Cycle Time Prediction: When Will This Case Finally Be Finished?

B.F. van Dongen, R.A. Crooy, and W.M.P. van der Aalst

Department of Mathematics and Computer Science
Technische Universiteit Eindhoven
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
{b.f.v.dongen,w.m.p.v.d.aalst}@tue.nl, r.a.crooy@student.tue.nl

Abstract. A typical question for people dealing with administrative processes is: "When will my case be finished?". In this paper, we show how this question can be answered, using historic information in the form of event logs of the systems supporting these administrative processes. Many information systems record information about activities performed for past cases in logs. Hence, to provide insights into the remaining cycle time of a case, the current case can be compared to all past ones.
The most trivial way of estimating the remaining cycle time of a case is by looking at the average cycle time and deducting the already past time of the case under consideration. However, in this paper we show how to compute the remaining cycle time using non-parametric regression on the data recorded in event logs. An experiment is presented that demonstrates that our techniques perform well on logs taken from practice.

1 Introduction

An often heard complaint about administrative processes is that no insights are given in the time needed to handle a case. A customer filing a claim with an insurance company will probably hear in the beginning that the claim will be handled in approximately 4 to 6 weeks. When s/he calls to ask about the status after a couple of weeks, the answer is usually still that the handling will take 4 to 6 weeks in total, i.e. the *average cycle time* is 5 weeks.

The estimate given to the customer over the phone is not just an arbitrary number. Instead, this is usually the average cycle time of a case, combined with a certain margin of error. Obviously, at the time when a new claim (or case in a more general setting) is entered into the system of the insurance company, the best estimate of the remaining cycle time is indeed the average cycle time. However, as soon as the case has been entered into the system, it is annotated with all kinds of information that might influence the remaining cycle time. For example, claims filed by customers filing claims more often, are more likely to be checked for fraud, whereas claims that do not exceed a certain amount are never checked for fraud. Since such a check costs time, knowing whether it will or will not be performed obviously has an influence on the cycle time.

As a case progresses in the process, i.e. as more and more activities are performed, the amount of information relating to the remaining cycle time of the

case is increasing. If, for example, fraud is expected, a case might be deferred to a different part of the organization, which significantly delays the handling. Such delays heavily influence the cycle time of a single case, but they also influence the average cycle time of all cases. Therefore, especially for cases in the middle of their handling, the quality of the average cycle time as an estimator is poor.

Rules such as the one stating that claims under a certain amount are never checked for fraud, are not likely to be made public. People working with a system handling a case generally do not have to know that such rules exist. Hence these people may not be able to give a better estimate of the remaining cycle time than the average cycle time.

Fortunately, information systems used in the handling of large administrative processes, store all kinds of information related to current and past cases in event logs. These event logs are typically annotated with information relevant to the remaining cycle time of a case.

In this paper, we focus on the issue of remaining cycle time. We consider event logs as a basis on which we predict, what the remaining cycle time of a specific partial case is, e.g. we accurately answer the question of the customer about the time needed to handle his/her claim. We use non-parametric regression [6] as opposed to other methods for prediction, mainly because non-parametric regression is most suitable in situations with little or no precedents are available.

The paper is organized as follows. First, in Section 2, we introduce some notations and we formally define the regression techniques used. In Section 3, we present five different predictors for the remaining cycle time and in Section 4 we present a case study where these predictors were put to the test on a real-life dataset. We conclude the paper with a section on the implementation (Section 5) and some conclusions.

2 Preliminaries

In this section, we introduce some basic concepts needed for prediction. We introduce logs, as well as the ideas behind non-parametric regression.

Let S be a set. The powerset of S is denoted by $\mathcal{P}(S) = \{S' | S' \subseteq S\}$. A *bag (multiset)* m over S is a function $S \to \mathbb{N}$, where $\mathbb{N} = \{0, 1, 2, \ldots\}$ denotes the set of natural numbers. The set of all bags over S is denoted by $\mathbb{N}^S$. We identify a bag with all elements occurring only once with the set containing these elements, and vice versa. We use $+$ and $-$ for the sum and difference of two bags, and $=, <, >, \leq, \geq$ for the comparison of two bags, which are defined in a standard way. We use $\emptyset$ for the empty bag, and $\in$ for the element inclusion. We write e.g. $m = [p^2, q]$ for a bag m with $m(p) = 2$, $m(q) = 1$ and $m(x) = 0$, for all $x \notin \{p, q\}$. We use the standard notation $|m|$ and $|S|$ to denote the number of elements in bags and sets.

A *sequence* over S of length n is a function $\sigma \colon \{0, \ldots, n-1\} \to S$. If $\sigma(0) = a_0, \ldots, \sigma(n-1) = a_{n-1}$, we write $\sigma = \langle a_0, \ldots, a_{n-1} \rangle$, and σ_i for $\sigma(i)$. The length of a sequence is denoted by $|\sigma|$. The sequence of length 0 is called the empty sequence, and is denoted by $\langle\rangle$. The set of finite sequences over S is denoted

by S^*. Let $\upsilon, \tau \in S^*$ be two sequences. Concatenation, denoted by $\sigma = \upsilon \cdot \tau$ is defined as $\sigma : \{0, \ldots, |\upsilon| + |\tau| - 1\} \rightarrow S$, such that for $0 \leq i < |\upsilon|$, $\sigma(i) = \upsilon(i)$, and for $|\upsilon| \leq i < |\sigma|$, $\sigma(i) = \tau(i - |\upsilon|)$.

Furthermore, we define the prefix from index i to j on sequences by $\tau' = \downarrow_{i,j} (\tau)$, such that if $i \geq j$, then $\tau' = \langle\rangle$, otherwise $\tau' = \langle \tau_i, \ldots, \tau_{j-1} \rangle$, i.e. for all sequences τ holds that $\downarrow_{0,|\tau|} (\tau) = \tau$.

We use $\vec{\mathcal{P}}(x)$ to denote column vectors and for a sequence $\sigma \in S^*$, the *Parikh vector* $\vec{\sigma}$: $S \rightarrow \mathbb{N}$ defines the number of occurrences of each element of S in the sequence, i.e. $\vec{\sigma}(s) = |\{i | 0 \leq i < |\sigma| \wedge \sigma(i) = s\}|$, for all $s \in S$.

2.1 Logs

Information systems typically log all kinds of events. Unfortunately, most systems use their own specific format. Therefore, we formalize the concept of a log. The basic assumption is that the log contains information about *activities* executed for specific *cases*, as well as their durations. Extensive practical experiences in the context of the process mining framework ProM [1] show that this assumption is valid in many applications [2].

Definition 2.1. (Case, Log) Let A be a set of activities. $\sigma \in A^*$ is a *case*, consisting of activities. A log W over A is defined as a bag of cases, i.e. $W \subseteq \mathbb{N}^{A^*}$.

Definition 2.2. (Sequence start, completion, duration) Let A be a set of activities and $\sigma \in A^*$ a sequence of activities. We define $\tau_s(\sigma) \in \mathbb{R}^+$ and $\tau_c(\sigma) \in \mathbb{R}^+$ to represent the start and completion times of the sequence σ. By definition, we say that $\tau_s(\langle\rangle) = \tau_c(\langle\rangle) = 0$ and we denote the duration of a sequence by $\delta(\sigma) = \tau_c(\sigma) - \tau_s(\sigma)$.

Note that τ_s, τ_c, and δ are *not* functions, as similar sequences might have different times attached to them. However, they are total, i.e. they do provide start, completion times and durations for all sequences.

Finally, if $W \subseteq A^*$ is a log, then we assume that all $\sigma \in W$ and $0 \leq i \leq |\sigma|$ holds that $\tau_{s|c}(\downarrow_{0,i} (\sigma)) \leq \tau_{s|c}(\sigma)$, i.e. the activities within each case are ordered in time.

Besides the minimal information of activity/duration pairs, logs often carry case-related information, such as the amount of money involved in a claim, or the data entered in an application form. As the nature of this information is not known up front, we leave that abstract for now and we define the case data as a map of key/value pairs.

Definition 2.3. (Sequence data) Let A be a set of activities, $\sigma \in A^*$ a sequence over A, K a set of attribute keys and V a set of attribute values. We denote sequence data by $\Delta(\sigma) : K \rightarrow V$, as a function from the keys in K to their corresponding value in V . The exact nature of these domains is left abstract for now. Again, Δ is not a function, but it is total.

2.2 Regression

Regression is a technique to fit a function to a set of measurements, i.e. to abstract from these measurements. Basically, there are two types of regression, namely *parametric*, where the function is assumed to be of a certain form (e.g. linear, exponential, quadratic, etc.) and *non-parametric*, where no assumptions are made about the function that should fit the measurements [5, 6].

In this paper we use non-parametric regression because, in order to predict cycle times in any unspecified business process, we cannot assume the cycle time to have a specific form or distribution, which is needed for parametric regression. The non-parametric approach only assumes there is some relationship between the predictor variables and the target variable, the form of this relationship need not be specified. A method called smoothing or "local averaging" is used in non-parametric regression to make estimations based the observed data, without a parameterized model.

At the basis of non-parametric regression lies a list of measurements $m = \langle m_0, m_1, \ldots, m_n \rangle$, such that each measurement $m_i = (\vec{x}_i, y_i) \in (X \times \mathbb{R})$ consists of a vector of k so-called predictor variables $\vec{x}_i \in X$ and a target variable ($y_i \in \mathbb{R}$). The domain $X = X_0 \times X_1 \times \ldots \times X_{k-1}$ are kept abstract for now.

The goal of regression is that, based on the measurements stored in m, the value of any new vector of predictor variables $\vec{x'}$ is estimated by a function $\gamma : X \to \mathbb{R}$. This function is such that it estimates the corresponding target value y', i.e. $y' \approx \gamma(\vec{x'})$. The estimate is such that the values of the target variables of measurements in m closest to the new vector $\vec{x'}$ have more influence than those measurements farther from $\vec{x'}$, i.e. $\gamma(\vec{x'})$ is interpolated from measurements in the larger vicinity of $\vec{x'}$.

The way the function γ computes the estimated target value is by taking the weighted average of the target values of the measurements, where the weight of the k components of the vector $\vec{x_i}$ is determined by a parameterized so-called *kernel function* $\phi : (X \times X \times \mathbb{R}^k) \to \mathbb{R}^+$, assigning an inverted weight to each of the k components representing the distance between $\vec{x'}$ and $\vec{x_i}$. The relative importance of each component of the vectors $\vec{x'}$ and $\vec{x_i}$ is denoted by the *bandwidth variable* $\vec{\lambda} \in \mathbb{R}^k$. As a result, the function γ, denoted by $\gamma_{\vec{\lambda}}$ to show the dependency on the bandwidth, looks as follows:

$$\gamma_{\vec{\lambda}}(\vec{x'}) = \frac{\sum_{i=0}^{n} \phi(\vec{x}_i, \vec{x'}, \vec{\lambda}).\, y_i}{\sum_{i=0}^{n} \phi(\vec{x}_i, \vec{x'}, \vec{\lambda})}, \text{for } m = \langle (\vec{x}_0, y_0), (\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n) \rangle \qquad (1)$$

In Section 3, we elaborate on how to select the right kernel function ϕ, taking into account the (so-far abstract) domains of each of the predictor variables. However, the process of finding the optimal values for the bandwidth variables, is similar for any particular kernel function.

2.3 Bandwidth Variable Optimization

In order to find the optimum bandwidth, we first define what optimality is. The goal of regression is to estimate $y' = \gamma_{\vec{\lambda}}(\vec{x'}) + \epsilon$ such that ϵ (the error) is minimal.

As this error depends on the bandwidth variables, the value of each component of the bandwidth variable $\vec{\lambda}$ is computed by minimizing $n^{-1}\sum_{i=0}^{n}(\gamma_{\vec{\lambda}}(\vec{x}_i) - y_i)^2$ using only the measurements m. To minimize bias standard cross-validation techniques are used [6].

In this paper we do not elaborate further on how to find the optimal bandwidth. Instead, we refer to [6] for various techniques to find an the optimum bandwidth efficiently and to [8, 10] for details about the implementation used in our case-study.

So far, we imposed the restriction that the target variable has a continuous domain. In the remainder of this paper, we use the remaining cycle time of a case as the target variable, which indeed has a continuous domain.

3 Cycle Time Prediction

Before presenting the results of our case study in Section 4, we first introduce several methods for estimating the remaining cycle time. First, we show a naive approach using only the average cycle time over a log. Then, we show three types of regression-based techniques, using the occurrences of activities, the durations of activities and the case data to base the prediction on. We conclude this section by showing how to combine the different regression-based techniques. The regression kernel functions presented in this section were inspired by [9], which proposes a method for non-parametric regression with both continuous and (un)ordered categorical variables.

3.1 Average Cycle Time Estimator

As mentioned in the introduction, the average cycle time is often used as an estimator for the remaining cycle time. Using the start and completion of a sequence, it is trivial to define the average cycle time of a log, which is just the sum of the durations of all cases divided by the number of cases.

Definition 3.1. (Average Cycle Time) Let W be a log. The average cycle time $\mathcal{A}_W$ of W is defined as $\mathcal{A}_W = \frac{\sum_{\sigma \in W} \delta(\sigma)}{|W|}$, where $|W|$ denotes the number of cases in W.

The remaining duration of a partial case under consideration is deducted from the average cycle time of a log to predict the remaining cycle time of the case. Since the result might be negative, the estimate is rounded to 0.

Definition 3.2. (Average Cycle Time Predictor) Let A be a set of activities and W be a log over A. Let $\mathcal{A}_W$ be the average cycle time over W. Let $\sigma' \in A^*$ be a partial case (i.e. a case that has not been completed yet). The average cycle time predictor of this partial case is defined as $\rho^{AVG}(\sigma') = \max(0, \mathcal{A}_W - \delta(\sigma'))$.

3.2 Activity Occurrence Estimator

The first regression-based estimator we present in this paper considers only the frequencies of activities within each case. As we have shown in Section 2.2, regression is based on measurements, which we have to define for logs.

Definition 3.3. (Activity Occurrence Measurement) Let A be a set of activities, let W be a log over A and let $\sigma \in W$ be a case. We define the list of case measurements as $\mathcal{M}_{\sigma}^{AO} = \langle m_1, \ldots, m_{|\sigma|} \rangle$, where for all $0 < i \leq |\sigma|$ holds that $m_i = (\vec{\mathcal{P}}(\downarrow_{0,i}(\sigma)), \tau_c(\sigma) - \tau_c \downarrow_{0,i}(\sigma))$.

We define the list of activity occurrence measurements $\mathcal{M}_{W}^{AO}$ over W as the concatenation of the case measurements.

In words, the list of activity occurrence measurements is such that for each (non-empty) prefix of a case, one measurement is taken, consisting of the number of occurrences of the elements of A in that prefix (the predictor variables $\vec{\mathcal{P}}(\sigma)$) and the remaining cycle time of that prefix (the target variable $\tau_c(\sigma) - \tau_c \downarrow_{0,i} (\sigma)$), which is defined as the difference between the latest time in the prefix and the latest time in the entire case. Note that, since the activities within each case are ordered in time, the remaining cycle time is a continuous variable greater or equal to 0, i.e. $\tau_c(\sigma) - \tau_c \downarrow_{0,i} (\sigma) \in \mathbb{R}^+$.

For a log W, the total number of measurements equals the sum of the lengths of all cases in the log.

So far, the domains of the measurement variables (denoted by X_i in Section 2.2) were kept abstract. For the activity occurrence measurement defined in Definition 3.3, we can make these domains concrete, since each variable represents the number of times an activity occurred, we know that all domains equal $\mathbb{N}$, which implies that we can define a concrete kernel function for these variables.

For this purpose, we use a variation of the Aitchison and Aitken's [3] kernel which was defined by Jeff Racine and Qi Li [9]. This function is parameterized by the bandwidth parameters $\lambda \in [0..1] \subset \mathbb{R}^+$. If a bandwidth parameter is 0 for a certain activity, then this activity has maximum influence on the remaining cycle time, whereas a value of 1 implies minimal influence.

Definition 3.4. (Activity Occurrence Kernel Function) Let A be a set of activities. We define the activity occurrence kernel function $\phi^{AO} : \mathbb{N}^{|A|} \times A^* \times [0..1]^{|A|} \to [0..1]$, such that

$$\phi^{AO}(\vec{x}, \sigma, \vec{\lambda}^{AO}) = \prod_{a \in A} (\vec{\lambda}^{AO}(a))^{|\vec{x}(a) - \vec{\mathcal{P}}(\sigma)(a)|} \text{ with } 0^0 := 1 \tag{2}$$

Using the activity occurrence kernel function, we now complete our activity occurrence predictor by substituting it in Equation 1.

Definition 3.5. (Activity Occurrence Predictor) Let A be a set of activities, let W be a log over A and let $\mathcal{M}_{W}^{AO}$ be the activity occurrence measurements

over W. Furthermore, let $\sigma \in A^*$ be a partial case. The expected remaining cycle time, given the bandwidth parameters $\vec{\lambda}^{AO}$, of this case is estimated by $\rho^{AO} : A^* \to \mathbb{R}$, as:

$$\rho^{AO}(\sigma) = \frac{\sum_{(\vec{x},y)\in\mathcal{M}_W^{AO}} \phi^{AO}(\vec{x},\sigma,\vec{\lambda}^{AO}).\, y}{\sum_{(\vec{x},y)\in\mathcal{M}_W^{AO}} \phi^{AO}(\vec{x},\sigma,\vec{\lambda}^{AO})} \tag{3}$$

With Definition 3.5, we have defined a prediction function that estimates the remaining cycle time of any given sequence, based on the occurrences of activities recorded in cases in the log. The quality of this prediction function depends on the bandwidth variable $\vec{\lambda}$, which is determined using the procedure described in Section 2.3.

3.3 Activity Duration Estimator

The second regression-based estimator we present in this paper considers the duration of each of the activities within cases. As we have shown in Section 2.2, regression is based on measurements, which we therefore define for our logs.

Definition 3.6. (Activity Duration Measurement) Let A be a set of activities, let W be a log over A and let $\sigma \in W$ be a case. We define the list of case measurements as $\mathcal{M}_\sigma^{AD} = \langle m_1, \ldots, m_{|\sigma|}\rangle$, where for all $0 < i \leq |\sigma|$ holds that $m_i = (\vec{x_i}, \tau_c(\sigma) - \tau_c(\downarrow_{0,i}(\sigma)))$, with $\vec{x_i}$ a vector such that for all $a \in A$ holds that

$$\vec{x}_i(a) = \frac{\sum_{\substack{0<j<i \\ \sigma_{j-1}=a}} \delta(\downarrow_{j-1,j}(\sigma))}{\vec{\mathcal{P}}(\downarrow_{0,i}(\sigma))(a)} \text{ with } \frac{0}{0} := 0. \tag{4}$$

We define the list of activity duration measurements $\mathcal{M}_W^{AD}$ over W as the concatenation of the case measurements.

Definition 3.6 defines the activity duration measurements, such that for each (non-empty) prefix of a case, the measurement consists of (i) the average duration of each activity $a \in A$ within this case (the sum of the durations of each occurrence of a divided by the number of occurrences) and (ii) the remaining cycle time of this case (the target variable).

In contrast to the activity occurrence measurements presented in Section 3.2, the domains of the measurement variables (denoted by X_i in Section 2.2) are not the natural numbers, but real numbers, i.e. each measurement is a vector of semi-positive real numbers, which implies that we can define a concrete kernel function for these variables.

For this purpose, we use a Gaussian kernel function [6], which is parameterized by the bandwidth parameters $\lambda \in \mathbb{R}^+$ for all activities[1]. A value close to 0 of the bandwidth parameter means that the influence of this parameter to the

[1] For continuous variables, the bandwidth is often denoted by h instead of λ to show the difference in domains. However, for consistency, we use λ.

remaining cycle time is maximal. A value of ∞ represents the case where the duration of an activity has no influence on the remaining cycle time.

We choose to use the Gaussian kernel, as it is the only kernel with infinite support, other kernels will assign a weight of 0 to cases if their difference is greater than a certain number, the Gaussian however will assign a calculated weight to all cases although it might go to 0. In this way a prediction for a partial case that has few to no precedents will be based on larger set.

Definition 3.7. (Activity Duration Kernel Function) Let A be a set of activities. We define the activity duration kernel function $\phi^{AD} : \mathbb{R}^{|A|} \times A^* \times \mathbb{R}^{|A|} \rightarrow \mathbb{R}^+$, such that

$$\phi^{AD}(\vec{x}, \sigma, \vec{\lambda}^{AD}) = \prod_{a \in A} \frac{\kappa\left(\frac{\vec{x}(a) - \vec{\mathcal{P}}(\sigma)(a)}{\vec{\lambda}^{AD}(a)}\right)}{\vec{\lambda}^{AD}(a)}, \text{ where } \kappa(u) = \frac{e^{-u^2/2}}{\sqrt{2\pi}} \tag{5}$$

Using the activity duration kernel function, we now complete our activity duration predictor by substituting it in Equation 1.

Definition 3.8. (Activity Duration Predictor) Let A be a set of activities, let W be a log over A and let $\mathcal{M}_W^{AD}$ be the activity duration measurements over W. Furthermore, let $\sigma' \in A^*$ be a partial case. The expected remaining cycle time, given the bandwidth parameters $\vec{\lambda}^{AD}$, of this case is estimated by $\rho^{AD} : A^* \rightarrow \mathbb{R}$, as:

$$\rho^{AD}(\sigma) = \frac{\sum_{(\vec{x}, y) \in \mathcal{M}_W^{AD}} \phi^{AD}(\vec{x}, \sigma, \vec{\lambda}^{AD}).\, y}{\sum_{(\vec{x}, y) \in \mathcal{M}_W^{AD}} \phi^{AD}(\vec{x}, \sigma, \vec{\lambda}^{AD})} \tag{6}$$

3.4 Case Attribute Estimator

The activity occurrence estimator of Section 3.2 considers the activity occurrences as measurements, which are variables from an ordered, ordinal domain ($\mathbb{N}$). The duration estimator however uses activity durations as measurements which are positive continuous variables ($\mathbb{R}^+$). In this subsection, we present the third type of variables, namely unordered ordinal variables.

Recall that Definition 2.3 presented a way to define arbitrary data attributes on each case in a log. As we do not assume any knowledge about this data in the log, we can only consider these data attributes to be unordered ordinal variables. However, they might still be of influence for the prediction of the remaining cycle time.

Definition 3.9. (Case Data Measurement) Let A be a set of activities, let W be a log over A, let K be a set of attribute keys and let $\sigma \in W$ be a case. We define the list of case data measurements as $\mathcal{M}_\sigma^{CD} = \langle m_1, \ldots, m_{|\sigma|} \rangle$, where for all $0 < i \leq |\sigma|$ holds that $m_i = (\vec{x_i}, \tau_c(\sigma) - \tau_c(\sigma_p))$, with x_i a vector, such that for all $k \in K$ holds that $x_i(k) = \Delta(\downarrow_{0,i} (\sigma))(k)$.

We define the list of activity occurrence measurements $\mathcal{M}_W^{CD}$ over W as the concatenation of the case measurements.

In other words, for each non-empty prefix of a case, a measurement is taken consisting of a vector representing the value of each attribute key for that prefix. Note that this allows for case data to change during execution, i.e. attributed might change value after certain activities have been performed.

If unordered ordinal variables are used for measuring, not much can be said about the distance between two measurements. In fact, only when two variables take the same value, they can be assumed to be close. Therefore, the kernel function is a modified version of Definition 3.4.

Definition 3.10. (Case Data Kernel Function) Let A be a set of activities, K a set of attribute keys and V the set of attribute values. We define the case data kernel function $\phi^{CD} : V^{|K|} \times A^* \times [0..1]^{|K|} \to [0..1]$, such that

$$\phi^{CD}(\vec{x}, \sigma, \vec{\lambda}^{CD}) = \prod_{k \in K} \begin{cases} 1, & \text{if } \vec{x}(k) = \Delta(\sigma)(k) \\ \vec{\lambda}(k), & \text{if } \vec{x}(k) \neq \Delta(\sigma)(k) \end{cases} \tag{7}$$

When comparing Definition 3.10 to Definition 3.4, it becomes clear that the only difference is in the "distance" of two variables, i.e. when assuming that for all unordered nominal variables (case data) the distance between different values equals 1 (i.e. $|\vec{x}(a) - \vec{x_p}(b)| := 1$), then these definitions are the same.

Finally, for case data, we define the predictor.

Definition 3.11. (Case Data Predictor) Let A be a set of activities, let W be a log over A, let K be a set of attribute keys, V the set of attribute values and let $\mathcal{M}_W^{CD}$ be the case data measurements over W. Furthermore, let $\sigma' \in A^*$ be a partial case. The expected remaining cycle time, given the bandwidth parameters $\vec{\lambda}^{CD}$, of this case is estimated by $\rho^{CD} : A^* \to \mathbb{R}$, as:

$$\rho^{CD}(\sigma) = \frac{\sum_{(\vec{x}, y) \in \mathcal{M}_W^{CD}} \phi^{CD}(\vec{x}, \sigma, \vec{\lambda}^{CD}).\, y}{\sum_{(\vec{x}, y) \in \mathcal{M}_W^{CD}} \phi^{CD}(\vec{x}, \sigma, \vec{\lambda}^{CD})} \tag{8}$$

3.5 Combining Regression Estimators

In the previous subsection, we have presented three regression-based estimators. The first estimator is based on the occurrences of activities within cases, which is an ordered ordinal variable. The second is based on activity durations, a continuous variable and the last on case data, which we considered to be unordered ordinal variables. All three estimators had the same structure, i.e. they consisted of a set of measurements, a kernel function and a predictor. In this subsection we show how to do regression on a mix of different variable types.

In definitions 3.3, 3.6 and 3.9, we presented the measurements for the different types of variables. These measurements consisted of two parts, namely the measurement variables and the target variable. This target variable is defined the same for all measurements. Furthermore, all lists of measurements have the same size, i.e. one measurement is taken per non-empty prefix of a case in the

log. This allows us to easily combine these measurements into one vector containing the activity occurrences, the activity duration and the case data for each prefix.

For the combined measurements, the kernel function used in the regression is simply the product of all individual kernel functions, applies to the relevant measurement variables, i.e., the kernel function of Definition 3.4 is multiplied with the kernel functions of definitions 3.7 and 3.10, where each of these functions is applied to the relevant part of the vector of measurement variables.

4 Case Study

We tested our prediction approach on a dataset taken from real-life. In this section, we present and discuss the results.

4.1 Case Description

For the verification of our approach, we used a dataset called "bezwaar WOZ" from a Dutch municipality [7,11]. The process described in the log is the process of handling objections filed against real estate taxes.

From the log that originally contains 1982 cases, we only kept those cases that were fully contained in the measurement period, i.e. both their first and last activity were performed in the measurement period. Furthermore, after consulting the process owner, we removed those activities not relating to the main procedure. This resulted in a log containing 706 cases, which were handled by the municipality between February 28^{th} 2005 and November 8^{th} 2005 (a period of 252 days). In total, 9218 events were recorded, relating to the start and completion of 12 activities. Note that the start events were only used to obtain the durations of each activity. The complete events were used in the measurements. Furthermore, all cases and all events were annotated with data attributes, which we all used in the analysis.

4.2 Experiment Setup

We conducted experiments using five different estimators. The experiments were set up as 10-fold cross validation experiments, meaning that each time, the original log was split into 10 partitions. Then, 9 partitions were used as measurements and using these measurements, the optimal values of the bandwidth parameters were computed. For all cases in the remaining partition, estimates of the remaining cycle time were computed using the optimal bandwidth parameters. These estimates were computed after the occurrence of each activity, except the last. [2] By repeating this procedure 10 times, the remaining cycle time is estimated, exactly once after completing each activity in the log (except for the last activity in each case).

The five estimators we used were:

[2] The last activity is not considered, as the case is than finished and therefore the remaining cycle time is known.

1. The naive approach of Section 3.1, i.e. the average cycle time over the 9 measurement parts minus the already passed time maximized with 0,
2. The estimator of Section 3.2, where only activity occurrences were taken into account as measurements for the non-parametric regression,
3. The estimator of Section 3.3, where only activity durations were taken into account as measurements,
4. The estimator of Section 3.4, where only case attributes were taken into account as measurements,
5. The estimator of Section 3.5, taking into account attributes, as well as activity occurrences and durations,

For each of the regression experiments, optimal values for the bandwidth parameters were calculated using R [8], which uses an internal cross-validation method for finding the optimal values of the bandwidth parameters. Bandwidth selection and the computation of the predictions for each partial prefix, were done using the software package R, running on four dual quad-core 2.66GHz Intel Xeon CPUs with 16 GB of memory each. This setup allowed us to run the 10 experiments of the 10-fold cross validation in parallel, as each experiment uses a single thread. The longest experiment, using the activity durations as measurements took 5 hours.

4.3 Discussion

Figures 1 to 5 show the results of our analysis. Each figure contains 3 lines, representing the actual remaining cycle time (in read, with square markers), the estimated remaining cycle time (in blue, with triangular markers) and the mean square error of the estimate (in pink, using circular markers), which uses a different scale. For sake of readability, the last part of each graph is zoomed out and those points that did not fit on the scale are annotated with their exact values.

On the x-axis, the time in days since the beginning of each case is depicted, i.e. the longest case took 252 days to complete, which spans the whole measuring period. On average, cases completed in 175 days. To get the points depicted in each graph, measurements were averaged over 7 day periods, i.e. the average time at which activities are performed within the first week of starting a case is 2.35 days, whereas the average remaining cycle time of those events is 173 days, thus yielding the first point $(2.35, 173)$ in the actual remaining cycle time graph.

All figures 1 to 5 show the same graph for the actual remaining cycle time. Interestingly, after approximately 180 days, the actual remaining cycle time stays constant at an average of 5 days.

Average Estimator: Figure 1 shows the performance of the average estimator. As expected, this estimate is an almost straight line from an estimate of 175 days at time 0 to an estimate of 0 at 175 days and more. The deviations from the straight line are caused by the nature of the 10-fold cross validation experiment, i.e. the average cycle time over 9 partitions deviates from the

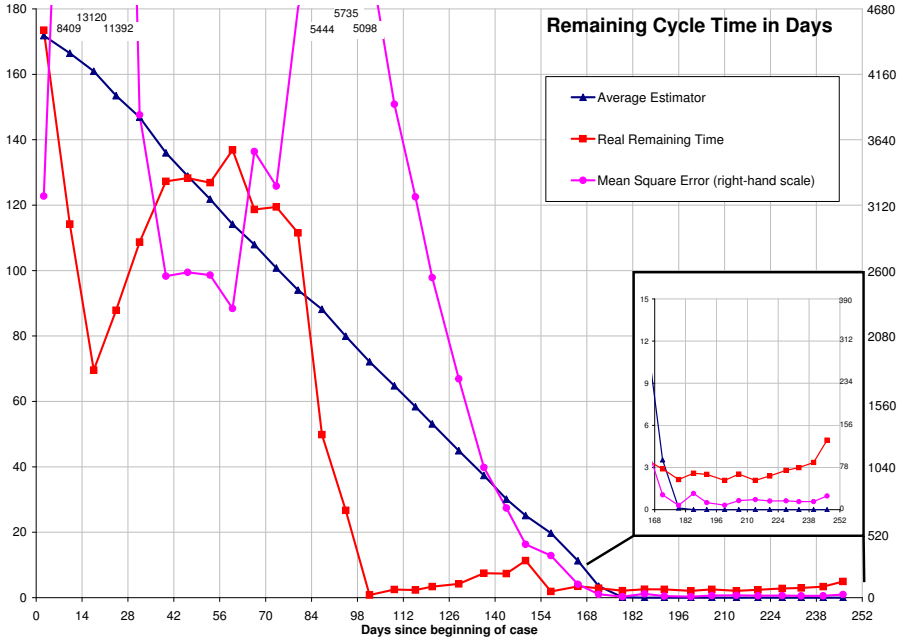

Fig. 1. Estimated values using average estimator.

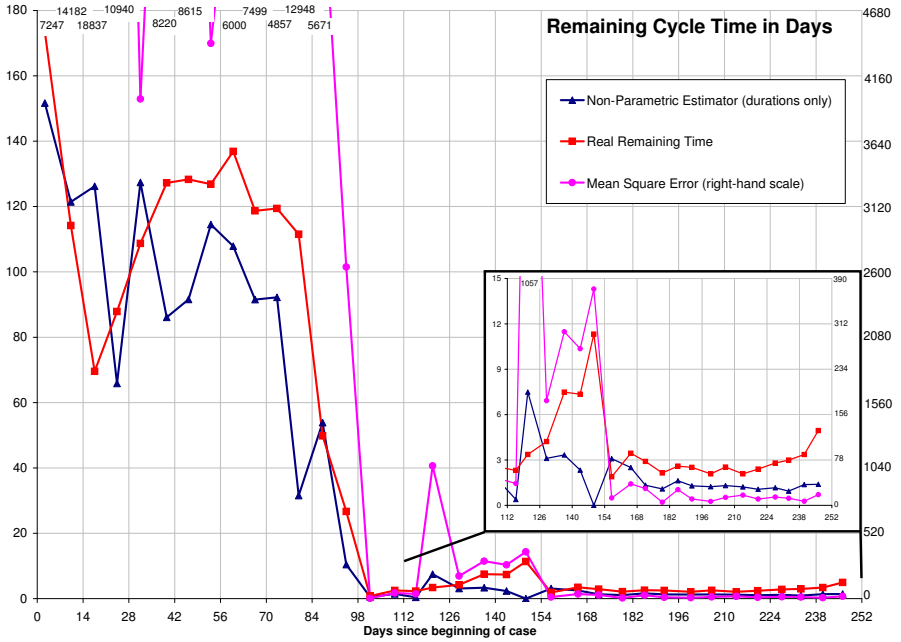

Fig. 2. Estimated values using only activity duration.

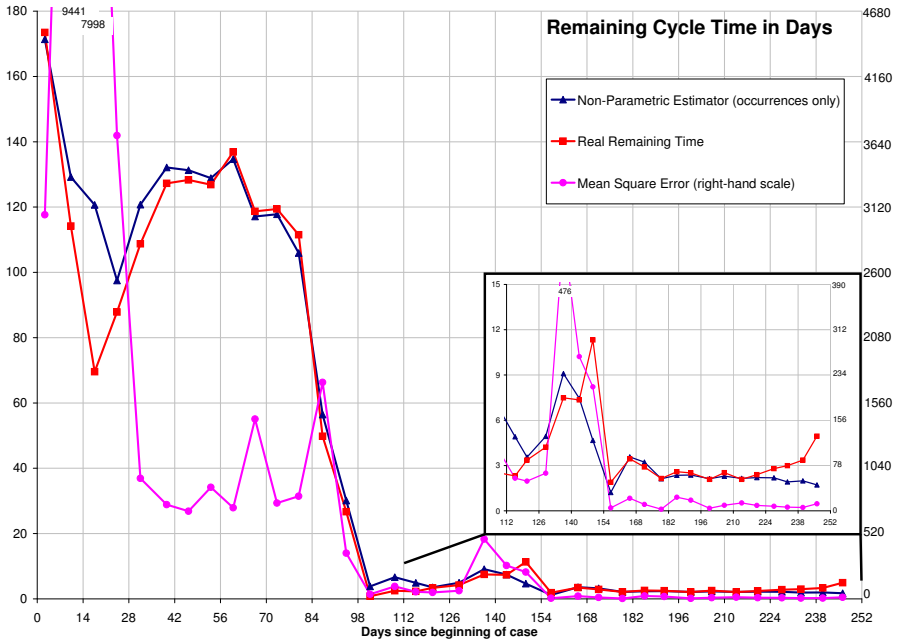

Fig. 3. Estimated values using only activity occurrences.

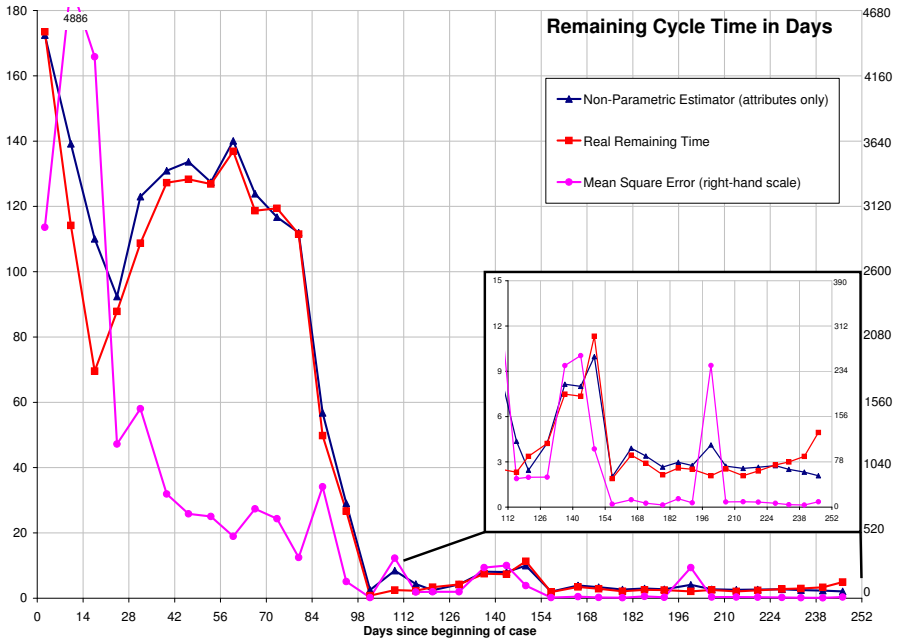

Fig. 4. Estimated values using only attributes.

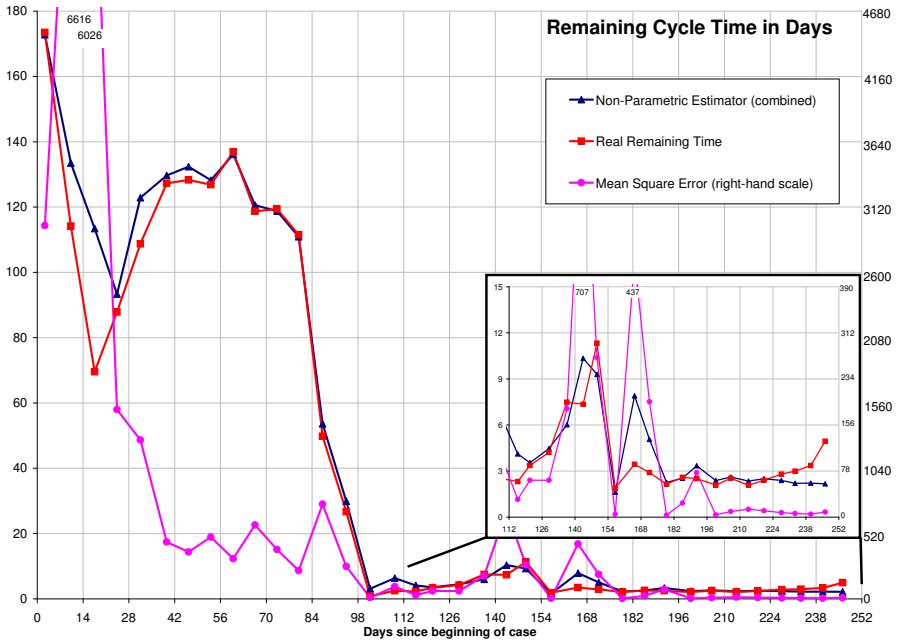

Fig. 5. Estimated values using durations, occurrences and attributes.

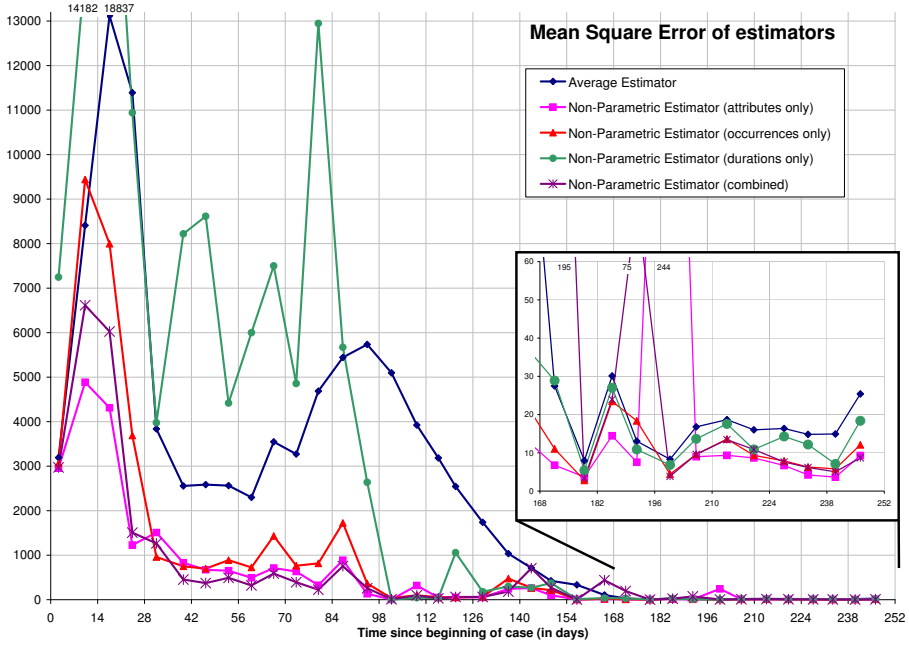

Fig. 6. Mean Square Errors of all estimators.

average cycle time over the log. Note that the scale of the mean-square error, shown on the right-hand side, is different from the scale on the left-hand side. The MSEs of all estimators are collected in Figure 6 and discussed in more detail in Subsection 4.4.

Duration Estimator: Figure 2 shows the result of a 10-fold cross validation experiment using only activity durations as measurements. The estimated remaining cycle time follows the actual remaining cycle time more closely than in Figure 1. However, the MSE of this estimator is far bigger. From 98 days after the start of a case, the duration estimator performs slightly better than the average estimator, which can also be observed in Figure 6. Nonetheless, it seems that using the durations as measurements does not provide much insights into the remaining cycle time.

Occurrences Estimator: Figure 3 shows the results when using activity occurrences as measurements. Here, the estimator follows the actual remaining time very closely, with a low value for the MSE. This indicates that using the activity occurrences as measurements is a good idea when trying to accurately predict the remaining cycle time of a case.

Attribute Estimator: As shown in Figure 4, using only the attribute values provides an even better estimate. The blue line with triangles in Figure 4, showing the estimator base on attributed follows the actual remaining cycle time very closely, with an even lower MSE than the occurrences-based estimator, as shown in Figure 6.

The bandwidth values indicate that the attributes "new_queue", "id", and "priority" are the most influential attributes. The "priority" attribute is a boolean and the name suggests that this attribute indeed should have big influence on the cycle time of a case.

The attribute "new_queue" indicates the next activity to be performed for a case, therefore this attribute changes as time passes and provides information about the future which makes it a good attribute to base predictions on. A fourth attribute that is relatively important is "queue", which indicates the activity that was just completed. As the non-parametric regression uses all variables to compare and select the most relevant cases, the combination of "queue" and "new_queue" provide a good basis for selecting the most relevant cases from the measurements.

The attribute "id", indicates the case-identifier. As the cross-validation splits up the log, it ensures that the measurement of a case is never used to make a prediction for that same case. Therefore, the weight of this attribute is irrelevant to the prediction. This shows that it is very difficult to derive information from the values in the bandwidth.

Combined Estimator: Figure 5 shows the results of the prediction when using all available information, i.e. attribute values, activity occurrences and activity durations. As shown in Figure 6, the combined estimator outperforms all estimators except the attribute-based one. Especially in the beginning and end of a case, the combined estimator performs worse.

Table 1. Mean Square Error of all estimators.

Estimator	Mean Square Error (with 5% confidence interval)
Average Estimator	2687.47 ± 133.69
Durations Estimator	5134.12 ± 320.02
Occurrences Estimator	1754.61 ± 115.38
Attribute Estimator	1532.85 ± 103.25
Combined Estimator	1537.42 ± 106.44

4.4 Error Analysis

In Figure 6, the mean square errors of all four estimators are combined. The figure shows that overall, the durations-based estimator performs worst, whereas the attribute-base estimator performs best. This is backed up by Table 1, which shows the mean square error of all estimates made for each estimator (i.e. not averaged per week as in Figure 6). It shows what we already concluded from the figures, i.e. that the durations estimator performs worst, but that the occurrences, attribute and combined estimators outperform the average estimator.

The best estimates are provided by attribute-based estimator and the combined estimator, which is not surprising as the attributes contain data relating to the difficulty of the case. What is interesting however, is that the estimator based on the occurrences of activities performs so well. This indicates that insights can be gained into the remaining cycle time, *without having to consider privacy sensitive data*. Especially in administrative processes, this can be valuable.

5 Implementation

The approach presented in this paper was implemented using the process mining framework ProM [1]. Figure 7 shows a screenshot of ProM showing the opened log we used in our case study on the top-right, together with two plugins we developed, namely the "Prediction Miner" and the "Event Data Attribute Visualizer". ProM can be downloaded from www.processmining.org and the plugins mentioned in this paper are available in the nightly builds.

5.1 Prediction Miner

The "Prediction Miner", shown on the left-hand side of Figure 7, provides a simple interface to the user for the analysis presented in this paper. It can be used to connect to our machines running R [8] via TCP/IP [10], but it also allows users to run R locally. Furthermore, it allows users to select the kernel functions to be used for different types of variables, as well as to set parameters not mentioned in this paper.

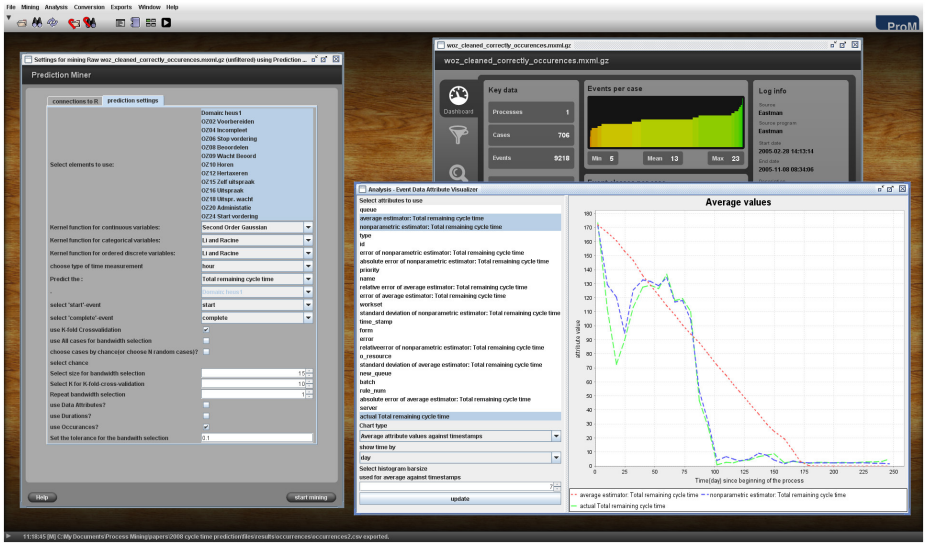

Fig. 7. ProM showing the opened log (top-right), the settings of the prediction miner (left) and the result as a graph (bottom-right).

5.2 Event Data Attribute Visualizer

The prediction miner annotates each "complete" event in the log with attributes relating to the remaining cycle time. In fact, it always stores (i) the actual remaining cycle time for each event, (ii) the non-parametric estimate of the remaining cycle time and (iii) the average estimator of the remaining cycle time. These attributes can be visualized using the "Event Data Attribute Visualizer", which in Figure 7 shows these estimators on the bottom-right. Also settings are provided for selecting the histogram size, in our case 7 days, i.e. the points in the graph are averaged over 7 day periods.

6 Conclusion and Future Work

In this paper, we presented a regression-based method for predicting the remaining cycle time of a case in a process. As input, we used an event log of the process under consideration, where we explicitly used information about the durations of all activities, the occurrence of all activities and any other case-related data. Using an example of a real-life process taken from practice, we have shown that our approach outperforms the naive approach of average cycle time minus the already spent time.

As a regression technique, we used non-parametric regression, where we assumed that activity durations are continuous variables, activity occurrences are

ordered ordinal variables and that all other case-data variables are unordered ordinal variables.

Although our results show that the predictions made under these assumptions are accurate, we expect that improvements can be made when case-data variables are not assumed to be unordered ordinal, but also to be ordered or even continuous. However, deciding about the type of variable for each data attribute is a human job, which can typically only be performed by the process owner, since it requires insights into the process at hand and the semantics of the data attributes.

To gain a deeper understanding of the situations in which the regression-based predictions perform well, we are currently conducting simulation experiments. For more information on these experiments and their results, we refer to [4].

The approach presented in this paper has been implemented in the process mining framework ProM and is available via www.processmining.org.

References

[1] W.M.P. van der Aalst, B.F. van Dongen, C.W. Günther, R.S. Mans, A.K. Alves de Medeiros, A. Rozinat, V. Rubin, M. Song, and H.M.W. Verbeek. ProM 4.0: Comprehensive Supports for Real Process Analysis. In *Application and Theory of Petri Nets 2007*, volume 4546 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 2007.

[2] W.M.P. van der Aalst, H.A. Reijers, A.J.M.M. Weijters, B.F. van Dongen, A.K. Alves de Medeiros, M. Song, and H.M.W. Verbeek. Business Process Mining: An Industrial Application. *Information Systems*, 32(5):713, 2007.

[3] J. AITCHISON and CGG AITKEN. Multivariate binary discrimination by the kernel method. *Biometrika*, 63(3):413–420, 1976.

[4] R.A. Crooy. Predictions in Information Systems, a process mining perspective. Master Thesis, Eindhoven University of Technology, 2008. To Appear in November 2008, via Digital Library of Eindhoven University of Technology.

[5] J. Dippon, P. Fritz, and M. Kohler. A statistical approach to case based reasoning, with application to breast cancer data. *Computational Statistics and Data Analysis*, 40(3):579–602, 2002.

[6] W. Hardle. *Applied Nonparametric Regression*. Cambridge University Press Cambridge [England, 1990.

[7] A.K.A. de Medeiros. *Genetic Process Mining*. PhD thesis, Eindhoven University of Technology, Eindhoven, 2006.

[8] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.

[9] J. Racine and Q. Li. Nonparametric estimation of regression functions with both categorical and continuous data. *Journal of Econometrics*, 119(1):99–130, 2004.

[10] S. Urbanek. Rserve - A Fast Way to Provide R Functionality to Applications. In K. Hornik, F. Leisch, and A. Zeileis, editors, *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*, 2003.

[11] Wil M. P. van der Aalst, Marlon Dumas, Chun Ouyang, Anne Rozinat, and Eric Verbeek. Conformance checking of service behavior. *ACM Trans. Interet Technol.*, 8(3):1–30, 2008.