

Bachelor Thesis by
Johannes Steinrücke
s1313134

Identifying Student Disengagement Using Time series Techniques

Department of Research Methodology, Measurement and Data

Analysis

Enschede, June 2015

Examination Committee

Prof. Dr. ir. Jean-Paul Fox

Drs. Sebie Oosterloo

Abstract

The use of online learning environments becomes more and more popular. Consequently also the problem of impairments to the learning process is arising. One of the known major impairments is the disengagement of students. To reduce the impact of disengagement on the learning process, a logistic autoregressive regression model was proposed and tested, using CUSUM control charts and log-likelihood analyses. By including indicators of disengagement the model became suitable for direct application. The chosen statistical methods and their suitability for analyzing the underlying processes are discussed.

A statistical process analysis was conducted on data retrieved from 'Leo's Pad', to evaluate the suitability of the proposed logistic autoregressive regression model. The model is a valuable tool for the detection of learners' disengagement and could help to improve the quality of online learning environments.

Acknowledgements

Special thanks go towards Prof. Dr. ir. Jean-Paul Fox. He patiently introduced me into methodological research methods, and also guided me, when I had questions concerning the statistical methods or questions concerning the programming with R. Learning R programming was crucial for my thesis. For a beginner it was demanding. I thank Prof. Fox for his ongoing support and for the introduction into R. He patiently answered all my questions, so that I was able to model the data in R correctly. Further thanks also go to Josine Verhagen, from Kidaptive, for providing the data I have analyzed.

Table of Contents

Introduction	5
General Introduction	5
Disengagement and its indicators	5
Leo's Pad	8
Statistical process control	8
CUSUM Control Chart	10
Method	12
Constructing the variables	13
The proposed logistic autoregressive regression model	13
Constructing time intervals	14
CUSUM Control Chart for Leo's Pad	15
Likelihood-ratio testing of the in-control versus out-control process status.....	16
Data-Analysis & Results	17
CUSUM Control Chart	17
Likelihood-ratio analysis	19
Testing effect of the parameters	19
Testing control status of learning process	20
Conclusion about the results	22
Discussion	23
References	25
Appendix	27

Introduction

General Introduction

Detecting disengagement of students, in modern learning environments like in the game ‘Leo’s Pad’ for the iPad, is an important topic. The typical users of the educational iPad game are preschool learners, which must solve tasks of different levels of difficulty. In the course of solving these tasks, data such as the time required and the correctness are saved per item and learner. A model for these data is presented to detect potential disengagement of students. This opens up a possibility to stimulate and support the process of learning through feedback, when disengagement occurs. Furthermore, feedback can be given to learners and developers. Through feedback loops, developers can introduce an intervention during the game, which can enhance the learning process. (Gowda, Baker, Corbett & Rossi, 2013)

First, an introduction is given to learner’s disengagement in modern learning environments. For this purpose, the game Leo’s Pad is described in more detail. That allows a better understanding of the context. Additionally, statistical process control is explained, together with suitable statistical models. Second, methods of modeling and analyzing the gained data are described. The methods are applied on fictional data and real data received from the game Leo’s Pad. Finally, a discussion of the results and a conclusion are given. The suitability of the proposed methods to identify student disengagement is discussed as well.

Disengagement and its indicators

Learners’ disengagement often results in less efficient learning. As Gowda et al. (2013) discussed, a student needs to understand the taught material deeply, when he/she wants to learn it. Otherwise, he/she experiences shallow learning, or no learning at all. One important defining factor for shallow learning is the student’s engagement-level in the learning environment. The learner can be engaged, thus interacting with the environment in an active way and trying to understand the provided knowledge. At the other end of the

spectrum there is the disengaged learner, with only shallow learning. It is typical for shallow learning, that such a student is only able to use his knowledge later in the exactly same environment. He/she is unable to transfer his knowledge to other environments (Gowda et al., 2013). In the game Leo's Pad this transfer also is important. The game was developed for preschoolers, who intend to gain basic knowledge, by playing Leo's Pad. Therefore, "gaming the system", how Gowda et al (2013) define disengaged playing or answering, will soon also imply a poor understanding of the offered knowledge. Consequently, that leads to what is termed as shallow learning. Under such conditions 'Leo's Pad' would be underperforming below its real possibilities. Therefore for improving effective learning with Leo's Pad, it is crucial to detect such disengagement events.

In the following, two indicators of disengagement are described. These indicators will be included in the statistical model.

To detect disengagement it is required to explore which data for disengagement exist, and which of them can be measured. Here, the response time of the learner and the accuracy of his/her response are used to identify disengagement. The objective is to identify inconsistencies in the responses of the learners. Any inconsistency can be viewed as a change in learner's response behavior, and might indicate disengagement of intended learning goals.

The response time of a learner is often used as an indicator of disengagement. For example, Cocea and Weibelzahl (2007) used this measure in a log file analysis towards motivation diagnosis. They made a distinction between moving very fast through pages, and spending too much time on one page, to describe two types of disengaged learners. This distinction can improve disengagement prediction and detection (Cocea & Weibelzahl, 2009). In the same study, they suggested to use this distinction in learning environments, where the occurrence of disengagement plays an important role. The negative consequences of these two disengagement behaviors were described in the study from Gowda et al. (2013). They

state, that “fast guesses may lead to no learning at all”. Furthermore, students who use this method of systematic guessing, as well as students who spend long pauses not of task, are more likely to learn just shallowly. They do not gain a deeper understanding of the given material. Deeper understanding would deliver the ability to use the gained knowledge also in other contexts, than those during the initial learning. . Shallow learning does not support this ability (Gowda et al., 2013).

Besides the response time, also the results of the previous responses are a useful indicator. This parameter has been also used by others like, for example Cocea and Weibelzahl (2011) in their validation studies. Apparently they did not find any additional use in analyzing this measure alone. Note, the score analyzed here is retrieved from a content delivery system. Cocea and Weibelzahl (2011) state that in problem-solving environments the score does have a significant influence in detecting if the learner is disengaged. In the case of Leo’s Pad it is possible, that the score of a learner becomes extreme in one of two possible directions. It can become too high, which may indicate that the learner gets significant help, or it can become too low, which may indicate that the student lacks motivation to solve the tasks, respectively that the student is following off-topic activities. In both cases it is useful to pause the game, so that the level of shallow learning can be limited.

Beck (2005) also used response time and response accuracy to model learners’ disengagement. Beck tried to introduce an approach to detect the level of engagement from individual students. Beck’s model (2005) is “sensitive enough to detect temporal changes in the student’s level of engagement within a single session of using the tutor” (Beck, 2005, p. 94). This tutor is, just as Leo’s Pad, an intelligent system, which gives the ability of measuring time and accuracy of an individual learner (Beck, 2005, p. 94).

Leo's Pad

As mentioned earlier, Leo's Pad is an educational iPad Game. It is developed by Kidaptive. Leo's Pad is intended for preschoolers, and it teaches them 25 different skills. The skills are taught by letting the learner explore a digital world. The virtual world places the various skills in meaningful contexts, so that, for example, the basics of math are taught in an entertaining way. The learners are allowed to learn naturally. They start by following their own ideas, and afterwards they observe what actually followed their actions.

As noted, the game takes place in a virtual world. Here, the preschoolers interact with different kinds of characters. The characters, called Leo and his friends, help the learner to understand the previous explored skills, by displaying the skills in a clear way for the child. The learner can play mini-games, so that he/she can practice his new skills.

Furthermore, Leo's Pad analyzes the current skill-level of a learner, so it can adapt to the skill level of him/her. Because of this, the tasks which are to solve do not become too difficult or too easy. Thus, Leo's Pad adjusts the level of difficulty dependent on the learner's skills. It behaves like an adaptive system. (Kidaptive, 2014)

Statistical process control

The data to be modeled is retrieved from Leo's Pad. It is longitudinal data, every observation is unique and obtained at a specific time point. Two types of data, response times and response accuracy, are used to apply statistical process control. The response accuracy consists of the outcomes of the individual items. The outcomes are dichotomous. On the one hand, it can be 'right', on the other hand, it can be 'wrong'. The time is continuous and restricted to be positive. For example, learner A needs just five seconds to answer, learner B needs ten seconds and learner C needs 17 seconds. It is possible, that all learners give the right answer. But also one or two, or all three of them could give an incorrect answer. These two sources of data can be used in statistical process control.

Woodall (2000) gives the following statement over statistical process control: “Statistical process control (SPC), a sub-area of SQC [Statistical Quality Control], consists of methods for understanding, monitoring, and improving process performance over time.” Statistical process control, from now on referred to as SPC, can be used to monitor an ongoing process. Various statistical methods can be used to do so, one combination of methods, suited for Leo’s Pad, will be discussed. SPC is monitoring an ongoing process, so constantly new data are observed. These new data are analyzed under the assumed process model, so significant process changes are detectable with SPC. When there is a significant change, the process can be out of control.

Figure 1 shows a fictional process, which at time point 6 shows a peak. This peak, marked red, is an example of the process being out of control.

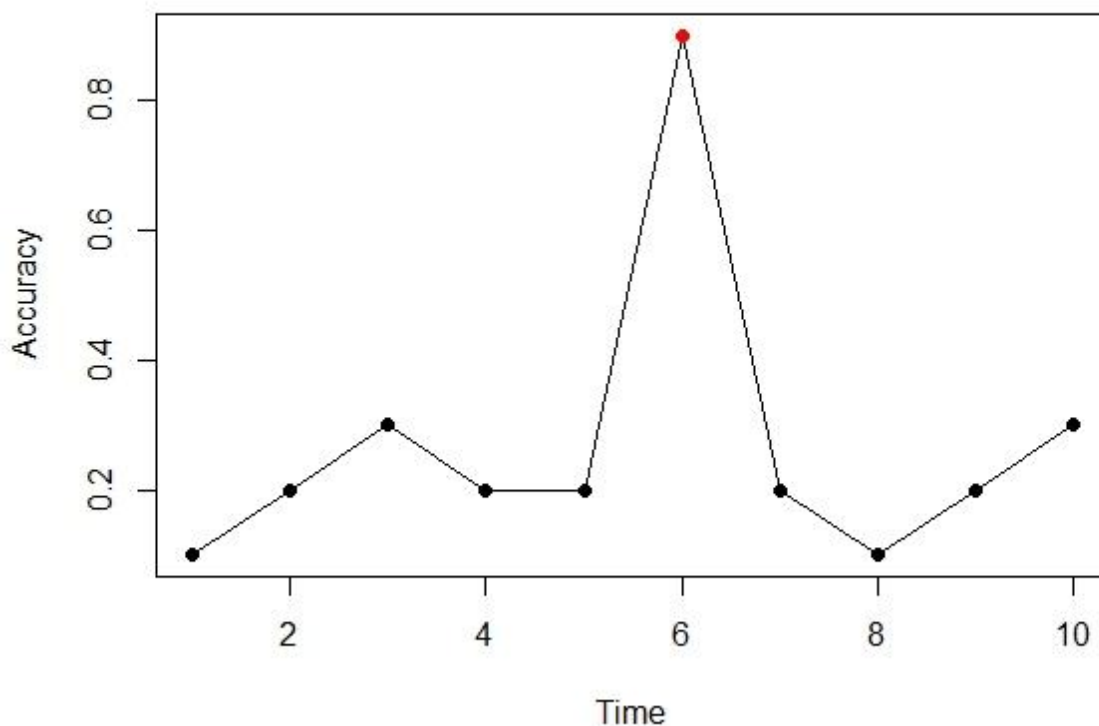


Figure 1. Process out of control.

With this detection method, the opportunity to introduce interventions is given. For example, this could be a pop up, which delivers a message to take a break, if the performance on a number of tasks gets significant worse. (Woodall, 2000)

A logistic regression model with an autoregressive component is used as the baseline model in applying SPC. The logistic regression model is used to describe and test hypotheses “about relationships between a categorical outcome variable and one or more categorical or continuous predictor variables.” (Peng, Lee & Ingersoll, 2002). So, it is suited in contexts where the outcome variable is dichotomous, thus ‘right’ or ‘wrong’. The autoregressive component derives from the AR-model (auto-regressive model), that broadly speaking states, that a following response is influenced by the previous responses (Fahrmeir & Tutz, 2001). Applied on Leo’s Pad, this means that the possibility of getting a right answer might be dependent on the results of previous tasks.

CUSUM Control Chart

One way of applying SPC is the cumulative sum (CUSUM) control chart. It gives the opportunity to monitor a time series process over several sessions, where a simple control chart would just include the most recent session (Barnard, 1959). The process over the last set of sessions stands in focus, so the CUSUM control chart will be suitable. In the context of Leo’s Pad, the CUSUM control chart uses data of previous sessions, to define an upper boundary and a lower boundary for the learner’s ability in the current session, based on a statistical model. This stands in contrast to a simple control chart, which would just use data of the most recent session. Concluding, a change point in a stochastic process is more accurately defined by a CUSUM control chart, than by a simple chart. As a result, smaller shifts can be detected more easily. (Barnard, 1959)

As Montgomery (1991) states, control charts allow the user to do a process control, by defining a suitable value range for the parameter under investigation. A CUSUM control

chart gives an upper control line (UCL) and a lower control line (LCL). The values of these two lines are normally two or three standard deviations larger, respectively, smaller, than the mean (Montgomery, 1991).

Figure 2 is a fictional example of a CUSUM control chart, showing an out of control process. The red points are indicators that the process can be out of control. The chart contains higher or lower values than the UCL, here in blue, respectively LCL, in green.

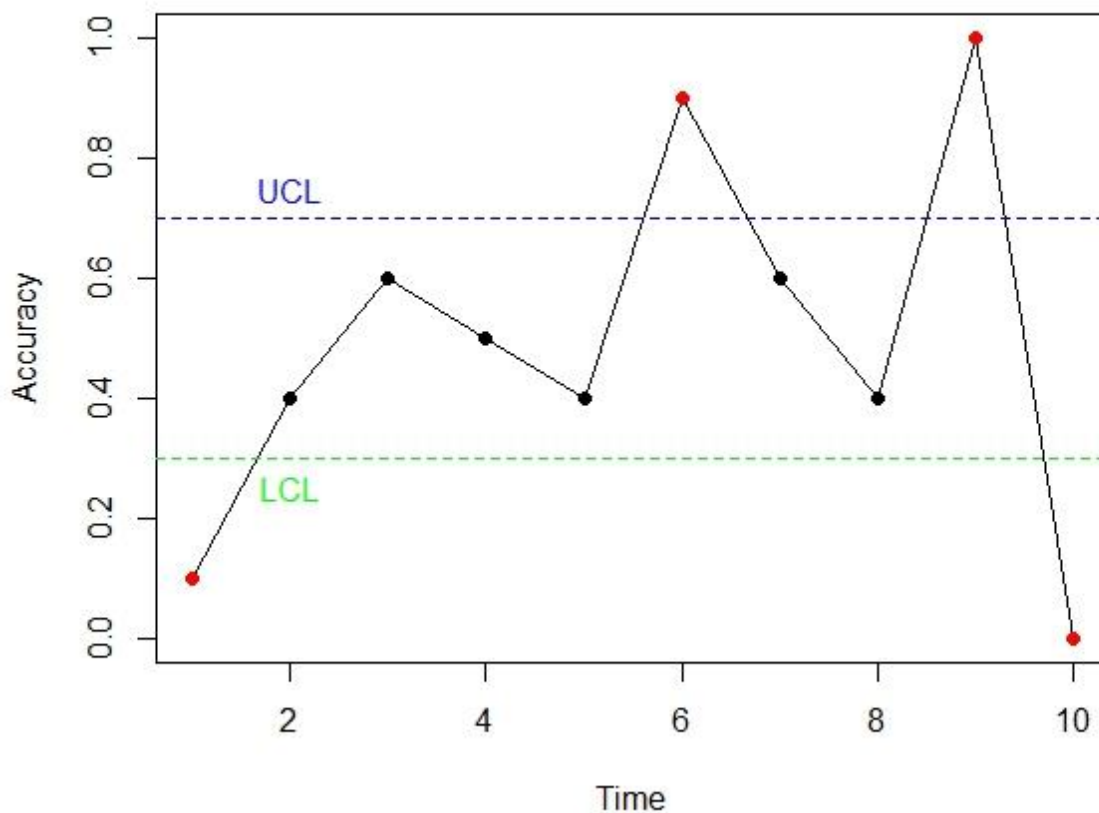


Figure 2. CUSUM control chart, of a process out of control.

Within Leo's Pad, a CUSUM control chart could show as UCL the highest possible intercept value of the learner, the LCL the lowest possible value, given data of a process that is in control under the assumed baseline model. The named intercept represents the learners' ability, given earlier observations. If a learner's intercept value is above the UCL, the learner

could be getting help, or experience a learning effect. Here, a further analysis is needed to give an accurate conclusion. If the intercept value remains larger than the UCL, a learning effect occurred, and the learner has consistently improved the accuracy of his/her responses. If it gets back to a value lower than the UCL, it could mean that the learner was getting assistance to improve his/her results. If the intercept value becomes lower than the LCL, the learner could be disengaged, the process would be out of control. By using a CUSUM control chart this change of the intercept value is detectable at an earlier state, than by using a simple control chart.

Method

A likelihood-ratio test and a logistic regression with an autoregressive component was conducted to apply SPC on data retrieved from Leo's Pad. Before the logistic regression was conducted, it was necessary to define the individual variables which were used. The first variable defined the autoregressive component. It consisted of four parts. Those parts were the response accuracy and the response times of the two previous sessions or time intervals. The second variable consisted of the difficulty of the items. It consisted of the mean results. The third variable was the response time of the current session, respectively time interval in question.

The logistic autoregressive regression model had to be tested. To do this, a CUSUM control chart was constructed. For this purpose additional steps were necessary. The data was naturally split up into a number of time intervals. For each time interval a specific intercept value was calculated, by fitting the model on the data. Based on this value the hypotheses H_0 and H_a were formulated. These hypotheses afterwards were tested by applying the CUSUM method and calculating a likelihood-ratio.

Constructing the variables

The analysis was conducted within the program R (R Core Team, 2014). R allows the user to create data files, extract important data out of them and analyze them in various ways. The used R code, to construct the variables, is given in Appendix A. These variables were the outcomes and response times of the time intervals t-1 and t-2.

Secondly, the difficulty per item was calculated. The lower the mean, the more difficult the item was for the learners. The retrieved mean value was inverted, so that the difficulty, instead of the ‘easiness’, was mathematically represented.

The response time, as the third variable, was directly observed in the retrieved data. In a real setting it would be measured time specifically. The response times were used to calculate the mean response time of the current session.

The mean values over a session are used, because a single response time and result, would have little to no effect on the result of a next item, which possibly asked a completely different skill. By doing so, multiple items of different kinds were included. For example, if item 1 was a task where one had to calculate the sum of 1+1, it would have had no forecasting value for item 2, which asked to find the wrong spelled word (Liepmann, Beauducel, Brocke & Amthauer, 2007). Furthermore, an increase in the ability of a learner was assumed to not be noticeable from item to item, since items of one session should reflect a similar level of difficulty.

The proposed logistic autoregressive regression model

Learners playing for more than 100 sessions were more likely to have experienced a learning effect and disengaged sessions. So the data analysis was conducted on two of those learners. The linear term of the logistic autoregressive regression model was given by,

$$f(y(t)) = \beta_0(t) + \beta_1 * y(t - 1) + \beta_2 * RT(t - 1) + \beta_3 * y(t - 2) + \beta_4 * RT(t - 2) + \beta_5 * DC(t) + \beta_4 * RT(t),$$

where 't' referred to the time window. 'RT' was short name for the response time, the item difficulty was given by 'DC'. The parameter β_0 gave the intercept value of an individual learner, in time window t. It was the most interesting parameter, since it reflected the learner's ability in the current time window.

Because the autoregressive component, the difficulty and the response time were used in the logistic autoregressive regression model, the intercept gave a quite good estimate of the learner's ability. To plot the intercept value over time, the longitudinal data was divided in a number of time intervals. The optimal number of intervals varied across the individual learners.

Two hypotheses were formulated to test the control status of the process. Note, the difficulty of the items was expected to increase over time, so the ability of the learner should have increased too. As the item difficulty was included in the model, it was already accordingly corrected. The function of the outcome was expected to remain on a constant level, which played a significant role in detecting learner's disengagement.

This led to the hypothesis H_0 , which stated that there was no significant difference between the intercept values in time interval t and time interval t-1. The alternative hypothesis H_a , stated that the two values differed significantly. The null hypothesis was constantly tested over time, such that a decision could be made whether a learner's ability has been changing.

Constructing time intervals

Changes in the learner's ability were shown, by plotting the intercept values of one learner in a CUSUM control chart. Therefore, the data of an individual learner needed to be blocked in different time intervals. However, it was required to guarantee a good balance between how many answers were given, and how much time lied between the answers. Using one short time interval with ten sessions completed, and one long time interval with two

sessions completed, would have resulted in an unreliable decision about the process control status. Not every learner completed the same number of sessions, and if so, they needed different amounts of time. It was therefore decided, to define the time intervals per individual learner, by time difference in days. The time intervals were created, by defining time differences in reference to the first completed item. The used R code to calculate the time differences and to compute the intervals, is given in Appendix B.

CUSUM Control Chart for Leo's Pad

As described a CUSUM Control Chart contained an UCL and a LCL, defined by the confidence interval borders of the first time window (time window 1). These two lines were the boundaries for the intercept values in the following time windows. Hypothesis H_0 stated here, that the intercept in the current time window lied within the boundaries. Hypothesis H_a stated, that the intercept in the subsequent time windows fell outside the earlier defined boundaries.

The parameters used to define H_0 were not updated, when the intercept of the ongoing process remained in, or kept going back to lie, between the UCL and LCL. However, if there was a significant difference between the intercept values in time interval 1 and interval t, the intercept and the boundaries were only updated, if the following intercepts differed significant as well. They had to point towards the same direction, thus either lower or higher. The intercept had to be replaced by the previous intercept value of the alternative hypothesis, the other parameters by the remaining parameters of H_a . When the process parameters were updated, new boundary values UCL and LCL were calculated. The data from the time window where the process was out of control, were used. To test the process status for the subsequent time windows, the updated boundaries would have been used. The R code is given in Appendix C.

Afterwards, the CUSUM control chart was constructed. Each intercept value was plotted together with the UCL and LCL. The corresponding R code is given in Appendix D.

Likelihood-ratio testing of the in-control versus out-control process status

To evaluate the evidence in favor of the in-control-process status to the out-control-status, a likelihood-ratio test was conducted. The log-likelihoods were calculated under the assumptions, that H_0 is true (LL0), and that H_a is true (LL1). The difference between log-likelihoods represented the evidence in favor of the null hypothesis to the alternative hypothesis. It could be larger than zero, and H_0 was supported, or lower than zero, and H_a was supported. It gave a direction of the data in evidence to one of the hypotheses.

The log-likelihood differences could be tested, since two times the absolute difference was approximately chi-squared distributed, that is $X^2 = -2 * (LL1 - LL0)$. A negative result, would have meant that the alternative hypothesis was more likely than the null hypothesis. A positive result, just as well as a non-significant negative result, indicated the opposite, H_0 had to be supported. The negativity only implied which of the hypotheses was more likely. The R code to create a likelihood function is given in Appendix E.

Furthermore, a likelihood-ratio was used to test if specific variables had a significant effect on the results. To test the variables, the log-likelihoods of the full model, with all variables considered, and of an alternative model, which did not contain the variable in question, were calculated. Again the differences of log-likelihoods were used to give a decision about the effect of the variable. In Appendix F the used R code is listed.

For each time interval log-likelihoods were calculated. Likelihood-ratio tests were conducted, to test if there were significant differences between the log-likelihoods of two time windows. The used R code is listed in Appendix G.

Data-Analysis & Results

Two kinds of analysis were carried out. First, a CUSUM control chart was generated. Second, the likelihood-ratio tests were conducted. With this method, the process status and the effect of the parameters on the current response were tested.

CUSUM Control Chart

The first learner analyzed played Leo's Pad regularly in eight time windows. In total he gave more than 500 answers. Learners giving that many answers were unlikely to have guessed all responses, so fluctuations in their abilities were likely to be based on their individual skill levels. Figure 3 showed the process of learner 1, the y-axis showed the intercept value, representing the learner's ability, the x-axis the corresponding time windows.

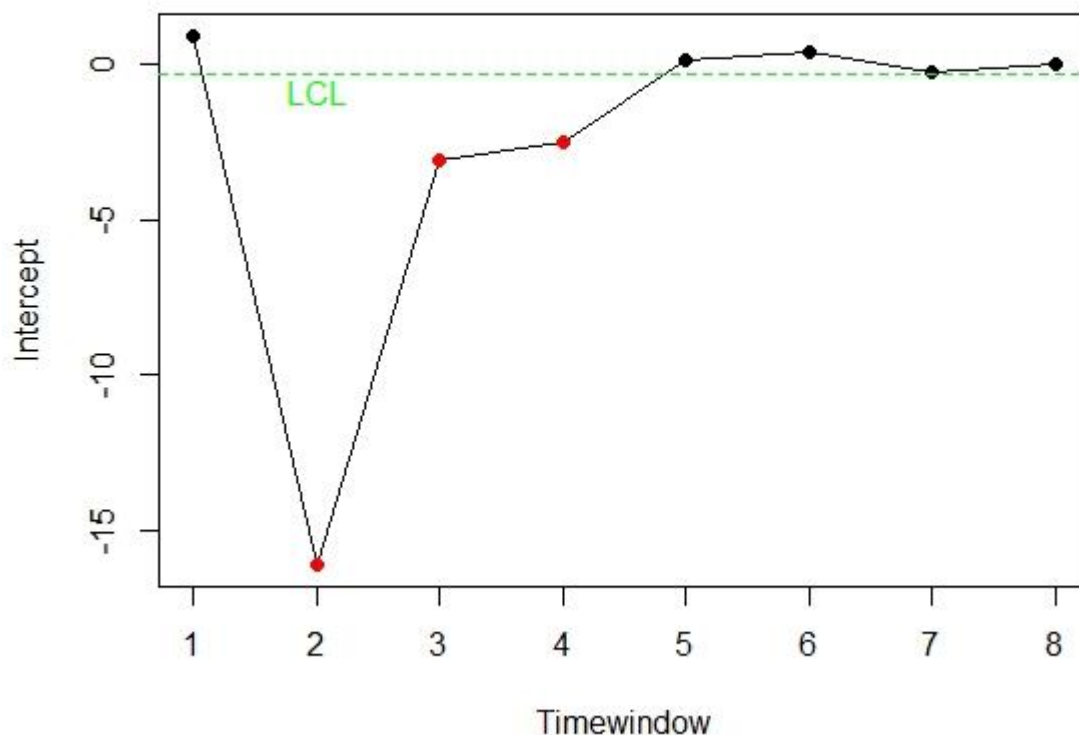


Figure 3. CUSUM Control Chart – learner 1

Figure 3 showed, that the process could have been out of control, in three time windows. The LCL showed the lower intercept boundary of an in-control process. The intercepts in time window 2, time window 3 and time window 4 lied below this boundary. At a first glance, the model needed to be updated, since the learner showed a big decrease in performance. However, in time window 5 the performance improved significantly. The intercept value lied above the LCL. This stated, that the process was again in control. As the intercept remained at this level, it was concluded that the process was out of control from time window 2 to time window 5.

The process shown in Figure 4 belonged to learner 2. The process of this learner got out of control in both possible ways. In time window 2 and 3, the intercept value became too small. In time window 6, the graph showed an intercept value, which was too big.

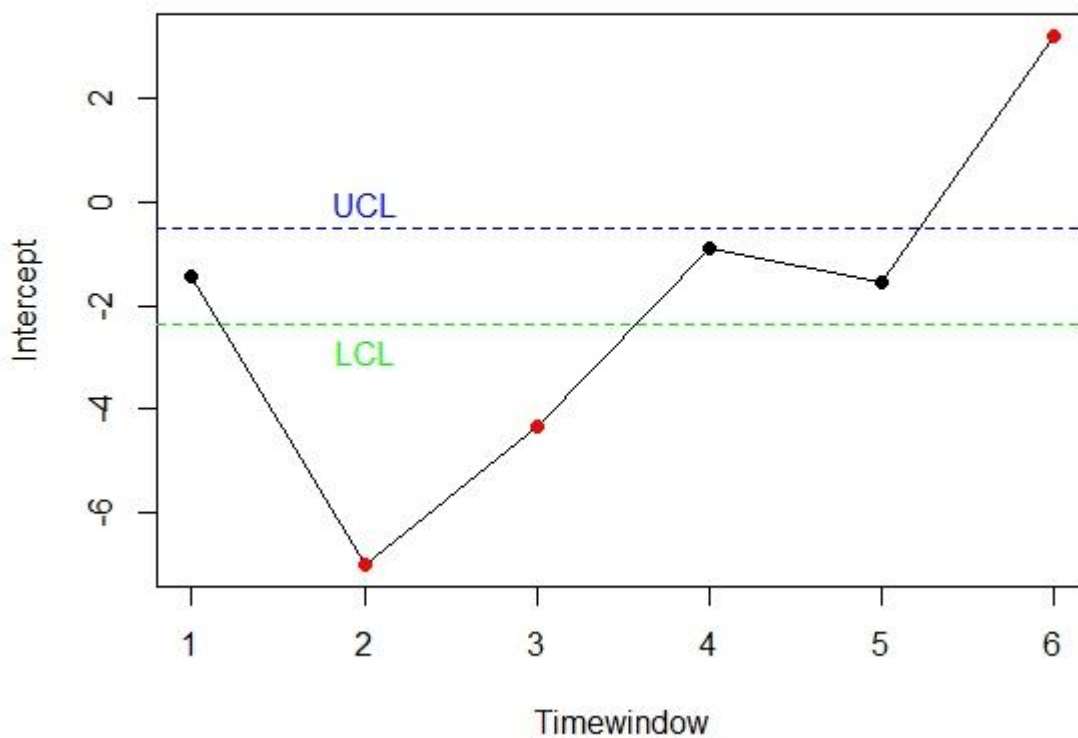


Figure 4. CUSUM Control Chart – learner 2

As for learner 1, the process was out of control in three time windows. In this case, the intercepts in time windows 2 and 3 could imply that learner 2 was disengaged, while the too-big intercept in time window 6 could imply that the learner experienced a learning effect or experienced help. No definitive conclusion could be made with the available data.

In both, Figure 3 and Figure 4, the process was at some point out of control. The CUSUM control charts showed in which time windows this was the case. However, it was required to fit the model in each time window to obtain all parameter estimates and to test the control status of the process with a CUSUM control chart.

Likelihood-ratio analysis

For the likelihood-ratio analysis just one learner was considered. The chosen learner provided a large set of answers, so that a relatively long time pattern was observed. The analysis started with testing the effect of the variables on the outcome in time window 1.

Testing effect of the parameters. For learner 2, just as for every other learner, the parameters of the logistic autoregressive regression model were used to calculate the intercept. The intercept represented the ability level of the learner. Subsequently, the effect of the item difficulty on the current response was tested. The log-likelihood of the full model (LL0) resulted in $L = -208.0735$, the log-likelihood of the alternative model (LL1) equaled $L = -243.8802$. As the difference was greater than zero ($LL0 - LL1 = 35.80672$), it was concluded that the difficulty had a significant effect on the response. The chi-square value equaled $X^2 = 71.61345$, $p < .01$, with one degree of freedom. With that, it was highly significant.

Testing the effect of the variable RT, representing the response time of the actual session, led to a chi-square value of $X^2 = 0.0041$. It was concluded that the response time of the current time window had no significant effect on the response. It was possible that in time

window 2, the RT actually had an effect. This was tested, and a log-likelihood difference value of $X^2 = 35.6034$, $p < .01$ was obtained, which indeed was significant.

The last four variables were not viewed separately, since they constituted the autoregressive component. So the effects of the response times and outcomes of time window t-1 and t-2, were tested together. The chi-square value equaled $X^2 = 36.24045$, $p < .01$. This implied a high significance, so it was to conclude, that the autoregressive component effected the response in the current time window.

Concluding, in the first time window of this learner, the autoregressive component and the item difficulty had a significant effect on the outcome. The response time was not relevant in time window 1. However, it was not disregarded. The response time had an effect in subsequent time windows.

Testing control status of learning process. Afterwards, attention was focused on the intercept. As from the CUSUM chart derived, the assumption was made, that under the same circumstances this value got lower in time window 2. So starting with the assumption that all parameters did not alter during the whole process, the process was in control, when the intercept in time window 2 lied between the boundaries. The null-hypothesis H_0 stated, that the intercept equaled the intercept calculated in time window 1. The alternative hypothesis H_a states, that the intercept in time window 2 was equal to, or smaller than the lower boundary. The results were clear. The chi-square value equaled $X^2 = 83.31489$, $p < .01$, the calculated log-likelihood difference was negative. This implied significant support for H_a , so it was concluded that the learner's ability in time window 2 was not as good as in time window 1. It was controlled for other differences. The process was, in time window 2, out of control.

Looking at the CUSUM chart, two more time windows seemed to be interesting. In time window 4, the learner's ability returned to control following this chart, and in time window 6 his ability became too high. This was tested with the likelihood-ratio analysis.

In time window 4, the alternative hypothesis H_a stated, that the intercept value differed significantly from the intercept in time window 1. H_0 stated, that the intercept in time window 4 equaled the intercept in time window 1. For this time window there were in total two chi-square values calculated, one to test if the intercept lied below the LCL, and one to test if it lied above the UCL. The first, testing against the LCL, indicated that H_0 was supported. The chi-square equaled $X^2 = 99.22657$, $p < .01$. The result from testing against the UCL was $X^2 = 9.577184$, $p < .01$, the difference in this case was negative. The alternative hypothesis was supported. The learner's ability in time window 4 tended to be higher than in time window 1, but still lied within the boundaries. Here the model could be updated. But because it was not practical to update the model's parameters too soon or too late, the following time window had to be considered. Following the CUSUM control chart the intercept was lower in time window 5 than in time window 4, so the conclusion was made to not update the null-model.

Following Figure 4, the graph showed a very large intercept value in time window 6, compared to time window 1. H_a stated, that the learner's ability was significantly higher than in time window 1. H_0 stated, the two intercept values differed significantly. With a chi-square value of $X^2 = 55.09325$, $p < .01$, and a negative deviance indicating that the alternative hypothesis had to be supported, it was highly significant. So, in the sixth time window the process seemed to be out of control. Since the CUSUM control chart showed such a large difference between the intercepts of time window 5 and time window 6, it was unlikely that all the differences were based on a learning effect. Under this assumption, the conclusion was made, that the process was out of control in time window 6.

In Table 1 the log-likelihoods and the corresponding differences were shown. These likelihoods derived from testing with the lower boundary as intercept for H_a .

Table 1

Lower Boundary as intercept from alternative hypothesis

	Time window					
	1	2	3	4	5	6
LL0	-208,0735	-251,3097	-112,9159	-159,0299	-178,5477	-178,267
LL1	-240,4269	-209,6523	-87,73059	-208,6432	-197,18498	-207,2132
Difference	32,35335	-41,65744	-25,18531	49,61329	19,18498	28,94626

Table 2 showed the log-likelihoods and differences, as derived from testing with the upper boundary as the intercept of H_a .

Table 2

Upper Boundary as intercept from alternative hypothesis

	Time window					
	1	2	3	4	5	6
LL0	-208,0735	-251,3097	-112,9159	-159,0299	-178,5477	-178,267
LL1	-237,3249	-291,9108	-139,4897	-154,2413	-210,9129	-150,7203
Difference	29,25139	40,60107	26,57378	-4,788592	32,36518	-27,54663

Conclusion about the results

Note, the differences given in Table 1 and in Table 2 supported the previous analysis. Significant differences supporting the alternative hypotheses, that the ability was much lower than in time window 1, were shown in time windows 2 and 3. Time window 6 contained a significant difference, which supported the alternative hypothesis, that the learner's ability is higher than in time window 1. Even in time window 4 a trend was shown, that there is a significant difference in the learner's ability towards time window 1.

All in all similar results were obtained for the likelihood-ratio test and the CUSUM chart. Both implied that the process of this learner was out of control in the second and in the sixth time window. Furthermore, both implied that the learner's process returned to control in time window 4.

Discussion

First, note the original purpose of this study. As learners' disengagement in online learning environments can impair the learning process, it is important to detect disengagement as soon as possible. As described in the introduction, disengagement may lead to a shallow learning experience, which means that no generalization of the obtained knowledge is possible. So to be able to detect learners' disengagement, and to prevent shallow learning, a model and suitable methods to monitor and evaluate the learning process are needed. The model requires the important indicators for learners' disengagement, in fact the response time and the response to each item.

To test the suitability of the logistic autoregressive regression model, a log-likelihood analysis was conducted. It was used to test the effect of the model's parameters on the response. The data supported the model. So under this model, the ongoing process could be evaluated using statistical tests. This was done using the likelihood-ratio test and the CUSUM control chart.

The autoregressive component was tested as a single parameter, since it reflected the performance of the previous sessions. Its effect on the outcome varied from time window to time window, while the effect remained significant. The second parameter, the difficulty, also had a significant effect. The probability of a correct response, depended on the difficulty of each question. The third parameter, the effect of the response time was also significant, except for time window 1. The response time remained part of the model. The parameter estimates were used to monitor the process.

Afterwards, the process was analyzed. First the CUSUM control chart showed, in which time windows the learner's ability was lower or higher than usual. Second, the likelihood-ratio analysis supported those results.

The retrieved results implied, that it was possible to monitor a learning process with the CUSUM control chart and to evaluate it with the likelihood-ratio analysis. Although no further distinction why the process was out of control was possible, a warning or an intervention could have been installed, whenever the process was out of control. Furthermore, by monitoring the process status also a message could be given, saying that the learner is doing good work. Intervening and supporting messages and/or pop ups can be put on display.

But this research has also some implications. First, the time windows used were comparatively big. So monitoring the process in short term environments was not possible. Each time interval was 50 days long, they overlapped 25 days. As disengagement can also occur within very short time periods, the tests were not sensitive enough to detect differences in a 'live' setting. Second, further testing needs to be done. The methods were tested with the data of just one learner. Besides testing more learners, other online learning environments need to be analyzed as well. Summarized, the developed methodology can be generalized to other games.

All in all detecting students' disengagement with the used techniques is possible. A logistic autoregressive regression model can be used to monitor ongoing learning processes with a CUSUM control chart. The log-likelihood analysis is suited to evaluate the process status. Further research could include a set of experiments, where learners are told to be disengaged at some point, or where disengagement is enforced from outside. Such approaches would allow to implement disengagement and evaluate the process in a more controlled way. This counts for small time windows as well. The refined methods would then also allow a valid explanation for the changes on shorter time scale.

References

- Barnard, G.A. (1959). Control Charts and Stochastic Processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, 21(2). 239-271.
- Beck, J. (2005). Engagement tracing: using response times to model student disengagement. In C. Looi, G. McCalla, B. Bredeweg & J. Breuker (Eds). *Artificial Intelligence in Education* (pp 88-95)
- Cocca, M. & Weibelzahl, S. (2007). Eliciting motivation knowledge from log files towards motivation diagnosis for adaptive systems. In C. Conati, K. McCoy & G. Paliouras (Eds). *User Modeling 2007* (pp 197-206). doi: 10.1007/978-3-540-73078-1_23
- Cocca, M. & Weibelzahl, S. (2009). Log file analysis for disengagement detection in e-Learning environments. *User Modeling and User-Adapted Interaction*, 19(4), 341-385. doi: 10.1007/s11257-009-9065-5
- Cocca, M. & Weibelzahl, S. (2011). Disengagement Detection in Online Learning: Validation Studies and Perspectives. *IEEE Transactions on Learning Technologies*, 4(2), 114-124. doi: 10.1109/TLT.2010.14
- Fahrmeir, L. & Tutz, G. (2001). *Multivariate Statistical Modelling Based on Generalized Linear Models*. doi: 10.1007/978-1-4757-3454-6
- Gowda, S.M., Baker, R.S., Corbett, A.T. & Rossi, L.M. (2013). Towards automatically detecting whether student learning is shallow. *International Journal of Artificial Intelligence in Education*, 23(1-4), 50-70. doi: 10.1007/s40593-013-0006-4
- Kidaptive Inc. (2014). Leo's Pad. Retrieved from <http://kidaptive.com/leos-pad/>
- Liepmann, D., Beauducel, A., Brocke, B. & Amthauer, R. (2007). *Intelligenz-Struktur-Test 2000 R (IST 2000 R). Manual (2. erweiterte und überarbeitete Aufl.)*. Göttingen: Hogrefe Verlag.

Looi, C., McCalla, G., Bredeweg, B. & Breuker, J. (Eds.). *Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology* (pp. 88–95). Amsterdam, NL: IOS Press.

Montgomery, D.C. (1991). *Introduction to statistical quality control*. New York: Wiley.

Peng, C.J., Lee, K.L. & Ingersoll, G.M. (2002). An Introduction to Logistic Regression Analysis and Reporting. *The Journal of Educational Research*, 96(1), 3-14. doi: 10.1080/00220670209598786

R Core Team (2014). R: A language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria. Retrieved from <http://www.R-project.org/>.

Scrucca, L. (2004). qcc: an R package for quality control charting and statistical process control. *R News* 4(1), 11-17.

Woodall, W.H. (2000). Controversies and contradictions in statistical process control. *Journal of Quality Technology*, 32(4), 341-350. Retrieved from http://myplace.frontier.com/~stevebrainerd1/STATISTICS/ECE-580-DOE%20WEEK%203_files/Woodall%20paper%20Control%20charts.pdf

Appendix

Appendix A: The variables of the proposed model

difficulty of the questions

```
moeilijk ← tapply(1- as.numeric(data$outcome), data$item, mean)
data$moeilijkheid ← matrix(0,ncol=1,nrow=N)
itemnummer ← tapply(data$item, data$item, mean)
for(ii in itemnummer){
  data$moeilijkheid[which(data$item==ii)] ←moeilijk[names(itemnummer)==ii]}
moeilijkheid ← moeilijk
```

Variables per session

```
Nsession ← tail(datax$session, n=1)
```

mean difficulty per session

```
datax$mo ← matrix(0,ncol=1,nrow(datax))
mo ← c(tapply(datax$moeilijkheid,datax$session,mean))
for(m in 1:Nsession){
  datax$mo[which(datax$session==m)] ← mo[m]}
```

mean response time per session

```
datax$rt ← matrix(0,ncol=1,nrow(datax))
rt ← c(tapply(datax$responsetime, datax$session, mean))
for (tt in 1:Nsession){
  datax$rt[which(datax$session==tt)] ← rt[tt]}
```

response time sessionB

```
datax$rtB ← matrix(0,ncol=1,nrow(datax))
for(ttt in 1: Nsession){
  datax$rtB[which(datax$sessionB==ttt)] ← rt[ttt]}
```

```
# response time sessionC
```

```
  datax$rtC ← matrix(0,ncol=1,nrow(datax))
```

```
  for(ttt in 1:Nsession){
```

```
    datax$rtC[which(datax$sessionC==ttt)] ← rt[ttt]}
```

```
# mean outcome per session
```

```
  datax$oc ← matrix(0,ncol=1,nrow(datax))
```

```
  oc ← c(tapply(as.numeric(datax$outcome), datax$session, mean))
```

```
  for (tt in 1:Nsession){
```

```
    datax$oc[which(datax$session==tt)] ← oc[tt]}
```

```
# outcome sessionB
```

```
  datax$ocB ← matrix(0,ncol=1,nrow(datax))
```

```
  for(ttt in 1:Nsession){
```

```
    datax$ocB[which(datax$sessionB==ttt)] ← oc[ttt]}
```

```
# outcome sessionC
```

```
  datax$ocC ← matrix(0,ncol=1,nrow(datax))
```

```
  for(ttt in 1:Nsession){
```

```
    datax$ocC[which(datax$sessionC==ttt)] ← oc[ttt]}
```

Appendix B: creating intervals based on time difference

```
# choose individual learner
```

```
  Datax ← data[data$learner==number of learner,]
```

```
# calculate time differences
```

```
datax$time ← as.POSIXlt(datax$timestamp)    # to be able to use timestamps
```

```
setT ← 1:nll["401481"]
```

```
datax$timediff ← difftime(datax$time[setT],datax$time[1],unit="days")
```

```
# create intervals dependent on time
```

```
  Ldatax ← length(datax$timediff)    # to retrieve LAST value in datax$timediff
```

```
  VHTD ← as.numeric(datax$timediff[Ldatax]) # maximum time difference as simple
  number
```

```
  VHTDX<-VHTD-10
```

```
  Nstep ← ceiling((VHTD-10)/5)    # number of steps, rounded up to include last one
```

```
  upper ← length interval, f.e.10    # start upper defined
```

```
  lower ← 0    # --"-- lower ---"---
```

```
  set ← list()    # create list
```

```
  for (i in 1:nstep){
```

```
    set[[i]] ← which((datax$timediff>=lower)&(datax$timediff<upper))
```

```
    upper ← upper+5
```

```
    lower ← lower+5 }
```

Appendix C: store Intercept values in list & calculating confidence intervals

```
# loop model over data
```

```
Value ← list()
```

```
for(j in 1:nstep){
```

```
  setJ ← unlist(set[j])
```

```
  value[[j]] ←
```

```
  glm(datax$outcome[setJ]~1+datax$ocB[setJ]+datax$rtB[setJ]+datax$ocC[setJ]+da  
  tax$rtC[setJ]+datax$mo[setJ]+datax$rt[setJ],data=datax,family=binomial(link="log  
  it"))}
```

```
# creating confidence intervals
```

```
CI ← list()
```

```
for (k in 1:nstep){
```

```
  intercept ← value[[k]]
```

```
  CI[[k]] ← confint(intercept,level=0.95)}
```

Appendix D: Using Confidence Intervals to apply the CUSUM-method

```
# extracting upper & lower boundaries of first time interval
```

```
Borders ← CI[[1]]
```

```
UpperBound ← Borders[1,2]
```

```
LowerBound ← Borders[1,1]
```

```
# "CUSUM" control
```

```
Control ← list()
```

```
for (k in 1:nstep){
```

```
  Intercept ← coef(value[[k]]["(Intercept)"]
```

```
  if ((LowerBound<Intercept)&(Intercept<UpperBound)){
```

```
    Control[[k]]=1 }else{
```

```
    Control[[k]]=0 } }
```

```
# plot CUSUM
```

```
IC ← list()
```

```
for (k in 1:nstep){
```

```
  IC[[k]] ← coef(value[[k]]["(Intercept)"] }
```

```
InterC ← unlist(IC)
```

```
Timewindow ← c(1:nstep)
```

```
plot(Timewindow,InterC,type="o",pch=16)
```

```
abline(h=UpperBound,col="blue",lty=2)
```

```
abline(h=LowerBound,col="green",lty=2)
```

```
text(x=2,y=UpperBound+0.5,labels="UCL",col="blue")
```

```
text(x=2,y=LowerBound-0.5,labels="LCL",col="green")
```

```
pnt ← identify(Timewindow,InterC,plot=F)
```

```
points(Timewindow[pnt],InterC[pnt],col="red",pch=16)
```

Appendix E: Likelihood-ratio – creating the function

```

# make function computelikelihood

  computelikelihood ← function(y,x,parameters,...){

# y new window: y

# x new window: x

# parameters : estimate log-likelihood given estimated parameters

  dots ← list(...)      ## ... Additional Arguments

# account for missing values, which will be ignored

  p ← length(parameters)

  x ← matrix(x,ncol=p,nrow=length(y))

  dum ← 1-apply(is.na(x),1,sum) # defines set of observed values

  setvalues ← which(dum==1)

  ynew ← y[setvalues]

# this corresponds to fitted values in GLM

# due to missing values in y or x: consider actual fitted cases

  estimatepar ← matrix(parameters,ncol=1)

  XBpred ← matrix(c(x%*%estimatepar)[setvalues],ncol=1)

  muhat ← exp(XBpred)/(1+exp(XBpred))

# log-likelihood binary data

  logl ← sum(ynew*log(muhat)+(1-ynew)*log(1-muhat))

  if(is.na(logl)){

    cat("log-likelihood is",logl,":: Missing values in predictors and/or
outcomes")
  }

  return(logl)}

```


Appendix F: Testing if parameters contribute to model - example

```
# Data in window 1
```

```
N ← length(datax$outcome[set[[1]])
```

```
y ← matrix(as.numeric(datax$outcome[set[[1]]]),ncol=1,nrow=N)
```

```
x ← matrix(c(rep(1,N),datax$ocB[set[[1]]],datax$rtB[set[[1]]],datax$ocC[set[[1]]],
  datax$rtC[set[[1]]],datax$moeilijkheid[set[[1]]],datax$rt[set[[1]]]),ncol=7,nrow=N)
```

```
# obtain glm object of window 1
```

```
out ← value[[1]]
```

```
p ← length(out$coefficients)      # number of model parameters
```

```
LL0 ← computelikelihood(y=y,x=x,param=out$coefficients)
```

```
# test if datax$moeilijkheid has an influence
```

```
paramH1 ← out$coefficients
```

```
paramH1[p-1] ← 0                  #restrict effect to be zero
```

```
LL1 ← computelikelihood(y=y,x=x,param=paramH1)
```

```
# test if datax$rt has an influence
```

```
paramH1 ← out$coefficients
```

```
paramH1[p] ← 0
```

```
LL1 ← computelikelihood(y=y,x=x,param=paramH1)
```

```
# influence of autoregressive component
```

```
paramH1 ← out$coefficients
```

```
paramH1[(p-5):(p-2)] ← 0
```

```
LL1 ← computelikelihood(y=y,x=x,param=paramH1)
```

```
# Deviance = -2*(LL1-LL0) is chi-square distributed: to test H0 versus H1
```

```
# Compute p-value
```

```
1-pchisq((-2*LL1)-(-2*LL0),1)
```

Appendix G: Testing intercept of following time window - example

```
# Data in window 2
```

```
p ← length(out$coefficients)
```

```
N ← length(datax$outcome[set[[2]])
```

```
y ← matrix(as.numeric(datax$outcome[set[[2]]),ncol=1,nrow=N)
```

```
x ←
```

```
matrix(c(rep(1,N),datax$ocB[set[[2]]],datax$rtB[set[[2]]],datax$ocC[set[[2]]],datax$rtC[set[[2]]],datax$moeilijkheid[set[[2]]],datax$rt[set[[2]]]),ncol=p,nrow=N)
```

```
# compute log-likelihood given parameter estimates window 2 using intercept window 1
```

```
out2 ← value[[2]]
```

```
out2$coefficients[1] <- -out$coefficients[1]
```

```
LL0W2 ← computelikelihood(y=y,x=x,param=out2$coefficients)
```

```
## H0 intercept equals intercept window 1
```

```
# H1 intercept equals UpperBound (Upper boundary of Confidence Interval of Intercept from window 1)
```

```
paramH1 ← out2$coefficients
```

```
paramH1[1] ← UpperBound
```

```
LL1W2 ← computelikelihood(y=y,x=x,param=paramH1)
```

```
1-pchisq((-2*LL1W2)-(-2*LL0W2),1)
```

```
# H1 intercept equals LowerBound (Lower boundary of Confidence Interval of Intercept from window 1)
```

```
paramH1 ← out2$coefficients
```

```
paramH1[1] ← LowerBound
```

```
LL1W2 ← computelikelihood(y=y,x=x,param=paramH1)
```

```
1-pchisq((-2*LL1W2)-(-2*LL0W2),1)
```