INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

Internship

Summer internship
presented by

Florian Pellegrin
Engineer student from INSA Rennes
Department INFO
Year 2018 - 2019

Task estimation from activity log

Location of the internship
Okayama University

Internship supervisor
Zeynep Yücel

Correspondant pédagogique INSA Rennes
Jean-Louis Pazat

岡山大学
OKAYAMA UNIV.

Résumé

Ce rapport de stage présente mon stage de recherche dans la ville d'Okayama au Japon dans le cadre de ma 4e année d'informatique à l'INSA de Rennes en France. Le rapport a pour but de survoler les différents aspects du stage sur lesquels j'ai pu travailler, car il est difficile d'entrer dans les détails de tout ce qui a été fait étant donné qu'une grande partie du travail consistait à essayer différentes méthodes sans forcément les implémenter à la fin. Tout le côté technique de la recherche a été réalisé par moi-même tandis que la rédaction était largement supportée par ma superviseure. Le but du projet était d'être capable d'estimer la tâche qu'un utilisateur vient d'effectuer parmi une liste prédéfinie. La méthode que nous présentons se base sur une approche Bayesienne, l'idée est de réussir à prédire la tâche qui vient d'être effectuée en fonction de celle qui a la plus grande probabilité de dérouler en sachant les tâches précédemment effectuées. Tout le défi réside dans le fait de savoir quelles informations sont utiles pour estimer ces probabilités.

Abstract

This internship report presents my research internship in the city of Okayama in Japan as part of my 4th year of computer science at INSA Rennes, France. The purpose of the report is to cover only the different aspects of the internship I was able to work on. Indeed it is difficult to go into the details of everything that has been done since a lot of the work involved trying different methods without necessarily implementing them at the end. The technical side of the research was done by myself while the writing was largely supported by my supervisor. The purpose of the project was to be able to estimate the task that a user has just performed from a predefined list. The method we present is based on a Bayesian approach, the idea is to succeed in predicting the task that has just been carried out according to the one that has the highest probability of happening knowing the tasks previously performed. The challenge lies in knowing what information is useful for estimating these probabilities.

INSA

INSA Rennes
20 Avenue des Buttes de Coësmes
CS 70839 - 35708 Rennes Cedex 7
Tél. +33 (0) 2 23 23 82 00 - Fax +33 (0) 2 23 23 83 96
www.insa-rennes.fr

1 Introduction

This 3 months internship took place in a research laboratory at the University of Okayama in Japan and more specifically in the department of Computer Science of the Graduate School of Natural Science and Technology. The laboratory consisted of 15 research students including 3 trainees. During the whole internship, I was working alone on the technical part and was under the supervision of Assistant Professor Dr. Zeynep Yücel. The purpose of the internship was to study the activity log of computer users and understand the underlying task they perform. I was in charge of finding significant patterns depending on the user background and profile. I also had to develop probabilistic models to represent the above-mentioned tasks and use those to estimate the associated task from the active application and user's inputs. Aside from all of these, I had to read scientific papers, select and summarize the relevant papers, give a midterm and final presentation on my research and finally to co-author a paper. In the following sections, I will give you an insight into why we did this work and what I did. At first, we will try to understand the context of the internship and then we will go through the approach I have implemented under the directions of Dr. Yücel.

2 Internship background

In the software industry, projects are often elicited through sophisticated software development processes (SDP). An SDP requires complex management of resources, therefore it entails collaboration of a large team with members of different roles, responsibilities, backgrounds, etc. Obviously, it is common to have some bottlenecks along the course of such elaborate project development. Unfortunately, the varying profile of partners makes it very hard to pinpoint the obstructions or resolve how to handle them.

Thus, in order to provide a smooth interaction between teams or partners and ensure efficient time-use in software development within an SDP, it is desirable to detect such complications promptly. It is crucial to recognize a set of basic constituent tasks for defining the regular course of progress (i.e. free from difficulties). In this respect, we propose to detect these complications such that this information can be fed-back to managers. In doing that, we ideally would like to minimize the involvement of human factors. One approach to detect these complications would be to monitor the behavior of each member, identify their tasks, define regular progress and identify out-of-ordinary episodes. Therefore, we will focus on the task identification problem and present our approach which achieves task estimation from Bayesian probabilities utilizing the aforementioned constituents.

3 Previous work

3.1 Data set

At first, we will start by explaining what had been done before I started the internship. We had records, dating back to 2013, related to several programmers from a Chinese company thanks to a software recording the history of actions performed on a computer. This software

associates each action performed with several descriptors that are: start and end time of the action, application name (i.e. exe name), title of the window, number of keystrokes and number of left and right clicks.
A former senior undergraduate student carried out manual annotation by assigning a single $task$ to each action (i.e. line of the activity log). To that end, he evaluated the information contained in the descriptors (Table 5 in Annexes) and selected one task from the $set\ of\ potential\ tasks$, which is determined as Programming, Test, Documentation, Administration ,and Leisure, considering the authority and responsibility of the users.

3.2 Association rules

Association rules are basically a set of conditions defined based on relationships among variables of a data set. An association rule is denoted by an expression $(A \Rightarrow B)$, where A is referred to as the $antecedent$ and B is referred to as the $consequent$ of the rule.
For our particular set, the same former student defined a set of rules represented as association relationships between several descriptors and tasks. In particular, amongst the set of descriptors presented in Table 5, the student found two descriptors especially useful to be deployed as antecedents, i.e. application name and window title, which can be seen in Table 6

Some association rules have a single outcome (e.g. rule-1 in Table 6), whereas some other association rules yield 2 or 3 outcomes (e.g. rules 2 and 12 in Table 6). We call these outcomes as $candidate\ tasks$ (or simply candidates).

Obviously, these association rules are strictly related to the properties of the data set and the targeted subjects, they cannot be generalized to any set with their current definitions. Moreover, there are a number of issues associated with them that are likely to occur for any set of association rules or data set (more details are available in Appendix - Section B).

3.3 Base code

One year ago, my supervisor has worked on this research and produced a first version of our Bayesian approach in Scilab (i.e. free open source alternative to MATLAB). This was the base for my work and the only thing I had to use as a reference when implementing our method in Python.

4 Actual work

It is possible to separate the internship in several phases. The first few days consisted of reading scientific papers, summarizing them to my supervisor and discussing about their relevance as background information for our final paper. After that, I started transpiling the existing Scilab code into Python code. To do so, I created a script based on regular expressions which would help me transpile correctly more than 80% of each file. Even if it should not have taken long, I lost around two days of work because I could not compile the Scilab scripts on my personal computer and needed my supervisor to forward me the output of the requested functions. Since I had to wait sometime, I took this opportunity

Table 1: Distribution of the ground truth labels assigned by the coder. N denotes number of instances and P denotes the percentage of occurrence.

Task	Developer		Leader	
	N	P	N	P
Programming	328	0.171	0	0.0
Testing	**1580**	**0.822**	7	0.005
Administration	13	0.007	43	0.034
Leisure	0	0.0	232	0.181
Documentation	0	0.0	**1001**	**0.78**
Tot.	1921	1	1283	1

to implement some classifiers (i.e *kNN*, *SVM* and *RandomForest*) so that we can have a baseline to compare with the association rules and our method. By doing that, I realized that our data set was unbalanced due to the non-even representation of the tasks (as it is presented in Table 1).

In this regard, I proposed solving this problem by over-sampling everything but the majority class with the SMOTE[1] technique.
Eventually, this led us to the midterm presentation, which took more than one week to fully prepare (the slides for this 45min presentation are available here). This presentation allowed us to put into light what had to be perfected and that we should compare all the possible approaches more fairly.

From this point, we started doing research more deeply and eventually we have been able to propose the following method. After finishing this, we had two weeks to write the paper and organize the final presentation (the slides are available here)

5 Proposed method

This internship led us to propose a method, which is capable of estimating users' tasks from activity logs without any need for extensive supervision provided by experts or any future information. In doing that, we employ empirical distributions of descriptors and try to relate the descriptor values of the actions with the associated tasks in a probabilistic manner utilizing a Bayesian approach.

5.1 Fundamentals of Bayesian estimation

Let us denote an action performed at line n of the activity log with $\alpha[n]$ and the set of descriptors associated with it with $\Delta[n]$. Suppose that a particular task is represented with

[1]Synthetic Minority Over-sampling TEchnique

t, where the set of all possible tasks is denoted with T, $t \in T$. We compute the probability that action $\alpha[n]$ belongs to a task t, $\alpha[n] \sim t$, $\forall t$ in the following manner[2].

The probability that the user performs a task t given an observation set $\Delta[n]$ is represented with $P_n(t|\Delta)$ and computed in a Bayesian manner,

$$P_n(t|\Delta) = \frac{P_n(\Delta|t)P_n(t)}{P_n(\Delta)}. \tag{1}$$

Here, $P_n(t|\Delta)$ is the posterior probability of the task t after observing the evidence provided by Δ, whereas $P_n(t)$ is the prior probability before any evidence is taken into account. In addition, $P_n(\Delta|t)$ is the likelihood term, which expresses how likely it is to observe the particular values of Δ for a given task t. On the other hand, in Bayesian statistics the term $P_n(\Delta)$ in the denominator is considered to represent the probability of evidence.

In order to compute the posterior probability by Equation 1, we need to compute, in particular, the prior probabilities $P_n(t)$ and the likelihood terms $P_n(\Delta|t)$, but it is not necessary to explicitly compute the probability of evidence. Namely, since the denominator term takes the same value for each task $t \in T$, the products in the numerator

$$\widehat{P}_n(t|\Delta) = P_n(\Delta|t)P_n(t), \tag{2}$$

can simply be scaled such that they sum up to 1. In other words, we make use of the fact that the action $\alpha[n]$ definitely belongs to one of the tasks in the list of potential tasks T and omit computing the probability of evidence.

In what follows, we elaborate on several extensions, which have the potential of enhancing the performance of this Bayesian approach, particularly when applied on our data set. We also deal with some practical issues concerning our specific descriptors.

5.2 Hierarchical implementation of the Bayesian approach

The dramatic bias in task distribution presented in Table 1 indicates a challenge in estimation of the less frequently occurring tasks.

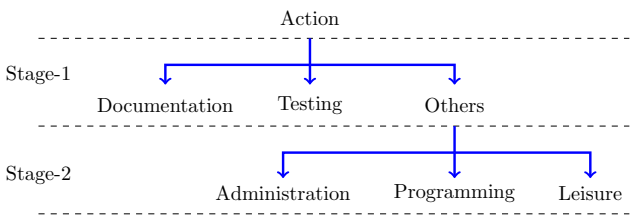

Figure 1: Hierarchical estimation of tasks.

In order to deal with our unbalanced data set, we propose using a hierarchical strategy in task estimation as presented in Figure 1. Namely, as a first stage in estimation, we determine

[2]Henceforth, we carry the index n of the set of descriptors d as well as the indices of its elements, to the subscript of the probability density function.

whether the action belongs to one of frequently occurring tasks or not. We consider Test and Documentation as frequently occurring tasks and put the remaining tasks (i.e. Programming, Administration and Leisure) under Others (see Table 1). If the action $\alpha[n]$ is determined to belong either to Test or to Documentation, estimation process is considered to be finalized. On the other hand, the process proceeds to second stage, in case $\alpha[n]$ is found to belong to Others (see Figure 1). At second stage we deal with actions, which are identified to be one of the infrequent tasks Stage-1, and determine whether it is Administration, Leisure or Programming.

5.3 Preprocessing prior to building empirical distributions

In addition to the unbalanced distribution of tasks, the concentration of descriptor values in narrow ranges as shown in Figure 2 point out to another challenge. In order to mitigate this issue, as well as to build a variable space with manageable size, we propose several preprocessing operations of descriptor values.

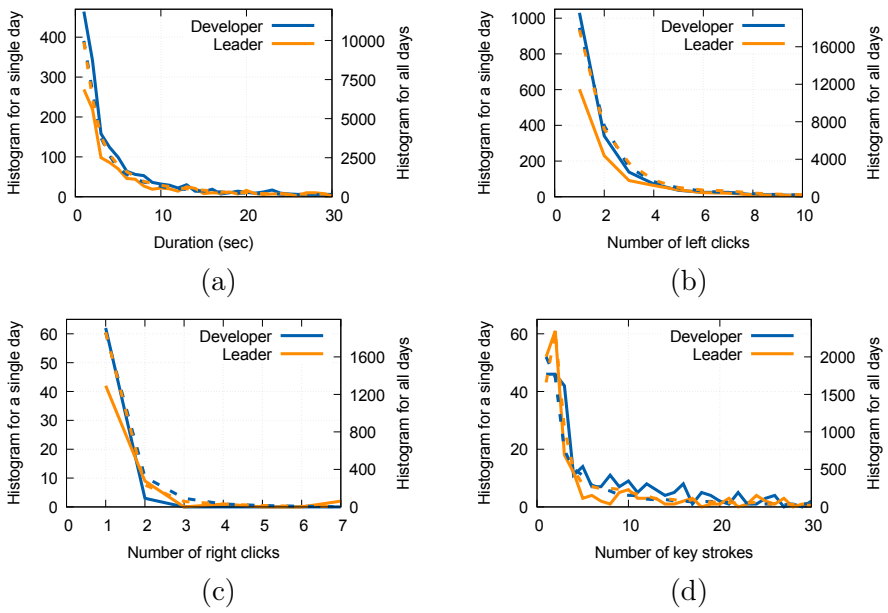

Figure 2: Histogram for (a) duration; and number of (b) left clicks, (c) right clicks, and (d) key strokes regarding the actions of the developer and leader. The solid curves relate actions on the particular day of annotation and the dashed ones reflect distributions over the entire monitoring duration.

Note that the descriptors can be considered as *variables* of the distribution functions. Therefore, we will henceforth use the terms descriptor and variable interchangeably.[3]

[3]However, note that it is not necessarily the case that every descriptor is a variable. Namely, according to the conditional dependence of descriptors, several descriptors can be combined in one variable.

The preprocessing of variable values refer to either quantization or categorization (or clustering) of values. Specifically, numerical variables (duration, number of left/right clicks and key strokes) are quantized, whereas nominal variables (i.e. applications and window titles) are categorized.

5.3.1 Preprocessing quantitative variables

Descriptors such as duration, number of left/right clicks and key strokes, are quantized so as to decrease the dimension of the variable space (see Appendix D for range of descriptor values). Clearly, duration is always larger than 0, whereas number of key strokes or left/right clicks attain 0 quite often. Therefore, for those descriptors we consider having a zero value and nonzero values, separately.

Moreover, we categorize the nonzero values by grouping them into c bins such that there are approximately same number of observations in each bin. To do so, we derive at first their empirical probability density functions (pdf) from relating histograms and then compute the cumulative density functions (cdf), which allow us to separate -almost- equitably the variables into different bins.

As an example, consider Figure 3 illustrating this procedure on duration descriptor of the developer. When the bin edges are chosen as illustrated on the x-axis of Figure 3 (and listed in column 2 of 2), the number of observations in each bin are found as in column 3 of Table 2.

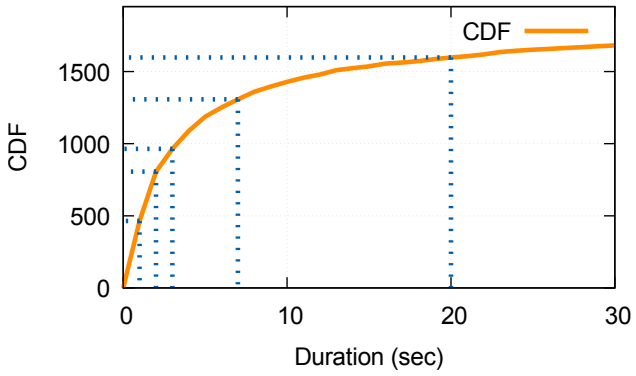

Figure 3: Quantization of task duration of the developer.

5.3.2 Preprocessing nominal variables

Since application names and window titles are nominal variables, which can potentially take a huge variety of values, we need to be careful about the size of those variable spaces.

Regarding applications, since the users utilize a small number of specific software tools (i.e. 20 unique applications) in comparison to the number of samples in the data set (see

Table 2: Details of quantization of task duration of the developer.

Bin number	Bin edges	Number of observations
1	(0,1]	464
2	(1,2]	342
3	(2,3]	158
4	(3,7]	343
5	(7,20]	290
6	(20,∞]	324
		1921

Table 1), the size of the relating variable space is considered to be manageable. Therefore, we consider all of those applications in our analysis.

In regard to window titles, since there is virtually an infinite number of possibilities, we only consider a set of 50 *key phrases*, which commonly occur in window titles. In the current analysis, the former student provided a list of 50 key phrases, some of which are redundant. Namely, 10 key phrases are utilized by the developer and 25 key phrases are used by the leader. In addition, some key phrases appear in the same window title. To solve this issue, we revised the set of key phrases by considering those that appear together as additional variable values. In that manner, we consider the developer to utilize a total of 11 distinct window titles and the developer to utilize 31 distinct window titles (aside from the alien titles). Obviously, an action may not contain any element of this set. If none of the key phrases match with a particular window tittle, we call it an *alien title*[4] Otherwise, we identify it with its state represented by One-hot-encoding[5].

5.4 Computing prior distributions

For computing the priors, we adopt an unbiased initial value. Namely, we assume that at first no task is favored above others for either of the subjects. Thus, we consider equal priors for each task for $n = 0$. Thus, as an initial value for our prior belief, $P_0(t)$, we consider,

$$P_0(t) = \begin{pmatrix} 1/|T| & \dots & 1/|T| \end{pmatrix},$$

where the operator $|\cdot|$ represents cardinality of a set.

Starting with fair priors is particularly beneficial for unbalanced data sets as ours (see Table 1), where less frequent tasks are inherently harder to estimate. Namely, reflecting the

[4]The number of actions with aliens titles are found to be 1377 for the developer and 514 for the leader.

[5]The window titles are not represented with IDs, since this would lead to ordinal variables, which introduces a variety of distances between pairs of window titles. However, the One-hot-encoding approach yields same euclidean distance between any pair of window titles, which makes more sense for our case.

actual task distribution on the priors has the risk of missing less frequent tasks, whereas unbiased priors put a larger emphasis on them, potentially increasing estimation rates of those rare tasks.

However, we propose changing this unbiased approach as time elapses, and favoring the previous task performed at $n-1$ over the other at varying levels. Namely, we propose updating the prior as in Equation 3, where a parameter α defines the rate of update,

$$P_n(t) = (1-\alpha)P_0(t) + \alpha P_{n-1}(t). \tag{3}$$

We contrast three cases, where (i) we do not update the priors, or (ii) update them using a linear combination of the initial value and last computed probability value or (iii) adopt the last computed value as the prior for the next step. The 3 cases described above realized using $\alpha = \{0, 0.5, 1\}$, respectively. In the main track, we will present estimation performance relating the case with $\alpha = 0.5$, since its is concluded to yield the best performance.

5.5 Computing likelihood

Essentially, the likelihood term in Equation 1 refers to a multidimensional distribution, since the variables (i.e. descriptors) are defined as a collection of several values. However, if these multidimensional variables come from a large space as in our case (See Table 9 in Appendix D for the range of values of quantitative variables.), the data set may not be big enough to populate it, and arises the risk of having a very sparse distribution.

One approach to tackle with this issue is to investigate the dependence of the descriptors (i.e. components of the variable vector). Namely, provided that the descriptors are independent, it is possible to decompose the likelihood term as follows,

$$P_n(\Delta|t_i) = \prod_{\delta} P_n(\delta|t_i), \tag{4}$$

where $\{\delta\}$ is the set of conditionally independent variables.

In this manner, several low dimensional (possibly 1D) δ distributions are derived from the observation set and a higher rate of samples is achieved in relation to the dimension of the space with a more accurate representation of distributions. However, it is necessary to first confirm the in/dependence of variables (i.e. descriptors), which will be addressed in Section 5.6.

5.6 Investigating dependence of variables

In order to decompose the conditional joint probabilities into a product as presented in Equation 4, we need to verify that these descriptors are conditionally independent.

To do so, we compute the *relative entropy distance* for each pair of descriptors presented in Table 5. Basically, relative entropy distance $D(X,Y)$ between two random variables as $X, Y \in \Delta$, is,

$$D(X,Y) = \frac{H(X,Y) - I(X,Y)}{H(X,Y)},$$

where $H(X,Y)$ is the joint entropy and $I(X,Y)$ is the mutual information defined as,

$$H(X,Y) = -\sum_{x\in X}\sum_{y\in Y} P(x,y)\log_2\left(P(x,y)\right),$$
$$I(X,Y) = \sum_{x\in X}\sum_{y\in Y} P(x,y)\log_2\left(\frac{P(x,y)}{P(x)P(y)}\right).$$

In addition, $D(X,Y)$ takes values in $[0,1]$, where correlated variables attain values close to 1 and uncorrelated ones attain values close to 0.

Table 3 presents the relative entropy distance values between each pair of descriptors regarding Stage-1 and Stage-2 of hierarchical estimation[6]. From these tables, we see that Application and Window title are the two descriptors, which are most likely to be dependent.

Particularly at Stage-1, there is a significant difference between their relative entropy distance and the ones of any other pair of descriptors. Therefore, we decided to consider these two descriptors as dependent variables at Stage-1 and modeled their behavior in a 2D space, whereas all other descriptors are considered to be independent.

Regarding Stage-2, although application and Window title have still a lower relative entropy distance, considering the significant increase in comparison to Stage-1, and the reduced number of samples at Stage-2[7], we opted for independence and modeled them as 1D variables just as the other descriptors.

5.7 Elimination of irrelevant of variables

The descriptors presented in Table 5 may or may not be relevant to determine the task of the user. Namely, certain descriptors may be highly correlated with the tasks, while some others may not present a distinction with respect to the nature of the task.

In order to assess the relevance of each descriptor in task estimation, we use Cramér's V, which is a measure of association between two variables based on Pearson's χ^2 statistics. In explicit terms, Cramér's V is computed as,

$$V = \sqrt{\frac{\chi^2/n}{\min(k-1, r-1)}}, \tag{5}$$

where χ^2 comes from Pearson's test, n is the number of observations; and k and r are number of values that each variable can attain respectively. Cramér's V attains a value in the range $[0,1]$.

Cramér's V values are computed after the preprocessing operation defined in Section 5.3. Note that to realize the operation defined in Section 5.3.1 it is necessary to decide the number of bins. To determine the best number of bins such that the correlation between the descriptors and tasks are highest, we propose using Cramér's V. Namely, we compute V by quantizing the quantitative variables for $1 \leq c \leq 6$ and picked the number of bins,

[6]Since the matrices in Table 3 are symmetric, only the upper triangular part is presented.

[7]Using only the infrequent tasks implies an inherently low number of samples.

Table 3: Relative entropy distance between pairs of descriptors regarding (a) Stage-1 and (b) Stage-2 of hierarchical estimation.

(a)

	Application	Window title	Key strokes	Left clicks	Right clicks	Duration
Application	-	**0.73**	0.98	0.99	0.99	0.98
Window title	-	-	0.97	0.98	0.99	0.97
key strokes	-	-	-	0.97	0.99	0.97
Left clicks	-	-	-	-	0.99	0.95
Right clicks	-	-	-	-	-	0.99
Duration	-	-	-	-	-	-

(b)

	Application	Window title	Key strokes	Left clicks	Right clicks	Duration
Application	-	**0.83**	0.96	0.97	0.99	0.97
Window title	-	-	0.92	0.95	0.99	0.95
key strokes	-	-	-	0.96	0.99	0.94
Left clicks	-	-	-	-	0.95	0.93
Right clicks	-	-	-	-	-	0.98
Duration	-	-	-	-	-	-

Table 4: Assessing relevance of descriptions with Cramér's V.

Descriptor	V	
	Stage-1	Stage-2
Application	0.68	0.89
Window title	0.59	0.46
Key strokes	0.14	0.19
Left clicks	0.11	0.15
Right clicks	0.10	0.17
Duration	0.10	0.14

which yield the highest V. In this respect, Table 4 presents the highest values of V for each quantitative variable.

As seen in Table 4 Application and Window title have by far the highest V values. Therefore, considering the significant difference between the values of V regarding these two descriptors and the remaining ones, we suggest using Application and Window title in estimation of task and omit the others. Although we do not make our choice based on certain

guidelines dividing the range of Cramér's V (into bands so as to interpret implications), in our method we prefer using V for $measuring\ the\ degree\ of\ correlation$ and determine our set of $relevant\ descriptors$ in rather an empirical manner.

6 Conclusion

At the end of this internship, we have been able to propose a robust method that has satisfying performance. Even if it is not a way to measure the success of an internship, it is still a clue on whether things went well or not. Dr. Yücel considered that the work I produced was outstanding and I would like to believe I greatly contributed to this research. I have been able to produce a lot of different scripts that would help me answer in the fastest possible way when I was requested to display or compute a new type of information. Thanks to this work, the research went a step forward and Dr. Yücel will be able to publish a paper.

As regards personal benefits, this internship gave me many opportunities, work experience as much as a social experience. I mainly discovered the world of research, I have always been very curious about it and I am now able to grasp an idea of what it is. One of my frustration was that I could not always work all the time because sometimes I was lost and didn't know what to do. Now, I also understand why you need a Ph.D. to do research on your own. Even though I was already familiar with Python, I feel I got more experience and I am more confident with different libraries like Numpy or Scikit. Meeting my supervisor every morning or every other day was a new way of working that I never encountered before. Not being able to ask a question at any time, as you would in a company, was disturbing in the beginning but it taught more about patience and how to work independently.
Working in Japan was also the opportunity to discover the Japanese culture, which is very different from ours. For example, the administration was surprisingly very rigorous about administrative papers and deadlines. Also, the student from the laboratory, but not only, were always willing to assist and were very kind. I could not talk about this work experience without mentioning the human experience. During this summer, I met so many people and not only Japanese people. Seeing such a varied panel of cultures truly opened my eyes, it made me think twice on the foundation of our own culture and I think that these kinds of experiences make you grow faster than you could have in years if you had stayed in France your whole life.
For that reason, I could not thank Dr. Zeynep Yücel and Dr. Akito Monden enough for this internship and the opportunity they gave me.

Appendices

A Appendix on subsets of the activity log and the rules

Table 5: An excerpt from the activity log data set.

Start time	End time	Left click	Right click	Key strokes	Application name	Window title
18:09:13	18:09:15	1	0	0	explorer.exe	hebijun
18:09:15	18:09:20	1	0	0	explorer.exe	TaskpitLog
18:09:20	18:09:22	1	0	0	sakura.exe	TaskPit\setting.ini
18:09:22	18:09:23	2	0	0	sakura.exe	sakura 1.6.3.0
18:09:23	18:09:30	1	0	0	explorer.exe	TaskpitLog
18:09:30	18:09:33	2	0	0	explorer.exe	hebijun
18:09:33	18:09:34	1	0	0	excel.exe	AIRS共通機能仕様書_ 0511.xls
18:09:34	18:09:35	1	0	0	explorer.exe	hebijun
18:09:35	18:09:37	0	1	0	excel.exe	AIRS共通機能仕様書_ 0511.xls
18:09:37	18:09:38	1	0	0	ipmsg.exe	IPMsg

Table 6: The set of association rules defined by the expert.

	Application	Window title	Candidate Tasks Candidate-1	Candidate-2	Candidate-3
1	devenv	ファイル内の検索	Programming	-	-
2	soffice	査読シート	Administration	Documentation	
3	tortoiseproc	*	Test	Programming	-
4	airs_ovly	*	Test	-	-
5	airs_psam	*	Test	-	-
6	aliim	*	Leisure	-	-
7	bcompare	*	Test	Documentation	-
8	devenv	*	Programming	Test	-
9	firefox	*	Programming	Leisure	Test
10	ipmsg	*	Administration	-	-
11	msimn	*	Administration	-	-
12	sakura	*	Programming	Test	Documentation
13	editplus	*	Documentation	-	-
14	taskpit	*	Administration	-	-
15	*	Test	Test	Documentation	-
16	*	淘宝	Leisure	-	-
17	*	taobao	Leisure	-	-
18	*	服装	Leisure	-	-
19	*	天猫	Leisure	-	-
20	*	支付宝	Leisure	-	-

[7] In this table, an entry of * denotes "any value" (of application or window title), whereas an entry

.

B Issues with association rules

Ideally, we would like to assign a single task to each action in the activity log. However, the set of rules provided in Table 6 may not always associate a single task with an action. Let us consider the hypothetical example given Table 7, which skips the descriptors and presents only the -candidate- tasks after directly applying the association rules on each action. Actions 1 and 9 in Table 7 satisfy a single association rule with a single outcome (i.e. task) and thus they can directly be associated with that task without any problems. On the other hand, some actions (e.g. actions 5 and 8 in Table 7), do not satisfy any of the conditions and thus the rules yield no candidate tasks for them. In addition, some actions (e.g. actions 2 and 3 in Table 7) receive multiple candidates. There may principally be two reasons leading to this discord. First of all, some conditions do not determine the task decisively, but yield a set of several possible tasks (e.g. rules 2 and 3 in Table 6)[8]. Secondly, an action may satisfy several conditions at once and thus receive multiple estimations. In either case, the task associated with that action cannot be determined decisively.

In particular, we call an action, which is associated with a single candidate task, to be *estimated*, while actions which are not associated with any candidates are termed as *inconclusive* and those associated with multiple candidates are called *uncertain*. An ideal set of association rules should not yield any inconclusive or uncertain outcomes.

Table 7: Application of association rules.

	Candidate 1	Candidate 2	Estimation
1	Test		**Test**
2	Documentation	Programming	
3	Programming	Test	
4	Programming	Test	
5			
6	Programming	Test	
7	Documentation	Leisure	
8			
9	Documentation		**Documentation**

For our data set, the number of inconclusive, estimated, and uncertain actions are as presented in Table 8. Note that this table relates only the number of candidate estimates and as such the 0.09 actions, which are estimated, are not necessarily estimated with the correct task. The discussion on the accuracy of these estimations will be provided in Section **??**.

In order to assign a single task to each action, it is necessary to do some post-processing operation such that the gaps in estimation are filled and multiple estimations are reduced

of '-' denotes "no value" (of candidate task).

[8]In particular, from the 20 rules presented in Table 6, 13 of them give a single estimation, whereas 5 rules give 2 estimations and 2 rules yield 3 estimations

Table 8: Estimation status of the actions after direct application of the rules.

	Percentage of actions
Inconclusive	0.57
Estimated	0.09
Uncertain	0.34

to a single one. The details of the post-processing operation, which is defined by the same expert, is be explained in the Section C.

C Post-processing outcomes of association rules

The expert defined a series of post-processing operations, so as to fill in missing estimations and reduce multiple estimations to single. In particular, the post-processing operations involve the following steps:

1. If the active application is a development environment and the number of key strokes is above 30, then the estimated task is Programming. If the active application is a browser and the time is between 12:30 and 13:00, then the estimated task is Leisure.
2. If two subsequent actions are not assigned any estimated task but have a single candidate task in common, then their estimated tasks are determined as this candidate (see Algorithm 1).
3. For each estimated action, the task associated to it is propagated to the preceding (not-estimated) action, provided that there is at least one common task among their candidates (see Algorithm 2). A similar procedure is applied for the succeeding action.
4. For each estimated action, the task associated to it is propagated to each preceding (not-estimated) action, if the estimation can be found amongst the candidates of that preceding action. This process is repeated until it reaches an estimated action (see Algorithm 3). The same is applied for the succeeding actions).
5. For each estimated action, the task associated to it is propagated to each preceding (not-estimated) action irrespective of their candidates, until an estimated action is found (see Algorithm 4). The same is applied for the succeeding actions.

Amongst these five post-processing steps, the first two are causal since they rely only on the past and current information. On the contrary, Steps 3, 4, and 5 are non-causal, i.e. each of them requires future estimations and propagates that information on preceding actions. Since these steps use some information, which is yet unknown at the time that the action is performed, they cannot be used on a real time system.

Algorithm 1: Step-2 of post-processing.

```
for n ← 1 : N − 1 do
    if ∄ EST[n] ∧ ∄ EST[n + 1] then
        if | CANDID[n] ∩ CANDID[n + 1] | = 1 then
            est ← | CANDID[n] ∩ CANDID[n + 1] |
            EST[n] ← est
            EST[n + 1] ← est
```

Algorithm 2: Step-3 of the post-processing.

```
for n ← 2 : N − 1 do
    if ∃ EST[n] then
        if | CANDID[n] ∩ CANDID[n − 1] | > 0 then
            EST[n − 1] ← EST[n]
        if | CANDID[n] ∩ CANDID[n + 1] | > 0 then
            EST[n + 1] ← EST[n]
```

Algorithm 3: Step-4 of the post-processing.

```
for n ← 2 : N − 1 do
    if ∃EST[n] then
        for j ← n : n − K | K = max(k), ∄EST[n − k], ∀n − k do
            if EST[n] ∈ CANDID[j] then
                EST[j] ← EST[n]
        for j ← n : n + K | K = max(k), ∄EST[n + k], ∀n + k do
            if EST[n] ∈ CANDID[j] then
                EST[j] ← EST[n]
```

Algorithm 4: Step 5 of post-processing

```
while ∃n | EST[n] = ∅ do
    for n ← 2 : N − 1 do
        if EST[n − 1] = ∅ then
            EST[n − 1] ← EST[n]
        if EST[n + 1] = ∅ then
            EST[n + 1] ← EST[n]
```

D Appendix on statistical properties of descriptor values

Table 9 presents the minimum, maximum, mean and standard deviation values relating the four quantitative descriptors. Although the tasks of the developer and the leader are distributed in a quite different manner (see Table 1, This table indicates that no significant distinction is present between descriptors of the developer and the leader. In other words, the variation between the mean value of any quantitative descriptor is within a single standard deviation concerning either of the subjects.

Table 9: The minimum, maximum, mean values and standard deviations for the four quantitative descriptors.

	Descriptor	Min	Max	Mean	Std
Developer	Duration (sec)	1	284	14.41	30.79
	Left click	0	87	2.41	4.34
	Right click	0	3	0.03	0.19
	key strokes	0	121	1.89	8.48
Leader	Duration (sec)	1	301	16.18	33.81
	Left click	0	55	2.50	4.36
	Right click	0	8	0.06	0.40
	key strokes	0	326	2.94	17.24