.011	(.001)	.005	(.000)

tween November 2007 and December 2011. For the analysis, we selected only these respondents that received at least one questionnaire, who had an identification number and started before December 2011. Total number of individuals used for the analysis is $N = 12,924$ and $n = 397.647$ observations. For the analysis of the LISS panel data, we had to drop one questionnaire (July, 2011) because none of the respondents had answered this questionnaire.

We analyze the data by replaying the data as if it is a data stream. To do so we kept the ordering of the questionnaires but randomly ordered the respondents within a questionnaire. We randomly selected the responses within a questionnaire because we do not have data about the order in which the data entered originally. We compare the results of the four shrinkage factors (online and offline) with the results by the GLMR function, like in the simulation study, in terms how well each of the methods can classify whether a respondent is or is not going to respond. We take into account whether a respondent indicated to stop the panel and correct the group statistics accordingly.

5.1 Results

Figure 4 presents the results of the replay of the data stream of the LISS panel questionnaires. The y-axis presents the percentage of correctly classified respondents. A respondent is correctly classified if the shrinkage model predicted the probability of a response greater than .5 and the respondent indeed answered the questionnaire, or when the predicted probability was below .5 and the responded failed to answer the questionnaire. The x-axis is the replay of the questionnaires as these are send out over time.

As expected from the simulation study, the differences between the offline and online estimation procedures are negligible. The classification performances of the offline BB and GLMR are exactly the same, and therefore, impossible to disentangle. Furthermore, the same clustering of shrinkage factors as in the simulation condition with the mixture of distribution appears in Figure 4: the JS and ML (online and offline) are less able to make accurate predictions with regard to non-response while the HN and BB perform equally well as the benchmark (GLMR). This is not much of a surprise, as the distribution of the individual-level effects estimated with GLMR (the MAP estimates) shows that the majority of the individual-level effects are on either end of the interval, see Fig. 5, just like the mixture of distributions of the simulation study. Even though BB and HN are less computationally complex than GLMR, the predictions made by BB and HN are equally accurate.

6 Conclusion and discussion

The most important conclusion we can draw is that we can make accurate predictions of the individual-level effects when the outcome variable is binary, the data have a nested structure, when the data enter over time, and predictions are required in real time. While the multilevel model with logit link function is the standard for analyzing nested data with a binary outcome, due to the computational complexity of that

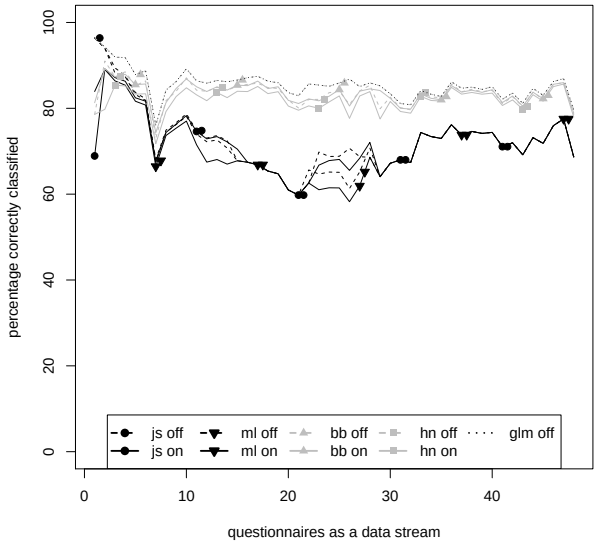


Fig. 4 Percentage correctly classified responses

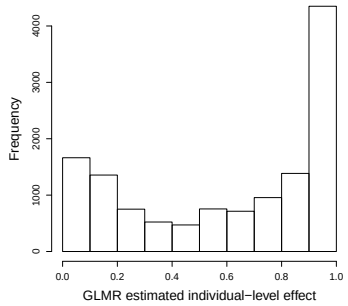


Fig. 5 Estimated μ_i using the GLMR function.

model, analyzing data streams of nested binary data becomes infeasible. To overcome this problem, we studied online – and computationally efficient – versions of four different shrinkage factors: the James-Stein estimator, the (approximate) Maximum Likelihood estimator, the Beta-Binomial estimator and lastly a heuristic estimator. In a simulation study, we showed that all these shrinkage factors on average make good predictions of the individual-level effects. However when there is a mixture of distributions of the individual-level effects, shrinkage factors that do not rely on the normal distribution of the individual-level effects do noticeably better. It appears that the data transformation suggested by Morris and Lysy (2012), in the studied situations

does not work well in situations where the number of observations is limited and the distribution of the individual-level effects deviates from the normal distribution.

There are differences between the shrinkage factors in how well they are able to predict the individual-level effects. When the true individual-level effects are close to normally distributed the prediction accuracy is very similar across all shrinkage factors. More importantly, the shrinkage factors implemented offline (making use of all individual data points) or online (incrementally and not revising previous data) perform similarly. However, when the distribution of the individual-level effects deviates from the normal distribution, the James-Stein (JS) estimator and the approximate Maximum Likelihood estimator perform less well than the Beta-Binomial and heuristic estimator.

In the current study, we assumed the data-generating process to be stationary; the possible effect of the time within the data stream is not explicitly modeled. As a result, the individual-level effects are estimated using the information of all data points equally, irrespective of their history. In practice, this assumption might, however, not hold. If the stationarity assumption does not hold, one might prefer to weigh the recent data points more heavily than the older data points when computing an estimate. All the online shrinkage factors presented in this paper are easily adapted to create such a moving window approach by changing the learn rate of the procedure to a fixed value: for example, when updating the sample mean $\bar{p}$ using Equation 2 we effectively use a learn rate of $\frac{1}{n}$ (which is easy to see since $\frac{x-\bar{p}}{n} = \frac{1}{n}(x-\bar{p})$). By choosing a fixed learn rate of, e.g., $\frac{1}{1000}$ instead we would (smoothly) decrease the value of older data points in the resulting estimate.

A possible advantage of the JS and ML could be that these methods are easier to extend to deal with covariates (see, for instance, Morris and Lysy, 2012). The JS can easily include fixed effects to improve the prediction of the individual effect. Including more random effects in this case might be less straightforward. The ML can, however, facilitate more random effects as well as fixed effects (Ippel, Kaptein, and Vermunt, 2016b) at the level of the group. Including fixed effects at the level of the observations seems challenging for both the JS as well as the ML since the suggested data transformation aggregates the information to the level of individuals.

Making real-time predictions without revising older data has great potential. These real-time predictions are not only beneficial in the context of e-commerce, or to encourage respondents that have a low probability to respond to the questionnaire. Other cases include classifying credit-card transactions (legitimate versus fraudulence transactions), monitoring patients' compliance with their medical regimen (medication was taken or not), or tracking students' progress in their educational career (passing exams or not), to name a few. The presented methods for estimating individual-level effects in data streams allow the researcher to take into account the dependence among the observations without losing the computational efficiency of the methods that do not take this dependency into account.

Acknowledgements We would like to thank Prof. dr. Peter Lugtig for preparing and sharing the LISS panel dataset. Additionally we would like to thank Prof. dr. Marcel Croon for his continuous help and feedback on this paper.

References

Aggarwal CC (2007) Data Streams: Models and Algorithms, vol 31, 2007th edn. Springer, DOI 10.1007/978-0-387-47534-9, arXiv:1310.8004v1

Agresti A, Booth JG, Hobert JP, Caffo B (2000) Random-Effects modeling of Categorical Response Data. Sociological Methodology 30(1):27–80, DOI 10.1017/CBO9781107415324.004, arXiv:1011.1669v3

Bock RD, Aitkin M (1981) EM Solution of the Marginal Likelihood Equations. Psychometrika 46(4):443–459

Bottou L (1998) Online learning and stochastic approximations. On-line learning in neural networks pp 1–34

Bottou L (2010) Large-Scale Machine Learning with Stochastic Gradient Descent. In: Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'2010), pp 177–187

Breslow NE, Clayton DG (1993) Approximate Inference in Generalized Linear Mixed Models. Journal of the American Statistical Association 88(421):9–25, DOI 10.2307/2290687

Cappé O (2011) Online Expectation-Maximization. In: Mengersen K, Titterington M, Robert C, Robert P (eds) Mixtures: Estimation and Applications, Wiley, pp 1–20

Cheng H, Cantú-Paz E (2010) Personalized click prediction in sponsored search. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, ACM, New York, NY, USA, WSDM '10, pp 351–360, DOI 10.1145/1718487.1718531

Curtin R, Singer E, Presser S (2007) Incentives in Random Digit Dial Telephone Surveys: A Replication and Extension. Journal of Official Statistics 23(1):91–105

Efraimidis PS, Spirakis PG (2006) Weighted random sampling with a reservoir. Information Processing Letters 97(5):181–185, DOI 10.1016/j.ipl.2005.11.003

Efron B, Morris C (1977) Stein's Paradox in Statistics. DOI 10.2307/2284155

Goldstein H (1986) Multilevel mixed linear model analysis using iterative generalized least squares. Biometrika 73(1):43–56, DOI 10.1093/biomet/73.1.43

Goodman J, Blum T (1996) Assessing the Non-Random Sampling Effects of Subject Attrition in Longitudinal Research. Journal of Management 22(4):627–652, DOI 10.1016/S0149-2063(96)90027-6

Ippel L, Kaptein MC, Vermunt JK (2016a) Dealing with Data Streams: an Online, Row-by-Row , Estimation Tutorial . Methodology 12:124–138

Ippel L, Kaptein MC, Vermunt JK (2016b) Estimating random-intercept models on data streams. Computational Statistics and Data Analysis 104:169–182, DOI 10.1016/j.csda.2016.06.008

James W, Stein C (1961) Estimation with Quadratic Loss. In: Neyman J (ed) Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics, University of California Press, Berkeley, Calif, vol 1, pp 361–379

Kaptein MC (2014) {RStorm}: Developing and Testing Streaming Algorithms in {R}. The R Journal 6(1):123–132

Leeuw EDD (2005) To Mix or Not to Mix Data Collection Modes in Surveys. Journal of Official Statistics 21(2):233–255

Linares B, Guizar JM, Amador N, Garcia A, Miranda V, Perez JR, Chapela R (2010) Impact of air pollution on pulmonary function and respiratory symptoms in children. Longitudinal repeated-measures study. BMC Pulmonary Medicine 10(1):62, DOI 10.1186/1471-2466-10-62

Manzo AN, Burke JM (2012) Increasing Response Rate in Web-Based/Internet Surveys, Springer New York, New York, NY, pp 327–343. DOI 10.1007/978-1-4614-3876-2_19

Moerbeek M, Van Breukelen G, Berger M (2003) A comparison of Estimation Methods for Multilevel Logistic Models. Computational Statistics 18:19–37, DOI 10.1007/s001800300130

Morris C, Lysy M (2012) Shrinkage Estimation in Multilevel Normal Models. Statistical Science 27(1):115–134, DOI 10.1214/11-STS363

Murnaghan DA, Sihvonen M, Leatherdale ST, Kekki P (2007) The relationship between school-based smoking policies and prevention programs on smoking behavior among grade 12 students in Prince Edward Island: A multilevel analysis. Preventive Medicine 44(4):317–322, DOI 10.1016/j.ypmed.2007.01.003

Neal R, Hinton GE (1998) A View Of The Em Algorithm That Justifies Incremental, Sparse, And Other Variants. Learning in Graphical Models pp 355–368, DOI 10.1007/978-94-011-5014-9-12

Pébay P, Terriberry TB, Kolla H, Bennett J (2016) Numerically stable, scalable formulas for parallel and online computation of higher-order multivariate central moments with arbitrary weights. Computational Statistics DOI 10.1007/s00180-015-0637-z

Quintelier E (2010) The effect of schools on political participation: a multilevel logistic analysis. Research Papers in Education 25(2):137–154, DOI 10.1080/02671520802524810

Rabe-Hesketh S, Skrondal A, Pickles A (2002) Reliable estimation of generalized linear mixed models using adaptive quadrature. Stata Journal 2(1):1–21

Schaul T, Zhang S, LeCun Y (2013) No More Pesky Learning Rates. Journal of Machine Learning Research 28(2):343–351, arXiv:1206.1106v2

Skrondal A, Rabe-Hesketh S (2004) Generalized latent variable models: multilevel, longitudinal, and structural equation models, vol 17. DOI 10.1007/BF02295939

Stein C (1956) Inadmissibility of the Usual Estimator for the Mean of a Multi-Variate Normal Distribution. In: Proc. Third Berkeley Symp. on Math. Statist. and Prob, vol 1, pp 197–206

Xu W (2011) Towards optimal one pass large scale learning with averaged stochastic gradient descent. CoRR abs/1107.2490

Young-Xu Y, Chan KA (2008) Pooling overdispersed binomial data to estimate event rate. BMC medical research methodology 8:58, DOI 10.1186/1471-2288-8-58