

Reports

Machine Learning and Inference Laboratory

**Learning User Models
for Computer Intrusion Detection:
Preliminary Results from Natural Induction Approach**

Ryszard S. Michalski
Kenneth A. Kaufman
Jarosław Pietrzykowski
Bartłomiej Snieżyński
Janusz Wojtusiak

MLI 05-3

P 05-6

November, 2005



School of Computational Sciences

George Mason University

LEARNING USER MODELS FOR COMPUTER INTRUSION DETECTION: PRELIMINARY RESULTS FROM NATURAL INDUCTION APPROACH

Ryszard S. Michalski*, Kenneth A. Kaufman, Jaroslaw Pietrzykowski,
Bartłomiej Snieżyński**, Janusz Wojtusiak

Machine Learning and Inference Laboratory, George Mason University,
Fairfax, VA 22030-4444

* Also with the Institute of Computer Science, Polish Academy of Sciences, Warsaw

** Also with Institute of Computer Science, AGH University of Science and Technology, Krakow

{kaufman, michalski, jarek, bsniezynski, jwojt}@mli.gmu.edu

<http://www.mli.gmu.edu>

Abstract

This paper presents a description of the LUS method for creating models (signatures) of computer users from datastreams that characterize users' interactions with computers, and the results of initial experiments with this method. By applying the models to new user activities, the system can detect an imposter, or verify a user's legitimate activity. In this research, original datastreams are lists of records extracted from the operating system's process table. The learned user signatures (LUS) are primarily in the reported results in the form of sets of *multistate templates (MTs)*, each characterizing one pattern in the user's behavior. Advantages of the method include the significant expressive power of the representation (a single template can characterize a large number of different user behaviors) and the ease of their interpretation, which makes possible their editing or enhancement by an expert. Presented initial results show a great promise and power of the method.

Keywords: Intrusion detection, learning user models, machine learning, rule learning, target data preparation, testing user models.

Acknowledgments

The authors express their deep gratitude to Dr. Thomas Goldring for general guidance and feedback on this project, and for supplying us the data used in the experiments. The development of heat maps was done due to his advice. Chien-Chih Lin contributed a statistical analysis of the LUS data.

This research was supported in part by the UMCB/LUCITE #32 grant, and in part by the National Science Foundation under Grants No. IIS-0097476 and IIS-9906858. The findings and opinions expressed here are those of the authors, and do not necessarily reflect those of the above sponsoring organizations.

Table of Contents

1	Research Goals	1
2	Basic Concepts and LUS Overview	2
2.1	LUS Methodology	2
2.2	Steps of the LUS Process.....	4
2.3	Related Research	9
3	Data and Problem Description	10
3.1	Available Datasets Collected from Computer Users.....	10
3.2	Description of the Target Dataset.....	10
4	Methods for Learning and Testing User Models.....	20
4.1	Learning User Models	20
4.2	Testing and Application.....	25
5	Data Preparation and Selection.....	30
5.1	Event Selection.....	30
5.2	Attribute Discretization	32
5.3	Window Size / Lookback.....	43
5.4	Attribute Selection.....	44
5.5	Determining Training and Testing Data Streams.....	46
6	Plan of Experiments.....	48
6.1	Experiment Set 1: Search for the Best Representation Space	48
6.2	Experiment Set 2: Search for the Minimum Amount of Data Needed for Learning	48
6.3	Experiment Set 3: Search for the Best Filtering Parameters	49
6.4	Experiment Set 4: Search for the Best AQ21 Parameters.....	50
7	Illustration and Validation of the LUS Method by Diagrammatic Visualization	53
8	Experimental Results and Evaluation.....	62
8.1	Measuring Similarity between Episodes.....	62
8.2	Event Selection Experiments	72
8.3	Output Value Selection for Prediction-based Model	74
8.4	AQ21 Experiments with Data from 10 Users, 10+5 Sessions	80
8.5	AQ21 Experiments on Data from 10 Users: Window Records Only, 10+5 Sessions.....	155
8.6	Experiments on 10 Users, All data	180
8.7	Summary of Experimental Results.....	181
9	Conclusion and Future Plans.....	182
	References.....	184
	Appendix A: Dictionary of LUS Methodology Terms.....	A1
	Appendix B: Description of Attributes	B1
	Appendix C: Selected MT User Models	C1
	C1 Experiment 040607-1: Filtered Data TR+TS, Discriminant Descriptions.....	C1
	C2 Experiment 040607-2: Filtered Data TR+TS, Characteristic Descriptions	C11
	C3 Experiment 040727-3: Unfiltered Data, Simplicity-based Descriptions.....	C26
	Appendix D: Heatmaps for selected Experiments	D1

1 RESEARCH GOALS

This report describes research on the development of a new approach to modeling users' interactions with a computer and using the models for detecting computer intrusion. The approach, called *Learning User Signatures (LUS)*, applies symbolic machine learning to discover general and consistent patterns in the interactions between users and computers, and then uses these patterns to confirm the legitimate use of a computer or indicate a possible computer intrusion.

Given data characterizing interactions between users and computers (in this research, records in a process table), LUS creates models of users' behavior, called *symbolic user signatures*, that capture regularities in the users' behavior that are both characteristic for each user and differentiate the users from each other. In this research, we have been developing and investigating user signatures in the form of *multistate templates* relating measured characteristics to individual users. Multistate templates are derived from expressions in *attributional calculus* that are generated by a learning program from training data. Attributional calculus is a highly expressive, logic-based language that can concisely represent complex inter-attribute relationships (Michalski, 2004).

An important aspect of the LUS methodology is that it strives to generate user models that not only have a high predictive accuracy in recognizing users but also are relatively easy to interpret and understand. This means that these models can be inspected and verified by experts, and possibly hand-adjusted or improved, if needed. In research reported here, the models were learned using our newest symbolic learning system, AQ21. Given training data in the form of a set of attribute-value vectors, AQ21 creates attributional rules that generalize the data and optimize a user-defined rule quality criterion (Wojtusiak, 2004).

The goals of the research presented here were to advance the LUS methodology by developing and implementing a variety of new ideas and methods, and to experimentally test it on datasets representing actual user activities. This report describes new developments, implemented methods, performed experiments, and their results. In the experiments, we used datasets generated from Windows-based operating systems, both unfiltered and filtered. The filtered datasets focused on records that were considered most characteristic of each user according to such measures as *commonality*, *distinctiveness*, and *significance*.

We developed four different general user model representations: *multistate template* (MT), *prediction-based* (PB), *hybrid rule-Bayesian* (RB), and *activity-based* (AB). To be able to test user models developed using different representations, we developed algorithms and implemented application programs for each of these representations. These application programs include EPIC-MT, for testing *multistate template* rule models; EPIC-P, for testing prediction-based rules, and EPIC-RB, for testing the hybrid rule-Bayesian model.

Because the scope of the proposed new methods and desirable testing experiments turned out to be exceedingly large, the current study only explores a relatively limited portion of them. Specifically, we concentrated primarily on a systematic experimentation and testing of the MT model representation and the exploration of other model representations was put on the agenda for future research.

For the reader’s convenience, Appendix A provides a dictionary of terms introduced in the LUS methodology. Appendix B provides a detailed description of attributes used in the experiments. Appendix C provides a selection of results obtained from the experiments that are not described in the main text. Appendix D illustrates selected results through heatmap visualizations.

2 BASIC CONCEPTS AND LUS OVERVIEW

2.1 LUS Methodology

To provide foundations for describing this research, we start with an explanation of the basic concepts and terms used in this report.

An *event* is a description of an entity or situation under consideration. In the case of LUS research, an event is a vector of attribute-values characterizing the use of the computer by a user at a specific time instance or during a specific time period. An example of an event is an n -gram, which is a list of n attribute values characterizing user behavior at n consecutive time instances. An extension of an n -gram is a *multi-attribute nxk -gram*, which is a sequence of values of k attributes occurring in n consecutive time instances. One of the main novel features of this research is that we have been working with nxk -grams, rather with n -grams, as we did before. To indicate simply the difference between these two approaches, we refer to n -grams as *unigrams*, and nxk -grams as *multigrams*.

A *session* is a sequence of events characterizing a user’s interaction with the computer from login to logoff. An *episode* is any sequence of events; it may contain just a few, typically, consecutive events, or all of the events in a session. In the training phase, it is generally desirable to use long episodes, or even whole sessions, in order to generate better user models. In the testing (or execution) phase, it is desirable to use short episodes, to identify a user from as little information as possible.

A *pattern* is a frequently occurring regularity in data. A pattern is characterized by a *pattern description*, which is an expression in a knowledge representation system. Such an expression can be in the form of, for example, decision rules, a decision tree, a neural network, a Bayesian net, or, as in the case of LUS, an *attributional ruleset*— a set of rules in Attributional Calculus (Michalski, 2004) that characterizes the interaction between a user and the computer.

Initial LUS experiments focused on user models employing values of a single attribute. Specifically, events were n -grams: sequences of n consecutive values of the *mode* attribute extracted from the user datastream. The behavior of a user was described by a set of consecutive, overlapping n -grams (events) spanning a given period of user interaction with the computer. In the current research, we used nxk -grams involving several attributes, selected as most relevant for characterizing individual states of users’ behavior. The attributes were selected from a repository of attributes constituting a union of attributes originally provided in the datastream and *derived* attributes, constructed from the original attributes or extracted from the given data files.

A *user model representation* is a general knowledge structure used for characterizing a user’s interaction with a computer. As mentioned above, we have developed several novel user model representations: Multistate Template (MT), Prediction-based (PB), Rule-Bayesian hybrid (RB),

and Activity-based (A). A *user model* is an instantiation of a user model representation that characterizes the behavior of a specific user.

User models can be developed and applied using a single representation, or a combination of representations. When user models in two or more representations are applied to a given datastream, a classification decision can be assigned by voting, which can be weighted or unweighted. Because of the possibility of using different model representations, and because each of the user models can be learned and/or applied with different parameters, the LUS methodology opens a very wide range of possible avenues for research and experimentation.

The LUS methodology aims at developing methods and computer programs able to create computer user models that resemble human recognition processes in terms of the following criteria:

- A. *Idiosyncrasy*: To discover patterns in user's interactions with the computer that are most characteristic of the given user, so that identification of the given user may be possible from short episodes that contain such patterns.
- B. *Satisfiability*: If at some point of observing datastreams characterizing behavior of different users, the observed behavior strongly matches one user model and only weakly matches other models, no further observation is conducted, and the decision identifying the user is reported. If the correct user model is not confirmed after a specified period of observation time, a possible intrusion is reported.
- C. *Understandability*: User models should be easily understandable and potentially modifiable by a person supervising the intrusion detection system.
- D. *Flexibility*: The methods of model creation and application should have potential to reflect various aspects of the problem according to the preferences of the user of the intrusion system.
- E. *Incrementality*: User models should be incrementally updatable in order to capture changes in the user's activities without completely re-learning all the user models.
- F. *Applicability*: User models should be in a form that can be efficiently applied to new data for recognizing users.

Due to the employment of the AQ learning methodology in LUS, all six criteria can be satisfied.

The LUS methodology assumes that during the process of using a computer system, some periods are designated randomly by a system manager to be *training phases*. In a training phase, a machine learning system "watches" activities of authorized users, and creates or updates models of legitimate uses of the computer system ("User Signatures"). To learn such descriptions, an advanced symbolic learning system is used. With the use of such a system, LUS produces effective user models that are understandable to humans. In LUS, the behavior of a user is characterized not by a single pattern, but by a collection of patterns that try to capture different kinds of user activity. Each pattern, represented as a single attributional rule, is associated with an estimated frequency of its occurrence (called *weight*), and some other parameters. Patterns represented this way can be updated relatively easily by a method of *incremental learning* in order to represent changes in the user's behavior.

During the testing or execution phase, LUS applies the user signatures to datastreams coming from individual users, and determines whether they sufficiently match the legitimate behavior of the purported users.

The sections that follow introduce a number of additional concepts and terms, specifically related to the topic of discussion. A dictionary of selected terms used in LUS is provided in Appendix 1.

2.2 Steps of the LUS Process

Let us assume that a set of users whose models we wish to build has been selected, and raw data streams characterizing their behavior (e.g., interaction with the computer) have already been collected.

A user raw data stream is in the form of a table in which columns represent attributes measuring a temporal process at given time instances or time periods (e.g., the state of the activity of a user or a group of users), and rows represent vector of attribute values in the consecutive states of user activity. A state can be measured at a time instance or over a time period. In the case of a time period, the time interval between two consecutive time units can be constant or variable.

The attributes can be of the following types: nominal, rank, cyclic, structured, interval, ratio or absolute (Michalski, 2004; Wojtusiak, 2004). In addition to the attributes characterizing states of users' activities, the data also include meta-attributes such as User ID (the user's identification number), and the time of the observation (expressed either as the time relative to an agreed starting point, or as the absolute time).

The LUS methodology consists of the following steps:

1: Define the attributes to be used in the target dataset.

This step involves the selection of attributes in datastreams to be used for creating user models. In addition to the attributes explicitly included in the raw datastream, additional attributes can be constructed from the raw data and metadata that appear relevant to the task at hand. These *derived* attributes may characterize entire episodes (e.g., date of observation or host machine), or can be computed from the attributes in the datastream. An example of a derived attribute is the number of characters in *protected* words in the window title.

2: Transform the raw datastreams into the initial target dataset to be used as input to the model learning system.

This is done by converting the input files into a form acceptable by the machine learning program. This step also includes adding to the data derived attributes and computing their values for each example. In the experiments using the *multistate template model*, the desired data format is a table of $n \times k$ -grams labeled by the user name and episode number meta-attributes. In the *prediction-based model*, the input data table consists of *lookback* k -grams labeled by the user name, episode number, and *lookforward* complexes (descriptions) representing what is observed subsequently.

In our experiments, records were of two types: *window* type (records describing the active window), and *activity* type (records describing processes taking place in the active windows). In

some experiments, only the window records were used. If the raw data are heterogeneous (records are of different types), an additional step must be taken. Either a *complete target dataset* (containing all records) is generated (with attribute values filled in where they were not in the original data schema for the record), or a *partial target dataset* (in which the records have been filtered based on type).

3: Select a scheme for discretizing numerical attributes

All numerical attributes in the data are discretized into a small number of ranges. Such a discretization is done in two steps:

3.1: Select candidate discretization schemas.

There are two types of schemas that may be used for discretizing attributes: manual, in which the user selects thresholds demarcating ranges of values, and automatic, in which a program automatically creates such thresholds.

In our experiments, we used two manual schemes, denoted Dis-1 and Dis-2, which were created by analyzing distributions of values of each attribute in the data for all users, and seven automatic schemas, Chi-3, Chi-4, ..., Chi-9, utilizing the ChiMerge method (Kerber, 1992), and set to discretize the values to 3-9 intervals in accordance with the schema name.

3.2: Select the best discretization schema, called Dis-Opt (originally, Dis-3), for each attribute from among all candidate schemas determined in 3.1.

This is done by applying programs that compute attribute quality measures (e.g., PROMISE, Gain Ratio) for each discretization schema generated in 3.1. Based on these values and the numbers of intervals created, the best discretization schema among the candidates is selected.

4: Select the training data size

Using a similarity measure among datasets, the consistency of a decision class's (e.g., a user's) behavior in the processed dataset is determined. If the consistency is very low, e.g., the behavior recorded in the later part of the data stream is very different from that in the early part, it will not be possible to learn a model from the early part that will reliably classify the behavior in the later part.

The low behavior consistency in the processed data may be due to employing insufficiently relevant attributes for characterizing behavior or due to significant changes in the actual behavior. To address the first problem, more relevant attributes need to be used for characterizing the observed behavior. To address the second problem, more data needs to be collected, so that it sufficiently reflects the range of different behaviors that may be observed.

This step seeks the minimum size of the training dataset for each decision class needed to creating a reliable model. For this purpose we developed the "sausage" method (Section 5.5)

5: Select the experiment's target dataset.

This step involves selecting the subset of the initial target dataset of analysis (training and testing). This means selecting a subset of the data from which training will take place, a subset

of the data from which testing will take place, and a subset of the values of the output attribute for which models will be learned and tested.

This step should ideally ensure that includes there is a sufficient amount of data for each model to be learned and tested, but not more than is necessary to achieve close to the best results possible.

For example, a number of LUS experiments had the form: Use the first ten episodes from each user for training, use the next five episodes from each user for testing, and only learn and test models for users who were represented by at least 35 episodes in the raw data stream.

6: Select the most relevant attributes

The relevance of all attributes in the target data is evaluated, and the most relevant attributes are selected in two steps:

6.1: Manual elimination

The user can mark certain attributes known to be irrelevant to be automatically ignored by the learning program. It is computationally more efficient to manually ignore irrelevant attributes than pass them to the algorithm in step 6.2 and let the program select them for removal.

6.2: Automatic selection

During this step, attribute quality is calculated for each attribute, and attributes are selected accordingly (e.g., choose the k best). Attributes may be selected for the entire dataset, or separately for each decision class (e.g., user model).

The former method will select the same set of attributes for each decision class. The quality of each attribute is evaluated using four different methods (Promise and Gain Ratio, based on average and maximum for a decision), and on this basis, attributes that score high on multiple lists are selected for use. The description of the selection algorithm can be found in Section 5.4.1.

It is, however, possible to use a different attribute set for learning each individual model. This can be done in the same way as described above (Promise, Gain Ratio, average, maximum, or a combination of those components). However, these methods are not being applied to separation of all decision classes, but rather the separation of one target class from the general negative class. Different attribute sets may then be chosen for each target class.

7: Select the most relevant training data

This step seeks events in the processed data that are most relevant to learning reliable models. The following methods of event selection are used in the experiments:

7.1: SB (Significance-based) methods

Apply different event significance measures to the dataset. Nine candidate measures have been developed, based on some combination of event distinctiveness and commonality (Section 5.1).

7.2: HCHD (High Commonality-High Distinctiveness) method

Select events with either high distinctiveness or high commonality (a disjunctive measure, as opposed to the conjunctive ones of 7.1).

7.3: Frequency-based method

Select events that have a high frequency of appearance for a user, regardless of their measured significance.

8: Learn models from training dataset

This step takes the training dataset and induces models for each decision from it. In the experiments, we used the newest AQ-type learning program, AQ21 (Wojtusiak, 2004), which allows the user to control the type of the model to be created and the way it is created. Through experiments, settings of parameters are sought that will lead to the best performance of the learned models. The parameters include:

8.1: Learning mode

- Theory Formation (TF), which creates models, in the form of attributional rulesets that are complete and consistent with regard to the training data.
- Pattern Discovery (PD), which creates models that represent strong patterns in the data. The patterns are also in the form of attributional rulesets, but these rulesets may be partially inconsistent and/or incomplete with regard to the data.
- Approximate Theory Formation (ATF), which first creates rulesets as in TF mode, but then optimizes rules in certain ways (Q-optimization). The obtained rulesets may be partially inconsistent and/or incomplete, as in PD mode.

8.2: Description type

- Characteristic, in which the program seeks for descriptions that cover most of the positive examples and list maximum number of features (selectors) that describe the the positive examples
- Discriminant, in which the program seeks the simplest descriptions that describe most of the positive examples
- Simplicity-based, in which the program seeks for the simplest descriptions, without checking how many examples are covered by the descriptions

8.3: Rule generality (the trim parameter provides instructions on how to postprocess a maximally general rule)

- Most specific: Intersect the rule with the *refunion* of positive covered examples, so that it results in the most specific rule possible with that coverage.
- Most general: Leave as is.

- Optimal: For each condition in the discriminant-form rule, intersect the condition with the *refunion* of the covered positive examples. The application of the reunion operator to a set of events creates the most specific attributional rule covering the events.

8.4: Ambiguity (*parameter to determine how to handle identical training examples in the positive and negative classes*)

- Ignore: do not use for learning this ruleset
- Include-in-positive: treat the ambiguous example as a positive example
- Include-in-negative: treat the ambiguous example as a negative example
- Include-in-majority: place the ambiguous example in the positive or negative class depending on which class has more occurrences of the example. If both classes have the same number of occurrences, simply ignore the example.

8.5: Exceptions

If this parameter is "On", the program seeks rulesets that may contain exception conditions (Michalski, 2004). If the parameter is set to "Off", learned rulesets will have no exception conditions.

8.6: AQ21 specific parameters such as *maxstar* and *maxrule*

maxstar defines the width of the beam search by setting a ceiling on the number of candidate hypotheses to be retained at each stage of star generation.

maxrule determines the number of hypotheses to be selected from a star for consideration in the final ruleset.

9: Testing and application of learned models

This step selects a testing method that applies some ruleset interpretation (evaluation) schema (Michalski, 2004) in order to match the events in the testing data against the learned models. Such matching is done using a combination of two programs: ATEST, which computes a degree of match between a ruleset and an individual event in the testing data, and EPIC, which combines degrees of match determined by ATEST for a sequence of events (a testing episode) to compute a degree of match between the episode and the different models. Both programs have different control parameters that can be appropriately tuned for the problem at hand (Section 4.2).

9.1: ATEST parameters

- Evaluation of selectors -- generating a degree of match between a condition and an event
- Evaluation of Conjunctions -- combining selector evaluations for an entire rule
- Evaluation of Disjunctions -- combining rule evaluations for an entire ruleset
- Acceptance Threshold -- minimum degree of match for a hypothesis to be considered
- Accuracy Tolerance -- range of consideration for degrees of match

9.2: *Episode classification algorithms:*

- EPIC-MT -- uses scheme described in 9.1 to evaluate events, and then aggregates them using the average function.
- EPIC-SDA -- Like MT, but halts before an entire episode is evaluated and reports a classification if the evidence up to that point warrants it.
- EPIC-RB -- Bayesian formula, for combining evaluations of events.
- EPIC-P (for prediction-based models) -- matches premises and consequents.

2.3 Related Research

Among the common approaches to intrusion detection are statistical methods, in particular, Bayesian methods (e.g., Goldring et al., 2000, Goldring, 2002, Valdes and Skinner, 2000; Scott, 2004), statistical modeling (e.g., Shah, Jonckheere and Bohacek, 2002), pattern matching (e.g., Hofmeyr, Forrest and Somayaji, 1998; Streilein, Cunningham and Webster, 2002), and various data analysis methods (e.g., Novak, Stark and Heinbuch, 2002; Mukkamala and Sung, 2002). A technique that learns from observation rather than from known examples, described by Eskin et al. (2002), applies multiple strategies to recorded system events mapped onto a feature space, in order to identify anomalous behavior. These methods all examine overall system or network behavior in search of indicators of misuse.

A variety of methods of user profiling, in which the activities of individual users or purported users are compared to their known patterns of legitimate use, have been developed. Traditionally, user profiling has relied heavily on the use of statistical approaches. For example, Valdes (2002) describes a multi-approach system that combines sequence matching and statistical analysis to identify user behaviors, and Goldring (2002) applies a probabilistic scoring system to match episodes with user profiles.

As does the method presented here, the work of Schonlau and Theus (2000) also bases its anomaly detection on observing the various processes run. Their approach is to compile a list of invoked commands, and generate a statistical plot of commands' popularity, both for individual users, and for all users together. During the application phase, episodes are then compared to these profiles, and high levels of uniqueness may set off alarms. Another approach, described by Lane and Brodley (1999), employs an n -gram-based representation of sequences, but rather than using processes as the basic units of the n -grams, their method uses command line tokens. Thus, a single command can result in several instances in the input data stream. Their approach is to apply a similarity-based measure between known legitimate behavior and new events, but it does not include a simple way of articulating learned user patterns.

Also of relevance to the work presented here is the method presented by Hätönen et al. (1996), which generates and selects association rules characterizing sequences of alarms in telecommunication networks. Their method employs a predictive model in which knowledge takes the form, "If this set of events has been observed, expect the following event(s) to occur."

The LUS approach is different from existing methods in many significant ways, which are described in the following sections.

3 DATA AND PROBLEM DESCRIPTION

3.1 Available Datasets Collected from Computer Users

In the process of our research, we had access to several datasets recording behavior of computer users. The first dataset, which recorded 400 user sessions, was collected from a Unix operating system's process table characterizing activities of four different users. We refer to this dataset as DS1. Prior work by Goldring et al. (2000) evaluated several existing methods for user modeling and scored them on these datasets. The user models created by these methods could be viewed as "black boxes" in the sense that, although they generally produced encouraging numerical results, they did not provide much insight into the nature of the users' "styles."

Two subsequent datasets were collected from Windows machines. The first contained information on 23 users, covering a total of 777 sessions. We refer to this dataset as DS2. The second, a subset of DS2, contained collected information on seven users (from 123 sessions). A new dataset DS3, also based on Windows use, contains information about activities of 20 users, covering a total of 702 sessions. A more extensive version of the DS3 data (including later sessions from these users, plus sessions from six additional users) has since been made publicly available at a New Jersey Institute of Technology website. Because it includes the original DS3 data, this dataset is referred to in this report as DS3.2, and the original one (provided on CD in 2002) as DS3.1. In this report, we concentrated on the analysis of DS3.2, and built and tested computer user models based on it.

3.2 Description of the Target Dataset

DS3.2 consists of 1292 sessions, characterizing activities of 26 users. Users, number of sessions, and dates of data recording are listed in Table 1 below.

The shaded rows indicate the ten users with the most recorded sessions in DS3.2; it was these users' activity that was examined in the experiments discussed here. Figure 1 depicts the timing of the session recordings in more detail. Each user's periods of recorded activity are mapped onto a single timeline. From this figure we can see, for example, that recording User 1's activities started on December 3, 2001 and ended on January 17, 2003. During such a long period, User 1's activities could have changed substantially.

We selected data from the above ten users for analysis in order to facilitate a comparison of our results with earlier research on profiling this data.

The DS3.2 dataset contains two types of records: (1) records announcing a new or killed active window, or a change to the title of the active window, and (2) records reporting process activity. We call the former records Window (W) Records, and the latter ones Activity (A) Records. The attributes used in these types of records are described in Table 2. Figure 2 shows a small session record in its raw form.

User	Number of Sessions	First Session Recording Date	Last Session Recording Date
1	287	12/3/01	1/17/03
2	54	11/29/01	2/28/02
3	37	12/14/01	10/21/02
4	134	11/28/01	11/5/02
5	34	11/29/01	9/5/02
6	1	6/3/02	6/3/02
7	193	11/27/01	12/17/02
8	167	11/26/01	11/12/02
9	21	5/8/02	7/23/02
10	17	12/28/01	9/19/02
11	6	6/28/02	8/12/02
12	35	11/27/01	1/22/02
13	14	12/3/01	4/8/02
14	1	12/28/01	12/28/01
15	1	10/16/02	10/16/02
16	7	4/4/02	4/29/02
17	15	11/29/01	2/20/02
18	5	12/26/01	12/26/01
19	134	11/27/01	8/19/02
20	10	12/3/01	5/16/02
21	5	12/27/01	1/18/02
22	1	12/4/01	12/4/01
23	1	2/5/02	2/5/02
24	4	5/31/02	8/21/02
25	99	1/29/02	7/5/02
26	9	11/26/01	8/21/02

The ten shaded users were selected for the initial experiments.

Table 1: User and session information in DS 3.2.

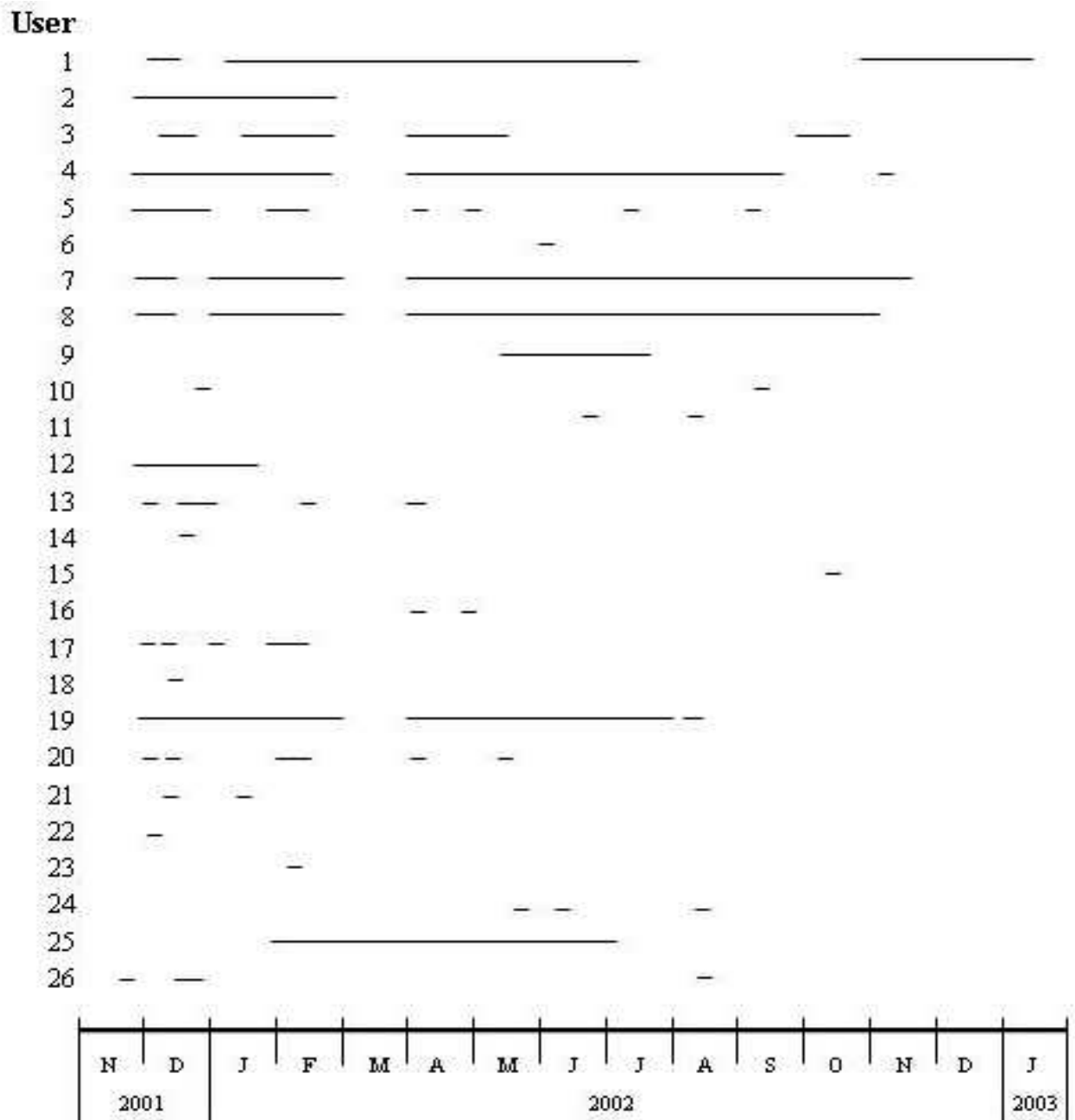


Figure 1: Timeline indicating dates of users' recorded sessions.

Unlike its predecessors, DS3.2 does not contain information from all records extracted from a user's session. Rather, it focuses on activity initiated by the user in the user's active window. To distinguish its various aspects, DS3.2 contained three copies of the data, identical save for the level of filtration:

- The dataset indicated by the file extension *1s* contains only process records whose process was the main active window process.

- The dataset indicated by the file extension *2s*, in addition to the *1s* records, also contains process records whose process name was the same as that of the main active process in the process' window.
- The dataset indicated by the file extension *3s*, in addition to the *2s* records, also contains records of processes that were spawned by other processes listed in the *2s* data.

No.	Attribute Name	Data Type	Record Type	Attribute Description
1	LineNo	Parenthesized Integer	W, A	Line number in the raw process table data corresponding to this record
2	Delta_t	Real	W, A	Number of seconds since start of session when this record was generated
3	ProcessName	String	W, A	Name of the user's current process or window
4	PID	Integer	W	Process ID of the active window
5	Status	Character	A	Indicates whether process is new, continuing, ending, or running in the background
6	CPU	Real	A	Total amount of CPU time used by the process when this record was generated
7	WinName	String	W	Name of the active window (sanitized)
8	Lineage	String	W, A	Lineage from the parent window(s)

Table 2: Original attributes used for characterizing user behavior in DS3.2.

An example of a Window record is presented below, where the attribute LineNo is “(532)“, Delta_t is “45.035“, ProcessName is not present, PID is “303“, Status and CPU are not applicable for this record type, WinName is “<<16624>> - (<<15265>>)” (contains only sanitized words), and Lineage is “:303:”.

```
(532)      45.035      pid = 303      <<16624>> - (<<15265>>)
                                         :303:
```

Another example below shows an Activity record, in which the attribute LineNo is “(680)“, Delta_t is “134.354“, ProcessName is “explorer“, PID and WinName are not applicable for this record type, Status is “c“, CPU is “0.841“, and Lineage is “:explorer:268:”.

```
(680)      134.354      explorer      c      0.841      :explorer:268:
```

Figure 2 shows a *.1s* file (in the form of a table) with a sample session containing both types of records that are relevant to the primary window process. Lines whose numbers are not consecutive indicate that lines between those numbers are removed from the raw data because the expert felt they would be less relevant to the user's conscious activity, and hence were not included in the *.1s* file.

LineNo	Delta_t	ProcessName	PID	WinName	
Lineage					
LineNo	Delta_t	ProcessName	Status	CPU	Lineage
(523)	44.064	explorer	pid = 256		Program Manager
(532)	45.035		:launch32:189:explorer:257:explorer:256:		
(536)	45.035	telnet	pid = 303		<<16624>> - <<15265>>
(539)	47.138	telnet	b	0.010	:303:
(665)	128.465	explorer	pid = 256		Program Manager
(667)	128.465	telnet	pid = 303		<<close>> Telnet -
(675)	131.179	explorer	pid = 268		<<14537>>
(680)	134.354	explorer	c	0.841	Shut Down Windows
(682)	134.654	explorer	c	0.991	:launch32:189:explorer:257:explorer:268:
(688)	137.118	explorer	c	1.142	:explorer:268:
(692)	137.418	explorer	c	1.192	:explorer:268:
(694)	137.819	explorer	c	1.322	:explorer:268:
(696)	138.129	explorer	c	1.592	:explorer:268:
(709)	139.02	explorer	c	1.722	:explorer:268:
(712)	139.321	explorer	c	1.993	:explorer:268:
(716)	139.922	explorer	c	2.123	:explorer:268:
(720)	140.222	explorer	c	2.393	:explorer:268:
(728)	141.053	explorer	c	2.594	:explorer:268:
(730)	141.384	explorer	c	2.774	:explorer:268:
(732)	141.684	explorer	c	2.794	:explorer:268:
(737)	142.075	explorer	c	2.924	:explorer:268:
(739)	142.375	explorer	c	2.994	:explorer:268:
(749)	142.976	explorer	c	3.004	:explorer:268:

Figure 2: Contents of a file documenting a small session from User 1.

For ease of reading, the table contains three top rows serving as header rows to list attributes used to characterize each record; in the actual data the header rows are not present. The two shaded header rows correspond to the Window records, which are also shaded, and the one unshaded header row corresponds to Activity records, which are left unshaded. The attributes, LineNo, Delta_t, and Process Name in the first and third row are used to characterize both window and activity records. Attributes PID and WinName characterize only the Window records, and Status and CPU only characterize Activity records. The attribute Lineage in the second and third rows also characterizes records of both types. Because of space constraints, it appeared in the data files on the second line of Window records.

For learning user models, a number of additional *derived attributes* were created in addition to the original attributes described in Table 2. The additional attributes that are used by the AQ21 learning program are presented in Table 3 together with original attributes. For each attribute, the table specifies an AQ type of the attribute and the domain size.

No.	AQ21 name	Description	AQ type (size)	Possible Values
1*	Host	Host machine ID	nominal (21)	<i>host1, host19,...</i>
2*	Day	Day of Week	linear (7)	<i>Mon, Tue, ...</i>
3*	Hour	Time of day	nominal (24)	<i>01, 08, ...</i>
4+	session_start_sec	Discretized number of seconds from session's start	linear (6)	<i>lte5000, from5000to7000,..</i>
5+	process_name	Name of active process	nominal (181)	<i>acord32, fastboot, nxk-....</i>
6+	event_status	In the case of a window record, event_status is <i>n</i> when the user creates a new window, or <i>o</i> when the user returns to the previous window. In the case of an activity record, event_status is <i>b</i> if it is a newly created process, or <i>c</i> , if it is a continuation of an existing process.	nominal (4)	<i>o, n, b, c,</i>
7+	Proc_cpu_time	CPU time used by process	linear (8)	<i>lte40, from40to70,</i>
8	proc_inactive_time	Process inactive time	Integer	<i>0,1,2, ...</i>
9	proc_inactive_time_lf	Discretized natural log of inactive time	linear (6)	<i>lte0d7, from0d7to1,...</i>
10	proc_inactive_time_gt1min	Flag if process inactive for over one minute	nominal (2)	<i>lte60, gt60, ...</i>
11+	win_pid	Process ID of window (ignored)	Integer	<i>1,2,3, ...</i>
12+	win_title	Name of window (ignored)	Integer	
13	proc_cpu_time_in_win	CPU time accrued by process during current stay in window	linear (6)	<i>lte10, from10to20, ...</i>
14	proc_cpu_time_in_win_lf	Natural log of process CPU time accrued during stay in window	linear (8)	<i>lte1d6, from1d6to2d3</i>

15	win_time_elapsed	Total elapsed time in active window	linear (8)	<i>lte90, from90to200</i>
16	win_time_elapsed_lf	Natural log of total elapsed time in active window	linear (6)	<i>lte5d8, from5d8to6d4</i>
17	proc_cpu_to_win_elapsed_ratio	Ratio of <i>proc_cpu_time_in_win</i> to <i>win_time_elapsed</i>	linear (11)	<i>lte0d87, from0d87to0d97</i>
18	delta_time_new_window	Time between last two window creations	linear (5)	<i>lte10500, from10500to11000</i>
19	delta_time_new_window_lf	Natural log of time between last two window creations	linear (6)	<i>lte3d5, from3d5to4d5</i>
20	new_win_time_elapsed	Elapsed time from login to creation of window	linear (9)	<i>lte800, from800to5000</i>
21	new_win_time_elapsed_lf	Natural log of elapsed time from login to creation of window	linear (5)	<i>lte8, from8to9</i>
22	prot_words_chars	Number of characters in protected words in window title	Integer	0, 9, 26
23	prot_words__chars_to_total_chars_ratio	Percentage of characters in window title in protected words	linear (6)	<i>lte0d08, from0d08to0d14</i>
24	win_title_total_words	Number of words in window title	linear (7)	<i>lte9, from9to12</i>
25	win_title_total_to_prot_words_ratio	Percentage of words in window title that are protected	linear (9)	<i>lte0d05, from0d05to0d06</i>
26	proc_count_in_win	Number of process records covered by active window record	Integer	1, 2,3,...
27	proc_count_in_win_lf	Natural log of number of process records in active window record	linear (10)	<i>lte2d5, from2d5to3,..</i>
28	win_opened	Total number of windows created during session	linear (8)	<i>lte1, from1to2,..</i>
29	win_opened_lf	Natural log of total number of windows created during session	linear (8)	<i>lte1, from1to1d5,..</i>
30	win_title_prot_words	Number of protected words in window title	integer (8)	0, 1,2,3,...
31	win_title_sani_words	Number of unprotected words in window title.	linear (8)	<i>lte1, from1to5,..</i>

Table 3: Attributes used in our experiments.

The attribute types recognized in AQ learning (Section 4.1) are:

nominal – whose domains are discrete, unordered sets of possible values;

structured – whose domains are partially ordered sets or hierarchies of possible values

linear – whose domains are small or medium-sized discrete, totally ordered sets of values;

integer – whose domains are large domains, and are handled more efficiently than linear types;

continuous – whose domains are sets of real values.

In this study, the *continuous* and *structured* types were not used (they are not present in Tables 2 and 3), but they can be potentially very useful

In Table 3, attributes marked by an asterisk (“*”) were determined from the file name; attributes marked by a plus (“+”) were taken from the raw data. The remaining (unmarked) attributes were *derived* from the original data. Derived attributes 8-10 and 13-31 were suggested to us by a domain expert.

The number of values and values themselves for some attributes vary because they depend on the discretization used (in Figure 3 below, discretization scheme Dis-2 (Section 5.2.1) was applied). The values of the discretized attributes have been created so as to have precise meaning. For example, value *lte1d6* means “interval of numbers less than or equal to 1.6”, value *bt400and500* means “interval of numbers greater than 400 and less than or equal to 500”, value *gt7d4* means “interval of numbers greater than 7.4”.

As can be seen in Table 2, several attributes refer to “protected” or “unprotected” words in the window title. Protected words are defined as those that were not converted into numbers by a sanitization preprocessor. These words generally identified the program that was running. Unprotected words were sanitized into four-digit numbers, so as to remove the names of files, individuals, and anything else not directly related to the operating system. Throughout the data, each such word was converted to exactly one number, and each number represented exactly one word.

Figure 3 shows a part of the training data prepared for input into AQ21. The first section, enclosed within the symbols “(“ and “)” contains comments for the user and is ignored by the program. Two subsequent paragraphs present two training events for User 1, which are multi-attribute 5-grams. Each event starts with the user ID, then lists the episode ID, and then lists 31 concatenated 5-grams, each involving values of one attribute from Table 3. (Appendix B describes these attributes in more detail). For technical reasons, the order of attributes in the event is different from the order in Table 3. In different experiments, we used different attributes in events.

In the events presented in Figure 3, the five consecutive (in time) values of the first attribute are listed together, then the five values of the next attribute, and so forth, rather than listing values of all attributes for the same instance in time together. The reason for this is that such an order is more convenient for experimentation, because it allows one to easily ignore one or more attributes in a given run of the learning program.

```
(#
The time of creation is 2004-02-19 13:24
Input data file name is /home/shared/data/lus-njit/user1-host19-12_12_01-09_35_4
5.1s
This output data filename is njit-top10ts5-all-lb4.lus
There is no parameters file for AQ created
Lookback parameter is 4 for all attributes
User parameter is ua
Episode number is 281
#)
user1,281,host19,host19,host19,host19,host19,Wed,Wed,Wed,Wed,Wed,09,09,09,09,09,
lt300,lt300,lt300,lt300,lt300,lt300,msoffice,msoffice,msoffice,msoffice,msoffice,c,c,c
,c,n,lt60,lt60,lt60,lt60,gtel80,20,0,20,0,N/A,3.04452,0,3.04452,0,N/A,lte60,lte6
0,lte60,lte60,N/A,252,252,252,252,252,532,532,532,532,532,lt60,lt60,lt60,lt60,lt
60,0,0,0,0,0,lt300,lt300,lt300,lt300,lt300,lt300,4.39445,4.39445,4.39445,4.394
45,bt0and02,bt0and02,bt0and02,bt0and02,bt0and02,lt300,lt300,lt300,lt300,lt300,0,
0,0,0,0,lt300,lt300,lt300,lt300,lt300,4.41884,4.41884,4.41884,4.41884,4.41884,bt
20and40,bt20and40,bt20and40,bt20and40,bt20and40,bt08and1,bt08and1,bt08and1,bt08a
nd1,bt08and1,lt10,lt10,lt10,lt10,lt10,lt10,bt08and1,bt08and1,bt08and1,bt08and1,bt0
8and1,lt100,lt100,lt100,lt100,lt100,1.79176,1.79176,1.79176,1.79176,1.79176,lt20,lt
20,lt20,lt20,lt20,0.693147,0.693147,0.693147,0.693147,0.693147,4,4,4,4,4,lt20,lt
20,lt20,lt20,lt20,lt20

user1,281,host19,host19,host19,host19,host19,Wed,Wed,Wed,Wed,Wed,09,09,09,09,09,
lt300,lt300,lt300,lt300,lt300,lt300,msoffice,msoffice,msoffice,msoffice,msoffice,c,c,c
,c,c,lt60,lt60,lt60,lt60,lt60,1,20,0,20,0,0.693147,3.04452,0,3.04452,0,lte60,lte
60,lte60,lte60,lte60,252,252,252,252,252,532,532,532,532,532,lt60,lt60,lt60,lt60
,lt60,0,0,0,0,0,lt300,lt300,lt300,lt300,lt300,lt300,4.39445,4.39445,4.39445,4.39445,4.
39445,bt0and02,bt0and02,bt0and02,bt0and02,bt0and02,lt300,lt300,lt300,lt300,lt300
,0,0,0,0,0,lt300,lt300,lt300,lt300,lt300,4.41884,4.41884,4.41884,4.41884,4.41884
,bt20and40,bt20and40,bt20and40,bt20and40,bt20and40,bt08and1,bt08and1,bt08and1,bt
08and1,bt08and1,lt10,lt10,lt10,lt10,lt10,lt10,bt08and1,bt08and1,bt08and1,bt08and1,bt0
8and1,lt100,lt100,lt100,lt100,lt100,1.79176,1.79176,1.79176,1.79176,1.79176,lt20
,lt20,lt20,lt20,lt20,0.693147,0.693147,0.693147,0.693147,0.693147,4,4,4,4,4,lt20
,lt20,lt20,lt20,lt20,lt20

. . . . .
```

Figure 3: An example of input data for AQ21.

There was a large difference in the amount of data available for each of the 26 users, as is indicated in Table 1. One user’s data consisted of 287 sessions, and five other users provided but one session. Therefore, we concentrated in the current study on the ten users with the most sessions, and prepared training and testing datasets for them. Following the lead of previous research, the first ten sessions (based on time of recording) from each of ten users were selected as training episodes, the next five were selected as testing episodes, and the rest were ignored. Thus, we used only 15 sessions from each user. This was done so that our results can be compared with other results using the same data. These datasets were subsequently transformed into multi-attribute n -grams.

Table 4 lists numbers of *events*, defined here as multi-attribute $n \times k$ -grams, with $n = 4$ and the number of attributes k equal to 31. The attributes, original and derived, were extracted from the original (unfiltered) data recording 15 training and testing sessions of each user. In different experiments, some attributes were ignored in the events.

Table 5 lists the number of events in each training session for each user. As one can see, some sessions were very short (e.g., session 1 for User 1).

User	Number of Training Events	Number of Testing Events
1	1843	2815
2	13712	5657
3	195	22
4	35326	5133
5	10402	5983
7	7006	1467
8	6137	5424
12	10054	7464
19	10654	3584
25	24137	5748

Table 4: Training and testing event counts.

User \ Session	Number of Training Events by Session									
	1	2	3	4	5	6	7	8	9	10
1	1	17	78	49	76	11	500	66	17	1028
2	520	3103	1483	1693	1655	437	178	1010	1486	2147
3	68	3	18	4	2	87	3	7	2	1
4	4750	6806	2623	5044	3364	819	1411	2881	4473	3155
5	96	482	515	269	718	3819	1053	395	607	2448
7	3291	34	290	258	240	155	1094	71	1543	30
8	7	201	655	805	4	1072	587	481	430	1895
12	816	1095	965	1486	1635	388	844	729	548	1548
19	276	666	706	299	849	1800	5386	211	352	109
25	1133	2769	2935	1194	4750	1680	3230	2830	1723	1893

Table 5: Training event counts by user and session.

Another type of data used in some experiments (called window-based data) characterizes users' behavior with a lower degree of time granularity. In this case, the prepared data contained only events corresponding to window records. Activity records were processed solely to construct attributes that needed their information in the former type of records. The data itself looks very similar to what was presented in Figure 3, except values of the attributes related to Activity records are replaced by “N/A” symbols (*Not Applicable*).

Table 6 shows characteristics of the window-based training and testing datasets of the 10 users.

	Training Sessions											Testing Sessions						Tot
U#	1	2	3	4	5	6	7	8	9	10	Tot	1	2	3	4	5	Tot	
1	0	3	29	17	15	2	138	30	2	158	394	130	75	28	31	190	454	848
2	380	174	52	200	163	213	220	50	23	110	1585	107	53	59	315	50	594	2179
3	4	25	0	8	1	0	11	2	0	0	51	1	0	5	0	0	6	57
4	178	341	185	255	80	31	102	98	310	95	1675	63	135	113	37	66	414	2089
5	10	35	40	26	84	75	59	13	17	144	503	71	116	44	118	95	444	947
7	81	12	79	57	75	25	61	19	169	6	584	31	71	13	22	66	203	787
8	4	40	84	32	0	91	61	36	67	100	515	96	41	144	229	76	586	1101
12	63	78	125	182	199	37	98	101	63	127	1073	67	54	61	109	261	552	1625
19	4	53	97	22	95	78	277	19	17	7	669	10	10	45	83	45	193	862
25	76	221	378	89	495	153	221	220	30	109	1992	86	105	33	86	393	703	2695

Table 6: Charecteristics of the window-based data for the 10 users' training and testing sets

4 METHODS FOR LEARNING AND TESTING USER MODELS

This section describes briefly the algorithms developed for generating user models, and for applying the models to the datastreams in order to recognize users.

4.1 Learning User Models

The main engine for acquiring user models used in this study is AQ21, our newest rule learning system (Wojtusiak, 2004). Given a set of positive and negative examples of a concept, AQ21 generates sets of general attributional rules (Michalski, 2004) that describe positive examples of the concept and are optimized according to a multicriterion user-defined optimality measure. The measure is defined by the user in order to tailor the learning process to the problem at hand.

In the experiments, we used three optimality measures depending on types of descriptions being investigated. The first one generates most general descriptions (which are obtained by seeking the shortest rules that cover the maximal number of examples). The second one generates very specific rules (which are obtained by seeking the longest rules that cover the maximal number of examples). The third measure seeks the simplest descriptions (which are obtained by seeking rules with the minimal number of conditions in them).

In our study using multistate template user model, examples were multi-attribute $n \times k$ -grams, such as those shown in Figure 3. Positive examples characterized the behavior of the user whose model was being learned, and negative examples characterized the behavior of other users. The

negative examples provide a contrast set, that is, they act as constraints on the scope of generalization of the description of a user's behavior.

AQ21 can be run in three modes:

TF—*Theory formation* mode, in which the learned rulesets are complete and consistent with regard to all the training data (that is, they describe all positive examples and none of the negative examples). In other modes, AQ21 generates rules that may be partially incomplete and/or inconsistent, reflecting strong patterns in the data.

PD—*Pattern Discovery* mode, in which ruleset inconsistency and incompleteness are allowed, if they result in more optimal rulesets according to a given quality measure (Kaufman and Michalski, 1999; Michalski and Kaufman, 2001).

ATF—*Approximate Theory Formation* mode, in which rules are learned as if in TF mode, but after they are generated, they are optimized as in PD mode. The ATF mode may include an additional optimization step, called TRUNC, in which some rules are removed.

The central procedure implemented in the AQ21 program concerns generating a *star*. Given a positive example, called the *seed*, and set of negative examples, a star is a set of alternative generalizations of the seed that do not cover negative examples. AQ21 creates rulesets describing individual concepts by selecting from consecutively generated stars the “best” rule, until all concept examples are explained (covered). AQ21 is employed in two basic user profiling models, described in Sections 4.1.1 and 4.1.2 below. A third model, the Activity-based model, which plays a supporting role, also utilizes AQ21, and is discussed in Section 4.1.3.

AQ21 includes a number of new features designed and implemented for the purpose of the LUS project that were not available in previous AQ program implementations. Among the new features are the ability to learn the Prediction-Based Model, several new rule matching methods implemented in the ATEST module of AQ21, new variants of the EPIC algorithm for matching rulesets with testing episodes, new methods for attribute quality evaluation/selection, and the ability to perform data-driven constructive induction. The latter feature is still under development. It draws upon the results published in (Bloedorn and Michalski, 1998).

4.1.1 Multistate Template Model Representation

The development of the *multistate template* model representation is a continuation and an extension of our earlier work on learning user models from n -grams. In this representation, multigrams (nxk -grams) are extracted from a training datastream in order to create a training set for learning. The attributes used in the multigrams are selected as the most relevant ones for developing models of a given set of users. Given the training data sets of multigrams, a learning program (in this research, AQ21) creates attributional rulesets characterizing the behavior of each user. These rulesets are transformed into multistate templates, which provide an easy to interpret representation of user models.

Figure 4 shows an example of such a multistate template for User 4, which was derived from an attributional ruleset learned by AQ21 characterizing that user (Figure 5). The first condition in the condition part of the template in Figure 4 indicates in the third position in its 4-gram a characteristic time interval pattern for user 4, namely, the period between 11:00 and 14:59 (on

the 24 hour clock). The number 3393, called *positive absolute support* or *p-number* for this condition is the number of $4xk$ -grams (events) in the training data from user 4 that satisfy this time period constraint, and the number 9171, called *negative absolute support*, or *n-number* for this condition is the number of negative events (events in the training data from other users) that satisfy this condition. In cases such as this one, in which only one of the slots in the n -gram is filled, and other time instances can accept any values, we have for the ease of understandability represented such conditions in double angle brackets, with a number in parentheses indicating the position the constraint takes in the n -gram. In this case, the hour condition is in the third time instance of four.

```
[user=user4]
  <-- [hour = << 11..14 : 3393,9171 >> (3)]
    [process_name = < netscape,outlook,winword : 3904,18376;
      csrss,netscape,outlook,winword : 3909,18413;
      csrss,netscape,outlook,winword : 3909,18397;
      csrss,netscape,outlook,winword : 3909,18379 > ]
    [event_status = < c,o : 3997,22090; c,o : 3997,22113;
      c,o : 3997,22123; * > ]
    [proc_cpu_time_in_win_lf = < 0.3466..4.049 : 3611,12784; *; *;
      lt_3.916 : 3994,20119 > ]
    [win_time_elapsed_lf = << gt_3.337 : 3251,13713 >> (1)]
    [delta_time_new_window = << lt_1800 : 3985,21445 >> (1)]
    [delta_time_new_window_lf = << lt_7.748 : 3987,21518 >> (4)]
    [new_win_time_elapsed = << 300..18000 : 3954,16719 >> (4)]
    [prot_words_chars = << lt_20 : 3980,17938 >> (1)]
    [proc_count_in_win_lf = << gt_4.063 : 3060,7992 >> (1)]
    [win_opened_lf = << 1.498..2.636 : 3600,13531 >> (4)]

p=2419, n=0
```

Figure 4: A template identifying User 4 in the *multistate template* model.

```
[user=user4]
  <-- [hour-1=11..14 : 3393,9171]
    [process_name=csrss,netscape,outlook,winword : 3909,18379]
    [process_name-1=csrss,netscape,outlook,winword : 3909,18397]
    [process_name-2=csrss,netscape,outlook,winword : 3909,18413]
    [process_name-3=netscape,outlook,winword : 3904,18376]
    [event_status-1=c,o : 3997,22123]
    [event_status-2=c,o : 3997,22113]
    [event_status-3=c,o : 3997,22090]
    [proc_cpu_time_in_win_lf<=3.916 : 3994,20119]
    [proc_cpu_time_in_win_lf-3=0.3466..4.049 : 3611,12784]
    [win_time_elapsed_lf-3>=3.337 : 3251,13713]
    [delta_time_new_window-3<=1800 : 3985,21445]
    [delta_time_new_window_lf<=7.748 : 3987,21518]
    [new_win_time_elapsed=300..18000 : 3954,16719]
    [prot_words_chars-3<=20 : 3980,17938]
    [proc_count_in_win_lf-3>=4.063 : 3060,7992]
    [win_opened_lf=1.498..2.636 : 3600,13531]

p=2419, n=0
```

Figure 5: A rule identifying User 4 in the *multistate template* model.

The second condition consists of 4 subconditions specifying values of the process name attribute in the four consecutive time instances. The first subcondition states that in the first time instance the process can be netscape, outlook, or winword. This subcondition, on its own, was satisfied by 3904 events from user 4 and 18376 events from all other users. The second subcondition states that in the second time instance the process should be csrss, netscape, outlook or winword. The interpretation of the remaining subconditions is analogical.

An asterisk “*” means that at this time instance any value can be present. The symbols gt_ and lt_ in front of some values indicate that the value of the corresponding attribute should be “greater than” or “less than” the value following this sign. Ranges of values are denoted by placing “..” between the end values. The numbers p and n at the end of the rule respectively denote the absolute positive support of the whole rule (the total number of positive training examples satisfying this rule) and the absolute negative support (the total number of negative training examples satisfying this rule), respectively. Thus, this multistate template satisfies p=2419 positive events and n=0 negative events in the training set that consisted of 4000 positive events and 25173 negative events for user 4.

In the attributional rule in Figure 5 from which the above template was derived, no suffix to the attribute name denotes the current time instance, and -1, -2 and -3 denote the three previous time instances.

Two models have been developed for applying multistate templates (Sections 4.2.1 and 4.2.4). The first, EPIC-MT applies these rules by aggregating their performance against the individual events in an episode. The second, EPIC-RB, combines this approach with Bayesian reasoning.

4.1.2 Prediction-Based Model

The prediction-based model discovers sets of conditions in the data that are associated with a subsequent set of conditions, and represents them in the form of if-then rules. The prediction-based model traces its origins to the program SPARC (Michalski, Ko and Chen, 1987). SPARC views a list of events not as individual occurrences, but rather as a set of points within a sequence. Thus, SPARC can recognize that an element that is appropriate at one point in a sequence may be completely out of place in another.

SPARC uses three separate models to characterize sequences:

1. The *DNF Model*, which uses the AQ algorithm to characterize the elements present in the sequence.
2. The *Decomposition Model*, which generates rules of the form “if recent events in the sequence has certain characteristics, the subsequent event will have some given characteristics”. For example, “If the previous event was more than 300 seconds prior to the next one, the next event’s process will be *mail* or *compile*.”
3. The *Periodic Model*, which looks for characterizable repeating patterns within the sequence.

The prediction-based model uses Decomposition-type rules as a basis for characterizing sequences of a user’s processes. Given window size parameters *lookback* and *lookforward*, the model expresses patterns in the form: For the given user, if a set of conditions was observed

during the last lookback, we expect a set of conditions to occur during the next lookforward. For example, the rules shown in Figure 6 were learned with a lookback of 5 events and a lookforward of 1, and they describe conditions in which a subsequent netscape process may be expected from User 14. The first rule, for example, identifies the processes that can be observed two events before the present in order that according to the rule, we can expect netscape from User 14, and the third rule similarly specifies that a netscape process five events ago may signal another one now. The second rule indicates that how many “protected” words were just seen in the window title may affect the expectation of a netscape process.

It has to be noted that learning a prediction-based model for a given user takes into consideration only data for the user regardless what is observed for other users.

```
[User = 14]:
[process_name=netscape]
  ← [process_name-2=calc,netscape : 1267,63]
  ← [prot_words_chars-1=8,11,20,23..24,27,31..33,35 : 1188,24]
  ← [process_name-5=netscape : 1216,107]
```

Figure 6: Rules characterizing process_name=netscape for User 14 in the prediction-based model.

4.1.3 Activity-Based Models

The activity-based models treat a user session as a whole, and characterize it based on the proportions of the session involving different activities. The differences between the activity patterns are encapsulated using AQ21, which generates rules for user identification. Since these models by nature require long episodes, they are only intended as supportive models.

We have identified four varieties of activity-based models to explore: Activity-Value (A-V), Activity-Event (A-E), Value-Next Value (V-NV), and Event-Next Event (E-NE). In the simplest model, the A-V model, attribute values are counted up for each user session, and then represented as histograms. AQ21 then learns rules based on the frequency of certain values. For instance, Figure 7 shows two A-V-based rules for identifying User 2 based on the frequency of certain processes in the session record. The first rule, for example, indicates approximately that csrss must occur less than 2.4% of the time, explorer at least 1.1%, netscape between 2.1% and 86.7%, and photoed less than 3.2%.

```
User 2
  ← [prob_csrss<=0.0240495 : 11,117] [prob_explorer>=0.0110005 : 10,87]
    [prob_netscape=0.0211505..0.867499 : 11,28]
    [prob_photoed<=0.0323995 : 11,122]: p=10,n=13
  ← [prob_netscape>=0.0256505 : 11,33] [prob_powerpnt<=0.0713995 : 11,120]
    [prob_winword>=0.116501 : 3,19]: p=3,n=0
```

Figure 7: Rules characterizing User 2 in the activity-based model.

Similarly, the A-E model counts instances of events, where events may be defined as vectors of attribute values, generalized n -grams, etc. The results from each of these models can easily be viewed as histograms.

The V-NV model tallies the count of each value-next value pair in a session, and similarly, the E-NV model counts event-next event pairs, where event is defined as in the A-E model. These models can easily be visualized by two dimensional models in which the x -axis represents the current value or event, and the y -axis represents the next one, while the thickness of points on the plotted graph represent the frequency of the associated value-next value pair.

4.2 Testing and Application

The EPIC series of programs are used for classifying episodes of data. It runs on top of the ATEST program (Reinke, 1984) that matches attributional rulesets with individual events. Given an episode, EPIC generates a classification of the episode with associated degrees of match for each user profile. To generate those degrees of match, EPIC applies a three-step process:

1. Generate a degree of match between each event in the episode and each rule in the user profiles by using ATEST to match the rule against the event.
2. Generate a degree of match between each event in the episode and each user profile as a whole by aggregating the degrees of match generated in (1) between the event and the profile's individual rules.
3. Generate a degree of match between the episode and each user profile by aggregating the degrees of match generated in (2) between the user profile and the episode's individual events.

Once a degree of match between the episode and each user profile is calculated, EPIC makes its calculations based on *threshold* and *tolerance* parameters. All profiles that return degrees of match both above the threshold, and within the tolerance of the highest degree of match attained by the episode are returned as possible classifications.

The different versions of EPIC differ in the representation of the knowledge they receive, and how they may apply the three stages of episode matching.

4.2.1 EPIC-MT

The EPIC-MT program for classifying episodes using multistate template user models applies the three-step matching process described above.

The pseudocode in Figure 8 presents the actual EPIC-MT algorithm implemented in AQ21. The sections below present in detail the matching strategies used in EPIC-MT.

Selector-event matching (for matching rules' individual conditions)

- *Match-no match*, in which an event-rule pair scores 1 if the event satisfies the selector's condition, and 0 otherwise.

- *Flexible*, in which an event-rule scores 1 if the event satisfies the selector's condition, and V otherwise, where V is a number between 0 and 1 that depends on distance from the values in the selector and the value in the event.

```

For all testing events
  If event is from a new episode
    Add the episode to list of episodes
    Reset episode counters
  Match event against all models
  If degree of match is below threshold for all classes
    Classify the event as "Other" with degree of match equal to
      1 / number of classes
  Else
    Classify the event to all classes within tolerance of the best-
      matched class

For all episodes
  Compute degree of match of the episode to all classes as the average
    of the degrees of match of all events from the episode
  If the degree of match of the episode is below threshold for all
    classes the episode is classified as "other"

```

Figure 8: EPIC-MT algorithm

Rule-event matching (conjunctions of selectors)

- *Match-no match*, in which an event-rule pair scores 1 if the event satisfies all of the rule's conditions, and 0 otherwise.
- *Selector ratio*, in which an event-rule pair scores a number between 0 and 1, inclusive. Specifically, the assigned score is ratio of the number of conditions in the rule satisfied by the event to the total number of conditions in the rule.
- *Coverage ratio*, in which an event-rule pair scores 0 if the event does not satisfy the rule's conditions, and a number between 0 and 1, inclusive, otherwise. This number is equal to the ratio of the number of training events of the rule's consequent class (user) satisfying the rule's conditions to the total number of training events of the consequent class.
- *Minimum (only for use with flexible selector match)*, in which an event-rule pair score is defined as the minimum of the degrees of match for all selectors from the rule.
- *Weighted Minimum (only for use with flexible selector match)*, in which an event-rule pair score is defined as the degree of match of the selector that minimizes $(1 - (1 - S_i) * w_i)$, where S_i is the degree of match of selector i , and w_i a weight based on the selector confidence.
- *Average (only for use with flexible selector match)*, in which an event-rule pair score is defined as the average of the degrees of match for all selectors from the rule.
- *Weighted Average*, in which an event-rule pair score is defined as the average of the degrees of match for all selectors from the rule, weighted by the confidence of the selectors.

- *Product (only for use with flexible selector match)*, in which an event-rule pair score is defined as the product of the degrees of match for all selectors from the rule.

Model-event matching (disjunctions of rules)

- *Average*, in which an event-model pair score is defined as the average of the degrees of match for all rules from the model.
- *Probabilistic sum*, in which an model-rule pair score is defined as the probabilistic sum of the degrees of match for all rules from the model.
- *Maximum*, in which an event-model pair score is defined as the maximum of the degrees of match for all rules from the model.
- *Weighted Maximum*, in which event-model pair score is defined using the following formula:

$$\max_j (c_{ijk} \times t_{ij})$$

where c_{ijk} is the degree of match between event k and rule R_{ij} , t_{ij} is the weight of the rule R_{ij} defined using two possible measures: (i) ratio of number of positive examples satisfying the rule to the total number of positive examples, (ii) significance of the rule, defined as sum of significances of the positive examples covered by the rule.

- *Best-only*, in which an event-model pair score is defined as the degree of match of the best rule from the model, evaluated using user-defined criteria.

Model-episode matching

- *Average*, in which an episode-model pair score is defined as the average of the degrees of match for all events from the episode.
- *Count matches*, in which an episode-model score is defined as the ratio of the number of events whose degree of match is above a user-defined threshold to the total number of events in the episode.

4.2.2 EPIC-SDA

The EPIC-SDA (*Stop when Decisive Advantage*) is a modification of the EPIC-MT method for classifying episodes using multistate template user models. EPIC-SDA matches examples from an episode until the degree of match of one of the classes has a decisive advantage over the degrees of match of other classes. The algorithm is presented in the pseudocode in Figure 9.

The EPIC-SDA algorithm is in practice a very useful modification of the EPIC-MT algorithm since it can significantly reduce amount of data (total time of observing users) needed for recognition of users.

```

For all testing events
  If event is from a new episode
    Add the episode to list of episodes
    Reset episode counters
  Match event against all models
  If degree of match is below threshold for all classes
    Classify the event as "Other" with degree of match 1/number of classes
  Else
    Classify the event to all classes within tolerance
  Update degree of match of the episode for all classes.
  If number of tested events for the episode > episode domination
    threshold
    If  $DM(i)/DM(j) > \text{episode domination threshold}$ , where  $i$  and  $j$  are the
      highest and second highest-matched models respectively
    Skip remainder of events for this episode

For all episodes
  Compute degree of match of the episode to all classes as average of
    degrees of match of all not-skipped events from the episode
  If degree of match of the episode is below threshold for all classes the
    episode is classified as "other"

```

Figure 9: EPIC-SDA algorithm

4.2.3 *EPIC-P*

The program EPIC-P has been developed to compare sequences in the testing sessions with rules based on the predictive model. It works by counting the number of times subsequences in episodes satisfy each of the rules, and applying threshold/tolerance to these counts to generate classifications. Specifically, for each user model, for each event in the episode, EPIC-P determines if (1) any premise in that user's model is matched by the event, and if so, (2) if any of the rules whose premises were matched also have their predicted behavior matched. The model's degree of match will simply be the ratio of the total generated by (2) to the total generated by (1), provided that (1) exceeds a frequency threshold.

4.2.4 *EPIC-RB*

EPIC-RB provides a methodology for combining learned decision rules based on multistate templates with Bayesian inference methodology. It begins by applying ATEST to generate degrees of match between each user profile and each event in the episode, using any of the available methods described in Section 4.2.1. These values are then aggregated in order to generate probabilities for each user as follows:

1. Initially, the probabilities of any profiled user being responsible for the episode are assumed to be equal. These values are the initial prior probabilities $P(U_i)$. For each event e_j in the episode, Steps 2-3 are applied in order to update these probabilities.
2. Using ATEST, degree of match $DM(e_j, U_i)$ is calculated. In addition, $P(e_j|U_i)$ and $P(e_j)$ are calculated based on the training data, the latter weighted according to the current $P(U_i)$ probabilities.
3. Based on these values, new probabilities for each user based on the event are calculated. In standard Bayesian reasoning, we would apply the formula $P(U_i|e_j) = P(U_i) * P(e_j|U_i) / P(e_j)$,

but instead EPIC-B modifies the second term to take into account the rule-based knowledge. Specifically: the formula is modified to $P(U_i|e_j) = f[P(U_i), DM(e_j, U_i)] * P(e_j|U_i) / P(e_j)$, where f is a combination function, and values are normalized so as to sum to 1. We are exploring different combination functions, such as maximum and probabilistic sum.

4. As in EPIC-MT, EPIC-RB returns classifications based on threshold and tolerance parameters.

Table 7 illustrates the EPIC-RB process. In this example, there are four users under consideration, and two events in an episode are viewed. Here, f is the maximum function. The first event produces higher degrees of match and frequencies in the training data for the first two users, and these values are propagated through the max function. Meanwhile, the degrees of match are poor for the last two users, and so the max function returns their .25 prior probabilities. When Bayes' formula is applied to these values, and the probabilities are normalized, the four users have probabilities of .36, .5, .07 and .07, respectively.

The first event produces a much higher degree of match for user 4, and a somewhat lower one for user 2. When Bayes' formula is applied to the numbers associated with this event, the probabilities are adjusted accordingly, and now User 4 is the leading candidate by a significant margin.

	User 1	User 2	User 3	User 4
<i>Initial Probability $P_0(U_i)$</i>	.25	.25	.25	.25
$DM(e_1, U_i)$.5	.7	.1	.2
$P(e_1 U_i)$.05	.06	.02	.02
$\max[P(U_i), DM(e_1, U_i)]$.5	.7	.25	.25
$P(U_i e_1)$.67	.92	.13	.13
<i>renormalized $P(U_i e_1)$</i>	.36	.5	.07	.07
$DM(e_2, U_i)$.5	.4	.2	.65
$P(e_2 U_i)$.06	.04	.03	.07
$\max[P(U_i), DM(e_2, U_i)]$.5	.5	.2	.65
$P(U_i e_2)$.62	.41	.12	.93
<i>renormalized $P(U_i e_2)$</i>	.3	.2	.06	.45

Table 7: An example of EPIC-RB using a maximum combination function.

This illustrates that with a maximum function in use, this method is subject to rapid reactions to a high degree of match. This is analogous to our idea of distinguishing features that when observed can alone provide recognition with high confidence. Thus, EPIC-RB, like EPIC-SDA, should have the capability of making identification decisions based on very short episodes.

5 DATA PREPARATION AND SELECTION

5.1 Event Selection

Input datasets contain many events that are common to a large number of users. They are not useful in learning, and slow down the process; therefore, it is desirable that they be omitted. The algorithm presented here selects n -tuples that satisfy a selection criterion from a potentially very large list of events. Three selection criteria have been proposed:

1. Commonality and distinctiveness criterion, in which events are chosen based on these two measures, which are computed for every event.
2. Significance-based criterion, in which events are chosen according to a significance measure that is an aggregation of commonality and distinctiveness.
3. Commonality and uniqueness-among-users criterion, which selects frequent events for one user that are not very common among different users.

In the experiments thus far, only the significance-based criterion has been used. Its algorithm is described below.

The input to data selection consists of a set of attributes (to be considered as useful), selection criteria and parameters, as well as training and tuning input lists of events (L1, L2). In our experiments we have examined two ways in which L1 and L2 can be assigned. The first approach (Tuning based on Training and Testing data - TTT) sets L1 to the entire training dataset and L2 to the whole testing dataset. The second method (Tuning based on Split Training data - TST) uses only training data, which split into two parts having equal numbers of sessions; the first part becomes L1 and the second becomes L2. The output from this process consists of a selected list of events. The selection is performed as follows:

1. For each distinct (when projected on the attributes selected) event e , count the number of times it occurs in each class (i.e., for each user) in the input lists of events assigned to different users. From these counts, for every event e and user u , generate two measures, *commonality* and *distinctiveness*. Commonality is defined as:

$$comm_u^e = \frac{p_u^e}{P_u^e}$$

and distinctiveness is defined as:

$$dist_u^e = \frac{comm_u^e}{comm_u^e + n_u^e / N_u^e}$$

where p_u^e is the minimum number of occurrences of e in the sets $L1_u$ and $L2_u$ (events of user u from L1 and L2), P_u^e is the total number of events in $L1_u$, n_u^e represents the total number of times the template e occurs in other users' data, and N_u^e is the total number of events for other users.

2. Compute significance (denoted *sig*) of event e for every user, using the commonality and distinctiveness measures.

3. Select events with significances that fulfill the selection criteria.

The selection algorithm presented above has several variants.

Ratio n_u^e/N_u^e used in step (1) can be computed using one of the following three schemas (called negative schemas):

- n1: n_u^e is the number of occurrences of e in sessions of other users from L1; N_u^e is the total number of events of other users in L1.
- nsum: n_u^e is the number of occurrences of e in sessions of other users from $L1 \cup L2$; N_u^e is the total number of events of other users in $L1 \cup L2$.
- nmax: n_u^e is the maximum number of occurrences of e in sessions of other users from L1 and L2; N_u^e is the total number of events of other users in L1.

The significance that is computed in step (2) can be also defined in several different ways. The following eight definitions were used in the presented experiments:

- sig1: $sig = dist$
- sig2: $sig = comm * dist$
- sig3: $sig = \log(1000 * comm. + 1) * dist$
- sig4: $sig = \log(100 * comm. + 100) * dist$
- sig5: $sig = comm. ^{0.5} * dist$
- sig6: $sig = comm. ^{0.25} * dist$
- sig7: $sig = comm. ^{0.75} * dist$
- sig8: $sig = \max(\text{normalized } comm, \text{normalized } dist)$

There are also two possible selection criteria that can be used in step (3): significance-rank-based, by which a certain number of events with the highest significance are selected; and significance-value-based, by which events with significance greater than a given threshold are selected.

We have also analyzed filtering methods based on the union of events having either high commonality or high distinctiveness. This method uses two rankings of events corresponding to commonality and distinctiveness and then finds the events that belong to the top of either of the rankings. The “top” ranks are established based on the input parameter specifying a percentage threshold. This method has two variants based on how this threshold is applied.

In the first variant (called *uni1*) the range and the maximum of the values from a given rank are computed and the user-specified tolerance threshold T is applied, selecting events that have commonality or distinctiveness not lower than T percent of the range from the maximum value.

The second form of this method (*uni2*) applies the threshold T to the list of numbers representing positions of the values of commonality or distinctiveness in their ranks. For example, if there are 200 distinct values of commonality (or distinctiveness) among the events and $T = 23\%$, then only events that have corresponding value of commonality (or distinctiveness) ranked between 1 and 46 will be selected.

The filtering process described above uses a conjunctive form of attributes. A disjunctive method can also be used to compute distinct significance measures for every attribute, and select events that contain at least one attribute with significance that fulfills the selection criterion.

5.2 Attribute Discretization

This section describes methods used to determine useful discretizations of large numeric domains. Initial experiments used very coarse hand-discretization encoded into the data preparation program; now the data preparation program reads the discretization scheme from the input file (which contains discretization points for selected attributes) and automatically applies this scheme when generating data. Two discretization schemes *dis-1* and *dis-2* were created based on charts of distributions of the values of the attributes, and discretizations were also evaluated using the PROMISE method (Baim, 1982; Kaufman, 1997), and Gain Ratio method (Quinlan, 1993) implemented in AQ21. Each distribution chart indicates how frequently the values of a given attribute occur in the selected target data (the first 15 sessions) for all users. The charts are used to determine points on the attribute's value scale that would best separate the users. This may be difficult manually, since some attributes have thousands of values.

In addition to manual discretization methods, an automatic algorithm has been implemented and applied to the LUS data, based on the ChiMerge method of incremental discretization (Kerber, 1992). Details of this algorithm are presented in Section 5.2.2.

5.2.1 Manual Discretization

Figure 10 presents an example of a distribution chart for the attribute *proc_cpu_time_in_win_lf*, indicating the natural log of process CPU time accrued during the current stay in the current window. Each curve represents a different user.

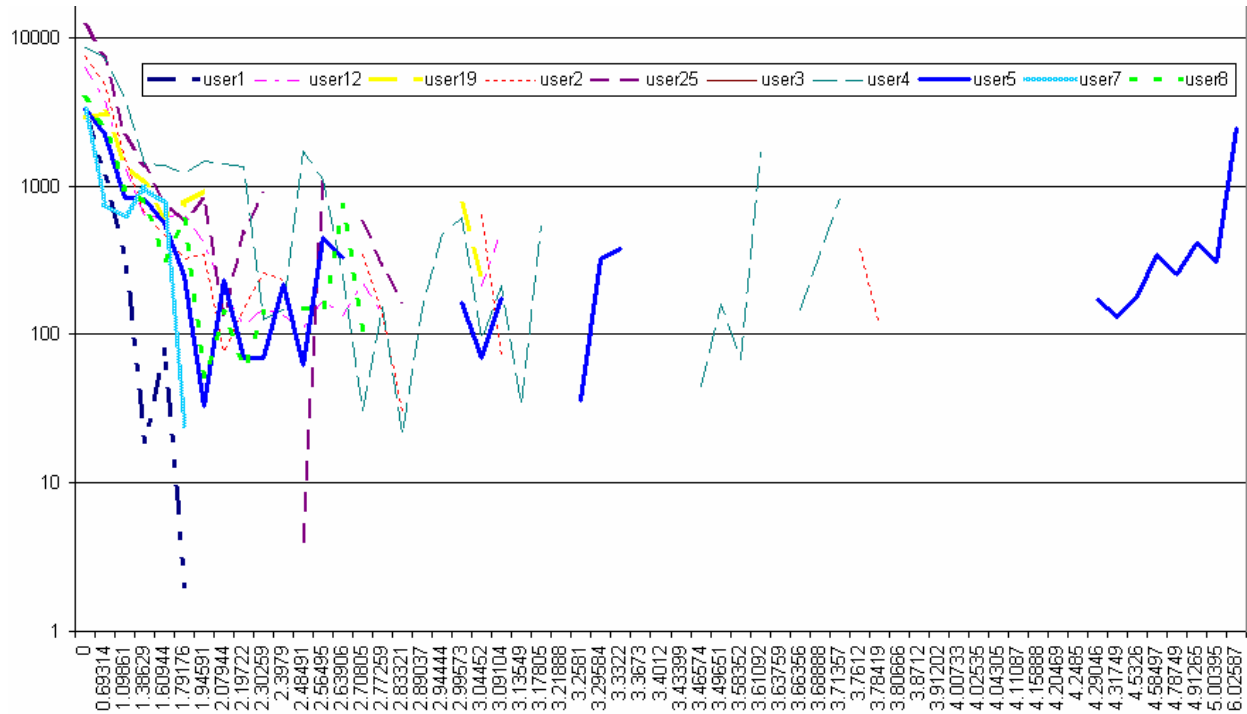


Figure 10: Frequency distribution of the values of the attribute *proc_cpu_time_in_win_lf* for each of ten users (logarithmic scale)

When needed, the data preparation program also creates corresponding AQ21 input parameter files for both multistate template and prediction-based models of user behavior. In these input files, tables of attribute domains, among which are the discretized domains, are created. The manual discretization scheme *Dis-2* is a coarser (fewer intervals) discretization than *Dis-1* as presented in Table 8 and Table 9.

Attribute	Discretization points
host	<i>not discretized</i>
day	<i>not discretized</i>
hour	<i>not discretized</i>
session_start_sec	320,1230,1700,2100,4000,5000,5400,6400,8200,12000,13700,14000,15000,16000,17600,18100,18400,19100,20000,21200,22100,22400,23700,25500,28900,30200,35300
process_name	<i>not discretized</i>
event_status	<i>not discretized</i>
proc_cpu_time	60,90,100,120,130,140,164,183,190,207,255,320,350,400
proc_inactive_time	<i>not discretized</i>
proc_inactive_time_lf	0.7,1.4,1.6,2.77,2.83,2.99,3.1,3.4,3.46
proc_inactive_time_gt_lmin	<i>not discretized</i>
win_pid	<i>not discretized</i>
win_title	<i>not discretized</i>
proc_cpu_time_in_win	13,16,20,24,32,42,55,60,70
proc_cpu_time_in_win_lf	0.69,1.1,1.38,1.79,2.08,2.3,2.63,2.83,3.1,3.25,3.45,3.65,3.75,4.25
win_time_elapsed	75,90,125,140,165,200,230,240,250,265,280,400,420,600,690,970,1500,1600,1800,2500,3500,6000,10000
win_time_elapsed_lf	5.8,6,6.05,6.4,6.6,6.8,7.3,7.4,7.5,8.2,8.4,8.9,9.1
proc_cpu_to_win_elapsed_ratio	0.87,0.97,1.3,4,5,6,7,20,500,1500
delta_time_new_window	10400,10700,11500,12500,13500,14300,15500,17500,19000,23000,24000,50000
delta_time_new_window_lf	1.3,3.3,3.5,4.2,4.4,5,5.5,6.5,7,7.5,8.5,9,9.5
new_win_time_elapsed	500,1500,2000,2500,3500,4500,5500,7500,8500,13500,14000,15000,18000,18500,20000,20500,21000,21500,25000,26000,29000,30500
new_win_time_elapsed_lf	6.5,7,7.9,8.6,8.9,9.5,9.6,9.9,10,10.3
prot_words_chars	<i>not discretized</i>
prot_words_chars_to_total_chars_ratio	0.09,0.13,0.21,0.215,0.234,0.3,0.4,0.46,0.53,0.62,0.65
win_title_total_words	<i>not discretized</i>
win_title_total_to_prot_words_ratio	0.67,1.11,1.74,2.20,2.25,2.85,2.9,3.6,3.7,0.4,0.44,0.45,0.6,0.65,0.71,0.9
proc_count_in_win	<i>not discretized</i>
proc_count_in_win_lf	0.69,1.3,1.6,2.3,2.77,3.73,4,4.15,4.24,4.3,4.35,4.7,4.8,4.85,5,6,7
win_opened	1,2,7,14,18,21,28
win_opened_lf	1,1.5,2.3,2.64,2.95,3.1,3.35
win_title_prot_words	<i>not discretized</i>
win_title_sani_words	<i>not discretized</i>

Table 8: Discretization scheme *Dis-1*

Attribute	Discretization points
host	<i>not discretized</i>
day	<i>not discretized</i>
hour	<i>not discretized</i>
session_start_sec	5000,7000,12000,19000,30000
process_name	<i>not discretized</i>
event_status	<i>not discretized</i>
Proc_cpu_time	40,70,100,130,180,220,350
proc_inactive_time	1,3,10
proc_inactive_time_lf	0.7,1,1.4,1.6,2
proc_inactive_time_gtlmin	<i>not discretized</i>
win_pid	<i>not discretized</i>
win_title	<i>not discretized</i>
proc_cpu_time_in_win	10,20,30,40,70
proc_cpu_time_in_win_lf	1.5,2.4,2.8,3.2,3.7,4
win_time_elapsed	200,350,500,650,1000,2000
win_time_elapsed_lf	5.8,6.4,7,7.3,7.4
proc_cpu_to_win_elapsed_ratio	1,1.4,5,10,100
delta_time_new_window	10500,11000,13000,24000,30000
delta_time_new_window_lf	3.5,4.5,5.5,6.5,7
new_win_time_elapsed	800,5000,5500,9000,10000,13000,15000,20000
new_win_time_elapsed_lf	8,9,9.5,10
prot_words_chars	5,15,25
prot_words__chars_to_total_cha rs_ratio	0.08,0.14,0.2,0.3,0.4
win_title_total_words	5,15
win_title_total_to_prot_words_ ratio	0.06,0.15,0.4
proc_count_in_win	20
proc_count_in_win_lf	2,3,4,5
win_opened	16,28
win_opened_lf	2.7,3.3
win_title_prot_words	<i>not discretized</i>
win_title_sani_words	10,15

Table 9: Discretization scheme *Dis-2*

The goal of the experiments with different discretization schemes is to create one with fewer discretization points but a better PROMISE evaluation. The PROMISE evaluations for the Dis-1 and Dis-2 schemes are presented in Table 10 and Table 11. The first column indicates the attribute, and the second column indicates the aggregate PROMISE measure considering each user as a separate decision class. The next ten columns indicate the attributes' PROMISE values for distinguishing one user (indicated by the column header) from the set of other users as a whole. The final column indicates the maximum PROMISE value from those ten columns.

The PROMISE values for individual users are useful in rule-based representations because our idea of a “good rule” tends to be one that is simple and understandable, and accurately provides a particular decision when it fires. Thus, if an attribute is useful for distinguishing one user, even if it provides little help in distinguishing among the other users, it may be the basis for a very useful rule, whose consequent is the one user it distinguishes well. In some non-rule-based representations, such as decision trees, it is more beneficial to have attributes that lead to quick decisions, even if they alone have little distinguishing power.

1		all	user1	user12	user19	user2	user25	user3	user4	user5	user7	user8	max
2	day	0.404093	0	0	0	0	0	0	0.3526	0	0	0.9984	0.9984
3	delta_time_new_window	0.61642	0.5853	0.7648	0	0.8649	0.5463	1	0	0.4105	0.3571	0.4286	1
4	delta_time_new_window_if	0.309149	0	0.3837	0	0	0	0	0.4806	0	0	0	0.4806
5	event_status	0.197723	0	0	0	0	0	0	0	0	0	0	0
6	host	0.998414	1	1	1	1	1	1	1	1	1	0.9897	1
7	hour	0.556284	0	0.3742	0.7309	0.4275	1	1	0.4107	0	0.8208	0	1
8	new_win_time_elapsed	0.366135	0	0.4775	0.5819	0	0.3986	0	0.3975	0.5822	0	0	0.5822
9	new_win_time_elapsed_if	0.322957	0	0.3556	0.4914	0	0	0	0.3899	0.3983	0	0	0.4914
10	proc_count_in_win	1	1	1	1	1	1	1	1	1	1	1	1
11	proc_count_in_win_if	0.284352	0	0	0	0	0.3612	0	0.5646	0	0	0	0.5646
12	proc_cpu_time	0.488095	0	0	0	0	0	0	0.5681	0.4302	0	0.48	0.5681
13	proc_cpu_time_in_win	0.512053	0	0	0.4341	0	0.3964	0	0.563	0.9082	0	0	0.9082
14	proc_cpu_time_in_win_if	0.427682	0	0	0	0	0	0	0.5164	0.8203	0.3788	0	0.8203
15	proc_cpu_to_win_elapsed_ratio	0.532523	1	0	0	1	0.597	0	0.3553	0.9375	0	0	1
16	proc_inactive_time	1	1	1	1	1	1	1	1	1	1	1	1
17	proc_inactive_time_gt1min	0.221149	0	0	0	0	0	0	0	0	0	0	0
18	proc_inactive_time_if	0.233013	0	0	0	0	0	0	0	0	0	0	0
19	process_name	0.756383	0.8688	0.9663	0.7893	0.8232	0.9549	0.8762	0.4405	0.9547	0.6874	1	1
20	prot_words_chars_to_total_chars_ratio	0.399101	0	0	0	0	0.4749	0	0.5057	0	0	0	0.5057
21	prot_words_chars	1	1	1	1	1	1	1	1	1	1	1	1
22	session_start_sec	0.377795	0.3524	0.4641	0.6405	0.3642	0.4714	0	0.4525	0.3859	0	0.4199	0.6405
23	user	1											0
24	win_opened	0.567757	0	0.4325	0	0.9331	0.72	0	0	0	0	0	0.9331
25	win_opened_if	0.53571	0	0.4325	0	0.8927	0.6561	0	0	0	0	0	0.8927
26	win_pid	1	1	1	1	1	1	1	1	1	1	1	1
27	win_time_elapsed	0.368485	0	0.3414	0.4392	0.4023	0.3361	0	0.4978	0.4513	0	0	0.4978
28	win_time_elapsed_if	0.386173	0	0	0.4076	0	0.3543	0	0.4569	0.451	0	0	0.4569
29	win_title	1	1	1	1	1	1	1	1	1	1	1	1
30	win_title_prot_words	0.511738	0	0	0	0	0.6723	0	0.3679	0	0	0	0.6723
31	win_title_sani_words	0.459106	0.7132	1	0.4013	0.5126	0.6786	0	0.3623	0	0	0.4986	1
32	win_title_total_to_prot_words_ratio	0.538086	0	0	0	0.7455	0.6895	0	0	0	0	0	0.7455
33	win_title_total_words	1	1	1	1	1	1	1	1	1	1	1	1

Table 10: Results of PROMISE attribute evaluation for the Dis-1 scheme.

1	attribute	all	user1	user12	user19	user2	user25	user3	user4	user5	user7	user8	max
2	day	0.404093	0	0	0	0	0	0	0.3526	0	0	0.9984	0.9984
3	delta_time_new_window	0.535649	0.3804	0.9268	0	0	0.3924	1	0	0	0	0	1
4	delta_time_new_window_if	0.306921	0	0	0	0	0	0	0.6235	0	0	0	0.6235
5	event_status	0.197723	0	0	0	0	0	0	0	0	0	0	0
6	host	0.998414	1	1	1	1	1	1	1	1	1	0.9897	1
7	hour	0.556284	0	0.3742	0.7309	0.4275	1	1	0.4107	0	0.8208	0	1
8	new_win_time_elapsed	0.335992	0	0	0	0	0	0	0.542	0.3461	0	0	0.542
9	new_win_time_elapsed_if	0.267801	0	0	0	0	0	0	0.4438	0	0	0	0.4438
10	proc_count_in_win	0.242969	0	0	0	0	0	0	0	0	0	0	0
11	proc_count_in_win_if	0.24752	0	0	0	0	0	0	0	0	0	0	0
12	proc_cpu_time	0.401348	0	0	0	0	0	0	0.5738	0.3618	0	0	0.5738
13	proc_cpu_time_in_win	0.461109	0	0	0	0	0	0	0.5379	0.8203	0	0	0.8203
14	proc_cpu_time_in_win_if	0.447389	0	0	0	0	0	0	0.5731	0.6298	0	0	0.6298
15	proc_cpu_to_win_elapsed_ratio	0.348415	0	0	0	0.75	0.45	0	0	0	0	0	0.75
16	proc_inactive_time	0.254149	0	0	0	0	0	0	0	0	0	0	0
17	proc_inactive_time_gt1min	0.221149	0	0	0	0	0	0	0	0	0	0	0
18	proc_inactive_time_if	0.253995	0	0	0	0	0	0	0	0	0	0	0
19	process_name	0.756383	0.8688	0.9663	0.7893	0.8232	0.9549	0.8762	0.4405	0.9547	0.6874	1	1
20	prot_words_chars_to_total_chars_ratio	0.337098	0	0	0	0	0	0	0.4092	0	0	0	0.4092
21	prot_words_chars	0.431889	0	0	0	0	0.5409	0	0.3362	0	0	0	0.5409
22	session_start_sec	0.35748	0	0.342	0.587	0	0	0	0.3983	0	0	0	0.587
23	user	1											0
24	win_opened	0.623385	0	0	0	0.6864	0.6187	0	0	0	0	0	0.6864
25	win_opened_if	0.447206	0	0	0	0.35	0.5388	0	0	0	0	0	0.5388
26	win_pid	1	1	1	1	1	1	1	1	1	1	1	1
27	win_time_elapsed	0.278555	0	0	0	0	0	0	0.3731	0	0	0	0.3731
28	win_time_elapsed_if	0.366861	0	0	0	0	0.34	0	0.4905	0.4546	0	0	0.4905
29	win_title	1	1	1	1	1	1	1	1	1	1	1	1
30	win_title_prot_words	0.511738	0	0	0	0	0.6723	0	0.3679	0	0	0	0.6723
31	win_title_sani_words	0.29302	0	0	0	0	0.3777	0	0	0	0	0	0.3777
32	win_title_total_to_prot_words_ratio	0.331588	0	0	0	0	0	0	0.346	0	0	0	0.346
33	win_title_total_words	0.347303	0	0	0	0	0.514	0	0	0	0	0	0.514

Table 11: Results of PROMISE attribute evaluation for the Dis-2 scheme.

5.2.2 Automatic Discretization

As in manual discretization described above, automatic discretization combines a discretization algorithm with the *computation* of a discretized attributes' quality measure, and a measure of similarity/dissimilarity between users' sessions. The discretization algorithm used here is ChiMerge (Kerber, 1992; Cichosz, 2000), which is described below. Also as in manual discretization, we use PROMISE and Gain Ratio as attribute quality measures.

The ChiMerge algorithm implemented in the AQ21 system is described by the pseudocode in Figure 11:

```

Intervals = {all values from data}
While number_of_intervals > threshold
  Compute values of  $\chi^2$  for all adjacent pairs of intervals
  Select and merge the pair of intervals with the lowest value of  $\chi^2$ 
  Add joint interval to list of intervals, replacing the intervals that were
  joined

```

Figure 11: ChiMerge algorithm implemented in AQ21

In the AQ21 implementation, the value of χ^2 is computed using the formula shown below.

$$X_{I_1, I_2}^2 = \sum_c \frac{(\#E_{I_1}^c - \#E_{I_1} \frac{\#E_{I_1 \cup I_2}^c}{\#E_{I_1 \cup I_2}})^2}{\#E_{I_1}^c \frac{\#E_{I_1 \cup I_2}^c}{\#E_{I_1 \cup I_2}}} + \sum_c \frac{(\#E_{I_2}^c - \#E_{I_2} \frac{\#E_{I_1 \cup I_2}^c}{\#E_{I_1 \cup I_2}})^2}{\#E_{I_2}^c \frac{\#E_{I_1 \cup I_2}^c}{\#E_{I_1 \cup I_2}}}$$

where: $\#E$ is total number of training examples, $\#E^c$ is number of examples in class c , $\#E_i$ is number of examples in interval i , $\#E_i^c$ is number of examples from class c in the interval i .

In our experiments, we computed ChiMerge for between 3 and 9 intervals, inclusive, per attribute in order to determine the smallest number of intervals with the highest value of PROMISE. The table with the promise values for 3 intervals per attribute is presented in Table 12 and Figure 12 below. The highlighted attributes have high values (above 0.5 or close to it) of overall PROMISE and max PROMISE for users. The values were computed for all data available from four selected users: Users 2, 5, 7, and 25.

Attribute	all	user2	user25	user5	user7	max	intervals
day	0.484616	0	0.392	0	0.4696	0.392	
delta_time_new_window	0.500093	0	0.4042	0	0.4865	0.4042	0, 3.5, 11.5, 32918
delta_time_new_window_if	0.500093	0	0.4042	0	0.4865	0.4042	0, 1.49787, 2.52493, 10.4018
event_status	0.515245	0	0.3864	0	0.5154	0.3864	
host	1	1	1	1	1	1	
hour	0.781307	0.344	0.8207	0	0.6649	0.8207	
new_win_time_elapsed	0.505749	0	0.3862	0	0.4875	0.3862	0, 7380.5, 21526.5, 45572
new_win_time_elapsed_if	0.498927	0	0.368	0	0.4807	0.368	0, 8.90673, 9.91238, 10.7271
proc_count_in_win	0.505842	0	0.4185	0	0.4857	0.4185	0, 15.5, 209.5, 4115
proc_count_in_win_if	0.505842	0	0.4185	0	0.4857	0.4185	0, 2.8029, 5.34948, 8.32264
proc_cpu_time	0.439918	0.3428	0.3637	0.4174	0.4882	0.4174	0, 159.5, 263.5, 429
proc_cpu_time_in_win	0.481349	0	0.415	0	0.4813	0.415	-1, 0.5, 7.5, 427
proc_cpu_time_in_win_if	0.481349	0	0.415	0	0.4813	0.415	0, 0.346573, 2.13833, 6.05912
proc_cpu_to_win_elapsed_ratio	0.434379	0.5	0.4185	0	0.4508	0.5	-1000, -0.495, 0.005, 34000
proc_inactive_time	0.477693	0	0.3686	0	0.5623	0.3686	0, 661.5, 2443.5, 24356
proc_inactive_time_gt1min	0.514954	0	0.3695	0	0.515	0.3695	
proc_inactive_time_if	0.477693	0	0.3686	0	0.5623	0.3686	0, 6.49602, 7.80159, 10.1006
process_name	0.848455	0.8509	0.8927	0.8117	0.8683	0.8927	
prot_words__chars_to_total_chars_ratio	0.553054	0	0.4689	0	0.5327	0.4689	0, 0.41898, 0.931034, 1
prot_words_chars	0.607497	0	0.5036	0	0.5789	0.5036	0, 13.5, 24, 52
session_start_sec	0.472753	0	0.3738	0	0.4568	0.3738	0, 3857.5, 15158.5, 45572
win_opened	0.463596	0	0.3985	0	0.4604	0.3985	1, 5.5, 8.5, 40
win_opened_if	0.463596	0	0.3985	0	0.4604	0.3985	0.693147, 1.86684, 2.24991, 3.71357
win_pid	0.877805	1	1	1	1	1	
win_time_elapsed	0.564891	0	0.4475	0	0.5228	0.4475	0, 256.5, 1713.5, 25716
win_time_elapsed_if	0.564891	0	0.4475	0	0.5228	0.4475	0, 5.55102, 7.44688, 10.1549
win_title	0.782032	1	1	1	1	1	
win_title_prot_words	0.547427	0	0.5945	0	0.4644	0.5945	0, 1.5, 2.5, 6
win_title_sani_words	0.499192	0	0.3795	0	0.4852	0.3795	0, 0.5, 4.5, 31
win_title_total_to_prot_words_ratio	0.499798	0	0.3987	0	0.48	0.3987	0, 0.244048, 0.928571, 1
win_title_total_words	0.479948	0	0.4276	0	0.4782	0.4276	1, 2.5, 4.5, 32

Table 12: Values of Promise for automatically discretized intervals for users 2, 5, 7, and 25 for all available data.

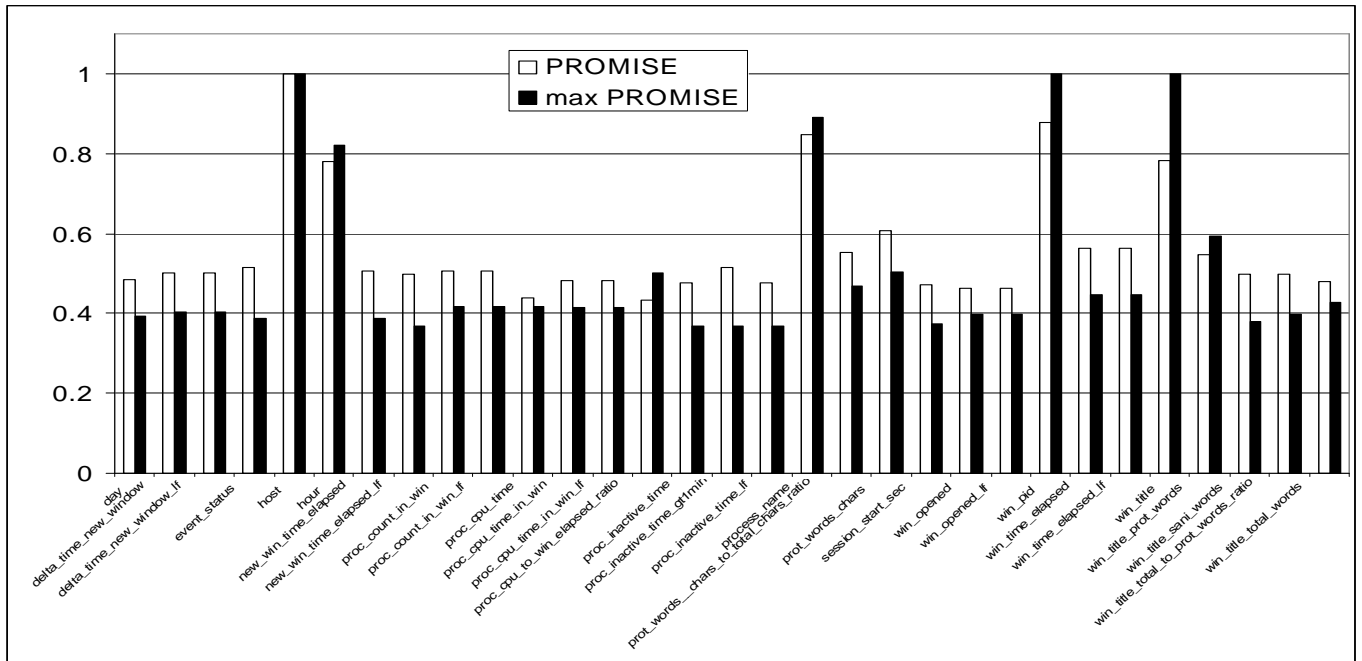


Figure 12: Values of Promise for automatically discretized intervals for users 2, 5, 7, and 25 for all available data.

The tables below present quality tables for different discretization schemas. The first column shows the discretization method, either ChiMerge with 3, 4, ..., 9 target intervals, or one of the two manual discretization schemas (Dis-1 and Dis-2). The second and third columns represent values of overall PROMISE and maximum of PROMISE for users for a given discretizations and attributes. The last column presents the number of users with a non-zero value of PROMISE for the given discretization. The values in the tables below were computed for 10 selected users (as indicated in Table 1) and 10 sessions per user (training data). Discretizations presented in bold and shaded were defined as the Dis-3 schema, and were used for further experiments with similarity and significance computation. For each attribute we have selected one discretization scheme for further investigation.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.33	0.37	1
4	0.35	0.45	1
5	0.33	0.45	1
6	0.33	0.41	1
7	0.33	0.41	1
8	0.34	0.48	2
9	0.34	0.48	2
Dis-1 (13)	0.62	1	8
Dis-2 (6)	0.53	1	4

Table 13: Discretization quality for delta_time_new_window attribute

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.33	0.37	1
4	0.35	0.45	1
5	0.33	0.45	1
6	0.33	0.41	1
7	0.33	0.41	1
8	0.34	0.48	2
9	0.34	0.48	2
Dis-1 (14)	0.31	0.48	2
Dis-2 (6)	0.31	0.62	1

Table 14: Discretization quality for delta_time_new_window_If attribute

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.31	0.45	1
4	0.29	0.45	1
5	0.31	0.4	1
6	0.31	0.4	1
7	0.33	0.41	1
8	0.31	0.41	1
9	0.34	0.54	2
Dis-1 (23)	0.37	0.58	5
Dis-2 (9)	0.33	0.54	2

Table 15: Discretization quality for new_win_time_elapsed attribute

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.3	0.47	1
4	0.29	0.47	1
5	0.32	0.47	1
6	0.32	0.47	1
7	0.33	0.47	2
8	0.32	0.47	2
9	0.34	0.47	2
Dis-1 (11)	0.32	0.49	4
Dis-2 (5)	0.27	0.44	1

Table 16: Discretization quality for new_win_time_elapsed_If attribute

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.31	0.39	1
4	0.29	0.39	1
5	0.29	0.39	1
6	0.34	0.67	1
7	0.33	0.67	1
8	0.33	0.67	1
9	0.33	0.67	1
Dis-1			
Dis-2 (2)	0.24	0	0

Table 17: Discretization quality for
proc_count_in_win attribute.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.31	0.39	1
4	0.29	0.39	1
5	0.29	0.39	1
6	0.34	0.67	1
7	0.33	0.67	1
8	0.33	0.67	1
9	0.33	0.67	1
Dis-1 (17)	0.28	0.56	2
Dis-2 (5)	0.24	0	0

Table 18: Discretization quality for
proc_count_in_win_lf attribute.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.49	0.60	2
4	0.54	0.63	2
5	0.5	0.63	2
6	0.52	0.63	2
7	0.54	0.55	2
8	0.53	0.58	2
9	0.54	0.6	2
Dis-1 (15)	0.49	0.56	3
Dis-2 (8)	0.4	0.57	2

Table 19: Discretization quality for
proc_cpu_time attribute.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.33	0.42	1
4	0.34	0.38	2
5	0.36	0.43	2
6	0.36	0.4	2
7	0.36	0.4	2
8	0.35	0.39	2
9	0.39	0.5	3
Dis-1 (11)	0.53	1	5
Dis-2 (6)	0.35	0.75	2

Table 20: Discretization quality for
proc_cpu_to_winelapsed_ratio attribute.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.31	0.40	1
4	0.33	0.4	1
5	0.32	0.4	1
6	0.33	0.44	1
7	0.44	0.77	2
8	0.42	0.77	2
9	0.43	0.77	2
Dis-1 (10)	0.51	0.9	4
Dis-2 (6)	0.46	0.82	2

Table 21: Discretization quality for
proc_cpu_time_in_win attribute.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.31	0.40	1
4	0.33	0.4	1
5	0.32	0.4	1
6	0.33	0.44	1
7	0.44	0.77	2
8	0.42	0.77	2
9	0.43	0.77	2
Dis-1 (14)	0.43	0.82	3
Dis-2 (7)	0.45	0.63	2

Table 22: Discretization quality for
proc_cpu_time_in_win_lf attribute.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.22	0	0
4	0.24	0	0
5	0.23	0	0
6	0.24	0	0
7	0.24	0	0
8	0.23	0	0
9	0.23	0	0
Dis-1			
Dis-2 (4)	0.25	0	0

Table 23: Discretization quality for
proc_inactive_time attribute.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.22	0	0
4	0.24	0	0
5	0.23	0	0
6	0.24	0	0
7	0.24	0	0
8	0.23	0	0
9	0.23	0	0
Dis-1 (10)	0.23	0	0
Dis-2 (6)	0.25	0	0

Table 24: Discretization quality for
proc_inactive_time_If attribute.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.29	0.00	0
4	0.3	0.35	1
5	4	0.51	2
6	0.41	0.68	2
7	0.43	0.68	2
8	0.41	0.68	2
9	0.41	0.68	2
Dis-1 (12)	0.4	0.5	2
Dis-2 (6)	0.38	0.4	1

Table 25: Discretization quality for
prot_words__chars_to_total_chars_ratio attribute.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.38	0.47	2
4	0.47	0.92	2
5	0.53	0.85	2
6	0.49	0.85	2
7	0.45	0.85	2
8	0.43	0.85	2
9	0.44	0.85	3
Dis-1			
Dis-2 (4)	0.43	0.54	2

Table 26: Discretization quality for
prot_words_chars attribute.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.3	0.35	2
4	0.32	0.4	2
5	0.3	0.4	2
6	0.34	0.46	2
7	0.32	0.46	2
8	0.34	0.46	2
9	0.33	0.46	2
Dis-1 (8)	0.57	0.93	3
Dis-2 (3)	0.62	0.69	2

Table 27: Discretization quality for
win_opened attribute.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.3	0.35	2
4	0.32	0.4	2
5	0.3	0.4	2
6	0.34	0.46	2
7	0.32	0.46	2
8	0.34	0.46	2
9	0.33	0.46	2
Dis-1 (8)	0.53	0.89	3
Dis-2 (3)	0.45	0.53	2

Table 28: Discretization quality for
win_opened_If attribute.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.31	0.34	1
4	0.33	0.4	1
5	0.31	0.4	1
6	0.35	0.47	1
7	0.34	0.47	1
8	0.33	0.47	1
9	0.33	0.43	1
Dis-1 (24)	0.37	0.5	6
Dis-2 (7)	0.28	0.37	1

Table 29: Discretization quality for win_time_elapsed attribute.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.31	0.34	1
4	0.33	0.4	1
5	0.31	0.4	1
6	0.35	0.47	1
7	0.34	0.47	1
8	0.33	0.47	1
9	0.33	0.43	1
Dis-1 (14)	0.39	0.45	4
Dis-2 (6)	0.37	0.49	3

Table 30: Discretization quality for win_time_elapsed_1f attribute.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.35	0.41	2
4	0.36	0.55	2
5	0.43	0.64	2
6	0.49	0.67	2
7	0.57	0.79	3
8			
9			
Dis-1 (7*)			
Dis-2 (7*)			

Table 31: Discretization quality for win_title_prot_words attribute.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.28	0.43	1
4	0.31	0.41	1
5	0.3	0.41	1
6	0.31	0.31	1
7	0.33	0.41	2
8	0.33	0.39	2
9	0.34	0.43	2
Dis-1 (23*)	0.46	1	7
Dis-2 (3)	0.29	0.38	1

Table 32: Discretization quality for win_title_sani_words attribute.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.42	0.62	2
4	0.41	0.62	2
5	0.39	0.62	2
6	0.42	0.62	2
7	0.4	0.62	2
8	0.4	0.54	2
9	0.43	0.57	2
Dis-1 (17)	0.54	0.74	2
Dis-2 (4)	0.33	0.35	1

Table 33: Discretization quality for win_title_total_to_prot_words attribute.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.3	0.34	1
4	0.31	0.44	1
5	0.35	0.45	1
6	0.34	0.45	1
7	0.34	0.45	1
8	0.33	0.45	2
9	0.34	0.43	2
Dis-1			
Dis-2 (3)	0.35	0.51	1

Table 34: Discretization quality for win_title_total_words attribute.

# Ranges	PROMISE	max PROMISE	# users with non-zero PROMISE
3	0.3	0.45	1
4	0.29	0.45	1
5	0.32	0.45	1
6	0.31	0.45	1
7	0.34	0.45	2
8	0.36	0.45	2
9	0.34	0.45	2
Dis-1 (28)	0.38	0.64	7
Dis-2 (6)	0.36	0.59	3

Table 35: Discretization quality for session_start_sec attribute.

5.2.3 Combined Discretization

The goal of this data preparation task was to combine the best outcomes from the manual and automated discretization. The method of choosing the most appropriate discretization for an attribute took three criteria into account. The first criterion, on which more emphasis was placed, was the number of split points used in a discretization – the smaller the number, the better, because it speeds up the learning process significantly and results in more comprehensible knowledge.

The second and third criteria used were the quality of an attribute represented by the PROMISE and MAX-PROMISE measures – higher values of these measures were preferred. Figure 12 shows the values of the criteria used in this task. As a result discretization scheme *Dis-3* was developed, which is presented in Table 36. Column **Attribute** corresponds to the attribute's name and column **Discretization points** corresponds to the number of the points of discretization (the number of intervals is greater by one. Column **Defined** indicates what type of definition was used in the data creation: *Prep* stands for the values created without use of any external discretization scheme, *Chi* means values resulting from application of the Chi-Merge method, and *Dis2* denotes the values from the discretization scheme *Dis-2*. In this column, the number after “-“ stands for the number of values in the domain of the attribute. The term “not discretized” is used to indicate attributes that are kept in their original form (they by nature can be already discrete, for example attributes host and day).

Attribute	Defined	Discretization points
host	Prep-21	<i>not discretized</i>
day	Prep-7	<i>not discretized</i>
hour	Prep-24	<i>not discretized</i>
session_start_sec	Chi-3	6899.5,15656.5,247561
process_name	Prep-181	<i>not discretized</i>
event_status	Prep-4	<i>not discretized</i>
proc_cpu_time	Chi-4	137.5,265.5,354.5,429
proc_inactive_time	Chi-3	471.5,1344.5,247517
proc_inactive_time_lf	Chi-3	6.15803,7.20452,12.4192
proc_inactive_time_gtlmin	Prep-2	<i>not discretized</i>
win_pid	Prep	<i>not discretized</i>
win_title	Prep	<i>not discretized</i>
proc_cpu_time_in_win	Chi-7	0.5,1.5,5.5,12.5,35.5,55.5,413
proc_cpu_time_in_win_lf	Chi-7	0.346573,0.895879,1.86884,2.60201,3.59722,4.0342,6.02587
win_time_elapsed	Chi-6	31.5,185.5,623.5,1541.5,2949,246762
win_time_elapsed_lf	Chi-6	3.48112,5.22843,6.43694,7.34116,7.98956,12.4162
proc_cpu_to_win_elapsed_ratio	Chi-3	0.005,0.045,2000
delta_time_new_window	Dis2-6	10500,11000,13000,24000,30000
delta_time_new_window_lf	Chi-4	1.49787,2.44141,3.02013,12.4192
new_win_time_elapsed	Chi-3	7409.5,15234,247561
new_win_time_elapsed_lf	Chi-4	0.693147,8.91065,9.81416,12.4194
prot_words_chars	Chi-5	7.5,8.5,24,25.5,50
prot_words_chars_to_total_chars_ratio	Chi-7	0.146087,0.242045,0.319091,0.320256,0.472999,0.914634,1
win_title_total_words	Dis2-2	5,15
win_title_total_to_prot_words_ratio	Chi-3	0,0.322916,0.348484
proc_count_in_win	Chi-6	6.5,32.5,103.5,260.5,1239,2435
proc_count_in_win_lf	Chi-6	2.01267,3.51143,4.64917,5.56641,7.12078,7.79811
win_opened	Dis2-2	16,28
win_opened_lf	Dis2-2	2.7,3.3
win_title_prot_words	Prep-8	<i>not discretized</i>
win_title_sani_words	Chi-3	1.5,3.5,23

Table 36: Discretization scheme *Dis-3*.

5.3 Window Size / Lookback

An important parameter in the LUS approach is the size of the time slice considered to be an event. For multistate template models, this is represented by the *window size*. For prediction-based models, this is represented by the *lookback*. This parameter can be adjusted by the user, and experimental testing can determine which setting provides the best results.

A user can specify this parameter to the data preparation program, which will build events accordingly.

5.4 Attribute Selection

Selecting which attributes are to be learned from is an important task during the data preparation stage. By removing less relevant attributes, the learning process will be faster, and the chance of the discovery of spurious rules may be lessened. One can remove by hand those that are clearly irrelevant (such as Process ID). An option we have explored is to apply the PROMISE algorithm (Baim, 1982, Kaufman 1997) and Gain Ratio algorithm (Quinlan, 1993) with a threshold to the training data set. Both algorithms were applied in two modes, standard and max, which is defined as the maximum of the evaluations of one class against other classes.

5.4.1 Common Attributes

Application of the above methods to LUS data provides four quality measures per attribute. To aggregate all four values, we followed the algorithm presented in Figure 13.

1. Compute: PROMISE, PROMISE max, Gain ratio, Gain Ratio max
2. For all four measures, select the five best attributes
3. Count attributes in the four sets and rank them according to their number of appearances
4. Select the six best attributes ranked in (3).

Figure 13: Attribute selection algorithm.

The method was applied to the LUS data with attributes discretized using the Dis-3 schema. Lists of attributes chosen by the four described criteria and the table with final ranking are presented in Figures 14-18 below.

Gain Ratio
win_opened_lf
prot_words_chars
process_name
win_opened
proc_inactive_time_gt1min

Figure 14: Attributes selected based on Gain Ratio.

Gain Ratio MAX
win_title_prot_words
prot_words_chars
process_name
win_opened
proc_inactive_time_gt1min

Figure 15: Attributes selected based on Gain Ratio MAX.

PROMISE
win_opened
process_name
delta_time_new_window
win_title_prot_words
win_opened_lf

Figure 16: Attributes selected based on PROMISE.

PROMISE MAX
process_name
delta_time_new_window
proc_count_in_win_lf
win_opened
prot_words_chars

Figure 17: Attributes selected based on PROMISE MAX.

Rank	
process_name	4
win_opened	4
prot_words_chars	3
delta_time_new_window	2
proc_count_in_win_lf	2
win_title_prot_words	2
win_opened_lf	1
proc_inactive_time_gt1min	1
win_opened_lf	1

Figure 18: Rank of attributes. The attributes in bold were selected.

As shown in Figure 18, the selected attributes were process_name, win_opened, prot_words_chars, delta_time_new_window, proc_count_in_win_lf, and win_title_prot_words.

5.4.2 User-oriented Attributes

Two sets of user-oriented attributes were prepared, based on the Promise and Gain Ratio attribute quality measures. The selected attributes for the Gain Ratio quality measure are presented in Figure 19. A description of the learning and testing methods applied the user-oriented attribute sets may be found in Section 8.

Attributes	User #									
	1	2	3	4	5	7	8	12	19	25
process_name										
event_status										
proc_cpu_time										
proc_inactive_time_gt1min										
proc_cpu_time_in_win_lf										
delta_time_new_window										
win_time_elapsed_lf										
prot_words_chars										
prot_words__chars_to_total_chars_ratio										
new_win_time_elapsed										
new_win_time_elapsed_lf										
proc_count_in_win										
win_opened										
win_opened_lf										
win_title_total_words										
win_title_prot_words										

Figure 19: User oriented attributes selected using Gain Ratio quality measure.

5.5 Determining Training and Testing Data Streams

Our initial efforts aimed at mirroring earlier experiments; therefore the first ten sessions of each of the ten selected users were designated as training data streams (i.e., used for creating user models), and the next five sections were designated as testing data streams (i.e., used for determining the predictive accuracy of the learned models).

Such a selection of training and testing of datastreams has a disadvantage in that learning an adequate model for different users may require training datastreams of different length. Moreover, it is not known a priori what should be the length of the training datastream for any given user.

To solve this problem, we have developed a “sausage” method for determining the “best” training datastream.

To explain the sausage method, assume that all records of each user’s behavior from all training sessions have been lined up in chronological order into a single, long string, resembling a sausage. Suppose now that the $CP_i\%$ of the records from the beginning of the sausage are selected to be training datastreams for each user, and $CP_i\%$ of the records from the end of the sausage are selected to be testing data streams (Figure 20). Suppose that such training and testing datastreams are created for different values of CP_i , called *cut points*, say, for $CP_1 = 10\%$, $CP_2 = 25\%$, $CP_3 = 50\%$, $CP_4 = 75\%$, $CP_5 = 90\%$ and $CP_6 = 100\%$. Clearly, for $CP_6 = 100\%$ the training and testing datastreams are identical, but for $CP_1 = 10\%$ they may be quite different, since the training and testing data streams not only do not overlap, but may also be significantly separated in time.

Let us introduce a measure of similarity, SIM , between any two datastreams, and determine a function $SIM(CP_i)$ that characterizes the dependence of the similarity measure between training and testing datastreams for different cut points. Assuming that the learning system works well, the value $SIM(CP_i)$ at any given cut point, CP_i , should indicate the chances that the model learned from the training datastream obtained at that cutpoint will work well on the testing datastream obtained at that cutpoint.

The $SIM(CP_i)$ function must clearly be monotonically increasing for CP_i -s greater than 50%, and for CP_i -s smaller than 50% it should be at least approximately monotonically increasing. Thus, the larger the CP_i , the better performance of the models should be. On the other hand, the higher the CP_i , the larger the training dataset, and thus the higher the computational cost of learning a user model. By determining the cut point of “diminishing returns,” one can select a desirable length of the training and testing data streams for each user.

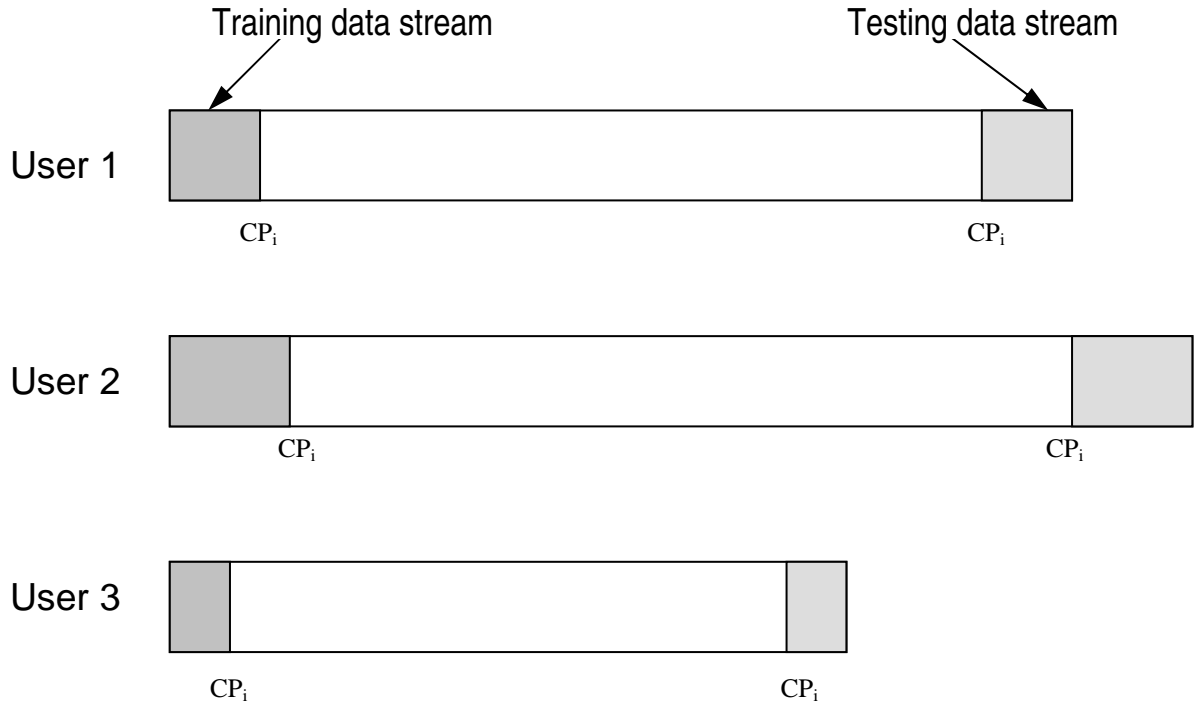


Figure 20: Illustration of the sausage model of episode size selection.

One can exercise various approaches to defining similarity between two datasets. In our research we wanted to concentrate on the issue of degree of consistency of user's behavior represented in two investigated datasets. Therefore we have developed a compound measure of similarity, called Combined Similarity (CS) that integrates two aspects of the user's observed actions represented by Forward Similarity (FS) and Backward Similarity (BS). We define FS as the fraction of the events in the dataset (let's call it DS1) pertaining to user's activity over some selected period of time, that match (over specified attributes) some events in the dataset (DS2) related to the user's activity that was observed later. High values of the FS indicate that there is a high chance that the observed behavior will manifest itself in the future, therefore models built from DS1 treated as training data should have strong ability to recognize such behaviour, if it occurs, in the stream of testing events. BS is successively defined as the fraction of events in DS2 that similarly match some event in DS1. High values of BS mean a low probability that the user's activity represented in DS2 has not been observed in the past. We consider the FS and BS measures treated jointly as Combined Similarity, which is computed as their product. This may have the ability to indicate high probability for both building a strong model and matching this model with the testing data.

Furthermore, we have extended the concept of similarity into self-similarity and cross-similarity. Self-similarity refers to the measurement of CS for both DS1 and DS2 belonging to the same user, and cross-similarity to the measurement of the data representing two different models. Cross-similarity between users U1 and U2 is computed from the FS between U1's training data U2's testing data, and the BS between U2's training data and U1's testing data. Thus, high values

of cross-similarity should indicate that the observed behavior of the users U1 and U2 is rather indistinguishable, and we should not expect good recognition performance of the developed models, even though the self-similarity for each user may be very high. Low self-similarity indicates a low consistency in a given user's behavior.

In order to have better insight into the nature of the data we deal with and also to have a better explanation of the results from the knowledge learning and testing, we have planned and conducted a number of experiments that measure CS between the training and testing data of each user. The results of these experiments as well as investigation of the “sausage” approach are presented in Section 8.

6 PLAN OF EXPERIMENTS

The goal of these experiments is to come up with the best set of parameters used for learning and testing users' models. This includes searches for the best discretization, the best attributes, and for the best combination of AQ21 parameters for learning and testing. The process needs to be done iteratively, and each time the representation space is modified, it is necessary to perform a set of learning and testing phases in order to evaluate the current data used. The most important parameters of the experiments are grouped below depending on their types.

6.1 Experiment Set 1: Search for the Best Representation Space

The goal of the first set of experiments is to determine the best representation space, containing the most suitable and best discretized attributes. In this study we compute various quality measures for different discretization methods. These experiments are based on methods described in Sections 5.2–5.5.

6.2 Experiment Set 2: Search for the Minimum Amount of Data Needed for Learning

In these experiments we search for the minimum amount of data needed to correctly learn and apply users' models. We use the “sausage” idea and SIMd measure described in Section 5.5. The initial experiment used the four best users as described below, and subsequent experiments used all 10 users who were most prolific in the data.

The search for the best SIMd value should include different discretization methods, event filtering methods and attribute selection methods on which the measure of similarity will be affected. A table of the proposed experiments is presented in Figure 21.

In the diagram, the three dimensions on the vertical axis are: window/lookback size, event filtering and attribute selection. Window/lookback sizes that are investigated are three, four, and five. Similar events may not be filtered, filtered only from training data, or filtered from both training and testing data. Selection of attributes includes use of all attributes, attributes with discriminatory power above 0.3, and attributes with discriminatory power above 0.6.

Window/lookback Size																																					
Filterit Selection																																					
3	NO	no 0.3 0.6	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7																
	TR	no 0.3 0.6																																			
	TR &	no 0.3																																			
	TS	0.6					3	3	3	3	3	3	3	3	3	3	3	3	3	3	3																
4	NO	no 0.3 0.6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																
	TR	no 0.3 0.6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																
	TR &	no 0.3																																			
	TS	0.6					3	3	3	3	3	3	3	3	3	3	3	3	3	3	3																
5	NO	no 0.3 0.6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5																
	TR	no 0.3 0.6																																			
	TR &	no 0.3																																			
	TS	0.6					2	2	2	2	2	2	2	2	2	2	2	2	2	2	2																
TR & TS Size		10%	30%	50%	70%	90%	100%	10%	30%	50%	70%	90%	100%	10%	30%	50%	70%	90%	100%	10%	30%	50%	70%	90%	100%												
Discretization		NO						DIS-1						DIS-2						ChiMerge-1						ChiMerge-2						ChiMerge-3					

Figure 21: Plan of tasks for data selection experiments.

Similarly the two dimensions on the horizontal axis represent the size of the training and testing data (the sausage idea described in Section 5.5), and the method used for discretization of continuous attributes. The sausage sizes investigated are: 10%, 30%, 50%, 70%, 90%, and 100%. A size of 100% implies that the training and testing datasets are the same.

The search for the best discretization includes experiments with undiscretized data, data discretized by hand (DIS-1, and DIS-2 described in Section 5.2.1), and data automatically discretized using the ChiMerge algorithm, described in Section 5.2.2.

It can be noted in Figure 13 that there are 972 possible combinations of parameters to be investigated. Since the size of the space is so large, we propose to investigate only the most promising areas of the experiment space. The numbers in the above diagram represent the order of performing sets of experiments.

The goal of the first, second, and third sets of experiments is to determine window/lookback size, using the simplest possible data -- data that is discretized, well filtered, and contains only the most relevant attributes.

6.3 Experiment Set 3: Search for the Best Filtering Parameters

Selection of the best filtering method requires a search through all defined methods and testing them on real data. Data filtered using methods described in Section 5.1 are passed to the AQ21

learning program, and the learned hypotheses are tested. The most promising filtering methods are then selected for further investigation.

Filtering methods are evaluated based on the number of correct and first choice correct answers of the EPIC-MT testing module. Parameters used for learning and testing hypotheses are presented below. Results of testing of are presented in Section 8.

AQ21 Learning Parameters:

maxstar = 1, 5, 10 maxrule = 1, 5, 10 ambiguity = ignore-for-learning

trim = mini exceptions = false mode = tf

Characteristic descriptions, discriminant descriptions, simplicity-based descriptions

Testing Parameters:

Evaluation of Conjunction = strict, coverage_ratio, selectors_ratio

Evaluation of Disjunction = max

Acceptance Threshold = 10%

Accuracy Tolerance = 5%

6.4 Experiment Set 4: Search for the Best AQ21 Parameters

The AQ21 Learning program has a number of parameters that can be optimized in order to meet different applications' requirements. The search for optimal parameters for LUS experiments should be conducted for both the Multistate Templates model and the Prediction-Based model. This search involves a search for the best learning parameters, testing parameters, types of descriptions, and various other technical parameters that control the AQ learning algorithm (e.g., maxstar and maxrule). These parameters are grouped by type and presented below.

Learning: User Models

- Multistate templates (viewing attributes over a window of raw events)
- Prediction-based (predicting user behavior)

Learning: parameters

- Learning mode (TF, ATF, PD)
- Ambiguity handling (Ignore, IncludeAsNeg, IncludeAsPos)
- Trimming (MostGen, Optimal, MostSpec)
- Threshold for truncation of rules with low unique coverage (no truncation, 5, 10, 20 examples)

Learning: Evaluation of rules

- Discriminative
- Characteristic
- Simplicity based

Testing

- Testing method for multistate templates model (EPIC-MT, EPIC-RB)
- Testing method for prediction-based model (EPIC-P)

- Matching – evaluation of conjunctions (strict, flexible, coverage ratio)
- Aggregation – evaluation of disjunctions (maximum, average)

The search space for the best parameter settings can be presented graphically using the following GLD (Figure 22). Since the graph presents only the most important parameters, the space is in fact much larger. For instance, it does not include evaluation methods used in testing, trimming and truncation options. In the GLD in Figure 22, the vertical axis consists of three dimensions: EPIC type, method of evaluating conjunctions, and method of evaluating disjunctions.

[illegible]

Figure 22: AQ21 parameter search space for multistate template model.

For datasets prepared using methods described in Section 4, we apply a standard set of experiments invoking AQ learning. Each cell in the diagram below identifies a set of nine experiments with AQ21 (maxstar = 1, 5, 10, maxrule = 1, 5, 10). Numbers in the cells correspond to order in which the AQ experiments are to be performed.

7 ILLUSTRATION AND VALIDATION OF THE LUS METHOD BY DIAGRAMMATIC VISUALIZATION

In order to illustrate and validate some aspects of the LUS methodology, we created two very simple, imaginary user data streams representing activities of hypothetical users User 1 and User 2. Using these datastreams, we illustrated selected steps of the LUS methodology using *diagrammatic visualization*. Diagrammatic visualization employs a *general logic diagram* to present a multidimensional discrete space on a plane (Michalski, 1978; Wnek, 1995; Zhang, 1997; Sniezynski, Szymacha, and Michalski, 2005).

The problem assumes a very small event space defined by three attributes, x_1 , x_2 , and x_3 , each of which can take on values 0, 1 and 2. We assume two users, User 1 and User 2, each of whom are observed performing some of the activities represented by the event space. Figure 25 shows a diagrammatic visualization of the event space. Each cell represents one combination of values of the attributes x_1 , x_2 and x_3 (an event). In each cell, the first and the second numbers indicate the frequency of nxk grams ($n=1$, $k=3$) occurring in the User1 and User2 training data streams, respectively. For example, it can be seen that the frequencies of the $1x3$ -gram $\langle 0,1,0 \rangle$ (corresponding to the event $[x_1=0, x_2=1, x_3=0]$) in User1 and User2 datastreams are 8 and 3, respectively.




0	0, 3	5, 8	2, 4	8, 3		1, 2	1, 7	4, 3	0, 4	
1	2, 4	7, 0	6, 3	2, 5	3, 3	2, 1	7, 5	5, 0	1, 4	
2	5, 4		2, 4	6, 2	9, 0	1, 4	0, 4	2, 2	0, 2	
x1	0	1	2	0	1	2	0	1	2	x3
		0			1			2		x2


Figure 25: Visualization of the event space with event frequencies.


Applying AQ21 to this data with the parameter `ambiguity=IgnoreInData`, which means that the events that occurred in the datastreams of both users are not used for learning rules, produced the training examples shown in Figure 26. In that figure, cells marked “1” represent events retained for learning User 1’s profile (i.e., User 1 had activity represented by that cell, but User 2 did not), and cells marked “2” similarly represent those retained as examples of User 2’s behavior. Figure 27 shows rules learned by AQ21 on the basis of that training data. The learned rules representing User 1’s profile are displayed in blue, and those representing User 2’s profile are in red. Links in the diagram connect separate parts of the same rule.

0	-	-	-	-		-	-	-	-
1	-	+	-	-	-	-	-	+	-
2	-		-	-	+	-	-	-	-
x1	0	1	2	0	1	2	0	1	2
		0			1			2	
									x3
									x2

Figure 28: Training data for learning User 1's profile after treating ambiguous events as negative.

0	-	-	-	-		-	-	-	-
1	-		-	-	-	-		-	-
2	-		-	-		-	-	-	-
x1	0	1	2	0	1	2	0	1	2
		0			1			2	
									x3
									x2

 $[x1=1] [x2=0] [x3=1]$

 $[x1=2] [x2=1] [x3=1]$


 $[x1=2] [x2=2] [x3=1]$

Figure 29: AQ21 rules learned for User 1 after treating ambiguous events as negative.

































0	2	2	2	1		2	2	1	2	
1	2	1	1	2		1	1	1	2	
2	1		2	1	1	2			2	
x1	0	1	2	0	1	2	0	1	2	x3
		0			1			2		x2

Figure 36: Training data after putting ambiguous events in the predominant class.

0										
1										
2										
x1	0	1	2	0	1	2	0	1	2	x3
		0			1			2		x2

User 1

-  [x1=1] [x2=2] [x3=0..1]
-  [x2=1,2] [x3=1]
-  [x1=1] [x2=0..1] [x3=1..2]
-  [x1=2] [x2=0..1] [x3=0..1]
-  [x1=0] [x2=1] [x3=0]

User 2







-  [x1=0] [x2=0]
-  [x2=2] [x3=2]
-  [x1=1] [x2=0..1] [x3=0]
-  [x1=2] [x3=2]
-  [x1=0] [x2=2] [x3=0]
-  [x1=0] [x3=2]

Figure 37: AQ21 rules learned after putting ambiguous events in the predominant class.

To test the LUS event filtering algorithms, we applied the Sig6 significance measure to the data, where the significance was defined as $p^{5/4} / (p + n)$. This simplification is equivalent to the Sig6 measure since $P = N$ in this problem. The significance numbers for the two classes for each event (User 1 on top, User 2 underneath) are shown in Figure 38. Events selected using significance threshold 1 are shown in Figure 39, and rules learned are shown in Figure 40.

0	0	.58	.40	1.22	0	.33	.13	.81	0
1	1.32	1.03	.94	.36	0	.79	1.42	.56	1.41
1	.40	1.63	1.04	.34	.66	.79	.95	1.5	.2
2	.94	0	.44	1.07	.66	.33	.62	0	1.13
2	.83	0	.40	1.17	1.73	.20	0	.59	0
x1	.63	0	.94	.30	0	1.13	1.41	.59	1.19
	0	1	2	0	1	2	0	1	2
		0			1			2	
									x3
									x2

Figure 38: Event significance based on the Sig6 measure.

0	2	2		1			2		2
1		1	1	2				1	2
2				1	1	2	2		2
x1	0	1	2	0	1	2	0	1	2
		0			1			2	
									x3
									x2

Figure 39: Training events after data filtering using the Sig6 measure.

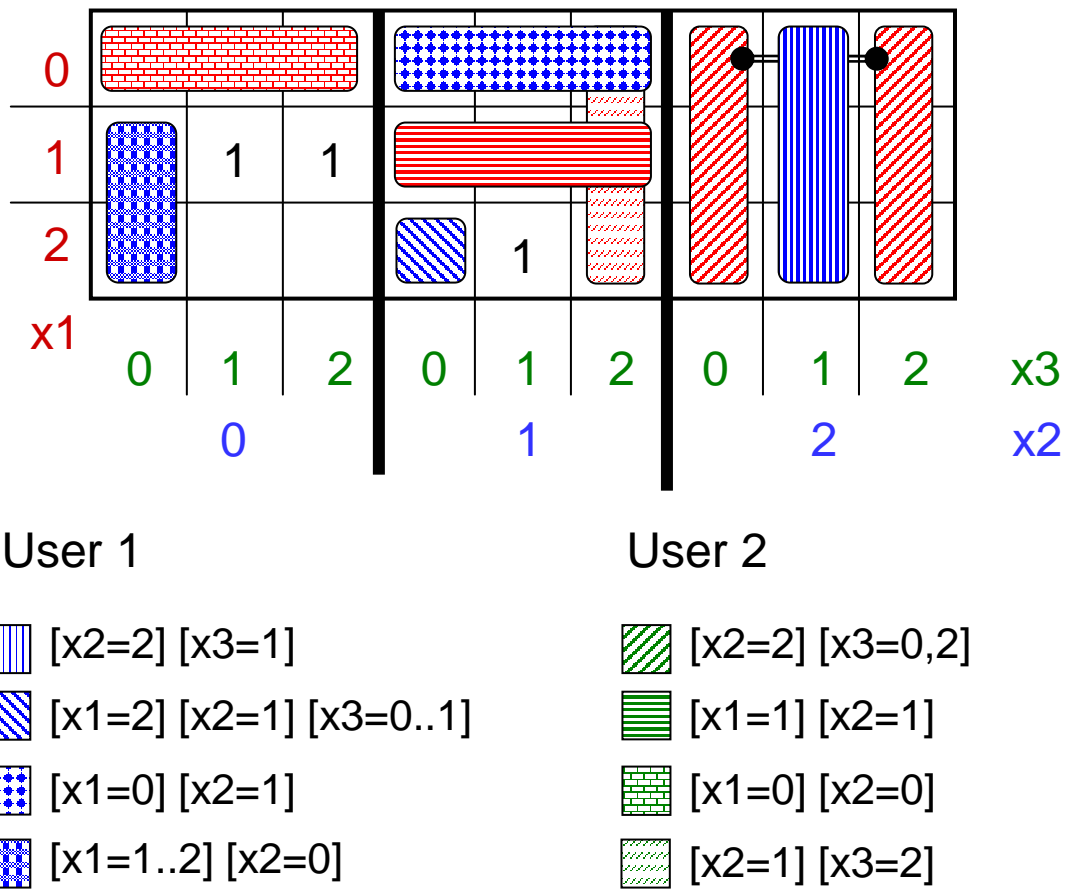


Figure 40: Rules learned after data filtering using the Sig6 measure.

8 EXPERIMENTAL RESULTS AND EVALUATION

8.1 Measuring Similarity between Episodes

8.1.1 *Testing the “Sausage”*

The purpose of the experiments described in this section was to determine if the measures of similarity generated using the “sausage” model are good predictors of the performance of the LUS-learned user models in identifying the users. The following are characteristics of the experiments:

- The data contained all sessions from 10 users as described in Table 1.
- The data (type1) contained both window-based and process-based events.
- The data was not filtered.
- Discretization scheme *Dis-3* (described in section 5.2.3) was used.
- The common set of attributes described in section 5.4.1 was applied (*5-grams*).
- Cutpoints of 10%, 30%, 50%, 70% and 90% were utilized.

The results are presented in Figures 41-47. In the first two figures, only self-similarity (computed as Combined Similarity, defined in section 5.5) for 10 users is shown. On the horizontal the cutpoints are ordered, from left to right, with increasing amountd of data from both ends of the “sausage”. For the cutpoints of 70% and 90%, overlapping of data occurs, which results in a sharp increase in the values of the smiliarity measure. More interesting are the results using the lower thresholds, because this reflects ton a greater degree the level of consistency of a users’ behavior. Based on these results the users were grouped into two equal, with respect to the number of members, groups of more promising (GR1) and less promising (GR2) users.

For the GR1 users, at the 10% cutpoint, the similarity of each is already above the 0.5 level, and often above the 0.6 level. At this point all similarity measures of the GR2 users are significantly below the 0.5 level, most of them being below the 0.4 level. Likewise, at the 30% cutpoint, most of the GR1 users show similarity above the 0.7 level, whereas the GR2 users do not exceed 0.65. It is also the case for the 50% cutpoint, with the exception of user 5 who comes close to the 0.7 value. This gap decreases at the 70% cutpoint and becomes insignificant at the 90% cutpoint, due to the big overlap in the data. One may note that the user 7 is on the borderline between these two groups.

These results might indicate that we should expect better results for the users belonging to the GR1 group (users 4, 7, 8, 19 and 25) than for the users from the group GR2 (users 1, 2, 3, 5, 12)On the other hand, self-similarity does not reflect how similar the users’ behaviors are to one another. Some suggestions may be taken from the outcomes shown in Figures 43-47, which show both self- and cross similarity between the ten users across the “sausage”. Each chart describes results using one of the cutpoints mentioned above. The bars are grouped by users, with the order in each group corresponding to the order of the users as shown on the horizontal axis. Black bars denote the highest value of the similarity between two users. If the remaining bars in a group are white, a given user has self-similarity higher than cross-similarity between him and any other user. Otherwise, the bar referring to user’s self-similarity is graye, and the number on the top of the highest (black) bar in the group indicates the number of the user that a given user is most akin

to. As one may note, the number of gray bars in the figures decreases with the increasing value of the cutpoints.

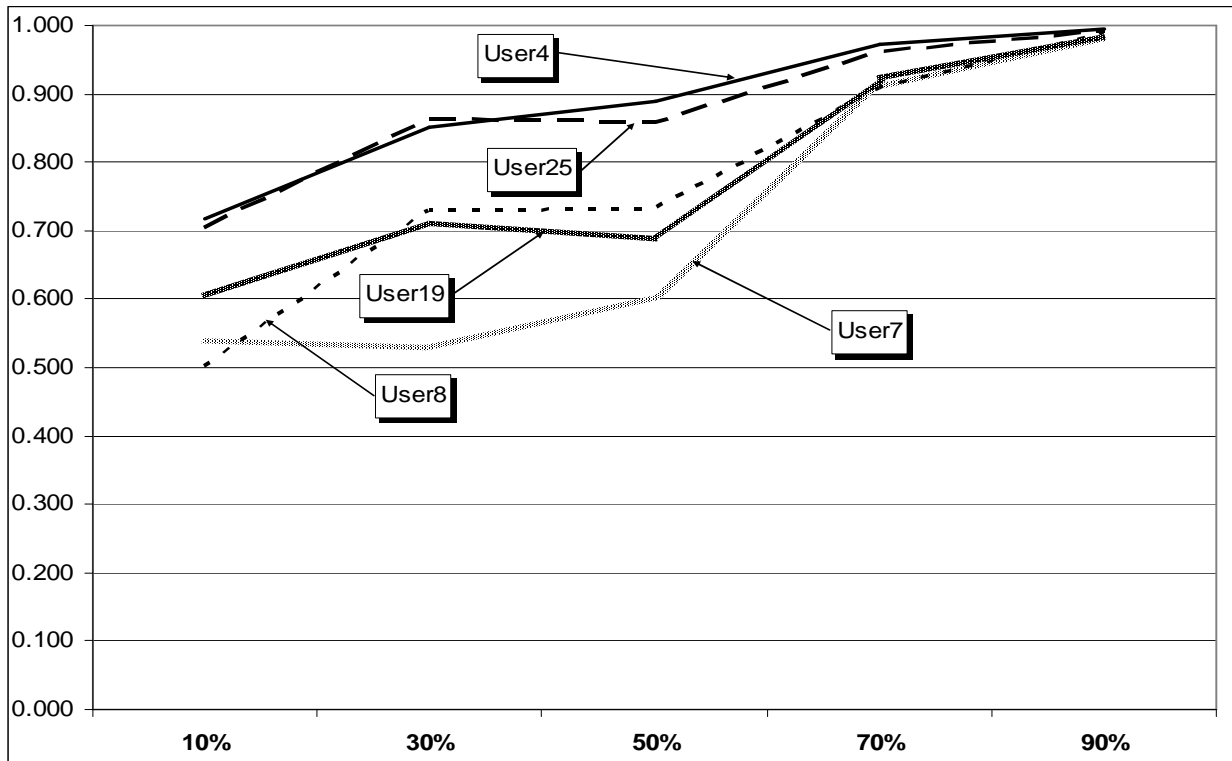


Figure 41: Self -similarity across the "sausage" for users with higher self-similarity (GR1).

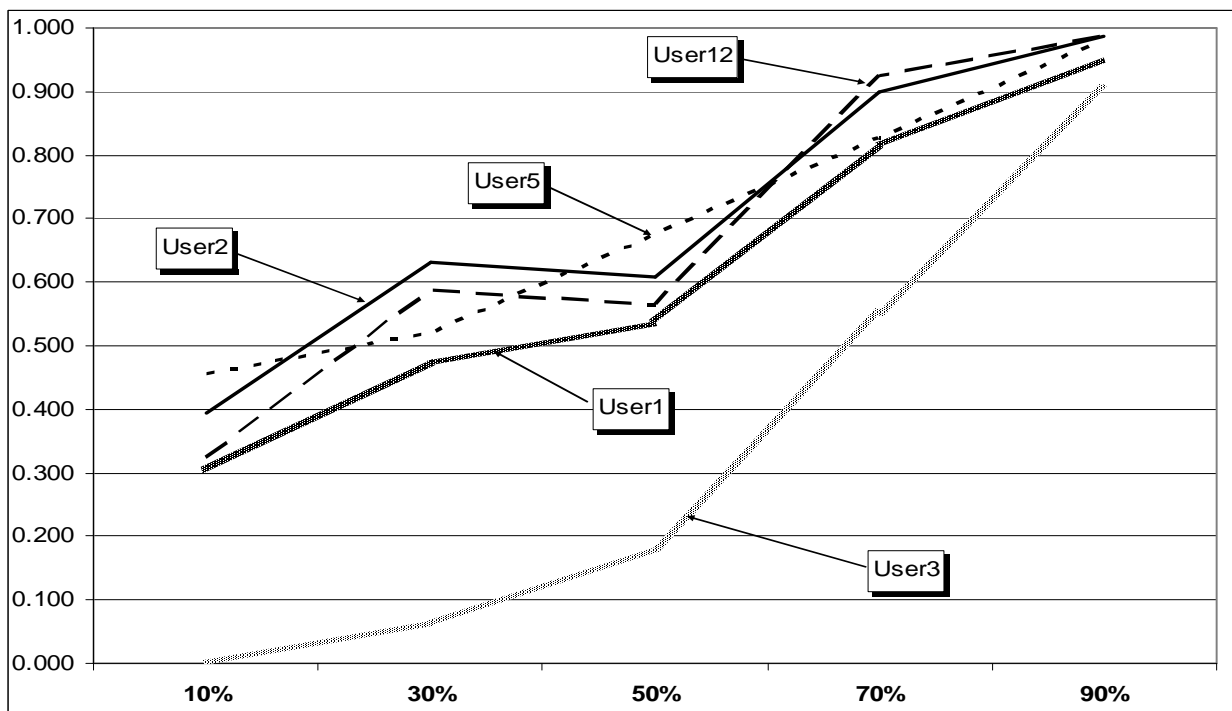


Figure 42: Self -similarity across the "sausage" for users with lower self-similarity (GR2).

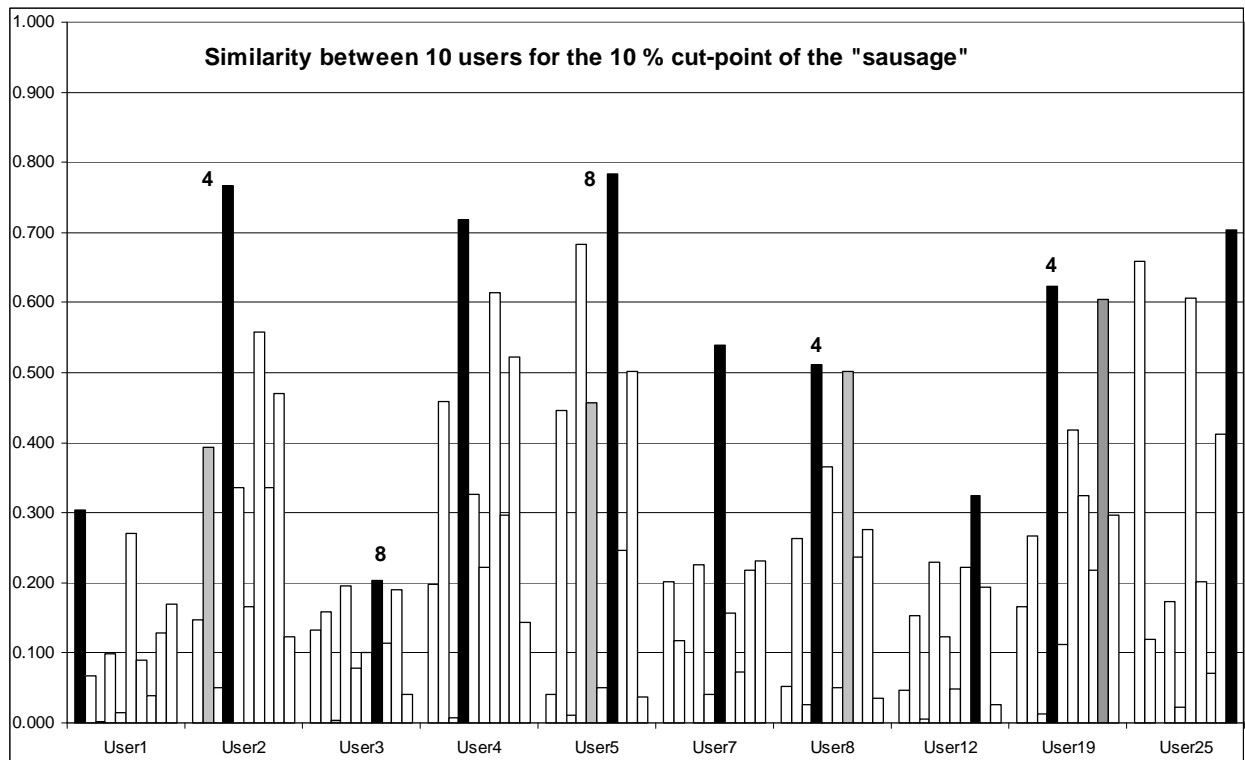


Figure 43: Self- and cross-similarity among 10 users for the 10% cutpoint.

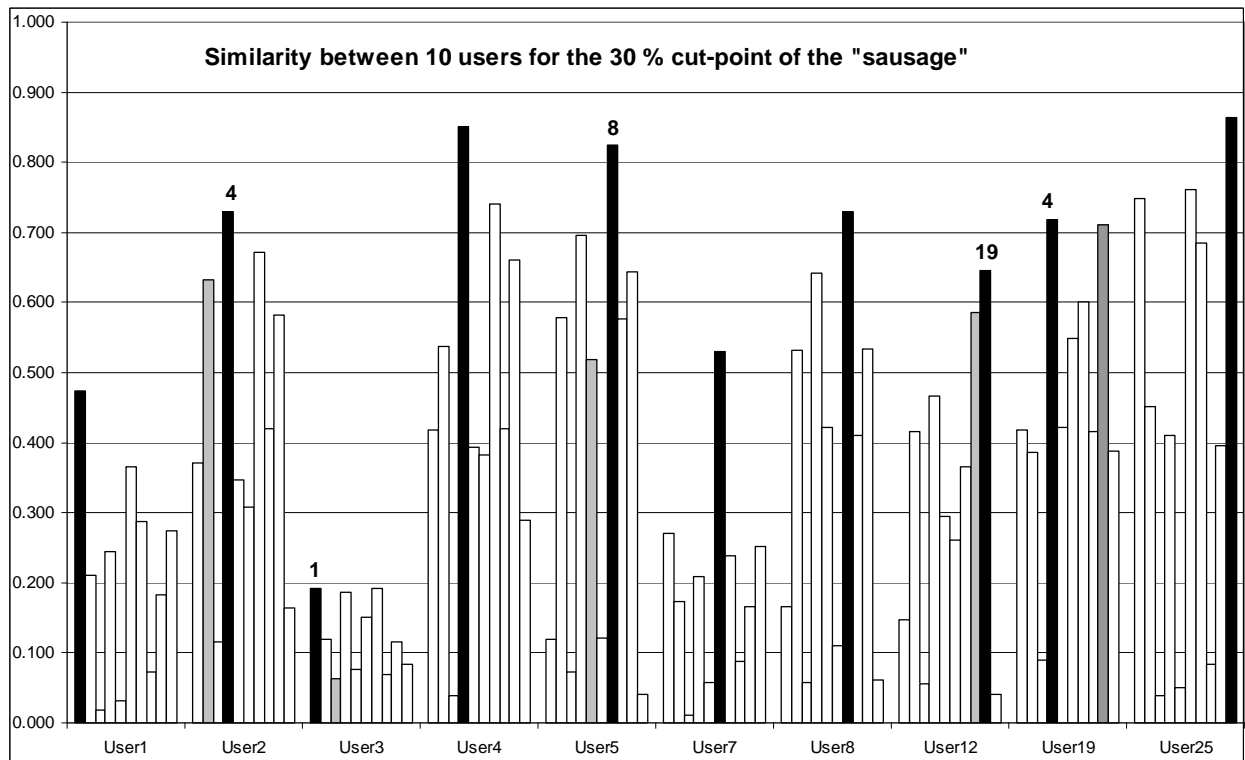


Figure 44: Self- and cross-similarity among 10 users for the 30% cutpoint.

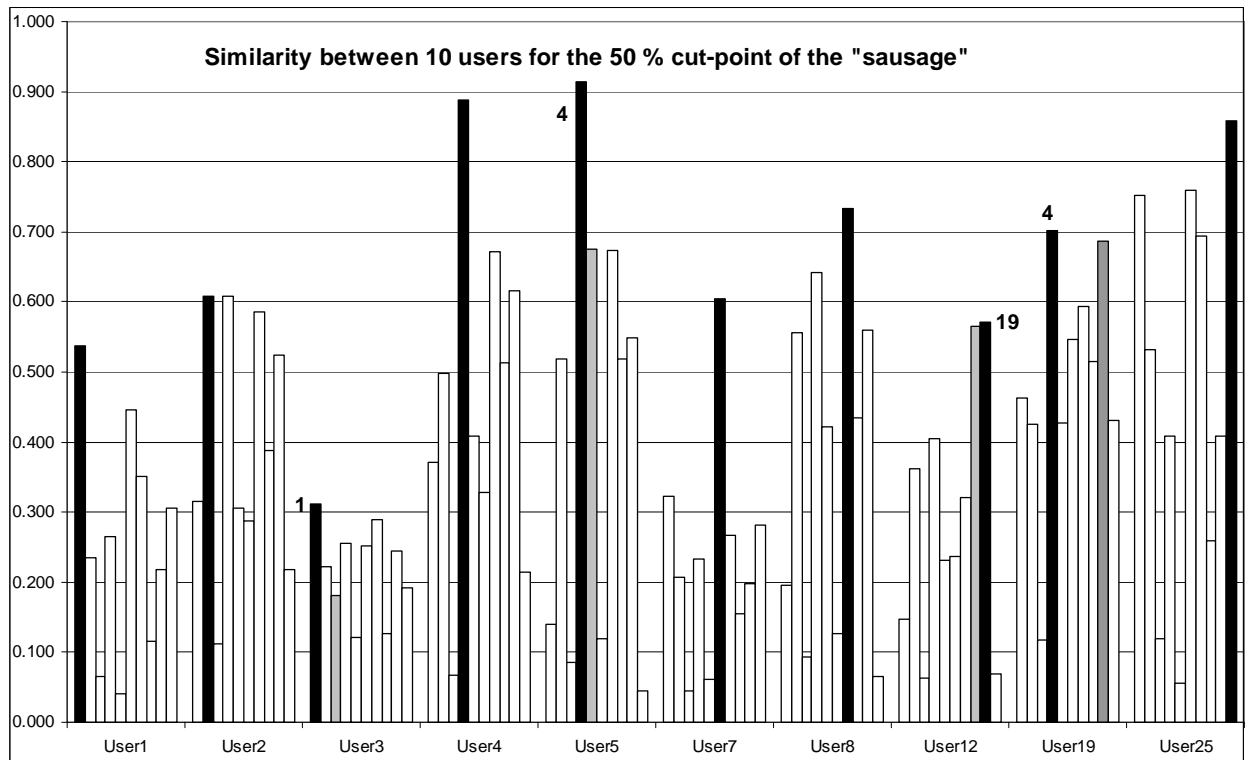


Figure 45: Self- and cross-similarity among 10 users for the 50% cutpoint.

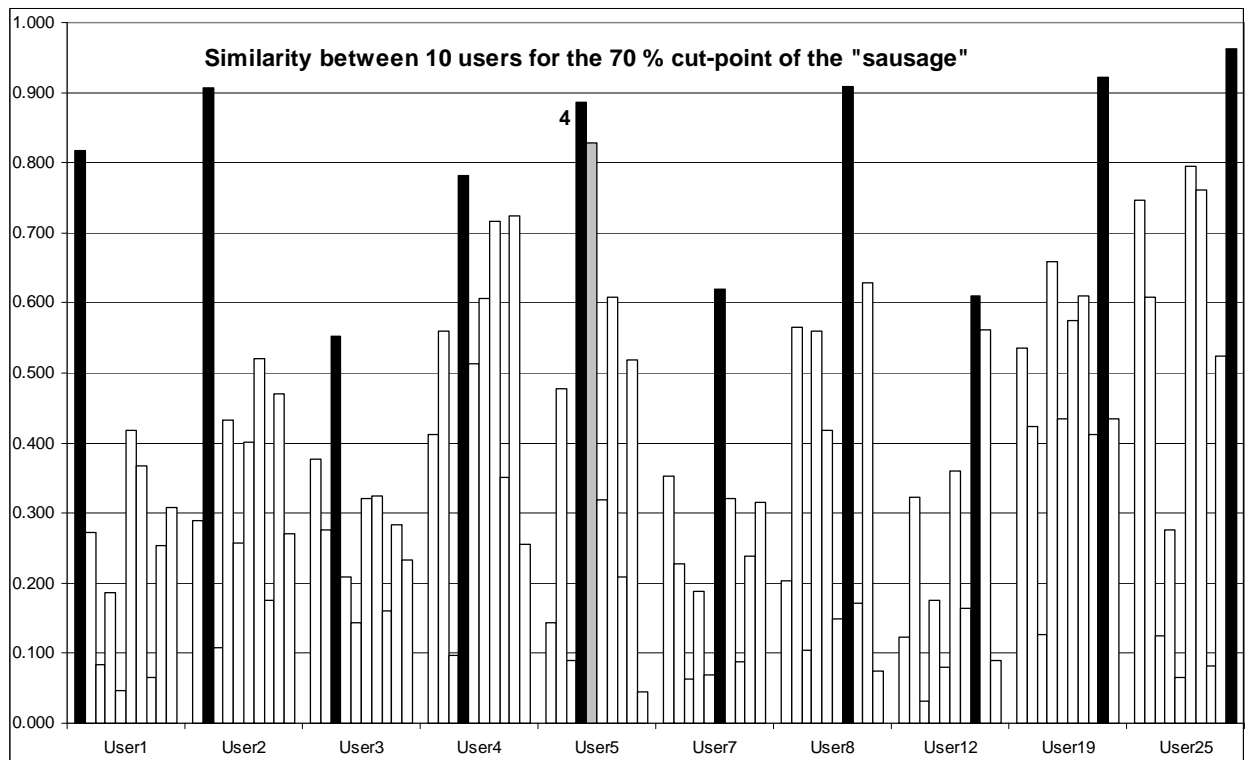


Figure 46: Self- and cross-similarity among 10 users for the 70% cutpoint.

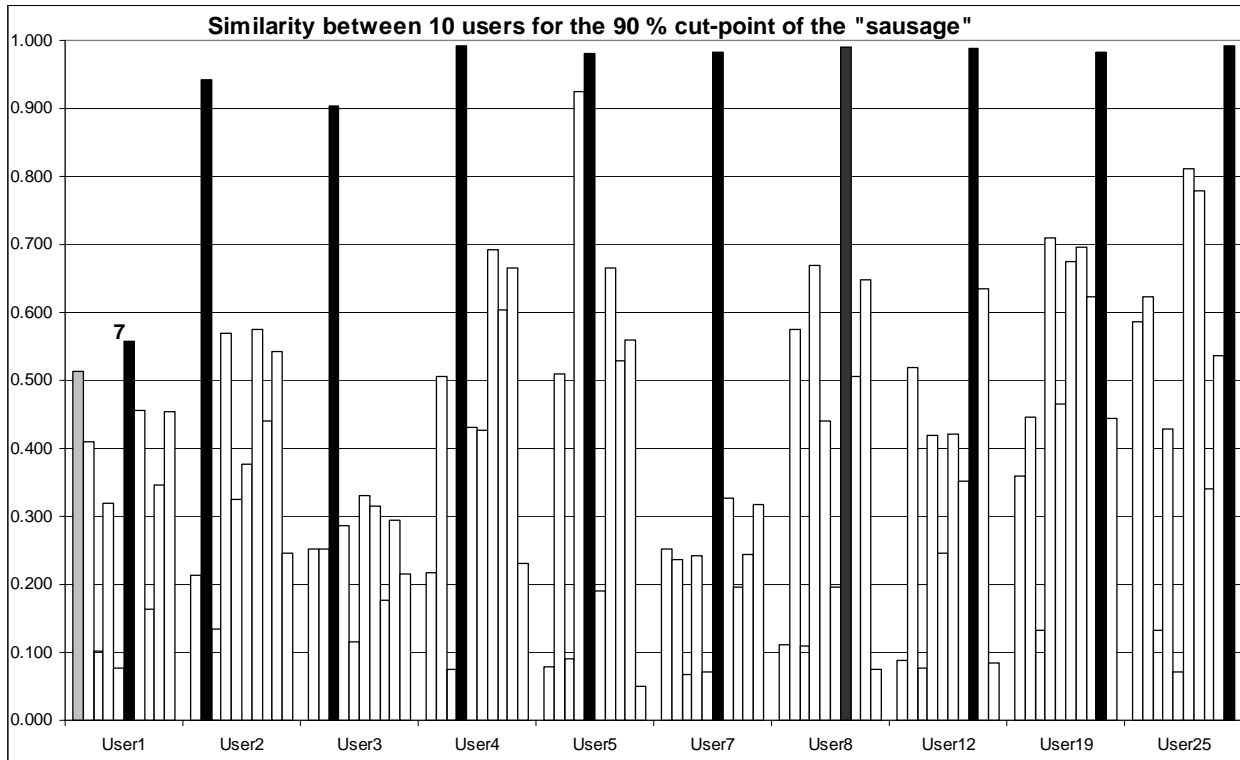


Figure 47: Self- and cross-similarity among 10 users for the 90% cutpoint.

Examination of Figure 43 may suggest which users will perform better, and which will perform worse. The group likely giving better results consists of users 1, 4, 7, 12 and 25. On the other hand we would not expect such a good performance of the models created for users 2, 3, 5, 8 and 19. This is partially confirmed in the outcomes of the knowledge creation and application experiments presented in Section 8.

The experiment in Section 8.4.5 utilized data most closely resembling the data used in conducting the experiments with the “sausage,” since the 10% cutpoint corresponds approximately to the amount of data used in this experiment (Table 37). The experiment shows that the prediction concerning users 7, 12, 25, 3, 5 and 8 was correct. We can also see that cross-similarity is an important factor determining the ability to create and test sound user models since high self-similarity can be dominated by significant cross-similarity to other users

User #	1	2	3	4	5	7	8	12	19	25
Training	1%	24%	3%	8%	30%	3%	3%	18%	7%	11%
Testing	1%	10%	0%	1%	17%	1%	3%	13%	2%	3%

Table 37: Size of the target data of 10 users as percentage of the total amount of their data.

Figures 44-47, reflecting higher “sausage” thresholds, show how the interrelation between behavior of the users changes with the increasing number of observations. For example, at the 30% cutpoint, the self-similarity of user 8 clearly becomes stronger than the cross-similarity of this user, so we can expect improved performance when more user 8 data is used.

On the other hand, results of the model of the user 12 will likely be worse for this cutpoint, since the self-similarity of this user is dominated by the cross-similarity with user 19.

8.1.2 Similarity in the Data used in Learning and Testing

This section presents the results of computing similarity (defined in section 5.5) for data used in the learning and testing experiments described below. In this particular case, the data examined consisted of the training and testing unfiltered datasets described in section 3.2. The discretization scheme used was *Dis-3* (Section 5.4.1), expressed as *5-grams*. In Figures 48-57, a black bar denotes the highest value of the Combined Similarity measure. Each chart also presents the similarity between a given user and other users (cross-similarity) as well as the self-similarity of this user.

For example in Figure 48, the first three bars represent self-similarity for User 1. The first bar is the forward similarity of User 1's training data to User 1's testing data, the second bar is the backward similarity (similarity between his testing and training data), and the third bar is the combined self-similarity for User 1. The next three bars represent cross-similarity between User 1 and User 2. In particular the fourth bar (from the left) in the figure represents the similarity between User 1's training data and User 2's testing data (forward similarity), the fifth bar represents the backward similarity between the testing data of User1 and the training data of User 2, and the sixth bar represents the combined similarity between Users 1 and 2. All other bars in the figure represent analogous similarities between User1 and the other users.

Please note that measure of similarity is not symmetric and the cross-similarity between User 1 and User 2 in Figure 48 is different from that between User 2 and User 1 in Figure 49.

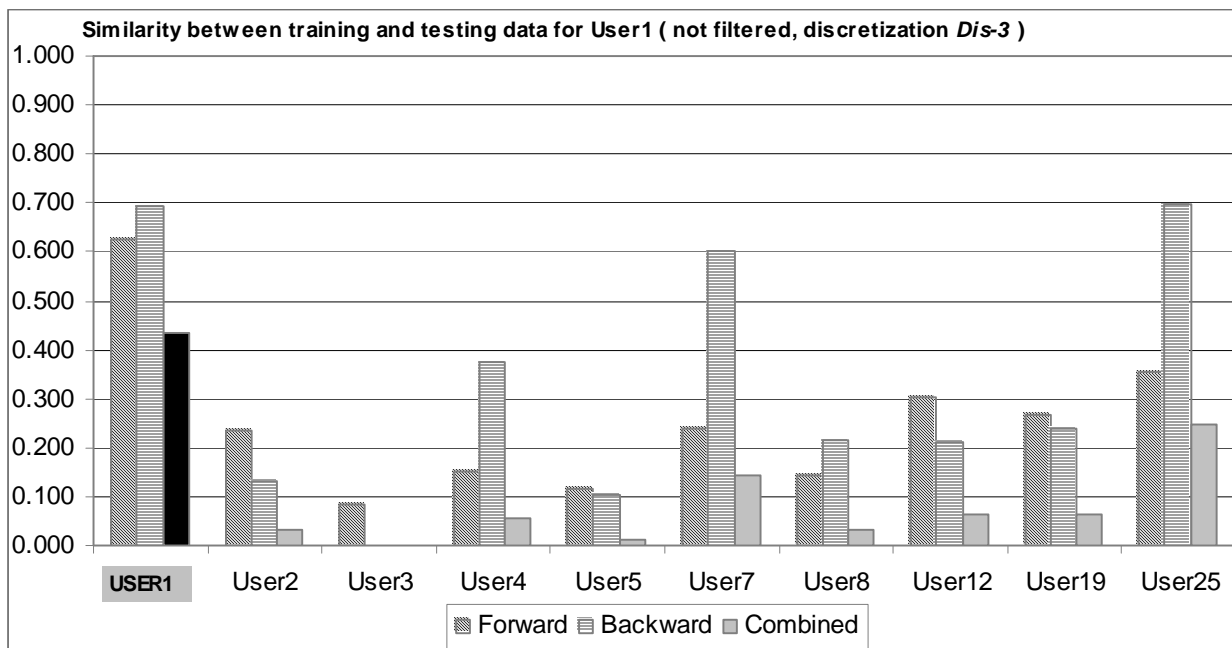


Figure 48: Self- and cross-similarity of the training and testing data for User 1.

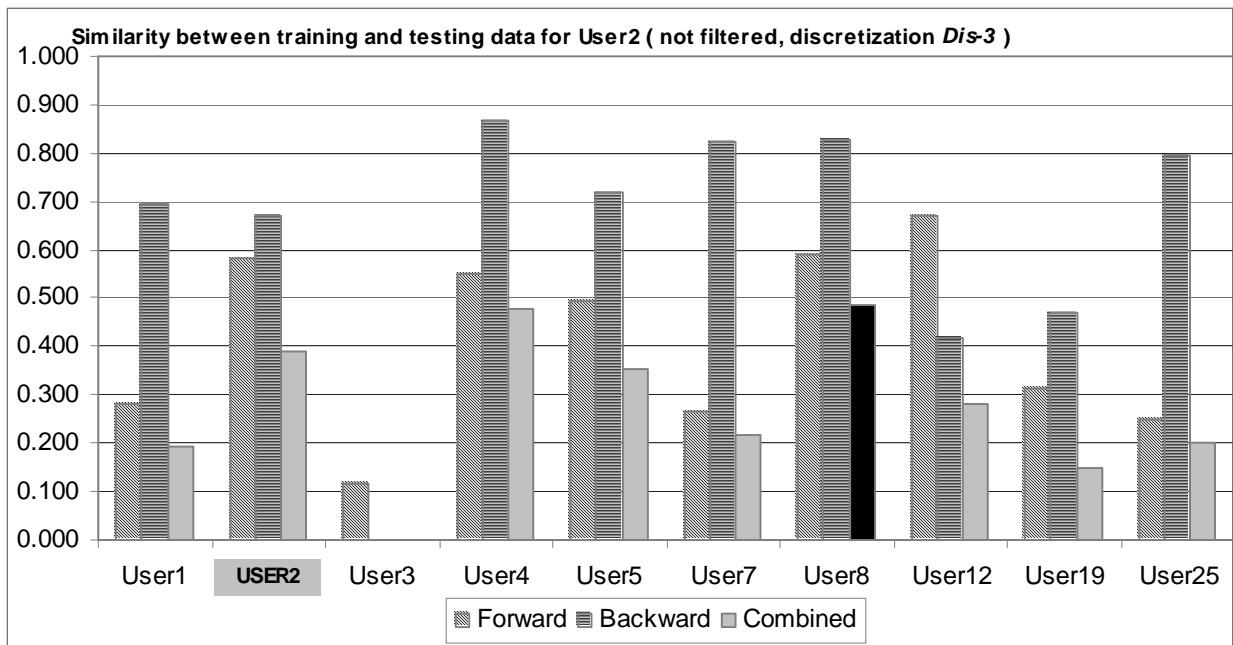


Figure 49: Self- and cross-similarity of the training and testing data for User 2.

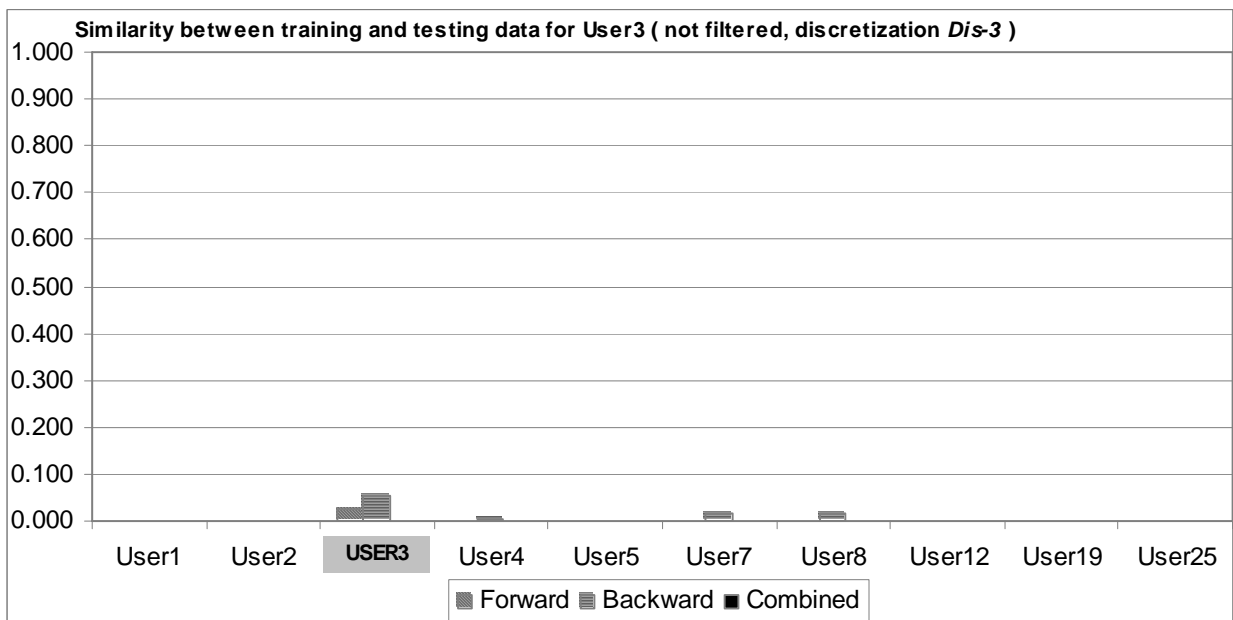


Figure 50: Self- and cross-similarity of the training and testing data for User 3.

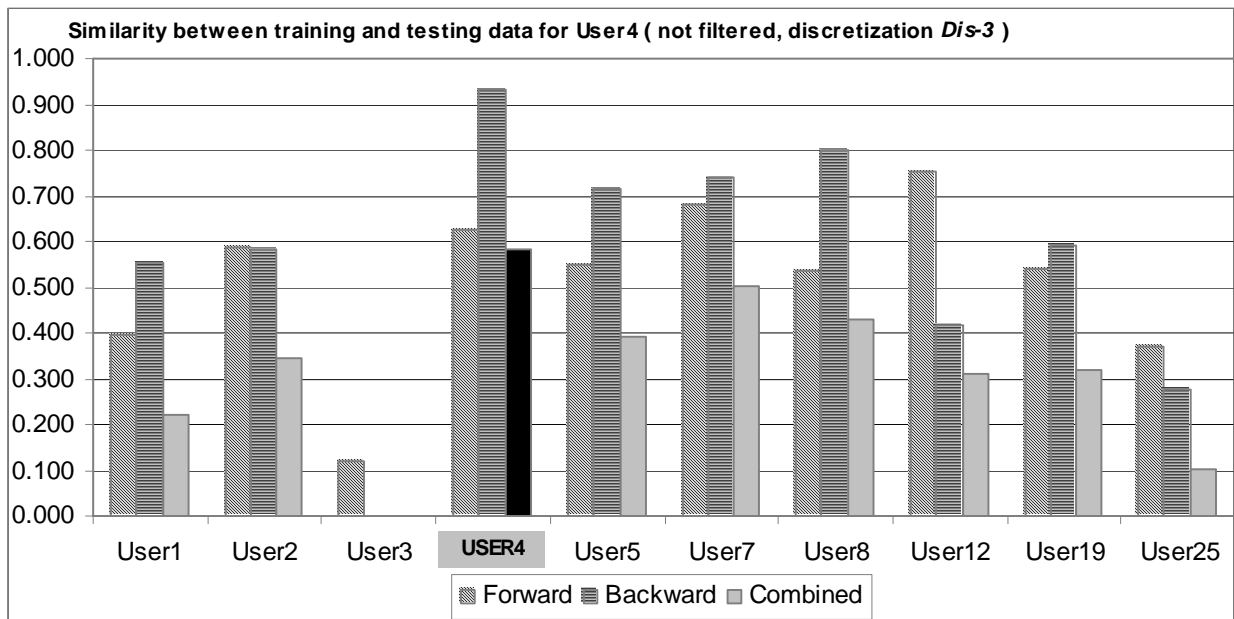


Figure 51: Self- and cross-similarity of the training and testing data for User 4.

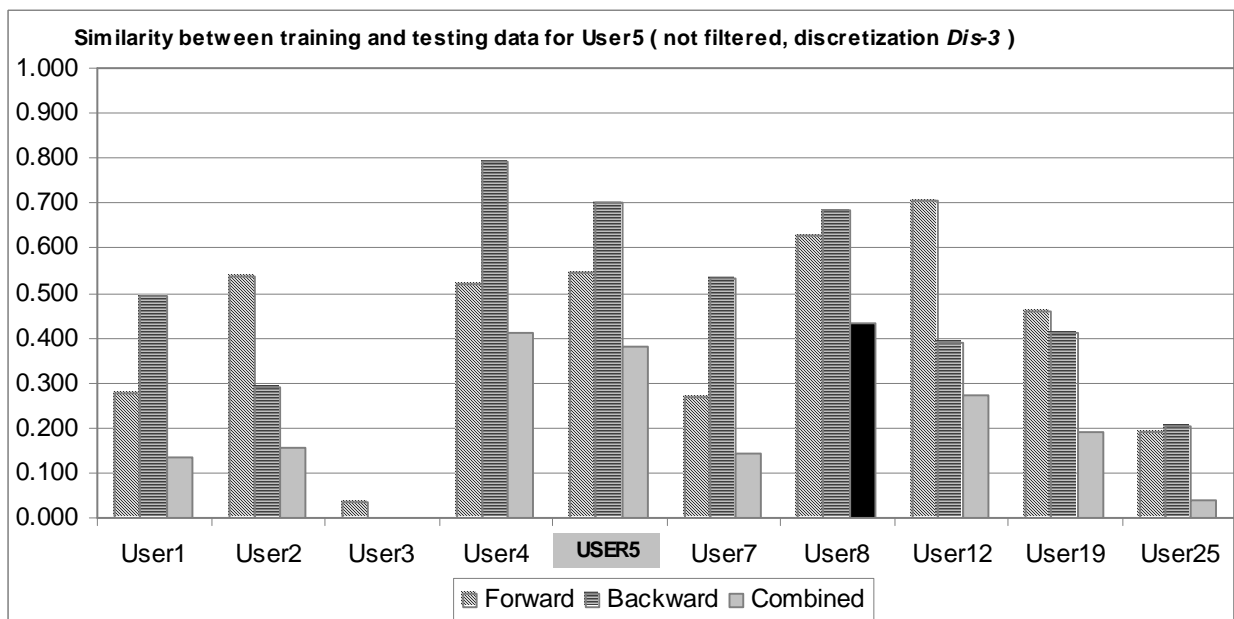


Figure 52: Self- and cross-similarity of the training and testing data for User 5.

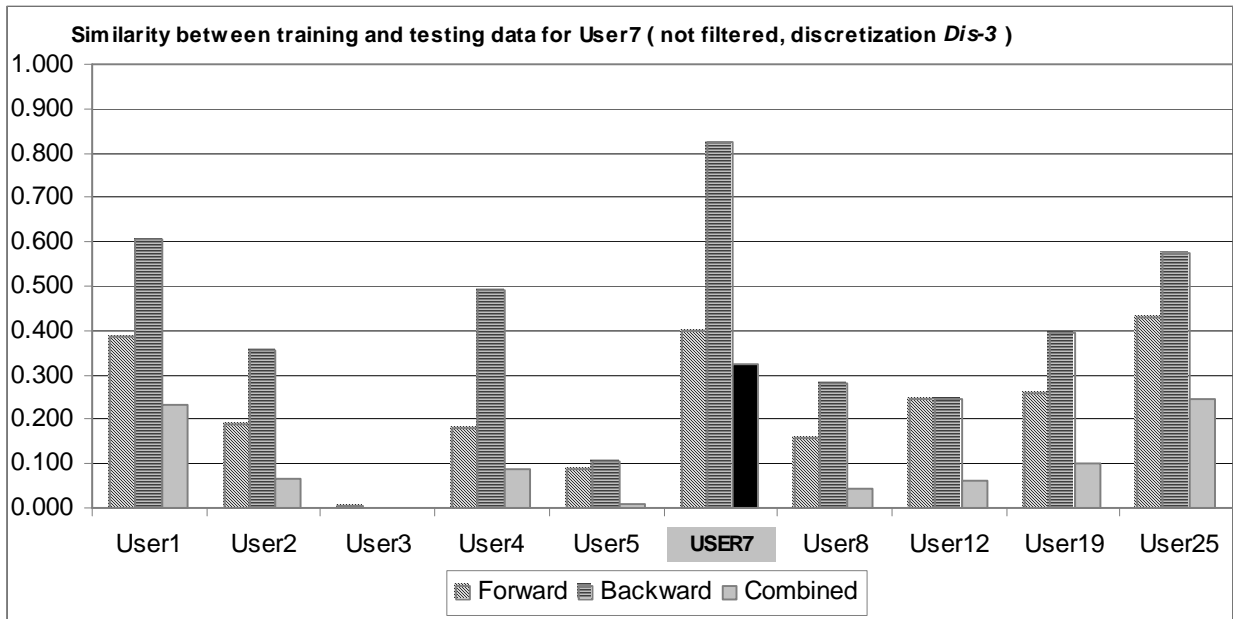


Figure 53: Self- and cross-similarity of the training and testing data for User 7.

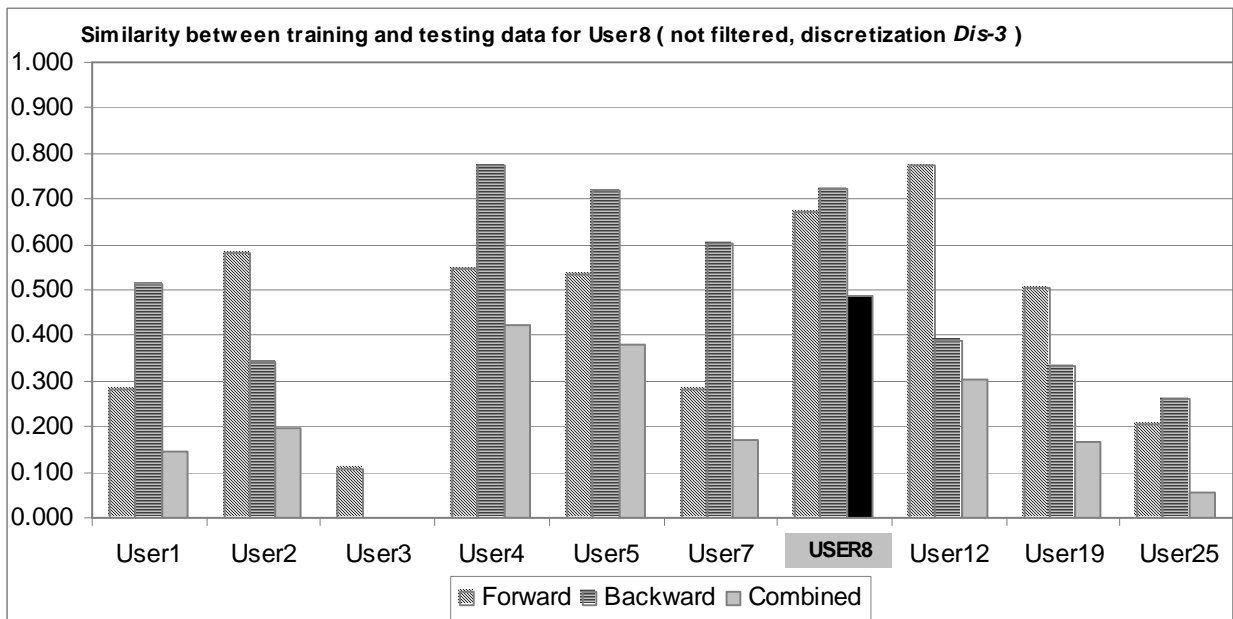


Figure 54: Self- and cross-similarity of the training and testing data for User 8.

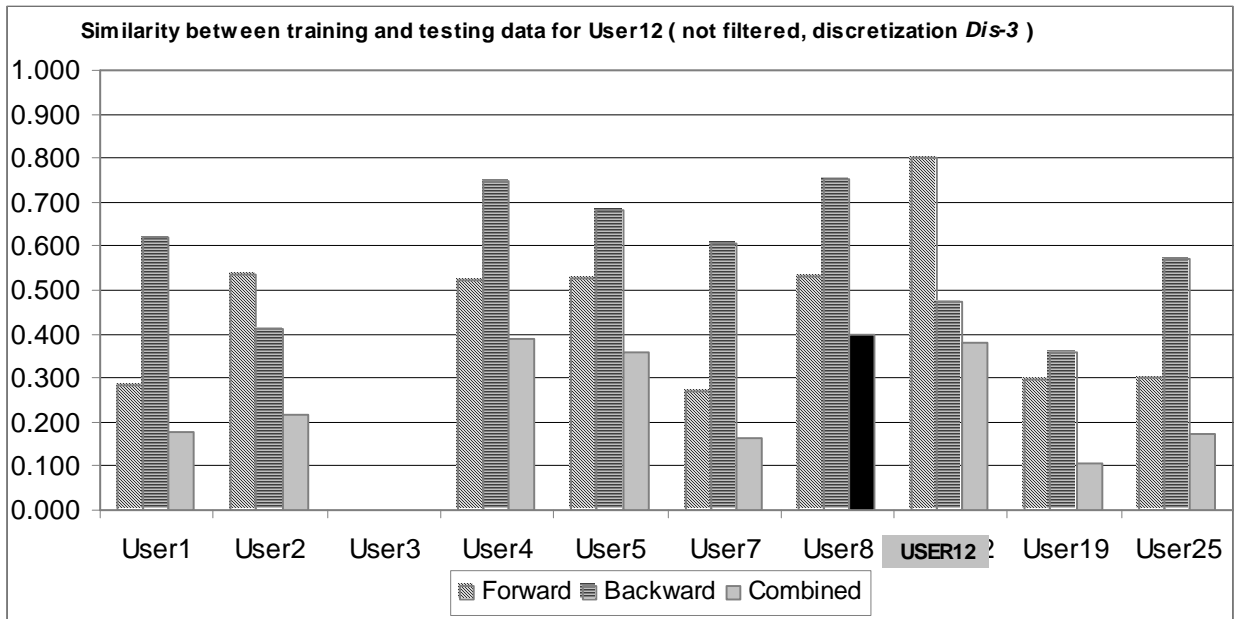


Figure 55: Self- and cross-similarity of the training and testing data for User 12.

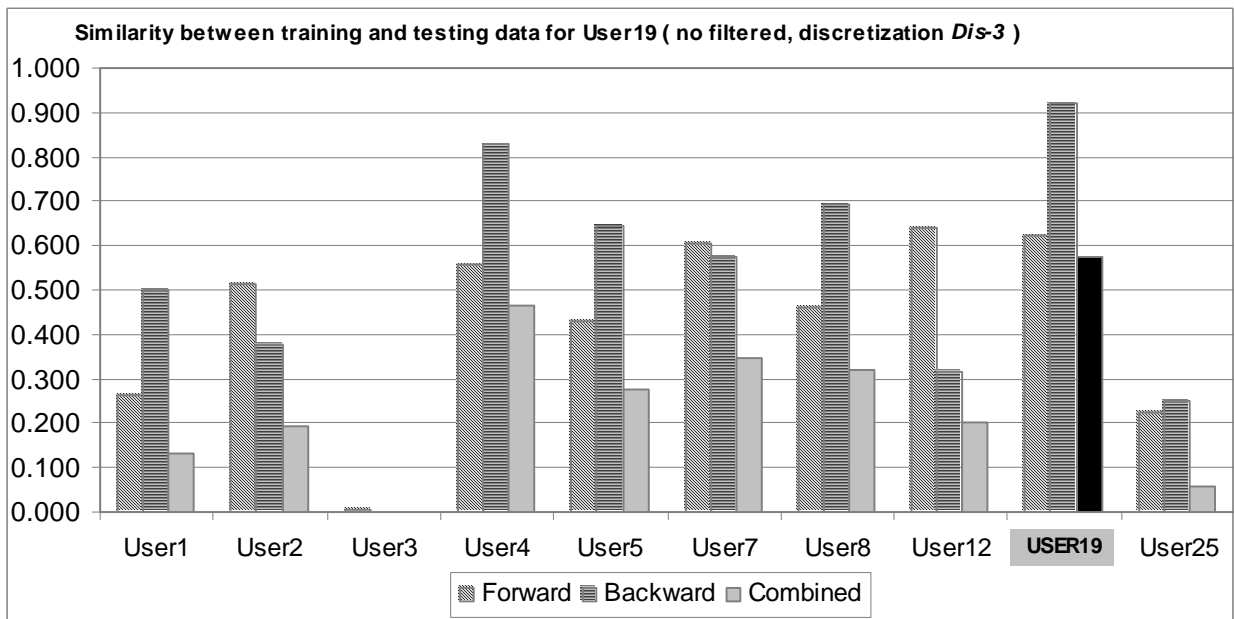


Figure 56: Self- and cross-similarity of the training and testing data for User 19.

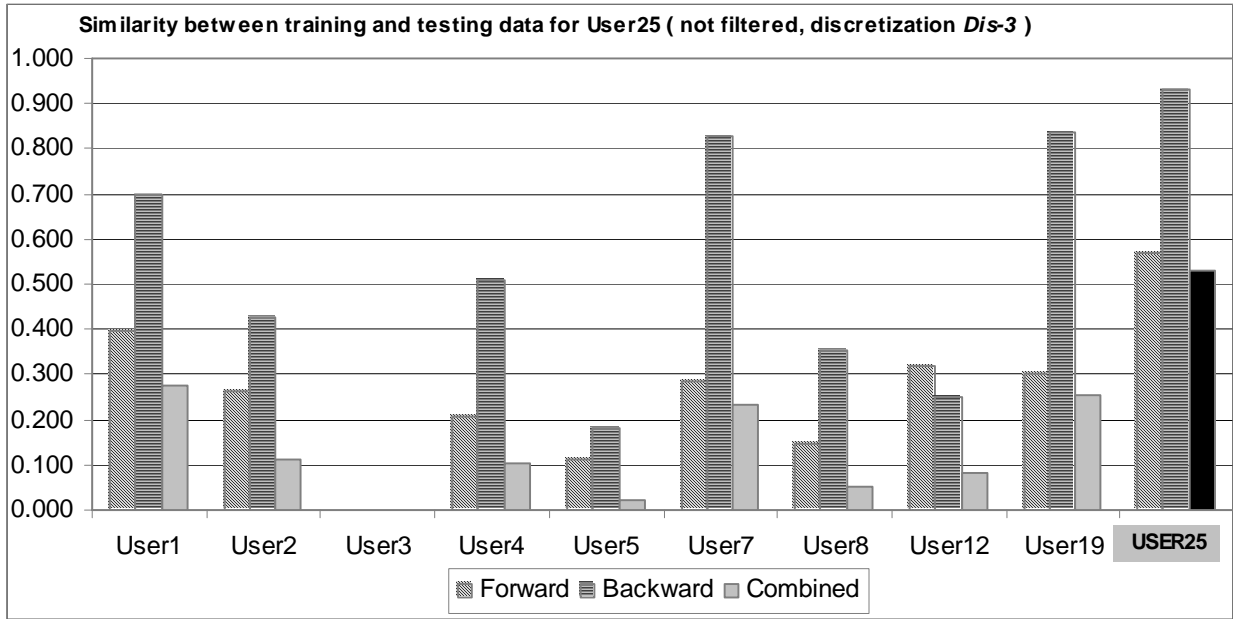


Figure 57: Self- and cross-similarity of the training and testing data for User 25.

8.2 Event Selection Experiments

Event selection experiments were performed in two phases. The goal of Phase 1 was to estimate the advantage that could be provided by event selection. Hence, as input to the selection algorithm, both training and testing data were used. The goal of Phase 2 was to prepare data for learning; therefore, selection was based on training data only. For this phase, the training data set was therefore divided into two parts: the first five and the second five sessions. Only the events that occur in both parts would be selected under this schema. In this set of experiments, different variants of selection methods were executed.

Phase 1

The selection algorithm used 4-grams of the following six attributes: process_name, win_opened, prot_words_chars, delta_time_new_window, proc_count_in_win_lf, and win_title_prot_words. Significance was calculated using the sig2 function ($sig = comm * dist$), and ratio n_u^e / N_u^e was calculated using negative schema n1. Significance values for the best 20 events for each user are presented in Figure 58. Because different users' significance values are different, it is difficult to choose one significance value threshold. Therefore, significance-rank-based selection criterion was used in these experiments; k=6, 10 and 14 best events were chosen.

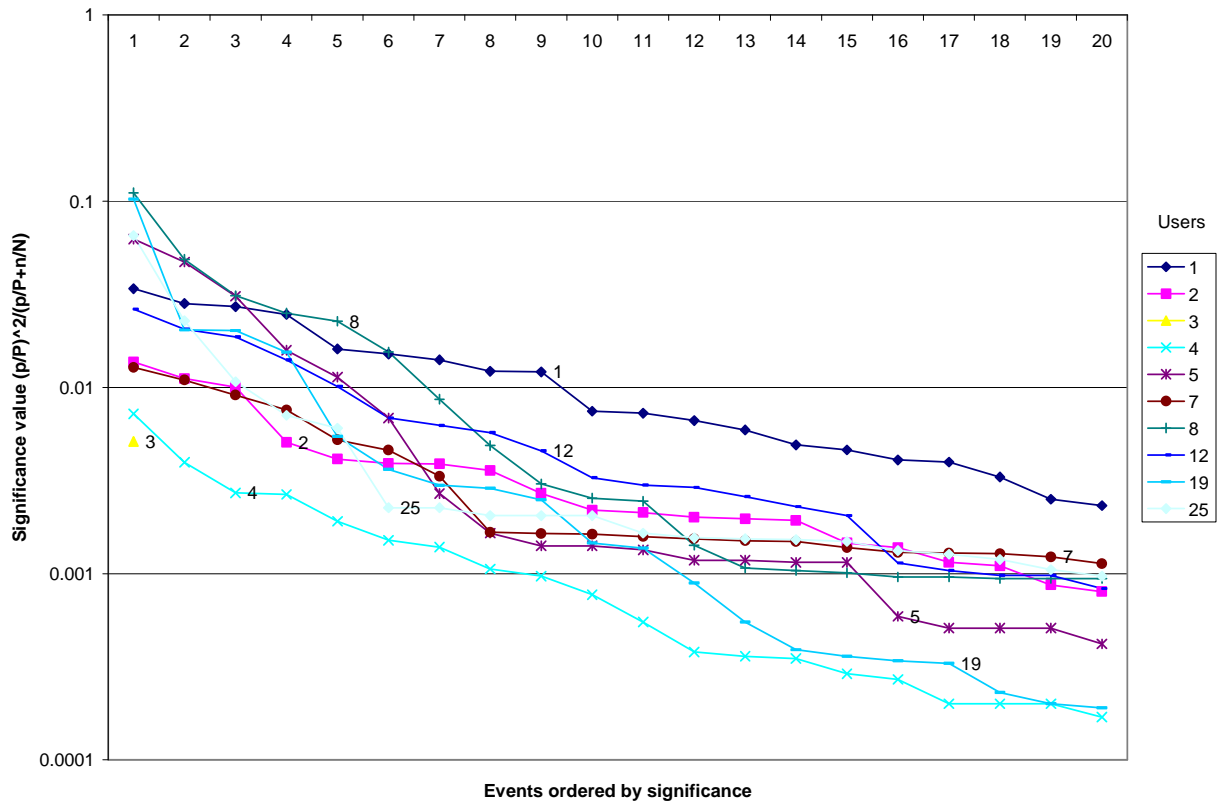


Figure 58: Significance for events with rank from 1 to 20 for all users, computed using training and testing data (logarithmic scale)

User	Number of events	Events selected using significance sig6, negative schema nmax, k best events							
		k=10		k=20		k=60		k=100	
		No	%	No	%	No	%	No	%
1	658	241	37%	267	41%	329	50%	329	50%
2	13898	5933	43%	6495	47%	6803	49%	7095	51%
3	52	26	50%	26	50%	26	50%	26	50%
4	56612	23791	42%	24674	44%	28255	50%	28357	50%
5	8876	3590	40%	3738	42%	4365	49%	4511	51%
7	4954	1553	31%	1762	36%	2388	48%	2566	52%
8	8502	3676	43%	3856	45%	4161	49%	4341	51%
12	15227	5954	39%	7097	47%	7505	49%	7722	51%
19	10964	4804	44%	5128	47%	5333	49%	5631	51%
25	27625	10354	37%	11751	43%	13642	49%	13983	51%
All	147368	59922	41%	64794	44%	72807	49%	74561	51%

Table 38: Number of selected events for all users (using significance sig6, negative schema nmax, and significance-rank-based selection criterion with k=10, 20, 60, 100)

Next, disjunctive filtering was also applied using the same attributes. In this case, the significance-rank-based selection criterion was also used with $k=6$ and 10. We found that filtering for $k=10$ is too weak; all events were selected. Learning results for $k=6$ were worse than these for conjunctive filtering with $k=10$, therefore in all subsequent experiments, only conjunctive filtering was used.

Phase 2

The selection algorithm was executed with many combinations of variants of negative schema, significance definition, number of events selected (10, 20, 60, 100). Numbers of selected events for chosen parameters are presented in Table 38.

8.3 Output Value Selection for Prediction-based Model

The primary output attribute used for prediction-based experiments was `process_name`. We used frequencies of values of the attribute to select values for which the models are built. The frequencies are presented graphically in Figures 59-68. Results of prediction-based learning and testing are presented in Sections 8.4.11 and 8.4.12.

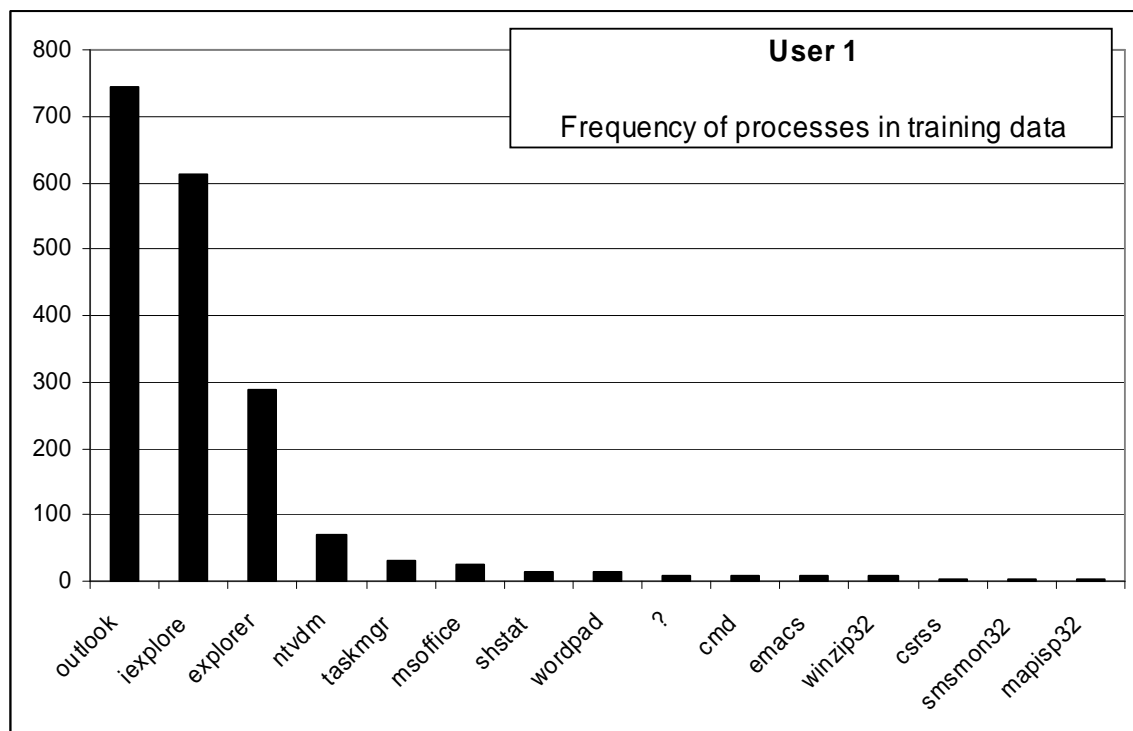


Figure 59: Frequency of processes for User 1

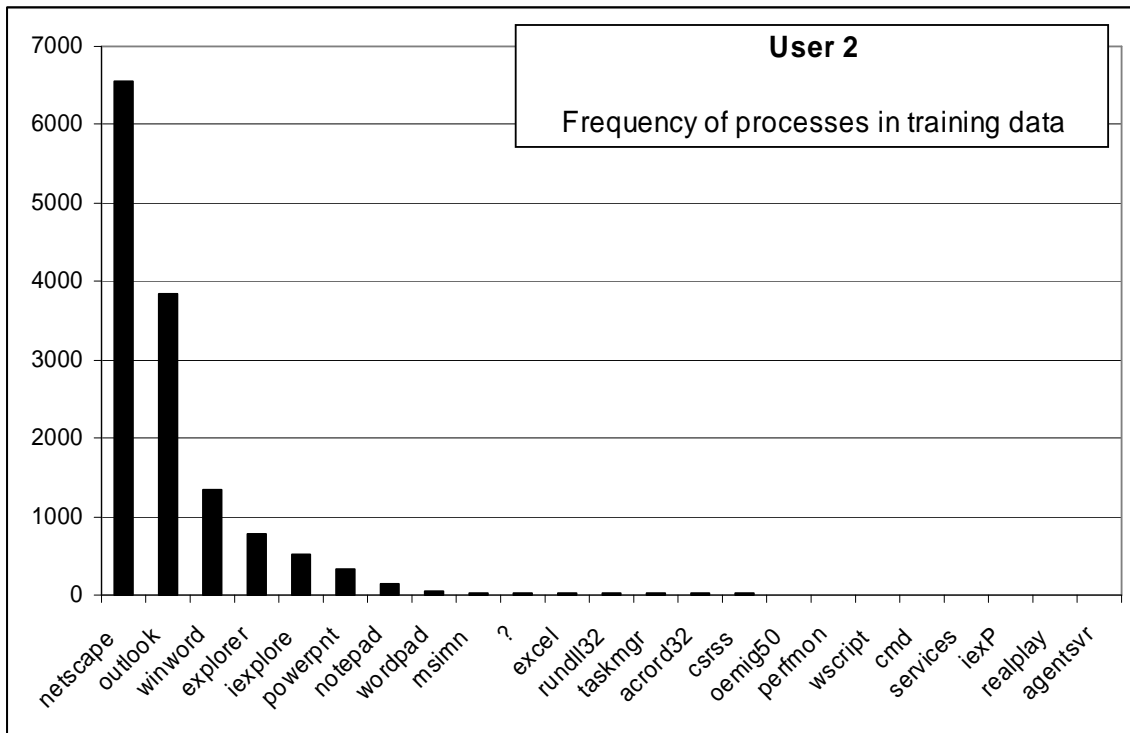


Figure 60: Frequency of processes for User 2

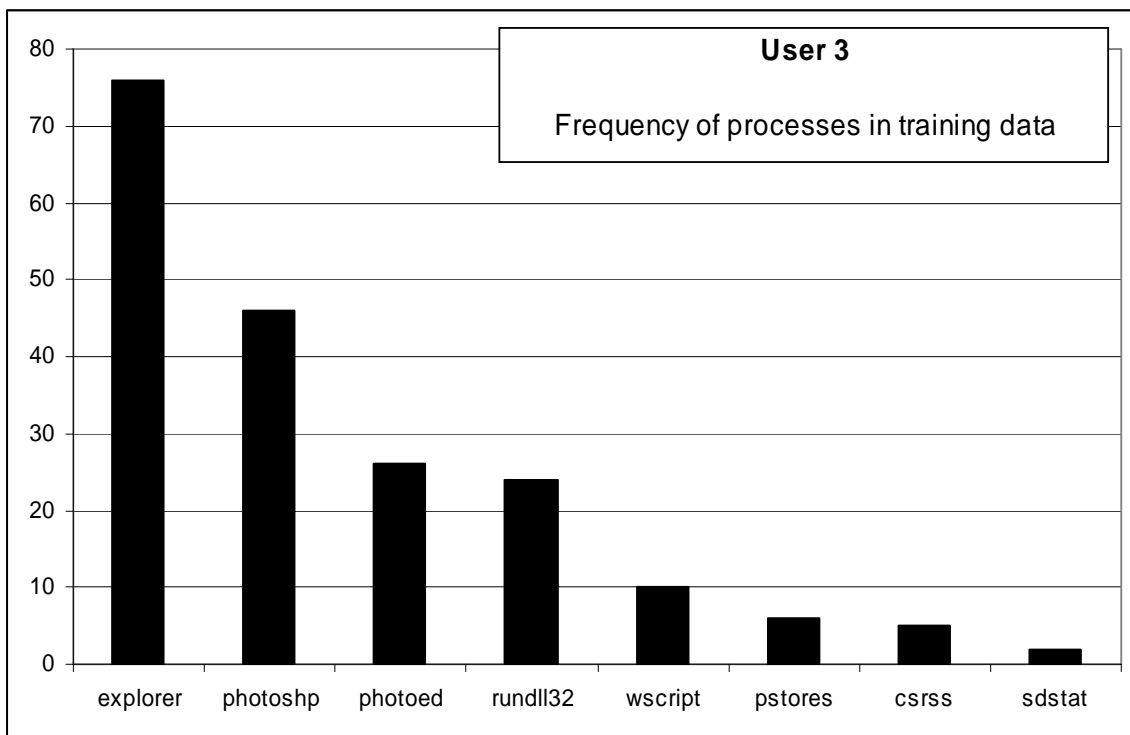


Figure 61: Frequency of processes for User 3

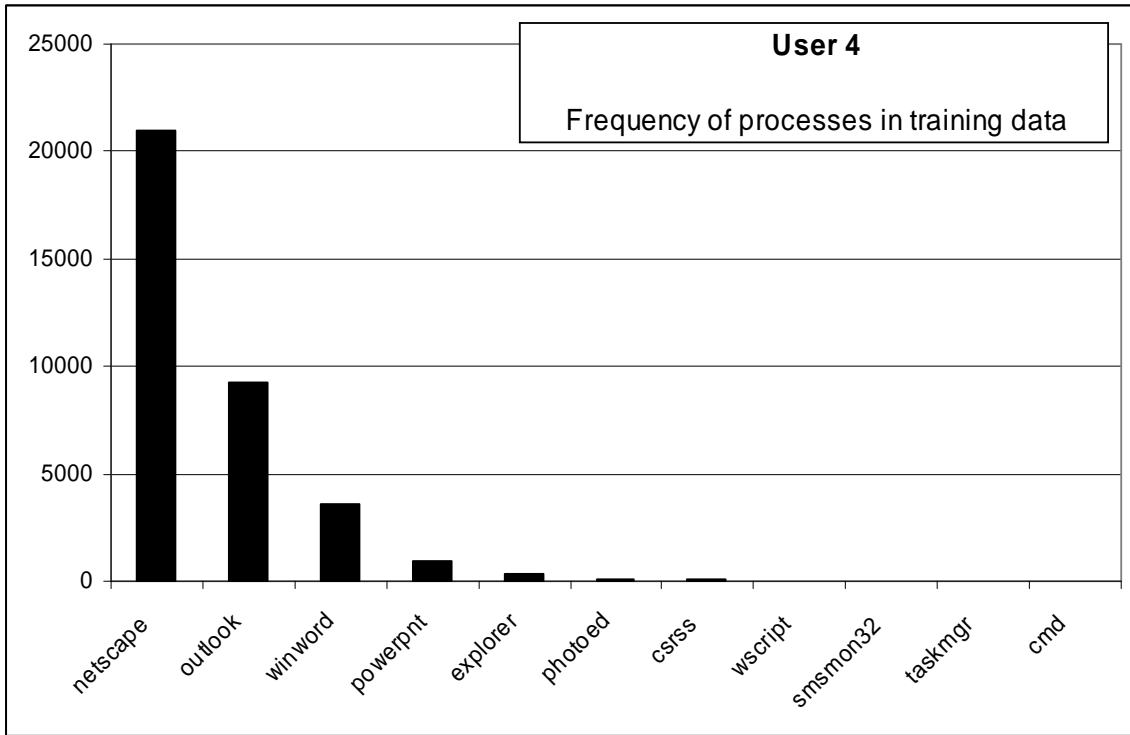


Figure 62: Frequency of processes for User 4

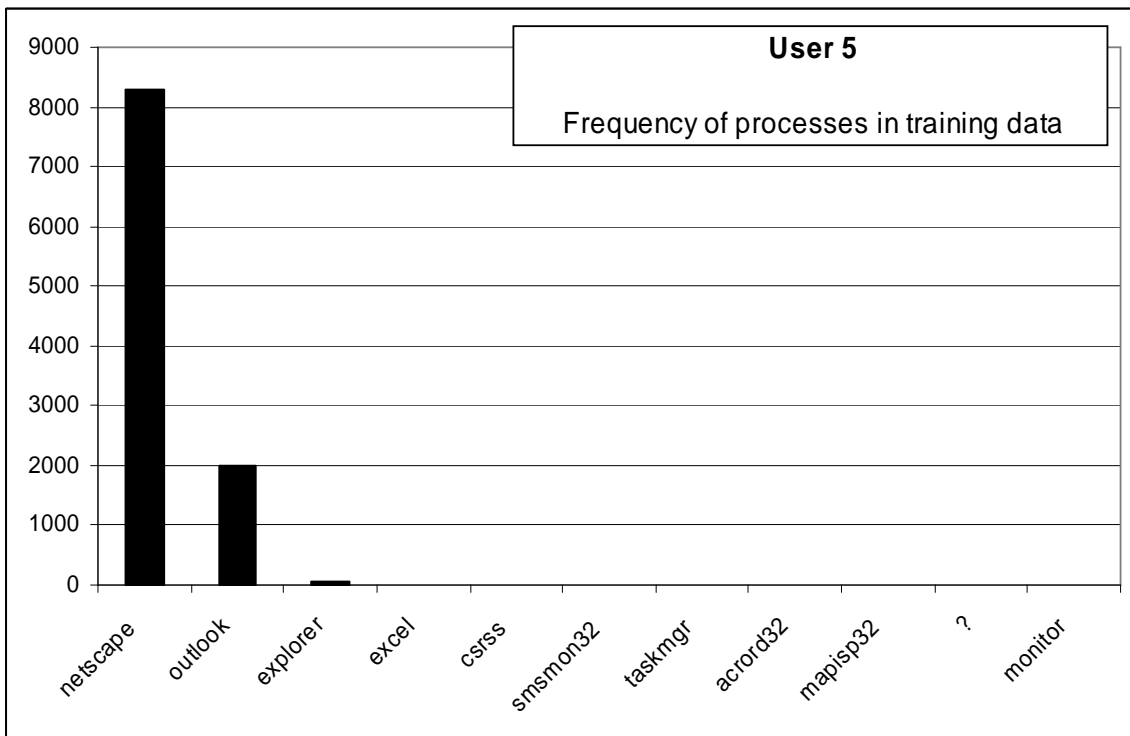


Figure 63: Frequency of processes for User 5

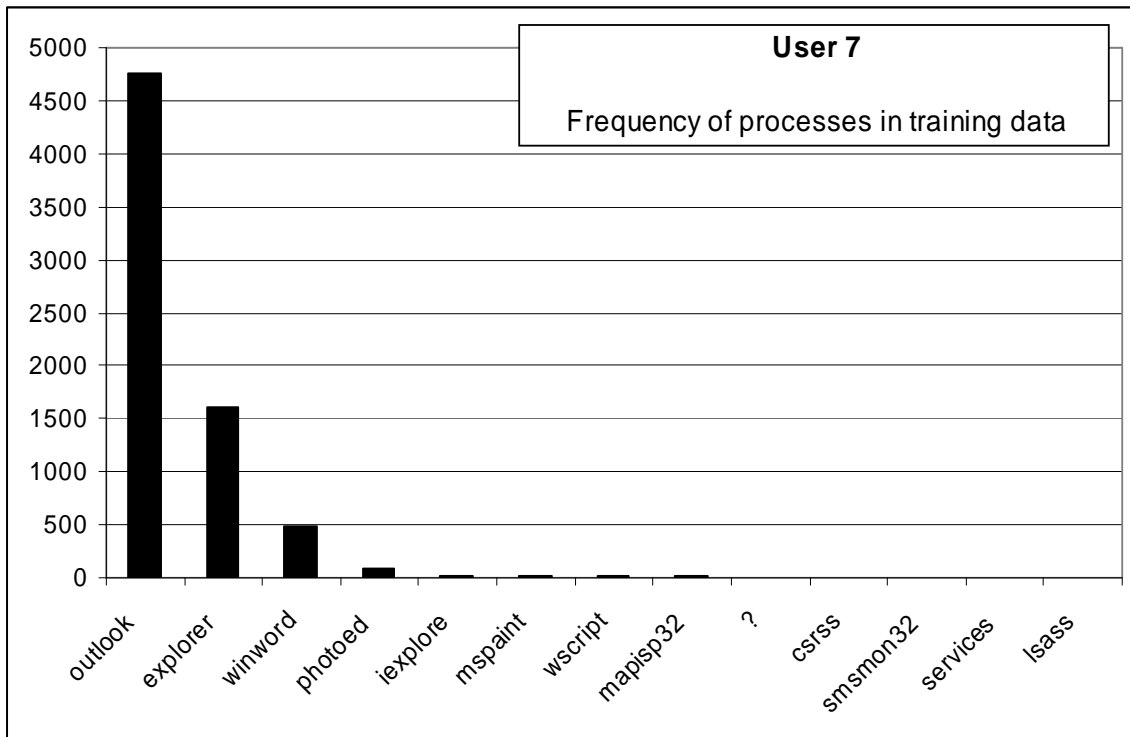


Figure 64: Frequency of processes for User 7

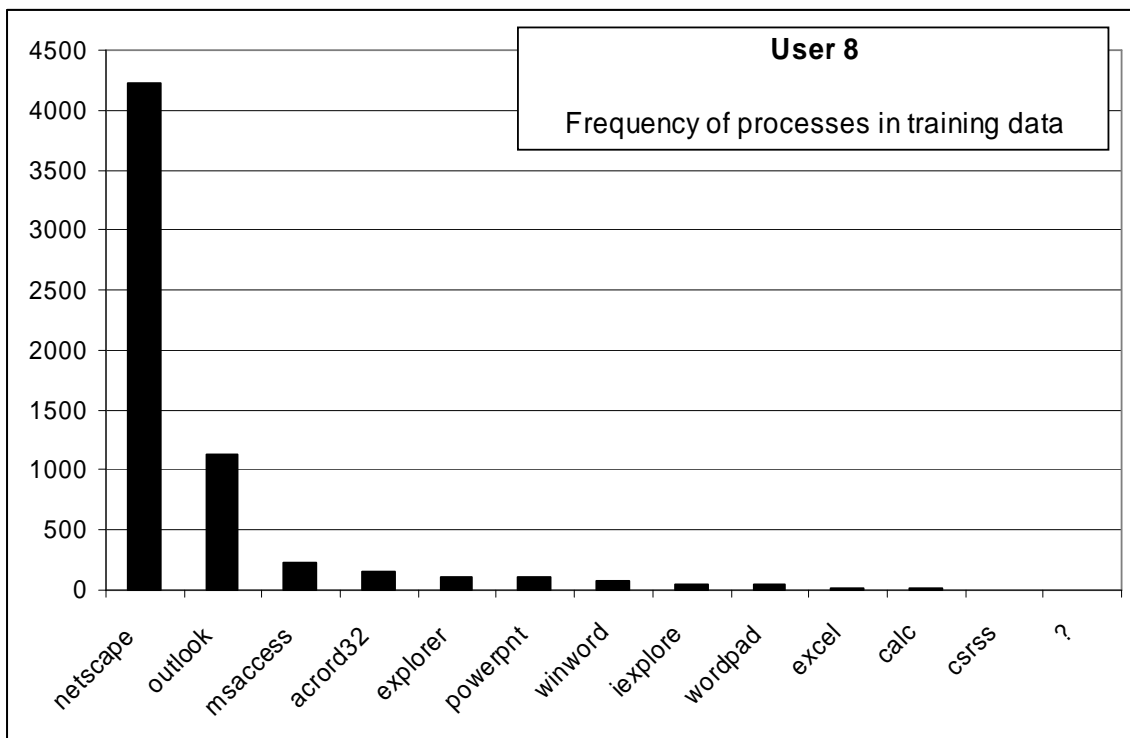


Figure 65: Frequency of processes for User 8

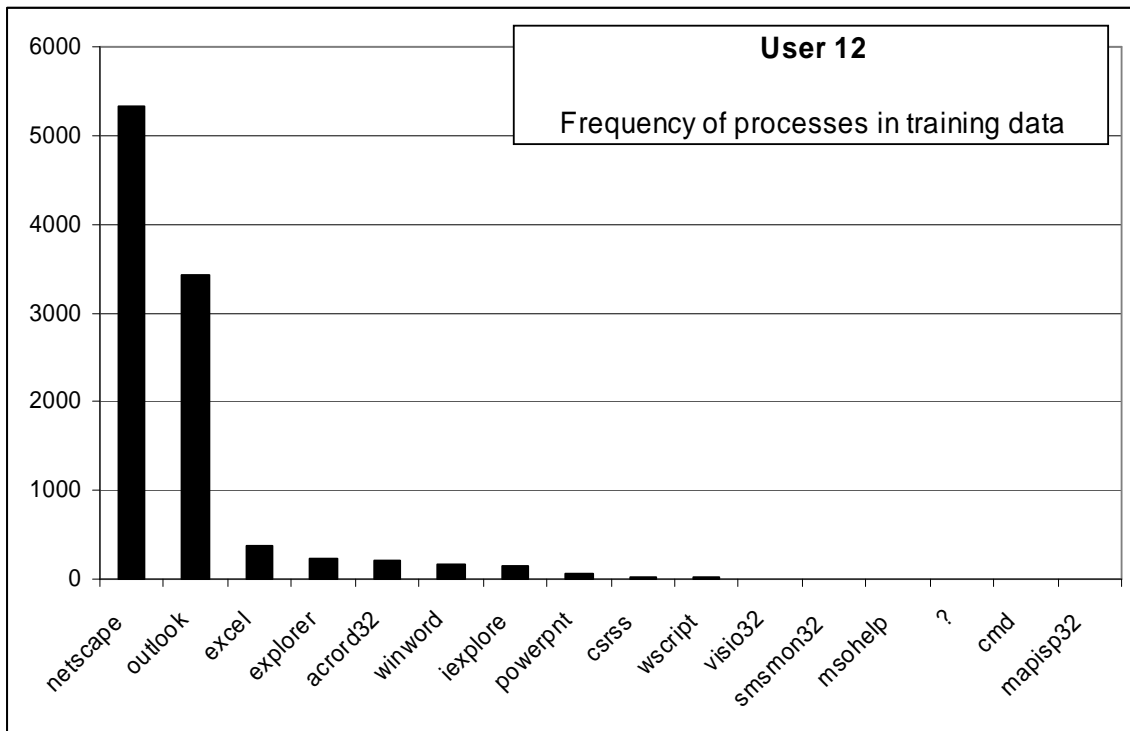


Figure 66: Frequency of processes for User 12

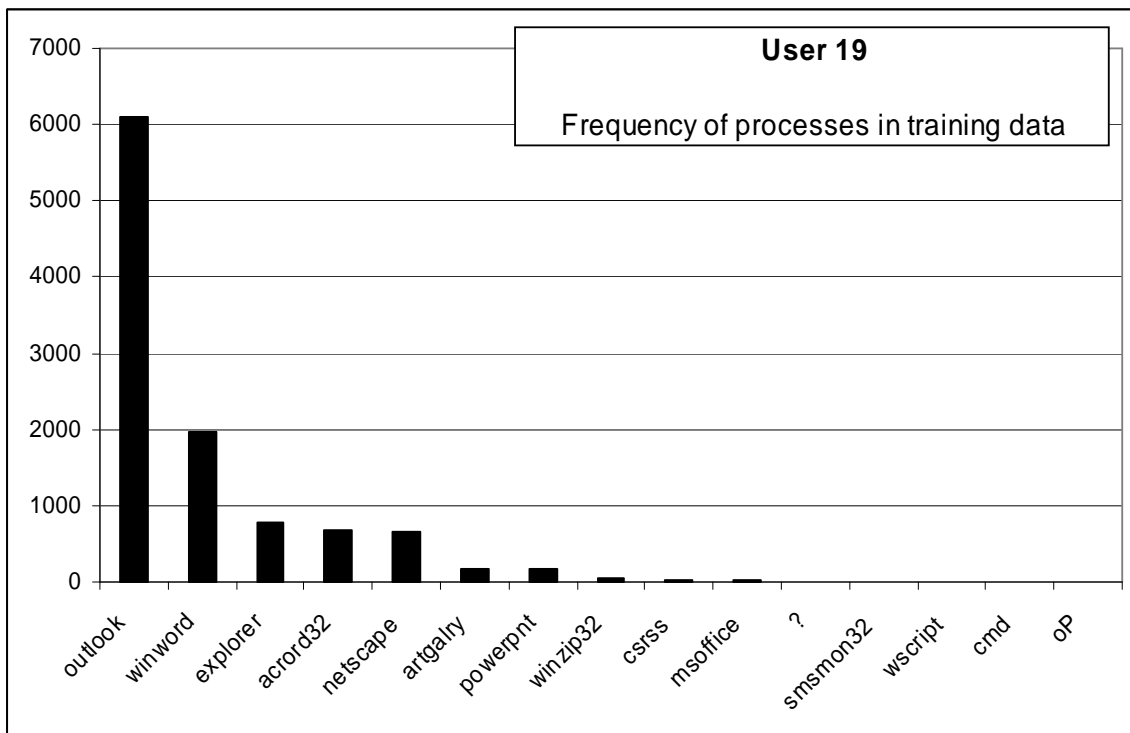


Figure 67: Frequency of processes for User 19.

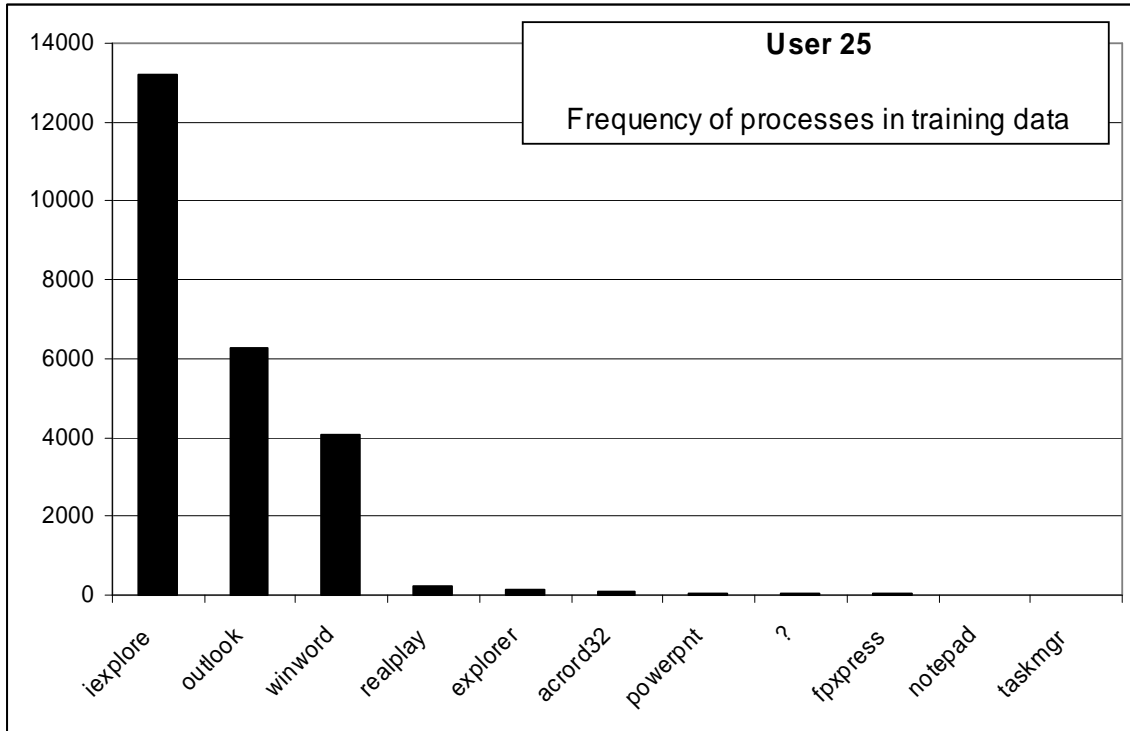


Figure 68: Frequency of processes for User 25.

Based on the above figures we selected output values of attribute process_name presented in Table 39.

User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
explorer	netscape	Explorer	netscape	netscape	explorer	netscape	netscape	outlook	iexplore
iexplore	outlook	photoshp	outlook	outlook	outlook		outlook	winword	outlook
outlook									winword

Table 39: Selected values of output attribute for prediction-based model experiments

The above values were selected manually, but in general modification of a PROMISE or Gain Ratio method can be used to automate process of the selection.

8.4 AQ21 Experiments with Data from 10 Users, 10+5 Sessions

These experiments used 10 sessions for training sessions, and 5 testing sessions from each user. The testing sessions were those that followed the training session in time. The purpose of the experiments was to investigate various combinations of AQ21 learning and testing parameters on datasets prepared using different filtering schemas. Data was discretized using the Dis-3 schema described in Section 5.2.2. Because these introduced ideas and novel methods opened a possibility for a very large number of lines of inquiry and different experiments, the experiments actually performed during this research period spanned only a subset of potential experiments and were limited to learning and testing multi-state user models.

8.4.1 *Experiment 040607-1: Filtered Data TR+TS, Discriminant Descriptions*

Training Dataset:

Discretization: Dis-3

Filtering: Significance based, conjunctive, rank-threshold = 10, TR+TS

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

AQ21 Learning Parameters:

maxstar = 1 maxrule = 1 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Discriminant descriptions

Testing Parameters:

Evaluation of Conjunction = strict

Evaluation of Disjunction = max

Acceptance Threshold = 10%

Accuracy Tolerance = 5%

Learning Results:

Total number of rules: 71

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	8	8	1	10	8	6	8	9	8	5

Table 40: Number of learned rules for 10 users

Testing Results:

Correct: 79.17%

Precision: 82.46%

First Choice Correct: 75%

First Choice Precision: 100%

Examples of learned rules:

```
[user=user1]
# Rule 1
<-- [process_name = explorer, outlook : 394,20140]
    [proc_count_in_win_lf = 2..3.5 : 377,21048]
    [win_title_prot_words = 3 : 269,14005]
    : p=160,u=98,cx=23

# Rule 2
<-- [process_name = explorer, ntvdm : 148,629]
    [proc_count_in_win_lf = 0..3.5 : 470,23150]
    [win_title_prot_words = 0..1 : 195,37455]
    : p=86,u=86,cx=23

# Rule 3
<-- [process_name = explorer, iexplore : 176,8290]
    [proc_count_in_win_lf = 2..3.5 : 377,21048]
    [win_title_prot_words = 3 : 269,14005]
    [win_title_prot_words-3 = 3 : 269,14007]
    : p=140,u=78,cx=30

# Rule 4
<-- [process_name = outlook : 296,19511]
    [proc_count_in_win_lf = 2..3.5 : 377,21048]
    [win_title_prot_words = 1 : 145,37350]
    : p=57,u=57,cx=21

[user=user2]
# Rule 1
<-- [process_name = netscape : 3083,22218]
    [prot_words_chars = 7.5..8.5 : 3083,21559]
    [proc_count_in_win_lf = 2..3.5 : 2577,18848]
    [proc_count_in_win_lf-2 = 2..3.5 : 2577,18878]
    [proc_count_in_win_lf-3 = 2..3.5 : 2577,18760]
    [win_title_prot_words = 1 : 3527,33968]
    : p=1731,u=1731,cx=42

# Rule 2
<-- [process_name = netscape : 3083,22218]
    [prot_words_chars = 7.5..8.5 : 3083,21559]
    [proc_count_in_win_lf = 3.5..4.6 : 1877,15967]
    : p=1352,u=1352,cx=21

# Rule 3
<-- [process_name = outlook : 849,18958]
    [proc_count_in_win_lf = 2..3.5 : 2577,18848]
    [win_title_prot_words = 3 : 611,13663]
    : p=611,u=611,cx=21
```

Figure 69: Examples of rules learned in experiment 040606-1.

The rules presented below are examples of learned rules for the first and second users. The first rule for the first user can be interpreted in the following way: User is User 1 if: it uses explorer or outlook and if logarithm of number of processes in current window is between 2 and 3.5 and if number of protected words in window title is 3. Numbers in parentheses represent positive and negative examples that satisfy a condition. For instance in condition

[process_name = explorer, outlook : 394,20140]

there are 394 positive and 20140 negative examples for class user = user1 in the training data. It can be seen that all conditions in the first rule cover negative examples, but their conjunction does not. Parameters displayed after each rule consist of the following values: p denotes the number of covered positive examples, u denotes the number of positive examples covered only by the rule (unique coverage), and cx denotes the complexity of the rule. Attributes used are described in Section 2. For instance there are 160 positive examples satisfying the first rule, 98 out of the examples are covered uniquely. Complexity of the first rule is 23.

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
First Ch. Correct	100%	100%	67%	80%	100%	80%	40%	100%	60%	100%

Table 41: Summary of correct answers for 10 users.

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
User1 (First Choice Correct: 100%)										
Epi.281	0.714	0.340	0.038	0.492	0.327	0.218	0.241	0.499	0.448	0.621
Epi.282	0.541	0.250	0.080	0.218	0.090	0.216	0.231	0.197	0.168	0.138
Epi.283	0.660	0.174	0.056	0.576	0.076	0.535	0.576	0.444	0.347	0.076
Epi.284	0.836	0.415	0.049	0.552	0.415	0.175	0.246	0.623	0.470	0.415
Epi.285	0.610	0.330	0.023	0.509	0.307	0.260	0.258	0.458	0.447	0.395
User2 (First Choice Correct: 100%)										
Epi.288	0.137	0.715	0.031	0.557	0.583	0.061	0.621	0.562	0.466	0.078
Epi.289	0.471	0.680	0.022	0.466	0.444	0.088	0.286	0.451	0.455	0.513
Epi.290	0.087	0.582	0.020	0.261	0.341	0.075	0.416	0.296	0.095	0.062
Epi.291	0.233	0.681	0.017	0.266	0.312	0.037	0.288	0.288	0.289	0.084
Epi.333	0.073	0.731	0.051	0.019	0.005	0.061	0.063	0.056	0.019	0.078
User3 (First Choice Correct: 67%)										
Epi.345	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Epi.347	0.000	0.000	0.286	0.000	0.000	0.143	0.000	0.000	0.000	0.000
Epi.349	0.000	0.000	*0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
User4 (First Choice Correct: 80%)										
Epi.391	0.198	0.719	0.007	0.843	0.793	0.160	0.874	0.829	0.423	0.076
Epi.392	0.459	0.415	0.011	0.611	0.385	0.303	0.553	0.660	0.566	0.131
Epi.393	0.252	0.366	0.007	0.870	0.634	0.269	0.513	0.588	0.637	0.151
Epi.394	0.222	0.278	0.119	*0.064	0.000	0.127	0.222	0.095	0.000	0.071
Epi.512	0.259	0.523	0.008	0.798	0.749	0.313	0.572	0.602	0.512	0.110
User5 (First Choice Correct: 100%)										
Epi.513	0.024	0.490	0.010	0.775	0.958	0.021	0.882	0.835	0.277	0.000
Epi.514	0.062	0.581	0.024	0.772	0.884	0.014	0.864	0.787	0.245	0.035

Epi.515	0.104	0.179	0.009	0.280	0.289	0.122	0.186	0.223	0.223	0.061
Epi.542	0.158	0.522	0.011	0.550	0.724	0.021	0.542	0.691	0.516	0.067
Epi.543	0.167	0.476	0.017	0.472	0.592	0.051	0.513	0.487	0.324	0.116
User7 (First Choice Correct: 80%)										
Epi.734	0.587	0.081	0.022	0.413	0.081	0.614	0.413	0.422	0.283	0.274
Epi.735	0.510	0.219	0.036	0.510	0.145	0.588	0.506	0.503	0.407	0.145
Epi.736	<u>0.442</u>	0.091	0.065	0.026	0.000	<u>*0.416</u>	0.026	0.156	0.000	0.325
Epi.737	0.163	0.044	0.020	0.785	0.638	0.912	0.191	0.163	0.669	0.044
Epi.738	0.392	0.019	0.015	0.267	0.026	0.580	0.203	0.269	0.183	0.241
User8 (First Choice Correct: 40%)										
Epi.741	<u>0.584</u>	0.239	0.041	<u>0.523</u>	<u>0.421</u>	0.209	<u>*0.312</u>	<u>0.618</u>	<u>0.620</u>	0.255
Epi.742	<u>0.395</u>	0.102	0.070	<u>0.358</u>	0.102	0.312	<u>*0.302</u>	<u>0.349</u>	0.288	0.130
Epi.743	0.152	0.442	0.031	0.587	0.577	0.037	0.780	0.606	0.206	0.021
Epi.744	0.162	0.575	0.016	0.660	0.658	0.073	0.735	0.653	0.294	0.050
Epi.897	<u>0.518</u>	0.324	0.029	<u>0.448</u>	0.411	0.212	<u>*0.434</u>	<u>0.570</u>	<u>0.697</u>	0.324
User12 (First Choice Correct: 100%)										
Epi.980	0.541	0.310	0.032	0.661	0.437	0.330	0.631	0.670	0.609	0.253
Epi.981	0.428	0.544	0.010	0.754	0.583	0.326	0.817	0.872	0.488	0.112
Epi.982	0.387	0.390	0.022	0.611	0.570	0.088	0.485	0.824	0.398	0.332
Epi.983	0.478	0.275	0.036	0.653	0.489	0.156	0.599	0.610	0.385	0.187
Epi.984	0.111	0.081	0.009	0.170	0.156	0.057	0.138	0.199	0.127	0.055
User19 (First Choice Correct: 60%)										
Epi.1040	0.521	0.422	0.026	0.510	0.844	0.099	0.151	0.932	0.917	0.422
Epi.1041	0.153	0.000	0.017	0.153	0.244	0.117	0.117	0.364	0.977	0.000
Epi.1042	0.441	0.417	0.083	0.361	0.484	0.115	0.413	<u>0.548</u>	<u>*0.516</u>	0.329
Epi.1043	0.129	0.124	0.010	0.386	0.192	0.075	0.122	0.241	0.888	0.061
Epi.1044	0.116	0.308	0.022	<u>0.777</u>	<u>0.750</u>	0.515	0.263	0.217	<u>*0.721</u>	0.053
User25 (First Choice Correct: 100%)										
Epi.1195	0.594	0.403	0.028	0.299	0.342	0.056	0.138	0.377	0.342	0.711
Epi.1196	0.433	0.176	0.014	0.295	0.141	0.163	0.174	0.282	0.278	0.661
Epi.1197	0.583	0.116	0.020	0.317	0.116	0.231	0.317	0.312	0.271	0.704
Epi.1198	0.485	0.172	0.020	0.222	0.142	0.144	0.142	0.259	0.206	0.784
Epi.1199	0.535	0.162	0.023	0.100	0.077	0.055	0.108	0.113	0.091	0.793

Table 42: Testing results for experiment 040606 (Discriminant Descriptions).

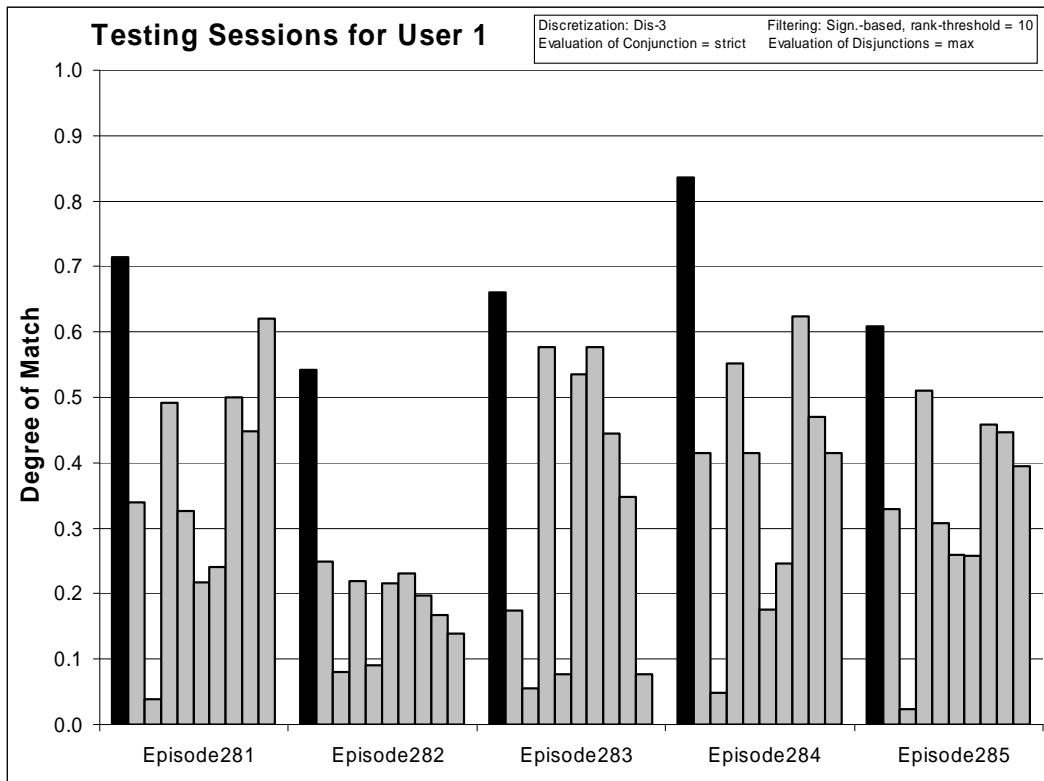


Figure 70: Degrees of match between 10 user models and 5 testing sessions from User 1.

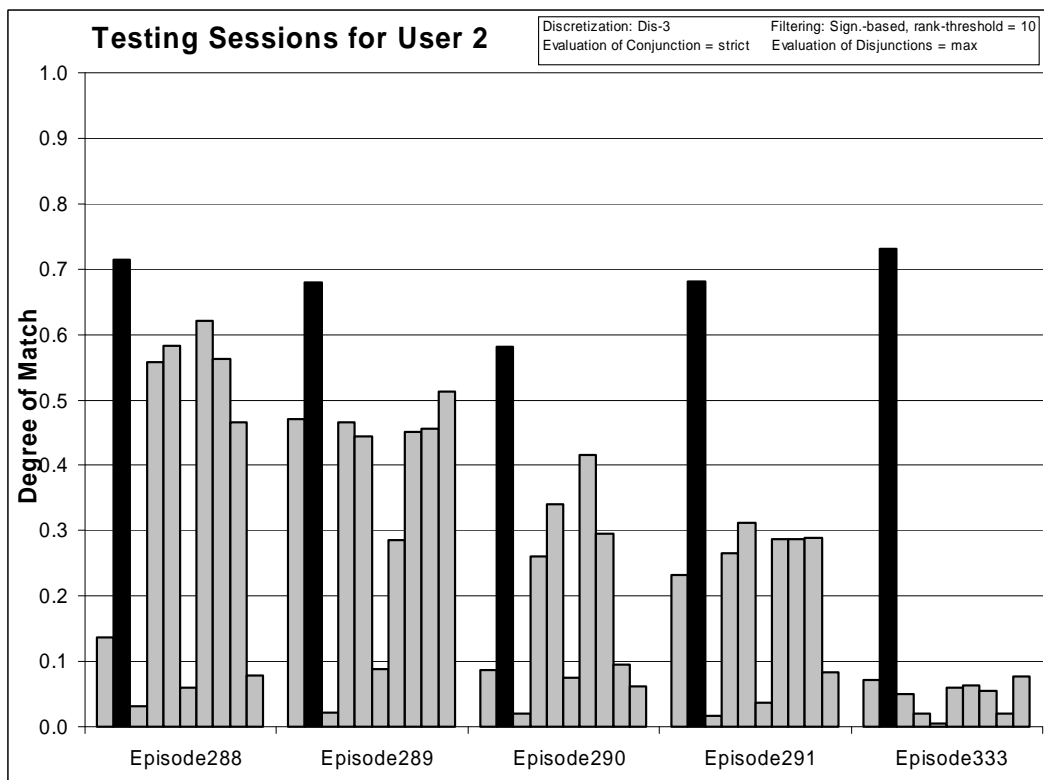


Figure 71 Degrees of match between 10 user models and 5 testing sessions from User 2.

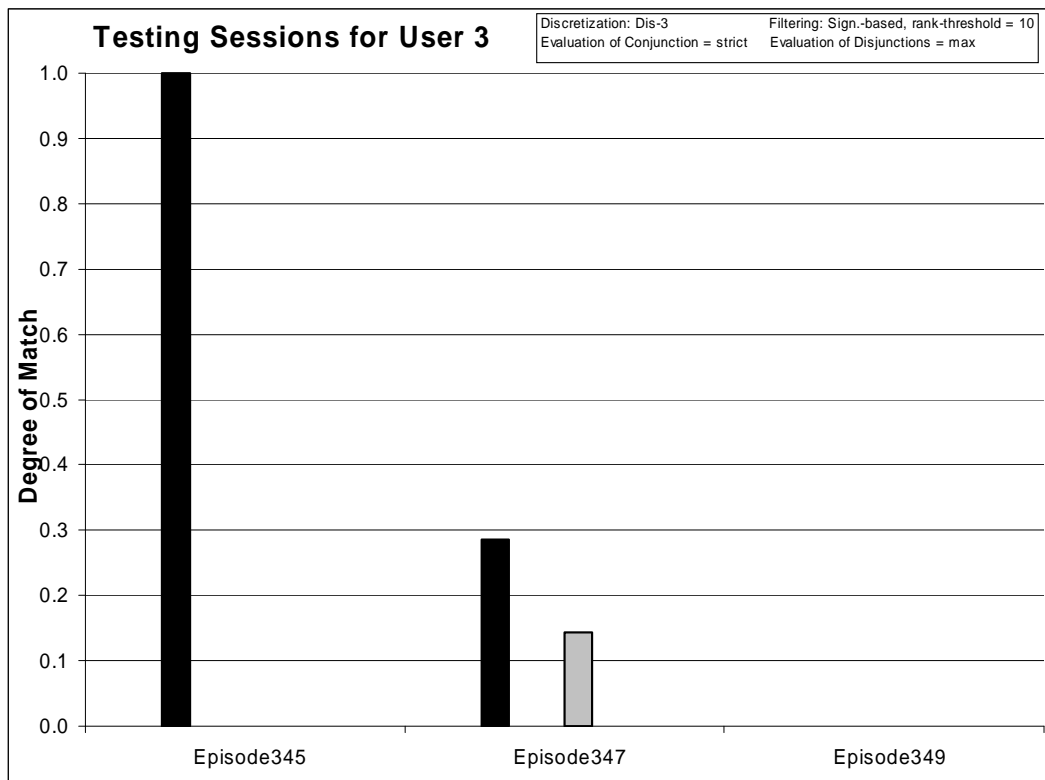


Figure 72: Degrees of match between 10 user models and 3 testing sessions from User 3.

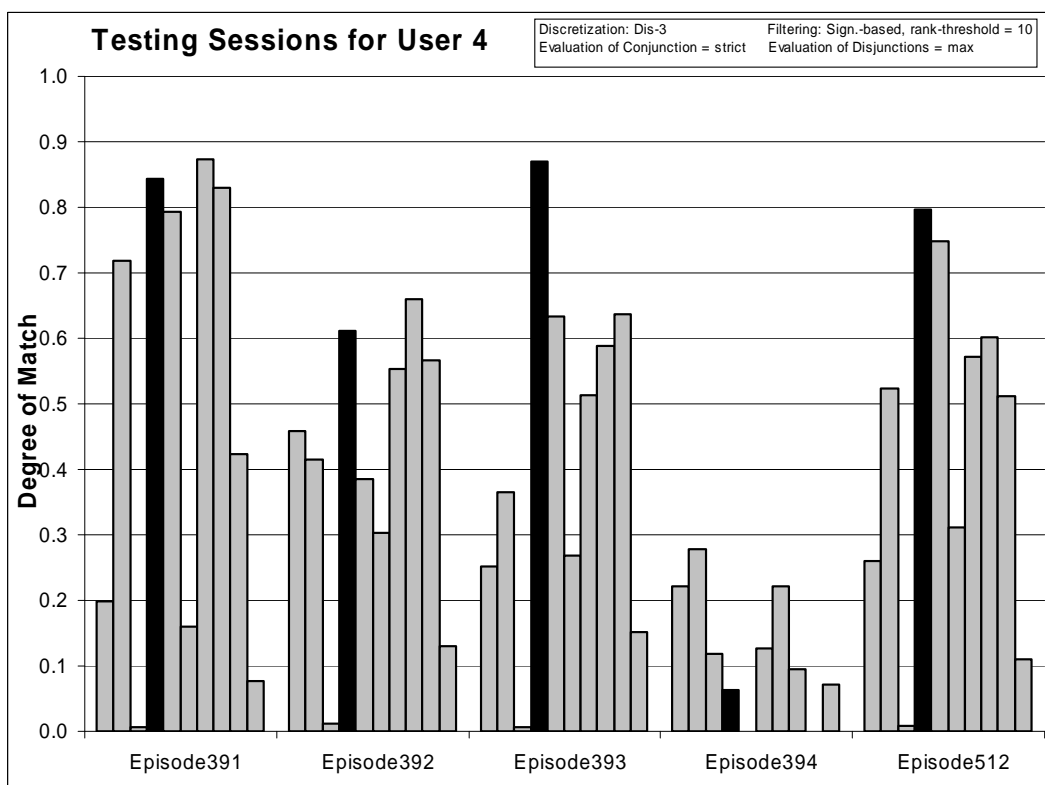


Figure 73: Degrees of match between 10 user models and 5 testing sessions from User 4.

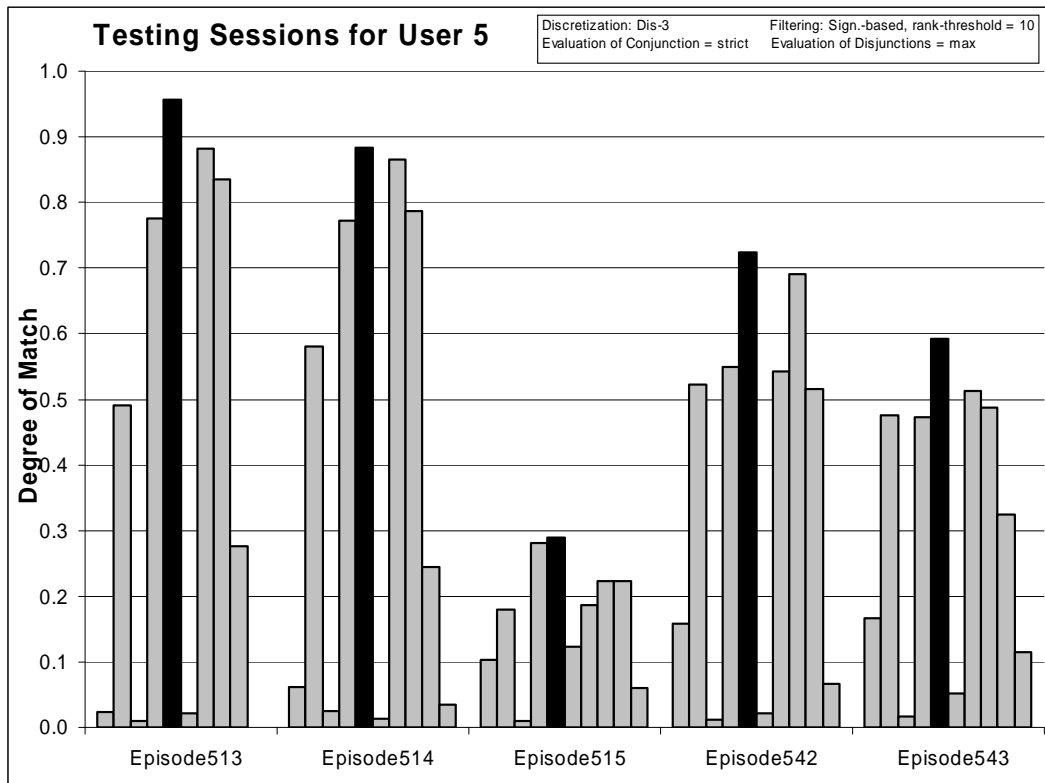


Figure 74: Degrees of match between 10 user models and 5 testing sessions from User 5.

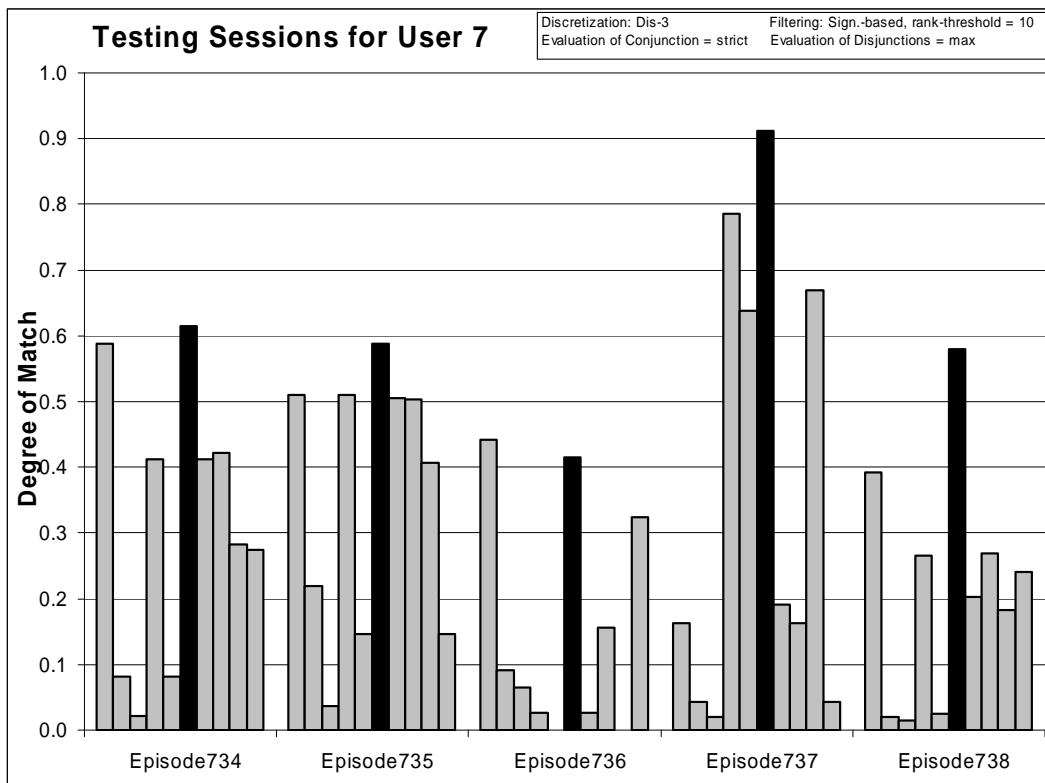


Figure 75: Degrees of match between 10 user models and 5 testing sessions from User 7.

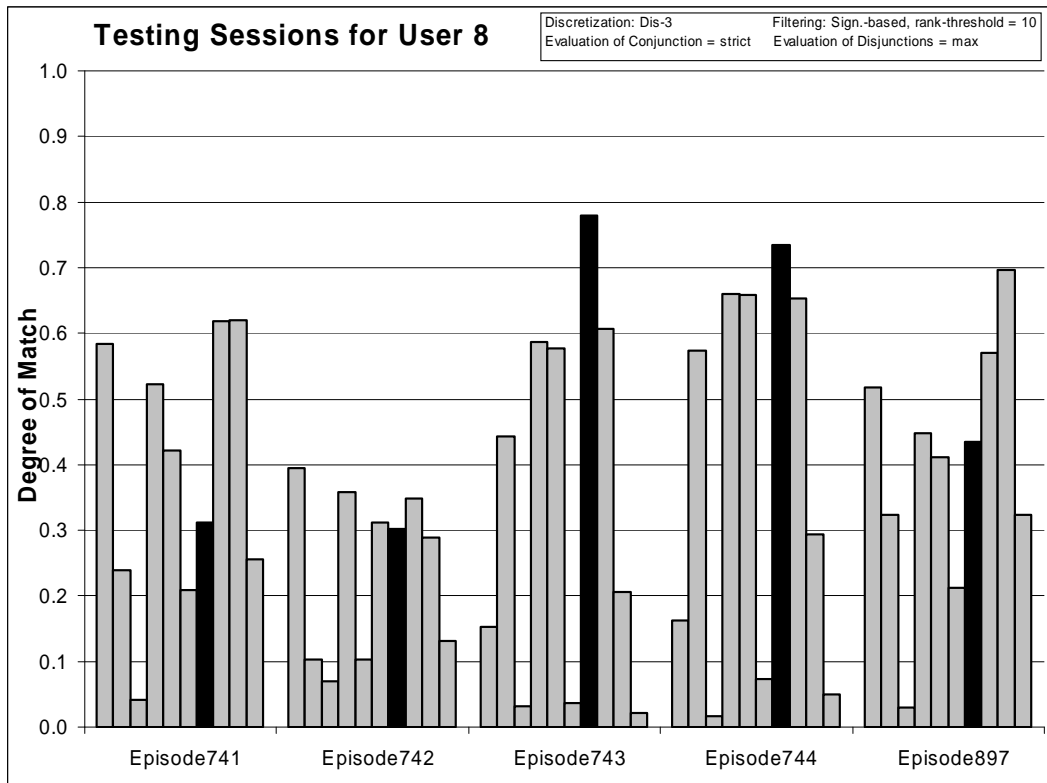


Figure 76: Degrees of match between 10 user models and 5 testing sessions from User 8.

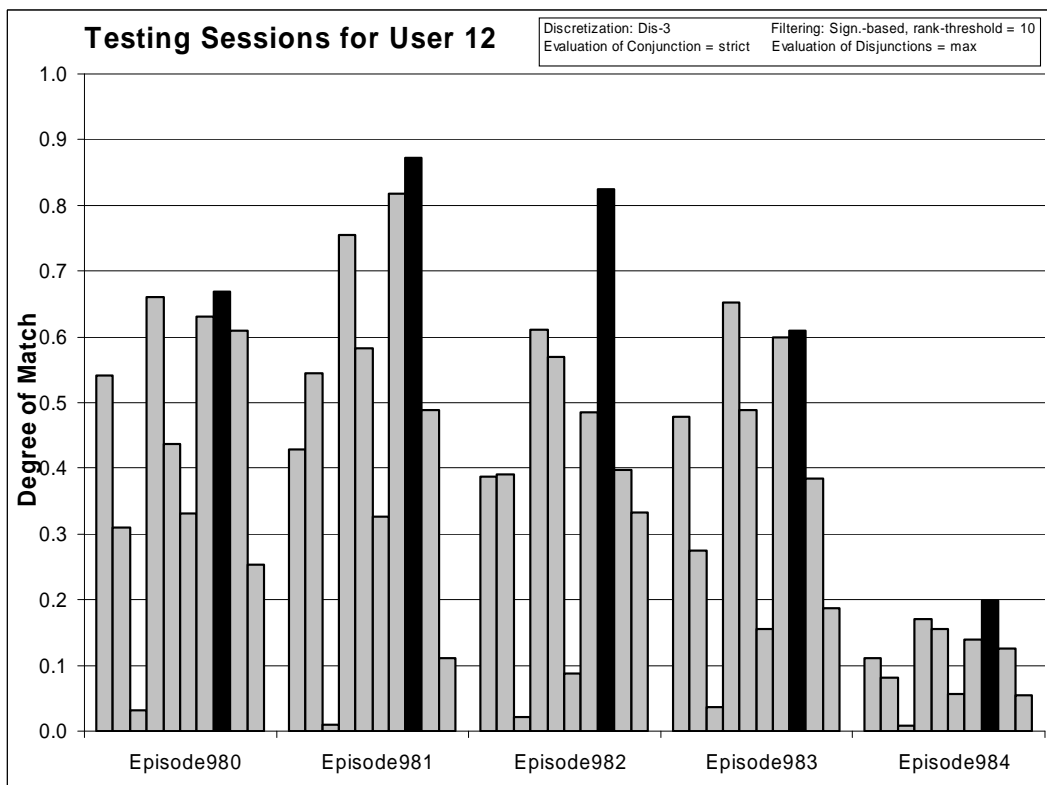


Figure 77: Degrees of match between 10 user models and 5 testing sessions from User 12.

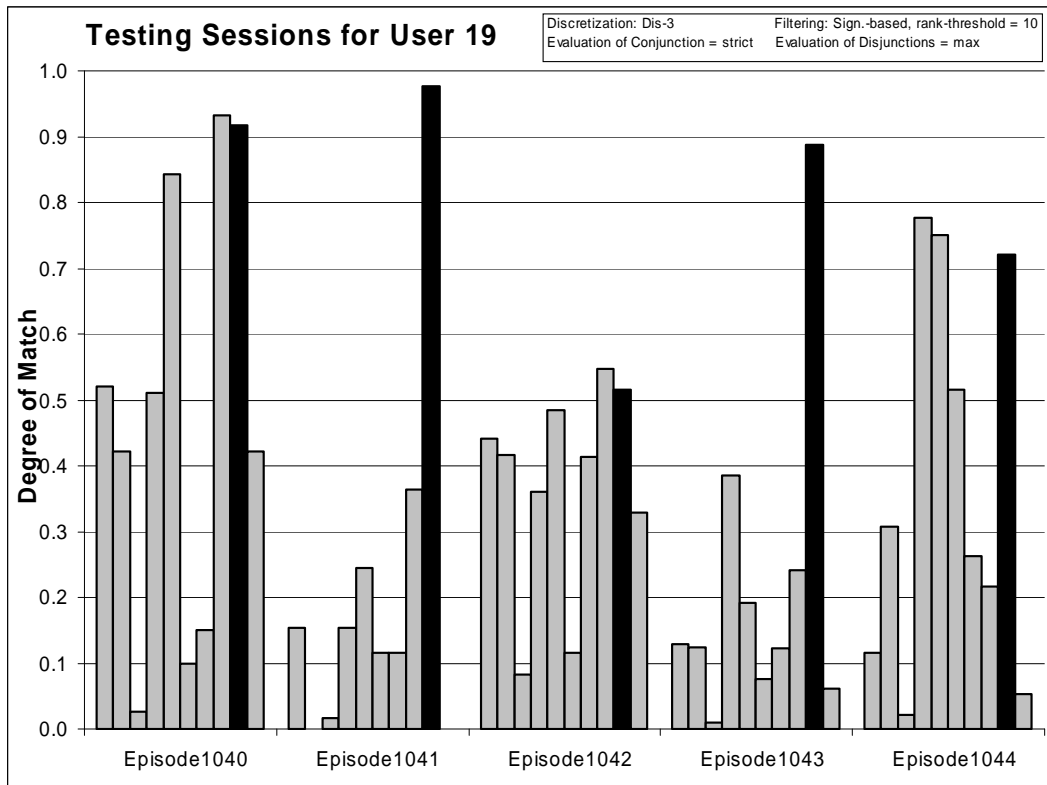


Figure 78: Degrees of match between 10 user models and 5 testing sessions from User 19.

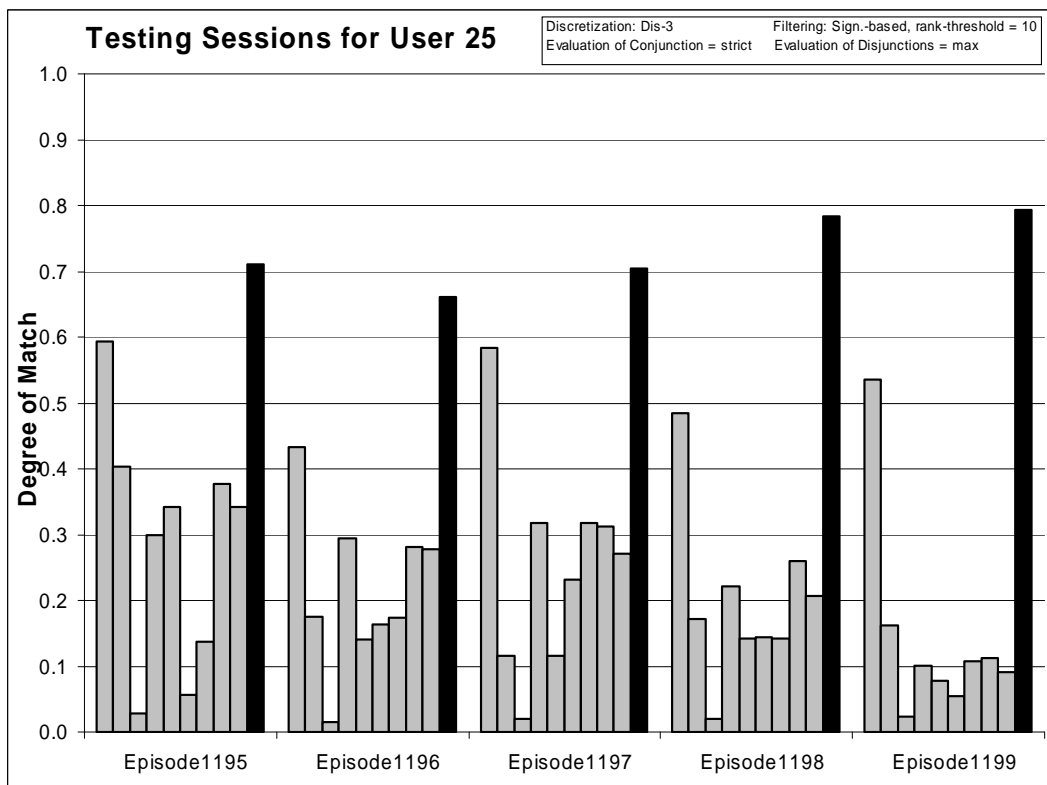


Figure 79: Degrees of match between 10 user models and 5 testing sessions from User 25

Experiment 060807-1 shows that the Multistate Template methodology gives very good results when provided adequate and correctly filtered data. It is not surprising that User 3 was correctly recognized not for all of his testing sessions. The very short Episode 349 is not similar to any episode observed in training data (all degrees of match are zero).

8.4.2 Experiment 040607-2: Filtered Data TR+TS, Characteristic Descriptions

Training Dataset:

Discretization: Dis-3

Filtering: Significance based, conjunctive, rank-threshold = 10, TR+TS

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

AQ21 Learning Parameters:

maxstar = 1 maxrule = 1 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Characteristic descriptions

Testing Parameters:

Evaluation of Conjunction = selectors ratio

Evaluation of Disjunction = max

Acceptance Threshold = 10%

Accuracy Tolerance = 5%

Learning Results:

Total number of rules: 71

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	8	8	1	10	8	6	8	9	8	5

Table 43: Number of learned rules for 10 Users

Testing Results:

Correct: 81.25%

Precision: 40.67%

First Choice Correct: 62.50%

First Choice Precision: 100.00%

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
First Ch. Correct	100%	100%	33%	40%	40%	40%	40%	80%	40%	100%

Table 44: Summary of correct answers for 10 users.

	User 1	User 2	User 3	User 4	User 5	User 7	User 8	User 12	User 19	User 25
User 1 (First Choice Correct: 100%)										
Epi.281	0.895	0.735	0.038	0.756	0.669	0.666	0.723	0.797	0.746	0.851
Epi.282	0.807	0.657	0.080	0.632	0.530	0.659	0.724	0.670	0.626	0.543
Epi.283	0.895	0.708	0.056	0.867	0.579	0.840	0.871	0.811	0.749	0.648
Epi.284	0.940	0.774	0.049	0.849	0.755	0.723	0.761	0.872	0.809	0.734
Epi.285	0.881	0.754	0.023	0.816	0.687	0.718	0.758	0.812	0.785	0.763
User 2 (First Choice Correct: 100%)										
Epi.288	0.633	0.894	0.031	0.811	0.792	0.561	0.871	0.870	0.790	0.485
Epi.289	0.764	0.866	0.022	0.745	0.729	0.569	0.749	0.788	0.757	0.773
Epi.290	0.508	0.844	0.020	0.683	0.673	0.564	0.801	0.795	0.681	0.503
Epi.291	0.675	0.890	0.017	0.715	0.679	0.560	0.734	0.787	0.731	0.535
Epi.333	0.499	0.890	0.051	0.512	0.509	0.498	0.690	0.628	0.587	0.580
User 3 (First Choice Correct: 33%)										
Epi.345	0.167	0.125	1.000	0.000	0.000	0.167	0.333	0.000	0.167	0.250
Epi.347	<u>0.566</u>	0.125	<u>*0.286</u>	<u>0.298</u>	<u>0.375</u>	<u>0.429</u>	<u>0.333</u>	<u>0.304</u>	<u>0.345</u>	<u>0.393</u>
Epi.349	<u>0.667</u>	0.000	<u>*0.000</u>	<u>0.333</u>	<u>0.500</u>	<u>0.333</u>	<u>0.333</u>	<u>0.250</u>	<u>0.333</u>	<u>0.500</u>
User 4 (First Choice Correct: 40%)										
Epi.391	0.674	0.904	0.007	<u>*0.951</u>	0.910	0.550	<u>0.958</u>	0.949	0.742	0.576
Epi.392	0.812	0.804	0.011	<u>*0.889</u>	0.735	0.722	0.864	<u>0.898</u>	0.841	0.646
Epi.393	0.701	0.777	0.007	0.954	0.823	0.683	0.840	0.860	0.853	0.649
Epi.394	<u>0.725</u>	<u>0.688</u>	0.119	<u>*0.580</u>	0.501	<u>0.631</u>	<u>0.669</u>	<u>0.646</u>	0.532	0.507
Epi.512	0.713	0.828	0.008	0.926	0.888	0.655	0.864	0.870	0.785	0.607
User 5 (First Choice Correct: 40%)										
Epi.513	0.540	0.832	0.010	0.932	0.983	0.495	0.966	0.956	0.707	0.505
Epi.514	0.558	0.842	0.024	0.911	<u>*0.939</u>	0.472	<u>0.948</u>	0.926	0.670	0.524
Epi.515	<u>0.492</u>	0.407	0.009	<u>0.617</u>	<u>*0.447</u>	<u>0.495</u>	<u>0.532</u>	0.384	<u>0.595</u>	0.359
Epi.542	0.690	0.837	0.011	0.873	<u>*0.906</u>	0.568	0.875	<u>0.915</u>	0.780	0.607
Epi.543	0.631	0.776	0.017	0.752	0.806	0.558	0.777	0.758	0.659	0.540
User 7 (First Choice Correct: 40%)										
Epi.734	<u>0.871</u>	0.703	0.022	0.799	0.567	<u>*0.841</u>	0.836	0.827	0.755	0.683
Epi.735	0.842	0.714	0.036	0.825	0.596	0.857	0.830	0.838	0.781	0.636
Epi.736	<u>0.780</u>	<u>0.720</u>	0.065	0.621	0.534	<u>*0.694</u>	<u>0.718</u>	<u>0.728</u>	0.615	<u>0.695</u>
Epi.737	0.729	0.672	0.020	0.927	0.837	0.972	0.721	0.726	0.882	0.650
Epi.738	<u>0.815</u>	0.625	0.015	0.697	0.587	<u>*0.785</u>	0.722	0.722	0.658	0.701
User 8 (First Choice Correct: 40%)										
Epi.741	<u>0.876</u>	0.729	0.041	<u>0.845</u>	0.746	0.760	<u>*0.786</u>	<u>0.871</u>	<u>0.857</u>	0.706
Epi.742	<u>0.727</u>	0.682	0.070	<u>0.693</u>	0.572	0.637	<u>*0.688</u>	<u>0.760</u>	<u>0.702</u>	0.612
Epi.743	0.617	0.808	0.031	0.854	0.799	0.520	0.917	0.866	0.691	0.499
Epi.744	0.635	0.846	0.016	0.861	0.832	0.520	0.903	0.877	0.698	0.538
Epi.897	<u>0.865</u>	0.752	0.029	<u>0.841</u>	0.787	0.785	<u>*0.834</u>	<u>0.870</u>	<u>0.892</u>	0.740
User 12 (First Choice Correct: 80%)										
Epi.980	0.858	0.751	0.032	0.879	0.738	0.765	0.879	0.883	0.850	0.691

Epi.981	0.763	0.844	0.010	0.928	0.810	0.664	0.948	0.963	0.775	0.644
Epi.982	0.660	0.749	0.022	0.825	0.787	0.659	0.745	0.936	0.784	0.666
Epi.983	0.764	0.744	0.036	0.875	0.783	0.665	0.869	0.875	0.767	0.631
Epi.984	<u>0.509</u>	<u>0.425</u>	0.009	<u>0.536</u>	<u>0.465</u>	<u>0.496</u>	<u>0.534</u>	<u>*0.422</u>	<u>0.560</u>	0.401

User 19 (First Choice Correct: 40%)

Epi.1040	0.865	0.793	0.026	0.855	0.916	0.726	0.740	<u>0.971</u>	<u>*0.949</u>	0.792
Epi.1041	0.783	0.665	0.017	0.780	0.773	0.768	0.773	0.835	0.985	0.662
Epi.1042	<u>0.812</u>	0.777	0.083	0.771	0.752	0.719	0.791	<u>0.840</u>	<u>*0.805</u>	0.678
Epi.1043	0.743	0.704	0.010	0.833	0.715	0.717	0.745	0.783	0.953	0.661
Epi.1044	0.680	0.758	0.022	<u>0.902</u>	0.885	0.765	0.760	0.732	<u>*0.879</u>	0.602

User 25 (First Choice Correct: 100%)

Epi.1195	0.853	0.787	0.028	0.705	0.676	0.595	0.716	0.778	0.724	0.857
Epi.1196	0.820	0.665	0.014	0.619	0.636	0.536	0.667	0.647	0.619	0.862
Epi.1197	0.857	0.641	0.020	0.647	0.562	0.639	0.714	0.709	0.638	0.894
Epi.1198	0.856	0.662	0.020	0.585	0.582	0.597	0.610	0.606	0.591	0.912
Epi.1199	0.821	0.699	0.023	0.559	0.555	0.498	0.696	0.651	0.589	0.903

Table 45: Testing results for experiment 040606-2 (Characteristic Descriptions).

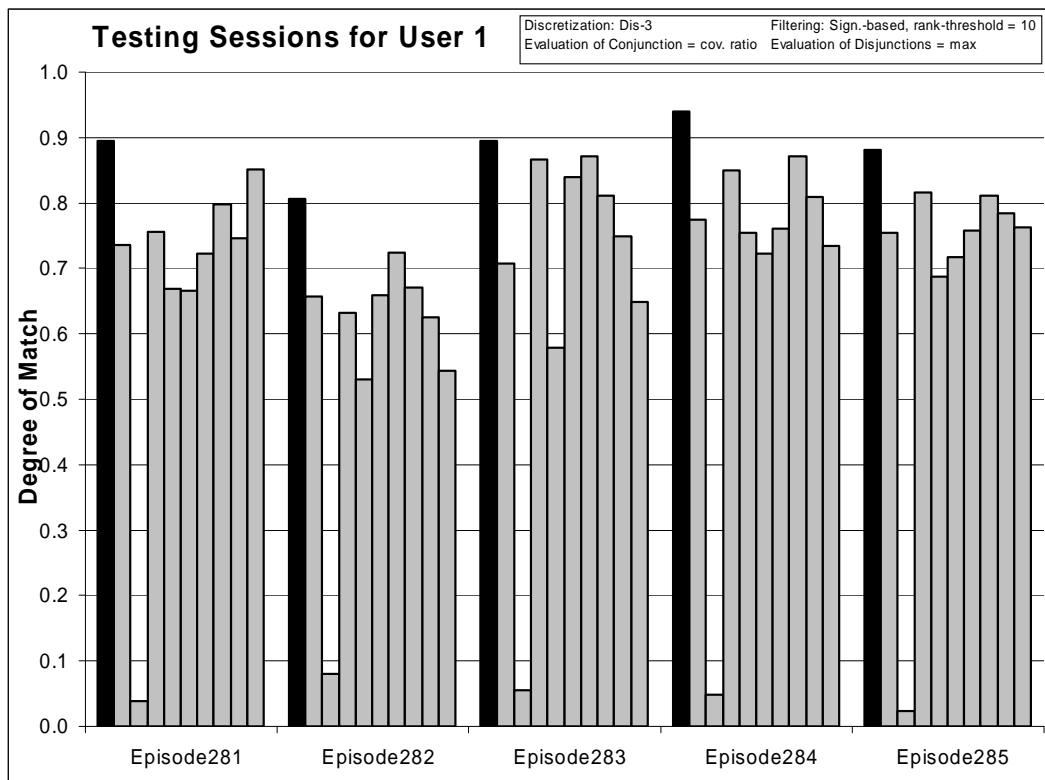


Figure 80: Degrees of match between 10 user models and 5 testing sessions from User 1.

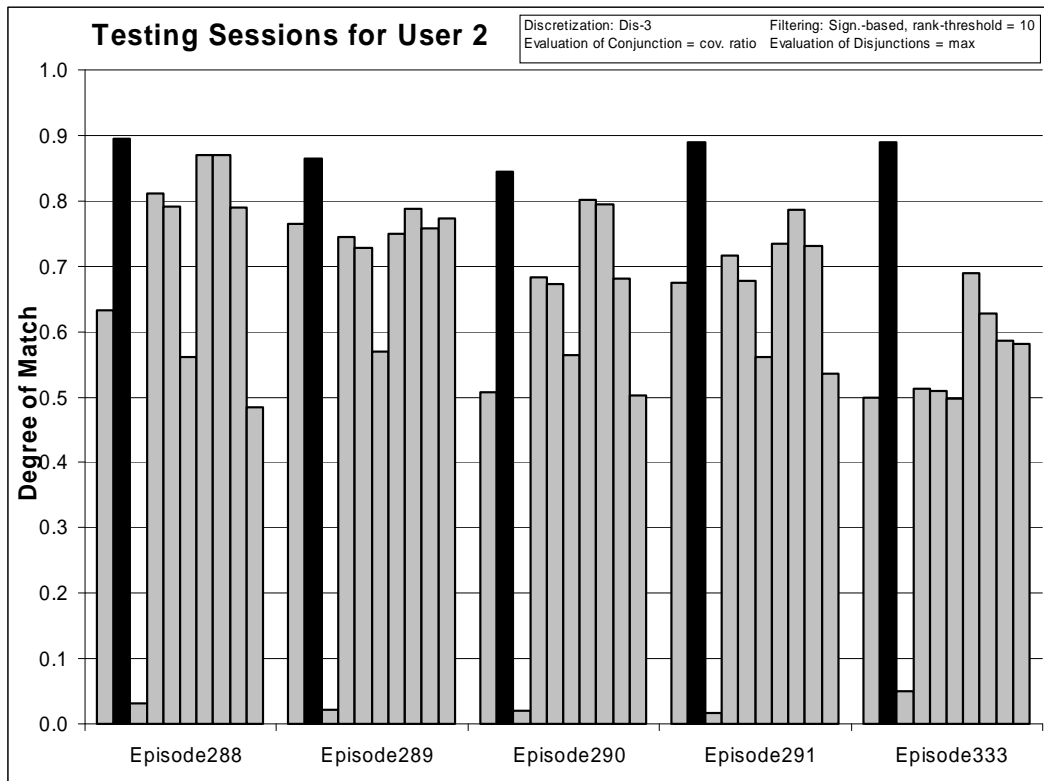


Figure 81: Degrees of match between 10 user models and 5 testing sessions from User 2.

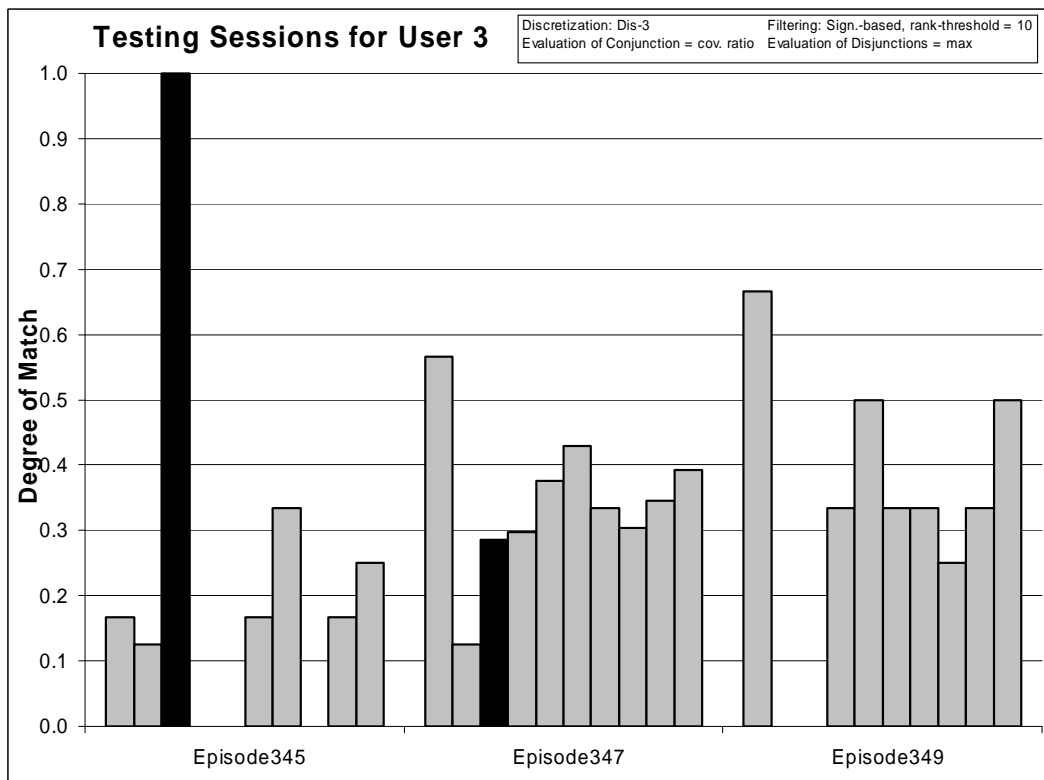


Figure 82: Degrees of match between 10 user models and 3 testing sessions from User 3.

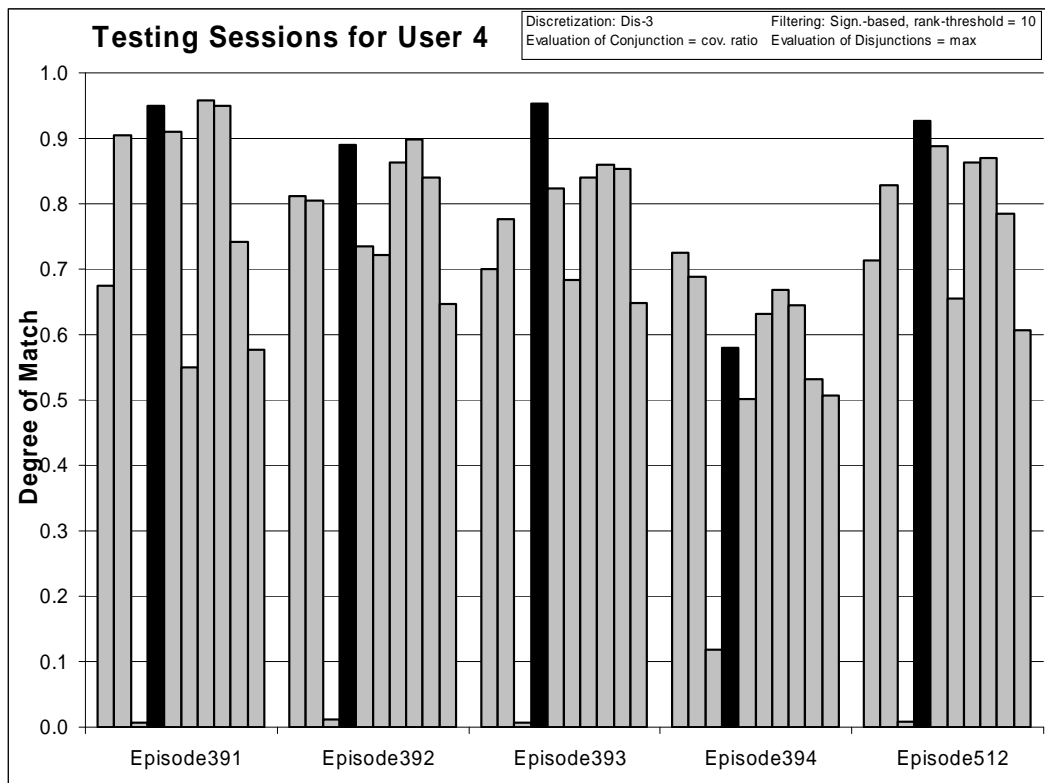


Figure 83: Degrees of match between 10 user models and 5 testing sessions from User 4.

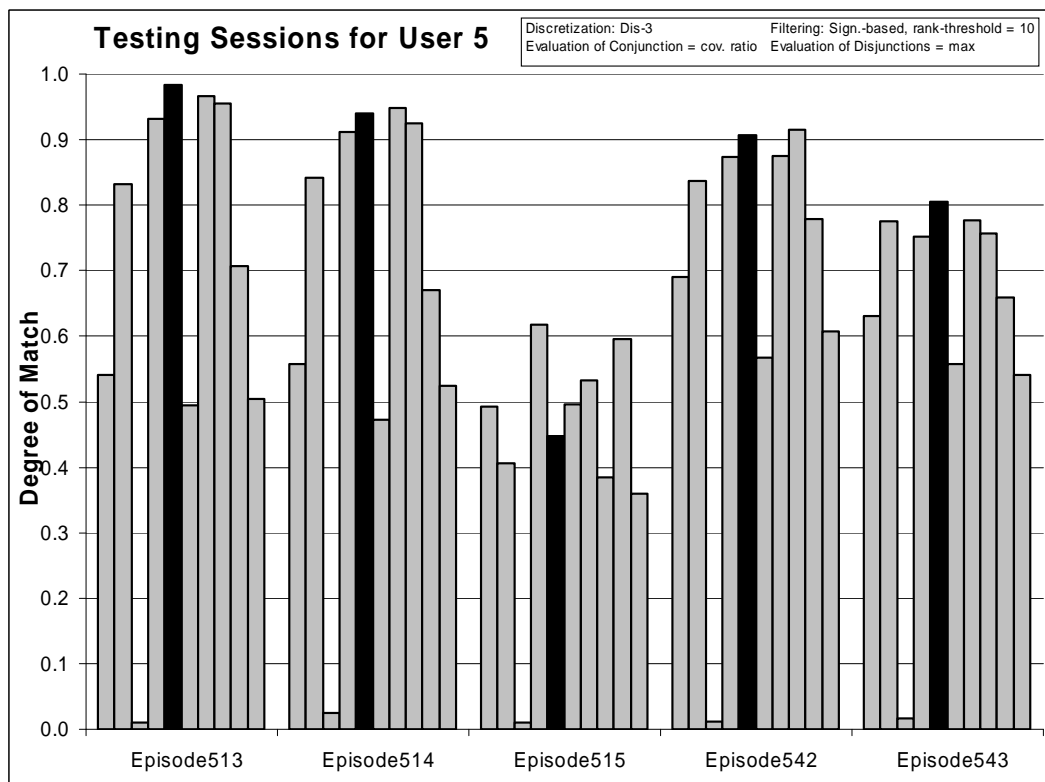


Figure 84: Degrees of match between 10 user models and 5 testing sessions from User 5.

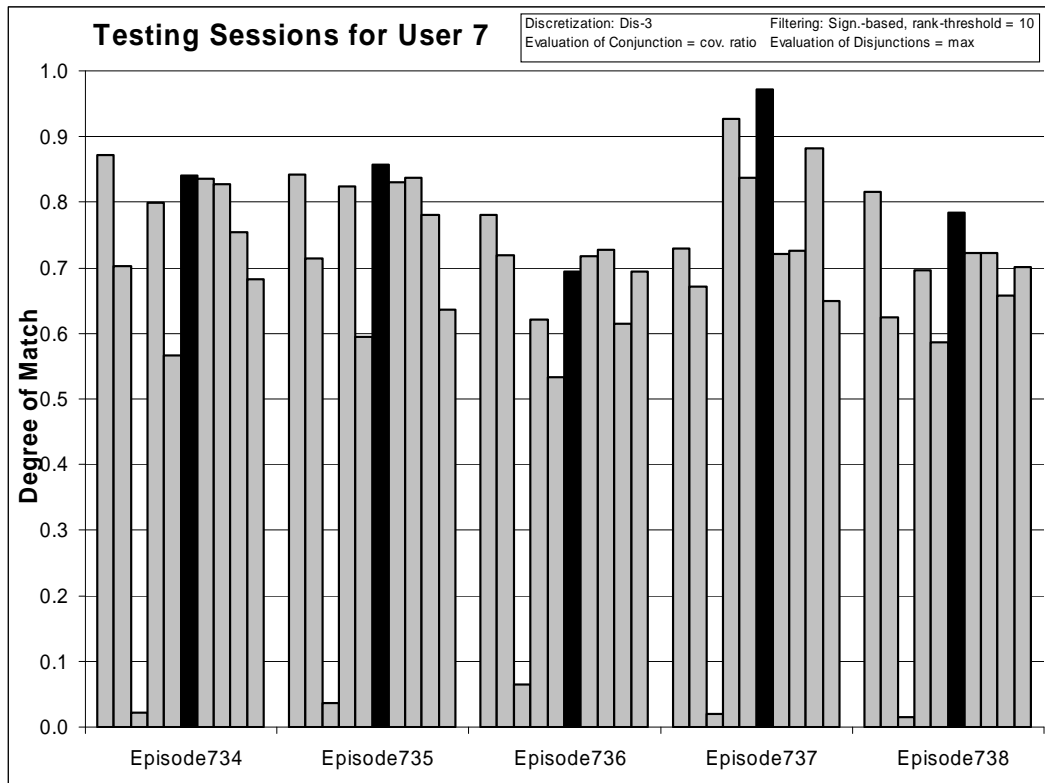


Figure 85: Degrees of match between 10 user models and 5 testing sessions from User 7.

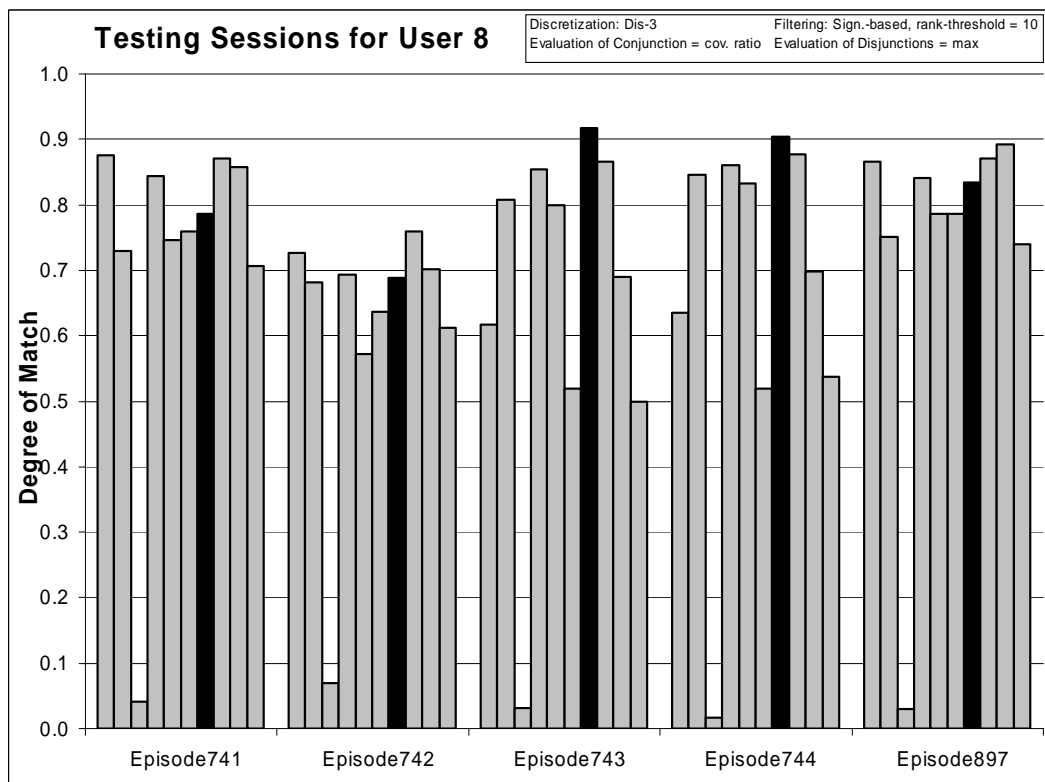


Figure 86: Degrees of match between 10 user models and 5 testing sessions from User 8.

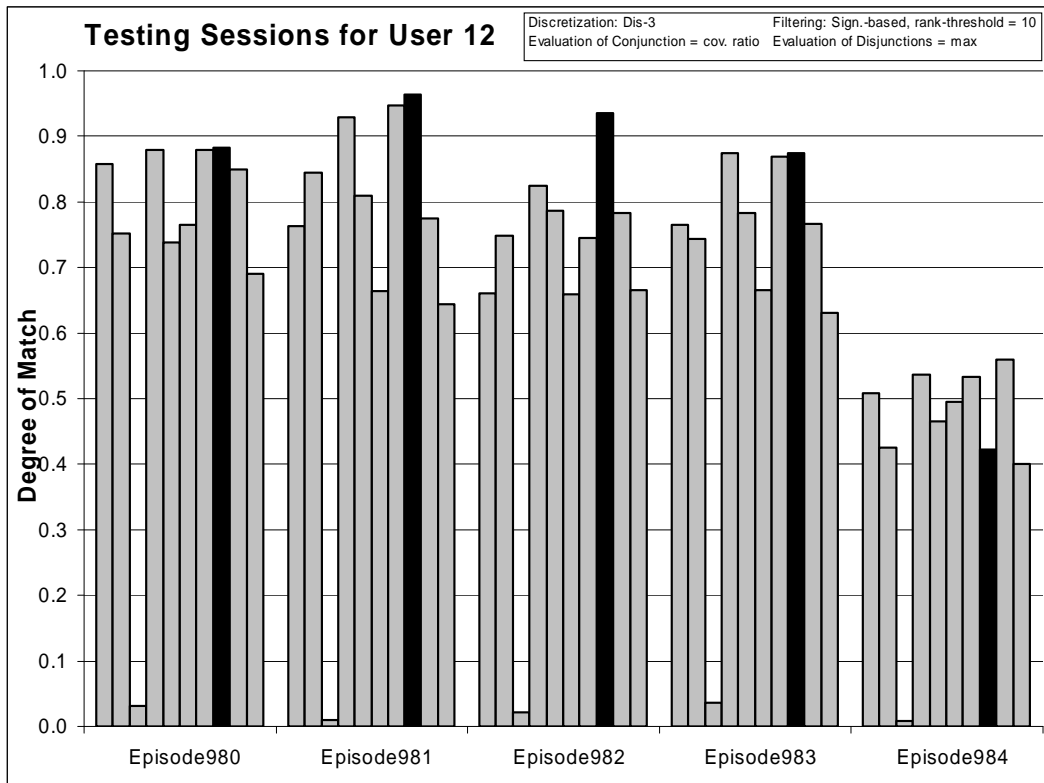


Figure 87: Degrees of match between 10 user models and 5 testing sessions from User 12.

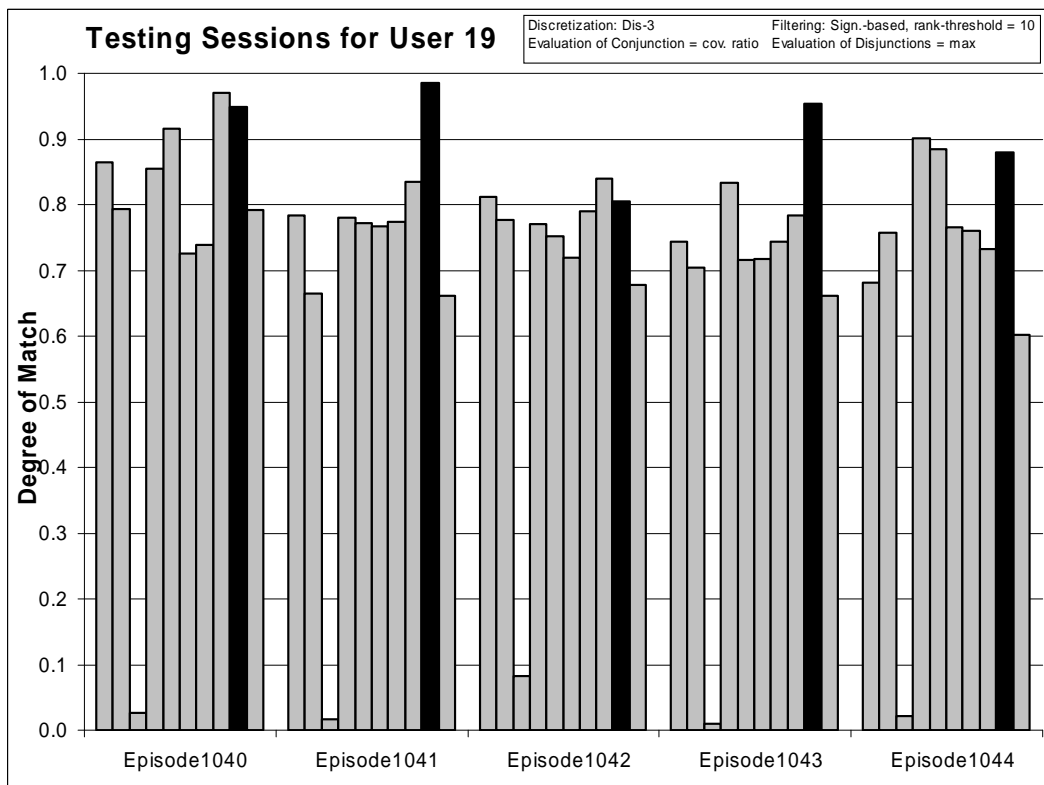


Figure 88: Degrees of match between 10 user models and 5 testing sessions from User 19.

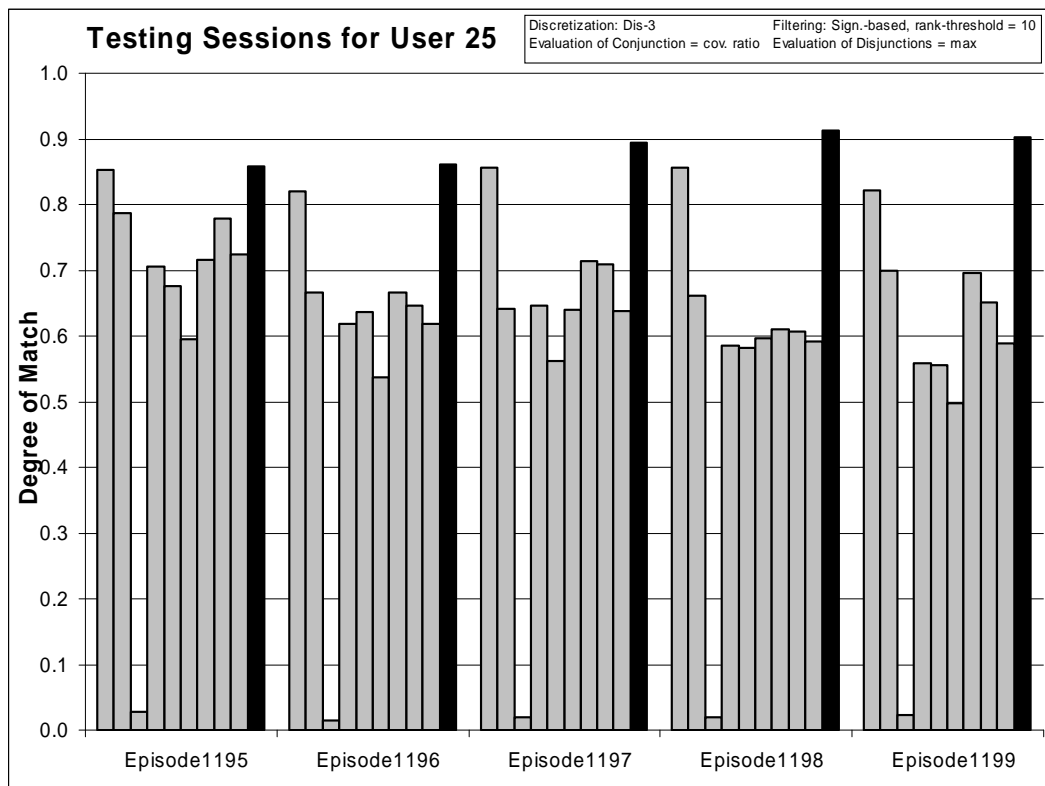


Figure 89: Degrees of match between 10 user models and 5 testing sessions from User 25.

Experiment 060807-2 shows that the Multistate Templates methodology gives very good results when provided adequate and correctly filtered data. Although characteristic descriptions provide models with comparable quality, use of the *selector ratio* evaluation of conjunctions made recognition more difficult. As shown in Table 45 and figures above, degrees of match to all models are high, with one exception, User 3, whose case was discussed in conclusion to experiment 040607-1. Comparison of degrees of match with those from experiment 040607-1 indicate that *strict* evaluation of selectors give more reliable results and less models have scores within tolerance. A comparative study of different testing methods is presented in experiments 040620-1 and 040620-2.

8.4.3 Experiment 040608: Filtered Data Using Small Numbers of Significant Events TR+TS

Training Dataset:

Discretization: Dis-3

Filtering: Significance based, conjunctive, rank-threshold = 6, TR+TS

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

AQ21 Learning Parameters:

maxstar = 1 maxrule = 1 ambiguity = ignore-for-learning
 trim = optimal exceptions = false mode = tf
 Discriminant descriptions

Testing Parameters:

Evaluation of Conjunction = strict
 Evaluation of Disjunction = max
 Acceptance Threshold = 10%
 Accuracy Tolerance = 5%

Learning Results:

Total number of rules: 71

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	4	5	1	6	5	4	4	5	4	3

Table 46: Number of learned rules for 10 Users

Testing Results:

Correct: 67%
 Precision: 91%
 First Choice Correct: 62%
 First Choice Precision: 98%

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
First Ch. Correct	100%	100%	33%	40%	40%	40%	40%	80%	40%	100%

Table 47: Summary of correct answers for 10 users.

	User 1	User 2	User 3	User 4	User 5	User 7	User 9	User 12	User 19	User 25
User 1 (20%)										
Epi.281	<u>*0.416</u>	0.088	0.038	0.197	0.068	0.238	0.068	<u>0.448</u>	0.259	<u>0.524</u>
Epi.282	0.575	0.257	0.080	0.165	0.090	0.170	0.102	0.231	0.017	0.015
Epi.283	<u>*0.563</u>	0.174	0.056	0.347	0.076	<u>0.604</u>	0.076	0.347	0.021	0.000
Epi.284	<u>*0.317</u>	0.159	0.049	0.213	0.159	0.312	0.164	<u>0.415</u>	<u>0.443</u>	0.257
Epi.285	<u>*0.286</u>	0.140	0.023	0.213	0.117	<u>0.330</u>	0.074	<u>0.385</u>	<u>0.312</u>	<u>0.315</u>
User 2 (100%)										
Epi.288	0.114	0.693	0.031	0.569	0.585	0.085	0.534	0.107	0.017	0.000
Epi.289	0.194	0.457	0.022	0.216	0.205	0.103	0.167	0.434	0.263	0.329
Epi.290	0.097	0.612	0.020	0.108	0.341	0.045	0.274	0.263	0.017	0.000
Epi.291	0.179	0.623	0.017	0.251	0.294	0.028	0.245	0.142	0.146	0.029
Epi.333	0.000	0.591	0.051	0.019	0.005	0.068	0.000	0.000	0.034	0.015

User 3 (67%)

Epi.345	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Epi.347	0.000	0.000	0.286	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Epi.349	0.000	0.000	*0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

User 4 (100%)

Epi.391	0.138	0.380	0.007	0.824	0.793	0.167	0.750	0.138	0.018	0.000
Epi.392	0.250	0.229	0.011	0.517	0.346	0.382	0.273	0.407	0.292	0.035
Epi.393	0.145	0.180	0.007	0.497	0.433	0.284	0.397	0.354	0.395	0.086
Epi.394	0.127	0.452	0.119	0.000	0.000	0.175	0.000	0.000	0.064	0.071
Epi.512	0.145	0.244	0.008	0.735	0.522	0.328	0.494	0.146	0.262	0.049

User 5 (80%)

Epi.513	0.000	0.277	0.010	0.528	0.956	0.033	0.811	0.321	0.047	0.000
Epi.514	0.046	0.292	0.024	0.595	0.884	0.019	0.816	0.241	0.006	0.000
Epi.515	0.084	0.097	0.009	<u>0.283</u>	<u>*0.199</u>	0.122	0.172	0.102	0.125	0.000
Epi.542	0.046	0.210	0.011	0.536	0.693	0.066	0.500	0.201	0.372	0.031
Epi.543	0.075	0.302	0.017	0.443	0.538	0.065	0.432	0.116	0.069	0.054

User 7 (100%)

Epi.734	0.448	0.081	0.022	0.283	0.081	0.677	0.081	0.283	0.031	0.193
Epi.735	0.336	0.195	0.036	0.329	0.067	0.620	0.067	0.407	0.090	0.078
Epi.736	0.299	0.117	0.065	0.000	0.000	0.390	0.000	0.000	0.000	0.325
Epi.737	0.076	0.044	0.020	0.669	0.044	0.892	0.044	0.076	0.610	0.000
Epi.738	0.179	0.019	0.015	0.183	0.019	0.597	0.019	0.119	0.096	0.209

User 8 (20%)

Epi.741	<u>0.236</u>	<u>0.168</u>	0.041	<u>0.365</u>	<u>0.350</u>	<u>0.331</u>	<u>*0.166</u>	<u>0.481</u>	<u>0.511</u>	0.080
Epi.742	<u>0.233</u>	<u>0.102</u>	0.070	<u>0.288</u>	<u>0.102</u>	<u>0.316</u>	<u>*0.102</u>	<u>0.233</u>	0.079	0.009
Epi.743	0.085	0.252	0.031	0.441	0.577	0.081	0.725	0.167	0.068	0.000
Epi.744	0.121	0.325	0.016	0.571	<u>0.660</u>	0.092	<u>*0.642</u>	0.170	0.041	0.009
Epi.897	<u>0.384</u>	<u>0.324</u>	0.029	<u>0.396</u>	<u>0.398</u>	0.270	<u>*0.324</u>	<u>0.456</u>	<u>0.329</u>	0.000

User 12 (40%)

Epi.980	0.396	0.324	0.032	<u>0.523</u>	0.437	0.423	0.324	<u>*0.468</u>	0.186	0.041
Epi.981	0.347	0.254	0.010	<u>0.745</u>	<u>0.583</u>	0.346	<u>0.509</u>	<u>*0.399</u>	0.068	0.000
Epi.982	0.174	0.214	0.022	0.215	0.380	0.109	0.349	0.765	0.192	0.203
Epi.983	0.222	0.249	0.036	<u>0.447</u>	<u>0.489</u>	0.281	0.434	<u>*0.365</u>	0.208	0.030
Epi.984	0.051	0.037	0.009	0.081	0.118	0.065	0.091	0.154	0.066	0.039

User 19 (60%)

Epi.1040	0.135	0.047	0.026	0.135	0.469	0.099	0.047	0.917	0.781	0.375
Epi.1041	0.117	0.000	0.017	0.153	0.244	0.117	0.000	0.361	0.867	0.000
Epi.1042	<u>0.345</u>	<u>0.441</u>	0.083	<u>0.361</u>	<u>0.484</u>	<u>0.135</u>	<u>0.329</u>	<u>0.484</u>	<u>*0.218</u>	0.000
Epi.1043	0.055	0.060	0.010	0.112	0.135	0.075	0.063	0.178	0.792	0.054
Epi.1044	0.053	0.279	0.022	<u>0.786</u>	0.242	0.525	0.220	0.053	<u>*0.569</u>	0.000

User 25 (100%)

Epi.1195	0.321	0.160	0.028	0.048	0.098	0.060	0.048	0.342	0.296	0.611
Epi.1196	0.261	0.079	0.014	0.181	0.044	0.180	0.044	0.240	0.136	0.579

Epi.1197	0.548	0.116	0.020	0.271	0.116	0.251	0.116	0.271	0.015	0.523
Epi.1198	0.325	0.063	0.020	0.097	0.033	0.149	0.033	0.206	0.114	0.740
Epi.1199	0.491	0.141	0.023	0.063	0.050	0.064	0.051	0.091	0.029	0.690

Table 48: Testing results for Experiment 040608 (rank-threshold = 6)

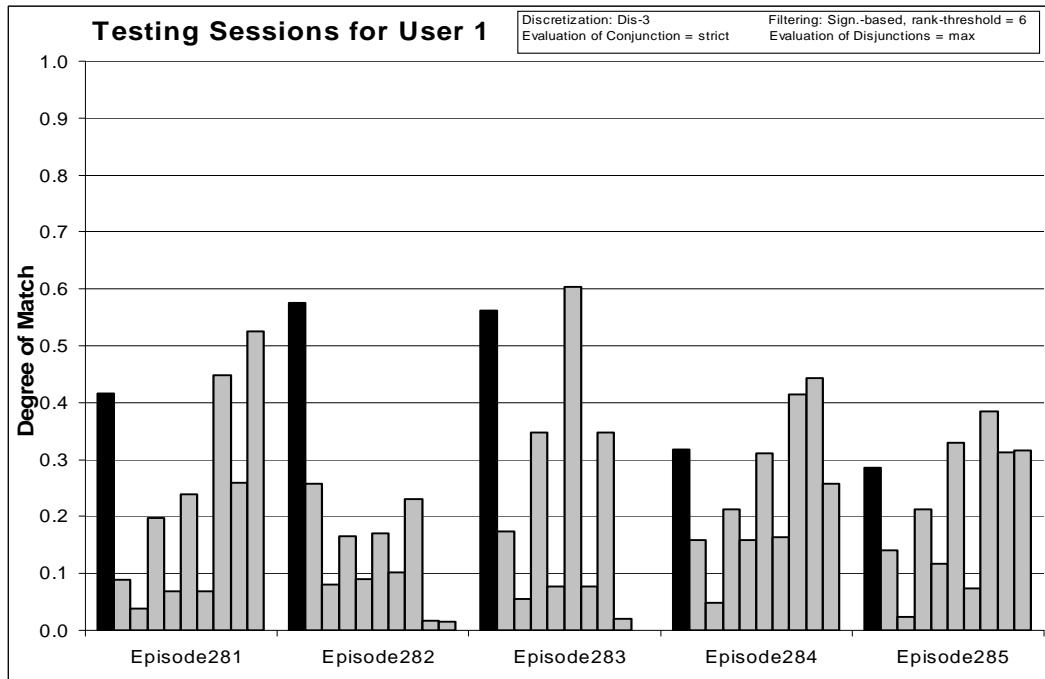


Figure 90: Degrees of match between 10 user models and 5 testing sessions from User 1.

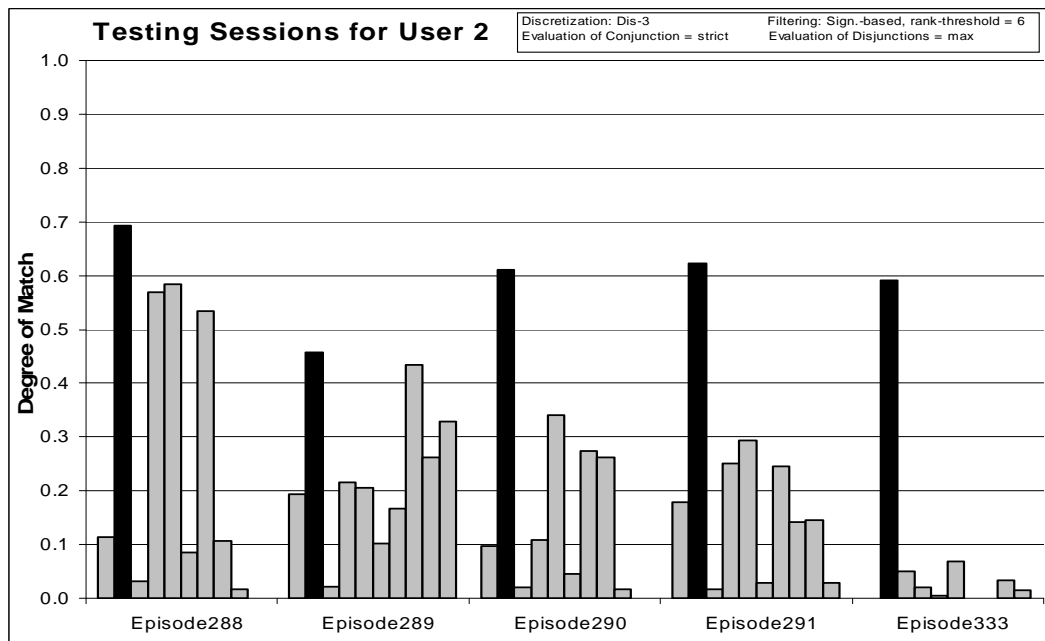


Figure 91: Degrees of match between 10 user models and 5 testing sessions from User 2.

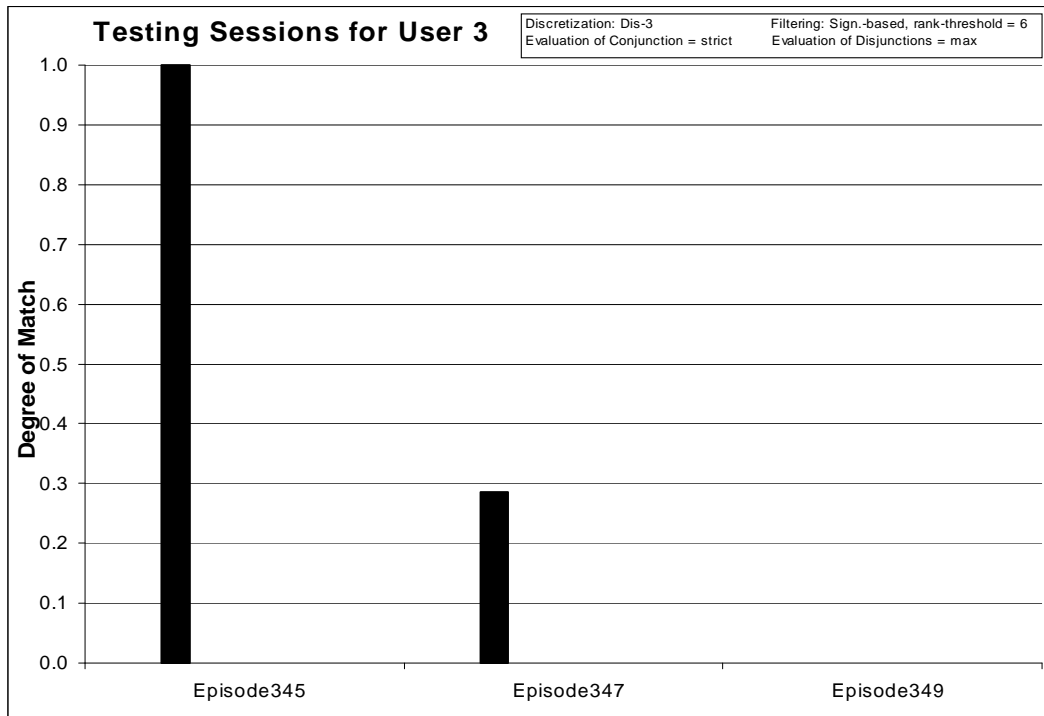


Figure 92: Degrees of match between 10 user models and 3 testing sessions from User 3.

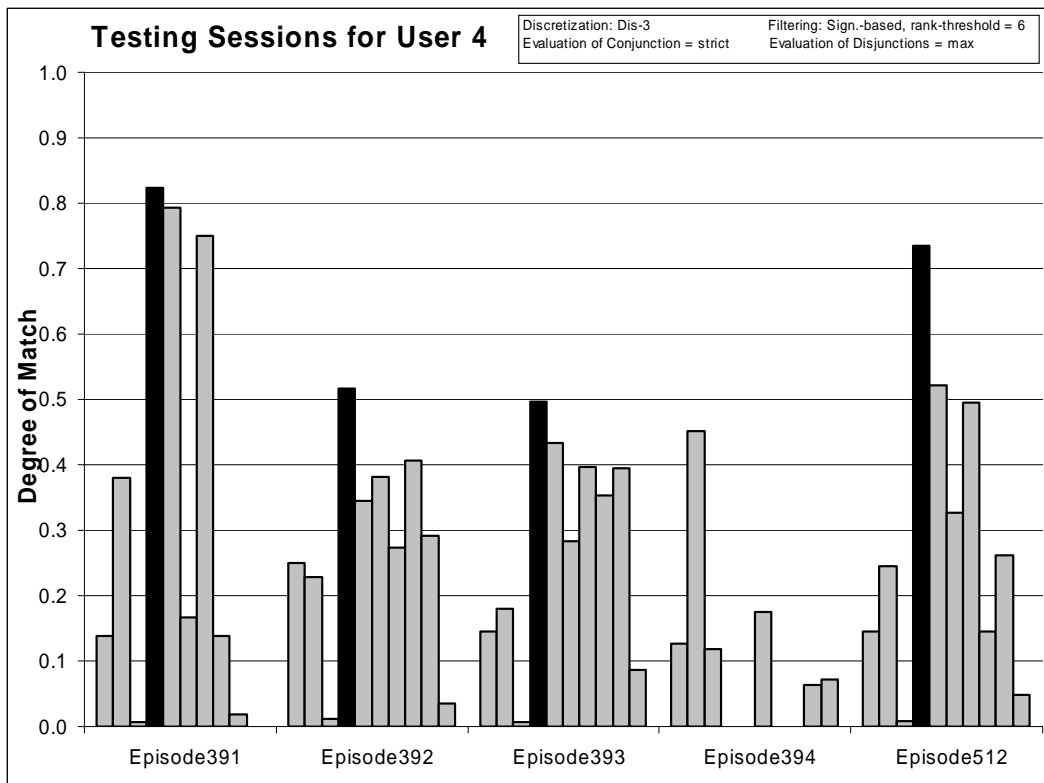


Figure 93: Degrees of match between 10 user models and 5 testing sessions from User 4

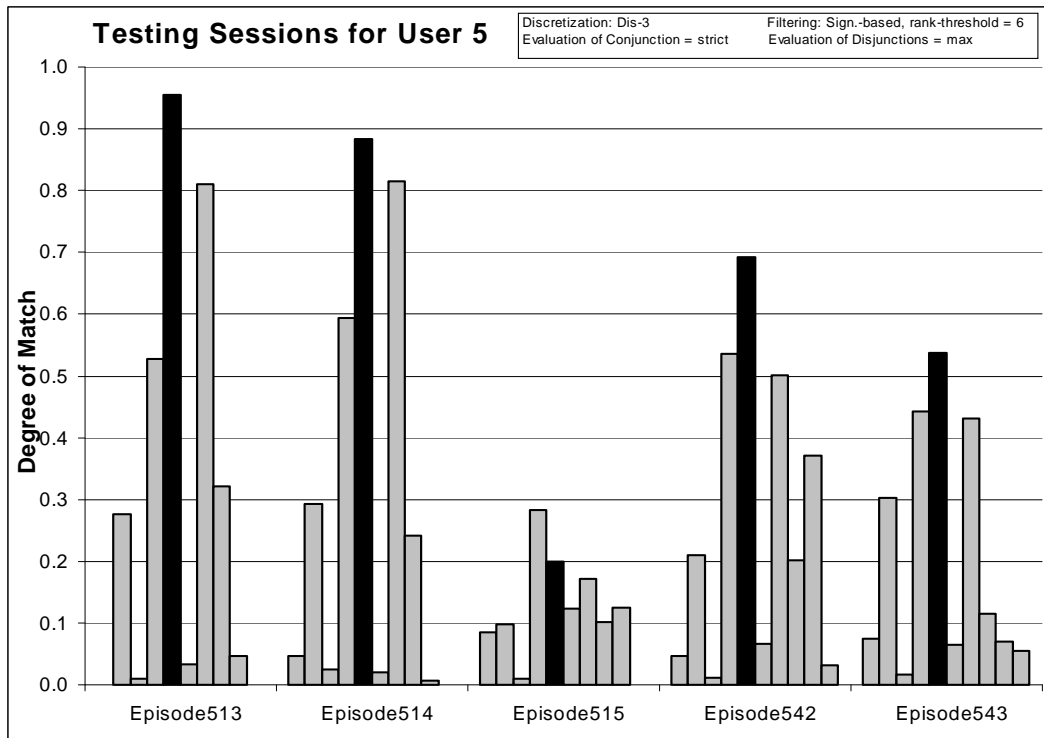


Figure 94: Degrees of match between 10 user models and 5 testing sessions from User 5.

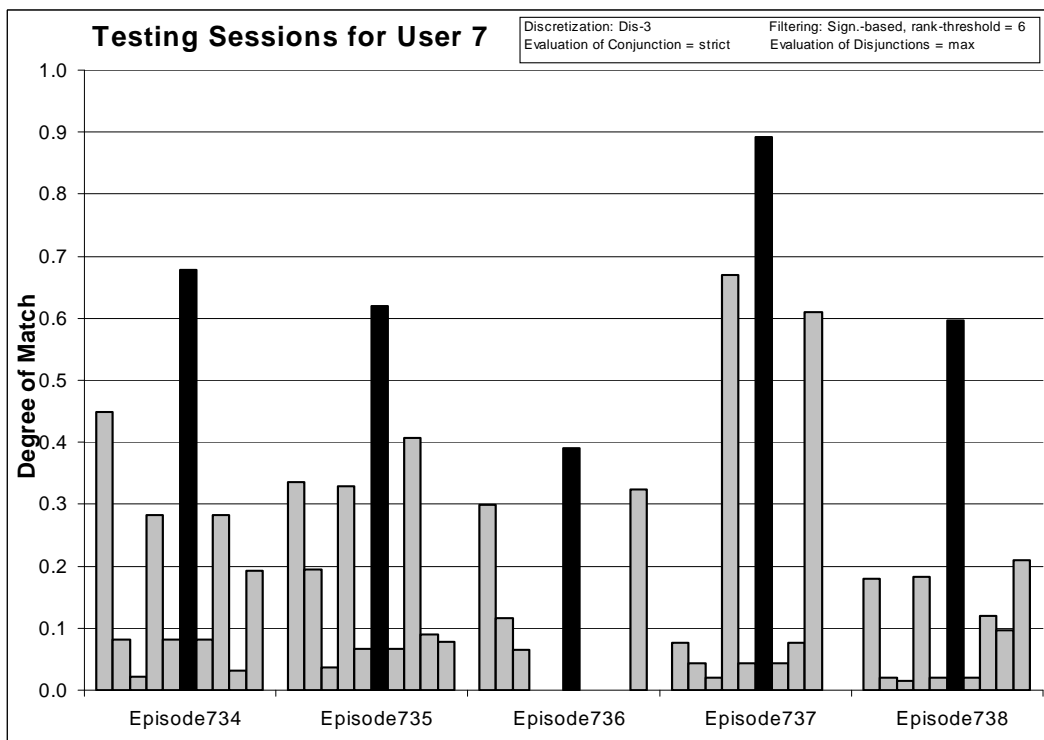


Figure 95: Degrees of match between 10 user models and 5 testing sessions from User 7

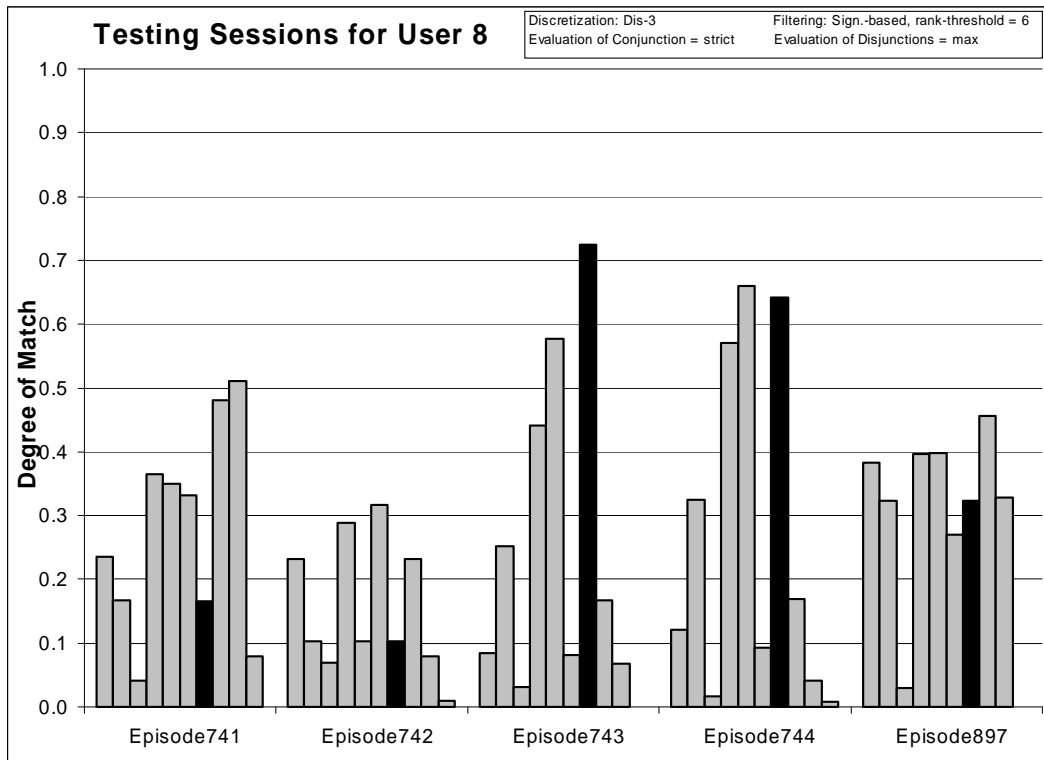


Figure 96: Degrees of match between 10 user models and 5 testing sessions from User 8.

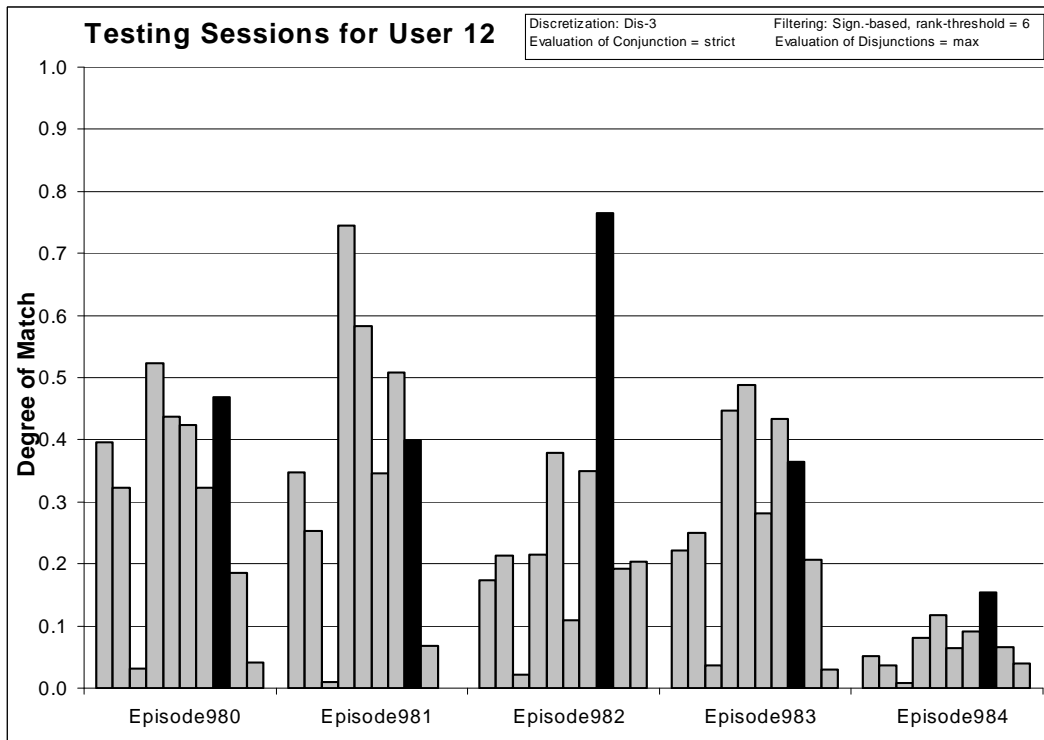


Figure 97: Degrees of match between 10 user models and 5 testing sessions from User 12

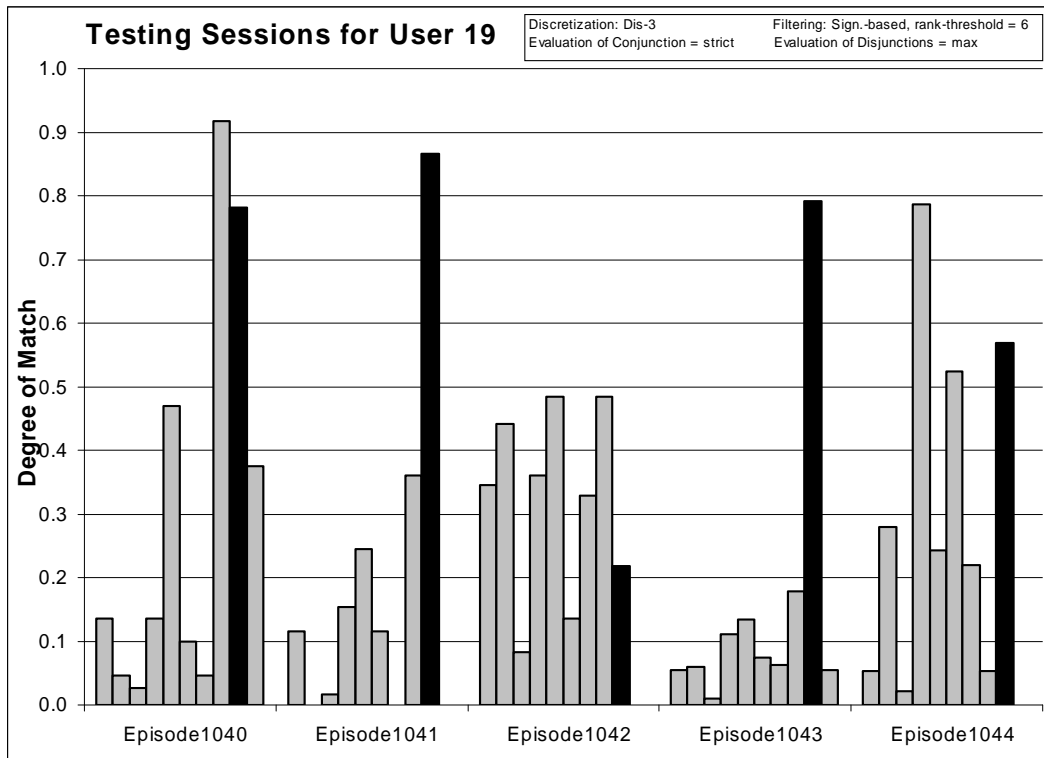


Figure 98: Degrees of match between 10 user models and 5 testing sessions from User 19.

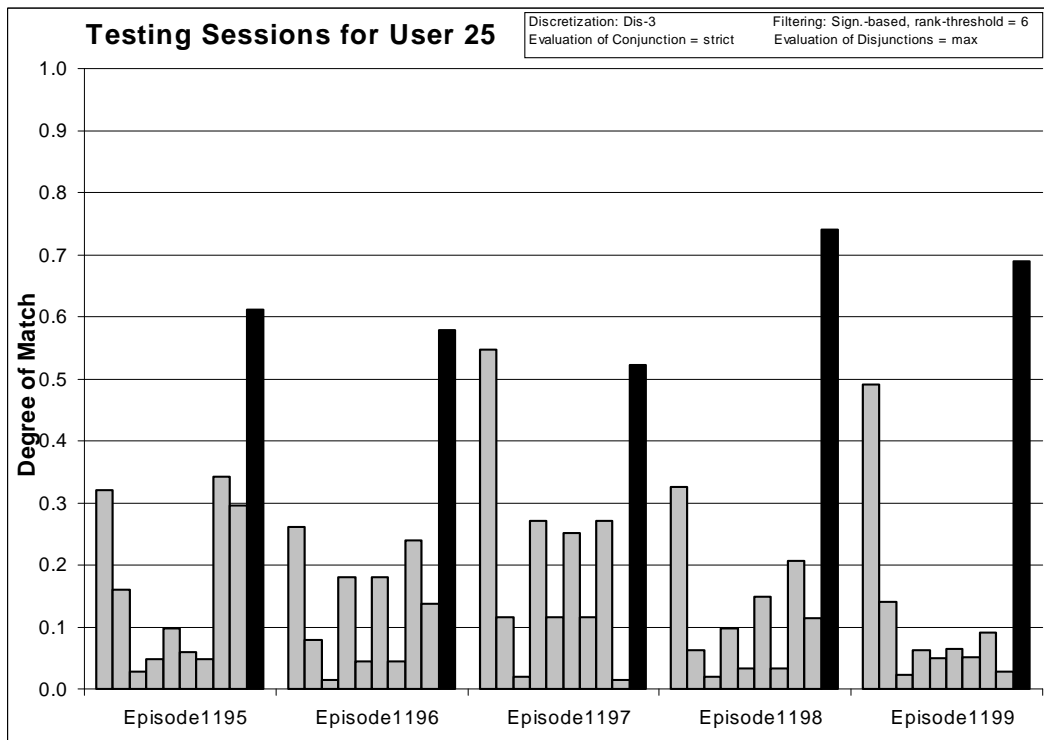


Figure 99: Degrees of match between 10 user models and 5 testing sessions from User 25

Experiment 060807-2 shows that the Multistate Templates methodology gives very good results when provided adequate and correctly filtered data. In this case, the number of selected significant events per user is 6, which gave worse results than in experiment 040607-1, in which 10 significant events per user were selected. Further investigation of the number of significant events needed for successful learning is presented in experiment 040610.

8.4.4 Experiment 040610: Summary of Results for Filtered Data based on Rank-Threshold

Training Dataset:

Discretization: Dis-3

Filtering: Significance based, conjunctive, rank-threshold = 6, 10, 14, 30, 40, TR+TS

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

AQ21 Learning Parameters:

maxstar = 1 maxrule = 1 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Discriminant descriptions

Testing Parameters:

Evaluation of Conjunction = strict

Evaluation of Disjunction = max

Acceptance Threshold = 10%

Accuracy Tolerance = 5%

Testing Results:

Correct: 79.17%

Precision: 82.46%

First Choice Correct: 75%

First Choice Precision: 100%

rank-threshold	% Correct	Correct Precision	% First Choice Correct	% First Choice Precision
6	66.67%	91.45%	62.5%	97.73%
10	79.17%	82.46%	75%	100%
14	79.17%	74.91%	75%	95.56%
30	83.33%	73.54%	75%	97.73%
40	83.33%	74.91%	75%	100%

Table 49: Summary of results for different values of Rank-Threshold

This experiment shows that the optimal number of significant events per user is 10. Models learned using 6 significant events per user provided worse results. When number of events is increased, the *First Choice Correct* score could not be improved. There is improvement in terms of *Correct* answers, but with loss of precision.

8.4.5 Experiment 040615-1: Discretized and Unfiltered Data

Training Dataset:

Discretization: Dis-3

Filtering: not filtered

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

AQ21 Learning Parameters:

maxstar = 1 maxrule = 10 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Characteristic descriptions

Testing Parameters:

Evaluation of Conjunction = strict

Evaluation of Disjunction = max

Acceptance Threshold = 10%

Accuracy Tolerance = 5%

Total number of rules: 5467

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	337	1001	42	956	396	452	456	596	608	623

Table 50: Number of learned rules for 10 users.

Testing Results:

Correct: 68.75%

Precision: 53.93%

First Choice Correct: 58.33%

First Choice Precision: 100.00%

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
First Ch. Correct	60%	80%	33%	60%	0%	80%	20%	100%	100%	100%

Table 51: Summary of correct answers for 10 users.

Epi.980	0.566	0.835	0.000	0.812	0.751	0.561	0.713	0.873	0.713	0.697
Epi.981	0.341	0.918	0.000	0.931	0.847	0.425	0.838	0.971	0.819	0.472
Epi.982	0.404	0.701	0.000	0.712	0.650	0.638	0.695	0.949	0.479	0.453
Epi.983	0.447	0.853	0.000	0.820	0.781	0.448	0.817	0.882	0.608	0.454
Epi.984	0.105	0.214	0.000	0.216	0.207	0.108	0.201	0.257	0.151	0.125

User 19 (First Choice Correct: 100%)

Epi.1040	0.474	0.901	0.010	0.938	0.901	0.490	0.500	0.906	0.922	0.896
Epi.1041	0.136	0.381	0.000	0.386	0.375	0.139	0.136	0.381	0.989	0.952
Epi.1042	0.341	0.746	0.004	0.774	0.520	0.441	0.500	0.663	0.857	0.706
Epi.1043	0.119	0.293	0.000	0.479	0.250	0.337	0.190	0.268	0.940	0.871
Epi.1044	0.597	0.892	0.002	0.944	0.796	0.684	0.817	0.380	0.901	0.687

User 25 (First Choice Correct: 100%)

Epi.1195	0.663	0.805	0.000	0.424	0.349	0.632	0.418	0.680	0.428	0.944
Epi.1196	0.794	0.865	0.000	0.355	0.321	0.472	0.355	0.510	0.379	0.932
Epi.1197	0.784	0.839	0.000	0.332	0.312	0.598	0.442	0.678	0.312	0.945
Epi.1198	0.508	0.568	0.000	0.302	0.261	0.492	0.314	0.524	0.275	0.984
Epi.1199	0.725	0.827	0.000	0.207	0.111	0.622	0.164	0.577	0.159	0.965

Table 52: Testing results for experiment 040615-1 (Unfiltered Data).

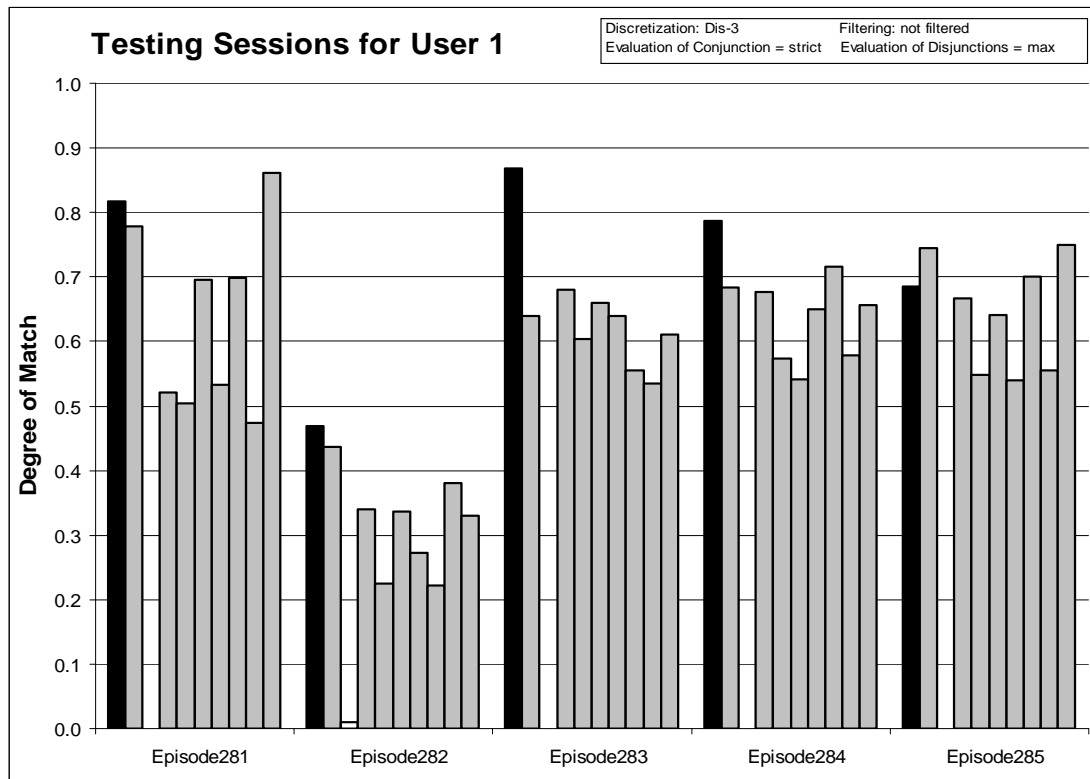


Figure 100: Degrees of match between 10 user models and 5 testing sessions from User 1.

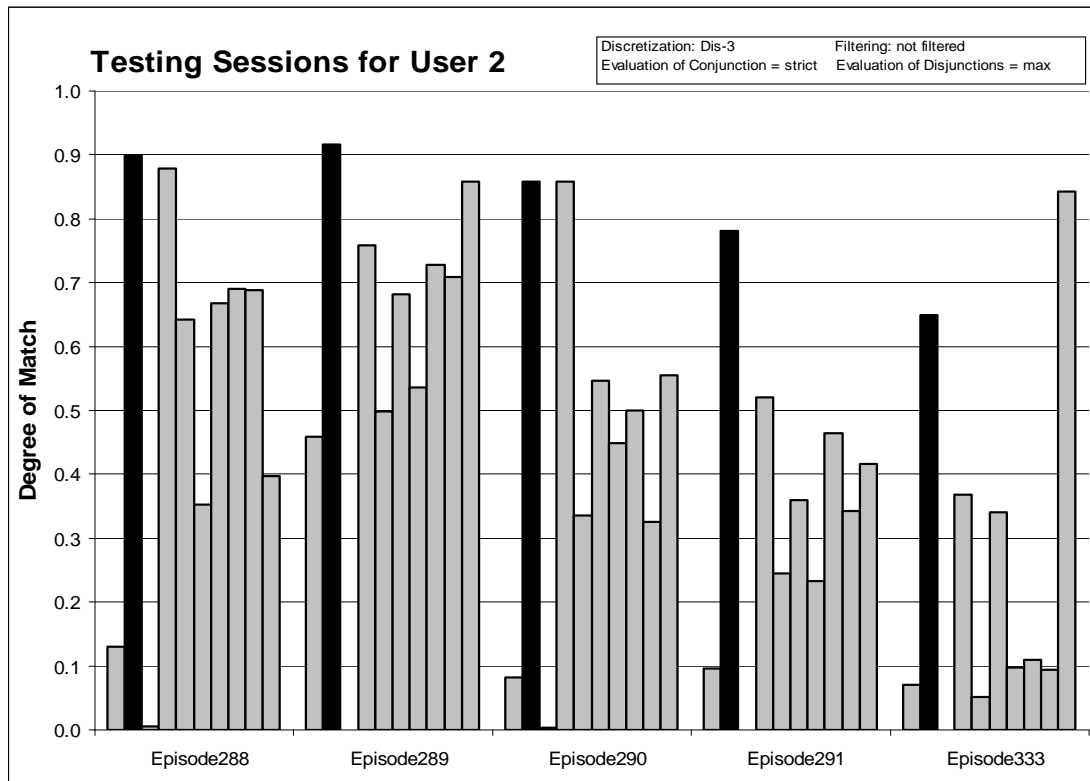


Figure 101: Degrees of match between 10 user models and 5 testing sessions from User 2.

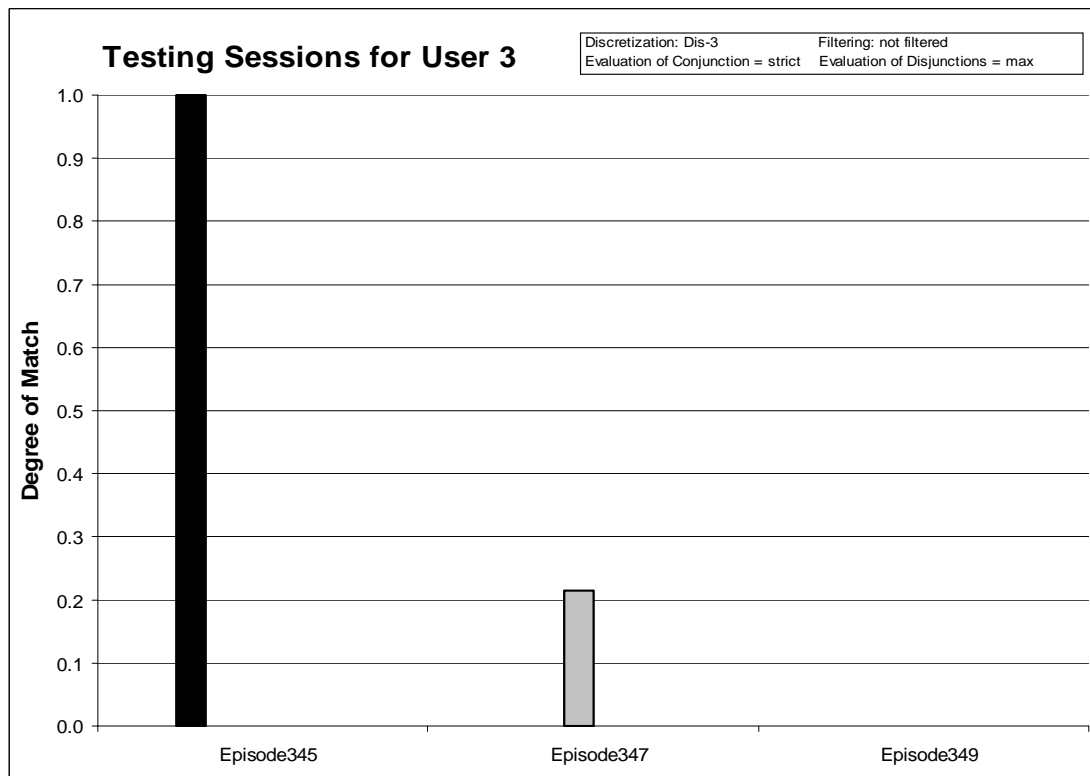


Figure 102: Degrees of match between 10 user models and 3 testing sessions from User 3.

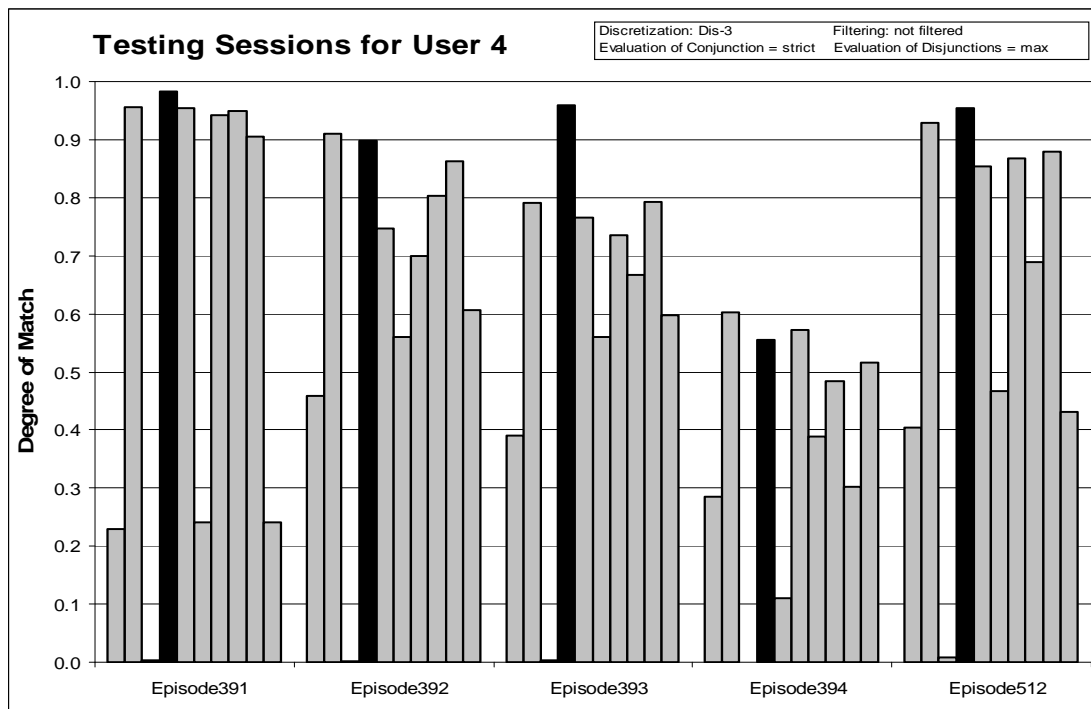


Figure 103: Degrees of match between 10 user models and 5 testing sessions from User 4.

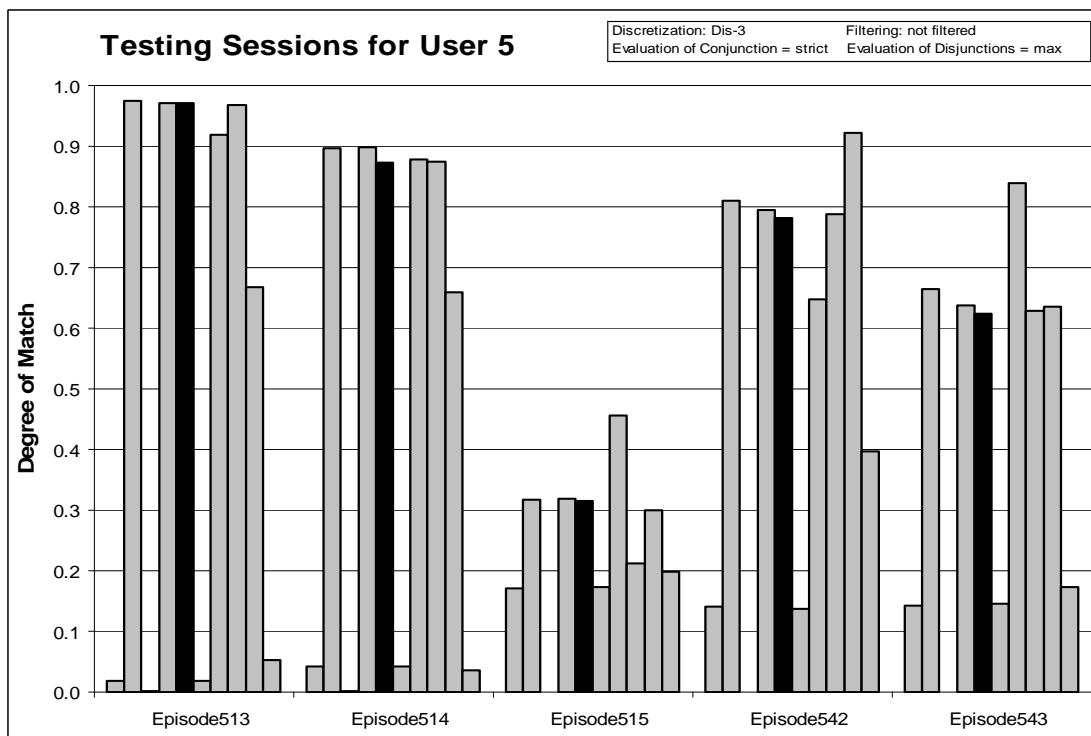


Figure 104: Degrees of match between 10 user models and 5 testing sessions from User 5.

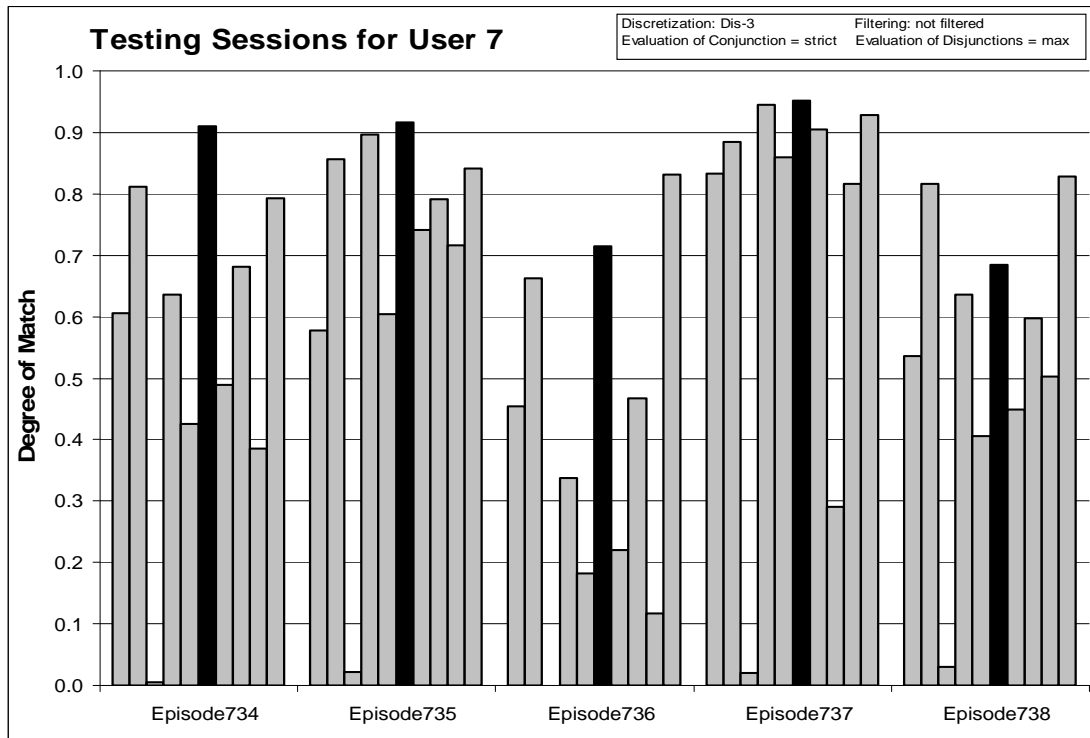


Figure 105: Degrees of match between 10 user models and 5 testing sessions from User 7.

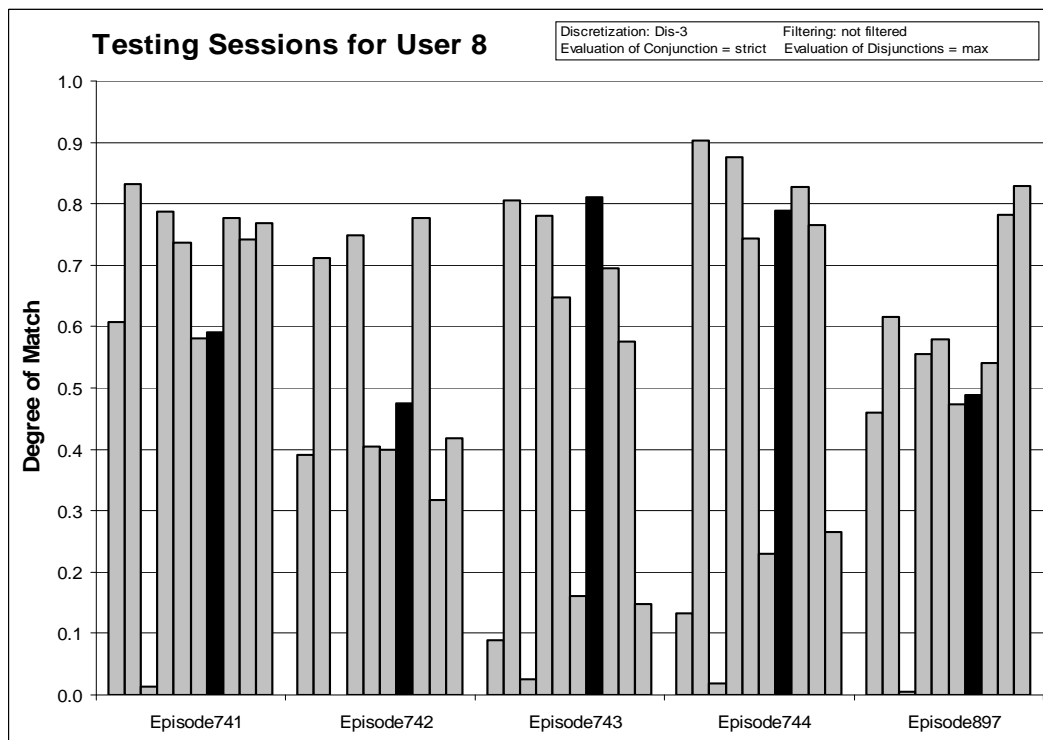


Figure 106: Degrees of match between 10 user models and 5 testing sessions from User 8.

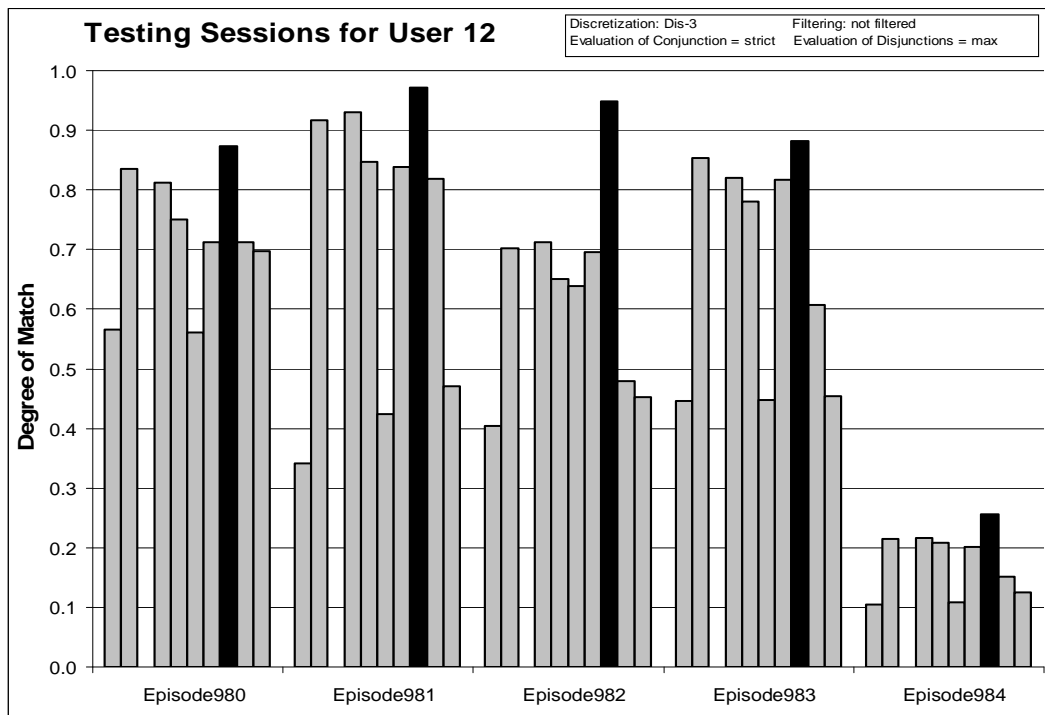


Figure 107: Degrees of match between 10 user models and 5 testing sessions from User 12.

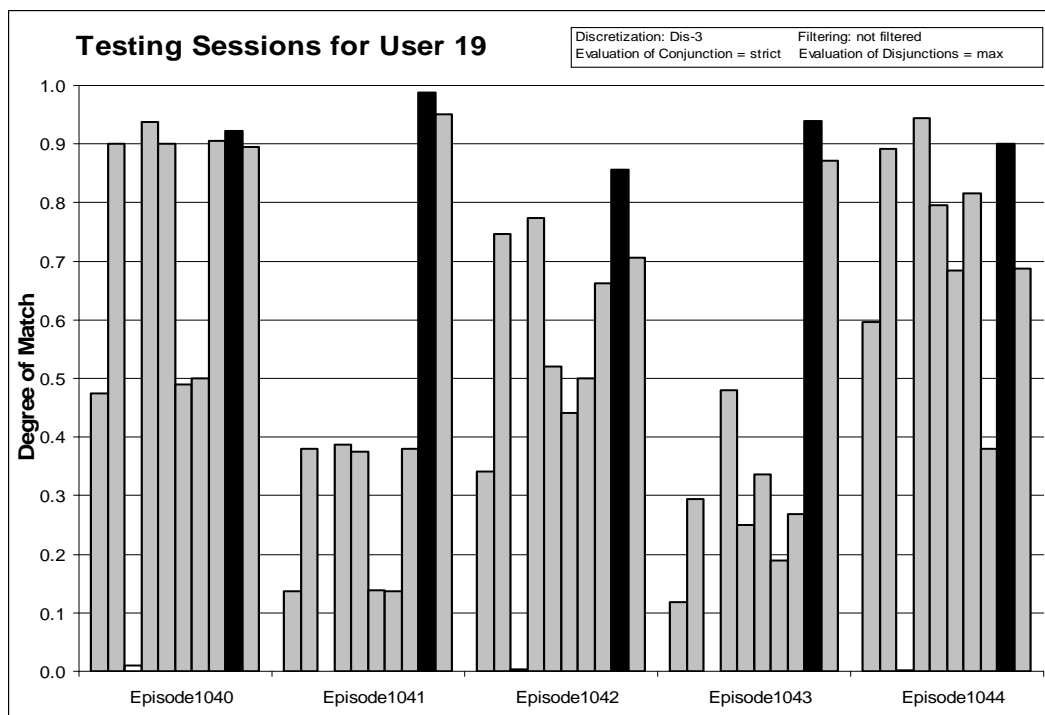


Figure 108: Degrees of match between 10 user models and 5 testing sessions from User 19

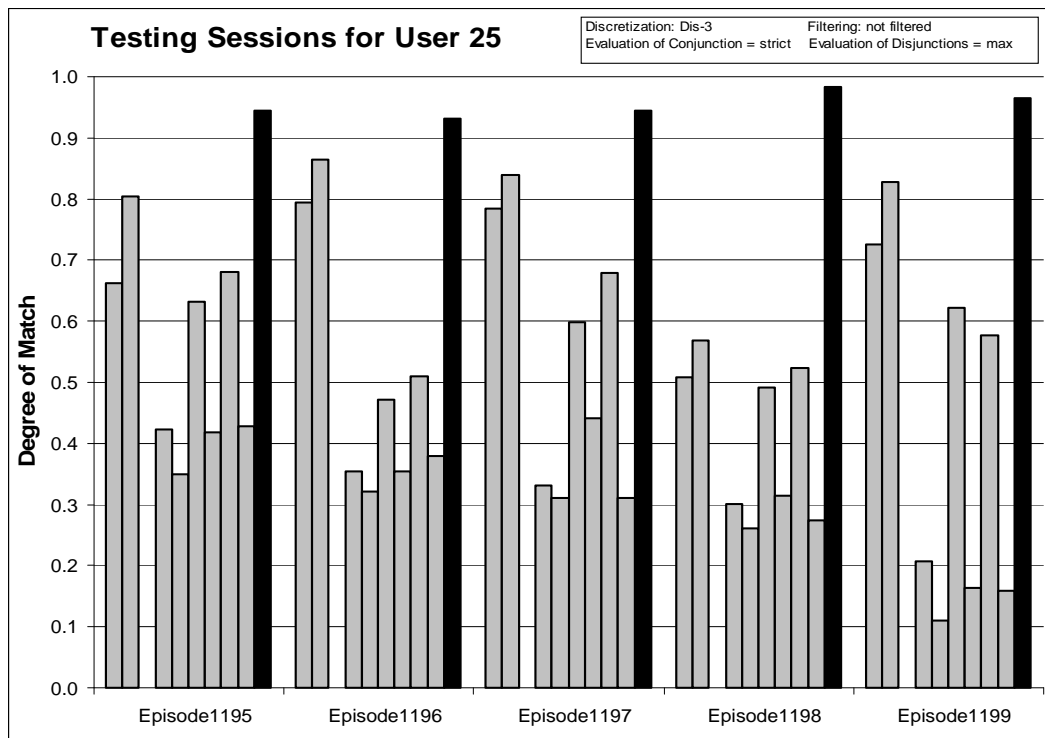


Figure 109: Degrees of match between 10 user models and 5 testing sessions from User 25.

Very good results for Users 2, 7, 12, 19, and 25 show that the Multistate Templates method can also be applied to unfiltered data. The investigation of similarity presented in Section 8.1.2 shows that that Users 4 and 8 can be easily confused with User 5, which in fact happened in the presented case. Degrees of match for User 5 were slightly too low for his Episodes so that his First Choice Correct score is 0%, but some of the answers are within 5% tolerance.

8.4.6 Experiment 040620-1: Comparison of Testing Methods on Discriminant Models

Training Dataset:

Discretization: Dis-3

Filtering: Significance based, conjunctive, rank-threshold = 10, TR+TS

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

AQ21 Learning Parameters:

maxstar = 1 maxrule = 1 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Discriminant descriptions

Evaluation of Conjunction, Evaluation of Disjunction	EPIC-MT				EPIC-RB			
	Correct	Prec.	First Choie Correct	First Choice Prec.	Correct	Prec.	First Choie Correct	First Choice Prec.
Selectors Ratio, Max	81.25%	40.67%	62.50%	100.00%	39.58%	87.65%	35.42%	100.00%
Selectors Ratio, avg.	52.08%	56.40%	43.75%	100.00%	39.58%	93.46%	37.50%	100.00%
Selectors Ratio, psum	85.42%	15.29%	35.42%	100.00%	37.50%	84.13%	35.42%	100.00%
Selectors Ratio, best.	45.83%	72.22%	33.33%	100.00%	35.42%	95.56%	35.42%	100.00%
Coverage Ratio, Max	37.5%	95.56%	35.42%	100.00%	39.58%	91.45%	39.58%	100.00%
Coverage Ratio, avg.	4.17%	100.00 %	4.17%	100.00%	37.50%	100.00 %	37.50%	100.00%
Coverage Ratio, psum	41.67%	95.56%	39.58%	100.00%	39.58%	91.45%	39.58%	100.00%
Coverage Ratio, best.	22.92%	97.73%	22.92%	100.00%	33.33%	93.46%	33.33%	100.00%
Strict, Max	79.17%	82.46%	75.00%	100.00%	43.75%	87.65%	37.50%	100.00%
Strict, avg.	37.50%	97.73%	37.50%	100.00%	33.33%	97.73%	33.33%	100.00%
Strict, psum	79.17%	82.46%	75.00%	100.00%	43.75%	87.65%	37.50%	100.00%
Strict, best.	37.50%	89.52%	37.50%	89.52%	35.42%	97.73%	35.42%	100.00%

Table 53: Results from different testing methods on discriminant rules

8.4.7 Experiment 040620-2: Comparison of Testing Methods on Characteristic Models

Training Dataset:

Discretization: Dis-3

Filtering: Significance based, conjunctive, rank-threshold = 10, TR+TS

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

AQ21 Learning Parameters:

maxstar = 1 maxrule = 1 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Characteristic descriptions

Evaluation of Conjunction, Evaluation of Disjunction	EPIC-MT				EPIC-RB			
	Correct	Prec.	First Choie Correct	First Choice Prec.	Correct	Prec.	First Choie Correct	First Choice Prec.
Selectors Ratio, Max	81.25%	45.63%	56.25%	100.00%	39.58%	89.52%	37.50%	100.00%
Selectors Ratio, avg.	45.83%	70.94%	39.58%	100.00%	37.50%	93.46%	35.42%	100.00%
Selectors Ratio, psum	81.25%	16.67%	37.50%	100.00%	37.50%	84.13%	35.42%	100.00%
Selectors Ratio, best.	43.75%	74.91%	37.50%	100.00%	35.42%	89.52%	33.33%	100.00%
Coverage Ratio, Max	33.33%	91.45%	31.25%	100.00%	39.58%	95.56%	37.50%	100.00%
Coverage Ratio, avg.	2.08%	100.00 %	2.08%	100.00%	39.58%	97.73%	39.58%	100.00%
Coverage Ratio, psum	37.50%	89.52%	35.42%	100.00%	39.58%	95.56%	37.50%	100.00%
Coverage Ratio, best.	20.83%	100.00 %	20.83%	100.00%	35.42%	97.73%	33.33%	100.00%
Strict, Max	81.25%	79.28%	75.00%	97.73%	45.83%	85.86%	41.67%	100.00%
Strict, avg.	27.08%	97.73%	25.00%	100.00%	33.33%	97.73%	33.33%	100.00%
Strict, psum	81.25%	79.28%	75.00%	97.73%	45.83%	85.86%	41.67%	100.00%
Strict, best.	37.50%	84.13%	35.42%	89.52%	35.42%	95.56%	35.42%	100.00%

Table 54: Results from different testing methods on characteristic rules

8.4.8 Summary of Experiments 040620-1 and 040620-2:

For both characteristic and discriminant descriptions, EPIC-MT performed better than EPIC-RB. It was not surprising that in all cases, strict rule match with maximum or probabilistic sum for evaluation of rulesets gave the best results. The selectors ratio method gives also reasonably good results but usually with low precision. Degrees of match to all models are very similar; for example, see experiment 040607-2.

This result encouraged further investigation of testing methods (experiment 040620-3, below).

8.4.9 Experiment 040620-3: EPIC-SDA Testing Method

Training Dataset:

Discretization: Dis-3

Filtering: Significance based, conjunctive, rank-threshold = 10, TR+TS

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

AQ21 Learning Parameters:

maxstar = 1 maxrule = 1 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Discriminant descriptions

Matching parameters:

Evaluation of Selector: Strict

Evaluation of Conjunction: Strict

Evaluation of Disjunction: Maximum

SDA threshold	SDA probe	Correct	Precision	First Choice Correct	First Choice Precision
1.5	10	50.00%	87.65%	47.92%	100.00%
	20	60.42%	87.65%	58.33%	100.00%
	50	72.92%	85.86%	68.75%	100.00%
	100	72.92%	85.86%	68.75%	100.00%
	200	77.08%	82.46%	72.92%	100.00%
	300	77.08%	82.46%	72.92%	100.00%
	500	79.17%	82.46%	75.00%	100.00%
2	10	64.58%	82.46%	60.42%	100.00%
	20	72.92%	82.46%	68.75%	100.00%
	50	77.08%	82.46%	72.92%	100.00%
	100	77.08%	82.46%	72.92%	100.00%
	200	77.08%	82.46%	72.92%	100.00%
	300	79.17%	82.46%	75.00%	100.00%
	500	79.17%	82.46%	75.00%	100.00%
3	10	72.92%	82.46%	68.75%	100.00%
	20	77.08%	82.46%	72.92%	100.00%
	50	77.08%	82.46%	72.92%	100.00%
	100	77.08%	82.46%	72.92%	100.00%
	200	79.17%	82.46%	75.00%	100.00%
	300	79.17%	82.46%	75.00%	100.00%
	500	79.17%	82.46%	75.00%	100.00%

Table 55: Results of testing discriminant descriptions for different settings of SDA threshold and SDA probe EPIC-SDA parameters.

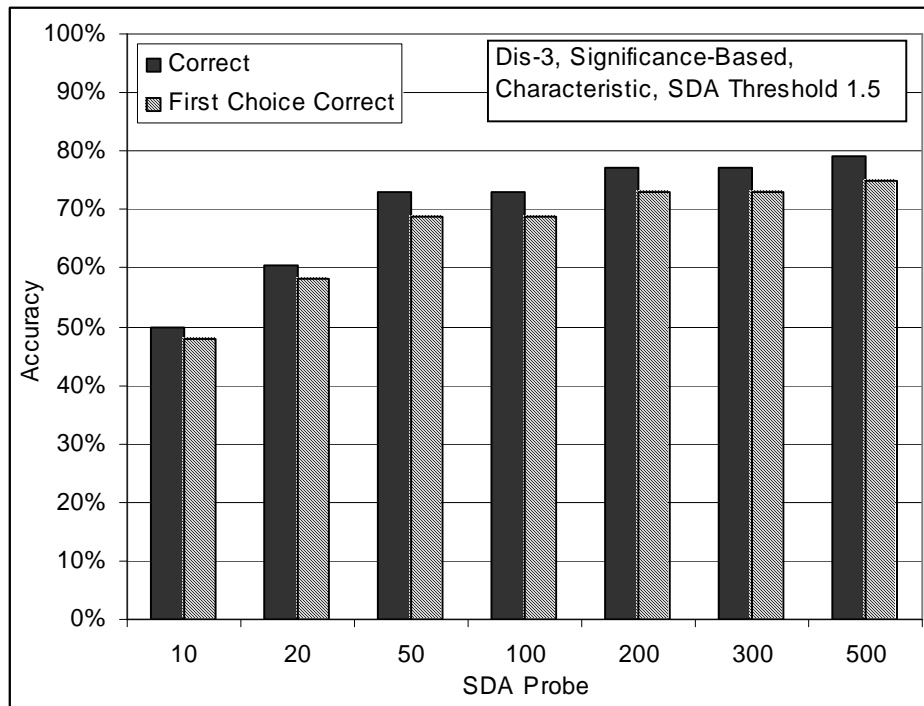


Figure 110: Classification accuracy for selected values of SDA Probe, SDA Threshold 1.5, characteristic rules.

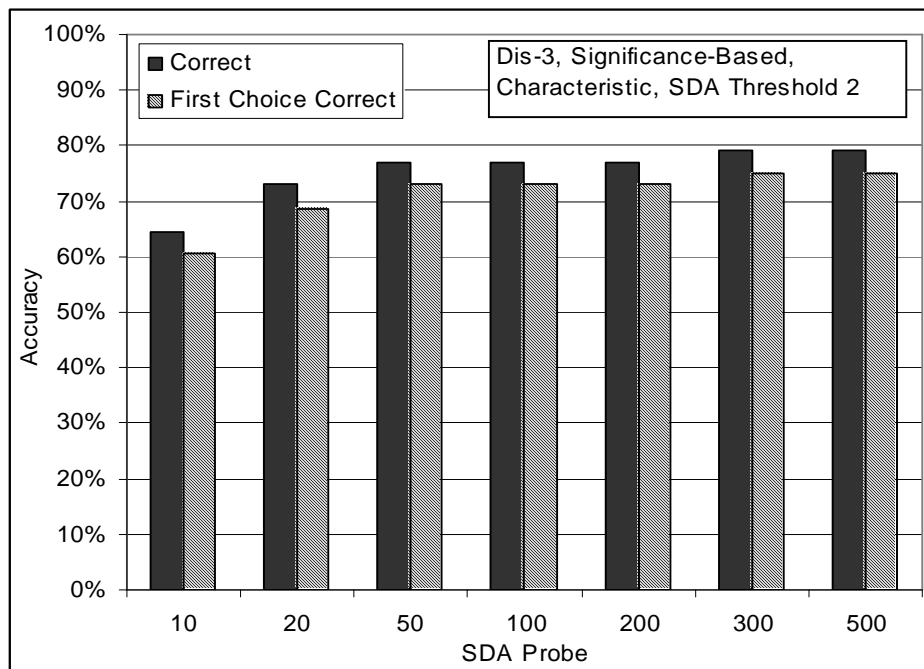


Figure 111: Classification accuracy for selected values of SDA Probe, SDA Threshold 2, characteristic rules

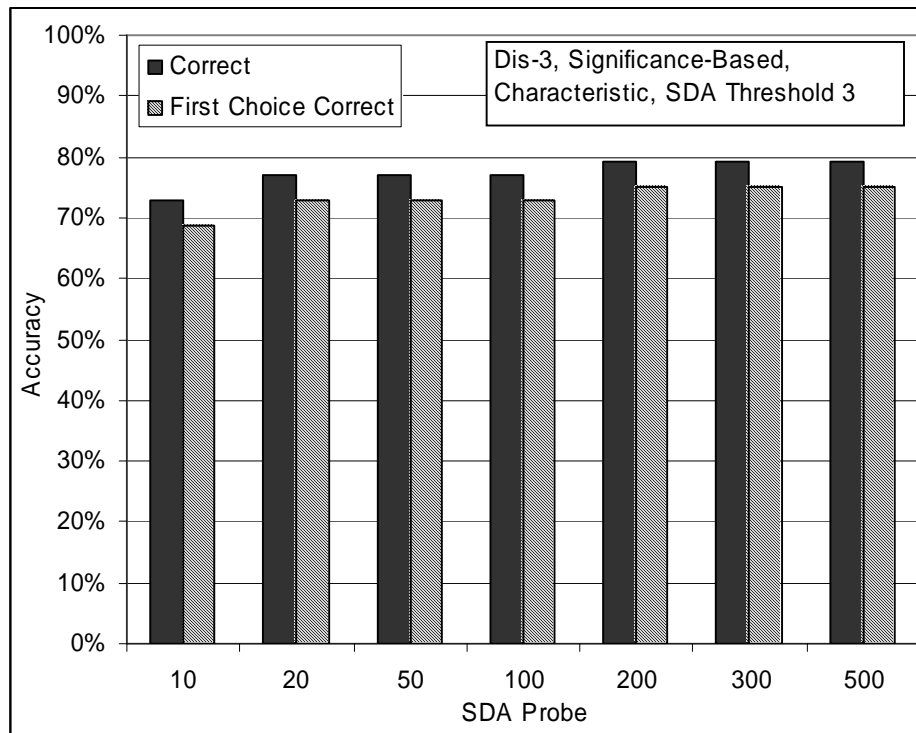


Figure 112: Classification accuracy for selected values of SDA Probe, SDA Threshold 3, characteristic rules.

EPIC-SDA when set up with sufficiently large values of SDA threshold and SDA probe provide as good results as a standard EPIC program. Figures above show that the same result can be obtained using different settings of the two parameters.

As described in Section 4.2.2, EPIC-SDA is a very useful modification of the EPIC algorithm that does not need entire episodes for classification, but instead stops whenever one model clearly “wins” over other models.

8.4.10 Experiment 040624-1: User-oriented Attribute Sets on Unfiltered Data, Characteristic Descriptions

Training Dataset:

Discretization: Dis-3

Filtering: not filtered

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

Attribute Selection: Based on Gain Ratio

Attributes	User #									
	1	2	3	4	5	7	8	12	19	25
process_name										
event_status										
proc_cpu_time										
proc_inactive_time_gt1min										
proc_cpu_time_in_win_lf										
delta_time_new_window										
win_time_elapsed_lf										
prot_words_chars										
prot_words_chars_to_total_chars_ratio										
new_win_time_elapsed										
new_win_time_elapsed_lf										
proc_count_in_win										
win_opened										
win_opened_lf										
win_title_total_words										
win_title_prot_words										

Table 56: User oriented attribute selection. Shaded attributes are selected.

AQ21 Learning Parameters:

maxstar = 1 maxrule = 1 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Characteristic descriptions

Testing Parameters:

Evaluation of Conjunction = strict

Evaluation of Disjunction = max

Acceptance Threshold = 10 %

Accuracy Tolerance = 5%

Testing Results:

Correct: 62.50%

Precision: 39.68%

First Choice Correct: 35.42%

First Choice Precision: 95.56%

	User 1	User 2	User 3	User 4	User 5	User 7	User 8	User 12	User 19	User 25
User 1 (Correct: 60%, First Choice Correct: 20%)										
Epi.281	0.799	0.764	0.012	0.533	0.488	0.430	0.555	0.681	0.490	0.890
Epi.282	0.534	0.740	0.063	0.464	0.187	0.432	0.131	0.189	0.398	0.556
Epi.283	0.806	0.819	0.000	0.715	0.667	0.729	0.653	0.667	0.653	0.625
Epi.284	0.869	0.836	0.000	0.743	0.705	0.601	0.721	0.738	0.727	0.727
Epi.285	0.789	0.821	0.005	0.692	0.488	0.656	0.475	0.608	0.631	0.796

User 2 (Correct: 40%, First Choice Correct: 20%)

Epi.288	0.137	0.864	0.000	0.896	0.270	0.351	0.276	0.468	0.485	0.413
Epi.289	0.577	0.682	0.000	0.768	0.514	0.660	0.554	0.727	0.710	0.892
Epi.290	0.074	0.919	0.001	0.880	0.186	0.486	0.375	0.847	0.857	0.573
Epi.291	0.273	0.403	0.007	0.453	0.254	0.517	0.164	0.143	0.295	0.592
Epi.333	0.264	0.617	0.000	0.550	0.058	0.361	0.116	0.424	0.361	0.864

User 3 (Correct: 33%, First Choice Correct: 33%)

Epi.345	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Epi.347	0.000	0.143	0.071	0.143	0.214	0.000	0.000	0.000	0.000	0.000
Epi.349	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

User 4 (Correct: 60%, First Choice Correct: 40%)

Epi.391	0.236	0.901	0.000	0.984	0.924	0.236	0.930	0.918	0.883	0.253
Epi.392	0.659	0.967	0.001	0.907	0.862	0.518	0.842	0.967	0.938	0.685
Epi.393	0.604	0.819	0.001	0.962	0.939	0.587	0.927	0.967	0.866	0.610
Epi.394	0.254	0.897	0.000	0.643	0.206	0.444	0.222	0.683	0.619	0.643
Epi.512	0.435	0.963	0.000	0.966	0.910	0.485	0.872	0.904	0.909	0.490

User 5 (Correct: 60%, First Choice Correct: 0%)

Epi.513	0.017	0.663	0.001	0.980	0.969	0.053	0.517	0.954	0.489	0.055
Epi.514	0.043	0.583	0.003	0.924	0.885	0.039	0.880	0.891	0.108	0.038
Epi.515	0.208	0.967	0.000	0.332	0.326	0.208	0.953	0.328	0.299	0.210
Epi.542	0.423	0.726	0.001	0.957	0.947	0.401	0.914	0.961	0.598	0.423
Epi.543	0.160	0.613	0.003	0.656	0.656	0.156	0.860	0.632	0.415	0.191

User 7 (Correct: 60%, First Choice Correct: 40%)

Epi.734	0.686	0.789	0.005	0.538	0.395	0.928	0.583	0.951	0.525	0.753
Epi.735	0.626	0.848	0.002	0.888	0.597	0.958	0.714	0.911	0.788	0.888
Epi.736	0.546	0.766	0.000	0.234	0.247	0.533	0.299	0.714	0.182	0.766
Epi.737	0.833	0.227	0.000	0.896	0.869	0.968	0.833	0.936	0.845	0.912
Epi.738	0.676	0.817	0.000	0.603	0.563	0.565	0.501	0.838	0.610	0.815

User 8 (Correct: 40%, First Choice Correct: 0%)

Epi.741	0.835	0.747	0.000	0.834	0.854	0.662	0.817	0.936	0.859	0.851
Epi.742	0.372	0.795	0.000	0.777	0.400	0.326	0.763	0.512	0.707	0.730
Epi.743	0.108	0.717	0.002	0.881	0.716	0.137	0.850	0.851	0.567	0.216
Epi.744	0.158	0.804	0.002	0.922	0.718	0.213	0.600	0.814	0.758	0.297
Epi.897	0.854	0.656	0.002	0.800	0.645	0.771	0.811	0.909	0.838	0.880

User 12 (Correct: 80%, First Choice Correct: 60%)

Epi.980	0.692	0.785	0.000	0.860	0.774	0.634	0.858	0.959	0.826	0.776
Epi.981	0.412	0.906	0.000	0.979	0.816	0.469	0.897	0.963	0.619	0.525
Epi.982	0.632	0.438	0.003	0.951	0.559	0.641	0.691	0.692	0.458	0.687
Epi.983	0.477	0.670	0.000	0.862	0.821	0.350	0.839	0.929	0.514	0.525
Epi.984	0.131	0.508	0.000	0.220	0.216	0.082	0.217	0.921	0.214	0.132

User 19 (Correct: 80%, First Choice Correct: 40%)

Epi.1040	0.912	0.948	0.000	0.943	0.932	0.922	0.912	0.932	0.912	0.906
Epi.1041	0.963	0.719	0.000	0.972	0.926	0.955	0.960	0.966	0.974	0.969

Epi.1042	0.552	0.873	0.000	0.814	0.532	0.651	0.587	0.889	0.937	0.790
Epi.1043	0.848	0.465	0.000	0.959	0.686	0.876	0.934	0.978	0.897	0.890
Epi.1044	0.625	0.970	0.000	0.963	0.796	0.652	0.818	0.944	0.935	0.710

User 25 (Correct: 100%, First Choice Correct: 100%)

Epi.1195	0.685	0.824	0.000	0.449	0.364	0.554	0.394	0.757	0.453	0.959
Epi.1196	0.818	0.944	0.000	0.356	0.340	0.374	0.403	0.910	0.421	0.985
Epi.1197	0.774	0.965	0.000	0.367	0.337	0.417	0.337	0.965	0.337	0.970
Epi.1198	0.815	0.925	0.000	0.310	0.265	0.341	0.288	0.978	0.290	0.993
Epi.1199	0.746	0.889	0.000	0.219	0.114	0.415	0.124	0.645	0.209	0.971

Table 57: Testing results for experiment 040624-1 (User-oriented attribute sets).

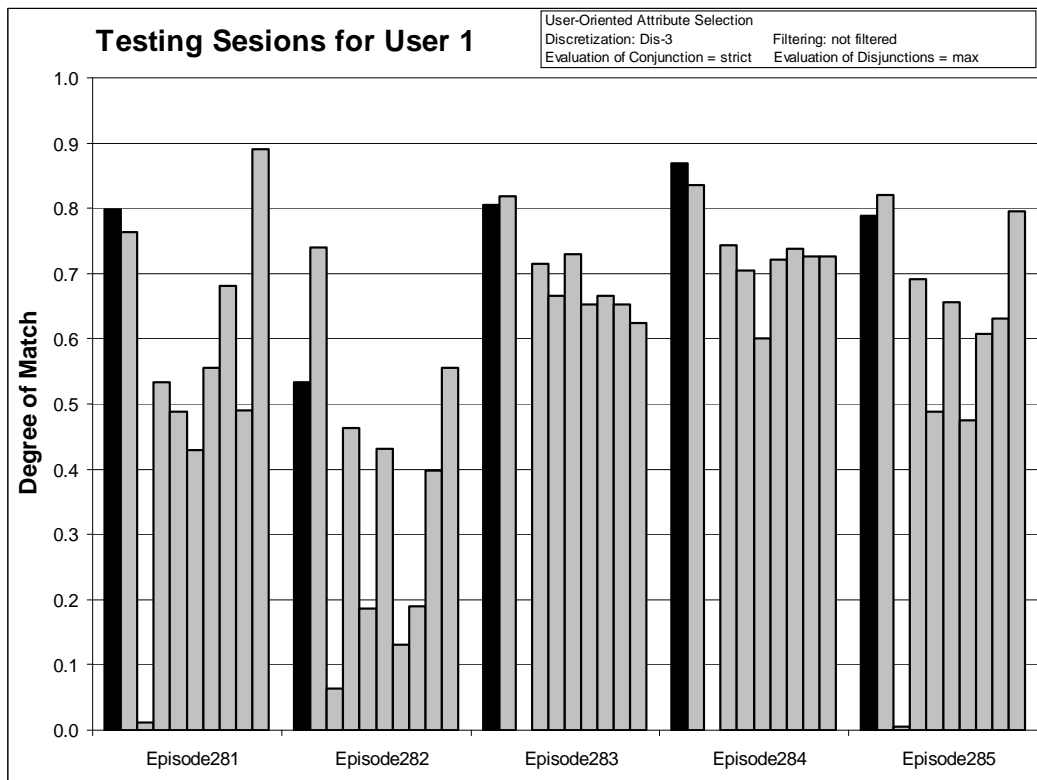


Figure 113: Degrees of match between 10 user models and 5 testing sessions from User 1.

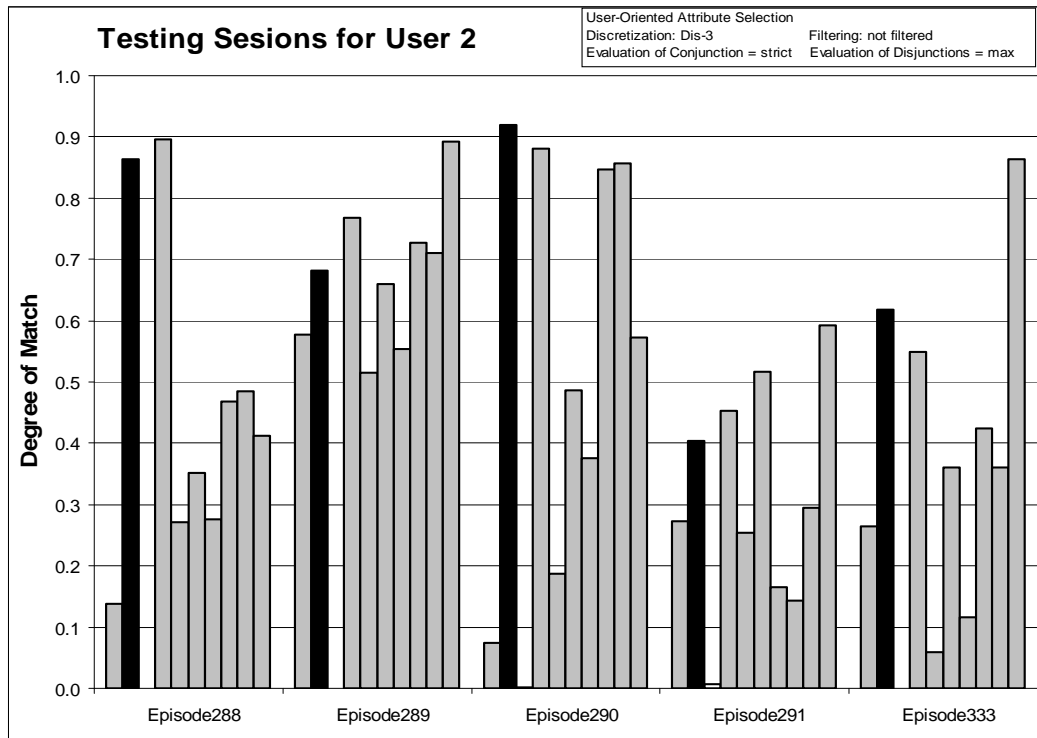


Figure 114: Degrees of match between 10 user models and 5 testing sessions from User 2.

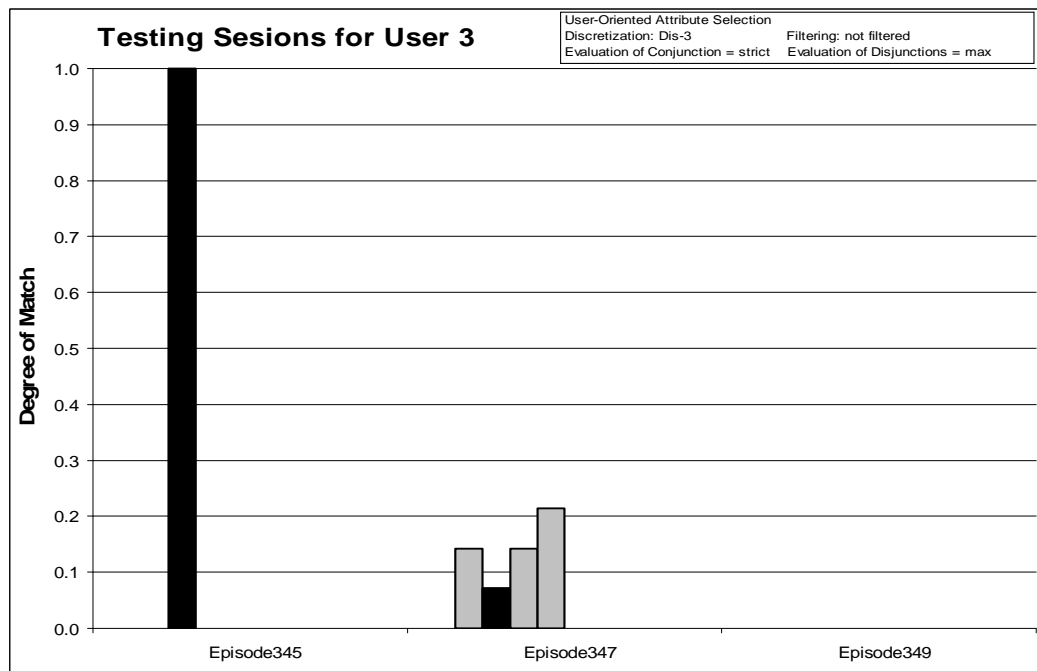


Figure 115: Degrees of match between 10 user models and 3 testing sessions from User 3.

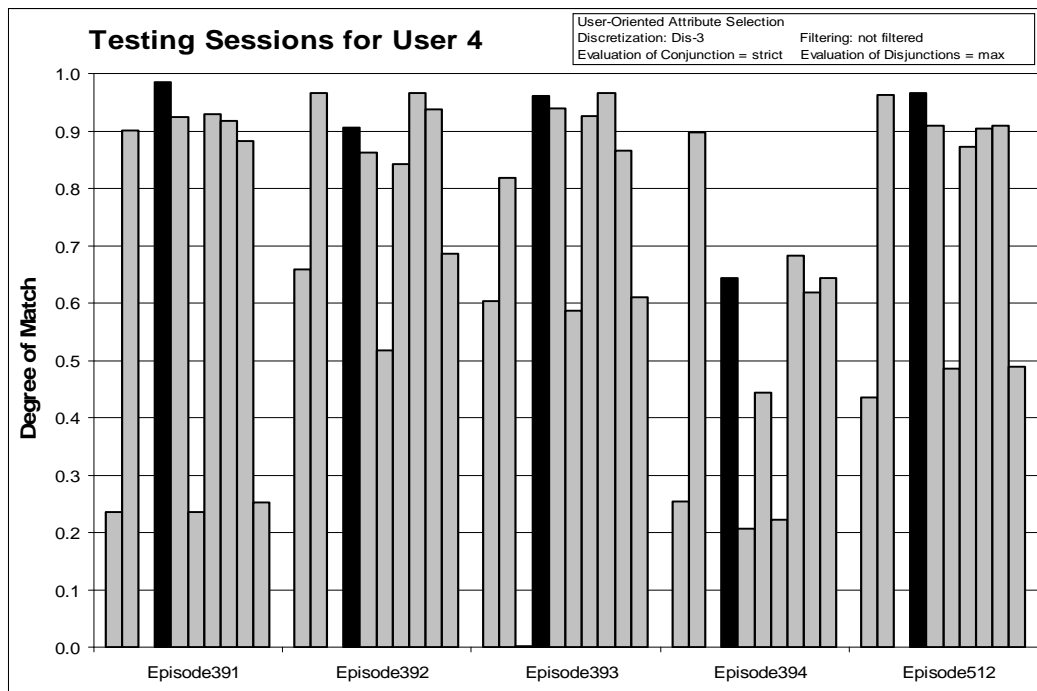


Figure 116: Degrees of match between 10 user models and 5 testing sessions from User 4.

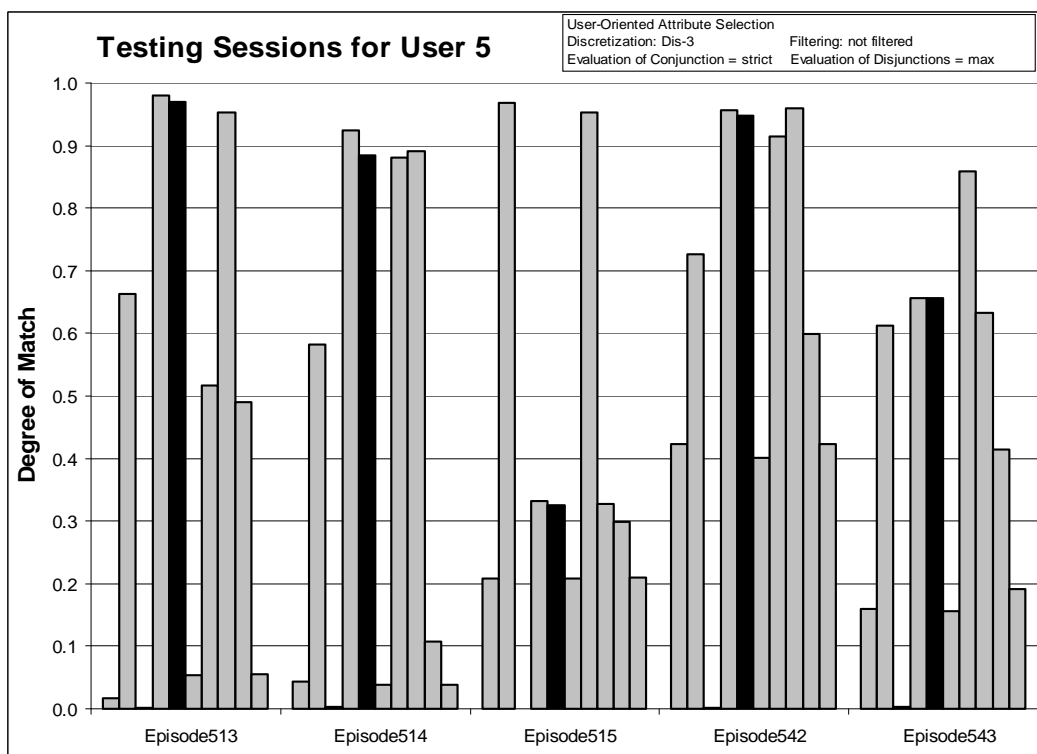


Figure 117: Degrees of match between 10 user models and 5 testing sessions from User 5

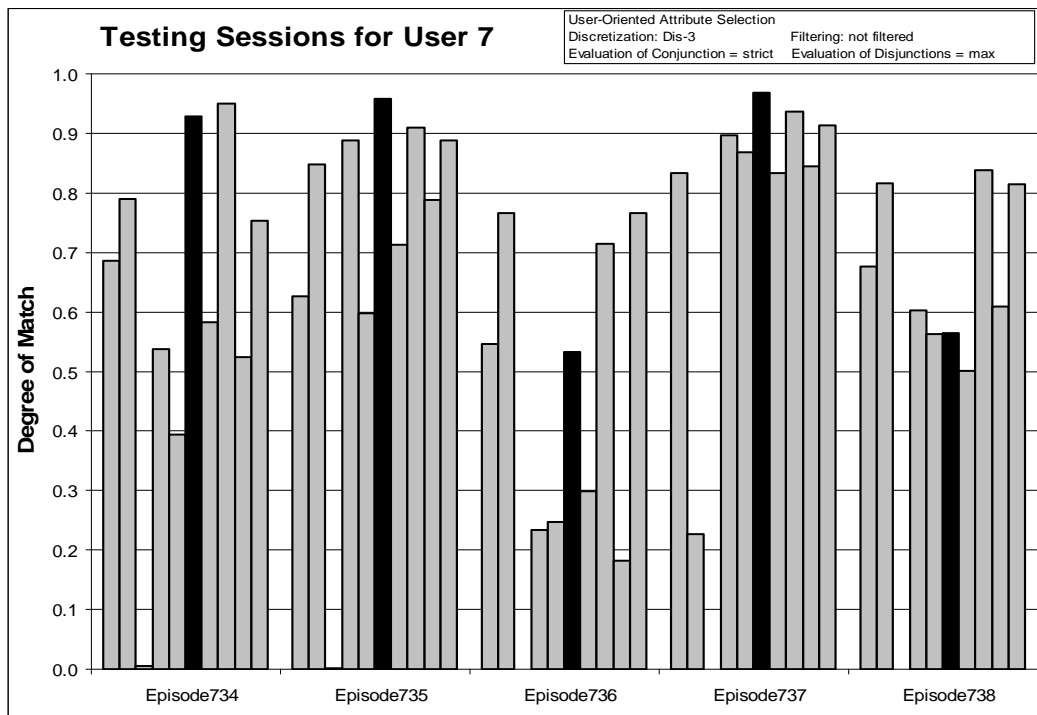


Figure 118: Degrees of match between 10 user models and 5 testing sessions from User 7.

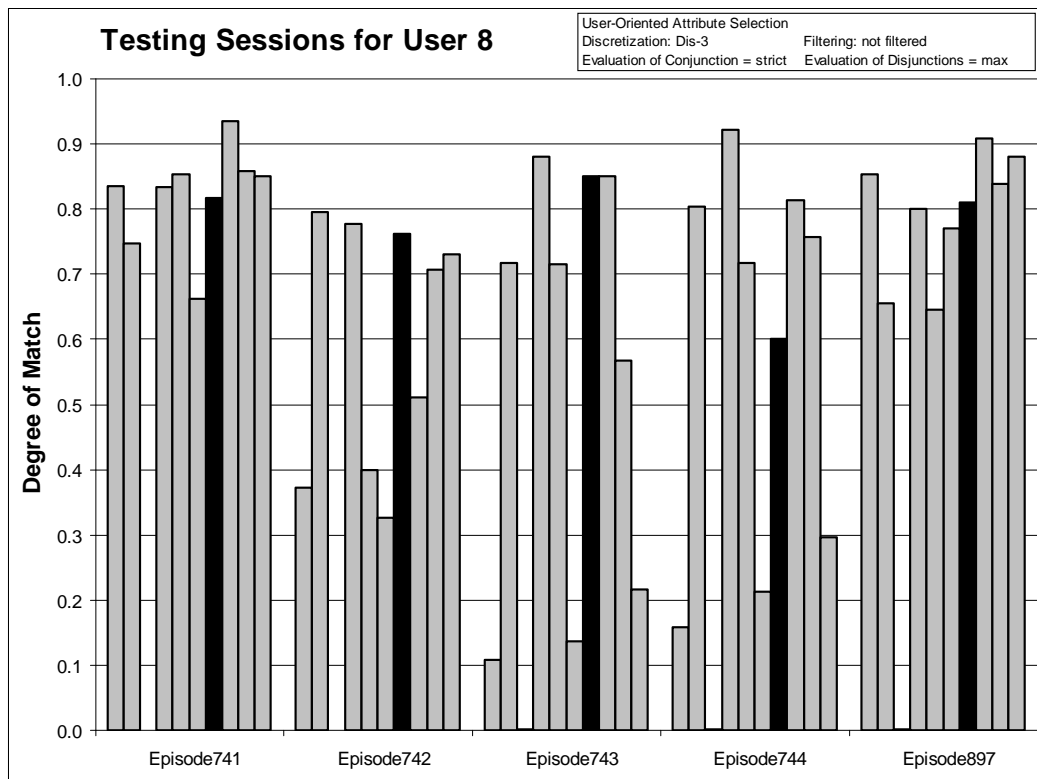


Figure 119: Degrees of match between 10 user models and 5 testing sessions from User 8.

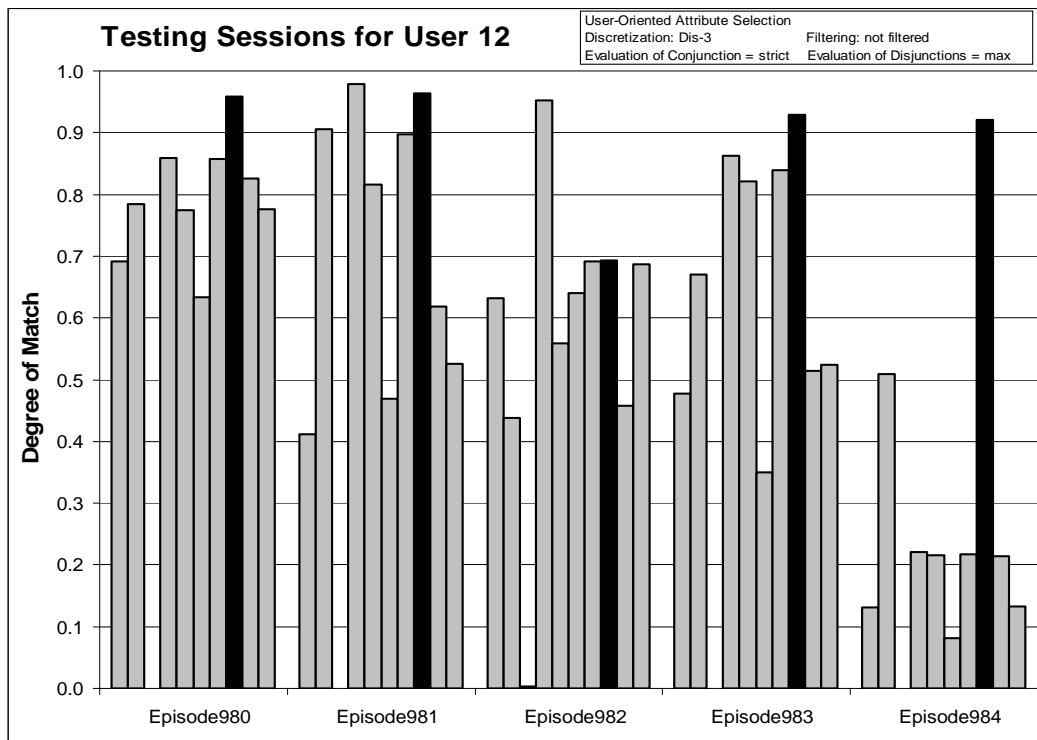


Figure 120: Degrees of match between 10 user models and 5 testing sessions from User 12.

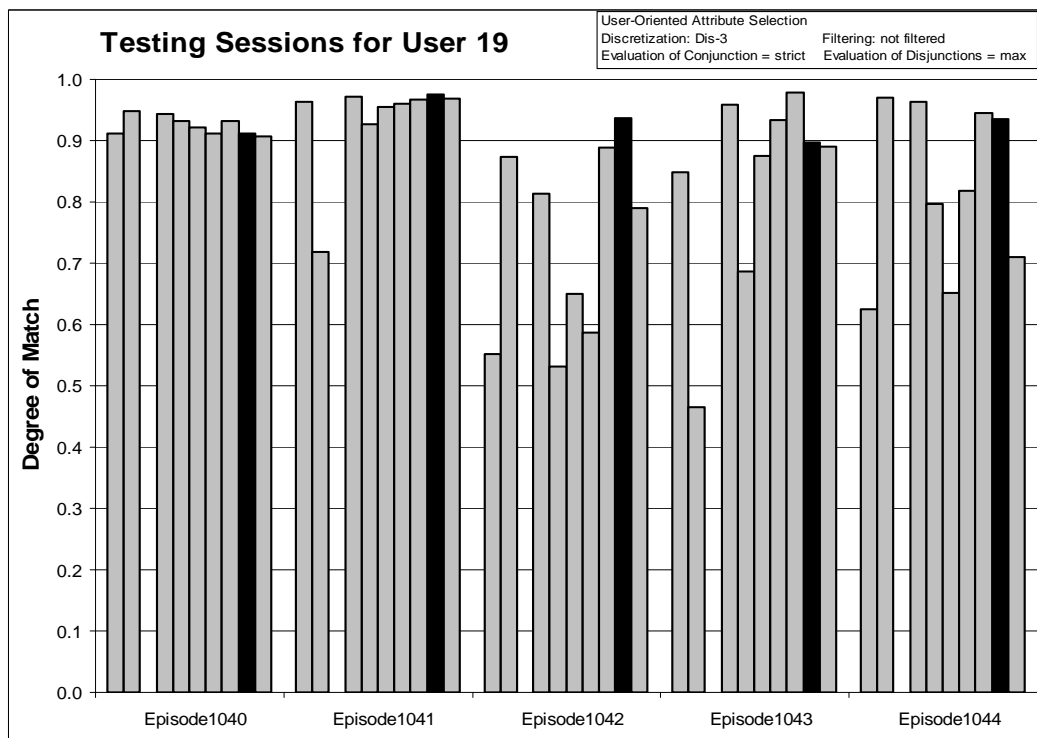


Figure 121: Degrees of match between 10 user models and 5 testing sessions from User 19.

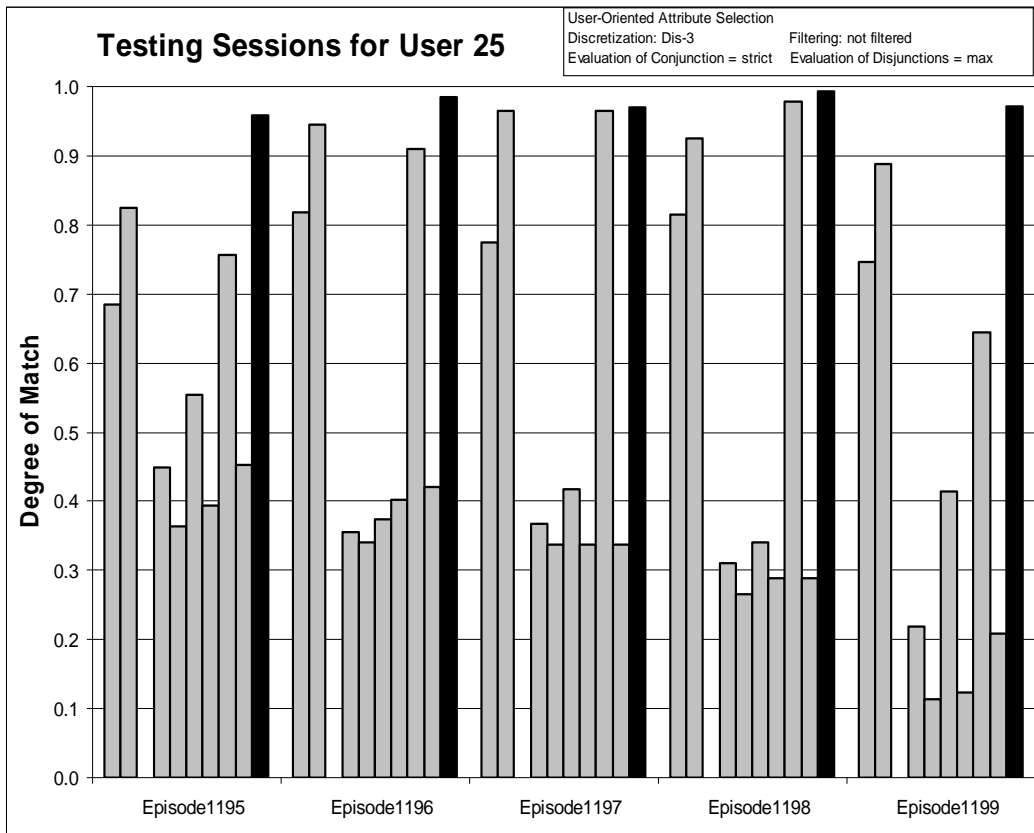


Figure 122: Degrees of match between 10 user models and 5 testing sessions from User 25.

8.4.11 Experiment 040624-2: User-oriented Attribute Sets on Unfiltered Data, Discriminant Descriptions

Training Dataset:

Discretization: Dis-3

Filtering: not filtered

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

Attribute Selection: Based on Gain Ratio

Attributes	User #									
	1	2	3	4	5	7	8	12	19	25
process_name										
event_status										
proc_cpu_time										
proc_inactive_time_gt1min										
proc_cpu_time_in_win_lf										
delta_time_new_window										
win_time_elapsed_lf										
prot_words_chars										
prot_words__chars_to_total _chars_ratio										
new_win_time_elapsed										
new_win_time_elapsed_lf										
proc_count_in_win										
win_opened										
win_opened_lf										
win_title_total_words										
win_title_prot_words										

Table 58: User oriented attribute selection. Shaded attributes are selected.

AQ21 Learning Parameters:

maxstar = 1 maxrule = 1 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Discriminant descriptions

Testing Parameters:

Evaluation of Conjunction = strict

Evaluation of Disjunction = max

Acceptance Threshold = 10 %

Accuracy Tolerance = 5%

Testing Results:

Correct: 62.50%

Precision: 39.68%

First Choice Correct: 35.42%

First Choice Precision: 95.56%

Learning Results:

Total number of rules: 4536

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	316	818	37	523	363	346	471	285	429	348

Table 59: Number of learned rules for 10 Users.

Testing Results:

Correct: 62.50%

Precision: 39.68%

First Choice Correct: 35.42%

First Choice Precision: 95.56%

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
Correct	60%	40%	33%	60%	60%	60%	40%	80%	80%	100%
First Ch. Correct	20%	20%	33%	40%	0%	40%	0%	60%	40%	100%

Table 60: Summary of correct answers for 10 users.

	User 1	User 2	User 3	User 4	User 5	User 7	User 8	User 12	User 19	User 25
User 1 (Correct: 60% First Choice Correct: 20%)										
Epi.281	0.801	0.766	0.011	0.533	0.488	0.454	0.557	0.681	0.491	0.892
Epi.282	0.551	0.752	0.066	0.466	0.197	0.435	0.141	0.197	0.418	0.556
Epi.283	0.806	0.826	0.000	0.715	0.681	0.729	0.660	0.667	0.646	0.632
Epi.284	0.863	0.847	0.000	0.743	0.710	0.601	0.721	0.738	0.727	0.727
Epi.285	0.789	0.823	0.006	0.693	0.499	0.657	0.480	0.610	0.635	0.794
User 2 (Correct: 40% First Choice Correct: 20%)										
Epi.288	0.137	0.865	0.000	0.898	0.274	0.354	0.278	0.470	0.488	0.414
Epi.289	0.579	0.682	0.000	0.766	0.514	0.660	0.554	0.732	0.714	0.896
Epi.290	0.074	0.921	0.001	0.880	0.186	0.489	0.381	0.849	0.856	0.573
Epi.291	0.274	0.415	0.007	0.467	0.261	0.517	0.166	0.161	0.300	0.594
Epi.333	0.264	0.620	0.000	0.550	0.061	0.361	0.116	0.426	0.366	0.864
User 3 (Correct: 33% First Choice Correct: 33%)										
Epi.345	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Epi.347	0.000	0.143	0.071	0.143	0.214	0.071	0.000	0.000	0.000	0.000
Epi.349	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
User 4 (Correct: 60% First Choice Correct: 40%)										
Epi.391	0.236	0.903	0.000	0.985	0.924	0.236	0.930	0.918	0.883	0.253
Epi.392	0.659	0.967	0.001	0.909	0.862	0.521	0.841	0.967	0.938	0.686
Epi.393	0.604	0.820	0.002	0.962	0.939	0.587	0.927	0.969	0.867	0.610
Epi.394	0.278	0.897	0.000	0.659	0.222	0.444	0.254	0.683	0.611	0.675
Epi.512	0.435	0.964	0.000	0.969	0.911	0.485	0.873	0.904	0.910	0.490
User 5 (Correct: 60% First Choice Correct: 0%)										
Epi.513	0.017	0.664	0.001	0.981	0.970	0.056	0.517	0.954	0.490	0.055
Epi.514	0.043	0.586	0.004	0.925	0.887	0.041	0.883	0.891	0.110	0.038
Epi.515	0.208	0.969	0.000	0.333	0.325	0.208	0.960	0.329	0.300	0.210
Epi.542	0.424	0.726	0.001	0.958	0.949	0.401	0.916	0.961	0.598	0.423
Epi.543	0.161	0.613	0.003	0.657	0.655	0.156	0.863	0.633	0.415	0.195
User 7 (Correct: 60% First Choice Correct: 40%)										
Epi.734	0.686	0.789	0.005	0.538	0.395	0.928	0.583	0.951	0.525	0.753
Epi.735	0.626	0.848	0.000	0.888	0.597	0.962	0.714	0.913	0.788	0.890

Epi.736	0.546	0.766	0.000	0.234	0.247	0.533	0.299	0.714	0.182	0.766
Epi.737	0.833	0.227	0.000	0.896	0.873	0.972	0.837	0.936	0.845	0.912
Epi.738	0.676	0.817	0.000	0.603	0.563	0.567	0.501	0.838	0.614	0.817

User 8 (Correct: 40% First Choice Correct: 0%)

Epi.741	0.834	0.752	0.000	0.839	0.863	0.664	0.823	0.937	0.863	0.856
Epi.742	0.367	0.800	0.000	0.777	0.400	0.330	0.767	0.516	0.721	0.735
Epi.743	0.108	0.718	0.002	0.884	0.716	0.138	0.855	0.851	0.568	0.219
Epi.744	0.157	0.804	0.002	0.922	0.719	0.214	0.601	0.814	0.761	0.297
Epi.897	0.861	0.658	0.002	0.802	0.645	0.773	0.814	0.911	0.859	0.883

User 12 (Correct: 80% First Choice Correct: 60%)

Epi.980	0.695	0.790	0.000	0.860	0.776	0.629	0.858	0.966	0.826	0.769
Epi.981	0.415	0.906	0.000	0.979	0.816	0.472	0.898	0.969	0.619	0.527
Epi.982	0.632	0.440	0.003	0.951	0.560	0.642	0.691	0.695	0.460	0.687
Epi.983	0.486	0.672	0.000	0.866	0.824	0.355	0.840	0.932	0.520	0.523
Epi.984	0.131	0.509	0.000	0.220	0.217	0.082	0.218	0.926	0.214	0.132

User 19 (Correct: 80% First Choice Correct: 40%)

Epi.1040	0.912	0.948	0.000	0.953	0.932	0.922	0.912	0.943	0.917	0.906
Epi.1041	0.963	0.719	0.000	0.972	0.932	0.955	0.960	0.966	0.977	0.969
Epi.1042	0.552	0.873	0.000	0.833	0.532	0.651	0.587	0.893	0.933	0.794
Epi.1043	0.848	0.465	0.000	0.961	0.688	0.876	0.934	0.978	0.898	0.890
Epi.1044	0.625	0.971	0.000	0.963	0.796	0.652	0.818	0.946	0.935	0.710

User 25 (Correct: 100% First Choice Correct: 100%)

Epi.1195	0.685	0.826	0.000	0.449	0.364	0.554	0.394	0.760	0.456	0.966
Epi.1196	0.819	0.946	0.000	0.358	0.340	0.374	0.404	0.910	0.423	0.986
Epi.1197	0.774	0.965	0.000	0.367	0.337	0.417	0.332	0.965	0.337	0.970
Epi.1198	0.815	0.926	0.000	0.310	0.265	0.341	0.288	0.978	0.291	0.994
Epi.1199	0.747	0.893	0.000	0.219	0.114	0.416	0.125	0.645	0.209	0.971

Table 61: Testing results for experiment 040624-2 (User-oriented attribute sets)

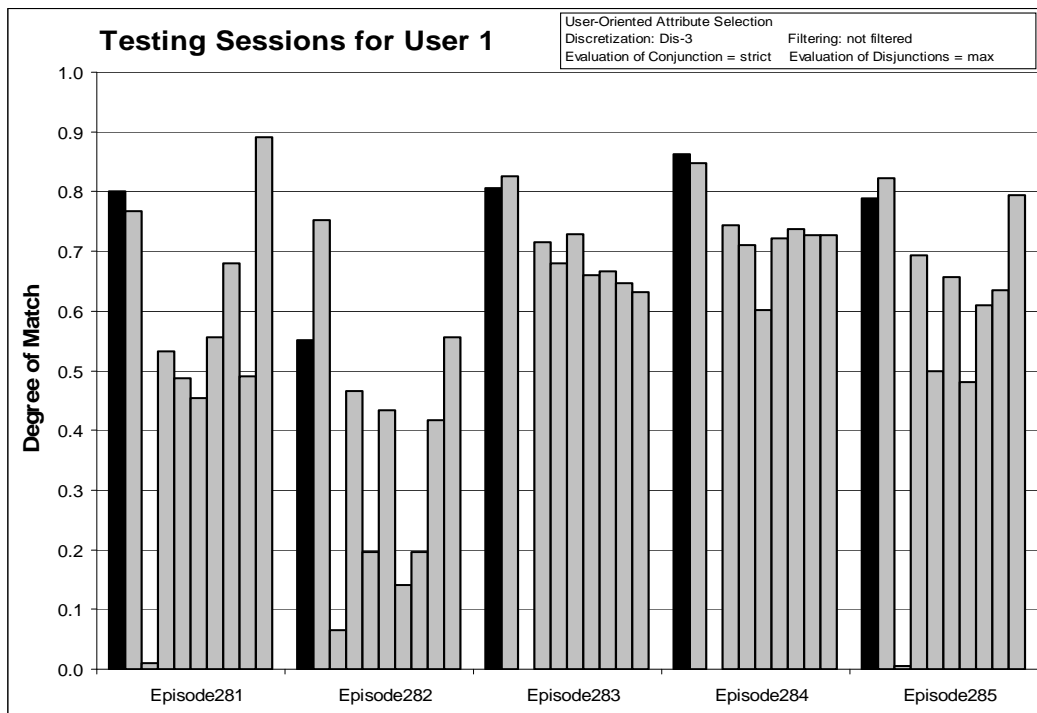


Figure 123: Degrees of match between 10 user models and 5 testing sessions from User 1.

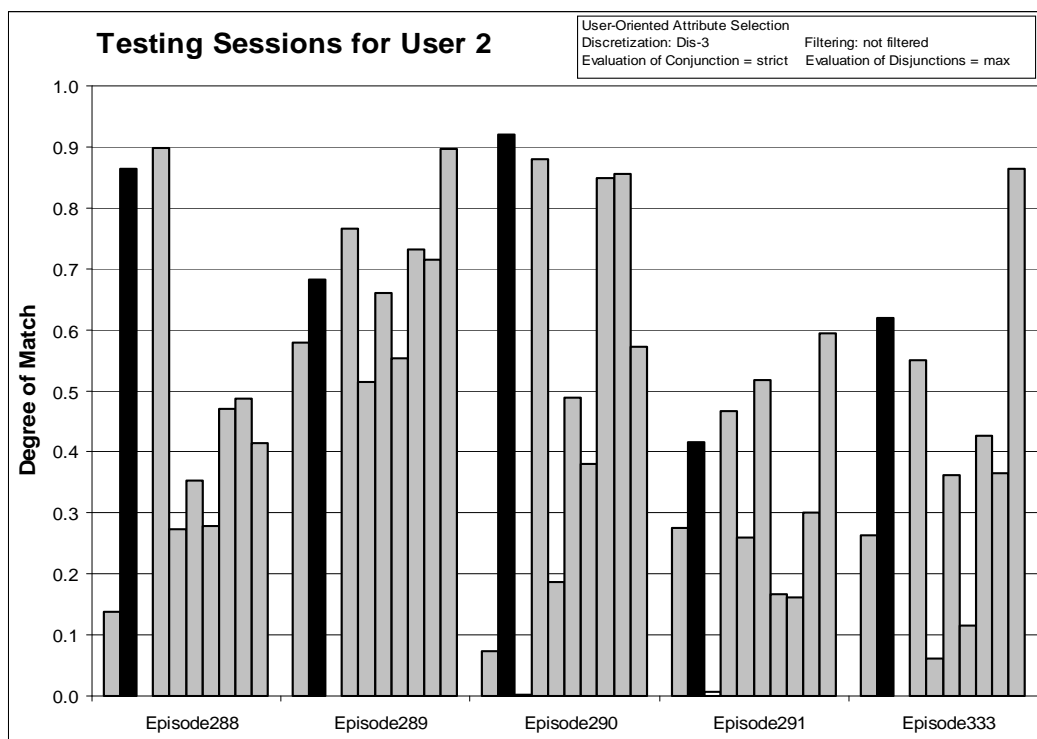


Figure 124: Degrees of match between 10 user models and 5 testing sessions from User 2

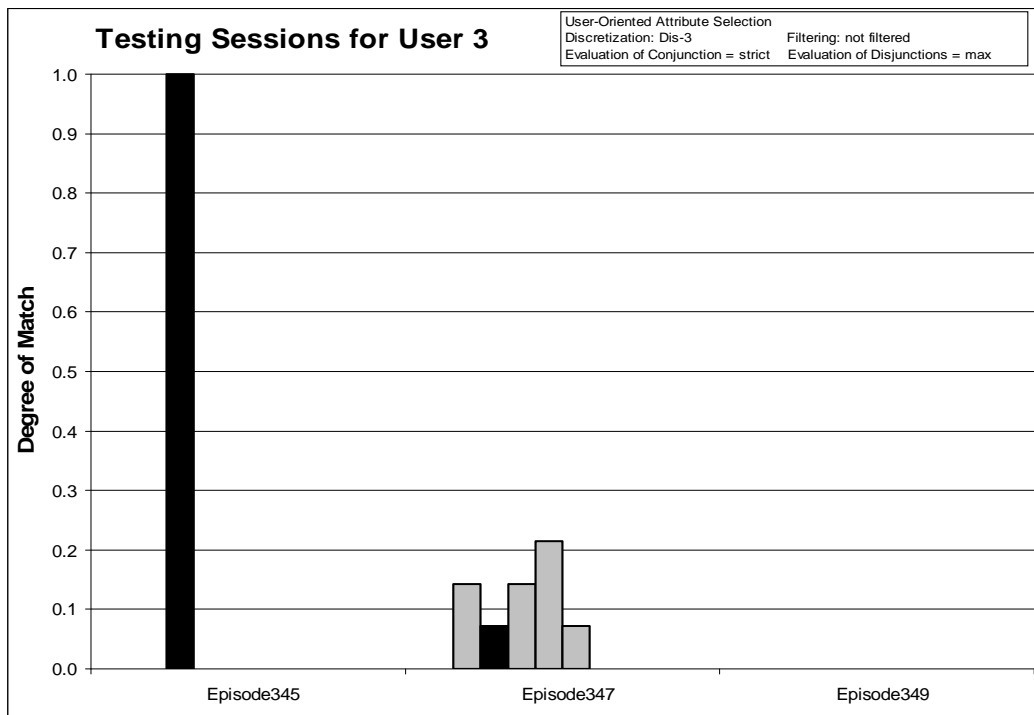


Figure 125: Degrees of match between 10 user models and 3 testing sessions from User 3.

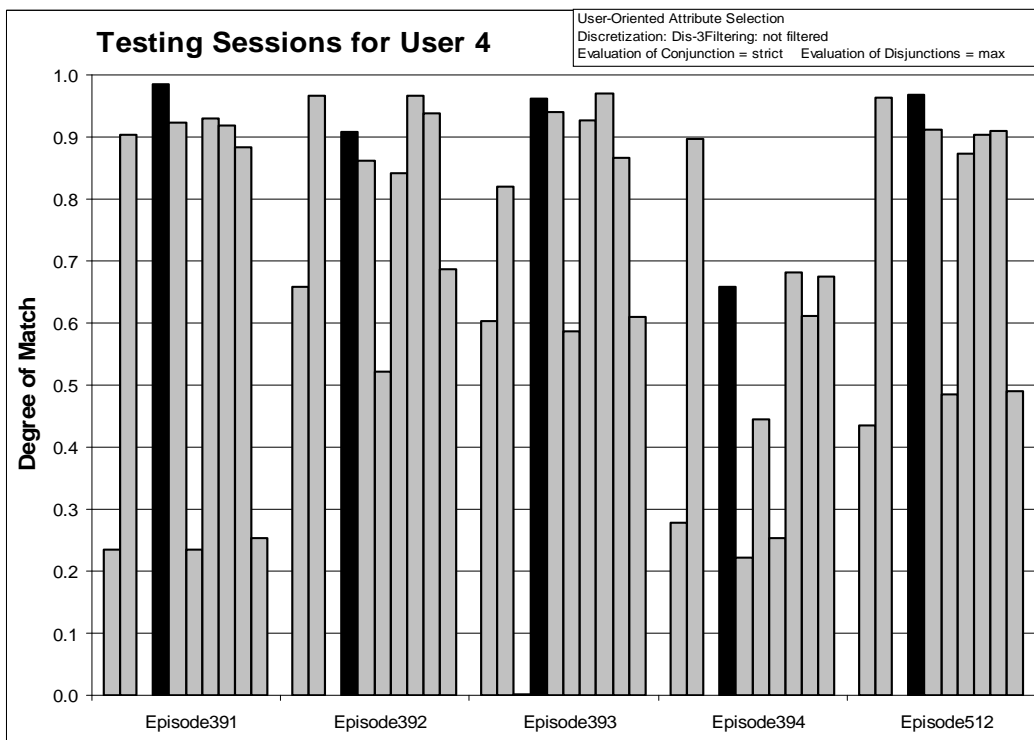


Figure 126: Degrees of match between 10 user models and 5 testing sessions from User 4.

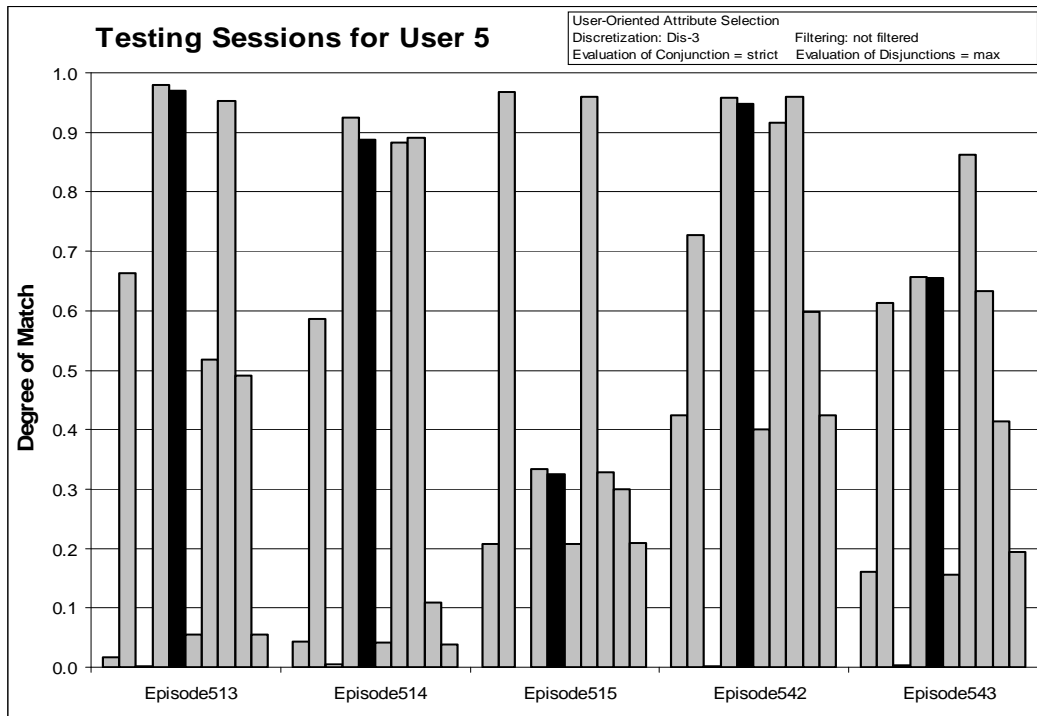


Figure 127: Degrees of match between 10 user models and 5 testing sessions from User 5.

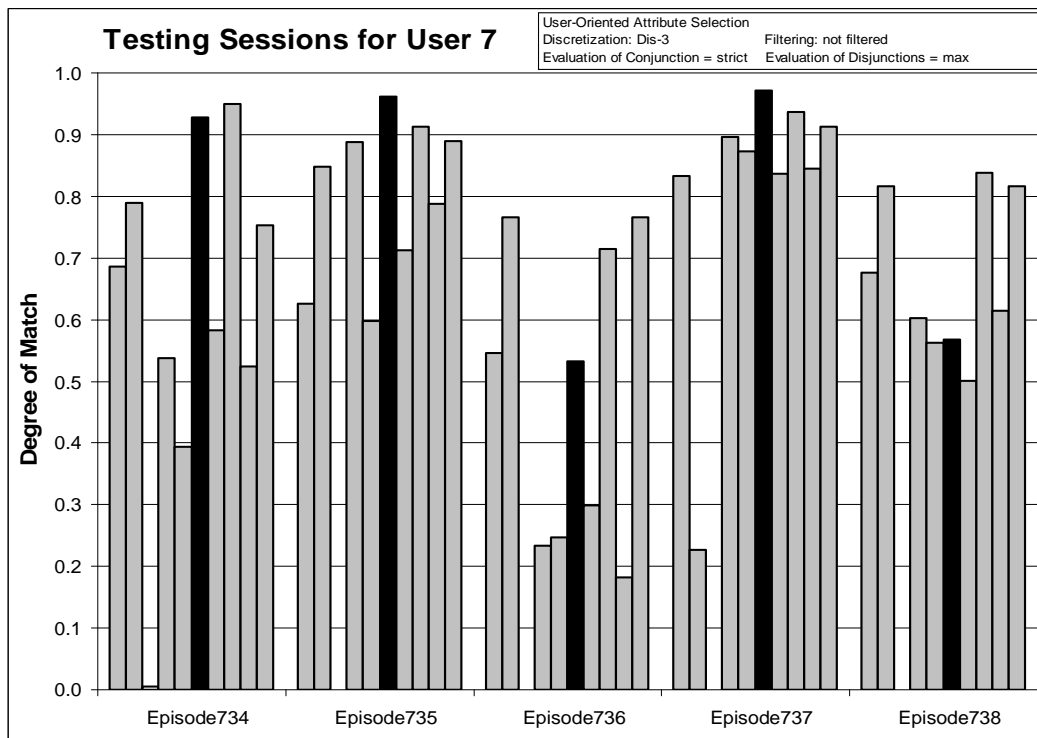


Figure 128: Degrees of match between 10 user models and 5 testing sessions from User 7.

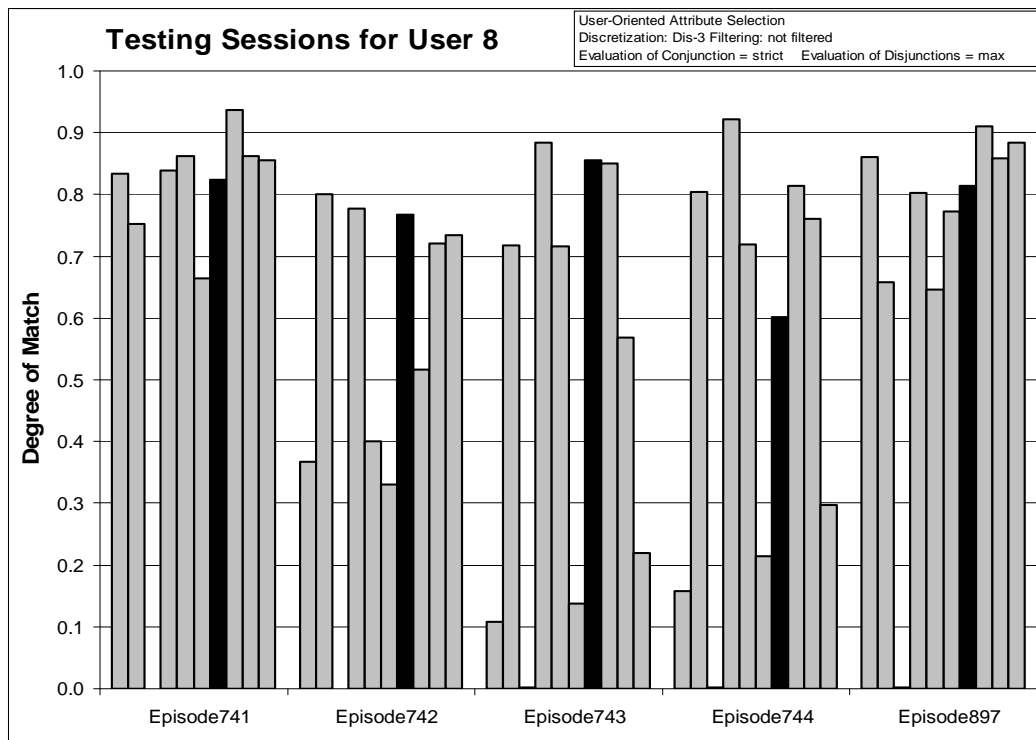


Figure 129: Degrees of match between 10 user models and 5 testing sessions from User 8.

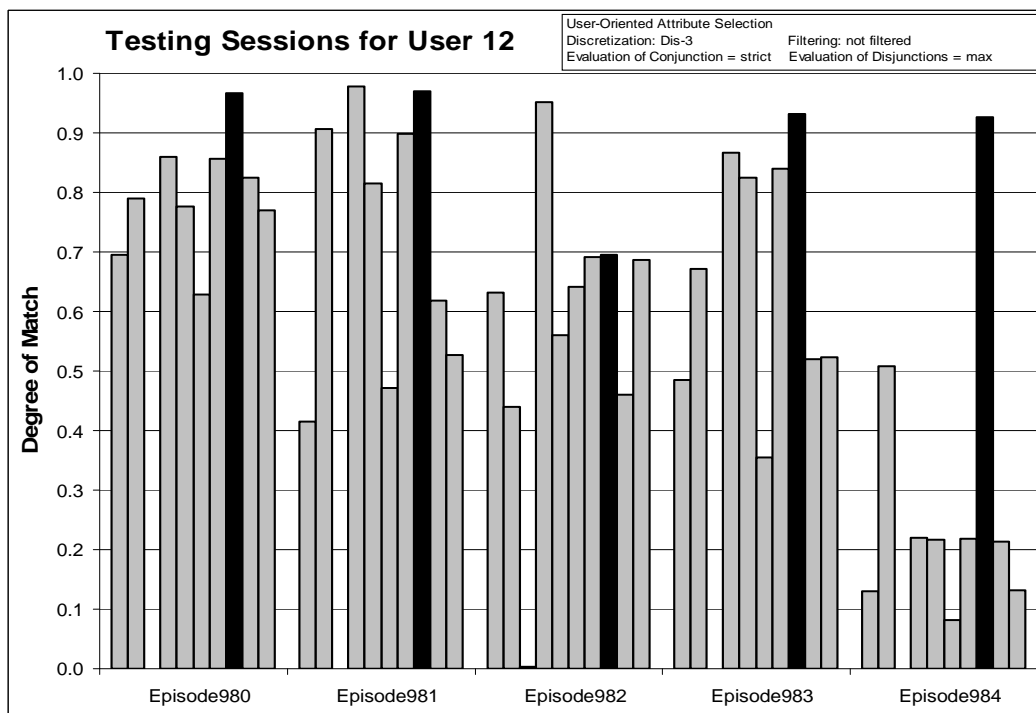


Figure 130: Degrees of match between 10 user models and 5 testing sessions from User 12.

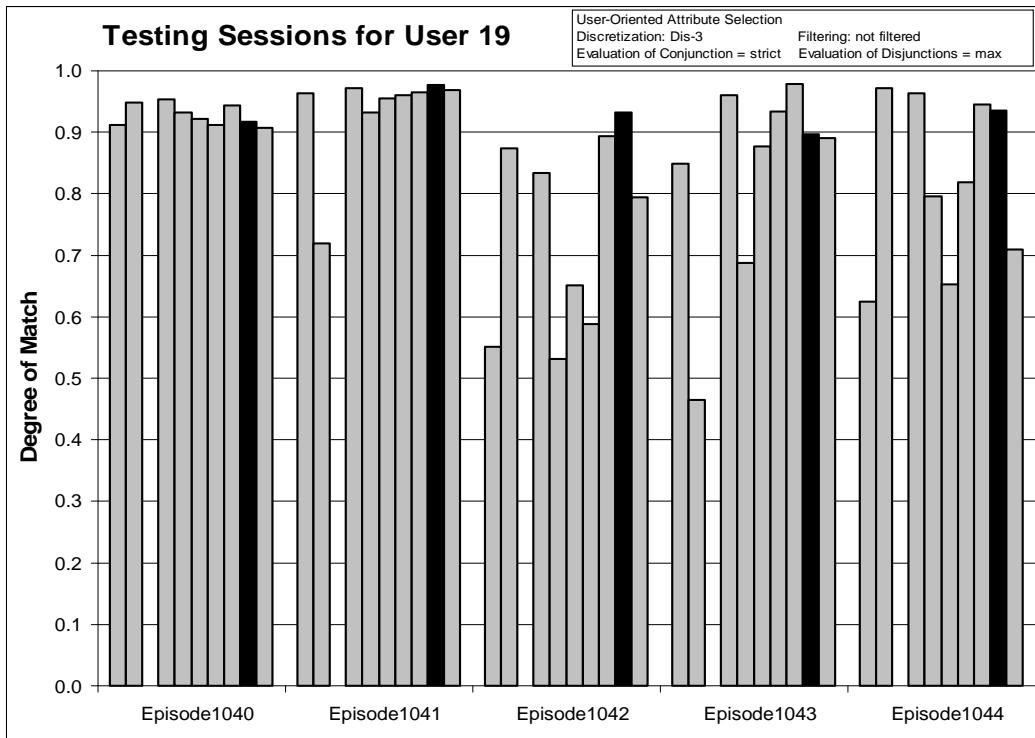


Figure 131: Degrees of match between 10 user models and 5 testing sessions from User 19.

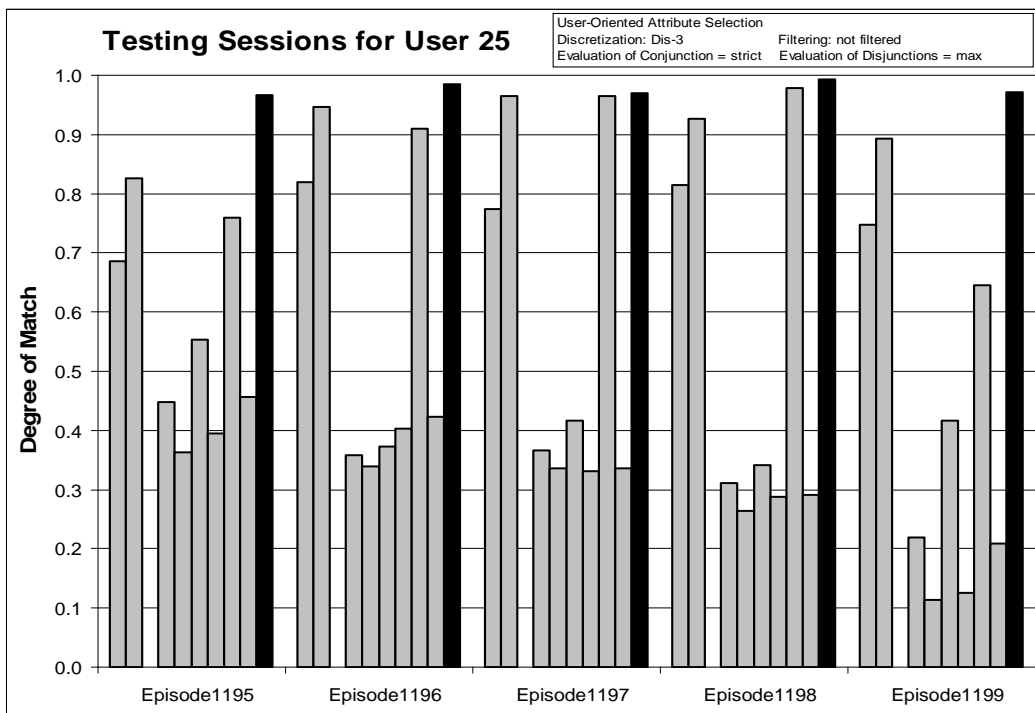


Figure 132: Degrees of match between 10 user models and 5 testing sessions from User 25.

User 4 (First Choice Correct: 80%)

Epi.391	0.714	0.812	0	1	1	0.433	0.998	1	1	0
Epi.392	0.0448	0.456	0	0.977	0.977	0.0202	0.996	0.977	0.977	0
Epi.393	0.75	0.708	0	1	1	0.508	0.995	1	1	0.769
Epi.394	0.828	0.215	0	0	0	0.0862	0	0	0	0.636
Epi.512	0.403	0.787	0	1	1	0.518	0.992	1	1	0

User 5 (First Choice Correct:100 %)

Epi.513	0	0.786	0	0.998	0.998	0	1	0.998	0.998	0
Epi.514	0.542	0.795	0	0.999	0.999	0.0323	0.994	0.999	0.999	0
Epi.515	0.211	0.754	0	1	1	0	0.993	1	1	0
Epi.542	0.583	0.777	0	0.998	0.998	0	0.998	0.998	0.998	0
Epi.543	0.239	0.676	0	1	1	0.118	1	1	1	0

User 7 (First Choice Correct: 20%)

Epi.734	0.941	0.12	0	0	0	0.863	0	0	0	0.915
Epi.735	0.576	0.169	0	0	0	0.431	0	0	0	0
Epi.736	0.925	0.241	0	0	0	1	0	0	0	0.925
Epi.737	0.857	0.219	0	0	0	0.852	0	0	0	0
Epi.738	0.653	0.24	0	0	0	0.567	0	0	0	0.962

User 8 (First Choice Correct: 20%)

Epi.741	0.444	0.121	0	0	0	0.132	0	0	0	0.5
Epi.742	0.545	0.185	0	0	0	0.111	0	0	0	0.3
Epi.743	0.182	0.765	0	1	1	0.04	0.997	1	1	0
Epi.744	0.23	0.792	0	0.998	0.998	0.0105	0.999	0.998	0.998	0.6
Epi.897	0.889	0.0839	0	0	0	0.216	0	0	0	0

User 12 (First Choice Correct: 100%)

Epi.980	0.652	0.278	0	1	1	0.0652	0.982	1	1	0.583
Epi.981	0.5	0.734	0	1	1	0.0833	0.996	1	1	0
Epi.982	0.974	0.303	0	1	1	0.841	0.99	1	1	0.583
Epi.983	0.643	0.288	0	1	1	0.0517	0.987	1	1	0.607
Epi.984	0.00749	0.241	0	1	1	0.0935	0.993	1	1	0

User 19 (First Choice Correct: 40%)

Epi.1040	1	0.2	0	1	1	0.5	0	1	1	0
Epi.1041	0.5	0	0	0	0	0	0	0	0	0
Epi.1042	0.167	0.0476	0	0	0	0.0133	0	0	0	0.167
Epi.1043	0.5	0.477	0	0.946	0.946	0	1	0.946	0.946	0
Epi.1044	1	0.627	0	1	1	0	0.973	1	1	0

User 25 (First Choice Correct: 40%)

Epi.1195	0.965	0.274	0	0	0	0.0676	0	0	0	0.952
Epi.1196	0.99	0.264	0	0	0	0.0236	0	0	0	0.986
Epi.1197	0.971	0.175	0	0	0	0.4	0	0	0	0.975
Epi.1198	0.979	0.0875	0	0	0	0.176	0	0	0	0.982
Epi.1199	0.965	0.262	0	0	0	0.0386	0	0	0	0.991

Table 62: Degrees of match for experiment 040627-1.

	User 1	User 2	User 3	User 4	User 5	User 7	User 8	User 12	User 19	User 25
User 1 (First Choice Correct: 40%)										
Epi.281	251/277	47/271	0/0	0/6	0/6	4/16	0/0	0/6	0/6	248/264
Epi.282	104/142	95/163	0/0	0/0	0/0	9/18	0/0	0/0	0/0	53/64
Epi.283	28/31	16/73	0/0	0/0	0/0	1/5	0/0	0/0	0/0	0/4
Epi.284	27/28	2/64	0/0	0/0	0/0	2/37	0/0	0/0	0/0	1/4
Epi.285	237/293	54/415	0/0	0/0	0/0	9/138	0/0	0/0	0/0	157/178

User 2 (First Choice Correct: 0%)

Epi.288	5/10	430/579	0/0	470/470	470/470	10/276	462/466	470/640	470/470	0/2
Epi.289	9/27	81/165	0/0	21/21	21/21	2/146	10/12	21/21	21/21	89/97
Epi.290	5/7	43/166	0/0	199/199	199/199	6/369	184/184	199/430	199/199	0/6
Epi.291	274/327	627/996	0/0	630/630	630/630	20/1388	650/655	630/923	630/630	15/22
Epi.333	7/10	140/217	0/0	3/3	3/3	5/153	0/0	3/109	3/3	144/153
Epi.288										

User 3 (First Choice Correct: 0%)

Epi.345	0/0	0/2	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
Epi.347	1/1	3/13	0/0	0/0	0/0	3/3	0/0	0/0	0/0	0/1
Epi.349	0/0	0/2	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0

User 4 (First Choice Correct: 80%)

Epi.391	10/14	599/738	0/0	630/630	630/630	13/30	616/617	630/630	630/630	0/0
Epi.392	3/67	233/511	0/0	260/266	260/266	7/346	243/244	260/266	260/266	0/1
Epi.393	36/48	467/660	0/0	668/668	668/668	32/63	655/658	668/668	668/668	20/26
Epi.394	24/29	20/93	0/0	0/0	0/0	5/58	0/0	0/0	0/0	7/11
Epi.512	25/62	395/502	0/0	395/395	395/395	29/56	392/395	395/395	395/395	0/1

User 5 (First Choice Correct: 100 %)

Epi.513	0/0	547/696	0/0	953/955	953/955	0/10	937/937	953/955	953/955	0/1
Epi.514	13/24	676/850	0/0	1012/1013	1012/1013	1/31	1012/1018	1012/1013	1012/1013	0/2
Epi.515	8/38	147/195	0/0	145/145	145/145	0/13	134/135	145/145	145/145	0/1
Epi.542	14/24	746/960	0/0	833/835	833/835	0/132	827/829	833/835	833/835	0/2
Epi.543	16/67	338/500	0/0	422/422	422/422	6/51	403/403	422/422	422/422	0/1

User 7 (First Choice Correct: 20%)

Epi.734	80/85	10/83	0/0	0/0	0/0	44/51	0/0	0/0	0/0	43/47
Epi.735	38/66	30/178	0/0	0/0	0/0	47/109	0/0	0/0	0/0	0/1
Epi.736	37/40	7/29	0/0	0/0	0/0	18/18	0/0	0/0	0/0	37/40
Epi.737	18/21	16/73	0/0	0/0	0/0	23/27	0/0	0/0	0/0	0/1
Epi.738	143/219	48/200	0/0	0/3	0/3	59/104	0/0	0/3	0/3	102/106

User 8 (First Choice Correct: 20%)

Epi.741	16/36	24/199	0/0	0/0	0/0	23/174	0/3	0/0	0/0	7/14
Epi.742	6/11	15/81	0/0	0/0	0/0	11/99	0/2	0/0	0/0	3/10
Epi.743	8/44	570/745	0/0	746/746	746/746	13/325	897/900	746/746	746/746	0/3
Epi.744	23/100	1368/1727	0/0	1616/1619	1616/1619	5/474	1726/1727	1616/1619	1616/1619	18/30
Epi.897	8/9	12/143	0/0	0/3	0/3	11/51	0/0	0/3	0/3	0/1

User 12 (First Choice Correct: 100%)

Epi.980	15/23	54/194	0/0	55/55	55/55	3/46	54/55	55/55	55/55	14/24
---------	-------	--------	-----	-------	-------	------	-------	-------	-------	-------

Epi.981	1/2	245/334	0/0	260/260	260/260	4/48	246/247	260/260	260/260	0/0
Epi.982	185/190	43/142	0/0	195/195	195/195	180/214	194/196	371/371	195/195	7/12
Epi.983	18/28	92/319	0/0	230/230	230/230	9/174	224/227	230/230	230/230	17/28
Epi.984	4/534	147/611	0/0	413/413	413/413	10/107	409/412	413/413	413/413	0/1

User 19 (First Choice Correct: 40%)

Epi.1040	5/5	3/15	0/0	3/3	3/3	1/2	0/0	3/3	3/3	0/2
Epi.1041	1/2	0/6	0/0	0/0	0/0	0/15	0/0	0/0	0/0	0/2
Epi.1042	2/12	3/63	0/0	0/0	0/0	1/75	0/0	0/0	0/0	1/6
Epi.1043	1/2	123/258	0/0	122/129	122/129	0/92	122/122	122/129	122/129	0/5
Epi.1044	3/3	101/161	0/0	119/119	119/119	0/112	109/112	119/119	119/119	0/3

User 25 (First Choice Correct: 40%)

Epi.1195	245/254	52/190	0/0	0/0	0/0	5/74	0/0	0/0	0/0	300/315
Epi.1196	501/506	69/261	0/0	0/0	0/0	3/127	0/0	0/0	0/0	568/576
Epi.1197	102/105	17/97	0/0	0/0	0/0	2/5	0/0	0/0	0/0	116/119
Epi.1198	512/523	14/160	0/0	0/0	0/0	6/34	0/0	0/0	0/0	531/541
Epi.1199	1937/2007	211/805	0/0	0/0	0/0	10/259	0/0	0/132	0/0	2247/2268

Table 63: Correct/total event matches in EPIC-P for experiment 040627-1.

8.4.13 Experiment 040720-1: Discretized and Filtered Data using Significance Measure 6

Training Dataset:

Discretization: Dis-3

Filtering: SIG-6, nmax, Rank-threshold 10, TR 5+5

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

AQ21 Learning Parameters:

maxstar = 5 maxrule = 10 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Simplicity-based descriptions

Testing Parameters:

Evaluation of Conjunction = coverage ratio

Evaluation of Disjunction = max

Acceptance Threshold = 10%

Accuracy Tolerance = 5%

Learning Results:

Total number of rules: 61

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	4	10	2	8	6	6	6	9	7	3

Table 64: Number of learned rules for 10 users.

Testing Results:

Correct: 62.50%

Precision: 56.40%

First Choice Correct: 56.25%

First Choice Precision: 100.00%

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
Correct	0%	100%	100%	20%	20%	100%	20%	80%	100%	100%
First Ch. Correct	0%	80%	100%	20%	20%	100%	0%	80%	80%	100%

Table 65: Summary of correct answers for 10 users.

	User 1	User 2	User 3	User 4	User 5	User 7	User 8	User 12	User 19	User 25
User 1 (Correct: 0% First Choice Correct: 0%)										
Epi.281	0.610	0.622	0.337	0.518	0.645	0.680	0.622	0.583	0.635	0.768
Epi.282	0.517	0.644	0.467	0.513	0.492	0.634	0.591	0.577	0.622	0.379
Epi.283	0.748	0.664	0.382	0.531	0.589	0.844	0.725	0.682	0.561	0.424
Epi.284	0.688	0.687	0.388	0.450	0.761	0.715	0.724	0.620	0.809	0.470
Epi.285	0.668	0.662	0.370	0.521	0.684	0.736	0.691	0.629	0.702	0.560
User 2 (Correct: 100% First Choice Correct: 80%)										
Epi.288	0.448	0.877	0.288	0.810	0.751	0.525	0.786	0.802	0.574	0.206
Epi.289	0.539	0.770	0.388	0.596	0.708	0.570	0.653	0.656	0.795	0.618
Epi.290	0.423	0.788	0.385	0.785	0.545	0.533	0.571	0.690	0.640	0.360
Epi.291	0.540	0.857	0.405	0.652	0.674	0.520	0.655	0.684	0.709	0.329
Epi.333	0.338	0.721	0.521	0.613	0.460	0.529	0.540	0.506	0.634	0.615
User 3 (Correct: 100% First Choice Correct: 100%)										
Epi.345	0.250	0.208	0.750	0.000	0.000	0.250	0.167	0.250	0.250	0.000
Epi.347	0.345	0.393	0.857	0.000	0.405	0.548	0.310	0.310	0.441	0.000
Epi.349	0.250	0.333	1.000	0.000	0.500	0.500	0.333	0.333	0.500	0.000
User 4 (Correct: 20% First Choice Correct: 20%)										
Epi.391	0.432	0.916	0.139	0.831	0.880	0.414	0.915	0.930	0.533	0.303
Epi.392	0.668	0.780	0.332	0.673	0.723	0.690	0.770	0.804	0.717	0.328
Epi.393	0.496	0.692	0.137	0.778	0.698	0.647	0.747	0.754	0.648	0.451
Epi.394	0.593	0.698	0.635	0.284	0.566	0.673	0.504	0.421	0.590	0.167
Epi.512	0.467	0.765	0.165	0.714	0.779	0.572	0.825	0.785	0.670	0.354
User 5 (Correct: 20% First Choice Correct: 20%)										
Epi.513	0.334	0.879	0.099	0.881	0.828	0.371	0.766	0.978	0.478	0.298
Epi.514	0.347	0.877	0.137	0.849	0.826	0.342	0.793	0.939	0.478	0.288
Epi.515	0.209	0.524	0.404	0.629	0.300	0.540	0.529	0.407	0.604	0.456
Epi.542	0.495	0.891	0.262	0.722	0.894	0.430	0.817	0.878	0.663	0.413
Epi.543	0.378	0.703	0.305	0.659	0.677	0.476	0.638	0.775	0.555	0.279
User 7 (Correct: 100% First Choice Correct: 100%)										
Epi.734	0.695	0.660	0.415	0.545	0.569	0.845	0.713	0.652	0.574	0.475

Epi.735	0.668	0.671	0.374	0.562	0.609	0.881	0.684	0.670	0.661	0.357
Epi.736	0.607	0.622	0.474	0.420	0.507	0.694	0.610	0.524	0.524	0.636
Epi.737	0.569	0.462	0.195	0.441	0.551	0.974	0.707	0.545	0.844	0.440
Epi.738	0.670	0.651	0.482	0.453	0.606	0.813	0.658	0.647	0.635	0.397

User 8 (Correct: 20% First Choice Correct: 0%)

Epi.741	0.746	0.763	0.474	0.486	0.768	0.753	0.742	0.745	0.781	0.350
Epi.742	0.581	0.667	0.477	0.525	0.646	0.654	0.633	0.693	0.629	0.400
Epi.743	0.436	0.811	0.261	0.751	0.773	0.454	0.815	0.818	0.562	0.260
Epi.744	0.414	0.860	0.227	0.785	0.797	0.418	0.807	0.857	0.574	0.302
Epi.897	0.765	0.776	0.497	0.494	0.676	0.774	0.774	0.779	0.859	0.535

User 12 (Correct: 80% First Choice Correct: 80%)

Epi.980	0.698	0.762	0.394	0.598	0.706	0.765	0.799	0.810	0.677	0.404
Epi.981	0.562	0.866	0.191	0.795	0.782	0.553	0.834	0.914	0.617	0.400
Epi.982	0.540	0.681	0.286	0.658	0.626	0.658	0.645	0.713	0.697	0.532
Epi.983	0.626	0.755	0.369	0.596	0.698	0.650	0.738	0.804	0.618	0.314
Epi.984	0.180	0.470	0.454	0.601	0.317	0.538	0.430	0.462	0.591	0.411

User 19 (Correct: 100% First Choice Correct: 80%)

Epi.1040	0.622	0.814	0.281	0.648	0.908	0.695	0.662	0.778	0.905	0.664
Epi.1041	0.690	0.750	0.449	0.588	0.722	0.701	0.666	0.787	0.925	0.760
Epi.1042	0.761	0.803	0.536	0.515	0.652	0.712	0.719	0.751	0.817	0.306
Epi.1043	0.597	0.674	0.323	0.661	0.676	0.673	0.680	0.672	0.804	0.703
Epi.1044	0.517	0.601	0.191	0.593	0.637	0.755	0.715	0.652	0.838	0.356

User 25 (Correct: 100% First Choice Correct: 100%)

Epi.1195	0.596	0.650	0.378	0.519	0.654	0.601	0.619	0.548	0.673	0.792
Epi.1196	0.591	0.643	0.416	0.575	0.624	0.572	0.592	0.631	0.596	0.776
Epi.1197	0.644	0.588	0.430	0.459	0.528	0.667	0.600	0.601	0.460	0.792
Epi.1198	0.591	0.532	0.424	0.512	0.513	0.602	0.476	0.550	0.565	0.864
Epi.1199	0.587	0.630	0.480	0.542	0.526	0.511	0.584	0.546	0.516	0.864

Table 66: Testing results for experiment 040720-1 (Significance measure 6).

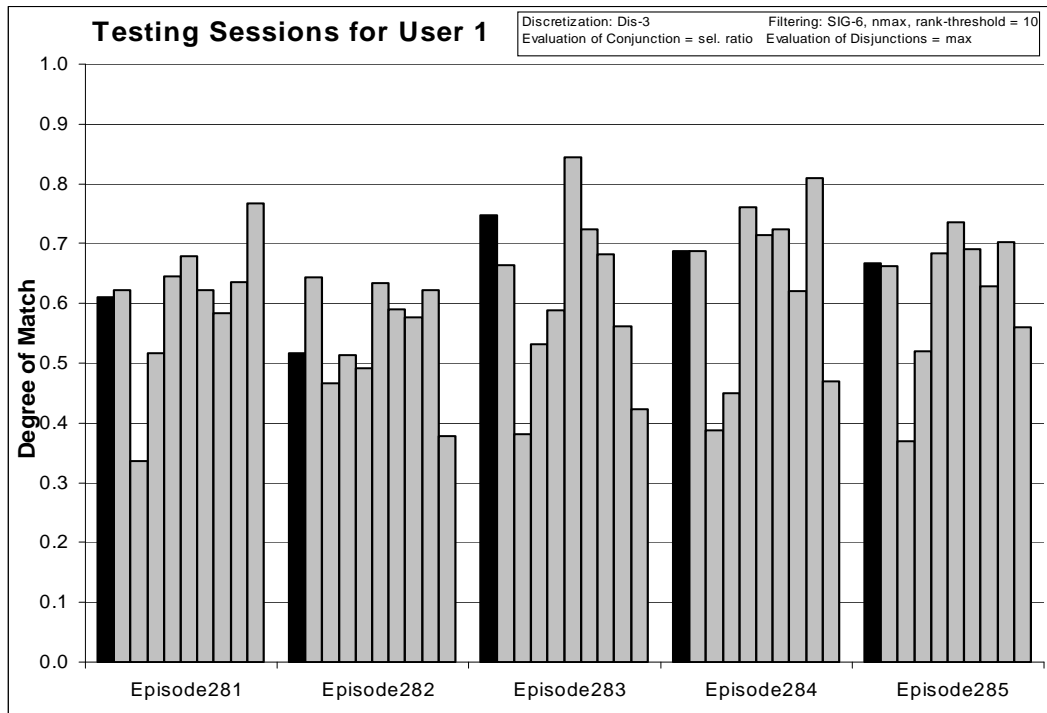


Figure 133: Degrees of match between 10 user models and 5 testing sessions from User 1.

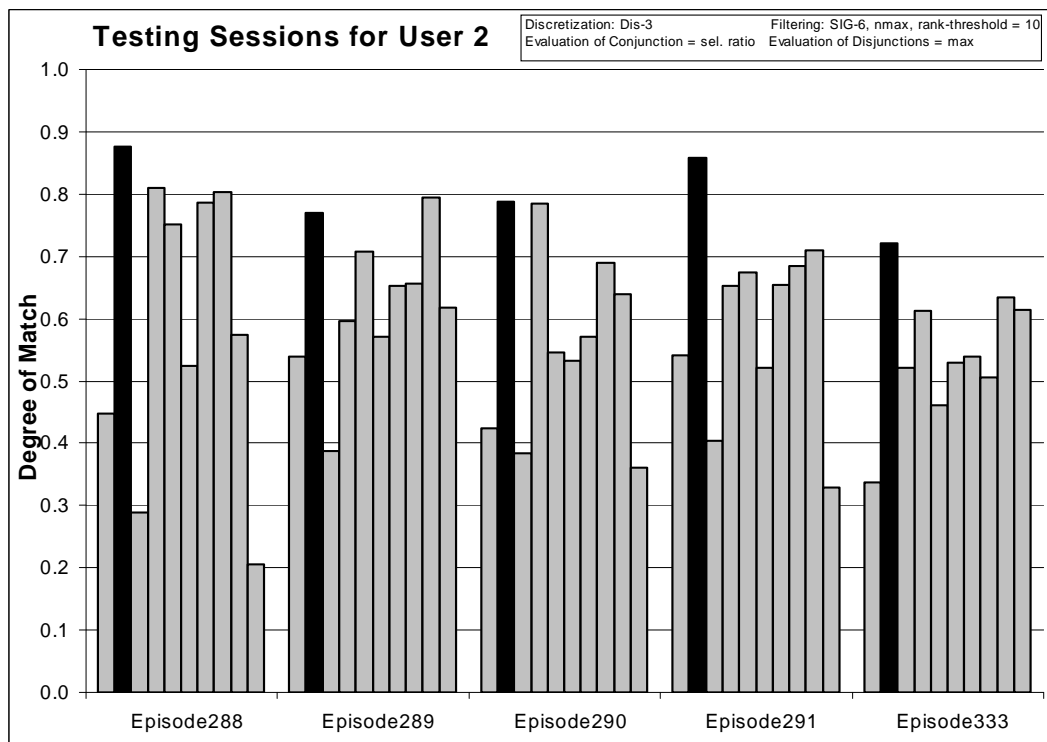


Figure 134: Degrees of match between 10 user models and 5 testing sessions from User 2

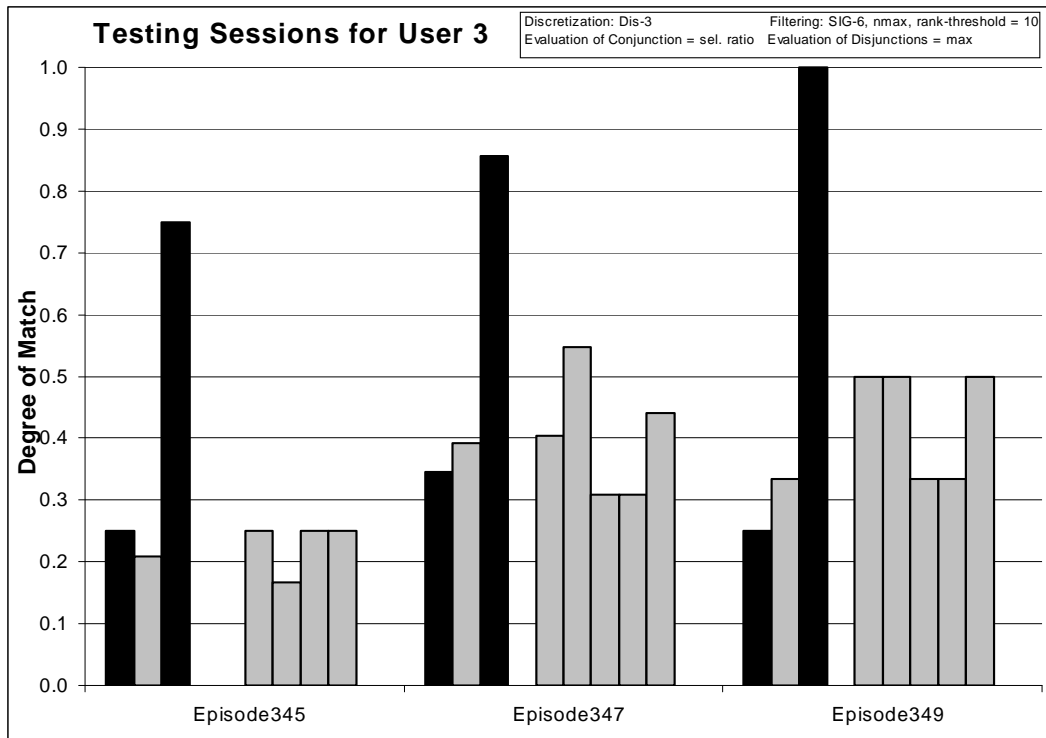


Figure 135: Degrees of match between 10 user models and 3 testing sessions from User 3.

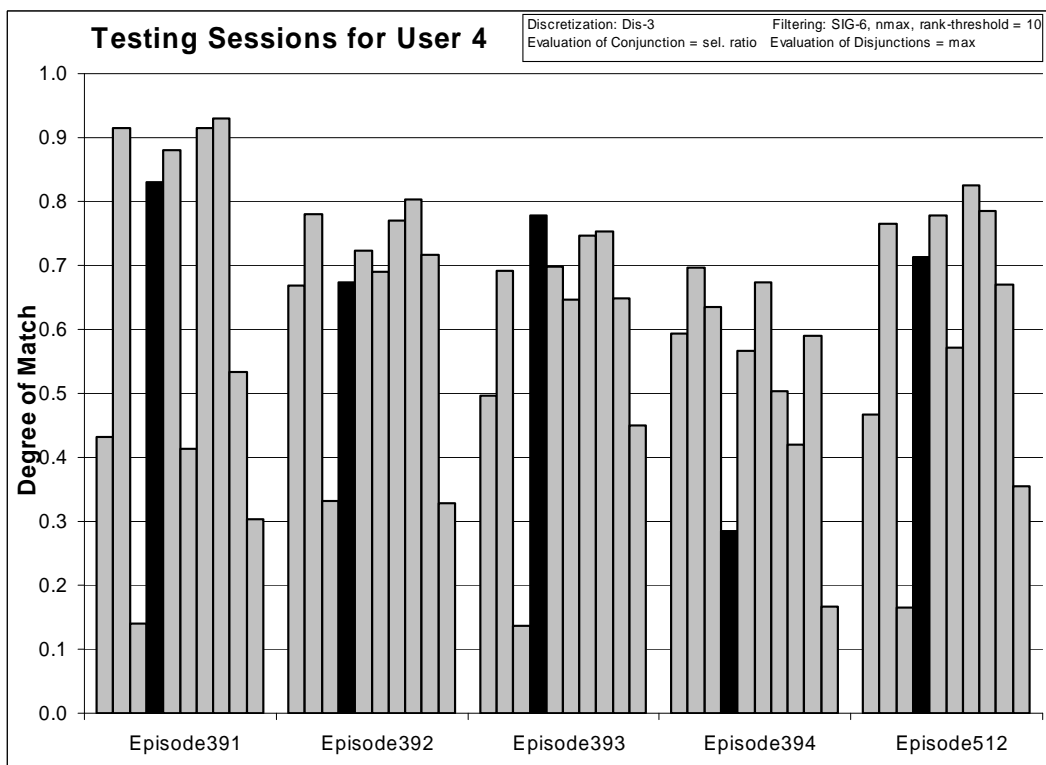


Figure 136: Degrees of match between 10 user models and 5 testing sessions from User 4

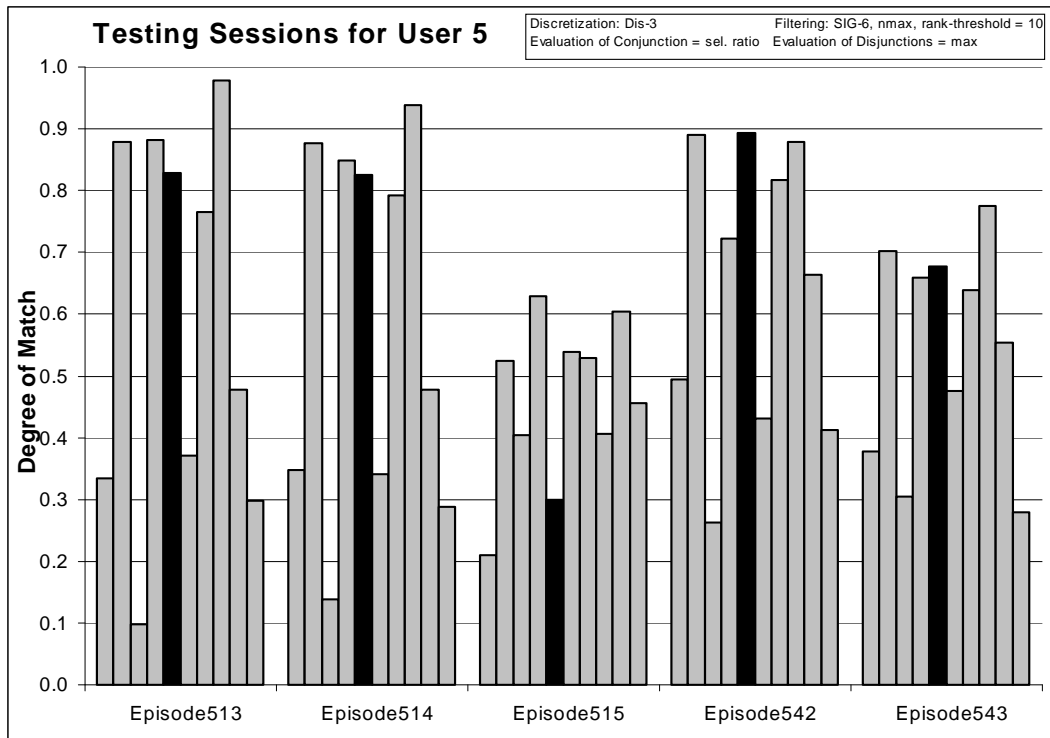


Figure 137: Degrees of match between 10 user models and 5 testing sessions from User 5.

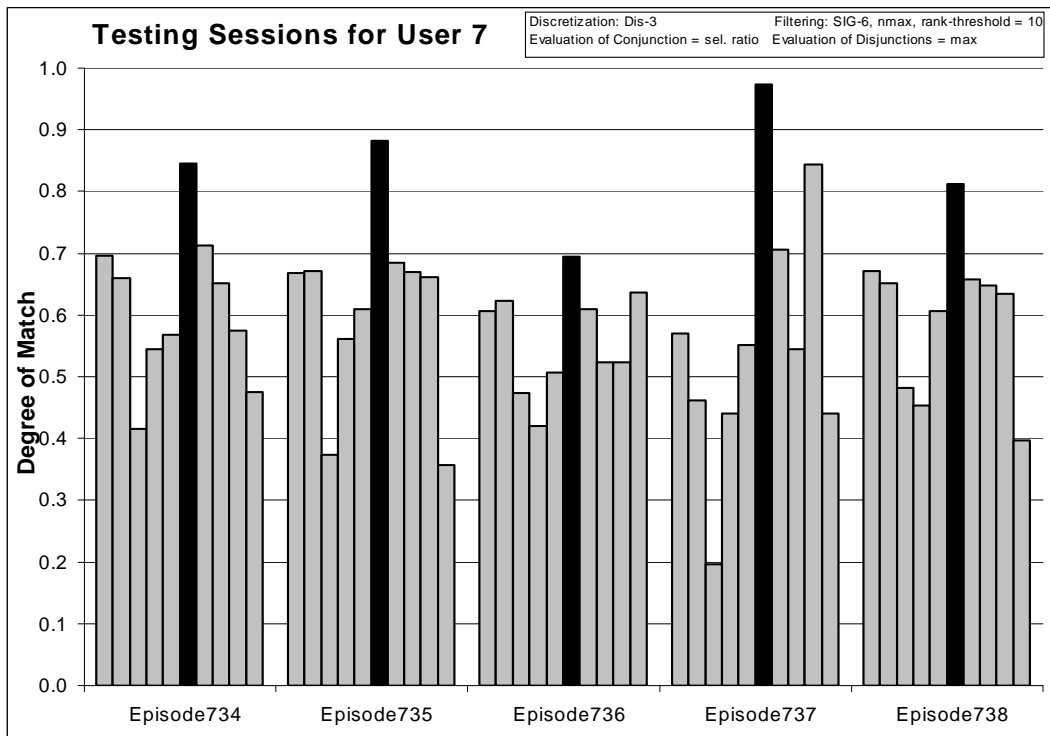


Figure 138: Degrees of match between 10 user models and 5 testing sessions from User 7.

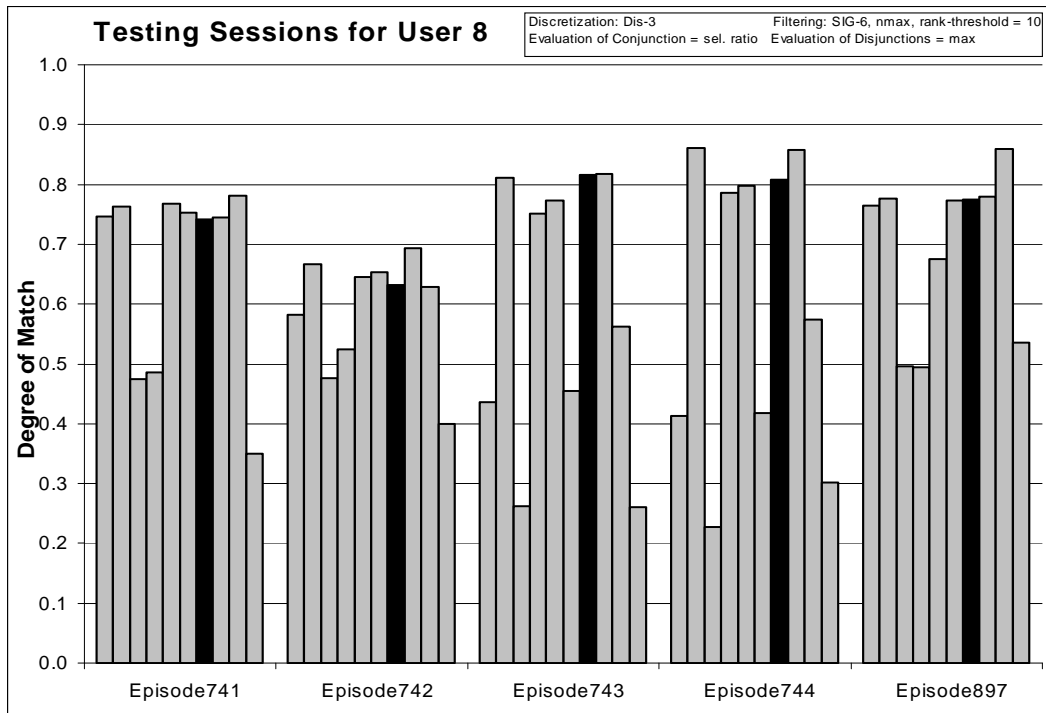


Figure 139: Degrees of match between 10 user models and 5 testing sessions from User 8.

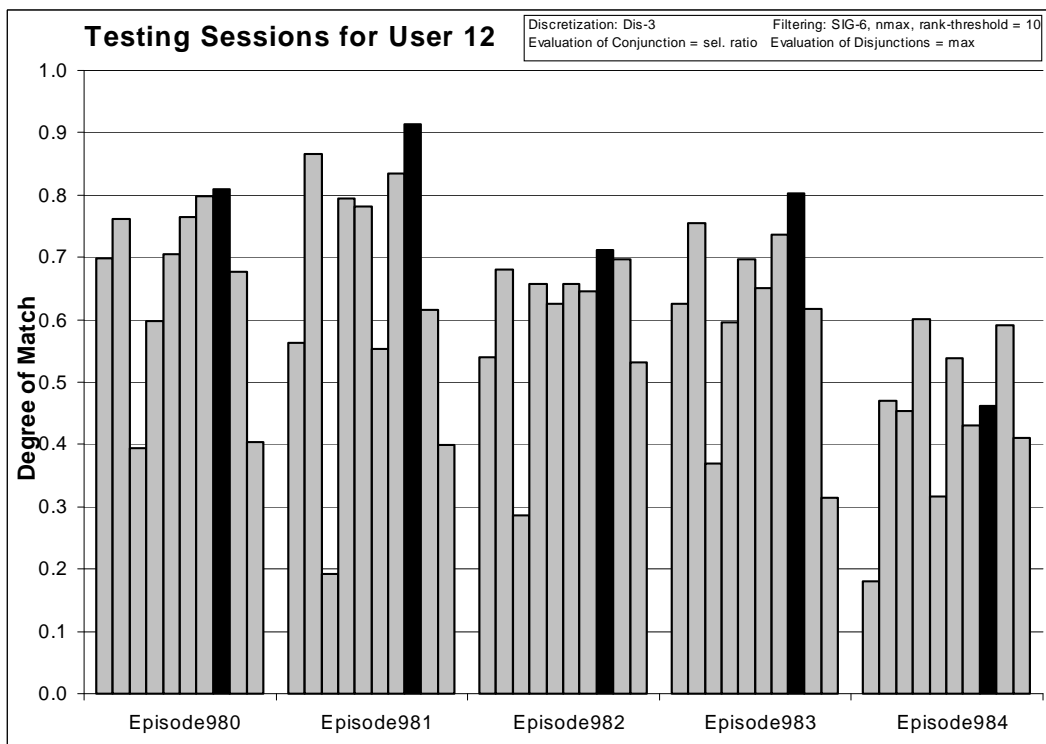


Figure 140: Degrees of match between 10 user models and 5 testing sessions from User 12.

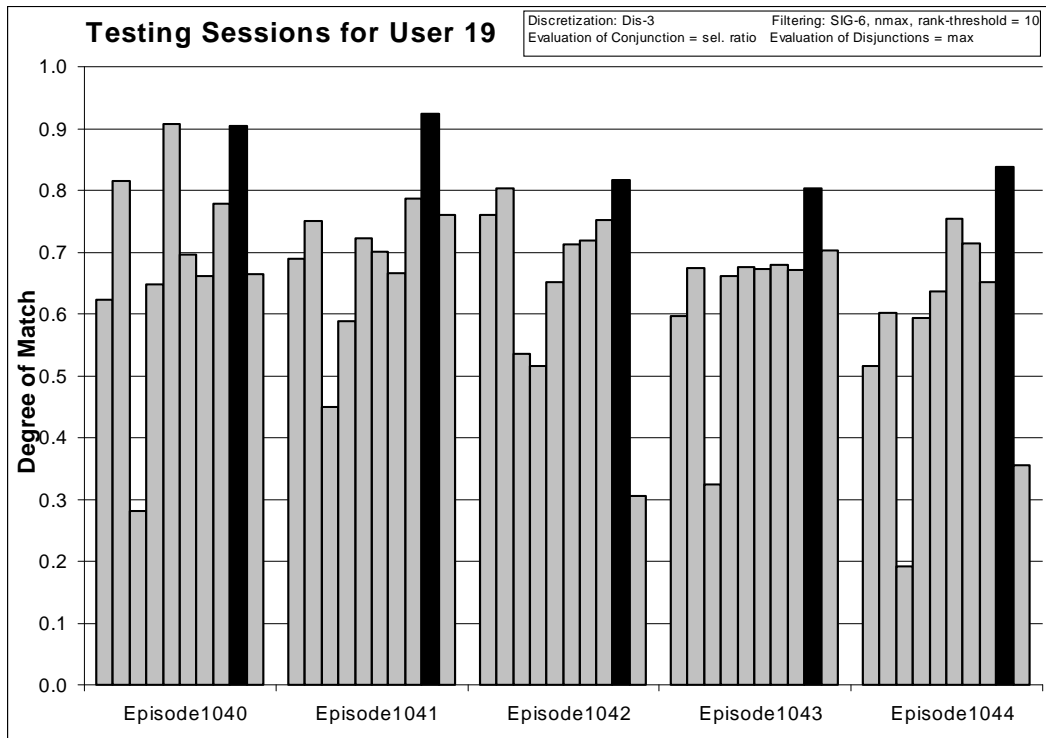


Figure 141: Degrees of match between 10 user models and 5 testing sessions from User 19.

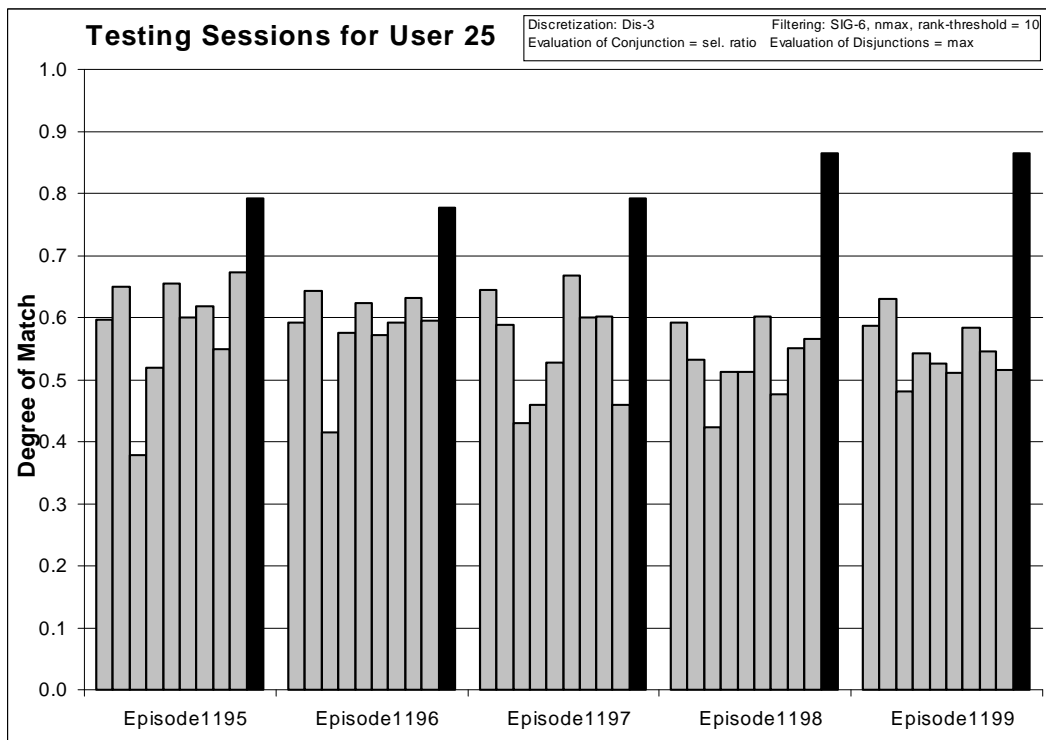


Figure 142: Degrees of match between 10 user models and 5 testing sessions from User 25.

8.4.14 Experiment 040720-2: Discretized and Filtered Data using Significance Measure 2

Training Dataset:

Discretization: Dis-3

Filtering: SIG-2, nmax, Rank-threshold 10, TR 5 + 5

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

AQ21 Learning Parameters:

maxstar = 10 maxrule = 1 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Simplicity-based descriptions

Testing Parameters:

Evaluation of Conjunction = coverage ratio

Evaluation of Disjunction = max

Acceptance Threshold = 10%

Accuracy Tolerance = 5%

Learning Results:

Total number of rules: 71

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	3	10	2	9	8	7	8	10	8	6

Table 67: Number of learned rules for 10 Users

Testing Results:

Correct: 58.33%

Precision: 74.91%

First Choice Correct: 54.17%

First Choice Precision: 100.00%

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
Correct	0%	40%	67%	40%	60%	100%	20%	80%	80%	100%
First Ch. Correct	0%	20%	67%	40%	40%	100%	20%	80%	80%	100%

Table 68: Summary of correct answers for 10 users.

	User 1	User 2	User 3	User 4	User 5	User 7	User 8	User 12	User 19	User 25
User 1 (Correct: 0% First Choice Correct: 0%)										
Epi.281	0.099	0.198	0.038	0.380	0.458	0.488	0.218	0.275	0.324	0.707
Epi.282	0.185	0.199	0.177	0.141	0.235	0.163	0.214	0.218	0.260	0.170
Epi.283	0.292	0.431	0.014	0.271	0.347	0.583	0.563	0.632	0.063	0.326

Epi.284	0.301	0.131	0.011	0.257	0.503	0.443	0.224	0.290	0.465	0.366
Epi.285	0.161	0.190	0.015	0.268	0.397	0.511	0.256	0.348	0.333	0.446

User 2 (Correct: 40% First Choice Correct: 20%)

Epi.288	0.085	0.632	0.010	0.647	0.616	0.076	0.626	0.621	0.095	0.094
Epi.289	0.189	0.410	0.146	0.266	0.451	0.342	0.237	0.257	0.630	0.434
Epi.290	0.056	0.260	0.010	0.583	0.229	0.032	0.170	0.354	0.199	0.046
Epi.291	0.157	0.559	0.009	0.289	0.348	0.051	0.293	0.314	0.420	0.134
Epi.333	0.019	0.015	0.344	0.257	0.024	0.061	0.010	0.058	0.235	0.015

User 3 (Correct: 67% First Choice Correct: 67%)

Epi.345	0.000	0.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Epi.347	0.000	0.000	0.143	0.000	0.000	0.071	0.000	0.000	0.000	0.000
Epi.349	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

User 4 (Correct: 40% First Choice Correct: 40%)

Epi.391	0.094	0.819	0.000	0.705	0.860	0.174	0.859	0.903	0.071	0.130
Epi.392	0.186	0.552	0.002	0.357	0.644	0.378	0.494	0.646	0.330	0.265
Epi.393	0.078	0.391	0.010	0.755	0.481	0.370	0.392	0.547	0.283	0.336
Epi.394	0.079	0.238	0.032	0.000	0.040	0.159	0.357	0.135	0.183	0.079
Epi.512	0.089	0.539	0.012	0.679	0.603	0.377	0.554	0.603	0.322	0.317

User 5 (Correct: 60% First Choice Correct: 40%)

Epi.513	0.000	0.637	0.002	0.773	0.684	0.023	0.599	0.939	0.035	0.000
Epi.514	0.045	0.671	0.006	0.736	0.679	0.014	0.654	0.878	0.039	0.026
Epi.515	0.075	0.213	0.000	0.213	0.213	0.121	0.182	0.221	0.196	0.160
Epi.542	0.079	0.688	0.002	0.477	0.758	0.064	0.541	0.695	0.336	0.200
Epi.543	0.087	0.523	0.002	0.398	0.605	0.119	0.507	0.544	0.164	0.111

User 7 (Correct: 100% First Choice Correct: 100%)

Epi.734	0.081	0.278	0.014	0.202	0.354	0.650	0.386	0.516	0.076	0.466
Epi.735	0.069	0.376	0.009	0.340	0.409	0.729	0.474	0.508	0.257	0.387
Epi.736	0.000	0.000	0.039	0.000	0.052	0.403	0.026	0.208	0.000	0.325
Epi.737	0.044	0.064	0.008	0.626	0.096	0.912	0.167	0.267	0.630	0.657
Epi.738	0.085	0.115	0.019	0.100	0.166	0.625	0.192	0.335	0.168	0.328

User 8 (Correct: 20% First Choice Correct: 20%)

Epi.741	0.312	0.402	0.024	0.131	0.564	0.316	0.290	0.508	0.409	0.272
Epi.742	0.167	0.265	0.033	0.130	0.265	0.316	0.274	0.419	0.088	0.214
Epi.743	0.036	0.451	0.036	0.531	0.485	0.038	0.616	0.605	0.118	0.021
Epi.744	0.058	0.670	0.034	0.574	0.618	0.084	0.652	0.720	0.148	0.077
Epi.897	0.343	0.425	0.007	0.059	0.460	0.270	0.387	0.563	0.615	0.558

User 12 (Correct: 80% First Choice Correct: 80%)

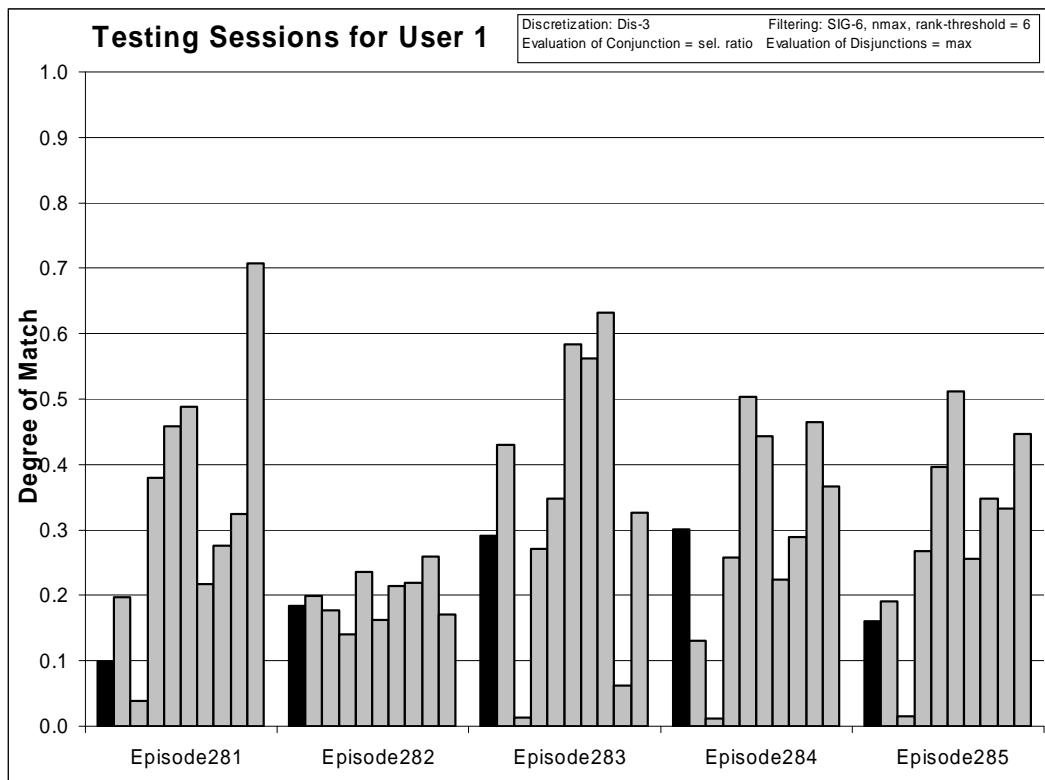
Epi.980	0.258	0.563	0.011	0.253	0.643	0.385	0.606	0.772	0.267	0.382
Epi.981	0.113	0.816	0.000	0.628	0.850	0.331	0.735	0.887	0.179	0.328
Epi.982	0.130	0.241	0.005	0.445	0.418	0.301	0.287	0.470	0.330	0.351
Epi.983	0.280	0.376	0.009	0.314	0.461	0.199	0.428	0.638	0.153	0.222
Epi.984	0.017	0.103	0.000	0.147	0.154	0.097	0.084	0.168	0.177	0.087

User 19 (Correct: 80% First Choice Correct: 80%)

Epi.1040	0.078	0.552	0.010	0.464	0.927	0.474	0.135	0.557	0.797	0.495
Epi.1041	0.046	0.361	0.006	0.117	0.369	0.117	0.108	0.361	0.815	0.699
Epi.1042	0.393	0.484	0.032	0.000	0.504	0.119	0.325	0.532	0.552	0.314
Epi.1043	0.019	0.181	0.005	0.361	0.249	0.149	0.119	0.203	0.581	0.541
Epi.1044	0.118	0.332	0.005	0.665	0.233	0.525	0.249	0.254	0.668	0.550

User 25 (Correct: 100% First Choice Correct: 100%)

Epi.1195	0.050	0.094	0.073	0.243	0.343	0.304	0.051	0.145	0.418	0.651
Epi.1196	0.082	0.140	0.064	0.196	0.256	0.277	0.142	0.196	0.192	0.717
Epi.1197	0.116	0.251	0.075	0.156	0.286	0.246	0.276	0.317	0.091	0.764
Epi.1198	0.033	0.092	0.016	0.173	0.222	0.258	0.096	0.158	0.166	0.837
Epi.1199	0.050	0.075	0.067	0.084	0.094	0.091	0.071	0.093	0.120	0.732

Table 69: Testing results for experiment 040720-2 (Significance measure 2)*Figure 143: Degrees of match between 10 user models and 5 testing sessions from User 1.*

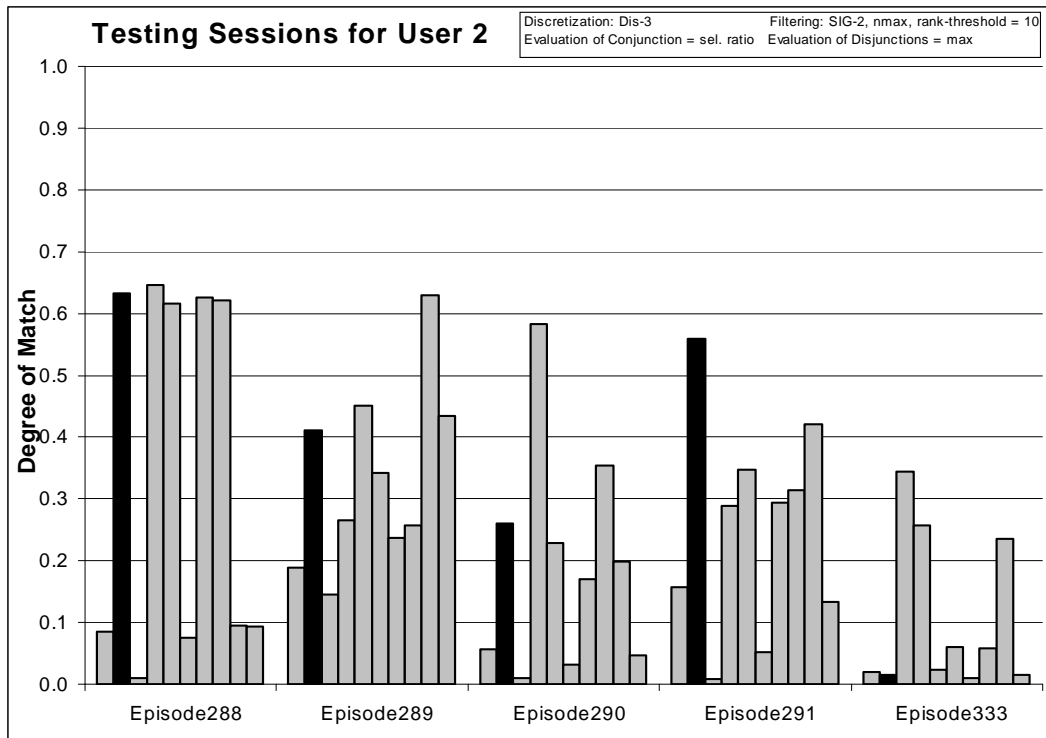


Figure 144: Degrees of match between 10 user models and 5 testing sessions from User 2.

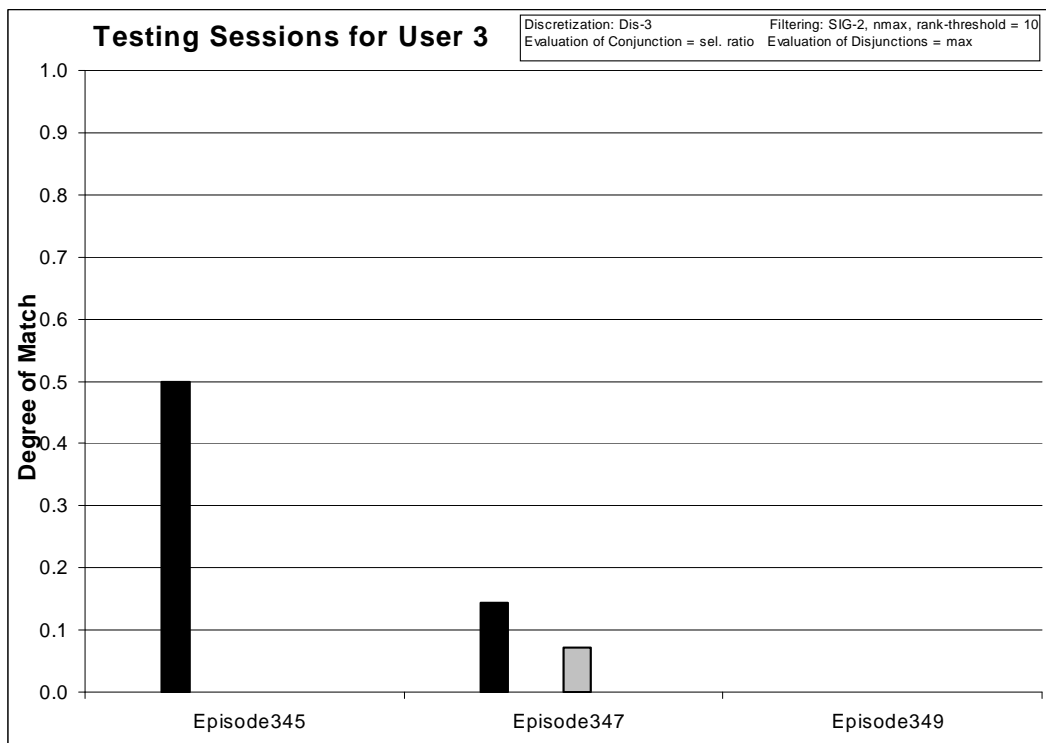


Figure 145: Degrees of match between 10 user models and 3 testing sessions from User 3.

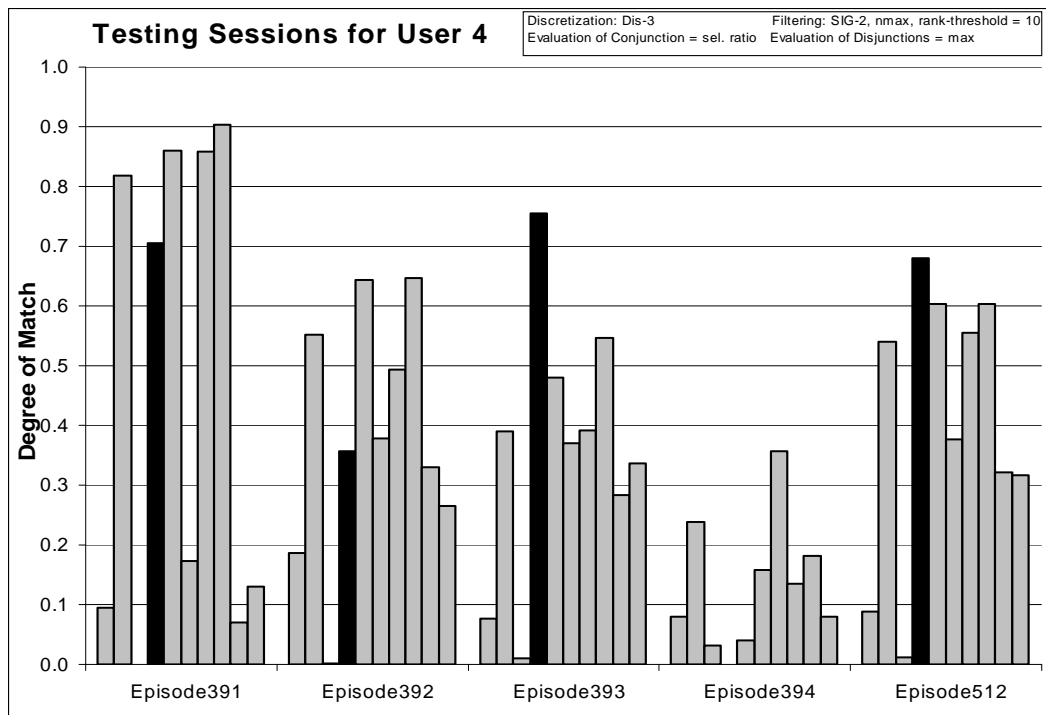


Figure 146: Degrees of match between 10 user models and 5 testing sessions from User 4.

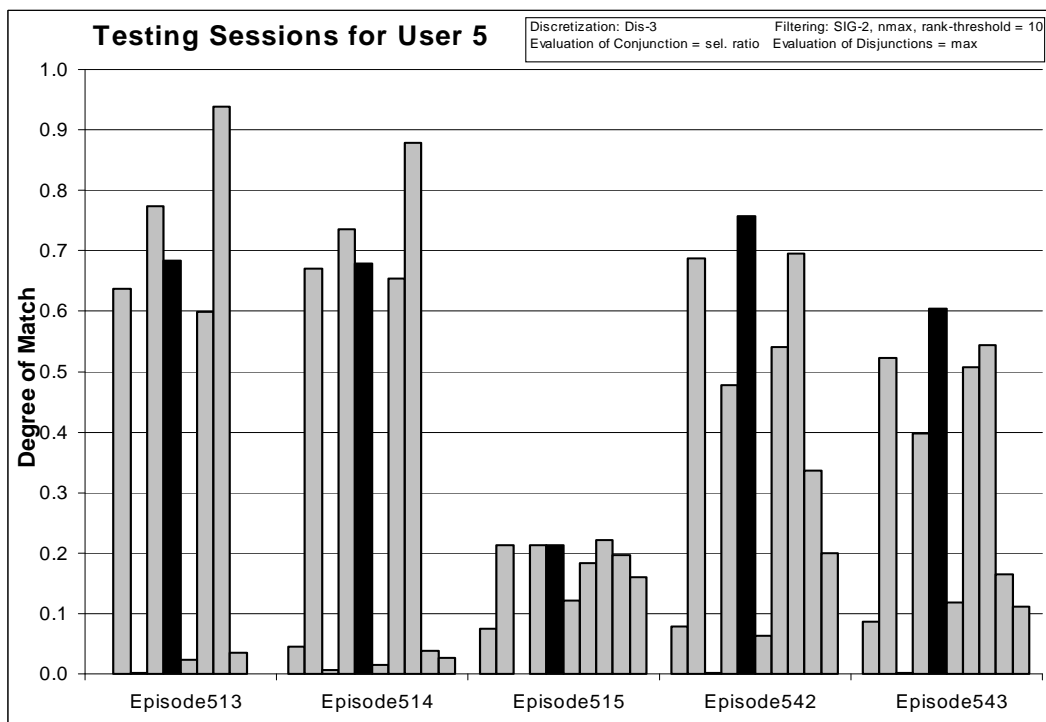


Figure 147: Degrees of match between 10 user models and 5 testing sessions from User 5.

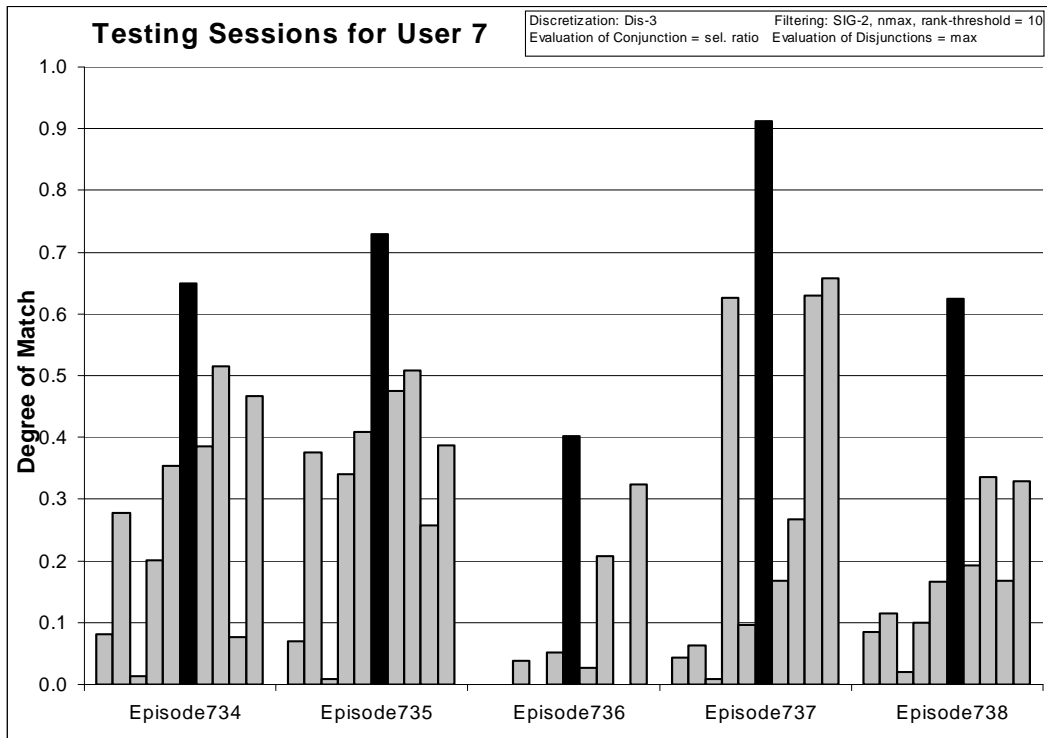


Figure 148: Degrees of match between 10 user models and 5 testing sessions from User 7.

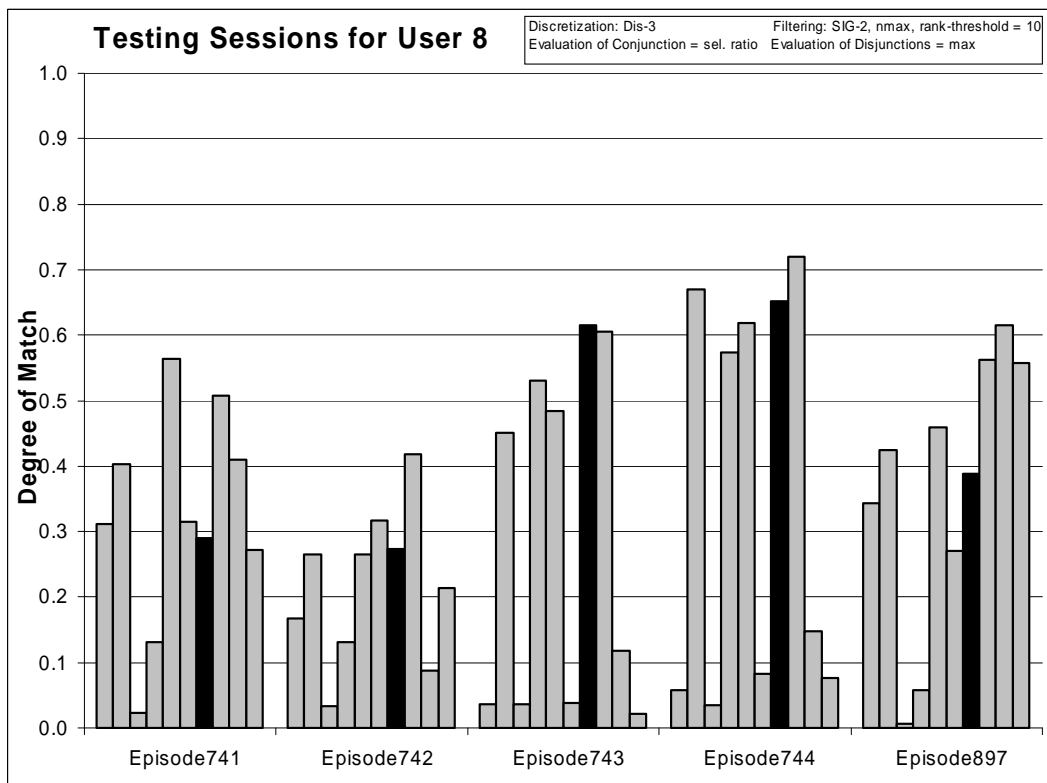


Figure 149: Degrees of match between 10 user models and 5 testing sessions from User 8.

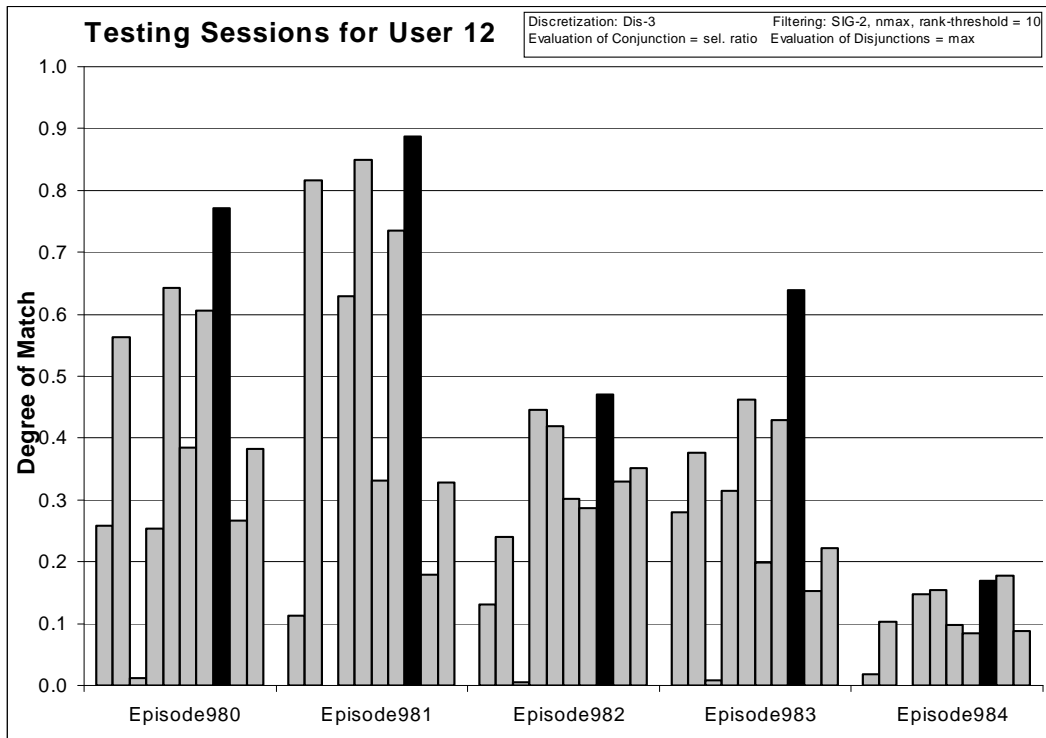


Figure 150: Degrees of match between 10 user models and 5 testing sessions from User 12.

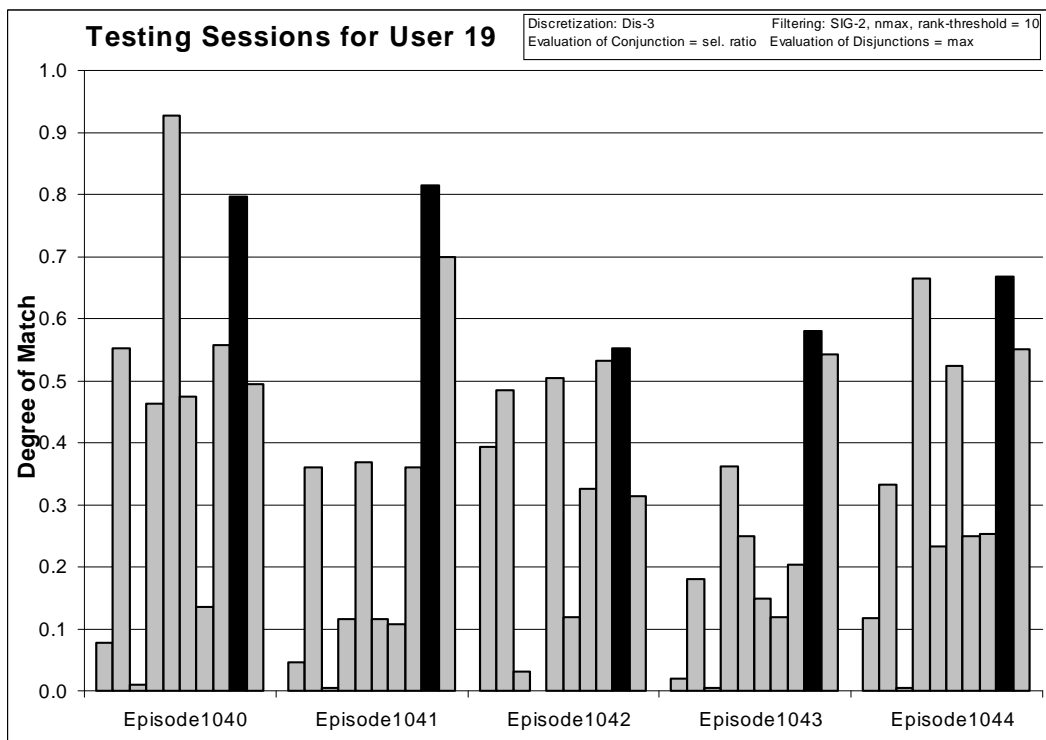


Figure 151: Degrees of match between 10 user models and 5 testing sessions from User 19.

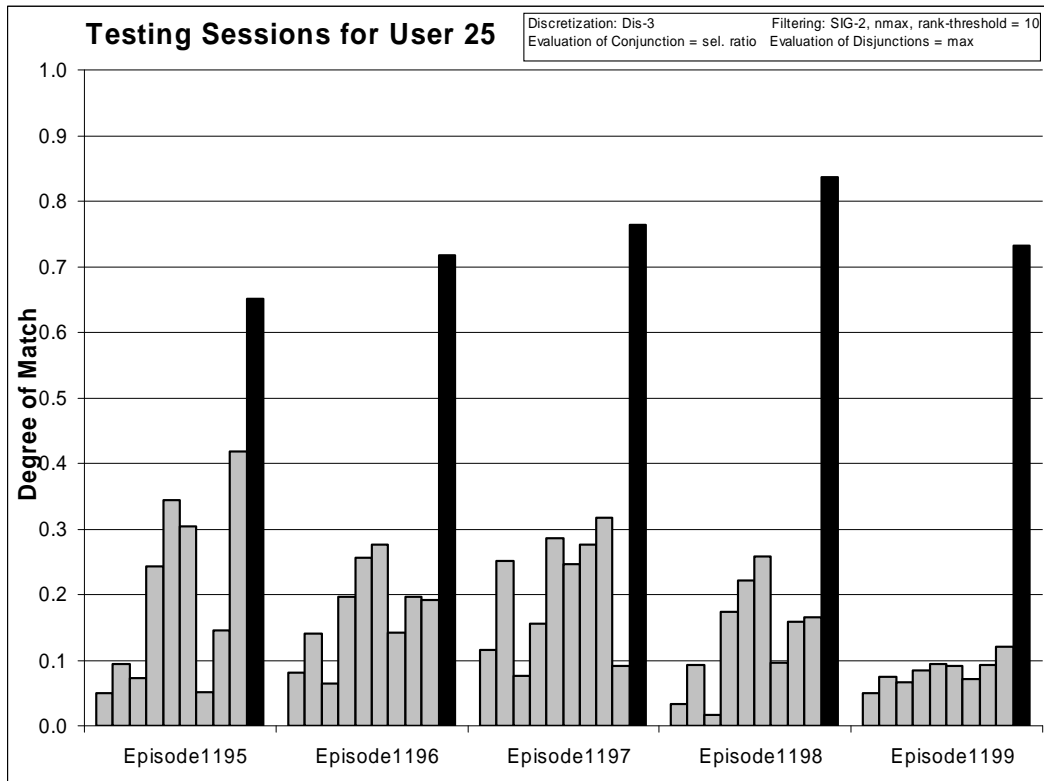


Figure 152: Degrees of match between 10 user models and 5 testing sessions from User 25.

8.4.15 $n \times 6$ -Grams for $n = 1, 2, 3, 4$, and 5

Training Dataset:

Discretization: Dis-3

Filtering: not filtered

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

AQ21 Learning Parameters:

maxstar = 1 maxrule = 1 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Characteristic descriptions

Testing Parameters:

Evaluation of Conjunction = strict

Evaluation of Disjunction = max

Acceptance Threshold = 10%

Accuracy Tolerance = 5%

n	Total # of rules	Correct	Precision	First Choice Correct	First Choice Precision
1	652	68.75%	33.71%	60.42%	82.46%
2	2637	72.92%	41.18%	60.42%	95.56%
3	4187	72.92%	44.44%	54.17%	91.45%
4	5044	70.83%	53.15%	58.33%	97.73%
5	6658	66.67%	57.26%	58.33%	100.00%

Table 70: Summary of learning and testing for n -gram sizes 1, 2, 3, 4, and 5.

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	55	103	15	81	50	57	56	71	80	84
Correct	60%	80%	33%	80%	40%	60%	20%	100%	100%	100%
First Ch. Correct	60%	60%	33%	80%	0%	60%	0%	100%	100%	100%

Table 71: Summary of learning and testing for $n=1$.

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	182	463	28	420	207	212	230	290	303	302
Correct	80%	80%	67%	80%	40%	60%	20%	100%	100%	100%
First Ch. Correct	80%	60%	67%	80%	0%	60%	0%	100%	60%	100%

Table 72: Summary of learning and testing for $n=2$.

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	263	745	43	706	312	348	363	455	465	487
Correct	80%	100%	33%	80%	40%	60%	20%	100%	100%	100%
First Ch. Correct	60%	60%	33%	60%	0%	60%	0%	100%	60%	100%

Table 73: Summary of learning and testing for $n=3$.

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	342	1007	40	960	398b	444	459	585	599	608
Correct	80%	80%	33%	80%	40%	60%	20%	100%	100%	100%
First Ch. Correct	60%	80%	33%	60%	0%	60%	20%	100%	60%	100%

Table 74: Summary of learning and testing for $n=4$.

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	364	1194	35	1204	487	552	580	740	718	784
Correct	60%	80%	33%	80%	40%	60%	20%	100%	80%	100%
First Ch. Correct	60%	80%	33%	60%	0%	60%	20%	100%	60%	100%

Table 75: Summary of learning and testing for $n=5$.

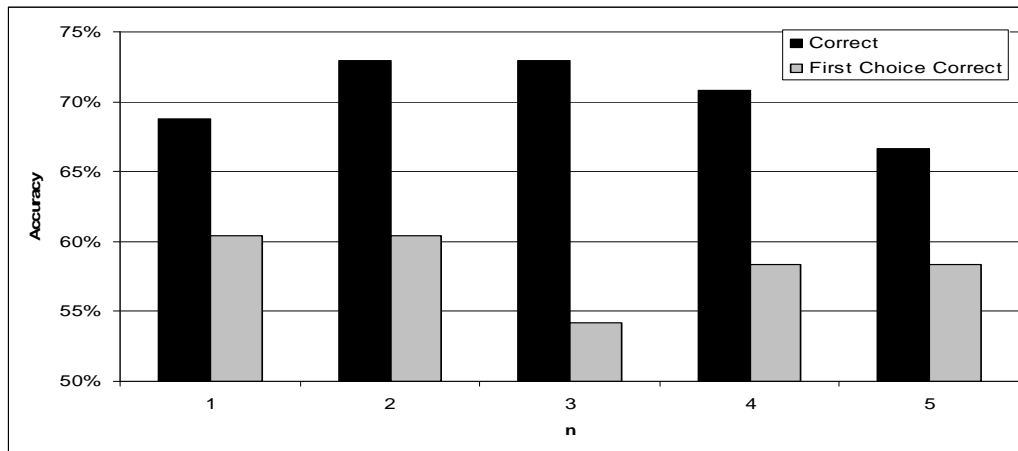


Figure 153: Number of correct and first choice correct answers for $n = 1, 2, 3, 4$, and 5 .

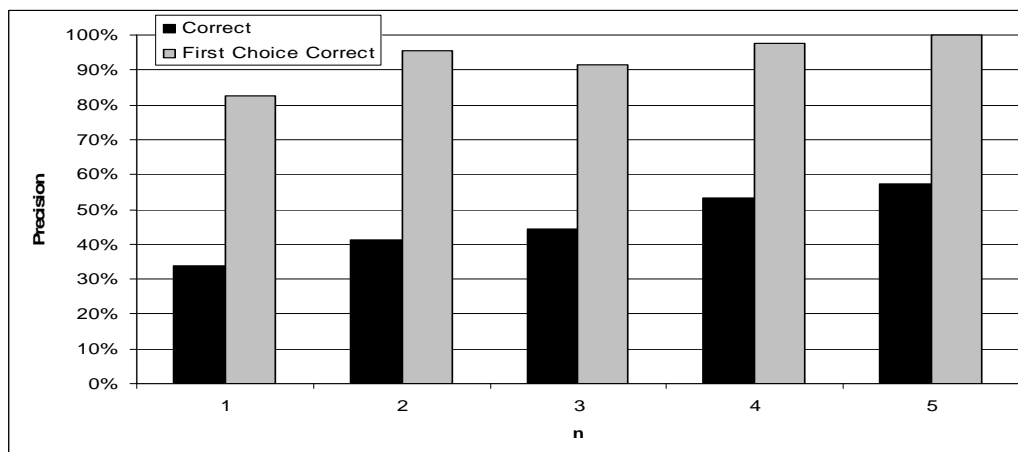


Figure 154: Precision and first choice precision for $n = 1, 2, 3, 4$, and 5 .

The charts and table show how accuracy and precision changes for different values of n in $n \times k$ -grams.

It should be noted that the number of correct answers is the highest for n equal to 3, but the number of first choice correct answers is the best for n equal to 1, which means that no past information is being used.

8.5 AQ21 Experiments on Data from 10 Users: Window Records Only, 10+5 Sessions

In this set of experiments we investigated if it is enough to use only window records from the source data to learn users' models and classify new sessions. The window records indicate actions that users consciously perform and ignore all other processes whose appearance is to high degree controlled by the operating system. Preliminary results shown in this section show that this is a very promising approach.

8.5.1 Experiment 040727-1: Unfiltered Data, Characteristic Descriptions

Source Data: window records

Training Dataset:

Discretization: Dis-3

Filtering: not filtered

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

AQ21 Learning Parameters:

maxstar = 1 maxrule = 1 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Characteristic descriptions

Testing Parameters:

Evaluation of Conjunction = strict

Evaluation of Disjunction = max

Acceptance Threshold = 10%

Accuracy Tolerance = 5%

Learning Results:

Total number of rules: 3172

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	152	569	16	645	228	242	279	424	254	363

Table 76: Number of learned rules for 10 users .

Testing Results:

Correct: 63.83%

Precision: 89.32%

First Choice Correct: 63.83%

First Choice Precision: 91.29%

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
Correct First Ch.	40%	80%	50%	80%	0%	100%	20%	100%	60%	100%
Correct	40%	80%	50%	80%	0%	100%	20%	100%	60%	100%

Table 77: Summary of correct answers for 10 users.

	User 1	User 2	User 3	User 4	User 5	User 7	User 8	User 12	User 19	User 25
User 1 (Correct: 40% First Choice Correct: 40%)										
Epi.281	0.162	0.085	0.000	0.100	0.000	0.092	0.023	0.046	0.000	0.339
Epi.282	0.120	0.013	0.013	0.027	0.013	0.013	0.000	0.040	0.013	0.067
Epi.283	0.286	0.143	0.000	0.179	0.000	0.143	0.036	0.071	0.000	0.000
Epi.284	0.032	0.097	0.000	0.097	0.032	0.032	0.000	0.032	0.000	0.065
Epi.285	0.100	0.084	0.000	0.111	0.000	0.090	0.058	0.100	0.037	0.221
User 2 (Correct: 80% First Choice Correct: 80%)										
Epi.288	0.019	0.318	0.000	0.271	0.243	0.009	0.187	0.252	0.178	0.075
Epi.289	0.000	0.245	0.000	0.038	0.076	0.000	0.019	0.019	0.038	0.151
Epi.290	0.000	0.170	0.000	0.051	0.068	0.000	0.034	0.034	0.034	0.000
Epi.291	0.108	0.470	0.000	0.108	0.067	0.003	0.019	0.067	0.073	0.089
Epi.333	0.000	0.040	0.000	0.020	0.020	0.000	0.000	0.020	0.020	0.260
User 3 (Correct: 50% First Choice Correct: 50%)										
Epi.345	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Epi.347	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
User 4 (Correct: 80% First Choice Correct: 80%)										
Epi.391	0.079	0.333	0.000	0.603	0.254	0.127	0.270	0.508	0.016	0.016
Epi.392	0.030	0.178	0.000	0.393	0.030	0.067	0.133	0.319	0.037	0.030
Epi.393	0.035	0.186	0.000	0.381	0.106	0.106	0.115	0.168	0.080	0.106
Epi.394	0.108	0.054	0.000	0.000	0.000	0.081	0.027	0.000	0.027	0.135
Epi.512	0.061	0.076	0.000	0.349	0.076	0.030	0.121	0.242	0.076	0.015
User 5 (Correct: 0% First Choice Correct: 0%)										
Epi.513	0.000	0.479	0.000	0.507	0.465	0.000	0.338	0.465	0.127	0.000
Epi.514	0.000	0.285	0.000	0.241	0.216	0.000	0.164	0.276	0.086	0.000
Epi.515	0.023	0.023	0.000	0.068	0.091	0.000	0.000	0.023	0.000	0.046
Epi.542	0.009	0.288	0.000	0.339	0.246	0.000	0.229	0.263	0.068	0.009
Epi.543	0.011	0.463	0.000	0.400	0.421	0.000	0.305	0.411	0.116	0.011
User 7 (Correct: 100% First Choice Correct: 100%)										
Epi.734	0.065	0.065	0.000	0.194	0.000	0.548	0.097	0.194	0.000	0.129
Epi.735	0.085	0.141	0.000	0.127	0.000	0.479	0.070	0.197	0.014	0.056
Epi.736	0.000	0.000	0.000	0.077	0.000	0.308	0.000	0.077	0.000	0.308
Epi.737	0.000	0.091	0.000	0.182	0.000	0.500	0.136	0.000	0.000	0.046
Epi.738	0.000	0.046	0.000	0.182	0.015	0.273	0.015	0.015	0.030	0.106
User 8 (Correct: 20% First Choice Correct: 20%)										
Epi.741	0.031	0.073	0.000	0.042	0.010	0.010	0.115	0.073	0.010	0.052
Epi.742	0.000	0.000	0.000	0.073	0.024	0.146	0.098	0.098	0.049	0.049
Epi.743	0.021	0.076	0.000	0.097	0.035	0.021	0.146	0.174	0.118	0.042

Epi.744	0.013	0.149	0.000	0.236	0.100	0.026	0.122	0.223	0.100	0.048
Epi.897	0.053	0.013	0.000	0.066	0.026	0.066	0.026	0.092	0.040	0.092

User 12 (Correct: 100% First Choice Correct: 100%)

Epi.980	0.045	0.149	0.000	0.254	0.060	0.090	0.045	0.284	0.030	0.060
Epi.981	0.000	0.204	0.000	0.482	0.130	0.093	0.148	0.556	0.037	0.019
Epi.982	0.000	0.180	0.016	0.230	0.115	0.016	0.213	0.426	0.049	0.049
Epi.983	0.018	0.156	0.000	0.174	0.092	0.009	0.092	0.385	0.018	0.101
Epi.984	0.004	0.054	0.000	0.149	0.050	0.035	0.058	0.199	0.012	0.008

User 19 (Correct: 60% First Choice Correct: 60%)

Epi.1040	0.000	0.100	0.000	0.100	0.000	0.000	0.000	0.100	0.000	0.100
Epi.1041	0.100	0.000	0.000	0.100	0.000	0.000	0.000	0.000	0.300	0.000
Epi.1042	0.000	0.000	0.000	0.022	0.022	0.022	0.000	0.000	0.244	0.089
Epi.1043	0.024	0.108	0.000	0.072	0.024	0.000	0.000	0.048	0.205	0.000
Epi.1044	0.000	0.044	0.000	0.222	0.022	0.000	0.022	0.022	0.133	0.000

User 25 (Correct: 100% First Choice Correct: 100%)

Epi.1195	0.023	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.663
Epi.1196	0.067	0.057	0.000	0.057	0.000	0.057	0.000	0.048	0.000	0.705
Epi.1197	0.000	0.030	0.000	0.030	0.000	0.030	0.000	0.030	0.000	0.849
Epi.1198	0.012	0.035	0.000	0.023	0.000	0.023	0.012	0.000	0.000	0.814
Epi.1199	0.059	0.097	0.000	0.013	0.005	0.020	0.003	0.010	0.015	0.842

Table 78: Testing results for experiment 040727-1.

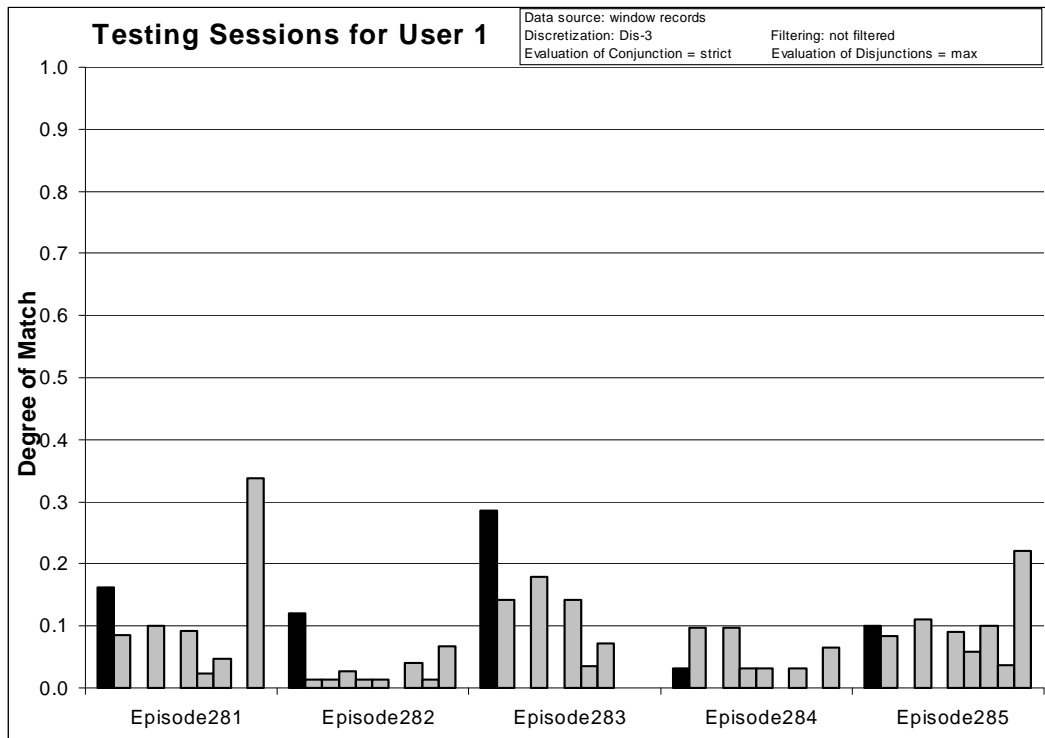


Figure 155: Degrees of match between 10 user models and 5 testing sessions from User 1.

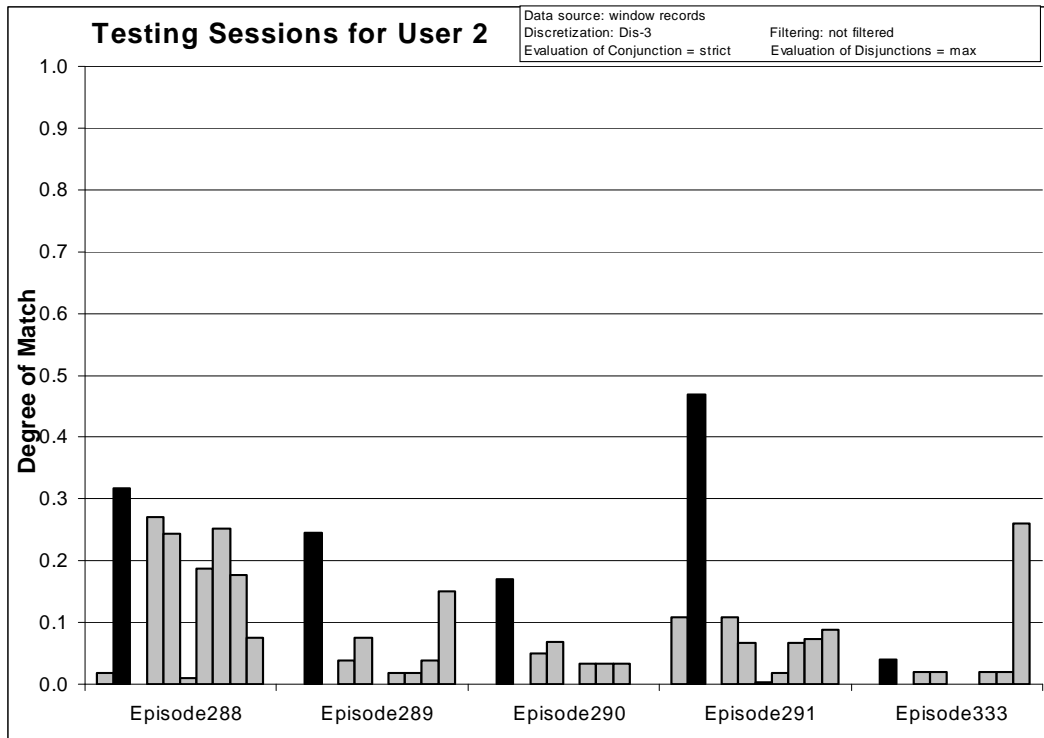


Figure 156: Degrees of match between 10 user models and 5 testing sessions from User 2.

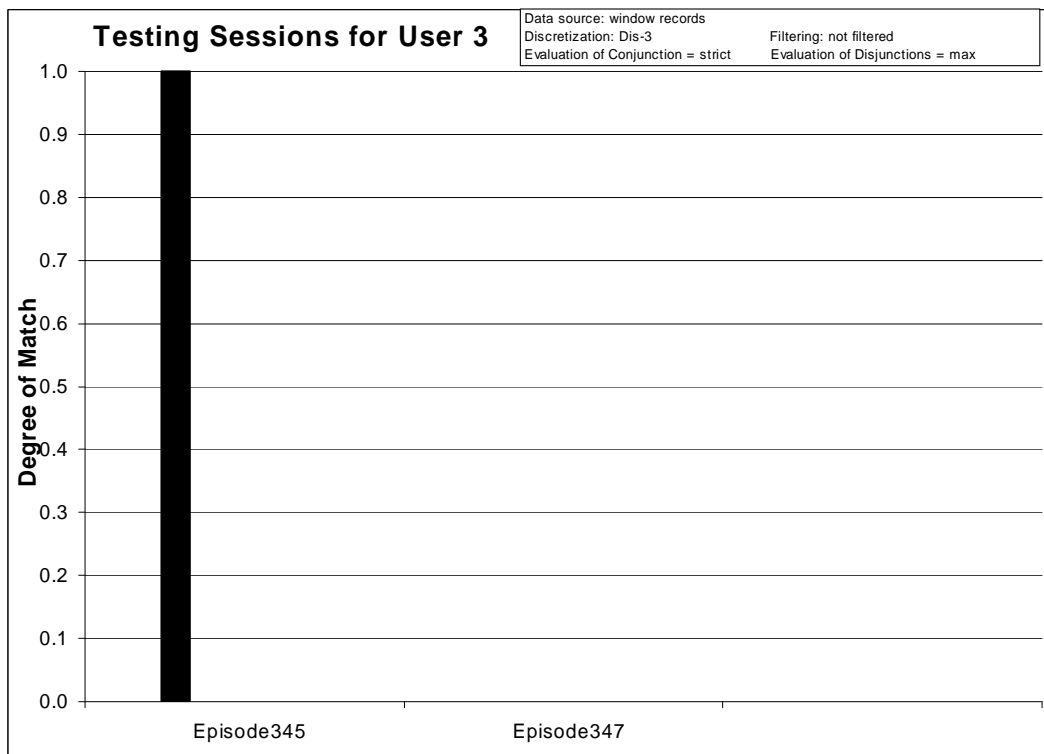


Figure 157: Degrees of match between 10 user models and 2 testing sessions from User 3.

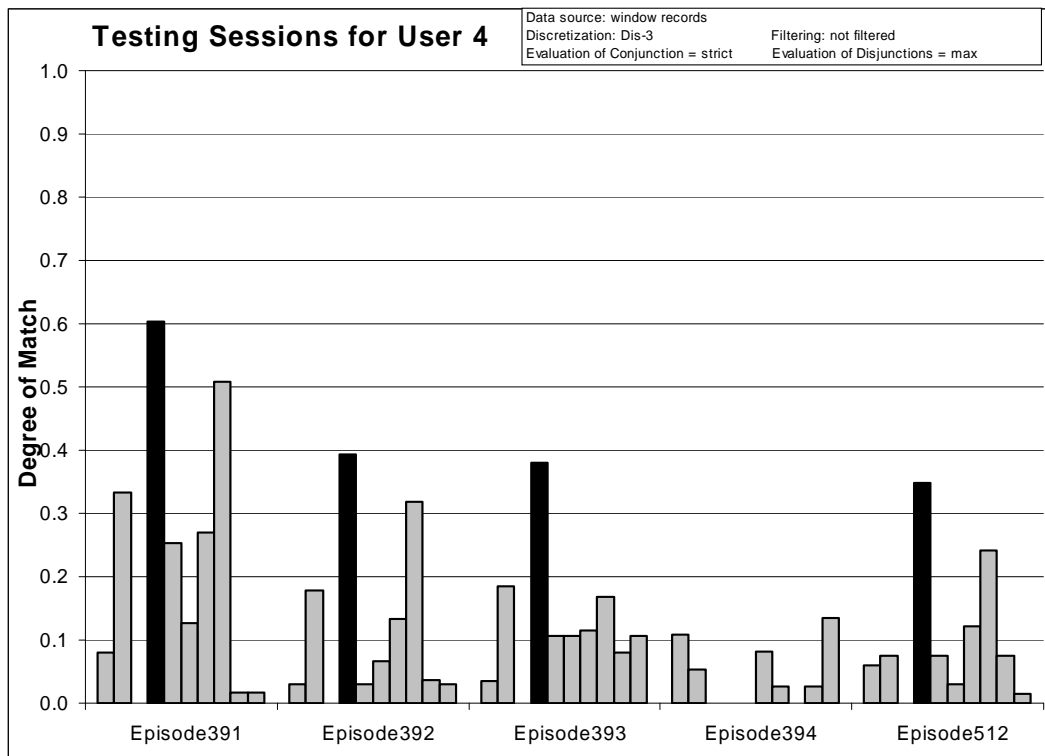


Figure 158: Degrees of match between 10 user models and 5 testing sessions from User 4.

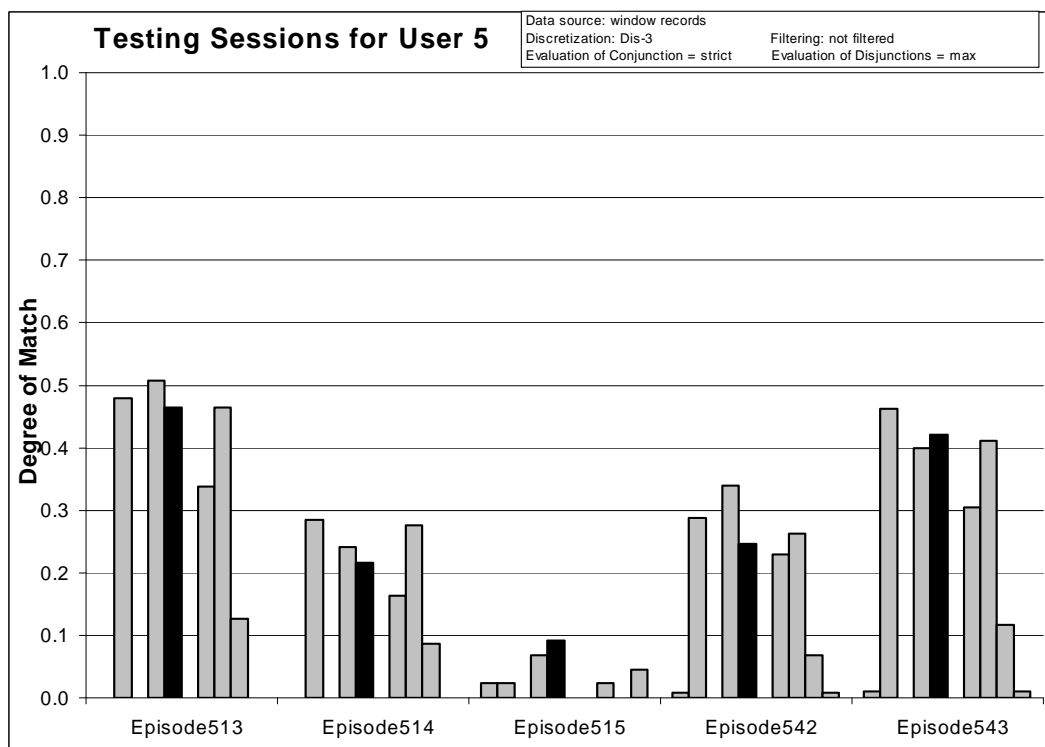


Figure 159: Degrees of match between 10 user models and 5 testing sessions from User 5.

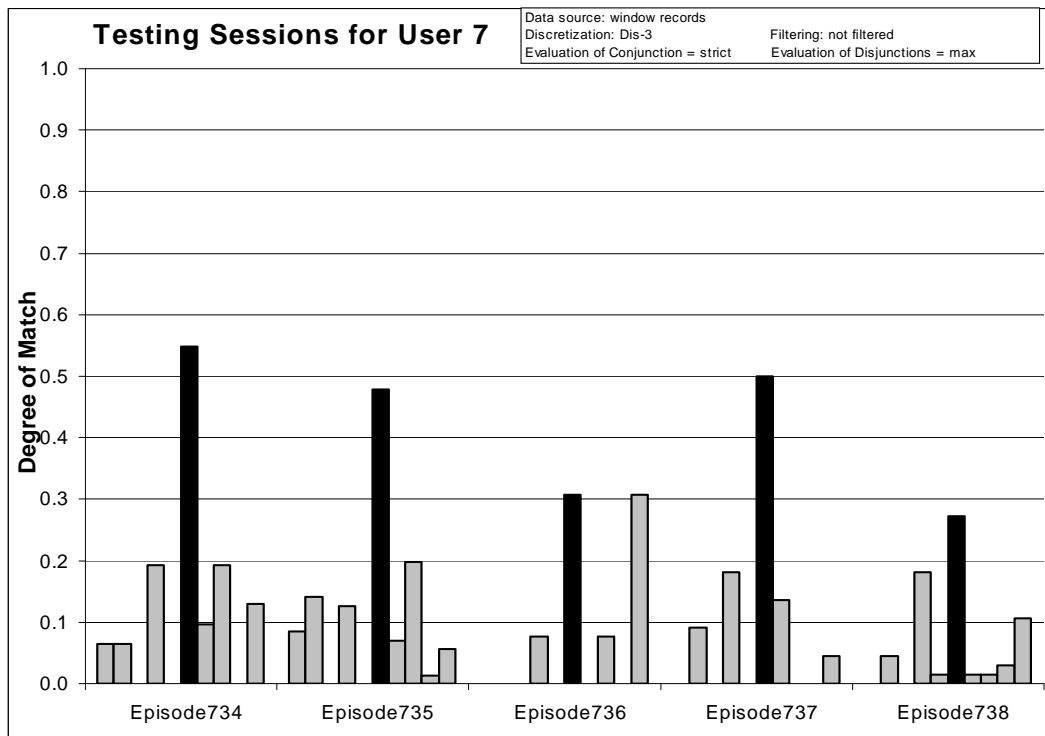


Figure 160: Degrees of match between 10 user models and 5 testing sessions from User 7.

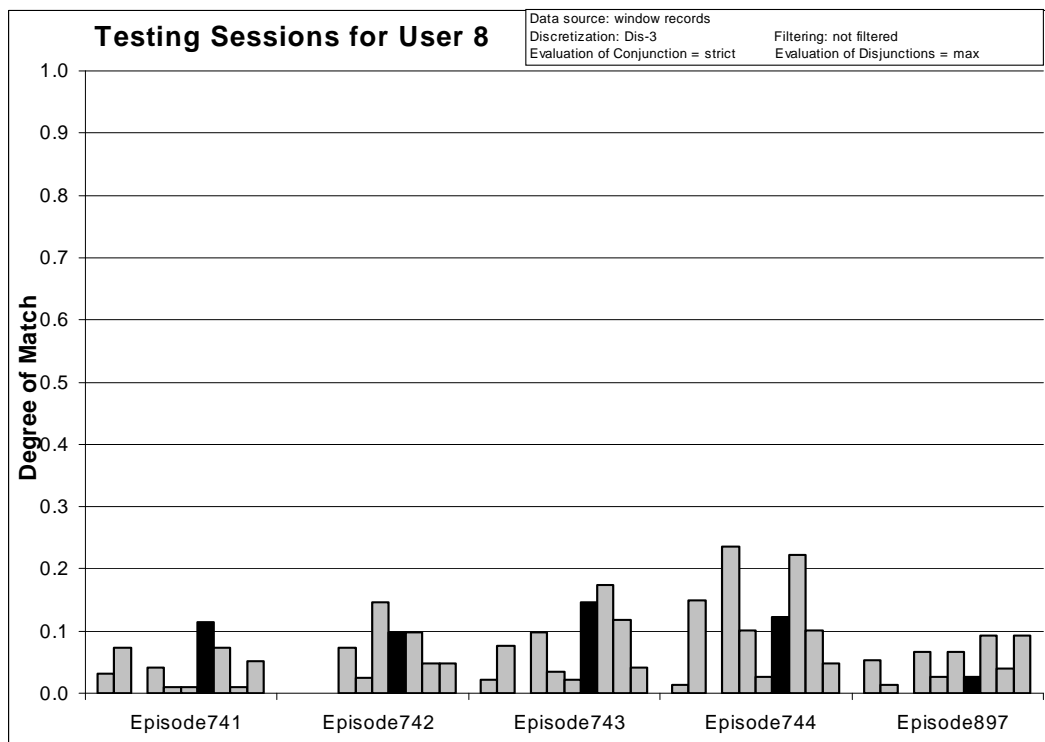


Figure 161: Degrees of match between 10 user models and 5 testing sessions from User 8.

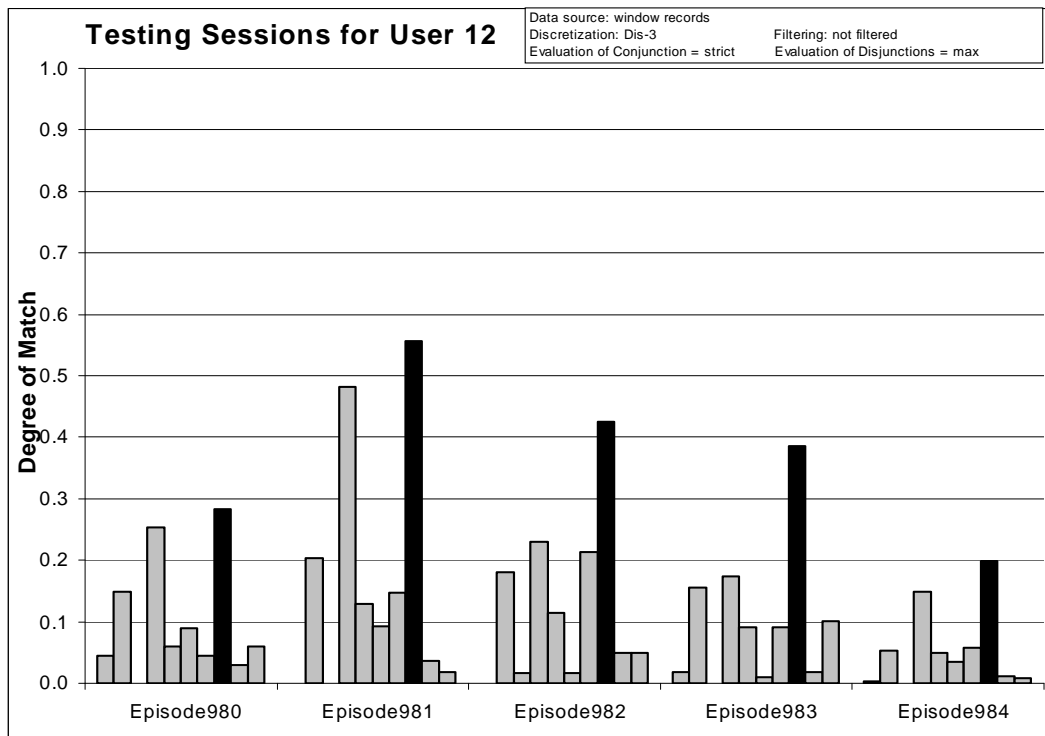


Figure 162: Degrees of match between 10 user models and 5 testing sessions from User 12.

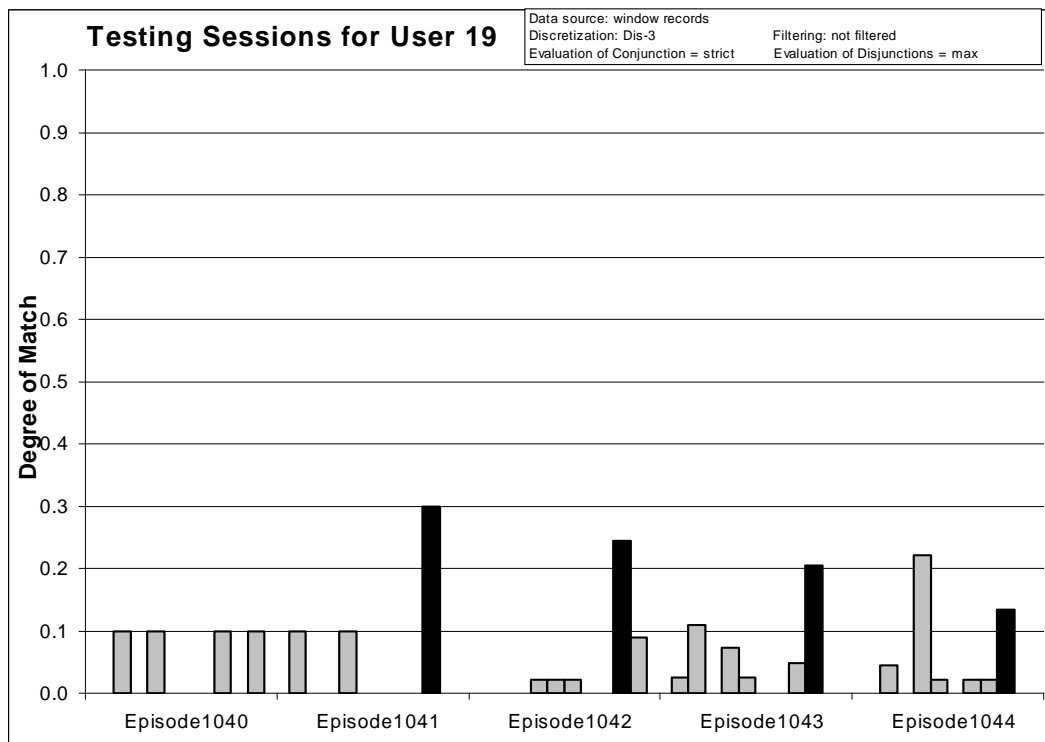


Figure 163: Degrees of match between 10 user models and 5 testing sessions from User 19.

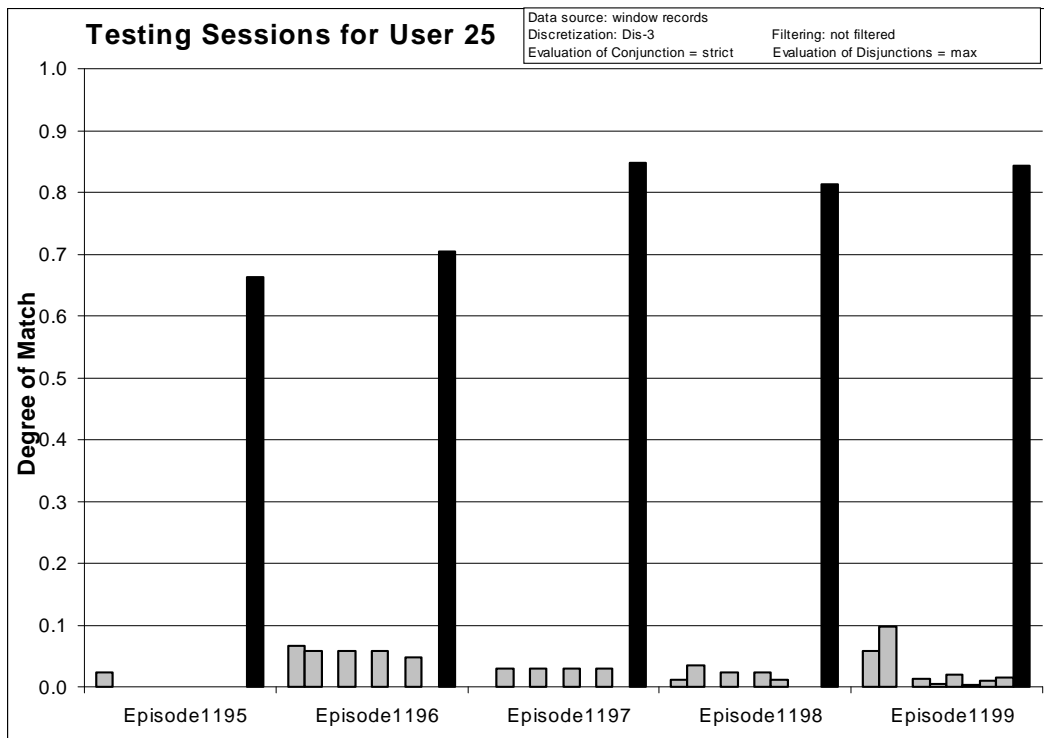


Figure 164: Degrees of match between 10 user models and 5 testing sessions from User 25.

8.5.2 Experiment 040727-2: Unfiltered Data, Characteristic Descriptions

Source Data: window records

Training Dataset:

Discretization: Dis-3

Filtering: not filtered

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

AQ21 Learning Parameters:

maxstar = 1 maxrule = 1 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Characteristic descriptions

Testing Parameters:

Evaluation of Conjunction = selectors ratio

Evaluation of Disjunction = max

Acceptance Threshold = 10%

Accuracy Tolerance = 5%

Learning Results:

Total number of rules: 3172

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	152	569	16	645	228	242	279	424	254	363

Table 79: Number of learned rules for 10 users.

Testing Results:

Correct: 93.62%

Precision: 20.35%

First Choice Correct: 65.96%

First Choice Precision: 100.00%

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
Correct	100%	100%	100%	80%	100%	80%	80%	100%	100%	100%
First Ch. Correct	40%	80%	100%	80%	0%	80%	20%	80%	100%	100%

Table 80: Summary of correct answers for 10 users.

	User 1	User 2	User 3	User 4	User 5	User 7	User 8	User 12	User 19	User 25
User 1 (Correct: 100% First Choice Correct: 40%)										
Epi.281	0.896	0.875	0.620	0.721	0.707	0.813	0.791	0.820	0.781	0.907
Epi.282	0.864	0.839	0.709	0.716	0.701	0.768	0.748	0.775	0.798	0.820
Epi.283	0.939	0.914	0.697	0.885	0.817	0.895	0.891	0.881	0.902	0.875
Epi.284	0.876	0.866	0.671	0.853	0.823	0.850	0.848	0.855	0.883	0.875
Epi.285	0.887	0.895	0.649	0.816	0.785	0.863	0.843	0.868	0.860	0.902
User 2 (Correct: 100% First Choice Correct: 80%)										
Epi.288	0.763	0.935	0.685	0.912	0.890	0.774	0.881	0.916	0.922	0.781
Epi.289	0.810	0.914	0.591	0.777	0.783	0.774	0.808	0.809	0.823	0.898
Epi.290	0.767	0.917	0.668	0.867	0.832	0.771	0.843	0.856	0.862	0.785
Epi.291	0.791	0.943	0.706	0.879	0.799	0.780	0.817	0.850	0.875	0.791
Epi.333	0.797	0.869	0.573	0.666	0.704	0.711	0.769	0.738	0.777	0.910
User 3 (Correct: 100% First Choice Correct: 100%)										
Epi.345	0.688	0.714	1.000	0.765	0.813	0.800	0.750	0.875	0.786	0.667
Epi.347	0.759	0.780	0.879	0.676	0.686	0.760	0.684	0.757	0.758	0.729
User 4 (Correct: 80% First Choice Correct: 80%)										
Epi.391	0.782	0.943	0.648	0.970	0.917	0.804	0.921	0.958	0.911	0.781
Epi.392	0.850	0.926	0.660	0.942	0.854	0.872	0.877	0.926	0.916	0.869
Epi.393	0.824	0.917	0.642	0.923	0.868	0.829	0.880	0.915	0.891	0.841
Epi.394	0.874	0.889	0.745	0.836	0.778	0.867	0.826	0.843	0.885	0.879
Epi.512	0.813	0.906	0.678	0.935	0.882	0.811	0.905	0.926	0.908	0.792
User 5 (Correct: 100% First Choice Correct: 0%)										
Epi.513	0.691	0.957	0.639	0.954	0.952	0.723	0.923	0.948	0.912	0.703

Epi.514	0.730	0.914	0.684	0.897	0.885	0.734	0.867	0.895	0.878	0.698
Epi.515	0.802	0.836	0.681	0.829	0.811	0.786	0.787	0.831	0.839	0.802
Epi.542	0.761	0.926	0.660	0.920	0.900	0.772	0.894	0.918	0.903	0.750
Epi.543	0.764	0.921	0.685	0.904	0.893	0.766	0.883	0.913	0.896	0.744

User 7 (Correct: 80% First Choice Correct: 80%)

Epi.734	0.878	0.895	0.641	0.841	0.794	0.934	0.875	0.900	0.852	0.931
Epi.735	0.856	0.908	0.697	0.896	0.821	0.944	0.872	0.909	0.894	0.896
Epi.736	0.834	0.866	0.600	0.715	0.708	0.814	0.813	0.776	0.770	0.938
Epi.737	0.894	0.906	0.693	0.903	0.853	0.956	0.910	0.907	0.896	0.916
Epi.738	0.881	0.903	0.700	0.877	0.830	0.916	0.867	0.877	0.892	0.912

User 8 (Correct: 80% First Choice Correct: 20%)

Epi.741	0.886	0.905	0.697	0.875	0.859	0.878	0.892	0.903	0.897	0.895
Epi.742	0.831	0.881	0.697	0.852	0.820	0.838	0.892	0.888	0.865	0.857
Epi.743	0.799	0.908	0.710	0.896	0.837	0.796	0.875	0.889	0.907	0.806
Epi.744	0.771	0.890	0.695	0.873	0.827	0.770	0.855	0.880	0.896	0.764
Epi.897	0.874	0.888	0.687	0.857	0.862	0.881	0.857	0.890	0.897	0.904

User 12 (Correct: 100% First Choice Correct: 80%)

Epi.980	0.860	0.903	0.640	0.865	0.818	0.852	0.874	0.932	0.863	0.879
Epi.981	0.820	0.937	0.646	0.958	0.884	0.858	0.909	0.967	0.895	0.860
Epi.982	0.818	0.915	0.673	0.883	0.831	0.827	0.879	0.939	0.883	0.840
Epi.983	0.845	0.917	0.671	0.871	0.846	0.838	0.895	0.933	0.875	0.872
Epi.984	0.779	0.768	0.681	0.746	0.717	0.732	0.715	0.781	0.820	0.730

User 19 (Correct: 100% First Choice Correct: 100%)

Epi.1040	0.861	0.872	0.653	0.817	0.822	0.769	0.833	0.846	0.897	0.843
Epi.1041	0.851	0.814	0.588	0.804	0.809	0.785	0.818	0.803	0.926	0.856
Epi.1042	0.817	0.847	0.676	0.821	0.791	0.787	0.807	0.803	0.933	0.853
Epi.1043	0.813	0.874	0.639	0.848	0.807	0.807	0.823	0.864	0.907	0.837
Epi.1044	0.749	0.886	0.661	0.892	0.820	0.772	0.828	0.863	0.899	0.783

User 25 (Correct: 100% First Choice Correct: 100%)

Epi.1195	0.835	0.866	0.525	0.609	0.649	0.733	0.766	0.752	0.734	0.969
Epi.1196	0.863	0.887	0.513	0.635	0.677	0.780	0.783	0.784	0.744	0.976
Epi.1197	0.888	0.907	0.526	0.622	0.671	0.803	0.801	0.815	0.744	0.991
Epi.1198	0.858	0.888	0.535	0.655	0.689	0.795	0.798	0.806	0.755	0.987
Epi.1199	0.861	0.898	0.474	0.553	0.610	0.750	0.751	0.741	0.691	0.984

Table 81: Testing results for experiment 040727-2.

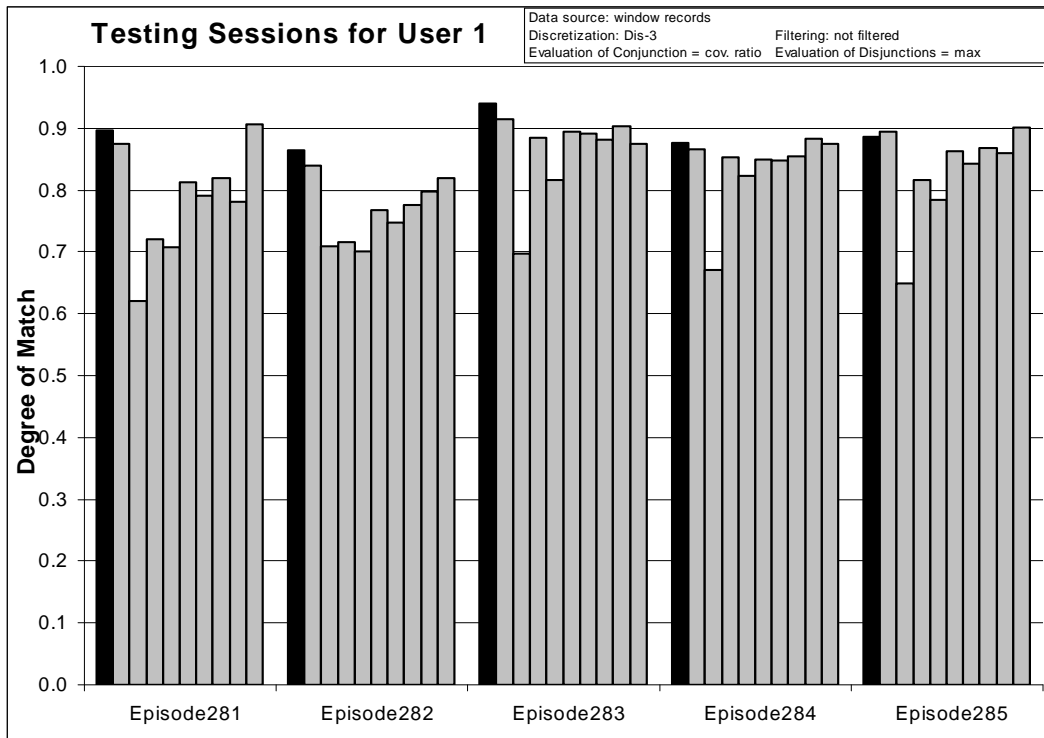


Figure 165: Degrees of match between 10 user models and 5 testing sessions from User 1.

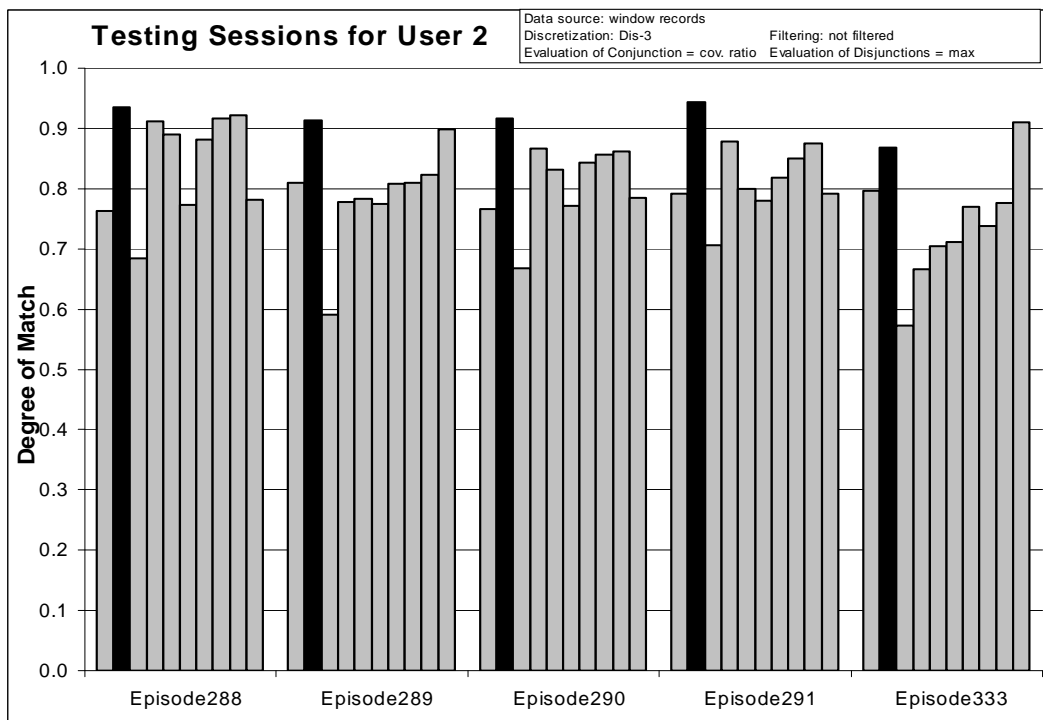


Figure 166: Degrees of match between 10 user models and 5 testing sessions from User 2.

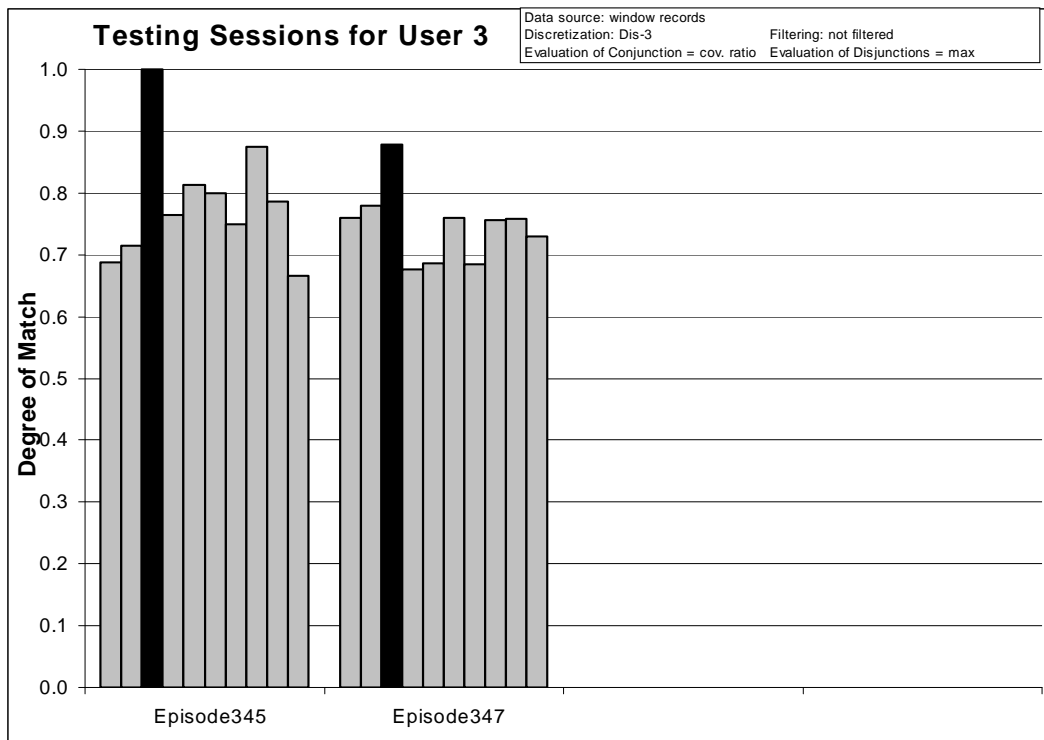


Figure 167: Degrees of match between 10 user models and 2 testing sessions from User 3.

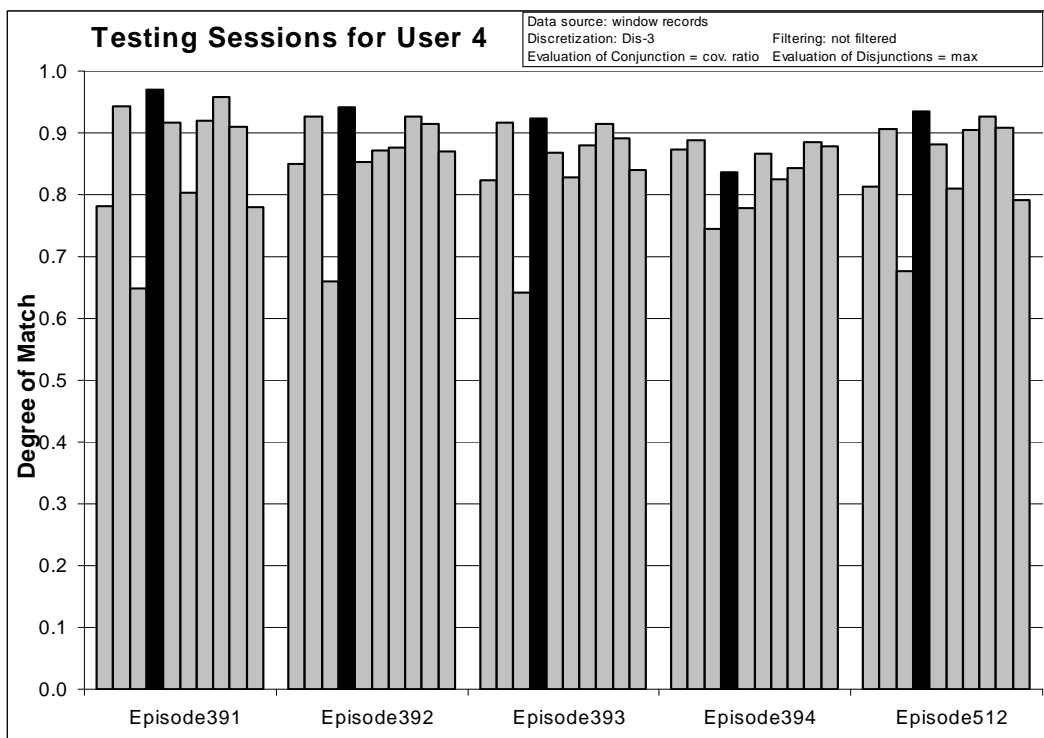


Figure 168: Degrees of match between 10 user models and 5 testing sessions from User 4.

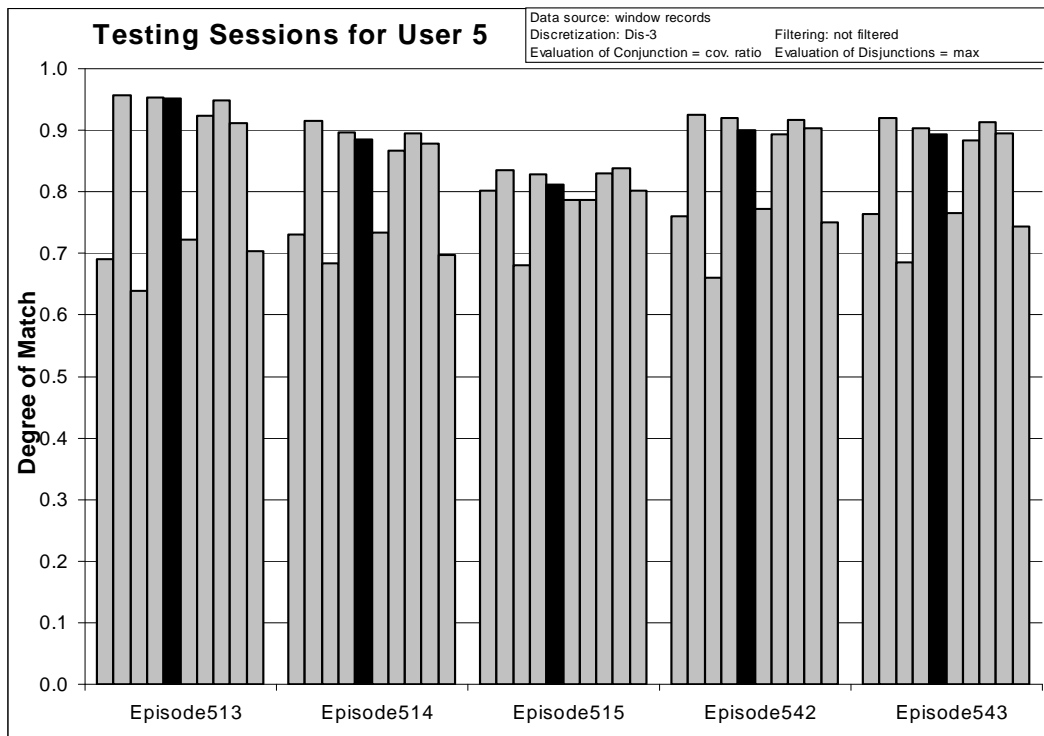


Figure 169: Degrees of match between 10 user models and 5 testing sessions from User 5.

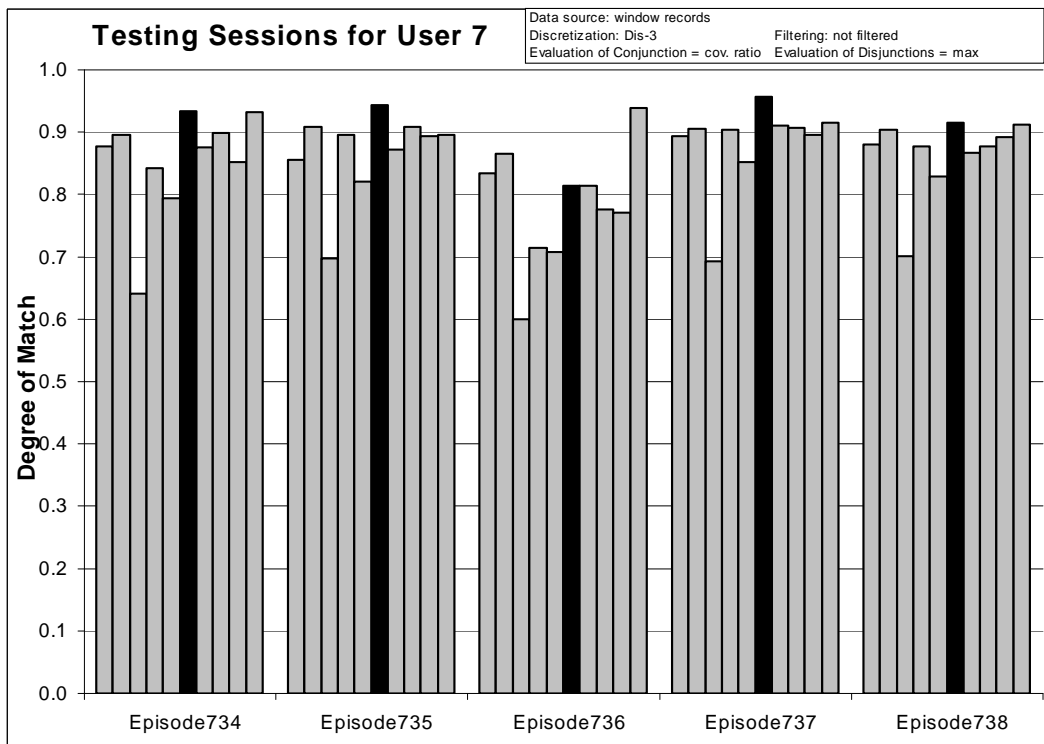


Figure 170: Degrees of match between 10 user models and 5 testing sessions from User 7.

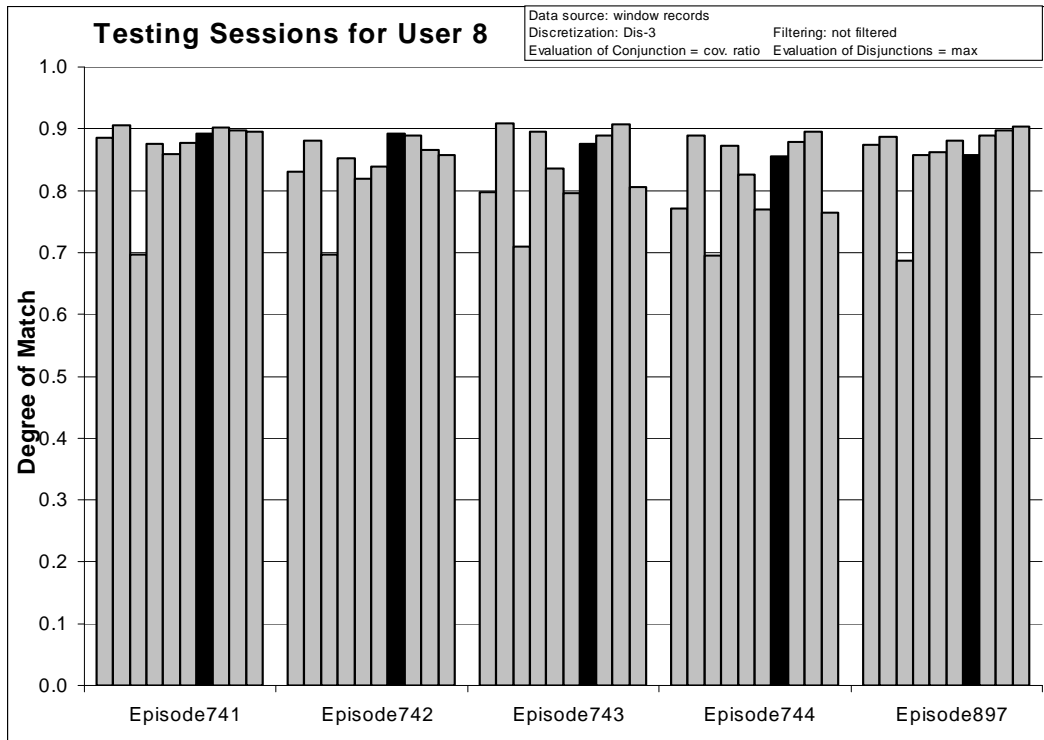


Figure 171: Degrees of match between 10 user models and 5 testing sessions from User 8.

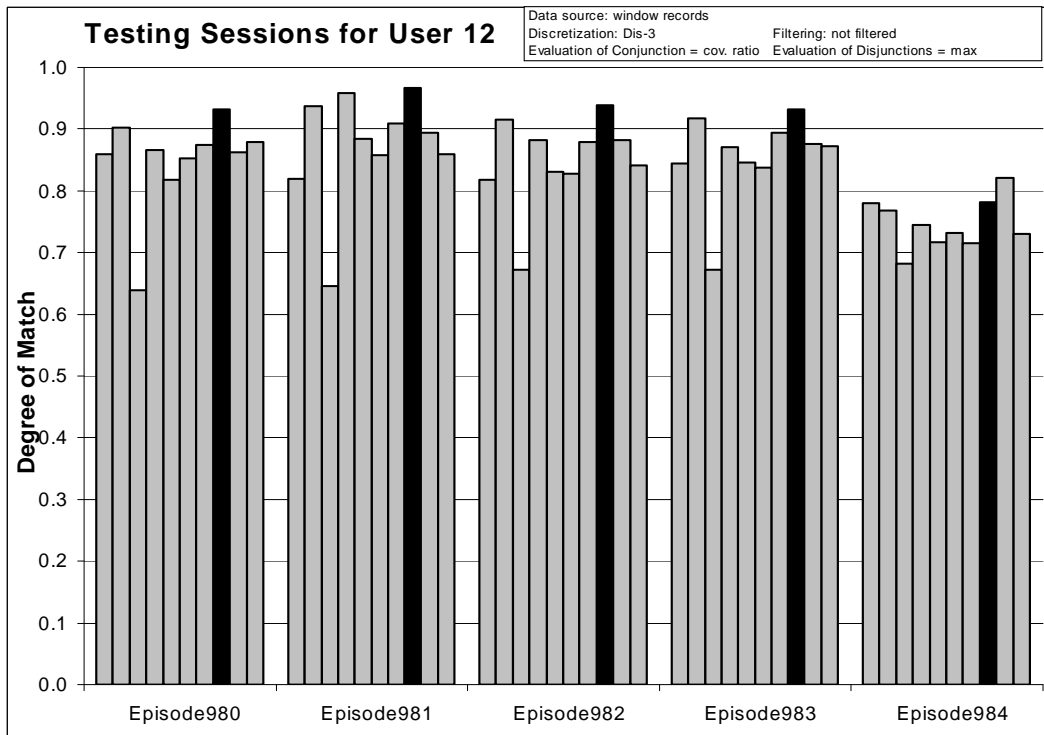


Figure 172: Degrees of match between 10 user models and 5 testing sessions from User 12.

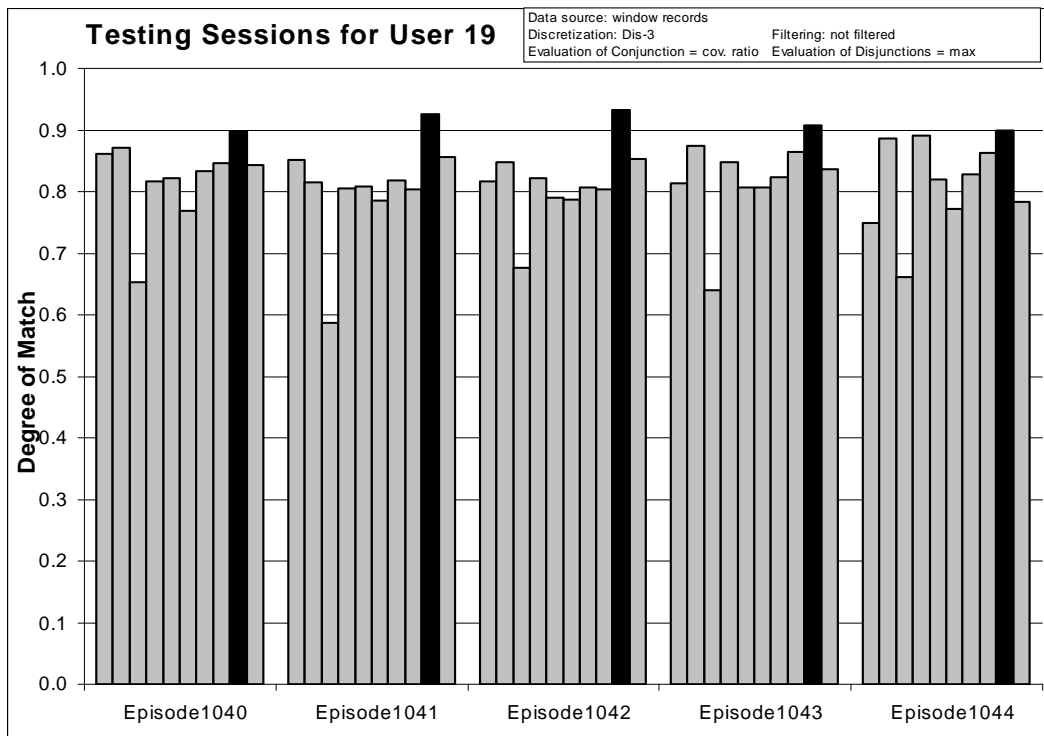


Figure 173: Degrees of match between 10 user models and 5 testing sessions from User 19.

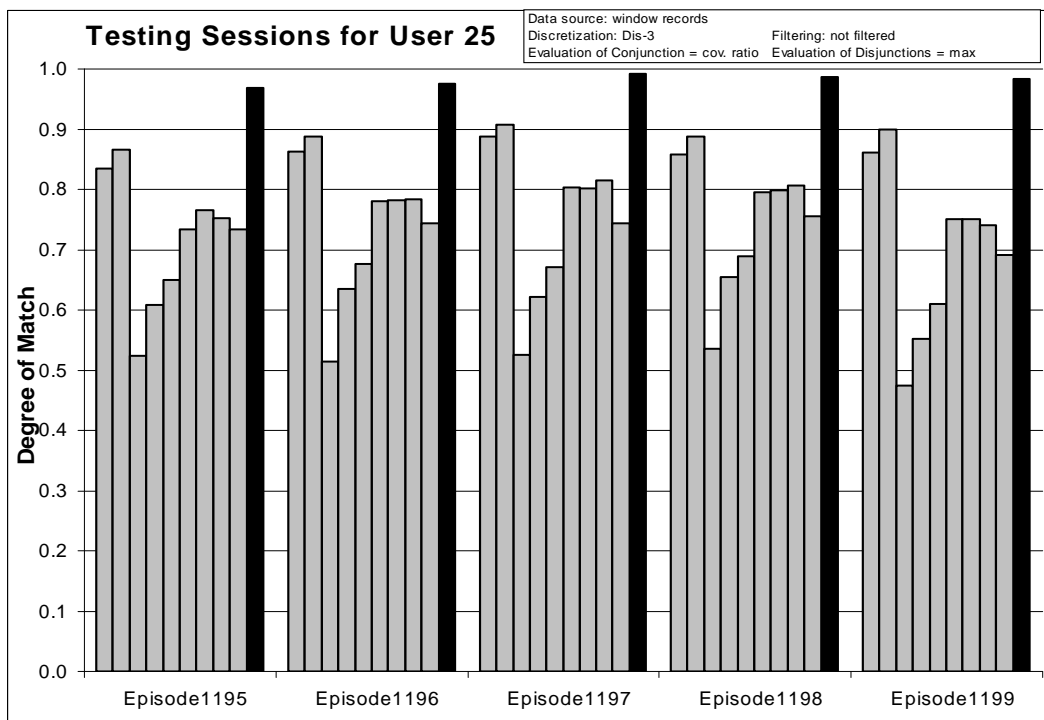


Figure 174: Degrees of match between 10 user models and 5 testing sessions from User 25.

8.5.3 Experiment 040727-3: Unfiltered Data, Simplicity-based Descriptions

Source Data: window records

Training Dataset:

Discretization: Dis-3

Filtering: not filtered

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

AQ21 Learning Parameters:

maxstar = 1 maxrule = 1 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Simplicity-based descriptions

Testing Parameters:

Evaluation of Conjunction = strict

Evaluation of Disjunction = max

Acceptance Threshold = 10%

Accuracy Tolerance = 5%

Learning Results:

Total number of rules: 3781

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	204	693	27	711	286	308	328	462	339	423

Table 82: Number of learned rules for 10 users.

Testing Results:

Correct: 72.34%

Precision: 89.32%

First Choice Correct: 70.21%

First Choice Precision: 100.00%

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
Correct	40%	80%	100%	80%	40%	100%	20%	100%	80%	100%
First Ch.										
Correct	40%	80%	100%	80%	20%	100%	20%	100%	80%	100%

Table 83: Summary of correct answers for 10 users.

	User 1	User 2	User 3	User 4	User 5	User 7	User 8	User 12	User 19	User 25
User 1 (Correct: 40% First Choice Correct: 40%)										
Epi.281	0.269	0.131	0.000	0.115	0.015	0.085	0.023	0.054	0.015	0.377
Epi.282	0.320	0.240	0.053	0.107	0.027	0.080	0.053	0.053	0.213	0.227
Epi.283	0.357	0.036	0.000	0.286	0.000	0.107	0.036	0.071	0.071	0.071
Epi.284	0.129	0.129	0.032	0.194	0.032	0.032	0.032	0.097	0.097	0.032
Epi.285	0.168	0.121	0.000	0.121	0.037	0.126	0.084	0.147	0.042	0.279
User 2 (Correct: 80% First Choice Correct: 80%)										
Epi.288	0.009	0.383	0.000	0.365	0.308	0.065	0.206	0.327	0.168	0.047
Epi.289	0.019	0.359	0.000	0.057	0.094	0.038	0.057	0.057	0.057	0.264
Epi.290	0.017	0.373	0.000	0.186	0.068	0.000	0.119	0.051	0.119	0.085
Epi.291	0.124	0.635	0.000	0.194	0.121	0.048	0.060	0.124	0.095	0.111
Epi.333	0.020	0.180	0.020	0.060	0.000	0.040	0.040	0.100	0.080	0.340
User 3 (Correct: 100% First Choice Correct: 100%)										
Epi.345	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Epi.347	0.000	0.000	0.400	0.000	0.000	0.000	0.000	0.200	0.000	0.000
User 4 (Correct: 80% First Choice Correct: 80%)										
Epi.391	0.095	0.413	0.000	0.730	0.286	0.143	0.270	0.556	0.016	0.016
Epi.392	0.030	0.244	0.007	0.437	0.052	0.067	0.141	0.348	0.104	0.022
Epi.393	0.062	0.195	0.000	0.513	0.150	0.159	0.168	0.239	0.124	0.142
Epi.394	0.189	0.162	0.000	0.135	0.054	0.135	0.054	0.027	0.027	0.135
Epi.512	0.106	0.121	0.015	0.470	0.091	0.121	0.197	0.242	0.076	0.046
User 5 (Correct: 40% First Choice Correct: 20%)										
Epi.513	0.000	0.578	0.014	0.535	0.465	0.000	0.366	0.507	0.141	0.028
Epi.514	0.026	0.267	0.000	0.328	0.250	0.009	0.190	0.336	0.095	0.000
Epi.515	0.068	0.114	0.000	0.114	0.159	0.023	0.023	0.068	0.114	0.046
Epi.542	0.017	0.364	0.000	0.415	0.314	0.009	0.220	0.305	0.186	0.009
Epi.543	0.063	0.474	0.000	0.421	0.453	0.011	0.316	0.421	0.179	0.021
User 7 (Correct: 100% First Choice Correct: 100%)										
Epi.734	0.065	0.097	0.000	0.258	0.000	0.677	0.129	0.226	0.000	0.161
Epi.735	0.085	0.239	0.000	0.254	0.000	0.676	0.113	0.282	0.042	0.113
Epi.736	0.000	0.077	0.000	0.077	0.000	0.539	0.077	0.077	0.000	0.385
Epi.737	0.046	0.091	0.000	0.227	0.000	0.591	0.182	0.000	0.046	0.046
Epi.738	0.015	0.106	0.000	0.242	0.000	0.318	0.015	0.091	0.167	0.167
User 8 (Correct: 20% First Choice Correct: 20%)										
Epi.741	0.156	0.188	0.021	0.083	0.042	0.094	0.115	0.135	0.063	0.167
Epi.742	0.024	0.000	0.024	0.098	0.049	0.122	0.171	0.317	0.000	0.073
Epi.743	0.028	0.153	0.007	0.194	0.042	0.049	0.215	0.194	0.132	0.028
Epi.744	0.039	0.201	0.009	0.297	0.153	0.052	0.205	0.301	0.183	0.074
Epi.897	0.092	0.158	0.000	0.118	0.092	0.118	0.079	0.145	0.132	0.197
User 12 (Correct: 100% First Choice Correct: 100%)										
Epi.980	0.075	0.209	0.000	0.328	0.075	0.075	0.075	0.343	0.060	0.090
Epi.981	0.000	0.222	0.000	0.537	0.167	0.093	0.204	0.667	0.074	0.037

Epi.982	0.000	0.197	0.000	0.279	0.115	0.131	0.180	0.492	0.066	0.049
Epi.983	0.073	0.248	0.000	0.229	0.119	0.055	0.147	0.431	0.064	0.064
Epi.984	0.008	0.069	0.004	0.172	0.061	0.035	0.061	0.422	0.046	0.004

User 19 (Correct: 80% First Choice Correct: 80%)

Epi.1040	0.000	0.100	0.000	0.100	0.000	0.000	0.000	0.100	0.200	0.100
Epi.1041	0.000	0.100	0.000	0.400	0.100	0.000	0.000	0.000	0.500	0.000
Epi.1042	0.022	0.067	0.000	0.111	0.044	0.000	0.044	0.000	0.422	0.022
Epi.1043	0.024	0.157	0.012	0.229	0.108	0.072	0.036	0.121	0.253	0.060
Epi.1044	0.022	0.156	0.000	0.511	0.067	0.044	0.067	0.133	0.156	0.022

User 25 (Correct: 100% First Choice Correct: 100%)

Epi.1195	0.047	0.058	0.000	0.035	0.012	0.012	0.012	0.000	0.012	0.779
Epi.1196	0.067	0.076	0.000	0.067	0.029	0.086	0.019	0.095	0.010	0.771
Epi.1197	0.030	0.030	0.000	0.061	0.030	0.091	0.000	0.091	0.000	0.818
Epi.1198	0.023	0.105	0.000	0.012	0.035	0.058	0.023	0.000	0.000	0.802
Epi.1199	0.069	0.132	0.000	0.020	0.008	0.025	0.015	0.018	0.025	0.863

Table 84: Testing results for experiment 040727-3.

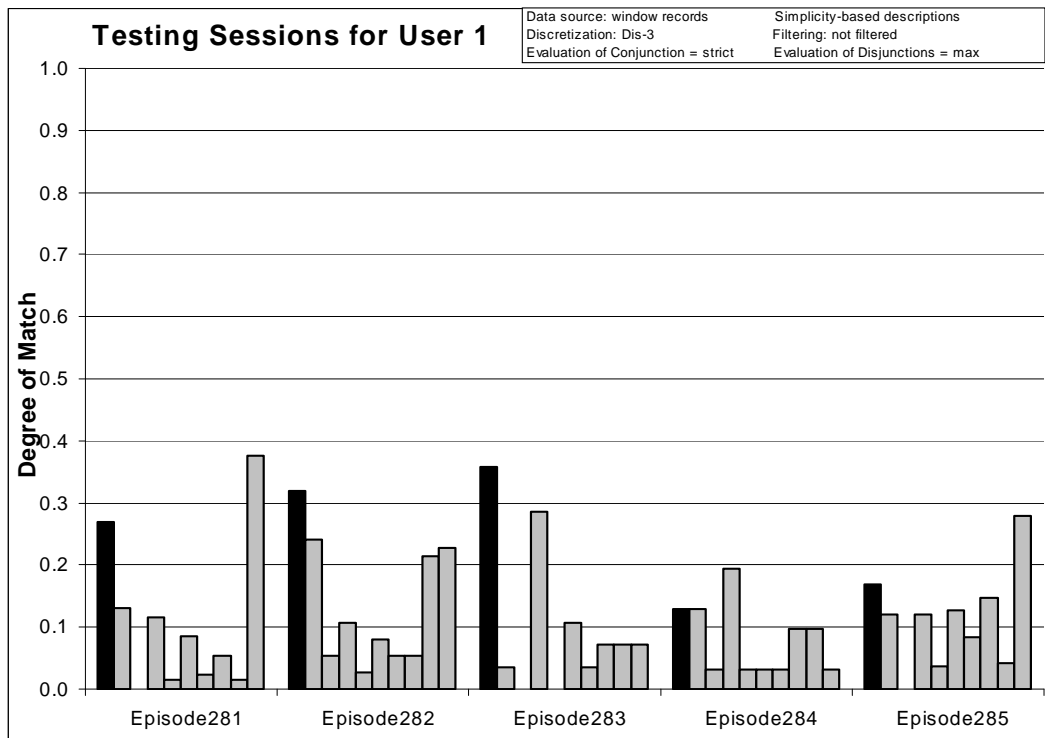


Figure 175: Degrees of match between 10 user models and 5 testing sessions from User 1.

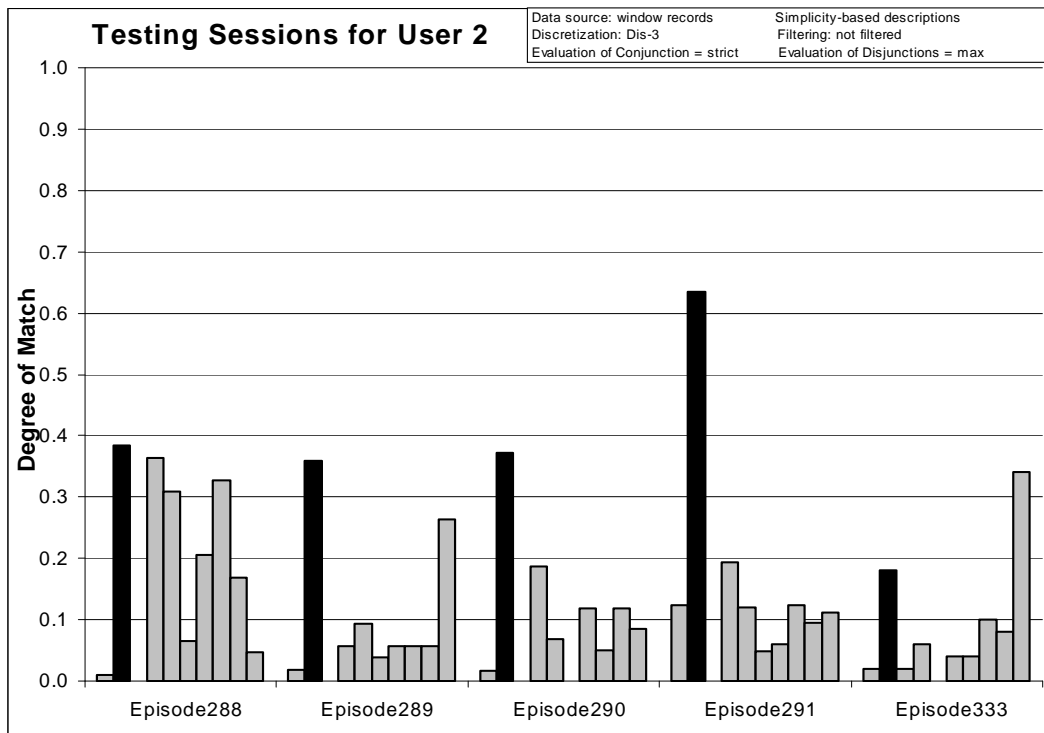


Figure 176: Degrees of match between 10 user models and 5 testing sessions from User 2.

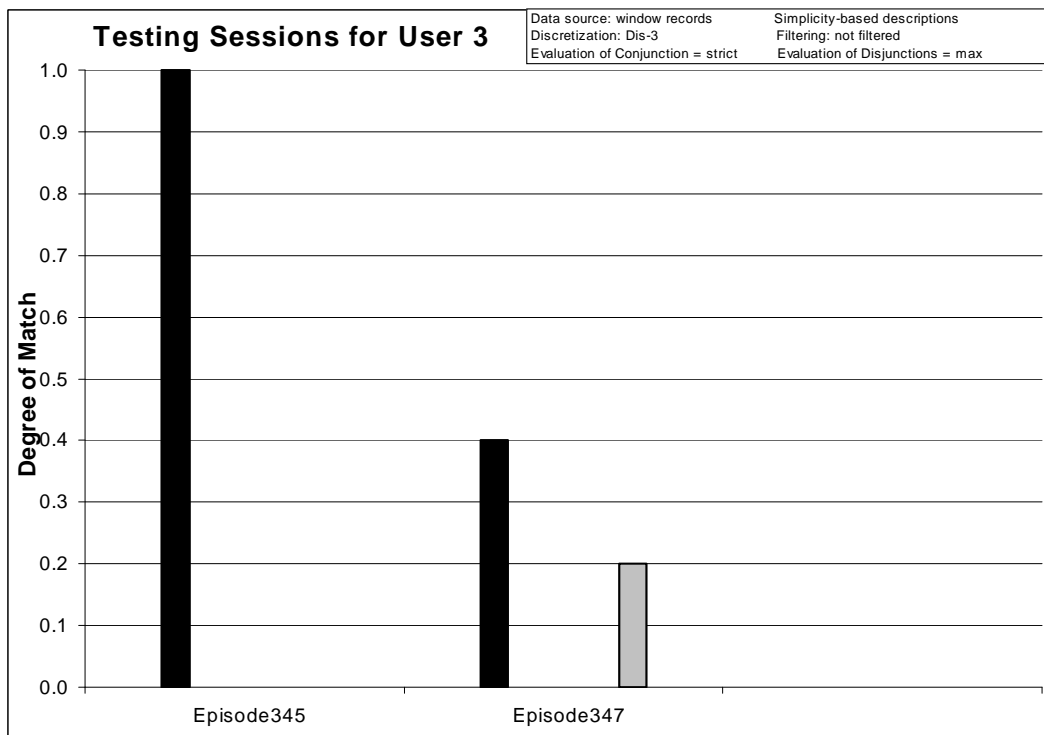


Figure 177: Degrees of match between 10 user models and 2 testing sessions from User 3.

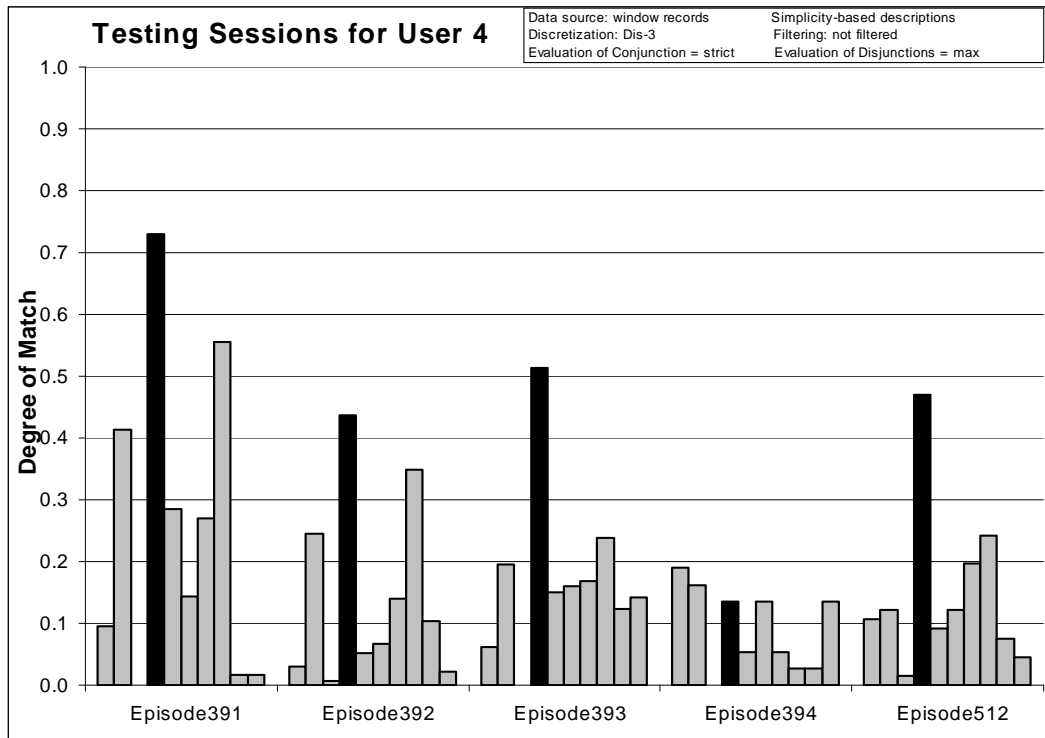


Figure 178: Degrees of match between 10 user models and 5 testing sessions from User 4.

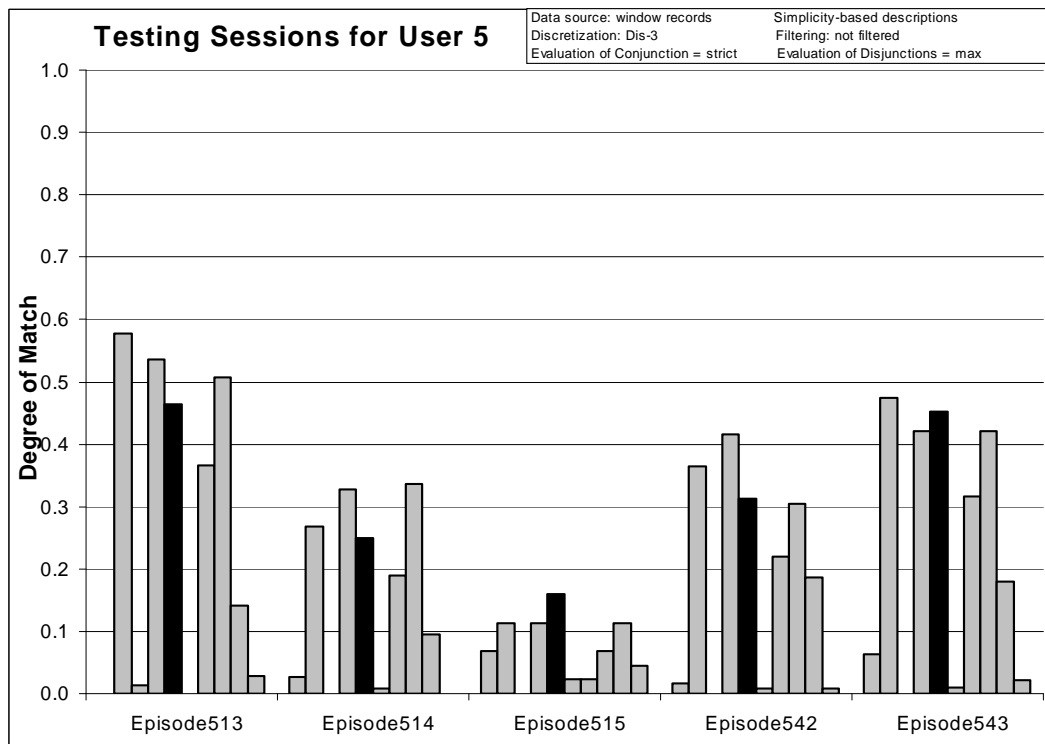


Figure 179: Degrees of match between 10 user models and 5 testing sessions from User 5.

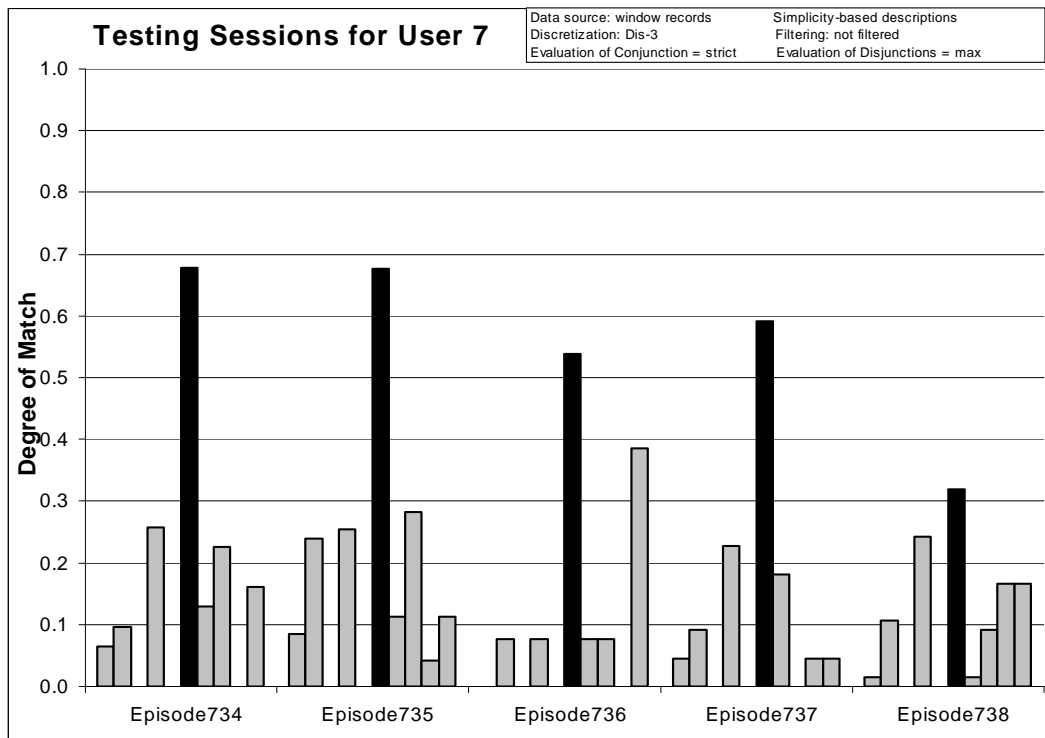


Figure 180: Degrees of match between 10 user models and 5 testing sessions from User 7.

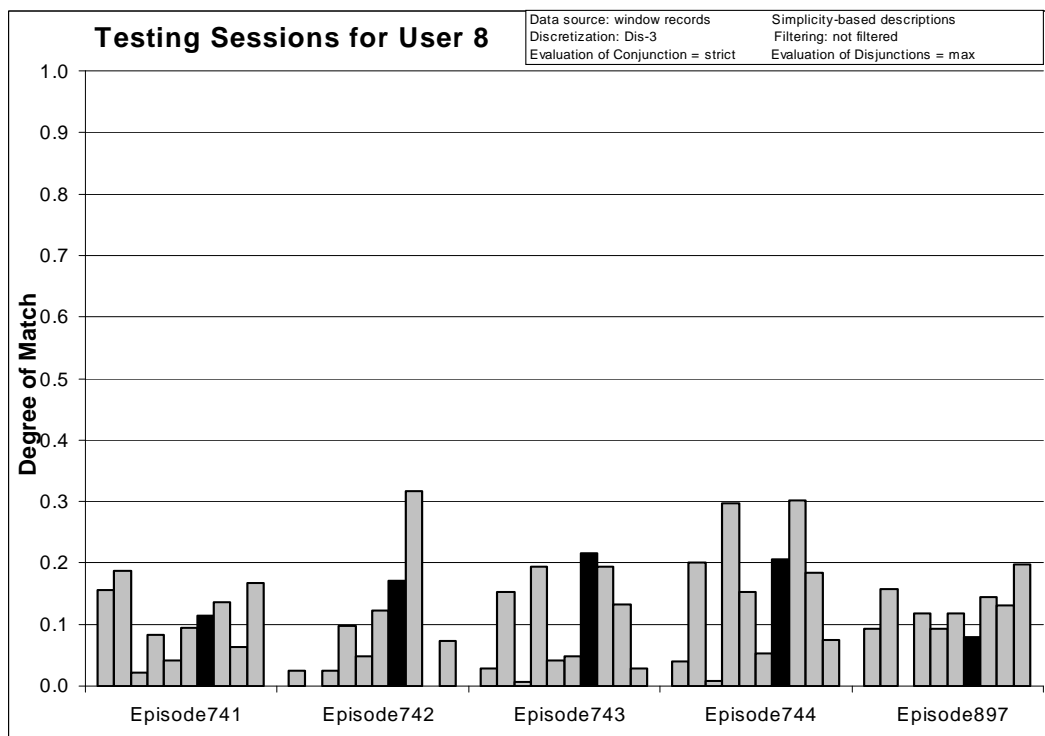


Figure 181: Degrees of match between 10 user models and 5 testing sessions from User 8.

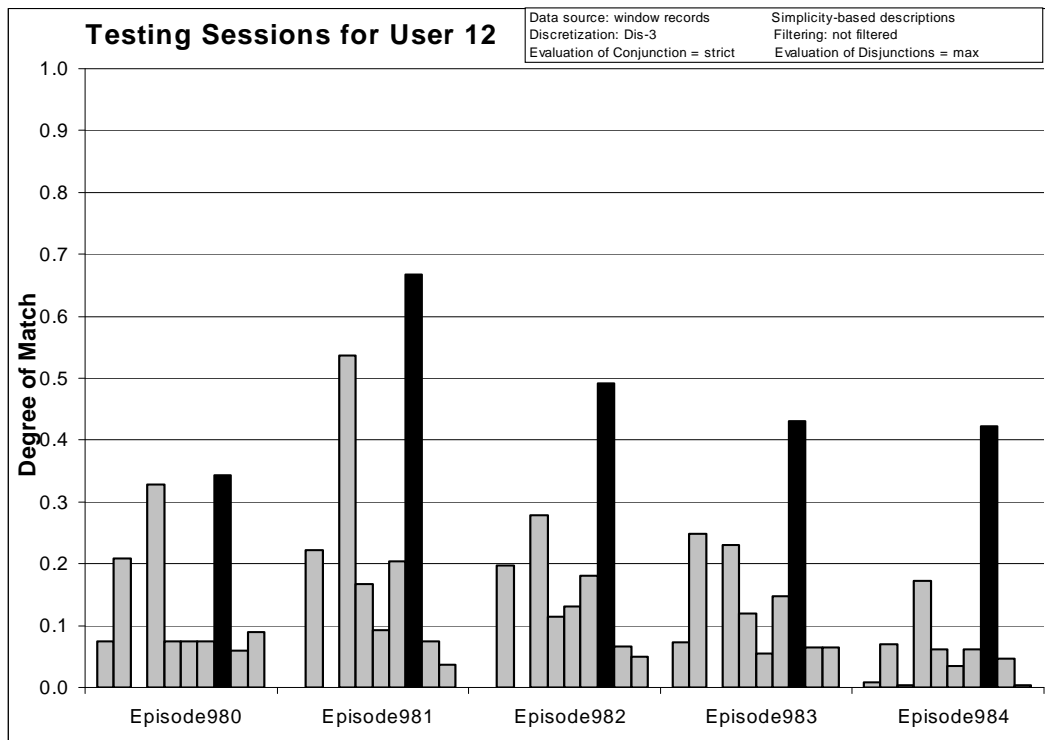


Figure 182: Degrees of match between 10 user models and 5 testing sessions from User 12.

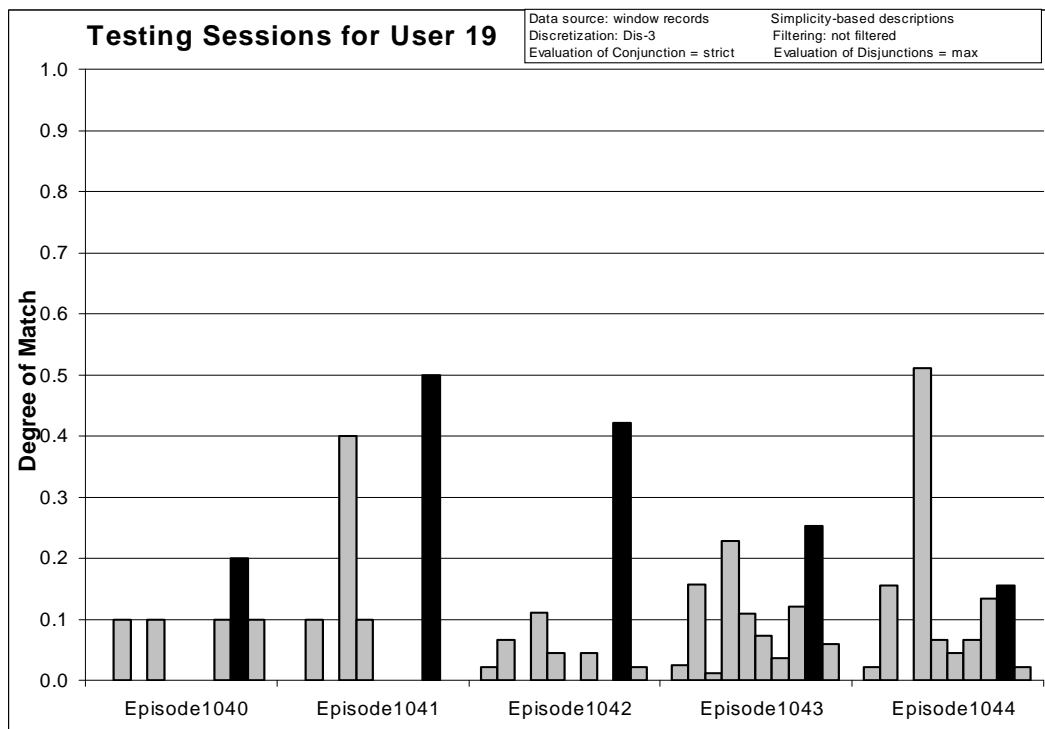


Figure 183: Degrees of match between 10 user models and 5 testing sessions from User 19.

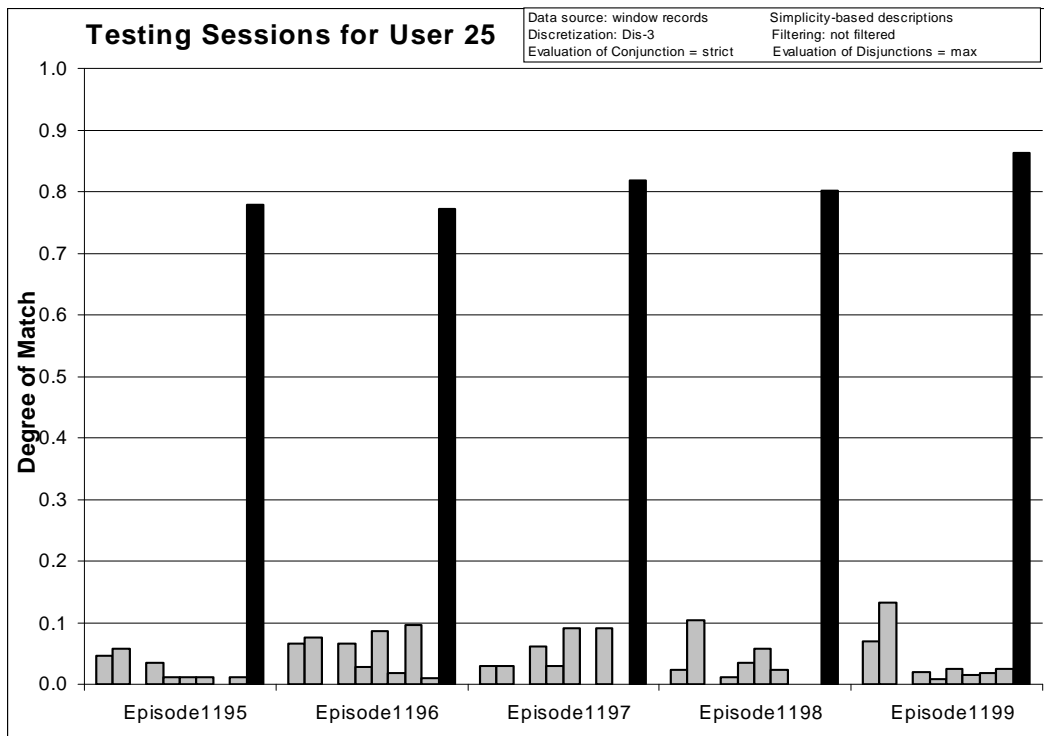


Figure 184: Degrees of match between 10 user models and 5 testing sessions from User 25.

8.5.4 $n \times 6$ -Grams for $n = 1, 2, 3, 4, 5, 6, 7$, and 8

Source Data: window records

Training Dataset:

Discretization: Dis-3

Filtering: not filtered

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

AQ21 Learning Parameters:

maxstar = 1 maxrule = 1 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Simplicity-based descriptions

Testing Parameters:

Evaluation of Conjunction = strict

Evaluation of Disjunction = max

Acceptance Threshold = 10%

Accuracy Tolerance = 5%

Learning and Testing Results:

n	Total # of rules	Correct	Precision	First Choice Correct	First Choice Precision
1	601	78.26%	37.57%	71.74%	66.33%
2	2230	65.22%	77.01%	60.87%	93.20%
3	3168	69.57%	87.18%	67.39%	93.20%
4	3097	67.39%	85.32%	65.22%	93.20%
5	3886	67.39%	87.18%	67.39%	93.20%
6	4081	65.22%	85.32%	65.22%	93.20%
7	4142	69.57%	91.11%	67.39%	95.37%
8	4165	67.39%	87.18%	65.22%	93.20%

Table 85: Summary of learning and testing for n -gram sizes 1, 2, 3, 4, 5, 6, 7, and 8.

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	49	94	11	79	47	53	58	66	71	73
Correct	100%	100%	100%	80%	20%	100%	0%	100%	100%	100%
First Ch. Correct	80%	100%	100%	80%	20%	80%	0%	80%	100%	100%

Table 86: Summary of learning and testing for $n=1$.

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	144	397	20	371	161	179	191	247	252	268
Correct	40%	80%	0%	60%	20%	60%	60%	100%	80%	100%
First Ch. Correct	20%	80%	0%	60%	20%	60%	60%	80%	80%	100%

Table 87: Summary of learning and testing for $n=2$.

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	180	570	23	585	235	258	264	374	308	371
Correct	40%	80%	0%	80%	20%	100%	60%	100%	60%	100%
First Ch. Correct	40%	80%	0%	80%	20%	100%	40%	100%	60%	100%

Table 88: Summary of learning and testing for $n=3$.

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	153	572	15	628	208	250	252	412	262	345
Correct	20%	80%	0%	80%	40%	100%	20%	100%	80%	100%
First Ch. Correct	20%	80%	0%	80%	20%	100%	20%	100%	80%	100%

Table 89: Summary of learning and testing for $n=4$.

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	196	729	24	774	294	301	321	511	327	409
Correct	40%	80%	0%	100%	40%	100%	0%	80%	80%	100%
First Ch. Correct	40%	80%	0%	100%	40%	100%	0%	80%	80%	100%

Table 90: Summary of learning and testing for $n=5$.

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	194	760	25	820	330	304	323	563	335	427
Correct	40%	60%	0%	100%	40%	100%	0%	80%	80%	100%
First Ch. Correct	40%	60%	0%	100%	40%	100%	0%	80%	80%	100%

Table 91: Summary of learning and testing for $n=6$.

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	189	783	24	823	341	300	323	593	335	431
Correct	20%	80%	0%	100%	40%	100%	20%	100%	80%	100%
First Ch. Correct	20%	80%	0%	100%	40%	100%	0%	100%	80%	100%

Table 92: Summary of learning and testing for $n=7$.

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	189	776	25	842	350	300	329	589	337	428
Correct	20%	80%	0%	80%	40%	100%	20%	100%	80%	100%
First Ch. Correct	20%	80%	0%	80%	40%	100%	0%	100%	80%	100%

Table 93: Summary of learning and testing for $n=8$.

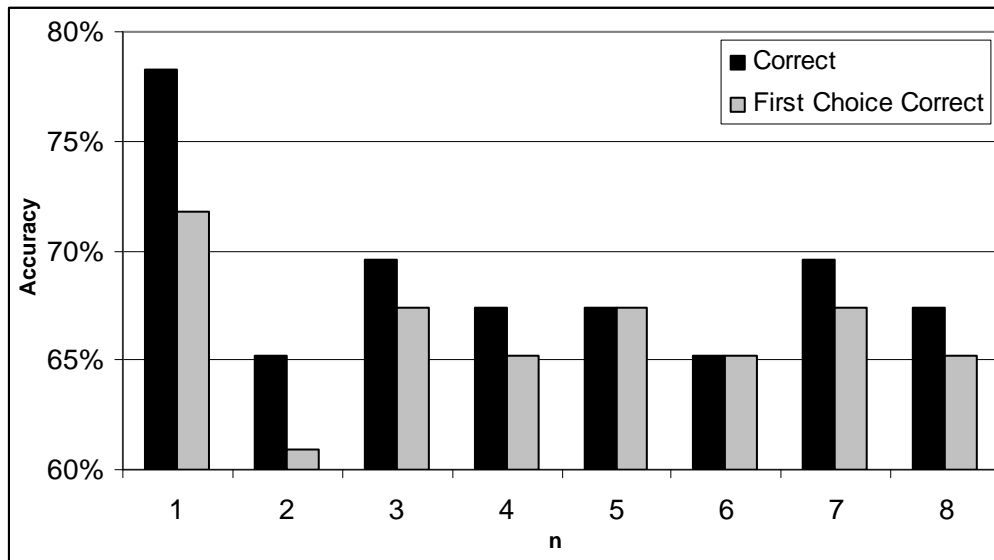


Figure 185: Number of correct and first choice correct answers for $n = 1-8$.

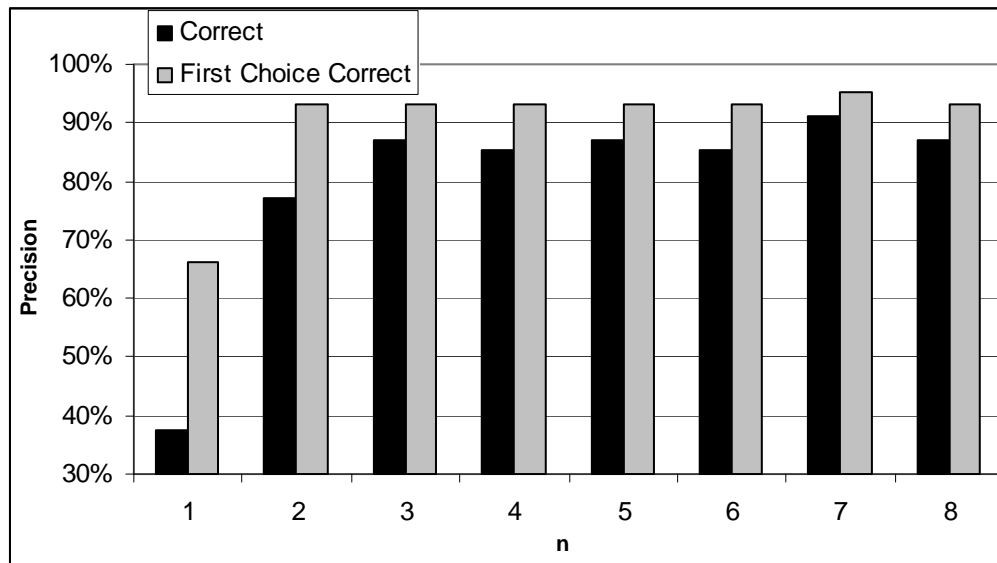


Figure 186: Precision and first choice precision for $n = 1-8$.

The charts and table show how accuracy and precision change for different values of n in $n \times k$ -grams.

The number of correct and first choice correct answers is clearly the highest for n equal to 1. The precision of these results is, however, very low. Because of that, it is reasonable to select n equal to 3 as the optimal n -gram size.

8.6 Experiments on 10 Users, All data

In this set of experiments we use all data available for 10 Users selected as in the previous experiments.

	# Learning Events	# Testing Events	# Total Episodes	# Correct	Accuracy	# 1st Choice	1st Choice Accuracy
User 1	21130	9097	106	96	91%	95	90%
User 2	4423	1861	21	10	48%	8	38%
User 3	1119	352	6	2	33%	2	33%
User 4	17990	7649	33	32	97%	32	97%
User 5	1457	879	9	2	22%	2	22%
User 7	15759	6757	51	48	94%	47	92%
User 8	11843	5088	52	50	96%	49	94%
User 12	2855	1310	15	2	13%	2	13%
User 19	6870	3097	35	31	89%	30	86%
User 25	14064	5929	25	24	96%	24	96%

Table 94: Summary of testing 4x6 grams for 10 selected users based on all data.

	# Learning Events	# Testing Events	# Total Episodes	# Correct	Accuracy	# 1st Choice	1st Choice Accuracy
User 1	394	454	5	2	40%	2	40%
User 2	1585	584	5	4	80%	4	80%
User 3	51	6	2	1	50%	1	50%
User 4	1675	414	5	4	80%	4	80%
User 5	503	444	5	0	0%	0	0%
User 7	584	203	5	5	100%	5	100%
User 8	515	586	5	1	20%	1	20%
User 12	1073	552	5	5	100%	5	100%
User 19	669	193	5	3	60%	3	60%
User 25	1992	703	5	5	100%	5	100%

Table 95: Summary of testing 4x6 grams for 10 selected users based on 10 training and 5 testing sessions.

8.7 Summary of Experimental Results

The experiments performed in this study represent only a subset of experiments that need to be done to sufficiently test the developed methods and determine optimal settings of parameters and modes operation of the learning and testing program in this area of application. These experiments also used only a relatively small subset of user data, because we wanted to facilitate a fair comparison our results with those obtained by researchers who used these same data, but different methods.

Despite these limitations, the experiments presented in this report show very promising results and brought some surprises. For example, it was very surprising that data filtering did not produce a noticeable improvement in performance. The filtering algorithms or the knowledge application applications may need further optimization for the kind of problem this data presents, and an analysis of the causes of the observed behavior is the subject of ongoing research.

The data preparation phase described in Section 5, and whose experimental results are presented as a part of Section 8 is critical for successful learning of users' models. This includes attribute and event selection and investigation of the "sausage" idea for data selection.

Correctly selected examples and representation spaces can be used by the AQ21 learning program to learn good models, like those presented in experiments 040606-1 and 040606-2. Relatively good results were also obtained from the multistate template model on unfiltered data (experiment 040615).

A sufficient similarity between sessions of the same user and a low similarity between sessions of different users is a necessary condition for successful learning of user models and correct recognition. Unfortunately, application of the methods described in Section 8.1 shows that many users behave similarly, and the task of discriminating among them is extremely difficult. Appropriate example selection can help achieving good results even in such situations; however, achieving high recognition of individual users is necessarily predicated upon a sufficient consistency in the given user's behavior, and a sufficient difference in the behavior of other users. What constitutes a sufficient consistency and sufficient difference depends on the required degree of certainty of correct recognition to be achieved

9 CONCLUSION AND FUTURE PLANS

This report described a wide range of ideas, methods and experimental results within the general theme of the LUS methodology developed for creating and testing user models for computer intrusion detection. The implementation and testing of all these ideas and methods goes far beyond the reported time period and the amount of work that can be done by the researchers supported in this research.

Therefore, we were able to conduct only a subset of experiments needed to be performed to explore and sufficiently test the developed methods, compare them, and determine optimal modes of operation and settings for parameters of the learning and testing program. Another limitation we experienced was that for some users we did not have sufficient data to allow us to get as good results as we obtained for the users for which the data was much more complete.

Despite these limitations, the experiments presented show very promising results and brought some positive surprises. One surprise was that learning rulesets using a simplicity-based method produced better results than those produced by learning based on statistical measures that require many runs through the entire dataset for each user. If this result holds consistently, it enables a significant speed-up in learning user models.

The results obtained demonstrate the feasibility of developing a useful and reliable computer intrusion detection system using the multistate template user model under the following conditions:

1. A sufficient amount of training data for each user is available.
2. The LUS-MT method is applied after appropriate target data preparation and parameter tuning.
3. There is sufficient similarity between the future user activity and the activity observed and represented in training data. What constitutes the “sufficient” similarity depends on the desired system performance, and has to be determined experimentally.

Future research needs to address more deeply the issues involved in satisfying the above conditions. Also, more datasets are needed to sufficiently test and evaluate the proposed methods.

In the future, we would like to work on further development, implementation and systematic testing, not only of the multiple-state template model, but also of other user models, such as the Prediction-based model, the Rule-Bayesian model, the Activity-based model, and combinations of the developed models.

This study of the LUS methodology and the obtained experimental results have opened many interesting and important topics for further investigation. These include:

1. Investigation of issues in satisfying the conditions for a successful application of the multistate template method to intrusion detection, specifically, determining the necessary size of the training sets, optimal procedures for target data preparation and parameter setting, and determining sufficient similarity between the training and testing data.
2. Investigation of the prediction-based, Rule-Bayesian and activity-based models and their comparison with the multistate template model.

3. Investigation of a multistrategy approach that would use the most desirable combination of the developed models.
4. Determination of the adequate size of the training and testing data streams for individual users using the “sausage” method.
5. Exploration of the utility of different methods for determining the most relevant attributes and events.
6. Application of the developed methodology to detect an inside intrusion.
7. Study of advantages of global vs. user-oriented attribute sets.
8. Investigation of new model-episode matching methods:
 - a. ATEST modifications, new evaluation methods
 - b. EPIC modifications, including the *count matches* method that classifies a possibly short series of events from an episode and then classifies the entire episode based on count of matches of such series; and the *select the best of the best* method that classifies a possibly short series of events from an episode and then classifies the entire episode based on the highest degree of match of one of the series.
9. Investigation of the sub-episode activity-based model that aggregates events into sub-episodes, and for each sub-episode computes activity-based user characteristics. This research would involve determining summaries of activities, and then applying the learning module to learn the user model from characteristics of the sub-episodes. Testing would be performed by a version of EPIC that classifies episodes based on the classification of its sub-episodes. The sub-episodes can be either of equal length (measured by time or number of events) or of lengths dependent on the user’s activity.
10. Extension of the knowledge representation power of AQ-learning.
11. Use of rule support estimates to speed up the learning process.

Finally, it should be noted that the LUS methodology is not limited only to problems of intrusion detection, but it could also be extended to a much wider class of problems concerning the analysis and modeling and characterization of temporal processes.

REFERENCES

- Baim, P., "The PROMISE Method For Selecting Most Relevant Attributes For Inductive Learning Systems," *Reports of the Intelligent Systems Group*, ISG 82-1, UIUCDCS-F-82-898, Department of Computer Science, University of Illinois, Urbana, September 1982.
- Bloedorn, E. and Michalski, R.S., "Data-Driven Constructive Induction," *IEEE Intelligent Systems*, Special issue on Feature Transformation and Subset Selection, pp. 30-37, March/April, 1998..
- Cichosz, P. "Systemy uczace sie," Wydawnictwa Naukowo-Techniczne, Warszawa, 2000
- Dufour, P. (eds.), *Current Issues in Expert Systems*, London: Academic Press Inc., pp. 125-158, 1987.
- Eskin, E., Arnold, A., Prerau, M., Portnoy, L. and Stolfo, S., "A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data," in Barbara, D. and Jajodia, S. (eds.), *Applications of Data Mining in Computer Security*, Kluwer, pp. 77-102, 2002.
- Goldring, T., "Recent Experiences with User Profiling for Windows NT," *Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection*, Baltimore, MD, 2002.
- Goldring, T., Shostak, J., Tessman, B. and Degenhardt, S., "User Profiling (Extended Abstract)," NSA unclassified internal report, 2000.
- Hätönen, K., Klemettinen, M., Mannila, H., Ronkainen, P. and Toivonen, H., "Knowledge discovery from Telecommunication Network Alarm Databases," *Proceedings of the Twelfth International Conference on Data Engineering (ICDE'96)*, New Orleans, LA, 1996.
- Hofmeyr, S., Forrest, S. and Somayaji, A., "Intrusion Detection using Sequences of System Calls," *Journal of Computer Security* 6, pp. 151-180, 1998
- Kaufman, K.A., "INLEN: A Methodology and Integrated System for Knowledge Discovery in Databases," Ph.D. Dissertation, School of Information Technology and Engineering, *Reports of the Machine Learning and Inference Laboratory*, MLI 97-15, George Mason University, Fairfax, VA, November, 1997.
- Kaufman, K.A. and Michalski, R.S., "Learning from Inconsistent and Noisy Data: The AQ18 Approach," *Proceedings of the Eleventh International Symposium on Methodologies for Intelligent Systems*, Warsaw, pp. 411-419, 1999.
- Kerber, R., "Chimerge: Discretization for Numeric Attributes," *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, AAAI Press, pp. 123-128, 1992.
- Lane, T. and Brodley, C.E., "Temporal Sequence Learning and Data Reduction for Anomaly Detection," *ACM Transactions on Information and System Security* 2, pp. 295-331, 1999.
- Michalski, R.S., "A Planar Geometrical Model for Representing Multi-Dimensional Discrete Spaces and Multiple-Valued Logic Functions," *Report No. 897*, Department of Computer Science, University of Illinois, Urbana, IL, January, 1978.

- Michalski, R.S., "ATTRIBUTIONAL CALCULUS: A Logic and Representation Language for Natural Induction," *Reports of the Machine Learning and Inference Laboratory*, MLI 04-2, George Mason University, Fairfax, VA, April, 2004.
- Michalski, R.S. and Kaufman K., "Learning Patterns in Noisy Data: The AQ Approach," in Paliouras, G., Karkaletsis, V. and Spyropoulos, C. (Eds.), *Machine Learning and its Applications*, Springer-Verlag, pp. 22-38, 2001.
- Michalski, R. S., Ko, H., and Chen, K., "Qualitative Prediction: the SPARC/G Methodology for Inductively Describing and Predicting Discrete Processes," Van Lamsweerde, A. and
- Mukkamala, S. and Sung, A., "Comparison of Neural Networks and Support Vector Machines in Intrusion Detection," *Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection*, Baltimore, MD, 2002.
- Novak, J., Stark, V. and Heinbuch, D., "Zombie Scan," *Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection*, Baltimore, MD, 2002.
- Reinke, R., "Knowledge Acquisition and Refinement Tools for the ADVISE Meta-Expert System," Master's Thesis, *Reports of the Intelligent Systems Group*, ISG 84-5, UIUCDCS-F-84-921, Department of Computer Science, University of Illinois, Urbana, 1984.
- Quinlan, J. R., "C4.5 Systems for Machine Learning," Morgan Kaufmann Publishers Inc., 1993
- Schonlau, M. and Theus, M., "Detecting Masquerades in Intrusion Detection Based on Unpopular Commands," *Information Processing Letters* 76, pp. 33-38, 2000.
- Scott, S., "A Bayesian Paradigm for Designing Intrusion Detection Systems," *Computational Statistics and Data Analysis*, 45, pp. 69-83, 2004.
- Shah, K., Jonckheere, E. and Bohacek, S., "Detecting Network Attacks through Traffic Modeling," *Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection*, Baltimore, MD, 2002.
- Snieszynski, B., Szymacha, R. And Michalski, R.S., "Knowledge Visualization Using Optimized General Logic Diagrams," *Proceedings of the Fourteenth International Symposium on Intelligent Information Systems*, Gdansk, Poland, 2005 (to appear).
- Streilein, W.W., Cunningham, R.K. and Webster, S.E., "Improved Detection of Low-Profile Probe and Novel Denial-of Service Attacks," *Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection*, Baltimore, MD, 2002.
- Valdes, A., "Profile Based Intrusion Detection: Lessons Learned, New Directions," *Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection*, Baltimore, MD, 2002.
- Valdes, A. and Skinner, K., "Adaptive, Model-based Monitoring for Cyber Attack Detection," in Debar, H., Me, L. and Wu, F. (eds.), *Lecture Notes in Computer Science #1907 (from Recent Advances in Intrusion Detection, RAID-2000)*, Springer-Verlag, 2000.
- Wnek, J., "DIAV 2.0 User Manual: Specification and Guide through the Diagrammatic Visualization System," *Reports of the Machine Learning and Inference Laboratory*, MLI 95-5, George Mason University, Fairfax, VA, 1995.

Wojtusiak, J., “AQ21 User’s Guide,” *Reports of the Machine Learning and Inference Laboratory*, MLI 04-3, George Mason University, Fairfax, VA, 2004.

Zhang, Q., “Knowledge Visualizer: A Software System for Visualizing Data, Patterns and Their Relationships,” *Reports of the Machine Learning and Inference Laboratory*, MLI 97-14, George Mason University, Fairfax, VA, 1997.

APPENDIX A: DICTIONARY OF LUS METHODOLOGY TERMS

Note: Terms typed in bold in the body of a definition have a separate definition in the dictionary.

episode: A **window**, or a set of non-overlapping **windows**.

episode similarity: A measure of similarity between two episodes, usually between a **training episode** and **testing episode** for a given user. It is used for estimating the chance for a learned user model to classify the **testing episode well**. One measure of episode similarity is the ratio p_{TE}/P_{TE} , where p_{TE} is the total number of (not necessarily distinct) events in the testing episode that also occur in the training episode, and P_{TE} is the total number of events in the testing episode.

episode distinctiveness: A measure of dissimilarity between a databag (a set of not necessarily distinct **events**) derived from a given **user's episode** and the databag derived from the windows other users' episodes. One way to measure such distinctiveness is $p/p+n$, where p is the total number of occurrences of (not necessarily distinct) significant events in the union of the **training and testing bags**, and n is the total number of occurrences of the same events in the training and testing bags of other users.

event: A description of a user behavior during a given time interval or at a given time instance. An event describing the behavior during a time interval is in the form of a sequence of values of a single attribute at consecutive time instances (case 1: single-attribute, multiple-time instances), or in the form of a conjunction of values of different attributes (case2: multi-attribute, single time instance). In the **multistate template model**, an event is a single n -gram involving one or several attributes. In the **prediction-based model**, an event is in the form of a **ground implication**, **A --> B**, where **A** is a sequence of consecutive events preceding a given time moment, and **B** a sequence of consecutive events following this time moment.

event significance: A measure of significance of an event in a **training set**. One measure of significance is $p^2/(p+n)$, where p is the number of examples of the target user's behavior matching the event, and n is the number of examples of other users' behavior matching it.

ground implication: An event in the form of a sequence of descriptions of the premise states of the length, called lookback, followed by a description of the states following the premise states of the length, called look-forward.

multi-event: A description of a user behavior in a given period of time using more than one attribute. A conjunctive multi-event is a conjunction of multi-attribute events characterizing behavior at different instances of time within this interval.

multistate template model: A user model in which a user's activities are represented by templates characterizing the activities at a small number of consecutive instances in time. These templates are based on $n \times k$ -grams.

n -gram: An n -gram is an event characterizing states of the user behavior during n time instances by values of a single attribute. It is in the form of a sequence of n attribute values representing values of this attribute in consecutive time instances.

nxk -gram: An *n -gram* in which the state in each time instance is described not by one, but by k attributes. The value of n is called the *length* and the value of k is called the *scope* of the nxk -gram.

prediction-based model: A user model in which user behavior is characterized by rules that predict future behavior when certain past behaviors have been observed.

testing episode: An **episode** used for testing a model of the behavior of a user or a group of users. A testing episode is evaluated and classified as a whole.

testing bag (TE): A bag of events describing a **testing episode** / a set of **testing episodes** to be assigned a classification decision/s

training episode: An **episode** used for developing a model of the behavior of a user or a group of users.

training session: A period of user activity from login to logout, or a set of **events** describing such a period.

training bag (TR): A set of events describing a **training episode** (used for learning a model of the behavior of a group of users)

user databag: A bag of all events in the data collected from observing a given user; it is the union of the **training** and **testing bags**.

window: A period of time during which the behavior of a user is being measured.

window size: A measure of maximum number of consecutive events in the raw data that can be combined to form a multistate event.

APPENDIX B: DESCRIPTION OF ATTRIBUTES

1 Host machine

Extracted from raw file name

Computed as substring of input file name, delimited by "-"

Possible values are: host*, where * is 1,2,3,4,5,6,7,8,9,11,12,13,14,15,16,17,18,19,20,21

Program variable name: host

This attribute has the same value for all records from the same session.

2 Day of week

Extracted from raw file name, and based in part on input variable "delta t (seconds) since login"

Computed as function of the value taken from the input record

Possible values are: Mon, Tue, Wed, Thu, Fri, Sat, Sun

Program variable name: day_of_week

3 Time of day(hour)

Extracted from raw file name, and based in part on input variable "delta t (seconds) since login"

Computed as function of the value taken from the input record

Possible values are: 00, 01, ... , 23

Program variable name: time_of_day

4 Number of seconds from the start of the session

Extracted from raw data process or window records, attribute delta_t.

Computed as value taken from the input record rounded to nearest second.

Program variables names: wdeltat_r, pdeltat_r

5 (Window) Process name

Extracted from raw data process or window records, process_name

Computed as value taken from the input record

Program variable name: window_process_name, process_name

.ls files filter out process records not corresponding to current window record.

Possible values are:

cmd config32 csrss dreamweaver drwtsn32 emacs eqnedt32 excel explorer fastboot
findfast fpxpress grpconv gsvie32 icwconn1 ie501dom ie5setup ie5wzd iexP iexplore
iexplorP ikernel installroot keyhh keyview kmi2000 loadwc lsass mapisp32 mcshield
monitor mouseworks mplayer2 msaccess msieexec msieexecP msimn msnt128 msoffice msohelp
mspaint musrmgr netP netscape netscapeP netscp6 netsP neW notepad ntvdm
o2ksr1adl odpusr32 oemig50 oP osa9 out128 outlook outP packet2k pbupdate
perfmon perlbu~1 perlbuilder photoed photoshop pid powerP powerpnt powP pstores
quikview rasphone rauninst realoneplayergo realplay regedit rndal rundll32 scan32 sdstat
services setup shstat smsapm32 smsmon32 smswiz32 sndvol32 spoolss sqlmangr system
tabletservice talkback taskmgr telnet uninstd update visio32 vsstat wanging welcome
windisk winfile winfile2 winhlp32 winmsd winproj winword winzip32 wordpad wscript
wuser32 xemacs

6 Window or Process status

Extracted from raw data status attribute

Based on input variable "status of the process: a(background), b(birth), c(continuation), d(death)"

Computed as value taken from the input record or one of the extra values described below

Program variable name: status

For process-type input records in .1s data, the set of possible values are as follows:

(background process records "a" are discarded, some birth process records "b" are discarded, "d"eath records do not occur)

b - record indicating the birth of a process

c - record indicating the continuation of the process

For window-type input records the set of possible values are following (definition is extended to window-type records):

n - means that the window-type input record indicates a new window process

o - means that the window-type input record indicates an old window process

7 CPU time accrued by process

Extracted from raw process data cpu attribute or N/A for a window record

Corresponds to attribute B4 in (4)

Computed as value taken from the input record rounded to nearest second.

Program variable name: cpu_time_accrued_r

8 Process inactive time

Derived from raw process data delta-t attribute or N/A for a window record

Based on input variable "delta t (seconds) since login"

Usually computed as difference between values taken from input variable or one of the extra values described below

If there is only one process-type input records for a given window unit the output value is "0"

If it is window-type input record the output value is N/A

Program variable name: inactive

9 Natural logarithm of process inactive time

Derived from raw process data delta-t attribute or N/A for a window record

Based on attribute 8

Computed as $\ln(1 + \text{attribute } 8)$ or is N/A if attribute 8. is N/A.

Program variable name: logf_inactive

10 Flag indicating process inactive time > 1 minute

Derived from raw process data delta-t attribute or N/A for a window record

Based on attribute 8

Computed as string "long" if the value of attribute 8 > 60, or one of the extra values described below

For window-type input recordd the output value is N/A

If there is only one process-type input record for a given window unit, the value is "lte60"

Program variable name: long_inactive

11 Window process ID*Extracted from raw window data pid attribute*

Program variable name: windows_pid

12 Window name (title) number based on lusprep-titles.txt file*Derived from raw window title attribute*

Based on input variable "window title as it appears in the window's title bar"

Program variable name: window_title

13 CPU time accrued by process within window*Derived from raw process data cpu attribute or N/A for a window record*

Computed as function of values taken from the input records or one of the extra values described below

For process records following the first in a window record, the value in the previous process record is subtracted from the value in the current process record

The result of this subtraction is rounded to the nearest second.

If there is only one process record the value is "0"; if there are none, the value is N/A.

Program variable name: window_time_accrued_r

14 Natural logarithm of CPU time accrued by process within window*Derived from raw process data cpu attribute or N/A for a window record*Computed as $\ln(1 + \text{attribute } 13)$ or N/A if attribute 13. is N/A.

Program variable name: logf_window_time_accrued_r

15 Total elapsed time in window*Derived from raw delta-t attribute*

Computed as difference of delta-t values in next window record and in previous window-type record (or 0 if none)

Program variable name: total_elapsed

16 Natural logarithm of Total elapsed time in window*Derived from raw delta-t attribute*Computed as $\ln(1 + \text{attribute } 15)$

Program variable name: logf_total_elapsed

17 Ratio of CPU time accrued by process within window to Total elapsed time in window*Derived from raw process data cpu and delta-t attributes*Computed as $\text{int}(100 * \text{attribute } 13) / (\text{attribute } 15 + 0.001) / 100$, (attribute 15 can be 0)

Program variable name: accrued2elapsed_ratio

18 Delta time between window titles whenever new window is opened*Derived from raw delta-t attribute*

Related to attributes 4, 6 and 15

Computed as difference between values taken for subsequent new window-type input records.

For the first window record in the input file the output value is "0"

Program variable name: delta_window

Auxiliary variable names: old_delta_new_window, old_delta_old_window, first_old

19 Natural logarithm of Delta time between window titles whenever *new* window is opened

Derived from raw process data delta-t attribute

Computed as $\ln(1 + \text{attribute } 18)$

Program variable name: logf_delta_window

20 Elapsed time since login whenever *new* window is opened

Derived from raw data delta-t attribute

Related to attributes 6 and 7.

Program variable name: elapsed_new_window

21 Natural logarithm of Elapsed time since login whenever *new* window is opened

Derived from raw data delta-t attribute

Computed as $\ln(1 + \text{attribute } 20)$

Program variable name: logf_elapsed_new_window

22 Number of characters in protected words

Derived from raw window title attribute

Program variable name: prot_chars

23 Number of characters in protected words / total number of characters in window title

Derived from raw window title attribute

Program variable name: prot2total_chars_ratio

24 Total number of words in window title

Derived from raw window title attribute

Program variable name: total_words

25 Ratio of Number of protected words / Total number of words in window title

Derived from raw window title attribute

Program variable name: prot2total_words_ratio

26 Number of process-level records in a single window unit

Derived from raw data record ordering

Program variable name: process_records

27 Natural logarithm of Number of process-level records in a single window unit

Derived from raw data record ordering

Computed as $\ln(1 + \text{attribute } 26)$

Program variable name: logf_process_records

28 Total number of windows opened

Derived from raw data record ordering

Program variable name: windows_count

29 Natural logarithm of Total number of windows opened

Derived from raw data record ordering

Computed as $\ln(1 + \text{attribute } 28)$

Program variable name: logf_windows_count

30 Number of protected words in window title

Derived from raw window title attribute

Program variable name: prot_words

31 Number of sanitized words in window title

Derived from raw window title attribute

Program variable name: san_words

APPENDIX C: SELECTED MT USER MODELS

This appendix presents selected multistate templates models learned using AQ21 system. For some of the models we present only selected rules since in number of cases their total number is very large. Presented models correspond to experiments presented in sections 7.4 and 7.5.

C1 Experiment 040607-1: Filtered Data TR+TS, Discriminant Descriptions

Training Dataset:

Discretization: Dis-3

Filtering: Significance based, conjunctive, rank-threshold = 10, TR+TS

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

AQ21 Learning Parameters:

maxstar = 1 maxrule = 1 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Discriminant descriptions

Learning Results:

Total number of rules: 71

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	8	8	1	10	8	6	8	9	8	5

Learned Models:

```
Output_Hypotheses User1
{
  # -- This learning took =
  # -- System (CPU) time  = 0.12
  # -- User (Total) time  = 0
  # -- Number of rules in the cover = 8
  # -- Number of conditions      = 25
  # -- Complexity for this cover  = 181
  # -- Average number of rules kept from each stars = 1
  # -- Uncovered Positives = 0

  positive_events      = 522
  negative_events      = 53733
  positive_distinct_events = 28
  negative_distinct_events = 513

[user=user1]
  # Rule 1
  <-- [process_name=explorer,outlook : 394,20140]
      [proc_count_in_win_1f=from2d01267to3d51143 : 377,21048]
      [win_title_prot_words=3 : 269,14005]
      : p=160,np=98,u=98,cx=23,c=1,s=160 # 1257
```

```

# Rule 2
<-- [process_name=explorer,ntvdm : 148,629]
[proc_count_in_win_lf=from0to2d01267..from2d01267to3d51143 :
470,23150]
[win_title_prot_words=0..1 : 195,37455]
: p=86,np=86,u=86,cx=23,s=86 # 1256

# Rule 3
<-- [process_name=explorer,iexplore : 176,8290]
[proc_count_in_win_lf=from2d01267to3d51143 : 377,21048]
[win_title_prot_words=3 : 269,14005]
[win_title_prot_words-3=3 : 269,14007]
: p=140,np=78,u=78,cx=30,c=1.75,s=140 # 1260

# Rule 4
<-- [process_name=outlook : 296,19511]
[proc_count_in_win_lf=from2d01267to3d51143 : 377,21048]
[win_title_prot_words=1 : 145,37350]
: p=57,np=57,u=57,cx=21,s=57 # 1255

# Rule 5
<-- [prot_words_chars=from0to7d5 : 109,12502]
[proc_count_in_win_lf=from3d51143to4d64917 : 52,17792]
: p=52,np=52,u=52,cx=14,s=52 # 1259

# Rule 6
<-- [process_name=outlook : 296,19511]
[proc_count_in_win_lf=from2d01267to3d51143 : 377,21048]
[win_title_prot_words=2 : 58,2058]
: p=32,np=32,u=32,cx=21,s=32 # 1262

# Rule 7
<-- [process_name=outlook : 296,19511]
[process_name-2=outlook : 296,19513]
[proc_count_in_win_lf=from0to2d01267 : 93,2102]
[win_title_prot_words=3 : 269,14005]
: p=31,np=31,u=31,cx=28,c=1.5,s=31 # 1258

# Rule 8
<-- [process_name=outlook : 296,19511]
[proc_count_in_win_lf=from0to2d01267 : 93,2102]
[win_title_prot_words=2 : 58,2058]
: p=26,np=26,u=26,cx=21,c=1,s=26 # 1261

}

```

Output_Hypotheses User2

```

{
# -- This learning took =
# -- System (CPU) time = 0.12
# -- User (Total) time = 0
# -- Number of rules in the cover = 8
# -- Number of conditions = 27
# -- Complexity for this cover = 191
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 4785
negative_events = 49470
positive_distinct_events = 46

```

```

negative_distinct_events = 495
[user=user2]
  # Rule 1
  <-- [process_name=netscape : 3083,22218]
      [prot_words_chars=from7d5to8d5 : 3083,21559]
      [proc_count_in_win_lf=from2d01267to3d51143 : 2577,18848]
      [proc_count_in_win_lf-2=from2d01267to3d51143 : 2577,18878]
      [proc_count_in_win_lf-3=from2d01267to3d51143 : 2577,18760]
      [win_title_prot_words=1 : 3527,33968]
      : p=1731,np=1731,u=1731,cx=42,c=1.83,s=1.73e+03 # 2605

  # Rule 2
  <-- [process_name=netscape : 3083,22218]
      [prot_words_chars=from7d5to8d5 : 3083,21559]
      [proc_count_in_win_lf=from3d51143to4d64917 : 1877,15967]
      : p=1352,np=1352,u=1352,cx=21,s=1.35e+03 # 2604

  # Rule 3
  <-- [process_name=outlook : 849,18958]
      [proc_count_in_win_lf=from2d01267to3d51143 : 2577,18848]
      [win_title_prot_words=3 : 611,13663]
      : p=611,np=611,u=611,cx=21,s=611 # 2607

  # Rule 4
  <-- [process_name=winword : 585,41]
      [prot_words_chars=from8d5to24 : 1402,7701]
      [proc_count_in_win_lf=from2d01267to3d51143..from3d51143to4d64917 :
4454,34815]
      [win_title_prot_words=2 : 585,1531]
      : p=460,np=460,u=173,cx=28,c=1,s=460 # 2606

  # Rule 5
  <-- [process_name=outlook,winword : 1434,18999]
      [proc_count_in_win_lf=from3d51143to4d64917 : 1877,15967]
      [win_title_prot_words=1..2 : 4112,35499]
      : p=525,np=238,u=238,cx=23,s=525 # 2609

  # Rule 6
  <-- [process_name=explorer : 206,521]
      [win_title_prot_words=1 : 3527,33968]
      : p=206,np=206,u=206,cx=14,s=206 # 2608

  # Rule 7
  <-- [process_name=winword : 585,41]
      [proc_count_in_win_lf=from3d51143to4d64917..from4d64917to5d56641 :
2002,25161]
      : p=412,np=125,u=125,cx=14,s=412 # 2610

  # Rule 8
  <-- [process_name=iexplore : 62,7677]
      [prot_words_chars=from25d5to50 : 62,153]
      [proc_count_in_win_lf=from2d01267to3d51143 : 2577,18848]
      [proc_count_in_win_lf-2=from2d01267to3d51143 : 2577,18878]
      : p=62,np=62,u=62,cx=28,c=1.5,s=62 # 2603
}

Output_Hypotheses User3
{
  # -- This learning took =
  # -- System (CPU) time = 0.06
  # -- User (Total) time = 0
  # -- Number of rules in the cover = 1

```

```

# -- Number of conditions          = 1
# -- Complexity for this cover    = 7
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events          = 5
negative_events          = 54250
[user=user3]
# Rule 1
<-- [proc_count_in_win_1f=1te0 : 5,0]
    : p=5,np=5,ep=5,n=0,en=0,u=5,cx=7,s=4 # 2790
}

Output_Hypotheses User4
{
# -- This learning took =
# -- System (CPU) time  = 0.14
# -- User (Total) time  = 0
# -- Number of rules in the cover = 10
# -- Number of conditions          = 33
# -- Complexity for this cover    = 231
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events          = 18396
negative_events          = 35859
positive_distinct_events = 73
negative_distinct_events = 468
[user=user4]
# Rule 1
<-- [process_name=netscape : 11570,13731]
    [proc_count_in_win_1f=from3d51143to4d64917 : 6324,11520]
    [win_title_prot_words=1 : 17564,19931]
    : p=5027,np=5027,u=5027,cx=21,s=5.03e+03 # 4241

# Rule 2
<-- [process_name=netscape : 11570,13731]
    [proc_count_in_win_1f=from2d01267to3d51143 : 6317,15108]
    [proc_count_in_win_1f-2=from2d01267to3d51143 : 6317,15138]
    [proc_count_in_win_1f-3=from2d01267to3d51143 : 6317,15020]
    [win_title_prot_words=1 : 17564,19931]
    : p=4248,np=4248,u=4248,cx=35,s=4.25e+03 # 4243

# Rule 3
<-- [proc_count_in_win_1f=from5d56641to7d12078 : 2387,1080]
    [win_title_prot_words=1 : 17564,19931]
    : p=2387,np=2387,u=2387,cx=14,s=2.39e+03 # 4244

# Rule 4
<-- [process_name=netscape : 11570,13731]
    [proc_count_in_win_1f=from4d64917to5d56641 : 3103,6216]
    [win_title_prot_words=1 : 17564,19931]
    : p=2295,np=2295,u=2295,cx=21,c=1,s=2.3e+03 # 4248

# Rule 5
<-- [process_name=outlook : 6826,12981]
    [proc_count_in_win_1f=from3d51143to4d64917 : 6324,11520]
    [win_title_prot_words=1 : 17564,19931]
    : p=1297,np=1297,u=1297,cx=21,c=1,s=1.3e+03 # 4239

# Rule 6
<-- [process_name=outlook : 6826,12981]

```



```

[proc_count_in_win_lf=from2d01267to3d51143 : 6317,15108]
[win_title_prot_words=1 : 17564,19931]
: p=1237,np=1237,u=1237,cx=21,s=1.24e+03 # 4245

# Rule 7
<-- [process_name=outlook : 6826,12981]
[proc_count_in_win_lf=from4d64917to5d56641 : 3103,6216]
[win_title_prot_words=1 : 17564,19931]
: p=808,np=808,u=808,cx=21,c=1,s=808 # 4242

# Rule 8
<-- [process_name=outlook : 6826,12981]
[prot_words_chars=from8d5to24 : 832,8271]
[proc_count_in_win_lf=from2d01267to3d51143 : 6317,15108]
[win_title_prot_words=3 : 639,13635]
: p=639,np=639,u=639,cx=28,s=639 # 4247

# Rule 9
<-- [process_name=outlook : 6826,12981]
[proc_count_in_win_lf=from0to2d01267 : 265,1930]
[win_title_prot_words=1 : 17564,19931]
: p=265,np=265,u=265,cx=21,s=265 # 4240

# Rule 10
<-- [process_name=outlook : 6826,12981]
[prot_words_chars=from8d5to24 : 832,8271]
[proc_count_in_win_lf=from2d01267to3d51143 : 6317,15108]
[win_title_prot_words=2 : 193,1923]
: p=193,np=193,u=193,cx=28,s=193 # 4246
}

Output_Hypotheses User5
{
# -- This learning took =
# -- System (CPU) time = 0.11
# -- User (Total) time = 1
# -- Number of rules in the cover = 8
# -- Number of conditions = 19
# -- Complexity for this cover = 133
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 5056
negative_events = 49199
positive_distinct_events = 61
negative_distinct_events = 480
[user=user5]
# Rule 1
<-- [prot_words_chars=from7d5to8d5 : 4065,20577]
[proc_count_in_win_lf=from2d01267to3d51143 : 1944,19481]
: p=1642,np=1574,u=1642,cx=14,s=1.64e+03 # 5219

# Rule 2
<-- [prot_words_chars=from7d5to8d5 : 4065,20577]
[proc_count_in_win_lf=from3d51143to4d64917 : 1683,16161]
: p=1126,np=1126,u=1126,cx=14,s=1.13e+03 # 5214

# Rule 3
<-- [prot_words_chars=from7d5to8d5 : 4065,20577]
[proc_count_in_win_lf=from4d64917to5d56641 : 1123,8196]
: p=991,np=991,u=991,cx=14,s=991 # 5220

```

```

# Rule 4
<-- [process_name=outlook : 991,18816]
    [proc_count_in_win_lf=from3d51143to4d64917 : 1683,16161]
    [win_title_prot_words=3 : 654,13620]
      : p=352,np=352,u=352,cx=21,c=1,s=352 # 5218

# Rule 5
<-- [prot_words_chars=from7d5to8d5 : 4065,20577]
    [proc_count_in_win_lf=from0to2d01267 : 306,1889]
      : p=306,np=306,u=306,cx=14,s=306 # 5216

# Rule 6
<-- [process_name=outlook : 991,18816]
    [prot_words_chars=from8d5to24 : 654,8449]
    [proc_count_in_win_lf=from2d01267to3d51143 : 1944,19481]
    [win_title_prot_words=3 : 654,13620]
      : p=302,np=302,u=302,cx=28,s=302 # 5215

# Rule 7
<-- [prot_words_chars=from0to7d5 : 337,12274]
    [proc_count_in_win_lf=from3d51143to4d64917 : 1683,16161]
      : p=205,np=205,u=205,cx=14,s=205 # 5217

# Rule 8
<-- [prot_words_chars=from0to7d5 : 337,12274]
    [proc_count_in_win_lf=from4d64917to5d56641 : 1123,8196]
      : p=132,np=132,u=132,cx=14,s=132 # 5221
}

```

Output_Hypotheses User7

```

{
# -- This learning took =
# -- System (CPU) time = 0.1
# -- User (Total) time = 0
# -- Number of rules in the cover = 6
# -- Number of conditions = 17
# -- Complexity for this cover = 119
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 1556
negative_events = 52699
positive_distinct_events = 73
negative_distinct_events = 468
[user=user7]
# Rule 1
<-- [prot_words_chars=from0to7d5 : 1092,11519]
    [proc_count_in_win_lf=from4d64917to5d56641 : 630,8689]
      : p=630,np=630,u=630,cx=14,s=630 # 5838

# Rule 2
<-- [prot_words_chars=from0to7d5 : 1092,11519]
    [proc_count_in_win_lf=from2d01267to3d51143 : 555,20870]
      : p=326,np=326,u=326,cx=14,s=326 # 5837

# Rule 3
<-- [process_name=explorer : 263,464]
    [proc_count_in_win_lf=from0to2d01267..from2d01267to3d51143 :
926,22694]
    [win_title_prot_words=2 : 263,1853]
      : p=263,np=263,u=263,cx=21,s=263 # 5833

```

```

# Rule 4
<-- [process_name=outlook : 1293,18514]
    [proc_count_in_win_lf=from0to2d01267 : 371,1824]
    [win_title_prot_words-2=1 : 1092,36403]
    : p=136,np=136,u=136,cx=21,s=136 # 5834

# Rule 5
<-- [process_name=outlook : 1293,18514]
    [prot_words_chars=from8d5to24 : 394,8709]
    [proc_count_in_win_lf=from0to2d01267 : 371,1824]
    [win_title_prot_words=3 : 131,14143]
    : p=131,np=131,u=131,cx=28,c=1,s=131 # 5836

# Rule 6
<-- [process_name=outlook : 1293,18514]
    [prot_words_chars=lte0 : 70,85]
    [proc_count_in_win_lf=from2d01267to3d51143 : 555,20870]
    : p=70,np=70,u=70,cx=21,s=70 # 5835
}

Output_Hypotheses User8
{
# -- This learning took =
# -- System (CPU) time = 0.13
# -- User (Total) time = 0
# -- Number of rules in the cover = 8
# -- Number of conditions = 26
# -- Complexity for this cover = 182
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 3249
negative_events = 51006
positive_distinct_events = 50
negative_distinct_events = 491
[user=user8]
# Rule 1
<-- [process_name=netscape : 2769,22532]
    [proc_count_in_win_lf=from3d51143to4d64917 : 1294,16550]
    : p=1294,np=1294,u=1294,cx=14,s=1.29e+03 # 7251

# Rule 2
<-- [process_name=netscape : 2769,22532]
    [proc_count_in_win_lf=from2d01267to3d51143 : 1430,19995]
    [proc_count_in_win_lf-2=from2d01267to3d51143 : 1430,20025]
    [proc_count_in_win_lf-3=from2d01267to3d51143 : 1430,19907]
    [win_opened=lte16 : 3249,50975]
    : p=1037,np=574,u=1037,cx=35,s=1.04e+03 # 7252

# Rule 3
<-- [process_name=netscape : 2769,22532]
    [proc_count_in_win_lf=from4d64917to5d56641 : 325,8994]
    [win_title_prot_words=1 : 2287,35208]
    : p=325,np=325,u=325,cx=21,c=1,s=325 # 7254

# Rule 4
<-- [process_name=outlook : 439,19368]
    [prot_words_chars=from8d5to24 : 962,8141]
    [proc_count_in_win_lf=from2d01267to3d51143 : 1430,19995]
    [win_title_prot_words=3 : 262,14012]

```

```

      : p=262,np=262,u=262,cx=28,s=262 # 7247

# Rule 5
<-- [process_name=netscape : 2769,22532]
    [proc_count_in_win_lf=from0to2d01267 : 200,1995]
    [win_title_prot_words=1 : 2287,35208]
      : p=113,np=113,u=113,cx=21,s=113 # 7248

# Rule 6
<-- [process_name=outlook : 439,19368]
    [proc_count_in_win_lf=from2d01267to3d51143 : 1430,19995]
    [win_title_prot_words=1 : 2287,35208]
      : p=90,np=90,u=90,cx=21,s=90 # 7250

# Rule 7
<-- [process_name=outlook : 439,19368]
    [proc_count_in_win_lf=from0to2d01267 : 200,1995]
    [win_title_prot_words=1 : 2287,35208]
      : p=87,np=87,u=87,cx=21,s=87 # 7249

# Rule 8
<-- [process_name=winword : 41,585]
    [proc_count_in_win_lf=from2d01267to3d51143 : 1430,19995]
    [win_opened=ltel6 : 3249,50975]
      : p=41,np=41,u=41,cx=21,s=41 # 7253
}

Output_Hypotheses User12
{
# -- This learning took =
# -- System (CPU) time = 0.13
# -- User (Total) time = 0
# -- Number of rules in the cover = 9
# -- Number of conditions = 32
# -- Complexity for this cover = 226
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 5920
negative_events = 48335
positive_distinct_events = 70
negative_distinct_events = 471
[user=user12]
# Rule 1
<-- [prot_words_chars=from7d5to8d5 : 3524,21118]
    [proc_count_in_win_lf=from2d01267to3d51143 : 3167,18258]
    [proc_count_in_win_lf-2=from2d01267to3d51143 : 3167,18288]
    [proc_count_in_win_lf-3=from2d01267to3d51143 : 3167,18170]
      : p=1948,np=1948,u=1948,cx=28,s=1.95e+03 # 8520

# Rule 2
<-- [process_name=netscape : 3524,21777]
    [proc_count_in_win_lf=from3d51143to4d64917 : 1799,16045]
    [win_title_prot_words=1 : 4255,33240]
      : p=955,np=955,u=955,cx=21,s=955 # 8524

# Rule 3
<-- [process_name=explorer,netscape : 3684,22344]
    [prot_words_chars=from7d5to8d5..from8d5to24 : 5189,28556]
    [proc_count_in_win_lf=from4d64917to5d56641 : 781,8538]
    [win_title_prot_words=1..2 : 4463,35148]
      : p=781,np=781,u=781,cx=30,s=781 # 8519

```

```

# Rule 4
<-- [process_name=outlook : 2236,17571]
    [prot_words_chars=from8d5to24 : 1665,7438]
    [proc_count_in_win_lf=from2d01267to3d51143 : 3167,18258]
    [win_title_prot_words=3 : 1457,12817]
    : p=742,np=742,u=742,cx=28,s=742 # 8521

# Rule 5
<-- [process_name=outlook : 2236,17571]
    [prot_words_chars=from8d5to24 : 1665,7438]
    [proc_count_in_win_lf=from3d51143to4d64917 : 1799,16045]
    : p=590,np=590,u=590,cx=21,s=590 # 8523

# Rule 6
<-- [process_name=outlook : 2236,17571]
    [proc_count_in_win_lf=from2d01267to3d51143 : 3167,18258]
    [win_title_prot_words=1 : 4255,33240]
    : p=477,np=477,u=477,cx=21,s=477 # 8525

# Rule 7
<-- [process_name=outlook : 2236,17571]
    [proc_count_in_win_lf=from3d51143to4d64917 : 1799,16045]
    [win_title_prot_words=1 : 4255,33240]
    : p=254,np=254,u=254,cx=21,c=1,s=254 # 8526

# Rule 8
<-- [process_name-2=outlook : 2236,17573]
    [prot_words_chars=from8d5to24 : 1665,7438]
    [proc_count_in_win_lf=from0to2d01267 : 173,2022]
    [win_title_prot_words=3 : 1457,12817]
    : p=125,np=125,u=125,cx=28,c=1.5,s=125 # 8518

# Rule 9
<-- [process_name=outlook : 2236,17571]
    [prot_words_chars=from8d5to24 : 1665,7438]
    [proc_count_in_win_lf=from0to2d01267 : 173,2022]
    [win_title_prot_words=2 : 208,1908]
    : p=48,np=48,u=48,cx=28,s=48 # 8522
}

Output_Hypotheses User19
{
# -- This learning took =
# -- System (CPU) time = 0.12
# -- User (Total) time = 0
# -- Number of rules in the cover = 8
# -- Number of conditions = 26
# -- Complexity for this cover = 182
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 5240
negative_events = 49015
positive_distinct_events = 44
negative_distinct_events = 497
[user=user19]
# Rule 1
<-- [process_name=outlook : 4950,14857]
    [prot_words_chars=from8d5to24 : 2324,6779]
    [proc_count_in_win_lf=from3d51143to4d64917..from4d64917to5d56641 :
3249,23914]

```

```

      : p=1857,np=1857,u=1857,cx=21,s=1.86e+03 # 9638

# Rule 2
<-- [proc_count_in_win_1f=from5d56641to7d12078 : 1080,2387]
    [win_title_prot_words=1 : 2881,34614]
      : p=1080,np=1080,u=1080,cx=14,c=1,s=1.08e+03 # 9643

# Rule 3
<-- [process_name=outlook : 4950,14857]
    [proc_count_in_win_1f=from4d64917to5d56641 : 1952,7367]
    [win_title_prot_words=1 : 2881,34614]
      : p=797,np=797,u=797,cx=21,c=1,s=797 # 9642

# Rule 4
<-- [process_name=outlook : 4950,14857]
    [proc_words_chars=lte0..from0to7d5 : 2626,10140]
    [proc_count_in_win_1f=from3d51143to4d64917 : 1297,16547]
      : p=595,np=595,u=595,cx=21,s=595 # 9645

# Rule 5
<-- [process_name=outlook : 4950,14857]
    [proc_count_in_win_1f=from2d01267to3d51143 : 911,20514]
    [win_title_prot_words=3 : 2215,12059]
      : p=358,np=358,u=358,cx=21,c=1,s=358 # 9639

# Rule 6
<-- [process_name=netscape : 290,25011]
    [proc_count_in_win_1f=from2d01267to3d51143 : 911,20514]
    [proc_count_in_win_1f-2=from2d01267to3d51143 : 911,20544]
    [proc_count_in_win_1f-3=from2d01267to3d51143 : 911,20426]
    [win_title_prot_words=1 : 2881,34614]
      : p=290,np=290,u=290,cx=35,s=290 # 9644

# Rule 7
<-- [process_name=outlook : 4950,14857]
    [proc_count_in_win_1f=from2d01267to3d51143 : 911,20514]
    [win_title_prot_words=1 : 2881,34614]
      : p=154,np=154,u=154,cx=21,c=1,s=154 # 9640

# Rule 8
<-- [process_name=outlook : 4950,14857]
    [prot_words_chars=from8d5to24 : 2324,6779]
    [proc_count_in_win_1f=from2d01267to3d51143 : 911,20514]
    [win_title_prot_words=2 : 109,2007]
      : p=109,np=109,u=109,cx=28,s=109 # 9641
}

```

Output_Hypotheses User25

```

{
# -- This learning took =
# -- System (CPU) time = 0.11
# -- User (Total) time = 0
# -- Number of rules in the cover = 5
# -- Number of conditions = 13
# -- Complexity for this cover = 91
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 9526
negative_events = 44729

```

```

positive_distinct_events = 91
negative_distinct_events = 450
[user=user25]
  # Rule 1
  <-- [process_name=iexplore : 7599,140]
      [proc_count_in_win_1f=from3d51143to4d64917..from4d64917to5d56641 :
4798,22365]
      : p=3456,np=3456,u=3456,cx=14,s=3.46e+03 # 10375

  # Rule 2
  <-- [process_name=iexplore : 7599,140]
      [proc_count_in_win_1f=from0to2d01267..from2d01267to3d51143 :
4728,18892]
      [win_title_prot_words=3 : 8036,6238]
      : p=3995,np=3394,u=3995,cx=21,s=4e+03 # 10378

  # Rule 3
  <-- [process_name=outlook : 1927,17880]
      [proc_count_in_win_1f=from3d51143to4d64917 : 3518,14326]
      [win_title_prot_words=1 : 1342,36153]
      : p=1342,np=1342,u=1342,cx=21,c=1,s=1.34e+03 # 10379

  # Rule 4
  <-- [process_name=outlook : 1927,17880]
      [proc_count_in_win_1f=from2d01267to3d51143 : 4147,17278]
      [win_title_prot_words=3 : 8036,6238]
      : p=585,np=585,u=585,cx=21,s=585 # 10376

  # Rule 5
  <-- [proc_count_in_win_1f=from0to2d01267 : 581,1614]
      [win_title_prot_words=4 : 148,67]
      : p=148,np=148,u=148,cx=14,s=148 # 10377

}

```

C2 Experiment 040607-2: Filtered Data TR+TS, Characteristic Descriptions

Training Dataset:

Discretization: Dis-3

Filtering: Significance based, conjunctive, rank-threshold = 10, TR+TS

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

AQ21 Learning Parameters:

maxstar = 1 maxrule = 1 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Characteristic descriptions

Testing Parameters:

Evaluation of Conjunction = selectors ratio

Evaluation of Disjunction = max

Acceptance Threshold = 10%

Accuracy Tolerance = 5%

Learning Results:

Total number of rules: 71

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	8	8	1	10	8	6	8	9	8	5

Learned Models:

Output_Hypotheses User1

```
{
# -- This learning took =
# -- System (CPU) time = 0.13
# -- User (Total) time = 0
# -- Number of rules in the cover = 8
# -- Number of conditions = 57
# -- Complexity for this cover = 409
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0
```

```
positive_events = 522
negative_events = 53733
positive_distinct_events = 28
negative_distinct_events = 513
```

[user=user1]

```
# Rule 1
<-- [process_name=explorer,outlook : 394,20140]
[prot_words_chars=from8d5to24 : 285,8818]
[proc_count_in_win_lf=from2d01267to3d51143 : 377,21048]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 377,21122]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 377,21078]
[win_title_prot_words=3 : 269,14005]
: p=160,np=98,u=98,cx=44,c=1.5,s=160 # 1361

# Rule 2
<-- [process_name=explorer,ntvdm : 148,629]
[process_name-1=explorer,ntvdm : 148,634]
[process_name-2=explorer,ntvdm : 148,627]
[prot_words_chars=lte0..from8d5to24 : 444,46067]
[proc_count_in_win_lf=from0to2d01267..from2d01267to3d51143 :
470,23150]
[proc_count_in_win_lf-1=from0to2d01267..from2d01267to3d51143 :
470,23150]
[win_opened=lte16 : 522,53702]
[win_title_prot_words=0..1 : 195,37455]
: p=86,np=86,u=86,cx=62,s=86 # 1360

# Rule 3
<-- [process_name=explorer,iexplore : 176,8290]
[prot_words_chars=from8d5to24..from24to25d5 : 363,16269]
[proc_count_in_win_lf=from2d01267to3d51143 : 377,21048]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 377,21122]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 377,21078]
[win_title_prot_words=3 : 269,14005]
```



```

[win_title_prot_words-3=3 : 269,14007]
  : p=140,np=78,u=78,cx=51,s=140 # 1364

# Rule 4
<-- [process_name=outlook : 296,19511]
    [prot_words_chars=from0to7d5 : 109,12502]
    [prot_words_chars-1=from0to7d5 : 109,12502]
    [proc_count_in_win_lf=from2d01267to3d51143 : 377,21048]
    [proc_count_in_win_lf-1=from2d01267to3d51143 : 377,21122]
    [proc_count_in_win_lf-2=from2d01267to3d51143 : 377,21078]
    [win_title_prot_words=1 : 145,37350]
    : p=57,np=57,u=57,cx=49,c=1.57,s=57 # 1359

# Rule 5
<-- [process_name=outlook : 296,19511]
    [prot_words_chars=from0to7d5 : 109,12502]
    [proc_count_in_win_lf=from3d51143to4d64917 : 52,17792]
    [proc_count_in_win_lf-1=from3d51143to4d64917 : 52,17792]
    [proc_count_in_win_lf-2=from3d51143to4d64917 : 52,17792]
    [win_title_prot_words=1 : 145,37350]
    : p=52,np=52,u=52,cx=42,c=1.5,s=52 # 1363

# Rule 6
<-- [process_name=outlook : 296,19511]
    [process_name-1=outlook : 296,19511]
    [process_name-2=outlook : 296,19513]
    [prot_words_chars=from8d5to24 : 285,8818]
    [prot_words_chars-1=from8d5to24 : 285,8823]
    [proc_count_in_win_lf=from2d01267to3d51143 : 377,21048]
    [proc_count_in_win_lf-1=from2d01267to3d51143 : 377,21122]
    [win_opened=1tel6 : 522,53702]
    [win_title_prot_words=2 : 58,2058]
    : p=32,np=32,u=32,cx=63,s=32 # 1366

# Rule 7
<-- [process_name=outlook : 296,19511]
    [process_name-1=outlook : 296,19511]
    [process_name-2=outlook : 296,19513]
    [prot_words_chars=from8d5to24 : 285,8818]
    [proc_count_in_win_lf=from0to2d01267 : 93,2102]
    [proc_count_in_win_lf-1=from0to2d01267 : 93,2028]
    [proc_count_in_win_lf-2=from0to2d01267 : 93,2072]
    [win_title_prot_words=3 : 269,14005]
    : p=31,np=31,u=31,cx=56,c=1.75,s=31 # 1362

# Rule 8
<-- [process_name=outlook : 296,19511]
    [prot_words_chars=from8d5to24 : 285,8818]
    [proc_count_in_win_lf=from0to2d01267 : 93,2102]
    [proc_count_in_win_lf-1=from0to2d01267 : 93,2028]
    [win_title_prot_words=2 : 58,2058]
    [win_title_prot_words-1=2 : 58,2063]
    : p=26,np=26,u=26,cx=42,c=1.33,s=26 # 1365
}

Output_Hypotheses User2
{
# -- This learning took =
# -- System (CPU) time = 0.12
# -- User (Total) time = 1
# -- Number of rules in the cover = 8
# -- Number of conditions = 54

```

```

# -- Complexity for this cover      = 378
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events      = 4785
negative_events      = 49470
positive_distinct_events = 46
negative_distinct_events = 495
[user=user2]
# Rule 1
<-- [process_name=netscape : 3083,22218]
    [process_name-1=netscape : 3083,22218]
    [prot_words_chars=from7d5to8d5 : 3083,21559]
    [prot_words_chars-1=from7d5to8d5 : 3083,21559]
    [proc_count_in_win_lf=from2d01267to3d51143 : 2577,18848]
    [proc_count_in_win_lf-1=from2d01267to3d51143 : 2577,18922]
    [proc_count_in_win_lf-2=from2d01267to3d51143 : 2577,18878]
    [proc_count_in_win_lf-3=from2d01267to3d51143 : 2577,18760]
    [win_title_prot_words=1 : 3527,33968]
    : p=1731,np=1731,u=1731,cx=63,c=1.89,s=1.73e+03 # 2631

# Rule 2
<-- [process_name=netscape : 3083,22218]
    [prot_words_chars=from7d5to8d5 : 3083,21559]
    [prot_words_chars-1=from7d5to8d5 : 3083,21559]
    [proc_count_in_win_lf=from3d51143to4d64917 : 1877,15967]
    [proc_count_in_win_lf-1=from3d51143to4d64917 : 1877,15967]
    [win_title_prot_words=1 : 3527,33968]
    : p=1352,np=1352,u=1352,cx=42,c=1.33,s=1.35e+03 # 2630

# Rule 3
<-- [process_name=outlook : 849,18958]
    [process_name-1=outlook : 849,18958]
    [prot_words_chars=from8d5to24 : 1402,7701]
    [proc_count_in_win_lf=from2d01267to3d51143 : 2577,18848]
    [proc_count_in_win_lf-1=from2d01267to3d51143 : 2577,18922]
    [proc_count_in_win_lf-2=from2d01267to3d51143 : 2577,18878]
    [win_title_prot_words=3 : 611,13663]
    : p=611,np=611,u=611,cx=49,c=1.57,s=611 # 2633

# Rule 4
<-- [process_name=winword : 585,41]
    [process_name-1=winword : 585,41]
    [process_name-2=winword : 585,41]
    [prot_words_chars=from8d5to24 : 1402,7701]
    [proc_count_in_win_lf=from2d01267to3d51143..from3d51143to4d64917 :
4454,34815]
    [proc_count_in_win_lf-1=from2d01267to3d51143..from3d51143to4d64917 :
4454,34889]
    [win_title_prot_words=2 : 585,1531]
    : p=460,np=460,u=173,cx=49,s=460 # 2632

# Rule 5
<-- [process_name=outlook : 849,18958]
    [process_name-1=outlook : 849,18958]
    [prot_words_chars=from0to7d5 : 238,12373]
    [prot_words_chars-1=from0to7d5 : 238,12373]
    [proc_count_in_win_lf=from3d51143to4d64917 : 1877,15967]
    [proc_count_in_win_lf-1=from3d51143to4d64917 : 1877,15967]
    [win_title_prot_words=1 : 3527,33968]
    : p=238,np=238,u=238,cx=49,c=1.43,s=238 # 2635

# Rule 6
<-- [process_name=explorer : 206,521]

```

```

[prot_words_chars=from8d5to24 : 1402,7701]
[proc_count_in_win_lf=from0to2d01267 : 206,1989]
[win_title_prot_words=1 : 3527,33968]
[win_title_prot_words-3=1 : 3527,33968]
: p=206,np=206,u=206,cx=35,s=206 # 2634

# Rule 7
<-- [process_name=winword : 585,41]
[process_name-1=winword : 585,41]
[process_name-2=winword : 585,41]
[prot_words_chars=from8d5to24 : 1402,7701]
[proc_count_in_win_lf=from3d51143to4d64917..from4d64917to5d56641 :
2002,25161]
[win_title_prot_words=2 : 585,1531]
: p=412,np=125,u=125,cx=42,s=412 # 2636

# Rule 8
<-- [process_name=iexplore : 62,7677]
[prot_words_chars=from25d5to50 : 62,153]
[prot_words_chars-1=from25d5to50 : 62,74]
[proc_count_in_win_lf=from2d01267to3d51143 : 2577,18848]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 2577,18878]
[win_title_prot_words=4 : 62,153]
[win_title_prot_words-3=4 : 62,168]
: p=62,np=62,u=62,cx=49,s=62 # 2629

}

```

Output_Hypotheses User3

```

{
# -- This learning took =
# -- System (CPU) time = 0.06
# -- User (Total) time = 0
# -- Number of rules in the cover = 1
# -- Number of conditions = 3
# -- Complexity for this cover = 21
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 5
negative_events = 54250
[user=user3]
# Rule 1
<-- [process_name=wscript : 5,0]
[proc_count_in_win_lf=lte0 : 5,0]
[win_title_prot_words=4 : 5,210]
: p=5,np=5,u=5,cx=21,c=1,s=5 # 2710

}

```

Output_Hypotheses User4

```

{
# -- This learning took =
# -- System (CPU) time = 0.14
# -- User (Total) time = 0
# -- Number of rules in the cover = 10
# -- Number of conditions = 60
# -- Complexity for this cover = 420
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 18396

```

```

negative_events      = 35859
positive_distinct_events = 73
negative_distinct_events = 468

```

```
[user=user4]
```

```

# Rule 1
<-- [process_name=netscape : 11570,13731]
    [prot_words_chars=from7d5to8d5 : 11570,13072]
    [proc_count_in_win_lf=from3d51143to4d64917 : 6324,11520]
    [proc_count_in_win_lf-1=from3d51143to4d64917 : 6324,11520]
    [win_title_prot_words=1 : 17564,19931]
    : p=5027,np=5027,u=5027,cx=35,c=1.2,s=5.03e+03 # 4005

# Rule 2
<-- [process_name=netscape : 11570,13731]
    [prot_words_chars=from7d5to8d5 : 11570,13072]
    [proc_count_in_win_lf=from2d01267to3d51143 : 6317,15108]
    [proc_count_in_win_lf-1=from2d01267to3d51143 : 6317,15182]
    [proc_count_in_win_lf-2=from2d01267to3d51143 : 6317,15138]
    [proc_count_in_win_lf-3=from2d01267to3d51143 : 6317,15020]
    [win_title_prot_words=1 : 17564,19931]
    [win_title_prot_words-1=1 : 17564,19931]
    : p=4248,np=4248,u=4248,cx=56,s=4.25e+03 # 4007

# Rule 3
<-- [prot_words_chars=from0to7d5 : 5994,6617]
    [proc_count_in_win_lf=from5d56641to7d12078 : 2387,1080]
    [proc_count_in_win_lf-3=from5d56641to7d12078 : 2387,1080]
    [win_title_prot_words=1 : 17564,19931]
    : p=2387,np=2387,u=2387,cx=28,s=2.39e+03 # 4008

# Rule 4
<-- [process_name=netscape : 11570,13731]
    [prot_words_chars=from7d5to8d5 : 11570,13072]
    [proc_count_in_win_lf=from4d64917to5d56641 : 3103,6216]
    [proc_count_in_win_lf-1=from4d64917to5d56641 : 3103,6216]
    [win_title_prot_words=1 : 17564,19931]
    : p=2295,np=2295,u=2295,cx=35,c=1.2,s=2.3e+03 # 4012

# Rule 5
<-- [process_name=outlook : 6826,12981]
    [prot_words_chars=from0to7d5 : 5994,6617]
    [prot_words_chars-1=from0to7d5 : 5994,6617]
    [proc_count_in_win_lf=from3d51143to4d64917 : 6324,11520]
    [proc_count_in_win_lf-1=from3d51143to4d64917 : 6324,11520]
    [win_title_prot_words=1 : 17564,19931]
    : p=1297,np=1297,u=1297,cx=42,c=1.33,s=1.3e+03 # 4003

# Rule 6
<-- [process_name=outlook : 6826,12981]
    [prot_words_chars=from0to7d5 : 5994,6617]
    [proc_count_in_win_lf=from2d01267to3d51143 : 6317,15108]
    [proc_count_in_win_lf-1=from2d01267to3d51143 : 6317,15182]
    [proc_count_in_win_lf-2=from2d01267to3d51143 : 6317,15138]
    [win_title_prot_words=1 : 17564,19931]
    [win_title_prot_words-1=1 : 17564,19931]
    : p=1237,np=1237,u=1237,cx=49,c=1.57,s=1.24e+03 # 4009

# Rule 7
<-- [process_name=outlook : 6826,12981]
    [prot_words_chars=from0to7d5 : 5994,6617]
    [proc_count_in_win_lf=from4d64917to5d56641 : 3103,6216]

```

```

[proc_count_in_win_lf-1=from4d64917to5d56641 : 3103,6216]
[win_title_prot_words=1 : 17564,19931]
: p=808,np=808,u=808,cx=35,c=1.2,s=808 # 4006

# Rule 8
<-- [process_name=outlook : 6826,12981]
[process_name-1=outlook : 6826,12981]
[process_name-2=outlook : 6826,12983]
[prot_words_chars=from8d5to24 : 832,8271]
[proc_count_in_win_lf=from2d01267to3d51143 : 6317,15108]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 6317,15182]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 6317,15138]
[win_title_prot_words=3 : 639,13635]
: p=639,np=639,u=639,cx=56,c=1.75,s=639 # 4011

# Rule 9
<-- [process_name=outlook : 6826,12981]
[prot_words_chars=from0to7d5 : 5994,6617]
[proc_count_in_win_lf=from0to2d01267 : 265,1930]
[proc_count_in_win_lf-1=from0to2d01267 : 265,1856]
[win_title_prot_words=1 : 17564,19931]
: p=265,np=265,u=265,cx=35,c=1.2,s=265 # 4004

# Rule 10
<-- [process_name=outlook : 6826,12981]
[process_name-1=outlook : 6826,12981]
[process_name-2=outlook : 6826,12983]
[prot_words_chars=from8d5to24 : 832,8271]
[proc_count_in_win_lf=from2d01267to3d51143 : 6317,15108]
[win_opened=1tel6 : 18396,35828]
[win_title_prot_words=2 : 193,1923]
: p=193,np=193,u=193,cx=49,s=193 # 4010
}

Output_Hypotheses User5
{
# -- This learning took =
# -- System (CPU) time = 0.12
# -- User (Total) time = 0
# -- Number of rules in the cover = 8
# -- Number of conditions = 39
# -- Complexity for this cover = 273
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 5056
negative_events = 49199
positive_distinct_events = 61
negative_distinct_events = 480
[user=user5]
# Rule 1
<-- [process_name=netscape : 4065,21236]
[prot_words_chars=from7d5to8d5 : 4065,20577]
[proc_count_in_win_lf=from2d01267to3d51143 : 1944,19481]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 1944,19555]
[win_title_prot_words=1 : 4402,33093]
: p=1642,np=1574,u=1642,cx=35,c=1.2,s=1.64e+03 # 4973

# Rule 2
<-- [process_name=netscape : 4065,21236]
[prot_words_chars=from7d5to8d5 : 4065,20577]
[proc_count_in_win_lf=from3d51143to4d64917 : 1683,16161]

```

```

[proc_count_in_win_lf-1=from3d51143to4d64917 : 1683,16161]
: p=1126,np=1126,u=1126,cx=28,c=1.25,s=1.13e+03 # 4968

# Rule 3
<-- [process_name=netscape : 4065,21236]
[prot_words_chars=from7d5to8d5 : 4065,20577]
[proc_count_in_win_lf=from4d64917to5d56641 : 1123,8196]
[proc_count_in_win_lf-1=from4d64917to5d56641 : 1123,8196]
: p=991,np=991,u=991,cx=28,c=1.25,s=991 # 4974

# Rule 4
<-- [process_name=outlook : 991,18816]
[proc_count_in_win_lf=from3d51143to4d64917 : 1683,16161]
[proc_count_in_win_lf-1=from3d51143to4d64917 : 1683,16161]
[proc_count_in_win_lf-2=from3d51143to4d64917 : 1683,16161]
[win_title_prot_words=3 : 654,13620]
: p=352,np=352,u=352,cx=35,c=1.6,s=352 # 4972

# Rule 5
<-- [process_name=netscape : 4065,21236]
[prot_words_chars=from7d5to8d5 : 4065,20577]
[proc_count_in_win_lf=from0to2d01267 : 306,1889]
[proc_count_in_win_lf-1=from0to2d01267 : 306,1815]
: p=306,np=306,u=306,cx=28,c=1.25,s=306 # 4970

# Rule 6
<-- [process_name=outlook : 991,18816]
[process_name-1=outlook : 991,18816]
[prot_words_chars=from8d5to24 : 654,8449]
[proc_count_in_win_lf=from2d01267to3d51143 : 1944,19481]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 1944,19555]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 1910,19545]
[win_title_prot_words=3 : 654,13620]
: p=302,np=302,u=302,cx=49,c=1.57,s=302 # 4969

# Rule 7
<-- [process_name=outlook : 991,18816]
[prot_words_chars=from0to7d5 : 337,12274]
[proc_count_in_win_lf=from3d51143to4d64917 : 1683,16161]
[proc_count_in_win_lf-1=from3d51143to4d64917 : 1683,16161]
[proc_count_in_win_lf-2=from3d51143to4d64917 : 1683,16161]
[win_title_prot_words=1 : 4402,33093]
: p=205,np=205,u=205,cx=42,c=1.5,s=205 # 4971

# Rule 8
<-- [process_name=outlook : 991,18816]
[prot_words_chars=from0to7d5 : 337,12274]
[proc_count_in_win_lf=from4d64917to5d56641 : 1123,8196]
[proc_count_in_win_lf-1=from4d64917to5d56641 : 1123,8196]
: p=132,np=132,u=132,cx=28,c=1.25,s=132 # 4975
}

```

Output_Hypotheses User7

```

{
# -- This learning took =
# -- System (CPU) time = 0.1
# -- User (Total) time = 0
# -- Number of rules in the cover = 6
# -- Number of conditions = 32
# -- Complexity for this cover = 226
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

```

```

positive_events          = 1556
negative_events          = 52699
positive_distinct_events = 73
negative_distinct_events = 468

[user=user7]
  # Rule 1
  <-- [process_name=outlook : 1293,18514]
      [prot_words_chars=from0to7d5 : 1092,11519]
      [proc_count_in_win_lf=from4d64917to5d56641 : 630,8689]
      [proc_count_in_win_lf-1=from4d64917to5d56641 : 630,8689]
      : p=630,np=630,u=630,cx=28,c=1.25,s=630 # 5574

  # Rule 2
  <-- [process_name=outlook : 1293,18514]
      [prot_words_chars=from0to7d5 : 1092,11519]
      [proc_count_in_win_lf=from2d01267to3d51143 : 555,20870]
      [proc_count_in_win_lf-1=from2d01267to3d51143 : 555,20944]
      [win_title_prot_words=1 : 1092,36403]
      : p=326,np=326,u=326,cx=35,c=1.2,s=326 # 5573

  # Rule 3
  <-- [process_name=explorer : 263,464]
      [process_name-1=explorer : 263,469]
      [process_name-2=explorer,outlook : 1556,18978]
      [prot_words_chars=from8d5to24 : 394,8709]
      [proc_count_in_win_lf=from0to2d01267..from2d01267to3d51143 :
926,22694]
      [win_opened=lte16 : 1556,52668]
      [win_title_prot_words=2 : 263,1853]
      [win_title_prot_words-1=2 : 263,1858]
      : p=263,np=263,u=263,cx=58,s=263 # 5569

  # Rule 4
  <-- [process_name=outlook : 1293,18514]
      [prot_words_chars=from0to7d5 : 1092,11519]
      [proc_count_in_win_lf=from0to2d01267 : 371,1824]
      [proc_count_in_win_lf-1=from0to2d01267 : 371,1750]
      [win_title_prot_words-2=1 : 1092,36403]
      : p=136,np=136,u=136,cx=35,c=1.6,s=136 # 5570

  # Rule 5
  <-- [process_name=outlook : 1293,18514]
      [prot_words_chars=from8d5to24 : 394,8709]
      [proc_count_in_win_lf=from0to2d01267 : 371,1824]
      [proc_count_in_win_lf-1=from0to2d01267 : 371,1750]
      [proc_count_in_win_lf-2=from0to2d01267 : 371,1794]
      [win_title_prot_words=3 : 131,14143]
      : p=131,np=131,u=131,cx=42,c=1.5,s=131 # 5572

  # Rule 6
  <-- [process_name=outlook : 1293,18514]
      [prot_words_chars=lte0 : 70,85]
      [proc_count_in_win_lf=from2d01267to3d51143 : 555,20870]
      [win_title_prot_words=0 : 70,85]
      : p=70,np=70,u=70,cx=28,s=70 # 5571

}

Output_Hypotheses User8
{
  # -- This learning took =

```

```

# -- System (CPU) time = 0.14
# -- User (Total) time = 0
# -- Number of rules in the cover = 8
# -- Number of conditions = 48
# -- Complexity for this cover = 340
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 3249
negative_events = 51006
positive_distinct_events = 50
negative_distinct_events = 491
[user=user8]

# Rule 1
<-- [process_name=netscape : 2769,22532]
[process_name-1=netscape : 2769,22532]
[prot_words_chars=from7d5to8d5..from8d5to24 : 3072,30673]
[proc_count_in_win_lf=from3d51143to4d64917 : 1294,16550]
[proc_count_in_win_lf-1=from3d51143to4d64917 : 1294,16550]
: p=1294,np=1294,u=1294,cx=35,c=1.4,s=1.29e+03 # 6871

# Rule 2
<-- [process_name=netscape : 2769,22532]
[process_name-1=netscape : 2769,22532]
[prot_words_chars=from7d5to8d5 : 2110,22532]
[prot_words_chars-1=from7d5to8d5 : 2110,22532]
[proc_count_in_win_lf=from2d01267to3d51143 : 1430,19995]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 1430,20069]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 1430,20025]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 1430,19907]
[win_opened=1tel6 : 3249,50975]
[win_title_prot_words=1 : 2287,35208]
[win_title_prot_words-1=1 : 2287,35208]
: p=574,np=574,u=574,cx=77,s=574 # 6872

# Rule 3
<-- [process_name=netscape : 2769,22532]
[prot_words_chars=from7d5to8d5 : 2110,22532]
[proc_count_in_win_lf=from4d64917to5d56641 : 325,8994]
[win_title_prot_words=1 : 2287,35208]
: p=325,np=325,u=325,cx=28,c=1,s=325 # 6874

# Rule 4
<-- [process_name=outlook : 439,19368]
[process_name-1=outlook : 439,19368]
[prot_words_chars=from8d5to24 : 962,8141]
[proc_count_in_win_lf=from2d01267to3d51143 : 1430,19995]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 1430,20069]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 1430,20025]
[win_title_prot_words=3 : 262,14012]
: p=262,np=262,u=262,cx=49,c=1.57,s=262 # 6867

# Rule 5
<-- [process_name=netscape : 2769,22532]
[prot_words_chars=from7d5to8d5 : 2110,22532]
[proc_count_in_win_lf=from0to2d01267 : 200,1995]
[win_title_prot_words=1 : 2287,35208]
: p=113,np=113,u=113,cx=28,c=1,s=113 # 6868

# Rule 6
<-- [process_name=outlook : 439,19368]
[prot_words_chars=from0to7d5 : 177,12434]
[proc_count_in_win_lf=from2d01267to3d51143 : 1430,19995]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 1430,20069]

```



```

[win_title_prot_words=1 : 2287,35208]
[win_title_prot_words-1=1 : 2287,35208]
: p=90,np=90,u=90,cx=42,c=1.33,s=90 # 6870

# Rule 7
<-- [process_name=outlook : 439,19368]
[prot_words_chars=from0to7d5 : 177,12434]
[proc_count_in_win_lf=from0to2d01267 : 200,1995]
[proc_count_in_win_lf-1=from0to2d01267 : 200,1921]
[win_title_prot_words=1 : 2287,35208]
: p=87,np=87,u=87,cx=35,c=1.2,s=87 # 6869

# Rule 8
<-- [process_name=netscape,winword : 2810,23117]
[process_name-1=netscape,winword : 2810,23117]
[prot_words_chars=from8d5to24 : 962,8141]
[proc_count_in_win_lf=from2d01267to3d51143 : 1430,19995]
[win_opened=ltel6 : 3249,50975]
[win_title_prot_words=2 : 700,1416]
: p=504,np=41,u=504,cx=46,c=1.17,s=504 # 6873
}

Output_Hypotheses User12
{
# -- This learning took =
# -- System (CPU) time = 0.13
# -- User (Total) time = 0
# -- Number of rules in the cover = 9
# -- Number of conditions = 60
# -- Complexity for this cover = 424
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 5920
negative_events = 48335
positive_distinct_events = 70
negative_distinct_events = 471
[user=user12]
# Rule 1
<-- [process_name=netscape : 3524,21777]
[prot_words_chars=from7d5to8d5 : 3524,21118]
[proc_count_in_win_lf=from2d01267to3d51143 : 3167,18258]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 3167,18332]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 3167,18288]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 3167,18170]
[win_opened=ltel6 : 5920,48304]
[win_title_prot_words=1 : 4255,33240]
[win_title_prot_words-1=1 : 4255,33240]
: p=1948,np=1948,u=1948,cx=63,s=1.95e+03 # 8210

# Rule 2
<-- [process_name=netscape : 3524,21777]
[prot_words_chars=from7d5to8d5 : 3524,21118]
[proc_count_in_win_lf=from3d51143to4d64917 : 1799,16045]
[proc_count_in_win_lf-1=from3d51143to4d64917 : 1799,16045]
[win_title_prot_words=1 : 4255,33240]
: p=955,np=955,u=955,cx=35,c=1.2,s=955 # 8214

# Rule 3
<-- [process_name=explorer,netscape : 3684,22344]
[process_name-1=explorer,netscape : 3684,22349]
[prot_words_chars=from7d5to8d5..from8d5to24 : 5189,28556]

```

```

[proc_count_in_win_lf=from4d64917to5d56641 : 781,8538]
[proc_count_in_win_lf-1=from4d64917to5d56641 : 781,8538]
[win_title_prot_words=1..2 : 4463,35148]
: p=781,np=781,u=781,cx=46,c=1.33,s=781 # 8209

# Rule 4
<-- [process_name=outlook : 2236,17571]
[process_name-1=outlook : 2236,17571]
[prot_words_chars=from8d5to24 : 1665,7438]
[proc_count_in_win_lf=from2d01267to3d51143 : 3167,18258]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 3167,18332]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 3167,18288]
[win_title_prot_words=3 : 1457,12817]
: p=742,np=742,u=742,cx=49,c=1.57,s=742 # 8211

# Rule 5
<-- [process_name=outlook : 2236,17571]
[process_name-1=outlook : 2236,17571]
[prot_words_chars=from8d5to24 : 1665,7438]
[proc_count_in_win_lf=from3d51143to4d64917 : 1799,16045]
[proc_count_in_win_lf-1=from3d51143to4d64917 : 1799,16045]
[win_title_prot_words=3 : 1457,12817]
[win_title_prot_words-3=3 : 1457,12819]
: p=590,np=590,u=590,cx=49,s=590 # 8213

# Rule 6
<-- [process_name=outlook : 2236,17571]
[prot_words_chars=from0to7d5 : 731,11880]
[proc_count_in_win_lf=from2d01267to3d51143 : 3167,18258]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 3167,18332]
[win_title_prot_words=1 : 4255,33240]
[win_title_prot_words-1=1 : 4255,33240]
: p=477,np=477,u=477,cx=42,c=1.33,s=477 # 8215

# Rule 7
<-- [process_name=outlook : 2236,17571]
[prot_words_chars=from0to7d5 : 731,11880]
[proc_count_in_win_lf=from3d51143to4d64917 : 1799,16045]
[proc_count_in_win_lf-1=from3d51143to4d64917 : 1799,16045]
[win_title_prot_words=1 : 4255,33240]
[win_title_prot_words-1=1 : 4255,33240]
: p=254,np=254,u=254,cx=42,c=1.33,s=254 # 8216

# Rule 8
<-- [process_name=outlook : 2236,17571]
[process_name-1=outlook : 2236,17571]
[process_name-2=outlook : 2236,17573]
[prot_words_chars=from8d5to24 : 1665,7438]
[proc_count_in_win_lf=from0to2d01267 : 173,2022]
[proc_count_in_win_lf-1=from0to2d01267 : 173,1948]
[win_title_prot_words=3 : 1457,12817]
: p=125,np=125,u=125,cx=49,c=1.57,s=125 # 8208

# Rule 9
<-- [process_name=outlook : 2236,17571]
[process_name-1=outlook : 2236,17571]
[prot_words_chars=from8d5to24 : 1665,7438]
[proc_count_in_win_lf=from0to2d01267 : 173,2022]
[proc_count_in_win_lf-1=from0to2d01267 : 173,1948]
[win_title_prot_words=2 : 208,1908]
[win_title_prot_words-1=2 : 208,1913]
: p=48,np=48,u=48,cx=49,c=1.43,s=48 # 8212

```

}

Output_Hypotheses User19

{

```
# -- This learning took =
# -- System (CPU) time   = 0.12
# -- User (Total) time   = 1
# -- Number of rules in the cover = 8
# -- Number of conditions      = 49
# -- Complexity for this cover = 343
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0
```

```
positive_events      = 5240
negative_events      = 49015
positive_distinct_events = 44
negative_distinct_events = 497
```

[user=user19]

Rule 1

```
<-- [process_name=outlook : 4950,14857]
[process_name-1=outlook : 4950,14857]
[prot_words_chars=from8d5to24 : 2324,6779]
[proc_count_in_win_lf=from3d51143to4d64917..from4d64917to5d56641 :
3249,23914]
[win_title_prot_words=3 : 2215,12059]
[win_title_prot_words-1=3 : 2215,12133]
: p=1857,np=1857,u=1857,cx=42,c=1.33,s=1.86e+03 # 9316
```

Rule 2

```
<-- [process_name=outlook : 4950,14857]
[proc_count_in_win_lf=from5d56641to7d12078 : 1080,2387]
[proc_count_in_win_lf-1=from5d56641to7d12078 : 1080,2387]
[win_title_prot_words=1 : 2881,34614]
: p=1080,np=1080,u=1080,cx=28,c=1.25,s=1.08e+03 # 9321
```

Rule 3

```
<-- [process_name=outlook : 4950,14857]
[prot_words_chars=from0to7d5 : 2591,10020]
[proc_count_in_win_lf=from4d64917to5d56641 : 1952,7367]
[proc_count_in_win_lf-1=from4d64917to5d56641 : 1952,7367]
[win_title_prot_words=1 : 2881,34614]
: p=797,np=797,u=797,cx=35,c=1.2,s=797 # 9320
```

Rule 4

```
<-- [process_name=outlook : 4950,14857]
[prot_words_chars=lte0..from0to7d5 : 2626,10140]
[proc_count_in_win_lf=from3d51143to4d64917 : 1297,16547]
[proc_count_in_win_lf-1=from3d51143to4d64917 : 1297,16547]
[proc_count_in_win_lf-2=from3d51143to4d64917 : 1297,16547]
: p=595,np=595,u=595,cx=35,c=1.6,s=595 # 9323
```

Rule 5

```
<-- [process_name=outlook : 4950,14857]
[process_name-1=outlook : 4950,14857]
[proc_count_in_win_lf=from2d01267to3d51143 : 911,20514]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 911,20588]
[win_title_prot_words=3 : 2215,12059]
: p=358,np=358,u=358,cx=35,c=1.4,s=358 # 9317
```

Rule 6

```
<-- [process_name=netscape : 290,25011]
[process_name-1=netscape : 290,25011]
```

```

[prot_words_chars=from7d5to8d5 : 290,24352]
[prot_words_chars-1=from7d5to8d5 : 290,24352]
[proc_count_in_win_lf=from2d01267to3d51143 : 911,20514]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 911,20588]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 911,20544]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 911,20426]
[win_title_prot_words=1 : 2881,34614]
: p=290,np=290,u=290,cx=63,s=290 # 9322

# Rule 7
<-- [process_name=outlook : 4950,14857]
[process_name-1=outlook : 4950,14857]
[prot_words_chars=from0to7d5 : 2591,10020]
[proc_count_in_win_lf=from2d01267to3d51143 : 911,20514]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 911,20588]
[win_title_prot_words=1 : 2881,34614]
[win_title_prot_words-1=1 : 2881,34614]
: p=154,np=154,u=154,cx=49,c=1.43,s=154 # 9318

# Rule 8
<-- [process_name=outlook : 4950,14857]
[process_name-1=outlook : 4950,14857]
[process_name-2=outlook : 4950,14859]
[prot_words_chars=from8d5to24 : 2324,6779]
[proc_count_in_win_lf=from2d01267to3d51143 : 911,20514]
[win_opened=1to16 : 5240,48984]
[win_title_prot_words=2 : 109,2007]
[win_title_prot_words-1=2 : 109,2012]
: p=109,np=109,u=109,cx=56,s=109 # 9319
}

Output_Hypotheses User25
{
# -- This learning took =
# -- System (CPU) time = 0.11
# -- User (Total) time = 0
# -- Number of rules in the cover = 5
# -- Number of conditions = 26
# -- Complexity for this cover = 182
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 9526
negative_events = 44729
positive_distinct_events = 91
negative_distinct_events = 450
[user=user25]
# Rule 1
<-- [process_name=iexplore : 7599,140]
[prot_words_chars=from24to25d5 : 7451,78]
[proc_count_in_win_lf=from3d51143to4d64917..from4d64917to5d56641 :
4798,22365]
[win_title_prot_words=3 : 8036,6238]
: p=3456,np=3456,u=1280,cx=28,c=1,s=3.46e+03 # 10039

# Rule 2
<-- [process_name=iexplore : 7599,140]
[prot_words_chars=from24to25d5..from25d5to50 : 7599,145]
[prot_words_chars-1=from24to25d5 : 7525,78]
[proc_count_in_win_lf=from0to2d01267..from3d51143to4d64917 :
8246,33218]

```

```

[proc_count_in_win_lf-1=from0to2d01267..from3d51143to4d64917 :
8246,33218]
[win_title_prot_words=3..4 : 8184,6305]
: p=6245,np=3394,u=3995,cx=42,s=6.24e+03 # 10042

# Rule 3
<-- [process_name=outlook : 1927,17880]
[prot_words_chars=from0to7d5 : 1342,11269]
[prot_words_chars-1=from0to7d5 : 1342,11269]
[proc_count_in_win_lf=from3d51143to4d64917 : 3518,14326]
[proc_count_in_win_lf-1=from3d51143to4d64917 : 3518,14326]
[win_title_prot_words=1 : 1342,36153]
: p=1342,np=1342,u=1342,cx=42,c=1.33,s=1.34e+03 # 10043

# Rule 4
<-- [process_name=outlook : 1927,17880]
[process_name-1=outlook : 1927,17880]
[prot_words_chars=from8d5to24 : 585,8518]
[proc_count_in_win_lf=from2d01267to3d51143 : 4147,17278]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 4221,17278]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 4211,17244]
[win_title_prot_words=3 : 8036,6238]
: p=585,np=585,u=585,cx=49,c=1.57,s=585 # 10040

# Rule 5
<-- [proc_count_in_win_lf=from0to2d01267 : 581,1614]
[win_title_prot_words=4 : 148,67]
[win_title_prot_words-1=3..4 : 8184,6300]
: p=148,np=148,u=74,cx=21,s=148 # 10041

```

C3 Experiment 040727-3: Unfiltered Data, Simplicity-based Descriptions

Source Data: window records

Training Dataset:

Discretization: Dis-3

Filtering: not filtered

Testing Dataset:

Discretization: Dis-3

Filtering: not filtered

AQ21 Learning Parameters:

maxstar = 1 maxrule = 1 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Simplicity-based descriptions

Learning Results:

Total number of rules: 3781

	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	204	693	27	711	286	308	328	462	339	423

Learned rules (selected 20 rules per user):

```
Output_Hypotheses User1
{
  # -- This learning took =
  # -- System (CPU) time = 15.2
  # -- User (Total) time = 15
  # -- Number of rules in the cover = 204
  # -- Number of conditions = 1400
  # -- Complexity for this cover = 10014
  # -- Average number of rules kept from each stars = 1
  # -- Uncovered Positives = 0

  positive_events = 394
  negative_events = 8647
  positive_distinct_events = 354
  negative_distinct_events = 7788
[user=user1]
  # Rule 1
  <-- [process_name=cmd,emacs,iexplore,ntvdm : 124,1496]
      [process_name-1=emacs,iexplore : 109,1492]
      [proc_count_in_win_lf=lte0 : 126,1638]
      [win_title_prot_words=0 : 116,984]
      [win_title_prot_words-1=0 : 117,1001]
      : p=9,np=9,u=6,cx=43,c=1.4,s=9 # 117810

  # Rule 2
  <-- [process_name=explorer,taskmgr : 101,924]
      [proc_count_in_win_lf=from2d01267to3d51143 : 50,2056]
      [win_title_prot_words=3..4 : 101,2732]
      : p=7,np=7,u=6,cx=23,s=7 # 117811
```

```

# Rule 3
<-- [process_name=iexplore : 104,1492]
    [process_name-1=emacs,iexplore : 109,1492]
    [process_name-2=emacs,iexplore : 109,1487]
    [prot_words_chars=lte0 : 116,984]
    [prot_words_chars-1=lte0 : 117,1001]
    [proc_count_in_win_lf=from0to2d01267 : 215,4363]
    [proc_count_in_win_lf-1=lte0 : 122,1606]
    [win_opened=lte16 : 394,8065]
    : p=6,np=6,u=1,cx=60,c=1.62,s=6 # 117863

# Rule 4
<-- [process_name=outlook : 143,2877]
    [process_name-1=outlook : 145,2913]
    [process_name-2=outlook : 149,2947]
    [process_name-3=outlook : 154,2987]
    [prot_words_chars=from0to7d5 : 49,1217]
    [proc_count_in_win_lf=from0to2d01267 : 215,4363]
    [proc_count_in_win_lf-1=from0to2d01267 : 217,4386]
    [proc_count_in_win_lf-2=lte0..from0to2d01267 : 337,5968]
    [proc_count_in_win_lf-3=from0to2d01267 : 227,4428]
    [win_title_prot_words=1 : 113,3351]
    [win_title_prot_words-1=3 : 77,2201]
    [win_title_prot_words-2=1..2 : 166,4828]
    [win_title_prot_words-3=3 : 76,2182]
    : p=7,np=6,u=6,cx=91,c=2.38,s=7 # 117927

# Rule 5
<-- [process_name=iexplore,ntvdm : 111,1492]
    [process_name-1=iexplore,ntvdm : 111,1492]
    [prot_words_chars=lte0 : 116,984]
    [prot_words_chars-2=lte0 : 118,1010]
    [proc_count_in_win_lf=from2d01267to3d51143 : 50,2056]
    : p=17,np=6,u=13,cx=39,s=17 # 117962

# Rule 6
<-- [process_name=iexplore : 104,1492]
    [prot_words_chars=lte0 : 116,984]
    [proc_count_in_win_lf=from0to2d01267 : 215,4363]
    [proc_count_in_win_lf-1=from0to2d01267 : 217,4386]
    [proc_count_in_win_lf-2=from2d01267to3d51143 : 54,2078]
    [win_title_prot_words-1=0 : 117,1001]
    [win_title_prot_words-2=0 : 118,1010]
    : p=5,np=5,u=2,cx=49,s=5 # 117801

# Rule 7
<-- [process_name=cmd,explorer : 105,911]
    [process_name-1=cmd,explorer : 107,868]
    [process_name-2=cmd,msoffice,shstat : 34,97]
    [proc_count_in_win_lf=lte0 : 126,1638]
    [proc_count_in_win_lf-1=lte0 : 122,1606]
    [win_opened=lte16 : 394,8065]
    [win_title_prot_words=2 : 53,1484]
    : p=5,np=5,u=1,cx=57,c=1.57,s=5 # 117807

# Rule 8
<-- [process_name=outlook : 143,2877]
    [process_name-1=outlook : 145,2913]
    [process_name-3=ntvdm,outlook : 161,2987]
    [prot_words_chars-2=from8d5to24 : 175,3097]
    [proc_count_in_win_lf=from0to2d01267 : 215,4363]
    [proc_count_in_win_lf-1=from0to2d01267 : 217,4386]
    [proc_count_in_win_lf-2=from0to2d01267 : 219,4402]

```

```

[proc_count_in_win_lf-3=lte0..from0to2d01267 : 342,5960]
[win_title_prot_words=3 : 75,2200]
[win_title_prot_words-1=1 : 112,3356]
[win_title_prot_words-2=3 : 80,2192]
[win_title_prot_words-3=1 : 106,3345]
: p=5,np=5,u=4,cx=86,s=5 # 117905

# Rule 9
<-- [process_name=explorer : 97,907]
[prot_words_chars=from8d5to24 : 170,3124]
[proc_count_in_win_lf=from0to2d01267 : 215,4363]
[proc_count_in_win_lf-1=lte0 : 122,1606]
[proc_count_in_win_lf-3=from0to2d01267 : 227,4428]
[win_opened=lte16 : 394,8065]
[win_title_prot_words=1 : 113,3351]
[win_title_prot_words-1=0..1 : 229,4357]
: p=4,np=4,u=1,cx=56,c=1.62,s=4 # 117805

# Rule 10
<-- [process_name-1=emacs,shstat : 25,78]
[proc_count_in_win_lf=lte0 : 126,1638]
[proc_count_in_win_lf-1=lte0 : 122,1606]
[win_title_prot_words=0..2 : 282,5819]
: p=4,np=4,u=1,cx=30,c=1.5,s=4 # 117806

# Rule 11
<-- [process_name=explorer : 97,907]
[prot_words_chars-1=lte0 : 117,1001]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 52,2059]
[win_title_prot_words=2 : 53,1484]
: p=4,np=4,u=4,cx=28,c=1.5,s=4 # 117848

# Rule 12
<-- [process_name=msoffice : 25,101]
[win_title_prot_words=5 : 11,92]
[win_title_prot_words-1=1 : 112,3356]
: p=4,np=4,u=4,cx=21,s=4 # 117896

# Rule 13
<-- [process_name=msoffice : 25,101]
[process_name-1=shstat,taskmgr : 24,95]
[win_title_prot_words=4 : 26,532]
: p=3,np=3,u=2,cx=23,c=1.33,s=3 # 117823

# Rule 14
<-- [process_name=iexplore : 104,1492]
[prot_words_chars=from24to25d5 : 20,876]
[proc_count_in_win_lf=from2d01267to3d51143 : 50,2056]
[win_title_prot_words-1=0 : 117,1001]
[win_title_prot_words-2=0 : 118,1010]
: p=3,np=3,u=3,cx=35,s=3 # 117843

# Rule 15
<-- [process_name=explorer : 97,907]
[process_name-1=explorer : 99,863]
[prot_words_chars=from25d5to50 : 38,635]
: p=3,np=3,u=3,cx=21,c=1.33,s=3 # 117846

# Rule 16
<-- [process_name=outlook : 143,2877]
[process_name-3=emacs,msoffice,shstat : 34,92]
[win_title_prot_words=4 : 26,532]
: p=3,np=3,u=1,cx=25,s=3 # 117895

```



```

# Rule 17
<-- [process_name=explorer,wordpad : 103,944]
    [proc_count_in_win_lf=lte0 : 126,1638]
    [proc_count_in_win_lf-1=from0to2d01267 : 217,4386]
    [win_opened=lte16 : 394,8065]
    [win_title_prot_words=1 : 113,3351]
    [win_title_prot_words-1=0 : 117,1001]
    [win_title_prot_words-2=1 : 108,3352]
    [win_title_prot_words-3=1 : 106,3345]
    : p=4,np=3,u=4,cx=58,s=4 # 117929

# Rule 18
<-- [process_name=explorer,smsmon32 : 99,922]
    [process_name-1=explorer : 99,863]
    [process_name-3=cmd,explorer : 109,821]
    [prot_words_chars=lte0 : 116,984]
    [proc_count_in_win_lf=from0to2d01267 : 215,4363]
    [proc_count_in_win_lf-1=from0to2d01267 : 217,4386]
    [win_opened=lte16 : 394,8065]
    [win_title_prot_words-1=1 : 112,3356]
    [win_title_prot_words-2=1..2 : 166,4828]
    : p=5,np=3,u=4,cx=67,c=1.89,s=5 # 117957

# Rule 19
<-- [process_name=explorer,shstat : 108,907]
    [proc_count_in_win_lf=lte0 : 126,1638]
    [proc_count_in_win_lf-1=from0to2d01267 : 217,4386]
    [win_title_prot_words=0 : 116,984]
    [win_title_prot_words-1=4 : 28,543]
    : p=2,np=2,u=1,cx=37,c=1.4,s=2 # 117809

# Rule 20
<-- [process_name=outlook : 143,2877]
    [process_name-1=outlook : 145,2913]
    [prot_words_chars=from0to7d5 : 49,1217]
    [prot_words_chars-2=from0to7d5 : 49,1230]
    [proc_count_in_win_lf=lte0..from0to2d01267 : 341,6001]
    [proc_count_in_win_lf-1=from0to2d01267 : 217,4386]
    [proc_count_in_win_lf-2=lte0 : 118,1566]
    [win_title_prot_words-1=3 : 77,2201]
    [win_title_prot_words-3=3 : 76,2182]
    : p=2,np=2,u=1,cx=63,c=2.11,s=2 # 117819

...

Output_Hypotheses User2
{
# -- This learning took =
# -- System (CPU) time = 58.5
# -- User (Total) time = 58
# -- Number of rules in the cover = 693
# -- Number of conditions = 4880
# -- Complexity for this cover = 35270
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 1585
negative_events = 7456
positive_distinct_events = 1425
negative_distinct_events = 6717
[user=user2]
# Rule 1
<-- [process_name=explorer : 335,669]
    [process_name-1=explorer,outlook : 758,3175]

```

```

[process_name-3=explorer,winword : 426,1062]
[prot_words_chars=from8d5to24 : 705,2589]
[prot_words_chars-2=from0to7d5..from8d5to24 : 1279,5083]
[proc_count_in_win_lf=from0to2d01267..from2d01267to3d51143 :
1177,5507]
[proc_count_in_win_lf-1=from0to2d01267 : 883,3720]
[proc_count_in_win_lf-2=from0to2d01267 : 887,3734]
[proc_count_in_win_lf-3=from0to2d01267..from3d51143to4d64917 :
1250,5989]
[win_title_prot_words=1 : 766,2698]
[win_title_prot_words-1=1 : 766,2702]
: p=52,np=42,u=37,cx=81,c=2.18,s=52 # 483906

# Rule 2
<-- [process_name=netscape : 522,1761]
[process_name-2=cmd,netscape : 534,1766]
[prot_words_chars=from7d5to8d5 : 410,1402]
[prot_words_chars-3=from7d5to8d5 : 410,1397]
[proc_count_in_win_lf=from0to2d01267 : 875,3703]
[proc_count_in_win_lf-1=from0to2d01267 : 883,3720]
[proc_count_in_win_lf-2=lte0..from0to2d01267 : 1213,5092]
[proc_count_in_win_lf-3=from0to2d01267 : 888,3767]
[win_title_prot_words-1=1 : 766,2702]
[win_title_prot_words-3=1 : 767,2684]
: p=32,np=32,u=31,cx=72,c=2.5,s=32 # 483919

# Rule 3
<-- [process_name=netscape : 522,1761]
[process_name-1=netscape : 528,1761]
[prot_words_chars=from7d5to8d5 : 410,1402]
[prot_words_chars-1=from7d5to8d5 : 410,1402]
[proc_count_in_win_lf=from0to2d01267 : 875,3703]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 302,1809]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 303,1829]
[proc_count_in_win_lf-3=from0to2d01267 : 888,3767]
: p=15,np=15,u=15,cx=56,s=15 # 484071

# Rule 4
<-- [process_name=netscape : 522,1761]
[prot_words_chars-1=from7d5to8d5 : 410,1402]
[prot_words_chars-2=from7d5to8d5 : 410,1401]
[prot_words_chars-3=from7d5to8d5 : 410,1397]
[proc_count_in_win_lf=from2d01267to3d51143 : 302,1804]
[proc_count_in_win_lf-1=from0to2d01267 : 883,3720]
[proc_count_in_win_lf-2=from0to2d01267 : 887,3734]
[proc_count_in_win_lf-3=from0to2d01267 : 888,3767]
[win_title_prot_words=1 : 766,2698]
: p=13,np=13,u=13,cx=63,s=13 # 484077

# Rule 5
<-- [process_name=netscape : 522,1761]
[proc_count_in_win_lf=from0to2d01267..from2d01267to3d51143 :
1177,5507]
[win_title_prot_words=3 : 246,2029]
: p=16,np=13,u=1,cx=21,c=1,s=16 # 483912

# Rule 6
<-- [process_name=netscape : 522,1761]
[process_name-2=netscape : 532,1755]
[prot_words_chars=from7d5to8d5 : 410,1402]
[prot_words_chars-1=from7d5to8d5 : 410,1402]
[proc_count_in_win_lf=from0to2d01267 : 875,3703]
[proc_count_in_win_lf-1=from0to2d01267 : 883,3720]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 303,1829]

```

```

[proc_count_in_win_lf-3=from2d01267to3d51143 : 303,1833]
[win_title_prot_words-1=1 : 766,2702]
[win_title_prot_words-2=1 : 767,2693]
: p=12,np=12,u=12,cx=70,s=12 # 483943

# Rule 7
<-- [process_name=netscape : 522,1761]
[process_name-3=netscape : 531,1750]
[prot_words_chars=from7d5to8d5 : 410,1402]
[prot_words_chars-2=from7d5to8d5..from8d5to24 : 1105,3978]
[prot_words_chars-3=from7d5to8d5 : 410,1397]
[proc_count_in_win_lf=from0to2d01267 : 875,3703]
[proc_count_in_win_lf-1=from0to2d01267 : 883,3720]
[proc_count_in_win_lf-2=from0to2d01267 : 887,3734]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 303,1833]
[win_title_prot_words-1=1 : 766,2702]
: p=12,np=12,u=11,cx=70,s=12 # 484015

# Rule 8
<-- [process_name=netscape : 522,1761]
[process_name-2=netscape : 532,1755]
[prot_words_chars-2=from7d5to8d5 : 410,1401]
[proc_count_in_win_lf=from2d01267to3d51143 : 302,1804]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 302,1809]
[proc_count_in_win_lf-2=from0to2d01267 : 887,3734]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 303,1833]
[win_title_prot_words=1 : 766,2698]
[win_title_prot_words-1=1 : 766,2702]
: p=11,np=11,u=11,cx=63,c=2.22,s=11 # 484017

# Rule 9
<-- [process_name=netscape : 522,1761]
[prot_words_chars-2=from7d5to8d5 : 410,1401]
[prot_words_chars-3=from7d5to8d5 : 410,1397]
[proc_count_in_win_lf=from2d01267to3d51143 : 302,1804]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 302,1809]
[proc_count_in_win_lf-2=from0to2d01267 : 887,3734]
[proc_count_in_win_lf-3=from0to2d01267 : 888,3767]
[win_title_prot_words=1 : 766,2698]
[win_title_prot_words-1=1 : 766,2702]
: p=11,np=11,u=11,cx=63,c=2.33,s=11 # 484054

# Rule 10
<-- [process_name=netscape : 522,1761]
[prot_words_chars-1=from7d5to8d5 : 410,1402]
[prot_words_chars-3=from0to7d5..from7d5to8d5 : 584,2501]
[proc_count_in_win_lf=from2d01267to3d51143 : 302,1804]
[proc_count_in_win_lf-1=from0to2d01267 : 883,3720]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 303,1829]
[proc_count_in_win_lf-3=from0to2d01267 : 888,3767]
[win_title_prot_words=1 : 766,2698]
: p=10,np=10,u=8,cx=56,s=10 # 483860

# Rule 11
<-- [process_name=netscape : 522,1761]
[prot_words_chars-1=from7d5to8d5 : 410,1402]
[proc_count_in_win_lf=from2d01267to3d51143 : 302,1804]
[proc_count_in_win_lf-1=from0to2d01267 : 883,3720]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 303,1829]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 303,1833]
[win_title_prot_words=1 : 766,2698]
[win_title_prot_words-2=1..3 : 1311,5955]
[win_title_prot_words-3=1 : 767,2684]
: p=10,np=10,u=9,cx=63,s=10 # 483916

```

```

# Rule 12
<-- [process_name=netscape : 522,1761]
    [process_name-1=netscape : 528,1761]
    [prot_words_chars=from7d5to8d5 : 410,1402]
    [prot_words_chars-1=from7d5to8d5 : 410,1402]
    [prot_words_chars-2=from7d5to8d5 : 410,1401]
    [prot_words_chars-3=from7d5to8d5 : 410,1397]
    [proc_count_in_win_lf=from0to2d01267 : 875,3703]
    [proc_count_in_win_lf-1=from2d01267to3d51143 : 302,1809]
    [proc_count_in_win_lf-2=from0to2d01267 : 887,3734]
    [proc_count_in_win_lf-3=from0to2d01267 : 888,3767]
    : p=10,np=10,u=10,cx=70,s=10 # 484087

# Rule 13
<-- [process_name=netscape : 522,1761]
    [process_name-3=netscape : 531,1750]
    [prot_words_chars-1=from7d5to8d5 : 410,1402]
    [prot_words_chars-2=from7d5to8d5 : 410,1401]
    [proc_count_in_win_lf=from2d01267to3d51143 : 302,1804]
    [proc_count_in_win_lf-1=from0to2d01267 : 883,3720]
    [proc_count_in_win_lf-2=from0to2d01267 : 887,3734]
    [proc_count_in_win_lf-3=from2d01267to3d51143 : 303,1833]
    [win_title_prot_words=1 : 766,2698]
    [win_title_prot_words-3=1..3 : 1310,5963]
    : p=10,np=10,u=10,cx=70,s=10 # 484152

# Rule 14
<-- [process_name=netscape : 522,1761]
    [process_name-3=netscape : 531,1750]
    [prot_words_chars-2=from7d5to8d5 : 410,1401]
    [proc_count_in_win_lf=from2d01267to3d51143 : 302,1804]
    [proc_count_in_win_lf-1=from2d01267to3d51143 : 302,1809]
    [proc_count_in_win_lf-2=from2d01267to3d51143 : 303,1829]
    [proc_count_in_win_lf-3=from0to2d01267 : 888,3767]
    [win_title_prot_words=1 : 766,2698]
    [win_title_prot_words-1=1 : 766,2702]
    : p=9,np=9,u=9,cx=63,s=9 # 483897

# Rule 15
<-- [process_name=netscape : 522,1761]
    [process_name-2=netscape : 532,1755]
    [prot_words_chars=from7d5to8d5 : 410,1402]
    [prot_words_chars-1=from7d5to8d5 : 410,1402]
    [prot_words_chars-3=from7d5to8d5 : 410,1397]
    [proc_count_in_win_lf=from0to2d01267 : 875,3703]
    [proc_count_in_win_lf-1=from0to2d01267 : 883,3720]
    [proc_count_in_win_lf-2=from2d01267to3d51143 : 303,1829]
    [proc_count_in_win_lf-3=from0to2d01267 : 888,3767]
    [win_title_prot_words-2=1 : 767,2693]
    : p=8,np=8,u=8,cx=70,s=8 # 484141

# Rule 16
<-- [process_name=winword : 139,528]
    [process_name-1=explorer : 327,635]
    [prot_words_chars-2=from0to7d5..from8d5to24 : 1279,5083]
    [proc_count_in_win_lf=from0to2d01267 : 875,3703]
    [win_title_prot_words=2 : 308,1229]
    [win_title_prot_words-1=1 : 766,2702]
    : p=7,np=7,u=2,cx=42,c=1.67,s=7 # 483870

# Rule 17
<-- [process_name=netscape : 522,1761]
    [process_name-1=netscape : 528,1761]

```

```

[process_name-2=outlook : 458,2638]
[prot_words_chars-2=from8d5to24 : 695,2577]
[proc_count_in_win_lf=from0to2d01267 : 875,3703]
[proc_count_in_win_lf-1=from0to2d01267 : 883,3720]
[win_title_prot_words=1 : 766,2698]
: p=7,np=7,u=6,cx=49,s=7 # 484080

# Rule 18
<-- [process_name=netscape : 522,1761]
[process_name-2=netscape : 532,1755]
[prot_words_chars=from7d5to8d5 : 410,1402]
[proc_count_in_win_lf=from0to2d01267 : 875,3703]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 302,1809]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 303,1829]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 303,1833]
[win_title_prot_words-1=1 : 766,2702]
[win_title_prot_words-3=1 : 767,2684]
: p=7,np=7,u=7,cx=63,s=7 # 484202

# Rule 19
<-- [process_name=netscape : 522,1761]
[prot_words_chars=from7d5to8d5 : 410,1402]
[prot_words_chars-2=from7d5to8d5 : 410,1401]
[proc_count_in_win_lf=from0to2d01267 : 875,3703]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 302,1809]
[proc_count_in_win_lf-2=from0to2d01267 : 887,3734]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 303,1833]
[win_title_prot_words-1=1 : 766,2702]
[win_title_prot_words-2=1 : 767,2693]
[win_title_prot_words-3=1 : 767,2684]
: p=6,np=6,u=6,cx=70,s=6 # 484189

# Rule 20
<-- [process_name=netscape : 522,1761]
[process_name-3=netscape : 531,1750]
[prot_words_chars=from0to7d5..from7d5to8d5 : 583,2495]
[prot_words_chars-2=from7d5to8d5 : 410,1401]
[proc_count_in_win_lf=from0to2d01267 : 875,3703]
[proc_count_in_win_lf-1=from3d51143to4d64917 : 59,386]
[proc_count_in_win_lf-2=from0to2d01267 : 887,3734]
[proc_count_in_win_lf-3=from0to2d01267 : 888,3767]
: p=5,np=5,u=5,cx=56,s=5 # 484088

```

...

Output_Hypotheses User3

```

{
# -- This learning took =
# -- System (CPU) time = 2.94
# -- User (Total) time = 3
# -- Number of rules in the cover = 27
# -- Number of conditions = 161
# -- Complexity for this cover = 1179
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 51
negative_events = 8990
[user=user3]
# Rule 1
<-- [process_name=photoshp,pstores,sdstat : 7,87]
[win_title_prot_words=1..3 : 25,7251]
: p=5,np=5,u=3,cx=18,s=5 # 504066

```

```

# Rule 2
<-- [process_name=explorer,rundll32 : 39,977]
    [process_name-1=explorer,rundll32 : 36,938]
    [process_name-2=explorer : 33,906]
    [process_name-3=explorer : 29,885]
    [prot_words_chars-1=lte0 : 25,1093]
    [prot_words_chars-3=from8d5to24 : 23,3264]
    [proc_count_in_win_lf=from0to2d01267 : 23,4555]
    [proc_count_in_win_lf-1=from0to2d01267..from2d01267to3d51143 :
24,6690]
    [win_title_prot_words=0 : 26,1074]
    [win_title_prot_words-2=0 : 24,1104]
    : p=3,np=3,u=3,cx=74,s=3 # 504065

# Rule 3
<-- [process_name-1=photoshp,pstores : 5,87]
    [win_title_prot_words-1=1..3 : 21,7239]
    : p=4,np=3,u=2,cx=16,s=4 # 504077

# Rule 4
<-- [process_name=explorer : 34,970]
    [process_name-1=explorer : 31,931]
    [prot_words_chars=from8d5to24 : 25,3269]
    [proc_count_in_win_lf=from0to2d01267 : 23,4555]
    [win_opened=lte16 : 51,8408]
    [win_title_prot_words=2 : 13,1524]
    [win_title_prot_words-1=0 : 25,1093]
    [win_title_prot_words-3=0 : 25,1116]
    : p=2,np=2,u=1,cx=56,s=2 # 504071

# Rule 5
<-- [process_name=explorer,sdstat : 36,970]
    [process_name-1=wscript : 5,112]
    [prot_words_chars=lte0 : 26,1074]
    [win_title_prot_words-1=4 : 5,566]
    : p=3,np=2,u=2,cx=30,s=3 # 504070

# Rule 6
<-- [process_name=explorer : 34,970]
    [process_name-1=explorer : 31,931]
    [process_name-2=explorer : 33,906]
    [prot_words_chars=from8d5to24 : 25,3269]
    [proc_count_in_win_lf=from0to2d01267 : 23,4555]
    [proc_count_in_win_lf-1=from0to2d01267 : 21,4582]
    [win_title_prot_words=2 : 13,1524]
    [win_title_prot_words-1=3 : 5,2273]
    : p=1,np=1,u=1,cx=56,c=1.62,s=1 # 504063

# Rule 7
<-- [process_name=explorer : 34,970]
    [prot_words_chars=from8d5to24 : 25,3269]
    [proc_count_in_win_lf=from0to2d01267 : 23,4555]
    [proc_count_in_win_lf-1=lte0 : 26,1702]
    [win_title_prot_words=2 : 13,1524]
    [win_title_prot_words-1=1 : 6,3462]
    [win_title_prot_words-2=0 : 24,1104]
    : p=1,np=1,u=1,cx=49,c=1.57,s=1 # 504064

# Rule 8
<-- [process_name=explorer : 34,970]
    [process_name-2=explorer : 33,906]
    [delta_time_new_window-2=from10500to11000 : 2,6]

```

```

[proc_count_in_win_lf=lte0 : 23,1741]
[proc_count_in_win_lf-1=lte0 : 26,1702]
[win_title_prot_words=1 : 7,3457]
[win_title_prot_words-1=0 : 25,1093]
: p=1,np=1,u=1,cx=49,c=1.86,s=1 # 504067

# Rule 9
<-- [process_name=explorer : 34,970]
[delta_time_new_window=from13000to24000 : 1,31]
[proc_count_in_win_lf=lte0 : 23,1741]
[win_title_prot_words=2 : 13,1524]
: p=1,np=1,u=1,cx=28,s=1 # 504068

# Rule 10
<-- [process_name=explorer : 34,970]
[process_name-1=explorer : 31,931]
[proc_count_in_win_lf=lte0 : 23,1741]
[win_opened=lte16 : 51,8408]
[win_title_prot_words=2 : 13,1524]
[win_title_prot_words-1=0 : 25,1093]
[win_title_prot_words-2=0 : 24,1104]
: p=1,np=1,u=1,cx=49,s=1 # 504069

# Rule 11
<-- [process_name=explorer : 34,970]
[prot_words_chars-3=lte0 : 25,1116]
[proc_count_in_win_lf=from0to2d01267 : 23,4555]
[proc_count_in_win_lf-1=lte0 : 26,1702]
[win_opened=lte16 : 51,8408]
[win_title_prot_words=0 : 26,1074]
[win_title_prot_words-1=1 : 6,3462]
[win_title_prot_words-2=2 : 15,1519]
: p=1,np=1,u=1,cx=56,s=1 # 504078

# Rule 12
<-- [process_name=explorer : 34,970]
[process_name-1=explorer : 31,931]
[proc_count_in_win_lf=lte0 : 23,1741]
[proc_count_in_win_lf-1=from0to2d01267 : 21,4582]
[proc_count_in_win_lf-2=lte0 : 27,1657]
[win_opened=lte16 : 51,8408]
[win_title_prot_words=1 : 7,3457]
[win_title_prot_words-1=2 : 10,1504]
: p=1,np=1,u=1,cx=56,c=1.62,s=1 # 504080

# Rule 13
<-- [process_name=explorer : 34,970]
[process_name-1=csrss : 2,194]
[process_name-2=pstores : 2,87]
[win_title_prot_words=0 : 26,1074]
[win_title_prot_words-2=1 : 6,3454]
: p=1,np=1,u=1,cx=35,c=2,s=1 # 504085

# Rule 14
<-- [process_name=explorer : 34,970]
[process_name-1=explorer : 31,931]
[proc_count_in_win_lf=from0to2d01267 : 23,4555]
[proc_count_in_win_lf-1=lte0 : 26,1702]
[win_title_prot_words=2 : 13,1524]
[win_title_prot_words-1=3 : 5,2273]
: p=1,np=1,u=1,cx=42,s=1 # 504086

# Rule 15
<-- [process_name=explorer : 34,970]

```

```

[delta_time_new_window=gt30000 : 3,0]
[win_title_prot_words=3 : 5,2270]
  : p=2,np=1,u=1,cx=21,s=2 # 504079

# Rule 16
<-- [process_name=explorer : 34,970]
    [process_name-1=explorer : 31,931]
    [process_name-3=csrss : 3,199]
    [prot_words_chars-2=lte0 : 24,1104]
    [proc_count_in_win_lf=lte0 : 23,1741]
    [proc_count_in_win_lf-1=lte0 : 26,1702]
    [win_opened=lte16 : 51,8408]
    [win_title_prot_words=1 : 7,3457]
    [win_title_prot_words-1=2 : 10,1504]
      : p=1,u=1,cx=63,s=1 # 504075

# Rule 17
<-- [process_name=explorer : 34,970]
    [process_name-1=explorer : 31,931]
    [prot_words_chars-2=lte0 : 24,1104]
    [proc_count_in_win_lf=lte0 : 23,1741]
    [proc_count_in_win_lf-1=from0to2d01267 : 21,4582]
    [win_opened=lte16 : 51,8408]
    [win_title_prot_words=1 : 7,3457]
    [win_title_prot_words-1=2 : 10,1504]
      : p=1,u=1,cx=56,s=1 # 504081

# Rule 18
<-- [process_name=explorer : 34,970]
    [prot_words_chars-3=from8d5to24 : 23,3264]
    [proc_count_in_win_lf=from0to2d01267 : 23,4555]
    [win_title_prot_words=3 : 5,2270]
    [win_title_prot_words-1=1 : 6,3462]
      : p=1,u=1,cx=35,s=1 # 504088

# Rule 19
<-- [process_name=explorer : 34,970]
    [process_name-1=explorer : 31,931]
    [process_name-2=csrss,explorer : 35,1013]
    [process_name-3=explorer,wscript : 37,927]
    [prot_words_chars=lte0 : 26,1074]
    [prot_words_chars-1=from8d5to24 : 21,3248]
    [proc_count_in_win_lf=from0to2d01267 : 23,4555]
    [win_opened=lte16 : 51,8408]
    [win_title_prot_words-1=2 : 10,1504]
      : p=2,u=2,cx=67,s=2 # 504061

# Rule 20
<-- [process_name=explorer,photoshp : 37,970]
    [prot_words_chars=from8d5to24 : 25,3269]
    [proc_count_in_win_lf=lte0 : 23,1741]
    [win_title_prot_words=3 : 5,2270]
    [win_title_prot_words-1=0..1 : 31,4555]
    [win_title_prot_words-2=2 : 15,1519]
      : p=2,u=1,cx=44,s=2 # 504074

...

Output_Hypotheses User4
{
  # -- This learning took =
  # -- System (CPU) time = 62.3
  # -- User (Total) time = 62
  # -- Number of rules in the cover = 711

```



```

# -- Number of conditions          = 5046
# -- Complexity for this cover     = 35806
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events          = 1675
negative_events          = 7366
positive_distinct_events = 1524
negative_distinct_events = 6618
[user=user4]
# Rule 1
<-- [process_name=outlook : 602,2418]
    [process_name-1=outlook : 605,2453]
    [process_name-3=outlook : 606,2535]
    [prot_words_chars-2=from8d5to24 : 589,2683]
    [proc_count_in_win_lf=from0to2d01267 : 769,3809]
    [proc_count_in_win_lf-1=from0to2d01267 : 770,3833]
    [proc_count_in_win_lf-2=from0to2d01267 : 767,3854]
    [proc_count_in_win_lf-3=from0to2d01267 : 773,3882]
    [win_title_prot_words=3 : 249,2026]
    [win_title_prot_words-1=1 : 943,2525]
    [win_title_prot_words-2=3 : 247,2025]
    [win_title_prot_words-3=1 : 941,2510]
    : p=21,np=21,u=21,cx=84,c=2.5,s=21 # 915330

# Rule 2
<-- [process_name=outlook : 602,2418]
    [process_name-1=outlook : 605,2453]
    [process_name-3=outlook : 606,2535]
    [prot_words_chars-2=from0to7d5 : 337,942]
    [prot_words_chars-3=from8d5to24 : 588,2699]
    [proc_count_in_win_lf=from0to2d01267 : 769,3809]
    [proc_count_in_win_lf-1=from0to2d01267 : 770,3833]
    [proc_count_in_win_lf-2=from0to2d01267 : 767,3854]
    [proc_count_in_win_lf-3=from0to2d01267 : 773,3882]
    [win_title_prot_words=1 : 942,2522]
    [win_title_prot_words-1=3 : 250,2028]
    : p=18,np=18,u=18,cx=77,s=18 # 915453

# Rule 3
<-- [process_name=netscape : 674,1609]
    [process_name-1=photoed,powerpnt,winword : 269,638]
    [process_name-2=csrss,photoed,powerpnt,winword : 302,710]
    [prot_words_chars=from7d5to8d5 : 574,1238]
    : p=21,np=18,u=6,cx=38,c=1.75,s=21 # 915247

# Rule 4
<-- [process_name=netscape : 674,1609]
    [process_name-1=netscape : 673,1616]
    [process_name-3=netscape : 671,1610]
    [prot_words_chars-2=from7d5to8d5 : 574,1237]
    [proc_count_in_win_lf=from2d01267to3d51143 : 539,1567]
    [proc_count_in_win_lf-1=from2d01267to3d51143 : 539,1572]
    [proc_count_in_win_lf-2=from0to2d01267 : 767,3854]
    [proc_count_in_win_lf-3=from0to2d01267 : 773,3882]
    [win_title_prot_words=1 : 942,2522]
    [win_title_prot_words-3=1 : 941,2510]
    : p=12,np=12,u=12,cx=70,c=2.5,s=12 # 915464

# Rule 5
<-- [process_name=netscape,photoed : 678,1643]
    [process_name-1=netscape : 673,1616]
    [prot_words_chars-1=from7d5to8d5 : 574,1238]
    [proc_count_in_win_lf=from0to2d01267 : 769,3809]

```

```

[proc_count_in_win_lf-1=from0to2d01267..from2d01267to3d51143 :
1309,5405]
[proc_count_in_win_lf-2=from2d01267to3d51143..from4d64917to5d56641 :
715,1969]
[win_opened=lte16 : 1675,6784]
[win_title_prot_words=0 : 159,941]
: p=13,np=12,u=7,cx=58,s=13 # 915222

# Rule 6
<-- [process_name=netscape : 674,1609]
[proc_count_in_win_lf=from2d01267to3d51143 : 539,1567]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 539,1572]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 542,1590]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 543,1593]
[win_title_prot_words=1 : 942,2522]
[win_title_prot_words-2=1 : 942,2518]
[win_title_prot_words-3=1 : 941,2510]
: p=13,np=12,u=12,cx=56,s=13 # 915288

# Rule 7
<-- [process_name=netscape : 674,1609]
[process_name-3=netscape : 671,1610]
[prot_words_chars-2=from7d5to8d5 : 574,1237]
[proc_count_in_win_lf=from2d01267to3d51143 : 539,1567]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 539,1572]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 542,1590]
[proc_count_in_win_lf-3=from0to2d01267 : 773,3882]
[win_title_prot_words=1 : 942,2522]
: p=11,np=11,u=11,cx=56,s=11 # 915239

# Rule 8
<-- [process_name=outlook : 602,2418]
[delta_time_new_window-2=lte10500 : 1672,7313]
[prot_words_chars-1=from0to7d5 : 336,939]
[proc_count_in_win_lf=from0to2d01267 : 769,3809]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 539,1572]
[proc_count_in_win_lf-2=from0to2d01267 : 767,3854]
[proc_count_in_win_lf-3=from2d01267to3d51143..from3d51143to4d64917 :
688,1896]
[win_title_prot_words=3 : 249,2026]
[win_title_prot_words-2=3 : 247,2025]
: p=9,np=9,u=9,cx=63,c=2.22,s=9 # 915158

# Rule 9
<-- [process_name=netscape : 674,1609]
[process_name-3=netscape : 671,1610]
[prot_words_chars-1=from7d5to8d5 : 574,1238]
[prot_words_chars-2=from7d5to8d5 : 574,1237]
[prot_words_chars-3=from7d5to8d5 : 574,1233]
[proc_count_in_win_lf=from2d01267to3d51143 : 539,1567]
[proc_count_in_win_lf-1=from0to2d01267 : 770,3833]
[proc_count_in_win_lf-2=from0to2d01267 : 767,3854]
[proc_count_in_win_lf-3=from0to2d01267 : 773,3882]
[win_title_prot_words=1 : 942,2522]
: p=9,np=9,u=9,cx=70,c=2.5,s=9 # 915162

# Rule 10
<-- [process_name=netscape : 674,1609]
[process_name-1=netscape : 673,1616]
[process_name-3=netscape : 671,1610]
[prot_words_chars-2=from7d5to8d5 : 574,1237]
[prot_words_chars-3=from7d5to8d5 : 574,1233]
[proc_count_in_win_lf=from2d01267to3d51143 : 539,1567]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 539,1572]

```

```

[proc_count_in_win_lf-2=from0to2d01267 : 767,3854]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 543,1593]
[win_title_prot_words=1 : 942,2522]
: p=9,np=9,u=9,cx=70,c=2.5,s=9 # 915373

# Rule 11
<-- [process_name=winword : 177,490]
[prot_words_chars=from0to7d5 : 335,931]
[proc_count_in_win_lf=from0to2d01267..from2d01267to3d51143 :
1308,5376]
[proc_count_in_win_lf-1=from0to2d01267..from4d64917to5d56641 :
1482,5779]
[win_title_prot_words-2=2 : 318,1216]
: p=10,np=9,u=3,cx=35,c=1.6,s=10 # 915169

# Rule 12
<-- [process_name=netscape : 674,1609]
[prot_words_chars-2=from7d5to8d5 : 574,1237]
[proc_count_in_win_lf=from2d01267to3d51143 : 539,1567]
[proc_count_in_win_lf-1=from0to2d01267 : 770,3833]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 542,1590]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 543,1593]
[win_title_prot_words=1 : 942,2522]
[win_title_prot_words-1=0..1 : 1101,3485]
[win_title_prot_words-3=1 : 941,2510]
: p=10,np=9,u=7,cx=63,s=10 # 915224

# Rule 13
<-- [process_name=netscape : 674,1609]
[process_name-1=outlook,powerpnt : 694,2568]
[proc_count_in_win_lf=lte0..from2d01267to3d51143 : 1484,6964]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 539,1572]
[win_title_prot_words=1 : 942,2522]
[win_title_prot_words-1=0..1 : 1101,3485]
: p=8,np=8,u=3,cx=44,s=8 # 915185

# Rule 14
<-- [process_name=netscape : 674,1609]
[process_name-3=netscape : 671,1610]
[prot_words_chars-1=from7d5to8d5 : 574,1238]
[proc_count_in_win_lf=from2d01267to3d51143 : 539,1567]
[proc_count_in_win_lf-1=from3d51143to4d64917..from4d64917to5d56641 :
173,374]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 542,1590]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 543,1593]
[win_title_prot_words=1 : 942,2522]
: p=8,np=8,u=8,cx=56,s=8 # 915193

# Rule 15
<-- [process_name=netscape : 674,1609]
[process_name-1=netscape : 673,1616]
[prot_words_chars=from7d5to8d5 : 574,1238]
[prot_words_chars-3=from7d5to8d5 : 574,1233]
[proc_count_in_win_lf=from0to2d01267 : 769,3809]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 539,1572]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 542,1590]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 543,1593]
[win_title_prot_words-1=1 : 943,2525]
: p=8,np=8,u=8,cx=63,c=2.22,s=8 # 915353

# Rule 16
<-- [process_name=outlook : 602,2418]
[process_name-1=outlook : 605,2453]
[process_name-2=netscape,outlook : 1278,4018]

```

```

[delta_time_new_window-3=lte10500 : 1672,7316]
[prot_words_chars-1=from8d5to24 : 590,2679]
[prot_words_chars-3=from7d5to8d5..from8d5to24 : 1162,3932]
[proc_count_in_win_lf=from2d01267to3d51143 : 539,1567]
[proc_count_in_win_lf-1=from0to2d01267 : 770,3833]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 542,1590]
[proc_count_in_win_lf-3=from0to2d01267 : 773,3882]
[win_title_prot_words=1 : 942,2522]
[win_title_prot_words-1=3 : 250,2028]
[win_title_prot_words-2=1 : 942,2518]
: p=8,np=8,u=7,cx=93,s=8 # 915461

# Rule 17
<-- [process_name=netscape : 674,1609]
[proc_count_in_win_lf=from3d51143to4d64917..from7d12078to7d79811 :
191,401]
[proc_count_in_win_lf-1=from3d51143to4d64917 : 145,300]
[proc_count_in_win_lf-2=from3d51143to4d64917..from4d64917to5d56641 :
173,379]
[win_title_prot_words=1 : 942,2522]
: p=7,np=7,u=6,cx=35,c=1.6,s=7 # 915167

# Rule 18
<-- [process_name=powerpnt : 88,203]
[process_name-1=netscape : 673,1616]
[prot_words_chars=from8d5to24 : 594,2700]
[prot_words_chars-1=from7d5to8d5 : 574,1238]
[proc_count_in_win_lf=from0to2d01267 : 769,3809]
: p=7,np=7,u=1,cx=35,c=1.4,s=7 # 915201

# Rule 19
<-- [process_name=netscape : 674,1609]
[prot_words_chars-1=lte0 : 158,960]
[proc_count_in_win_lf=lte0..from0to2d01267 : 945,5397]
[proc_count_in_win_lf-1=from0to2d01267 : 770,3833]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 542,1590]
[win_title_prot_words=0 : 159,941]
[win_title_prot_words-2=1 : 942,2518]
: p=7,np=7,u=5,cx=49,s=7 # 915248

# Rule 20
<-- [process_name=netscape : 674,1609]
[process_name-1=netscape : 673,1616]
[proc_count_in_win_lf=from2d01267to3d51143 : 539,1567]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 539,1572]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 542,1590]
[proc_count_in_win_lf-3=from3d51143to4d64917 : 145,303]
[win_title_prot_words=1 : 942,2522]
: p=7,np=7,u=7,cx=49,s=7 # 915477

...

Output_Hypotheses User5
{
# -- This learning took =
# -- System (CPU) time = 18.9
# -- User (Total) time = 19
# -- Number of rules in the cover = 286
# -- Number of conditions = 2117
# -- Complexity for this cover = 14885
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 503

```

```

negative_events          = 8538
positive_distinct_events = 411
negative_distinct_events = 7731
[user=user5]
# Rule 1
<-- [process_name=netscape : 311,1972]
[process_name-2=netscape : 310,1977]
[prot_words_chars=from7d5to8d5 : 278,1534]
[prot_words_chars-3=from7d5to8d5 : 277,1530]
[proc_count_in_win_lf=from0to2d01267 : 235,4343]
[proc_count_in_win_lf-1=from0to2d01267 : 241,4362]
[proc_count_in_win_lf-2=from0to2d01267 : 241,4380]
[proc_count_in_win_lf-3=from0to2d01267 : 245,4410]
[win_opened=lte16 : 503,7956]
[win_title_prot_words-1=1 : 307,3161]
[win_title_prot_words-2=1 : 306,3154]
: p=33,np=33,u=33,cx=77,s=33 # 1063337

# Rule 2
<-- [process_name=netscape : 311,1972]
[process_name-1=netscape : 311,1978]
[proc_count_in_win_lf=from2d01267to3d51143 : 139,1967]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 140,1971]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 144,1988]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 145,1991]
[win_title_prot_words=1 : 305,3159]
[win_title_prot_words-2=1 : 306,3154]
[win_title_prot_words-3=1 : 304,3147]
: p=15,np=15,u=15,cx=63,s=15 # 1063474

# Rule 3
<-- [process_name=netscape : 311,1972]
[process_name-3=netscape : 307,1974]
[prot_words_chars-2=from7d5to8d5 : 279,1532]
[proc_count_in_win_lf=from2d01267to3d51143 : 139,1967]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 140,1971]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 144,1988]
[proc_count_in_win_lf-3=lte0..from0to2d01267 : 314,5988]
[win_title_prot_words=1 : 305,3159]
[win_title_prot_words-1=1 : 307,3161]
: p=13,np=12,u=13,cx=63,s=13 # 1063342

# Rule 4
<-- [process_name=netscape : 311,1972]
[process_name-2=netscape : 310,1977]
[prot_words_chars=from7d5to8d5 : 278,1534]
[prot_words_chars-1=from7d5to8d5 : 279,1533]
[proc_count_in_win_lf=from0to2d01267 : 235,4343]
[proc_count_in_win_lf-1=from0to2d01267 : 241,4362]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 144,1988]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 145,1991]
[win_title_prot_words-1=1 : 307,3161]
[win_title_prot_words-2=1 : 306,3154]
: p=11,np=11,u=11,cx=70,c=2.2,s=11 # 1063323

# Rule 5
<-- [process_name=netscape : 311,1972]
[process_name-1=netscape : 311,1978]
[process_name-3=netscape : 307,1974]
[prot_words_chars-2=from7d5to8d5 : 279,1532]
[prot_words_chars-3=from7d5to8d5 : 277,1530]
[proc_count_in_win_lf=from2d01267to3d51143 : 139,1967]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 140,1971]
[proc_count_in_win_lf-2=from0to2d01267 : 241,4380]

```

```

[proc_count_in_win_lf-3=from0to2d01267 : 245,4410]
[win_title_prot_words=1 : 305,3159]
: p=11,np=11,u=11,cx=70,c=2.5,s=11 # 1063401

# Rule 6
<-- [process_name=netscape : 311,1972]
[process_name-1=netscape : 311,1978]
[prot_words_chars=from7d5to8d5 : 278,1534]
[prot_words_chars-1=from7d5to8d5 : 279,1533]
[proc_count_in_win_lf=from0to2d01267 : 235,4343]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 140,1971]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 144,1988]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 145,1991]
[win_title_prot_words-3=1 : 304,3147]
: p=10,np=10,u=10,cx=63,s=10 # 1063368

# Rule 7
<-- [process_name=netscape : 311,1972]
[process_name-2=netscape : 310,1977]
[prot_words_chars=from7d5to8d5 : 278,1534]
[prot_words_chars-1=from7d5to8d5 : 279,1533]
[prot_words_chars-2=from7d5to8d5 : 279,1532]
[prot_words_chars-3=from7d5to8d5 : 277,1530]
[proc_count_in_win_lf=from0to2d01267 : 235,4343]
[proc_count_in_win_lf-1=from0to2d01267 : 241,4362]
[proc_count_in_win_lf-2=from0to2d01267 : 241,4380]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 145,1991]
[win_opened=ltel6 : 503,7956]
[win_title_prot_words-1=1 : 307,3161]
: p=10,np=10,u=10,cx=84,s=10 # 1063385

# Rule 8
<-- [process_name=netscape : 311,1972]
[process_name-1=netscape : 311,1978]
[prot_words_chars=from7d5to8d5 : 278,1534]
[prot_words_chars-1=from7d5to8d5 : 279,1533]
[prot_words_chars-3=from7d5to8d5 : 277,1530]
[proc_count_in_win_lf=from0to2d01267 : 235,4343]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 140,1971]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 144,1988]
[proc_count_in_win_lf-3=from0to2d01267 : 245,4410]
: p=8,np=8,u=8,cx=63,c=2.22,s=8 # 1063364

# Rule 9
<-- [process_name=netscape : 311,1972]
[process_name-1=netscape : 311,1978]
[process_name-2=netscape : 310,1977]
[prot_words_chars-2=from7d5to8d5 : 279,1532]
[proc_count_in_win_lf=from2d01267to3d51143 : 139,1967]
[proc_count_in_win_lf-1=from2d01267to3d51143..from4d64917to5d56641 :
176,2482]
[proc_count_in_win_lf-2=from0to2d01267 : 241,4380]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 145,1991]
[win_title_prot_words=1 : 305,3159]
[win_title_prot_words-3=1 : 304,3147]
: p=8,np=8,u=8,cx=70,c=2.4,s=8 # 1063371

# Rule 10
<-- [process_name=netscape : 311,1972]
[prot_words_chars-1=from7d5to8d5 : 279,1533]
[proc_count_in_win_lf=from2d01267to3d51143 : 139,1967]
[proc_count_in_win_lf-1=from0to2d01267 : 241,4362]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 144,1988]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 145,1991]

```

```

[win_title_prot_words=1 : 305,3159]
[win_title_prot_words-2=1 : 306,3154]
[win_title_prot_words-3=1 : 304,3147]
: p=8,np=8,u=8,cx=63,s=8 # 1063415

# Rule 11
<-- [process_name=netscape : 311,1972]
[process_name-3=netscape : 307,1974]
[prot_words_chars-1=from7d5to8d5 : 279,1533]
[prot_words_chars-2=from7d5to8d5 : 279,1532]
[prot_words_chars-3=from7d5to8d5 : 277,1530]
[proc_count_in_win_lf=from2d01267to3d51143 : 139,1967]
[proc_count_in_win_lf-1=lte0..from0to2d01267 : 322,6009]
[proc_count_in_win_lf-2=from0to2d01267 : 241,4380]
[proc_count_in_win_lf-3=from0to2d01267 : 245,4410]
[win_title_prot_words=1 : 305,3159]
: p=8,np=7,u=7,cx=70,s=8 # 1063473

# Rule 12
<-- [process_name=netscape : 311,1972]
[process_name-2=netscape : 310,1977]
[process_name-3=netscape : 307,1974]
[prot_words_chars-1=from7d5to8d5 : 279,1533]
[prot_words_chars-2=from7d5to8d5 : 279,1532]
[proc_count_in_win_lf=from2d01267to3d51143 : 139,1967]
[proc_count_in_win_lf-1=lte0..from0to2d01267 : 322,6009]
[proc_count_in_win_lf-2=from0to2d01267 : 241,4380]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 145,1991]
[win_title_prot_words=1 : 305,3159]
[win_title_prot_words-3=1 : 304,3147]
: p=7,np=6,u=6,cx=77,c=2.55,s=7 # 1063426

# Rule 13
<-- [process_name=netscape : 311,1972]
[process_name-1=netscape : 311,1978]
[prot_words_chars=from7d5to8d5 : 278,1534]
[prot_words_chars-1=from7d5to8d5 : 279,1533]
[prot_words_chars-3=from7d5to8d5 : 277,1530]
[proc_count_in_win_lf=from0to2d01267 : 235,4343]
[proc_count_in_win_lf-1=from0to2d01267 : 241,4362]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 144,1988]
[proc_count_in_win_lf-3=from0to2d01267 : 245,4410]
[win_title_prot_words-1=1 : 307,3161]
[win_title_prot_words-2=1 : 306,3154]
[win_title_prot_words-3=1 : 304,3147]
: p=3,np=3,u=3,cx=84,s=3 # 1063355

# Rule 14
<-- [process_name=explorer : 36,968]
[process_name-1=netscape : 311,1978]
[process_name-2=explorer : 25,914]
[proc_count_in_win_lf=lte0 : 88,1676]
[win_title_prot_words=2 : 61,1476]
[win_title_prot_words-1=2 : 56,1458]
: p=3,np=3,u=3,cx=42,c=1.67,s=3 # 1063379

# Rule 15
<-- [process_name=netscape : 311,1972]
[process_name-1=netscape : 311,1978]
[process_name-2=netscape : 310,1977]
[proc_count_in_win_lf=from2d01267to3d51143 : 139,1967]
[proc_count_in_win_lf-1=lte0 : 81,1647]
[proc_count_in_win_lf-2=from0to2d01267 : 241,4380]
[win_title_prot_words=1 : 305,3159]

```

```

[win_title_prot_words-1=1 : 307,3161]
: p=3,np=3,u=1,cx=56,c=1.88,s=3 # 1063383

# Rule 16
<-- [process_name=netscape : 311,1972]
[prot_words_chars-1=from7d5to8d5 : 279,1533]
[prot_words_chars-3=from7d5to8d5 : 277,1530]
[proc_count_in_win_lf=from2d01267to3d51143 : 139,1967]
[proc_count_in_win_lf-1=from0to2d01267 : 241,4362]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 144,1988]
[proc_count_in_win_lf-3=from0to2d01267 : 245,4410]
[win_title_prot_words=1 : 305,3159]
[win_title_prot_words-2=1 : 306,3154]
: p=3,np=3,u=3,cx=63,s=3 # 1063389

# Rule 17
<-- [process_name=netscape : 311,1972]
[process_name-1=netscape : 311,1978]
[prot_words_chars=from7d5to8d5 : 278,1534]
[prot_words_chars-1=from7d5to8d5 : 279,1533]
[prot_words_chars-2=from7d5to8d5 : 279,1532]
[proc_count_in_win_lf=from0to2d01267 : 235,4343]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 140,1971]
[proc_count_in_win_lf-2=from0to2d01267 : 241,4380]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 145,1991]
[win_title_prot_words-2=1 : 306,3154]
[win_title_prot_words-3=1 : 304,3147]
: p=3,np=3,u=3,cx=77,s=3 # 1063430

# Rule 18
<-- [process_name=netscape : 311,1972]
[process_name-1=netscape : 311,1978]
[prot_words_chars=from7d5to8d5 : 278,1534]
[proc_count_in_win_lf=from0to2d01267 : 235,4343]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 140,1971]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 144,1988]
[proc_count_in_win_lf-3=from3d51143to4d64917 : 31,417]
: p=3,np=3,u=3,cx=49,s=3 # 1063456

# Rule 19
<-- [process_name=netscape : 311,1972]
[process_name-3=netscape : 307,1974]
[proc_count_in_win_lf=from2d01267to3d51143 : 139,1967]
[proc_count_in_win_lf-1=from3d51143to4d64917 : 28,417]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 144,1988]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 145,1991]
[win_title_prot_words=1 : 305,3159]
[win_title_prot_words-1=1 : 307,3161]
[win_title_prot_words-2=1 : 306,3154]
: p=3,np=3,u=3,cx=63,s=3 # 1063478

# Rule 20
<-- [process_name=outlook : 135,2885]
[prot_words_chars=from8d5to24 : 151,3143]
[prot_words_chars-1=from8d5to24 : 149,3120]
[proc_count_in_win_lf=lte0 : 88,1676]
[proc_count_in_win_lf-1=from0to2d01267 : 241,4362]
[proc_count_in_win_lf-2=from0to2d01267..from2d01267to3d51143 :
385,6368]
[win_title_prot_words=2 : 61,1476]
[win_title_prot_words-1=3 : 90,2188]
: p=3,np=3,u=3,cx=56,s=3 # 1063547

```

...

Output_Hypotheses User7

```

{
# -- This learning took =
# -- System (CPU) time = 23.3
# -- User (Total) time = 24
# -- Number of rules in the cover = 308
# -- Number of conditions = 2238
# -- Complexity for this cover = 15834
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 584
negative_events = 8457
positive_distinct_events = 567
negative_distinct_events = 7575
[user=user7]
# Rule 1
<-- [process_name=outlook : 379,2641]
[process_name-1=explorer : 92,870]
[prot_words_chars-1=from8d5to24 : 294,2975]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 87,2024]
[win_title_prot_words=3 : 144,2131]
[win_title_prot_words-1=2 : 115,1399]
: p=16,np=13,u=13,cx=42,c=1.67,s=16 # 1233901

# Rule 2
<-- [process_name=explorer : 97,907]
[process_name-1=outlook : 386,2672]
[process_name-2=mapisp32,outlook : 395,2712]
[prot_words_chars=from8d5to24 : 298,2996]
[prot_words_chars-1=from8d5to24 : 294,2975]
[prot_words_chars-2=lte0..from0to7d5 : 272,2135]
[proc_count_in_win_lf=from0to2d01267..from5d56641to7d12078 :
439,6834]
[proc_count_in_win_lf-1=from0to2d01267 : 337,4266]
[proc_count_in_win_lf-2=lte0..from0to2d01267 : 477,5828]
[proc_count_in_win_lf-3=from0to2d01267 : 348,4307]
[win_title_prot_words=2 : 119,1418]
: p=12,np=11,u=9,cx=79,c=2.09,s=12 # 1233917

# Rule 3
<-- [process_name=photoed : 39,86]
[prot_words_chars-1=from0to7d5 : 163,1112]
[proc_count_in_win_lf-1=lte0 : 140,1588]
: p=11,np=10,u=10,cx=21,c=1.67,s=11 # 1233993

# Rule 4
<-- [process_name=outlook : 379,2641]
[process_name-1=explorer : 92,870]
[process_name-2=explorer,outlook : 471,3477]
[prot_words_chars=from8d5to24 : 298,2996]
[proc_count_in_win_lf=from0to2d01267 : 332,4246]
[proc_count_in_win_lf-1=from0to2d01267..from2d01267to3d51143 :
424,6290]
[proc_count_in_win_lf-3=lte0 : 129,1518]
[win_title_prot_words=3 : 144,2131]
[win_title_prot_words-1=2 : 115,1399]
: p=7,np=7,u=1,cx=65,s=7 # 1233853

# Rule 5
<-- [process_name=outlook,photoed,services : 411,2649]
[prot_words_chars=lte0 : 102,998]
[prot_words_chars-1=from0to7d5 : 163,1112]
[proc_count_in_win_lf=from0to2d01267 : 332,4246]

```

```

[proc_count_in_win_lf-1=lte0..from0to2d01267 : 477,5854]
[proc_count_in_win_lf-2=lte0 : 133,1551]
[win_title_prot_words-1=1 : 199,3269]
[win_title_prot_words-2=0 : 109,1019]
: p=7,np=6,u=6,cx=60,s=7 # 1234025

# Rule 6
<-- [process_name=wscript : 15,88]
[proc_count_in_win_lf-1=from0to2d01267 : 337,4266]
[win_title_prot_words=4 : 18,540]
[win_title_prot_words-1=3 : 146,2132]
: p=5,np=5,u=5,cx=28,c=1.5,s=5 # 1233881

# Rule 7
<-- [process_name=outlook : 379,2641]
[process_name-1=outlook : 386,2672]
[process_name-3=outlook : 393,2748]
[prot_words_chars-2=from8d5to24 : 295,2977]
[proc_count_in_win_lf=from0to2d01267 : 332,4246]
[proc_count_in_win_lf-1=from0to2d01267 : 337,4266]
[proc_count_in_win_lf-2=from0to2d01267 : 344,4277]
[proc_count_in_win_lf-3=from0to2d01267 : 348,4307]
[win_title_prot_words=3 : 144,2131]
[win_title_prot_words-1=1 : 199,3269]
[win_title_prot_words-2=3 : 148,2124]
[win_title_prot_words-3=0..1 : 305,4287]
: p=5,np=5,u=5,cx=84,c=2.5,s=5 # 1233889

# Rule 8
<-- [process_name=mapisp32 : 17,86]
[proc_count_in_win_lf=lte0 : 145,1619]
[win_title_prot_words-2=2 : 117,1417]
: p=6,np=5,u=6,cx=21,s=6 # 1233948

# Rule 9
<-- [process_name=mapisp32,mspaint,photoed : 55,93]
[prot_words_chars-2=lte0 : 109,1019]
[proc_count_in_win_lf=lte0 : 145,1619]
[proc_count_in_win_lf-1=from0to2d01267 : 337,4266]
: p=12,np=5,u=8,cx=32,s=12 # 1234078

# Rule 10
<-- [process_name=outlook : 379,2641]
[process_name-2=outlook : 390,2706]
[process_name-3=winword,wscript : 65,651]
[proc_count_in_win_lf=from0to2d01267 : 332,4246]
[proc_count_in_win_lf-1=from0to2d01267 : 337,4266]
[win_title_prot_words=1 : 201,3263]
[win_title_prot_words-1=3 : 146,2132]
[win_title_prot_words-2=0..2 : 422,5700]
: p=4,np=4,u=3,cx=58,s=4 # 1233851

# Rule 11
<-- [process_name=outlook : 379,2641]
[process_name-1=outlook : 386,2672]
[process_name-2=explorer : 89,850]
[delta_time_new_window=lte10500 : 577,8401]
[proc_count_in_win_lf=from0to2d01267 : 332,4246]
[win_title_prot_words=3 : 144,2131]
[win_title_prot_words-1=3 : 146,2132]

```

```

: p=4,np=4,u=1,cx=49,s=4 # 1233895

# Rule 12
<-- [process_name=winword : 56,611]
[process_name-3=outlook : 393,2748]
[prot_words_chars-1=from0to7d5 : 163,1112]
[proc_count_in_win_lf-2=from0to2d01267 : 344,4277]
[win_title_prot_words=3 : 144,2131]
: p=4,np=4,u=2,cx=35,c=2.2,s=4 # 1233922

# Rule 13
<-- [process_name=explorer : 97,907]
[process_name-1=outlook : 386,2672]
[prot_words_chars=from8d5to24 : 298,2996]
[prot_words_chars-3=from8d5to24 : 299,2988]
[proc_count_in_win_lf=from0to2d01267 : 332,4246]
[proc_count_in_win_lf-1=from0to2d01267 : 337,4266]
[proc_count_in_win_lf-2=from2d01267to3d51143..from4d64917to5d56641 :
101,2583]
[proc_count_in_win_lf-3=from0to2d01267 : 348,4307]
[win_title_prot_words=2 : 119,1418]
[win_title_prot_words-1=3 : 146,2132]
[win_title_prot_words-2=1 : 196,3264]
: p=4,np=4,u=4,cx=77,s=4 # 1233965

# Rule 14
<-- [process_name=outlook : 379,2641]
[prot_words_chars-2=from0to7d5 : 163,1116]
[prot_words_chars-3=from0to7d5 : 163,1115]
[proc_count_in_win_lf=from0to2d01267 : 332,4246]
[proc_count_in_win_lf-1=from0to2d01267 : 337,4266]
[proc_count_in_win_lf-2=from0to2d01267 : 344,4277]
[proc_count_in_win_lf-3=from0to2d01267 : 348,4307]
[win_title_prot_words=1 : 201,3263]
[win_title_prot_words-1=1 : 199,3269]
: p=4,np=4,u=4,cx=63,s=4 # 1234017

# Rule 15
<-- [process_name=outlook : 379,2641]
[process_name-1=outlook : 386,2672]
[process_name-2=explorer : 89,850]
[prot_words_chars=from0to7d5 : 163,1103]
[proc_count_in_win_lf=from0to2d01267 : 332,4246]
[proc_count_in_win_lf-1=from0to2d01267 : 337,4266]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 87,2045]
[win_title_prot_words-2=2 : 117,1417]
: p=4,np=4,u=4,cx=56,c=2,s=4 # 1234044

# Rule 16
<-- [process_name=outlook : 379,2641]
[process_name-1=outlook : 386,2672]
[prot_words_chars=from8d5to24 : 298,2996]
[proc_count_in_win_lf=from0to2d01267 : 332,4246]
[proc_count_in_win_lf-1=from0to2d01267 : 337,4266]
[win_title_prot_words=3 : 144,2131]
[win_title_prot_words-1=1 : 199,3269]
[win_title_prot_words-2=2 : 117,1417]
[win_title_prot_words-3=0 : 109,1032]
: p=4,np=4,u=3,cx=63,c=1.89,s=4 # 1234095

# Rule 17
<-- [process_name=explorer,winword : 145,1439]
[process_name-1=lsass,smsmon32,wscript : 19,115]
[prot_words_chars=from8d5to24 : 298,2996]

```

```

[proc_count_in_win_lf=from0to2d01267 : 332,4246]
[proc_count_in_win_lf-3=from0to2d01267 : 348,4307]
[win_title_prot_words=2 : 119,1418]
: p=5,np=4,u=5,cx=48,c=1.67,s=5 # 1234053

# Rule 18
<-- [process_name=outlook : 379,2641]
[prot_words_chars-1=from0to7d5 : 163,1112]
[proc_count_in_win_lf=from0to2d01267 : 332,4246]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 87,2024]
[proc_count_in_win_lf-2=from0to2d01267 : 344,4277]
[proc_count_in_win_lf-3=from5d56641to7d12078 : 6,42]
[win_title_prot_words=3 : 144,2131]
[win_title_prot_words-2=3 : 148,2124]
: p=3,np=3,u=2,cx=56,s=3 # 1233862

# Rule 19
<-- [process_name=outlook,photoed : 410,2648]
[process_name-2=outlook,wscript : 400,2737]
[prot_words_chars-1=from0to7d5 : 163,1112]
[proc_count_in_win_lf=lte0 : 145,1619]
[proc_count_in_win_lf-3=lte0 : 129,1518]
[win_title_prot_words=0 : 102,998]
[win_title_prot_words-1=1 : 199,3269]
: p=3,np=3,u=2,cx=53,c=2,s=3 # 1233872

# Rule 20
<-- [process_name=outlook : 379,2641]
[process_name-1=outlook : 386,2672]
[process_name-3=explorer,wscript : 96,868]
[prot_words_chars=from0to7d5 : 163,1103]
[proc_count_in_win_lf=from0to2d01267 : 332,4246]
[proc_count_in_win_lf-1=from0to2d01267 : 337,4266]
[win_title_prot_words-1=1 : 199,3269]
: p=3,np=3,u=2,cx=51,c=1.86,s=3 # 1233905

...

Output_Hypotheses User8
{
# -- This learning took =
# -- System (CPU) time = 22.5
# -- User (Total) time = 23
# -- Number of rules in the cover = 328
# -- Number of conditions = 2406
# -- Complexity for this cover = 16934
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 515
negative_events = 8526
positive_distinct_events = 501
negative_distinct_events = 7641
[user=user8]
# Rule 1
<-- [process_name=netscape : 227,2056]
[prot_words_chars=from8d5to24 : 230,3064]
[prot_words_chars-2=from8d5to24 : 220,3052]
[proc_count_in_win_lf=from2d01267to3d51143..from3d51143to4d64917 :
138,2410]
[win_title_prot_words=2 : 158,1379]
: p=18,np=18,u=7,cx=35,s=18 # 1405977

```

```

# Rule 2
<-- [process_name=netscape : 227,2056]
    [prot_words_chars=from8d5to24 : 230,3064]
    [prot_words_chars-1=from8d5to24..from24to25d5 : 235,3930]
    [proc_count_in_win_lf=from2d01267to3d51143..from3d51143to4d64917 :
138,2410]
    [win_title_prot_words=2 : 158,1379]
    : p=12,np=7,u=1,cx=35,c=1.2,s=12 # 1405992

# Rule 3
<-- [process_name=explorer : 52,952]
    [process_name-1=explorer : 44,918]
    [process_name-2=outlook : 171,2925]
    [delta_time_new_window-1=lte10500..from10500to11000 : 512,8478]
    [prot_words_chars-2=from25d5to50 : 21,634]
    [proc_count_in_win_lf=lte0 : 111,1653]
    [win_title_prot_words=2 : 158,1379]
    [win_title_prot_words-1=3 : 83,2195]
    : p=6,np=6,u=6,cx=56,c=1.88,s=6 # 1405972

# Rule 4
<-- [process_name=outlook : 165,2855]
    [process_name-1=outlook : 169,2889]
    [prot_words_chars=from0to7d5 : 80,1186]
    [prot_words_chars-1=from0to7d5..from7d5to8d5 : 220,2867]
    [prot_words_chars-2=from0to7d5 : 83,1196]
    [prot_words_chars-3=from0to7d5 : 83,1195]
    [proc_count_in_win_lf=from0to2d01267 : 259,4319]
    [proc_count_in_win_lf-1=from0to2d01267 : 264,4339]
    [proc_count_in_win_lf-2=from0to2d01267 : 265,4356]
    [proc_count_in_win_lf-3=lte0..from0to2d01267 : 362,5940]
    [win_title_prot_words=1 : 223,3241]
    [win_title_prot_words-1=1 : 224,3244]
    : p=6,np=6,u=6,cx=84,s=6 # 1406151

# Rule 5
<-- [process_name=netscape : 227,2056]
    [process_name-1=explorer : 44,918]
    [delta_time_new_window-2=lte10500 : 512,8473]
    [prot_words_chars=from7d5to8d5 : 139,1673]
    [proc_count_in_win_lf=from0to2d01267 : 259,4319]
    [proc_count_in_win_lf-1=lte0 : 106,1622]
    [proc_count_in_win_lf-2=from0to2d01267 : 265,4356]
    [proc_count_in_win_lf-3=from0to2d01267 : 266,4389]
    [win_title_prot_words-1=2 : 150,1364]
    [win_title_prot_words-3=0 : 39,1102]
    : p=5,np=5,u=5,cx=70,s=5 # 1405970

# Rule 6
<-- [process_name=netscape : 227,2056]
    [process_name-1=netscape : 230,2059]
    [process_name-3=netscape : 228,2053]
    [prot_words_chars-1=from8d5to24 : 223,3046]
    [proc_count_in_win_lf=from2d01267to3d51143 : 113,1993]
    [proc_count_in_win_lf-1=from0to2d01267 : 264,4339]
    [proc_count_in_win_lf-2=from0to2d01267 : 265,4356]
    [win_title_prot_words=1 : 223,3241]
    [win_title_prot_words-1=2 : 150,1364]
    : p=4,np=4,u=4,cx=63,s=4 # 1405948

# Rule 7
<-- [process_name=outlook : 165,2855]

```

```

[process_name-2=outlook : 171,2925]
[process_name-3=outlook : 172,2969]
[prot_words_chars=from0to7d5 : 80,1186]
[prot_words_chars-3=from8d5to24 : 227,3060]
[proc_count_in_win_lf=from0to2d01267 : 259,4319]
[proc_count_in_win_lf-1=from0to2d01267 : 264,4339]
[proc_count_in_win_lf-2=from0to2d01267 : 265,4356]
[proc_count_in_win_lf-3=from0to2d01267 : 266,4389]
[win_title_prot_words=1 : 223,3241]
[win_title_prot_words-1=3 : 83,2195]
[win_title_prot_words-2=1 : 227,3233]
: p=4,np=4,u=4,cx=84,c=2.42,s=4 # 1405955

# Rule 8
<-- [process_name=netscape : 227,2056]
[process_name-2=netscape : 230,2057]
[prot_words_chars-2=from8d5to24 : 220,3052]
[proc_count_in_win_lf=from0to2d01267 : 259,4319]
[proc_count_in_win_lf-1=from0to2d01267 : 264,4339]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 114,2018]
[win_title_prot_words=2 : 158,1379]
: p=4,np=4,u=4,cx=49,s=4 # 1406054

# Rule 9
<-- [process_name=outlook : 165,2855]
[process_name-1=iexplore,winword : 27,2149]
[proc_count_in_win_lf=lte0 : 111,1653]
[proc_count_in_win_lf-2=from0to2d01267 : 265,4356]
[win_opened=lte16 : 515,7944]
[win_title_prot_words=1 : 223,3241]
[win_title_prot_words-1=3 : 83,2195]
[win_title_prot_words-2=0..1 : 267,4321]
: p=4,np=4,u=4,cx=58,c=1.75,s=4 # 1406056

# Rule 10
<-- [process_name=netscape : 227,2056]
[process_name-1=netscape : 230,2059]
[prot_words_chars-1=from8d5to24 : 223,3046]
[proc_count_in_win_lf=from2d01267to3d51143 : 113,1993]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 113,1998]
[win_title_prot_words=1 : 223,3241]
: p=7,np=4,u=5,cx=42,s=7 # 1406113

# Rule 11
<-- [process_name=explorer : 52,952]
[process_name-1=outlook : 169,2889]
[process_name-2=explorer : 41,898]
[delta_time_new_window=lte10500 : 511,8467]
[proc_count_in_win_lf=lte0 : 111,1653]
[proc_count_in_win_lf-2=lte0..from0to2d01267 : 368,5937]
[win_opened=lte16 : 515,7944]
[win_title_prot_words=3 : 82,2193]
[win_title_prot_words-1=4 : 19,552]
: p=3,np=3,u=3,cx=63,s=3 # 1405983

# Rule 12
<-- [process_name=iexplore : 17,1579]
[process_name-1=outlook : 169,2889]
[prot_words_chars-1=from0to7d5 : 81,1194]
[proc_count_in_win_lf=lte0 : 111,1653]
[proc_count_in_win_lf-1=lte0 : 106,1622]
[proc_count_in_win_lf-3=from0to2d01267 : 266,4389]
[win_opened=lte16 : 515,7944]
[win_title_prot_words=4 : 18,540]

```

```

      : p=3,np=3,u=2,cx=56,s=3 # 1405987

# Rule 13
<-- [process_name=netscape : 227,2056]
    [prot_words_chars-2=from7d5to8d5 : 139,1672]
    [prot_words_chars-3=from7d5to8d5 : 139,1668]
    [proc_count_in_win_lf=from2d01267to3d51143 : 113,1993]
    [proc_count_in_win_lf-1=from2d01267to3d51143 : 113,1998]
    [proc_count_in_win_lf-2=from0to2d01267 : 265,4356]
    [proc_count_in_win_lf-3=lte0..from0to2d01267 : 362,5940]
    [win_title_prot_words=1..2 : 381,4620]
    [win_title_prot_words-1=1 : 224,3244]
      : p=3,np=3,u=2,cx=63,s=3 # 1405991

# Rule 14
<-- [process_name=explorer : 52,952]
    [process_name-1=netscape : 230,2059]
    [proc_count_in_win_lf=from0to2d01267 : 259,4319]
    [proc_count_in_win_lf-1=from0to2d01267..from3d51143to4d64917 :
402,6757]
    [win_title_prot_words=2 : 158,1379]
    [win_title_prot_words-1=2 : 150,1364]
      : p=3,np=3,u=1,cx=42,c=1.5,s=3 # 1406005

# Rule 15
<-- [process_name=netscape : 227,2056]
    [prot_words_chars=from24to25d5 : 12,884]
    [win_title_prot_words=3 : 82,2193]
      : p=3,np=3,u=2,cx=21,c=1,s=3 # 1406013

# Rule 16
<-- [process_name=iexplore,wordpad : 28,1611]
    [process_name-1=outlook,wordpad : 180,2921]
    [process_name-2=csrss,msaccess,wordpad : 22,221]
    [delta_time_new_window=lte10500 : 511,8467]
    [prot_words_chars=from0to7d5..from24to25d5 : 461,6807]
    [prot_words_chars-1=from0to7d5 : 81,1194]
    [proc_count_in_win_lf=from0to2d01267 : 259,4319]
    [proc_count_in_win_lf-1=from0to2d01267 : 264,4339]
      : p=3,np=3,u=3,cx=64,c=1.62,s=3 # 1406032

# Rule 17
<-- [process_name=netscape : 227,2056]
    [process_name-2=netscape : 230,2057]
    [prot_words_chars=from7d5to8d5 : 139,1673]
    [prot_words_chars-1=from7d5to8d5 : 139,1673]
    [proc_count_in_win_lf=from0to2d01267 : 259,4319]
    [proc_count_in_win_lf-1=from0to2d01267 : 264,4339]
    [proc_count_in_win_lf-2=from2d01267to3d51143 : 114,2018]
    [proc_count_in_win_lf-3=from2d01267to3d51143 : 120,2016]
    [win_title_prot_words-1=1 : 224,3244]
    [win_title_prot_words-2=1 : 227,3233]
      : p=3,np=3,u=3,cx=70,c=2.2,s=3 # 1406034

# Rule 18
<-- [process_name-2=netscape : 230,2057]
    [prot_words_chars-2=from8d5to24 : 220,3052]
    [proc_count_in_win_lf=from2d01267to3d51143 : 113,1993]
    [proc_count_in_win_lf-1=from3d51143to4d64917 : 25,420]
    [proc_count_in_win_lf-2=from2d01267to3d51143 : 114,2018]
    [win_title_prot_words=1 : 223,3241]
      : p=3,np=3,u=2,cx=42,s=3 # 1406064

# Rule 19

```

```

<-- [process_name=powerpnt : 28,263]
[process_name-2=outlook : 171,2925]
[prot_words_chars=from8d5to24 : 230,3064]
[proc_count_in_win_lf=from0to2d01267 : 259,4319]
[win_title_prot_words-1=2 : 150,1364]
[win_title_prot_words-2=1 : 227,3233]
: p=4,np=3,u=3,cx=42,s=4 # 1406107

# Rule 20
<-- [process_name=netscape : 227,2056]
[process_name-2=acrord32,netscape : 237,2121]
[prot_words_chars=from8d5to24 : 230,3064]
[proc_count_in_win_lf=from0to2d01267 : 259,4319]
[proc_count_in_win_lf-1=lte0..from0to2d01267 : 370,5961]
[win_title_prot_words-3=2 : 152,1412]
: p=5,np=3,u=4,cx=44,c=2,s=5 # 1406203
...

Output_Hypotheses User12
{
# -- This learning took =
# -- System (CPU) time = 36.3
# -- User (Total) time = 36
# -- Number of rules in the cover = 462
# -- Number of conditions = 3485
# -- Complexity for this cover = 24729
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 1073
negative_events = 7968
positive_distinct_events = 935
negative_distinct_events = 7207
[user=user12]
# Rule 1
<-- [process_name=netscape : 440,1843]
[process_name-1=netscape : 439,1850]
[prot_words_chars=from0to7d5..from7d5to8d5 : 540,2538]
[prot_words_chars-2=from7d5to8d5 : 367,1444]
[proc_count_in_win_lf=lte0..from0to2d01267 : 741,5601]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 281,1830]
[proc_count_in_win_lf-2=lte0..from0to2d01267 : 741,5564]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 281,1855]
[win_title_prot_words-1=1 : 549,2919]
[win_title_prot_words-2=1 : 547,2913]
[win_title_prot_words-3=1 : 545,2906]
: p=17,np=17,u=16,cx=77,s=17 # 1672775

# Rule 2
<-- [process_name=netscape : 440,1843]
[prot_words_chars-1=from7d5to8d5 : 367,1445]
[prot_words_chars-3=from7d5to8d5 : 366,1441]
[proc_count_in_win_lf=from2d01267to3d51143 : 281,1825]
[proc_count_in_win_lf-1=lte0..from0to2d01267 : 741,5590]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 281,1851]
[proc_count_in_win_lf-3=lte0..from0to2d01267 : 742,5560]
[win_title_prot_words=1 : 549,2915]
[win_title_prot_words-2=1 : 547,2913]
: p=16,np=16,u=16,cx=63,c=2.33,s=16 # 1672811

# Rule 3
<-- [process_name=netscape : 440,1843]
[process_name-1=netscape : 439,1850]
[proc_count_in_win_lf=from2d01267to3d51143 : 281,1825]

```



```

[proc_count_in_win_lf-1=from2d01267to3d51143 : 281,1830]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 281,1851]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 281,1855]
[win_title_prot_words=1 : 549,2915]
[win_title_prot_words-1=1 : 549,2919]
[win_title_prot_words-2=1 : 547,2913]
[win_title_prot_words-3=1 : 545,2906]
: p=16,np=16,u=16,cx=70,s=16 # 1672833

# Rule 4
<-- [process_name=netscape : 440,1843]
[prot_words_chars-1=from7d5to8d5 : 367,1445]
[prot_words_chars-2=from7d5to8d5 : 367,1444]
[proc_count_in_win_lf=from2d01267to3d51143 : 281,1825]
[proc_count_in_win_lf-1=lte0..from0to2d01267 : 741,5590]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 281,1851]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 281,1855]
[win_title_prot_words=1 : 549,2915]
: p=17,np=16,u=16,cx=56,s=17 # 1672949

# Rule 5
<-- [process_name=outlook : 382,2638]
[process_name-1=iexplore : 65,1531]
[prot_words_chars=from0to7d5 : 173,1093]
[prot_words_chars-2=from24to25d5 : 57,839]
[proc_count_in_win_lf=lte0..from0to2d01267 : 741,5601]
[proc_count_in_win_lf-1=from0to2d01267 : 533,4070]
[proc_count_in_win_lf-2=lte0..from0to2d01267 : 741,5564]
[win_title_prot_words-1=3 : 233,2045]
: p=22,np=16,u=22,cx=56,c=1.88,s=22 # 1672914

# Rule 6
<-- [process_name=netscape : 440,1843]
[process_name-2=netscape : 437,1850]
[prot_words_chars=from7d5to8d5 : 367,1445]
[prot_words_chars-3=from7d5to8d5 : 366,1441]
[proc_count_in_win_lf=from0to2d01267 : 536,4042]
[proc_count_in_win_lf-1=from0to2d01267 : 533,4070]
[proc_count_in_win_lf-2=from0to2d01267 : 537,4084]
[proc_count_in_win_lf-3=from0to2d01267 : 540,4115]
[win_opened=lte16 : 1073,7386]
[win_title_prot_words-1=1 : 549,2919]
[win_title_prot_words-2=1 : 547,2913]
: p=13,np=13,u=13,cx=77,s=13 # 1672749

# Rule 7
<-- [process_name=netscape : 440,1843]
[process_name-3=netscape : 434,1847]
[prot_words_chars-2=from7d5to8d5 : 367,1444]
[proc_count_in_win_lf=from2d01267to3d51143 : 281,1825]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 281,1830]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 281,1851]
[proc_count_in_win_lf-3=from0to2d01267 : 540,4115]
[win_title_prot_words=1 : 549,2915]
: p=13,np=13,u=13,cx=56,s=13 # 1672791

# Rule 8
<-- [process_name=netscape : 440,1843]
[process_name-2=netscape : 437,1850]
[proc_count_in_win_lf=from2d01267to3d51143 : 281,1825]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 281,1830]
[proc_count_in_win_lf-2=lte0..from0to2d01267 : 741,5564]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 281,1855]
[win_title_prot_words=1 : 549,2915]

```

```

[win_title_prot_words-3=1 : 545,2906]
: p=13,np=13,u=13,cx=56,s=13 # 1672795

# Rule 9
<-- [process_name=iexplore : 65,1531]
[process_name-1=outlook : 382,2676]
[prot_words_chars=from24to25d5 : 57,839]
[proc_count_in_win_lf=from0to2d01267 : 536,4042]
[proc_count_in_win_lf-1=lte0 : 208,1520]
[win_title_prot_words-1=1 : 549,2919]
: p=12,np=12,u=2,cx=42,c=1.5,s=12 # 1672765

# Rule 10
<-- [process_name=excel,outlook : 444,2655]
[process_name-1=excel,msohelp,smsmon32 : 72,112]
[win_title_prot_words-1=1 : 549,2919]
: p=17,np=11,u=3,cx=27,s=17 # 1672824

# Rule 11
<-- [process_name=netscape : 440,1843]
[process_name-2=explorer : 47,892]
[prot_words_chars-1=from7d5to8d5 : 367,1445]
[proc_count_in_win_lf=from2d01267to3d51143 : 281,1825]
[proc_count_in_win_lf-1=lte0..from0to2d01267 : 741,5590]
[proc_count_in_win_lf-2=lte0 : 204,1480]
[win_title_prot_words=1 : 549,2915]
[win_title_prot_words-2=2 : 180,1354]
: p=11,np=10,u=11,cx=56,s=11 # 1672821

# Rule 12
<-- [process_name=csrss,msohelp : 26,170]
[process_name-1=excel : 65,101]
[win_title_prot_words-1=2 : 179,1335]
: p=12,np=10,u=4,cx=23,c=1.67,s=12 # 1672770

# Rule 13
<-- [process_name=netscape : 440,1843]
[process_name-1=netscape : 439,1850]
[prot_words_chars=from0to7d5..from7d5to8d5 : 540,2538]
[prot_words_chars-1=from7d5to8d5 : 367,1445]
[proc_count_in_win_lf=from0to2d01267 : 536,4042]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 281,1830]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 281,1851]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 281,1855]
[win_title_prot_words-3=1 : 545,2906]
: p=12,np=10,u=10,cx=63,s=12 # 1673041

# Rule 14
<-- [process_name=netscape : 440,1843]
[process_name-1=netscape : 439,1850]
[prot_words_chars=from7d5to8d5 : 367,1445]
[prot_words_chars-1=from7d5to8d5 : 367,1445]
[prot_words_chars-3=from7d5to8d5 : 366,1441]
[proc_count_in_win_lf=from0to2d01267 : 536,4042]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 281,1830]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 281,1851]
[proc_count_in_win_lf-3=from0to2d01267 : 540,4115]
: p=9,np=9,u=9,cx=63,c=2.22,s=9 # 1672790

# Rule 15
<-- [process_name=netscape : 440,1843]
[process_name-1=explorer : 52,910]
[prot_words_chars=from7d5to8d5 : 367,1445]
[proc_count_in_win_lf=from0to2d01267 : 536,4042]
[proc_count_in_win_lf-1=lte0 : 208,1520]

```

```

[proc_count_in_win_lf-3=from0to2d01267 : 540,4115]
[win_title_prot_words-1=2 : 179,1335]
[win_title_prot_words-3=0 : 101,1040]
: p=8,np=8,u=8,cx=56,s=8 # 1672753

# Rule 16
<-- [process_name=outlook : 382,2638]
[process_name-1=outlook : 382,2676]
[prot_words_chars-2=lte0..from0to7d5 : 272,2135]
[proc_count_in_win_lf=from0to2d01267 : 536,4042]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 281,1830]
[proc_count_in_win_lf-2=from0to2d01267 : 537,4084]
[proc_count_in_win_lf-3=from2d01267to3d51143..from3d51143to4d64917 :
322,2262]
[win_opened=lte16 : 1073,7386]
[win_title_prot_words=1 : 549,2915]
[win_title_prot_words-1=3 : 233,2045]
[win_title_prot_words-3=3 : 229,2029]
: p=7,np=7,u=5,cx=77,s=7 # 1672763

# Rule 17
<-- [process_name=outlook : 382,2638]
[process_name-2=smsmon32,wscript : 22,117]
[proc_count_in_win_lf=lte0 : 205,1559]
[win_title_prot_words=2 : 175,1362]
: p=7,np=7,u=7,cx=30,s=7 # 1672766

# Rule 18
<-- [process_name=netscape : 440,1843]
[process_name-3=explorer : 43,871]
[proc_count_in_win_lf=from2d01267to3d51143 : 281,1825]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 281,1830]
[proc_count_in_win_lf-2=from0to2d01267 : 537,4084]
[proc_count_in_win_lf-3=lte0 : 202,1445]
[win_title_prot_words=1 : 549,2915]
[win_title_prot_words-1=1 : 549,2919]
[win_title_prot_words-3=2 : 185,1379]
: p=7,np=7,u=7,cx=63,s=7 # 1672771

# Rule 19
<-- [process_name=outlook : 382,2638]
[process_name-1=outlook : 382,2676]
[prot_words_chars-1=from8d5to24 : 362,2907]
[proc_count_in_win_lf=from2d01267to3d51143 : 281,1825]
[proc_count_in_win_lf-1=from0to2d01267..from2d01267to3d51143 :
814,5900]
[proc_count_in_win_lf-2=from0to2d01267 : 537,4084]
[proc_count_in_win_lf-3=from2d01267to3d51143..from3d51143to4d64917 :
322,2262]
[win_title_prot_words=1 : 549,2915]
[win_title_prot_words-1=3 : 233,2045]
[win_title_prot_words-2=1 : 547,2913]
[win_title_prot_words-3=3 : 229,2029]
: p=7,np=7,u=5,cx=77,s=7 # 1672781

# Rule 20
<-- [process_name=outlook : 382,2638]
[delta_time_new_window-2=lte10500 : 1061,7924]
[prot_words_chars-1=from0to7d5 : 173,1102]
[prot_words_chars-2=from8d5to24 : 362,2910]
[proc_count_in_win_lf=from0to2d01267 : 536,4042]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 281,1830]
[proc_count_in_win_lf-2=from0to2d01267 : 537,4084]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 281,1855]

```

```

[win_title_prot_words=3 : 240,2035]
: p=7,np=7,u=7,cx=63,c=2.22,s=7 # 1672842

...

Output_Hypotheses User19
{
# -- This learning took =
# -- System (CPU) time = 27.9
# -- User (Total) time = 28
# -- Number of rules in the cover = 339
# -- Number of conditions = 2148
# -- Complexity for this cover = 15568
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 669
negative_events = 8372
positive_distinct_events = 631
negative_distinct_events = 7511
[user=user19]
# Rule 1
<-- [process_name=winzip32 : 30,88]
[process_name-1=explorer,msoffice,outlook,winword,winzip32 :
498,4074]
[win_title_prot_words=2..3 : 300,3512]
: p=19,np=10,u=3,cx=29,c=1.33,s=19 # 1861428

# Rule 2
<-- [process_name=outlook : 221,2799]
[proc_count_in_win_lf=from2d01267to3d51143..from4d64917to5d56641 :
190,2457]
[proc_count_in_win_lf-1=from3d51143to4d64917..from5d56641to7d12078 :
65,530]
[proc_count_in_win_lf-2=from3d51143to4d64917..from5d56641to7d12078 :
66,534]
[win_opened=lte16 : 669,7790]
[win_title_prot_words=3 : 107,2168]
[win_title_prot_words-1=1..2 : 342,4640]
[win_title_prot_words-2=1..3 : 458,6808]
: p=9,np=7,u=8,cx=56,c=1.75,s=9 # 1861575

# Rule 3
<-- [process_name=outlook : 221,2799]
[prot_words_chars-1=lte0 : 188,930]
[proc_count_in_win_lf=from2d01267to3d51143..from3d51143to4d64917 :
179,2369]
[win_title_prot_words=2 : 193,1344]
: p=6,np=6,u=6,cx=28,s=6 # 1861484

# Rule 4
<-- [process_name=winword,winzip32 : 141,557]
[process_name-1=winword,winzip32 : 141,557]
[prot_words_chars=from8d5to24 : 317,2977]
[proc_count_in_win_lf=from0to2d01267 : 306,4272]
[proc_count_in_win_lf-1=from0to2d01267 : 307,4296]
[proc_count_in_win_lf-2=from0to2d01267 : 309,4312]
[win_title_prot_words-1=1 : 150,3318]
[win_title_prot_words-2=1 : 149,3311]
: p=7,np=6,u=5,cx=60,c=1.88,s=7 # 1861425

# Rule 5
<-- [process_name=artgalry,explorer : 131,883]
[process_name-1=artgalry,explorer,netscape : 186,2988]

```

```

[process_name-3=artgalry,csrss,msoffice,netscape : 103,2320]
[prot_words_chars-1=lte0 : 188,930]
[prot_words_chars-2=lte0..from0to7d5 : 276,2131]
[win_opened=lte16 : 669,7790]
[win_title_prot_words=0 : 188,912]
: p=7,np=6,u=2,cx=61,c=2,s=7 # 1861459

# Rule 6
<-- [process_name=artgalry,explorer : 131,883]
[process_name-1=artgalry,explorer : 131,841]
[prot_words_chars-1=lte0 : 188,930]
[prot_words_chars-2=lte0 : 188,940]
[prot_words_chars-3=lte0..from0to7d5 : 276,2143]
[proc_count_in_win_lf=from0to2d01267..from4d64917to5d56641 :
496,6729]
[proc_count_in_win_lf-1=from0to2d01267..from4d64917to5d56641 :
502,6759]
[win_opened=lte16 : 669,7790]
[win_title_prot_words=0 : 188,912]
: p=53,np=6,u=17,cx=67,c=1.89,s=53 # 1861720

# Rule 7
<-- [process_name=acrord32,winword : 146,592]
[process_name-2=acrord32,explorer,outlook,winword : 491,4107]
[prot_words_chars=from0to7d5 : 87,1179]
[proc_count_in_win_lf=lte0 : 168,1596]
: p=5,np=5,u=2,cx=36,c=1.5,s=5 # 1861445

# Rule 8
<-- [process_name=winword : 115,552]
[process_name-1=winword : 115,552]
[process_name-3=explorer,outlook : 354,3609]
[prot_words_chars=from8d5to24 : 317,2977]
[prot_words_chars-1=from0to7d5 : 88,1187]
[proc_count_in_win_lf=from0to2d01267 : 306,4272]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 139,1993]
[win_title_prot_words-1=1 : 150,3318]
: p=4,np=4,u=3,cx=58,c=2,s=4 # 1861434

# Rule 9
<-- [process_name=acrord32,winword : 146,592]
[process_name-1=acrord32,winword : 146,592]
[proc_count_in_win_lf=from0to2d01267 : 306,4272]
[proc_count_in_win_lf-1=lte0 : 162,1566]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 140,1996]
[win_opened=lte16 : 669,7790]
[win_title_prot_words=2 : 193,1344]
[win_title_prot_words-1=2 : 192,1322]
: p=4,np=4,u=2,cx=60,s=4 # 1861439

# Rule 10
<-- [process_name=msoffice : 27,99]
[process_name-1=outlook,winzip32 : 254,2835]
[proc_count_in_win_lf-1=lte0 : 162,1566]
[win_title_prot_words=4 : 24,534]
: p=4,np=4,u=3,cx=30,s=4 # 1861525

# Rule 11
<-- [process_name-1=artgalry,csrss,wscript : 39,197]
[proc_count_in_win_lf=from0to2d01267 : 306,4272]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 139,1993]
[win_title_prot_words=0 : 188,912]
: p=4,np=4,u=1,cx=32,s=4 # 1861553

```

```

# Rule 12
<-- [process_name=acrord32,outlook : 252,2839]
    [process_name-1=acrord32,explorer,winword : 263,1350]
    [process_name-2=csrss,outlook,winzip32 : 287,2949]
    [prot_words_chars=from8d5to24 : 317,2977]
    [proc_count_in_win_lf=from3d51143to4d64917 : 45,397]
    [win_title_prot_words-1=2 : 192,1322]
    : p=6,np=4,u=2,cx=52,c=1.67,s=6 # 1861466

# Rule 13
<-- [process_name=powerpnt,winzip32 : 79,243]
    [process_name-1=netscape,outlook,winzip32 : 309,4982]
    [prot_words_chars-1=from7d5to8d5..from8d5to24 : 363,4718]
    [proc_count_in_win_lf=lte0 : 168,1596]
    [proc_count_in_win_lf-1=from0to2d01267..from2d01267to3d51143 :
442,6272]
    [win_title_prot_words=2 : 193,1344]
    : p=8,np=4,u=3,cx=48,s=8 # 1861458

# Rule 14
<-- [process_name=artgalry,explorer : 131,883]
    [process_name-1=acrord32,powerpnt,winword,winzip32 : 220,753]
    [prot_words_chars=lte0 : 188,912]
    [win_title_prot_words-1=2 : 192,1322]
    [win_title_prot_words-2=0..1 : 337,4251]
    : p=10,np=4,u=4,cx=43,c=1.8,s=10 # 1861593

# Rule 15
<-- [process_name=acrord32 : 35,123]
    [process_name-1=outlook : 228,2830]
    [proc_count_in_win_lf=from2d01267to3d51143 : 134,1972]
    [win_title_prot_words=2 : 193,1344]
    : p=3,np=3,u=2,cx=28,c=1.25,s=3 # 1861412

# Rule 16
<-- [process_name=netscape : 60,2223]
    [process_name-1=acrord32,powerpnt : 83,279]
    [win_opened=lte16 : 669,7790]
    [win_title_prot_words=2 : 193,1344]
    [win_title_prot_words-1=2 : 192,1322]
    : p=3,np=3,u=1,cx=37,s=3 # 1861420

# Rule 17
<-- [process_name=acrord32,outlook,powerpnt : 301,2994]
    [process_name-1=outlook,winword : 339,3299]
    [prot_words_chars=from8d5to24 : 317,2977]
    [prot_words_chars-2=from0to7d5..from8d5to24 : 453,5909]
    [prot_words_chars-3=from0to7d5 : 88,1190]
    [proc_count_in_win_lf=lte0..from0to2d01267 : 474,5868]
    [proc_count_in_win_lf-1=from3d51143to4d64917 : 47,398]
    [proc_count_in_win_lf-2=lte0..from0to2d01267 : 464,5841]
    [proc_count_in_win_lf-3=from0to2d01267 : 308,4347]
    [win_title_prot_words-1=1..2 : 342,4640]
    : p=3,np=3,u=3,cx=76,s=3 # 1861443

# Rule 18
<-- [process_name=acrord32,outlook : 252,2839]
    [process_name-1=acrord32,cmd,csrss : 59,221]
    [prot_words_chars=from8d5to24 : 317,2977]
    [proc_count_in_win_lf=from3d51143to4d64917 : 45,397]
    [proc_count_in_win_lf-1=lte0 : 162,1566]
    : p=3,np=3,u=2,cx=41,s=3 # 1861452

# Rule 19

```

```

<-- [process_name=csrss : 27,167]
[process_name-1=outlook : 228,2830]
[proc_count_in_win_lf=lte0 : 168,1596]
[proc_count_in_win_lf-1=from2d01267to3d51143..from3d51143to4d64917 :
182,2374]
[win_title_prot_words-1=1 : 150,3318]
: p=3,np=3,u=2,cx=35,c=1.6,s=3 # 1861462

# Rule 20
<-- [process_name=netscape : 60,2223]
[prot_words_chars=from7d5to8d5 : 40,1772]
[prot_words_chars-1=from7d5to8d5 : 39,1773]
[prot_words_chars-3=from7d5to8d5 : 37,1770]
[proc_count_in_win_lf=from0to2d01267 : 306,4272]
[proc_count_in_win_lf-1=from2d01267to3d51143 : 135,1976]
[proc_count_in_win_lf-2=from2d01267to3d51143 : 139,1993]
[proc_count_in_win_lf-3=from0to2d01267 : 308,4347]
[win_opened=lte16 : 669,7790]
: p=3,np=3,u=3,cx=63,s=3 # 1861464

...

Output_Hypotheses User25
{
# -- This learning took =
# -- System (CPU) time = 53.3
# -- User (Total) time = 53
# -- Number of rules in the cover = 423
# -- Number of conditions = 2805
# -- Complexity for this cover = 20039
# -- Average number of rules kept from each stars = 1
# -- Uncovered Positives = 0

positive_events = 1992
negative_events = 7049
positive_distinct_events = 1743
negative_distinct_events = 6399
[user=user25]
# Rule 1
<-- [proc_count_in_win_lf=from0to2d01267..from4d64917to5d56641 :
1599,5626]
[win_opened=from16to28 : 304,131]
[win_title_prot_words=3..5 : 1504,1432]
: p=211,np=124,u=2,cx=21,c=1,s=211 # 2135287

# Rule 2
<-- [process_name=iexplore : 1282,314]
[prot_words_chars=lte0..from24to25d5 : 1525,6843]
[prot_words_chars-1=from24to25d5 : 763,133]
[proc_count_in_win_lf-2=from2d01267to3d51143..from3d51143to4d64917 :
554,2027]
[win_title_prot_words-2=3 : 1038,1234]
: p=77,np=34,u=2,cx=35,c=2,s=77 # 2135331

# Rule 3
<-- [process_name=iexplore : 1282,314]
[prot_words_chars=from25d5to50 : 467,206]
[proc_count_in_win_lf=from0to2d01267..from2d01267to3d51143 :
1485,5199]
[proc_count_in_win_lf-1=from0to2d01267 : 1030,3573]
[win_opened=lte16..from16to28 : 1990,6904]
[win_title_prot_words-1=3 : 1034,1244]
[win_title_prot_words-2=3 : 1038,1234]

```

```

      : p=44,np=32,u=5,cx=49,s=44 # 2135288

# Rule 4
<-- [process_name=iexplore : 1282,314]
    [prot_words_chars=from24to25d5 : 763,133]
    [prot_words_chars-3=from25d5to50 : 454,181]
    [proc_count_in_win_lf=lte0..from0to2d01267 : 1411,4931]
    [proc_count_in_win_lf-2=lte0..from0to2d01267 : 1399,4906]
    [win_opened=lte16..from16to28 : 1990,6904]
    [win_title_prot_words-1=3..5 : 1497,1440]
    [win_title_prot_words-2=3 : 1038,1234]
      : p=59,np=29,u=42,cx=56,s=59 # 2135402

# Rule 5
<-- [process_name=iexplore : 1282,314]
    [prot_words_chars-1=from25d5to50 : 466,205]
    [proc_count_in_win_lf=from3d51143to4d64917..from5d56641to7d12078 :
124,465]
    [win_title_prot_words=3 : 1040,1235]
      : p=32,np=28,u=7,cx=28,s=32 # 2135284

# Rule 6
<-- [process_name=iexplore : 1282,314]
    [prot_words_chars-3=from25d5to50 : 454,181]
    [proc_count_in_win_lf=from2d01267to3d51143 : 457,1649]
    [proc_count_in_win_lf-2=from2d01267to3d51143..from4d64917to5d56641 :
583,2101]
    [win_opened=lte16..from16to28 : 1990,6904]
    [win_title_prot_words=3 : 1040,1235]
    [win_title_prot_words-1=3..5 : 1497,1440]
      : p=67,np=28,u=28,cx=49,s=67 # 2135382

# Rule 7
<-- [process_name=outlook,winword : 639,2961]
    [process_name-1=iexplore : 1282,314]
    [prot_words_chars=from8d5to24 : 439,2855]
    [proc_count_in_win_lf=from0to2d01267 : 1028,3550]
    [win_title_prot_words-1=3 : 1034,1244]
      : p=24,np=23,u=4,cx=37,s=24 # 2135292

# Rule 8
<-- [process_name=iexplore : 1282,314]
    [proc_count_in_win_lf=lte0 : 383,1381]
    [proc_count_in_win_lf-1=from2d01267to3d51143..from3d51143to4d64917 :
547,2009]
    [win_opened=lte16..from16to28 : 1990,6904]
    [win_title_prot_words=3 : 1040,1235]
      : p=20,np=17,u=5,cx=35,s=20 # 2135350

# Rule 9
<-- [process_name=iexplore : 1282,314]
    [proc_count_in_win_lf=lte0..from0to2d01267 : 1411,4931]
    [proc_count_in_win_lf-1=from3d51143to4d64917..from5d56641to7d12078 :
127,468]
    [win_title_prot_words=3..4 : 1439,1394]
      : p=40,np=13,u=4,cx=28,c=1.25,s=40 # 2135317

# Rule 10
<-- [process_name=iexplore : 1282,314]
    [prot_words_chars=from25d5to50 : 467,206]
    [prot_words_chars-1=from8d5to24..from24to25d5 : 1197,2968]
    [proc_count_in_win_lf-1=lte0..from0to2d01267 : 1406,4925]
    [proc_count_in_win_lf-2=lte0 : 368,1316]
    [win_opened=lte16..from16to28 : 1990,6904]

```



```

[win_title_prot_words=4 : 399,159]
  : p=40,np=12,u=3,cx=49,c=1.57,s=40 # 2135338

# Rule 11
<-- [process_name=fpexpress,iexplore : 1285,314]
[prot_words_chars=from8d5to24 : 439,2855]
[win_title_prot_words=1 : 209,3255]
  : p=11,np=11,u=2,cx=23,s=11 # 2135281

# Rule 12
<-- [process_name=outlook : 537,2483]
[process_name-1=iexplore,winword : 1385,791]
[process_name-3=iexplore,outlook : 1795,2843]
[proc_count_in_win_lf-1=from3d51143to4d64917 : 88,357]
[win_title_prot_words=3 : 1040,1235]
  : p=14,np=9,u=5,cx=39,s=14 # 2135324

# Rule 13
<-- [process_name=iexplore : 1282,314]
[process_name-3=iexplore,winword : 1371,792]
[prot_words_chars=from25d5to50 : 467,206]
[prot_words_chars-1=from8d5to24..from24to25d5 : 1197,2968]
[prot_words_chars-2=from24to25d5..from25d5to50 : 1223,328]
[proc_count_in_win_lf=from0to2d01267 : 1028,3550]
[proc_count_in_win_lf-1=lte0..from0to2d01267 : 1406,4925]
[proc_count_in_win_lf-3=lte0..from0to2d01267 : 1399,4903]
[win_opened=lte16..from16to28 : 1990,6904]
  : p=79,np=9,u=22,cx=65,s=79 # 2135537

# Rule 14
<-- [process_name-1=explorer,fpexpress,iexplore,winword : 1439,1615]
[process_name-2=fpexpress,iexplore,winword : 1382,791]
[prot_words_chars=from24to25d5 : 763,133]
[proc_count_in_win_lf=lte0..from0to2d01267 : 1411,4931]
[win_title_prot_words-3=3 : 1035,1223]
  : p=186,np=9,u=40,cx=45,s=186 # 2135467

# Rule 15
<-- [process_name=iexplore : 1282,314]
[proc_count_in_win_lf=lte0 : 383,1381]
[proc_count_in_win_lf-1=lte0..from0to2d01267 : 1406,4925]
[win_title_prot_words=5 : 65,38]
[win_title_prot_words-1=3 : 1034,1244]
  : p=8,np=8,u=1,cx=35,c=1.4,s=8 # 2135298

# Rule 16
<-- [process_name=outlook : 537,2483]
[process_name-2=explorer,iexplore : 1327,1116]
[proc_count_in_win_lf=from0to2d01267 : 1028,3550]
[proc_count_in_win_lf-1=from2d01267to3d51143..from4d64917to5d56641 :
576,2082]
[proc_count_in_win_lf-3=from2d01267to3d51143 : 466,1670]
[win_title_prot_words=1 : 209,3255]
[win_title_prot_words-1=3 : 1034,1244]
[win_title_prot_words-2=2..5 : 1637,2812]
  : p=8,np=8,u=6,cx=58,c=2.12,s=8 # 2135299

# Rule 17
<-- [process_name=iexplore : 1282,314]
[prot_words_chars=from25d5to50 : 467,206]
[proc_count_in_win_lf=from0to2d01267 : 1028,3550]
[proc_count_in_win_lf-2=from2d01267to3d51143..from3d51143to4d64917 :
554,2027]
[win_title_prot_words-1=0..2 : 493,5607]

```

```

      : p=8,np=8,u=1,cx=35,c=1.6,s=8 # 2135330

# Rule 18
<-- [process_name=outlook : 537,2483]
    [process_name-1=iexplore : 1282,314]
    [win_title_prot_words=3 : 1040,1235]
    [win_title_prot_words-1=4..5 : 463,196]
    [win_title_prot_words-2=3 : 1038,1234]
      : p=11,np=8,u=6,cx=35,s=11 # 2135326

# Rule 19
<-- [process_name=iexplore : 1282,314]
    [proc_count_in_win_lf=from0to2d01267..from2d01267to3d51143 :
1485,5199]
    [proc_count_in_win_lf-1=lte0..from2d01267to3d51143 : 1865,6577]
    [proc_count_in_win_lf-2=from0to2d01267 : 1031,3590]
    [win_title_prot_words=5 : 65,38]
      : p=32,np=8,u=6,cx=35,s=32 # 2135302

# Rule 20
<-- [process_name=iexplore,outlook,winword : 1889,3220]
    [prot_words_chars=from7d5to8d5..from8d5to24 : 442,4664]
    [proc_count_in_win_lf=lte0 : 383,1381]
    [win_title_prot_words=1 : 209,3255]
    [win_title_prot_words-1=2 : 141,1373]
      : p=8,np=7,u=7,cx=39,s=8 # 2135314

```

APPENDIX D: HEATMAPS FOR SELECTED EXPERIMENTS

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = strict Evaluation of Disjunction = max



Figure 187: Heat map for testing session 281 (User 1) with degrees of match to 10 Users. Black color indicates no match, white color indicates match.

Source Data: window records **Training Dataset:** Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = strict Evaluation of Disjunction = max



Figure 188: Heat map for testing session 282 (User 1) with degrees of match to 10 Users. Black color indicates no match, white color indicates match.

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = strict Evaluation of Disjunction = max

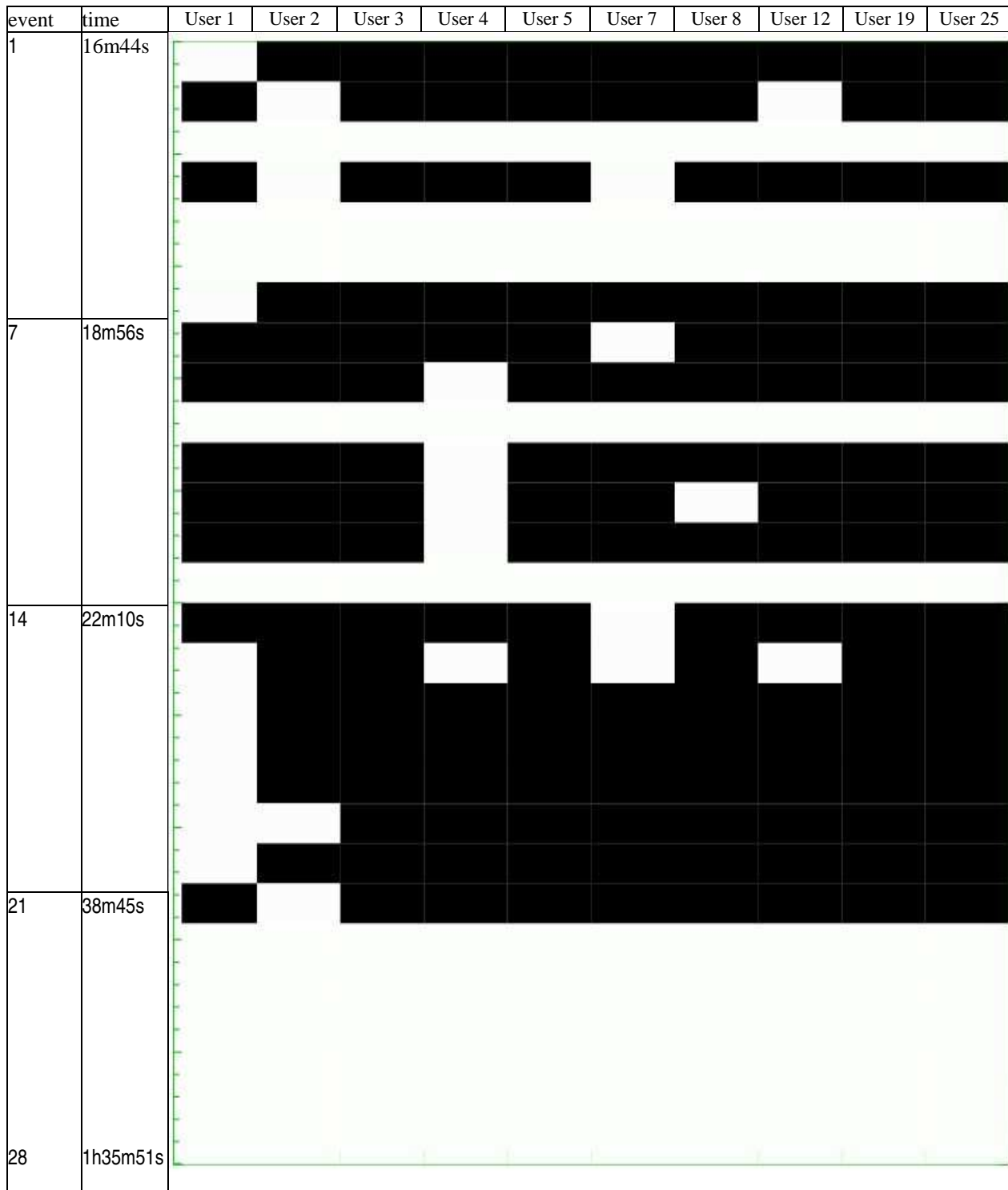


Figure 189: Heat map for testing session 283 (User 1) with degrees of match to 10 Users. Black color indicates no match, white color indicates match.

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = strict Evaluation of Disjunction = max

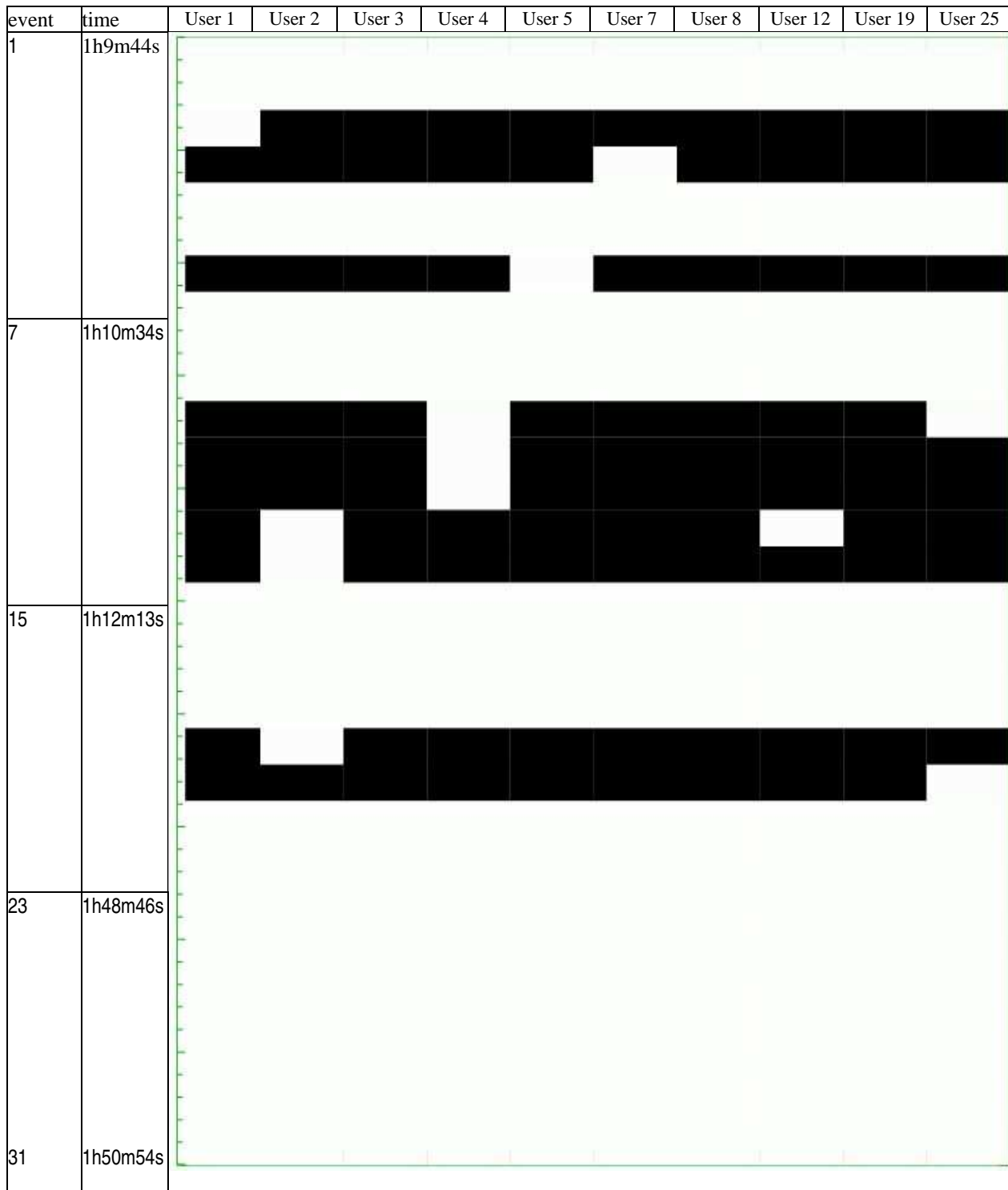


Figure190: Heat map for testing session 284 (User 1) with degrees of match to 10 Users. Black color indicates no match, white color indicates match.

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = strict Evaluation of Disjunction = max



Figure191: Heat map for testing session 285 (User 1) with degrees of match to 10 Users. Black color indicates no match, white color indicates match.

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = strict Evaluation of Disjunction = max



Figure192: Heat map for testing session 1195 (User 25) with degrees of match to 10 Users.
Black color indicates no match, white color indicates match.

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = strict Evaluation of Disjunction = max



Figure193: Heat map for testing session 1196 (User 25) with degrees of match to 10 Users.
Black color indicates no match, white color indicates match.

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = strict Evaluation of Disjunction = max

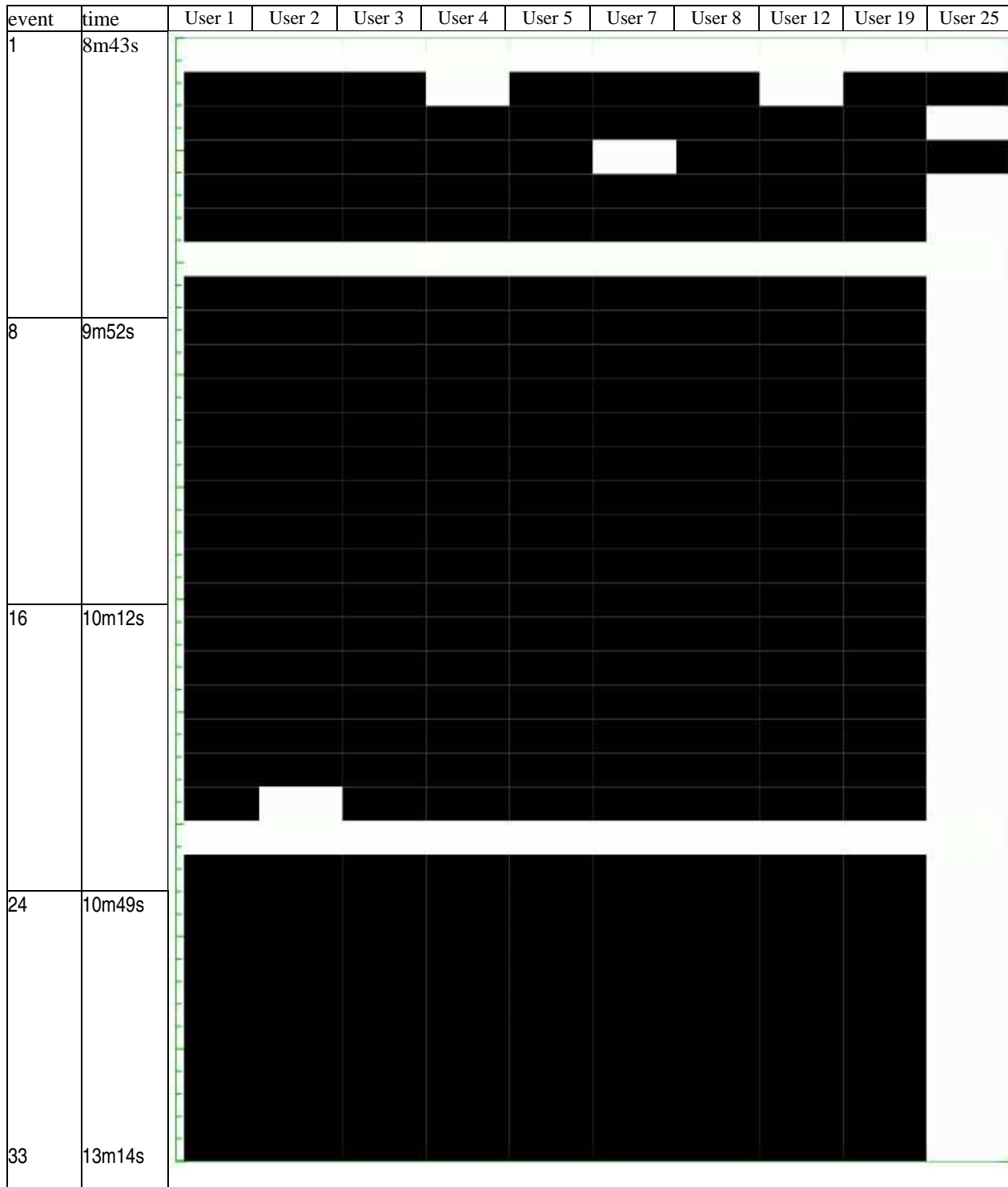


Figure194: Heat map for testing session 1197 (User 25) with degrees of match to 10 Users.
Black color indicates no match, white color indicates match.

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = strict Evaluation of Disjunction = max



Figure195: Heat map for testing session 1198 (User 25) with degrees of match to 10 Users.
Black color indicates no match, white color indicates match.

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = strict Evaluation of Disjunction = max



Figure196: Heat map for testing session 1199 (User 25) with degrees of match to 10 Users.
Black color indicates no match, white color indicates match.

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = cov. ratio Evaluation of Disjunction = max

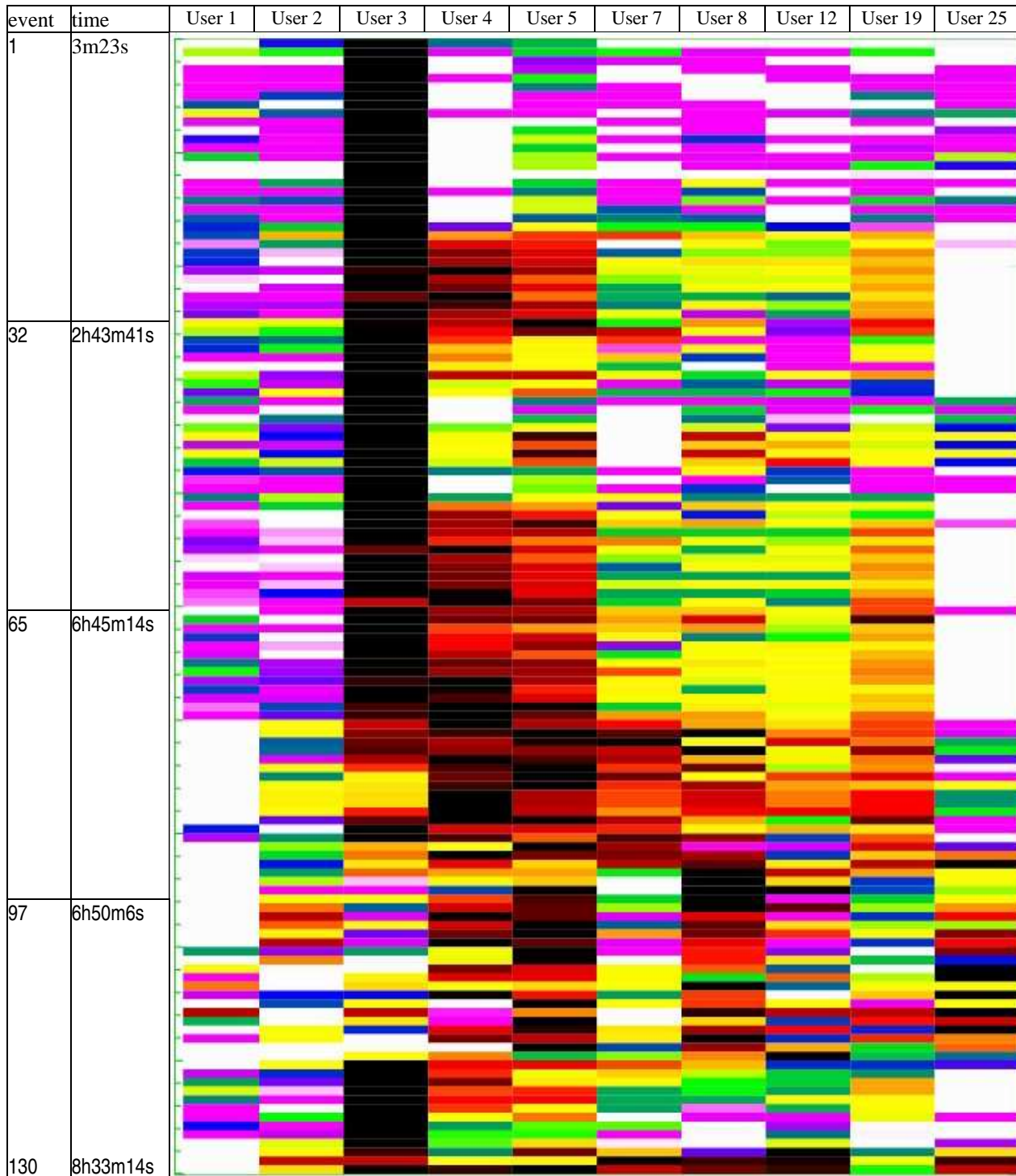


Figure 197: Heat map for testing session 281 (User 1) with degrees of match to 10 Users. Black color indicates no match, white color indicates match.

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = cov. ratio Evaluation of Disjunction = max

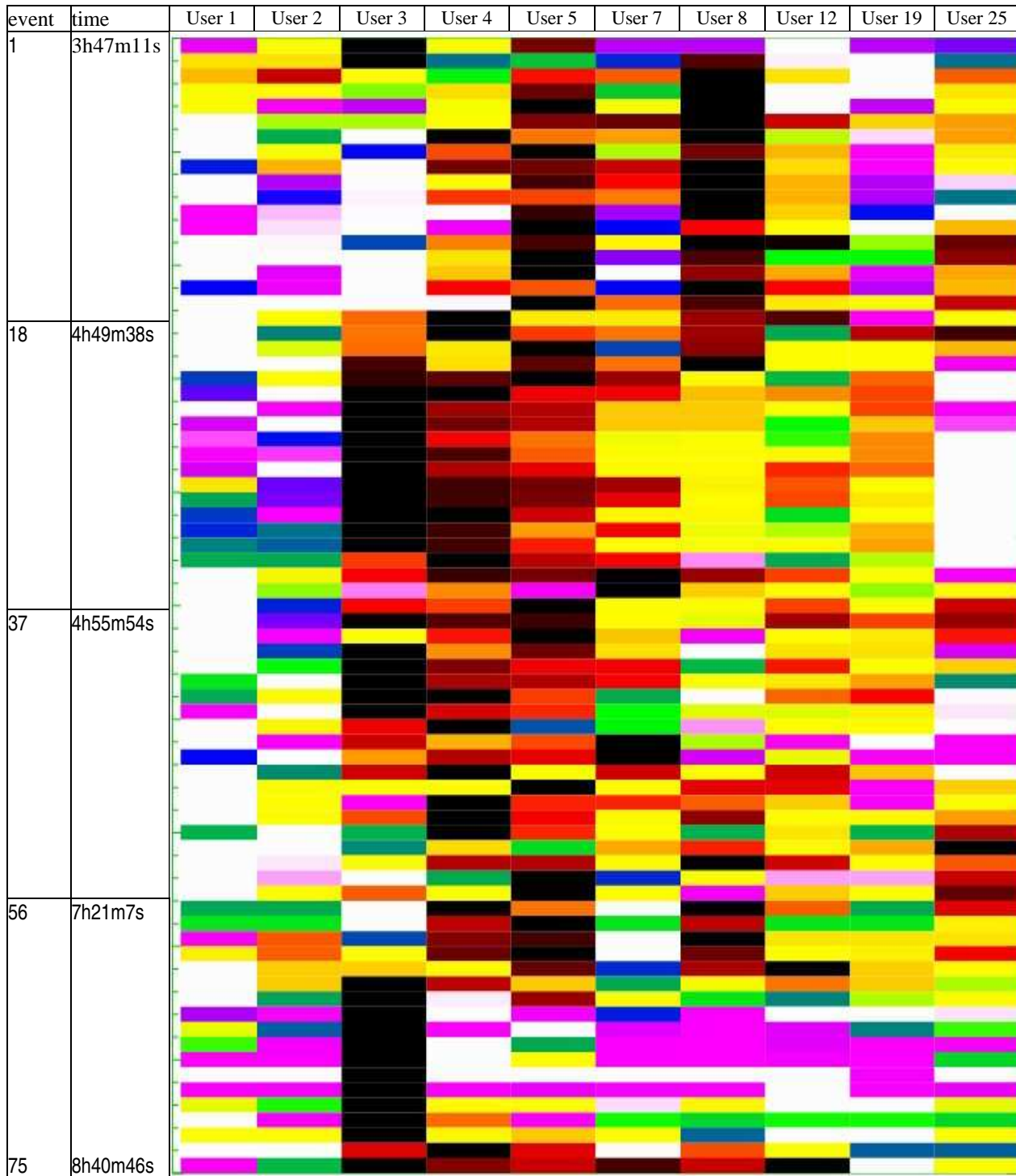


Figure 198: Heat map for testing session 282 (User 1) with degrees of match to 10 Users. Black color indicates no match, white color indicates match.

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = cov. ratio Evaluation of Disjunction = max

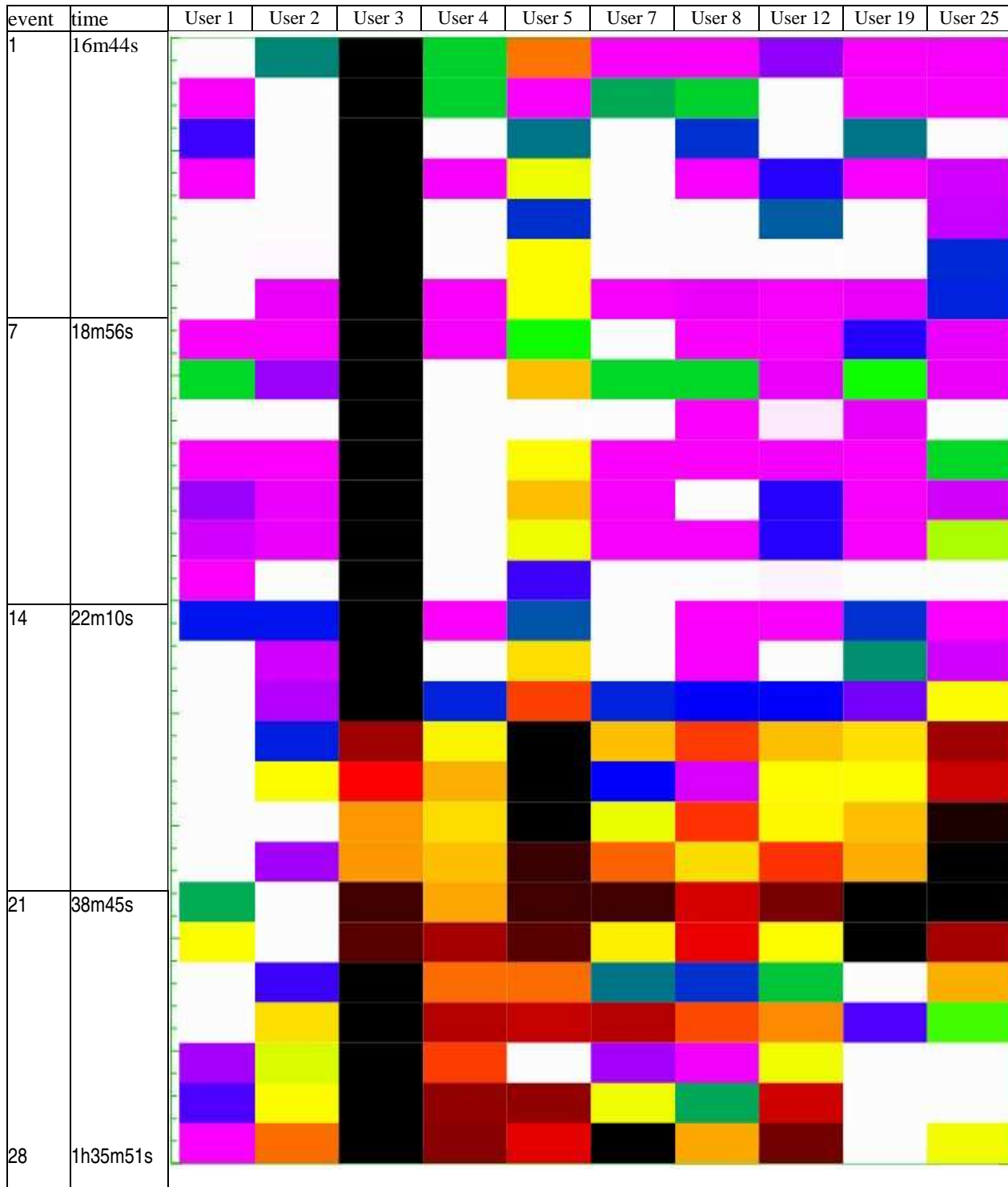


Figure 199: Heat map for testing session 283 (User 1) with degrees of match to 10 Users. Black color indicates no match, white color indicates match.

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = cov. ratio Evaluation of Disjunction = max

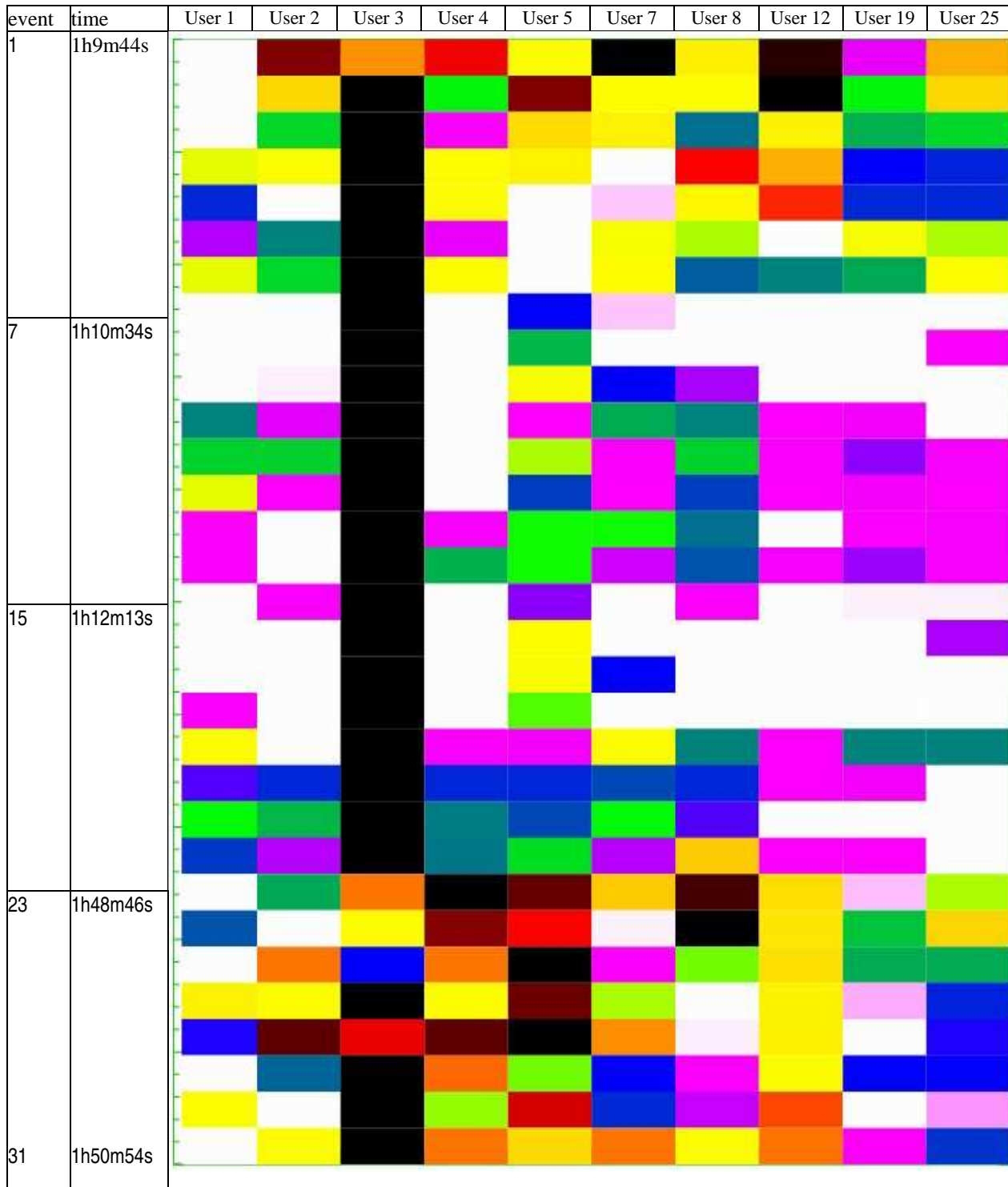


Figure 200: Heat map for testing session 284 (User 1) with degrees of match to 10 Users. Black color indicates no match, white color indicates match.

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = cov. ratio Evaluation of Disjunction = max

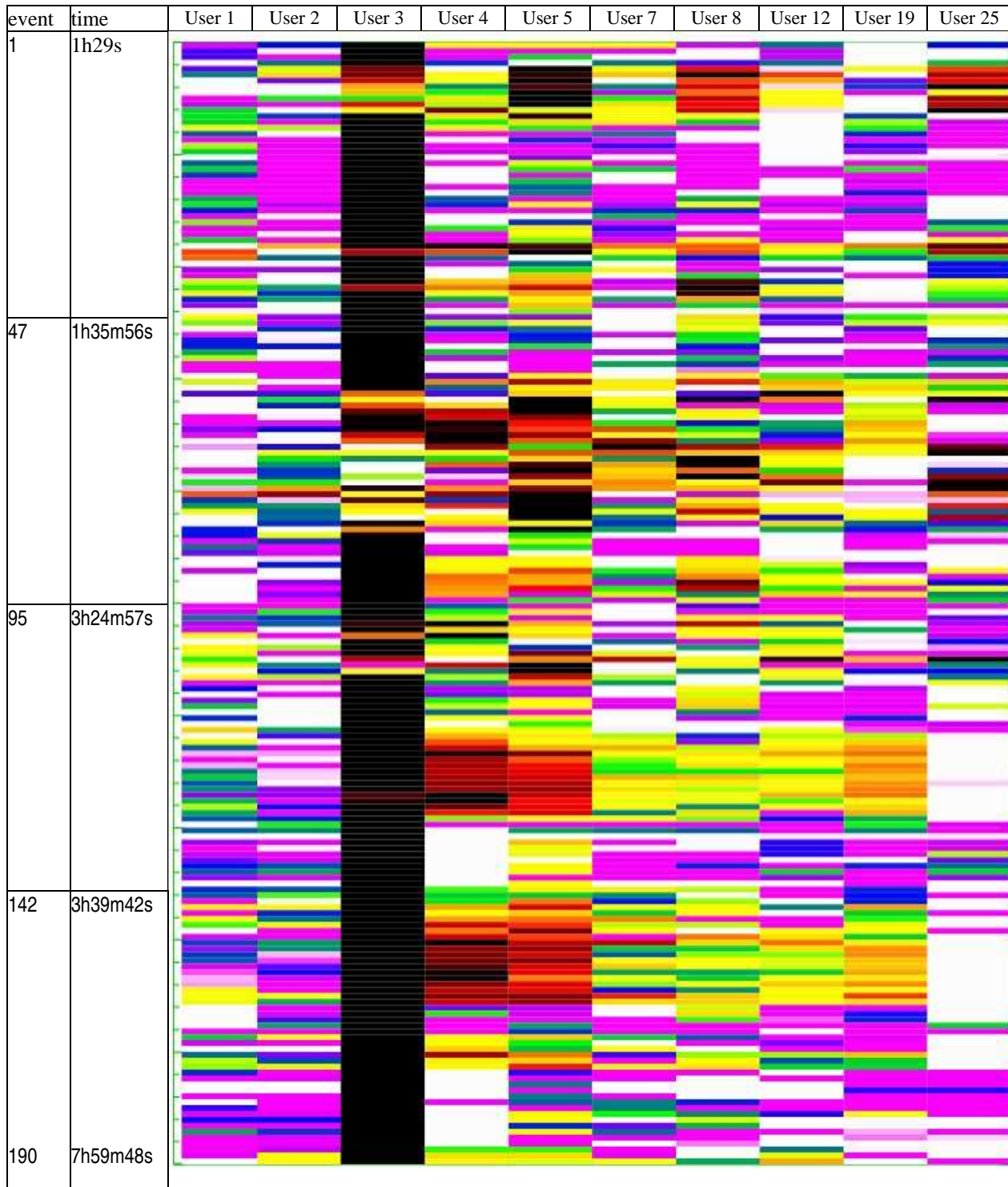


Figure 201: Heat map for testing session 285 (User 1) with degrees of match to 10 Users. Black color indicates no match, white color indicates match.

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = cov. ratio Evaluation of Disjunction = max

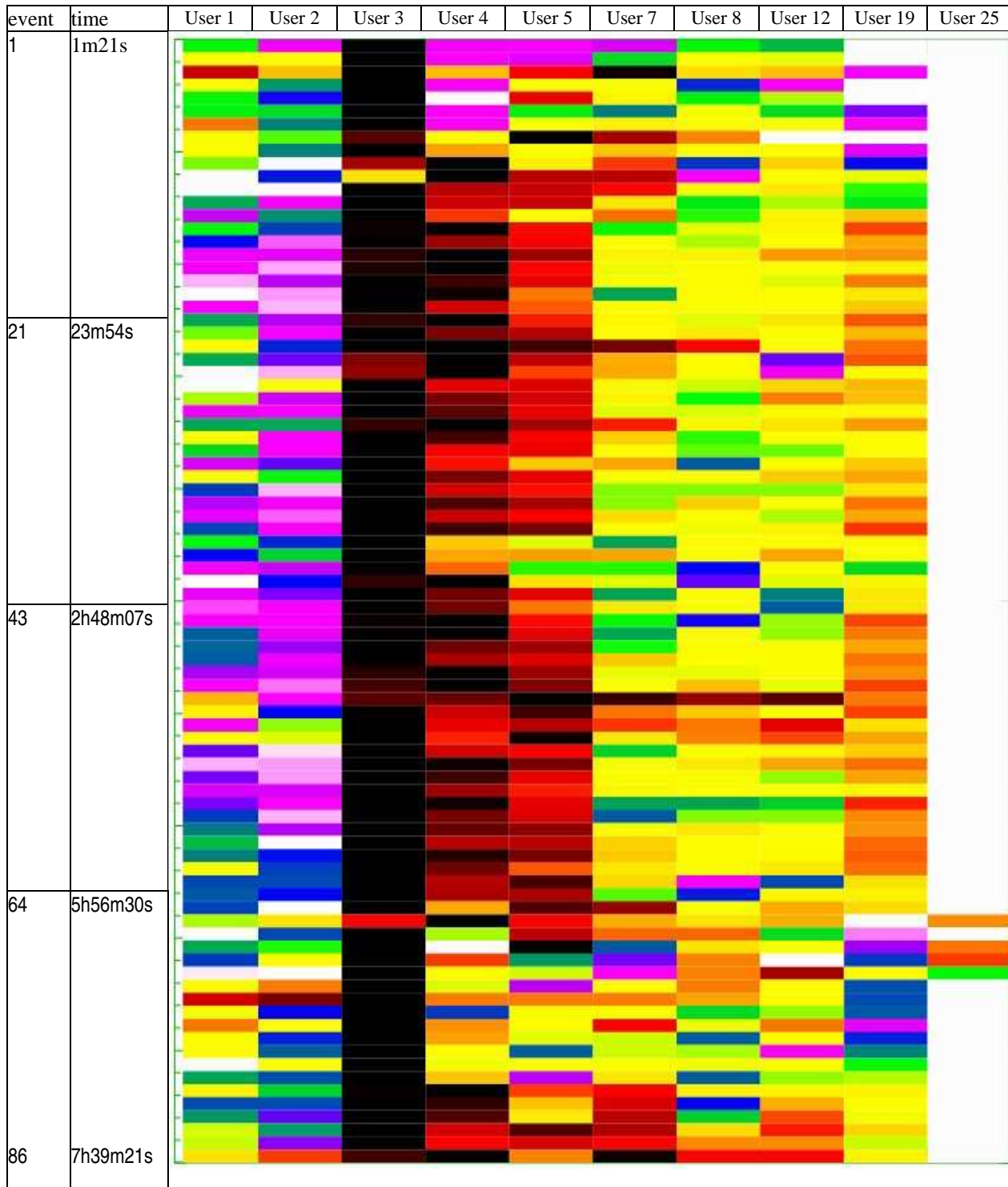


Figure 202: Heat map for testing session 1195 (User 25) with degrees of match to 10 Users. Black color indicates no match, white color indicates match.

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = cov. ratio Evaluation of Disjunction = max

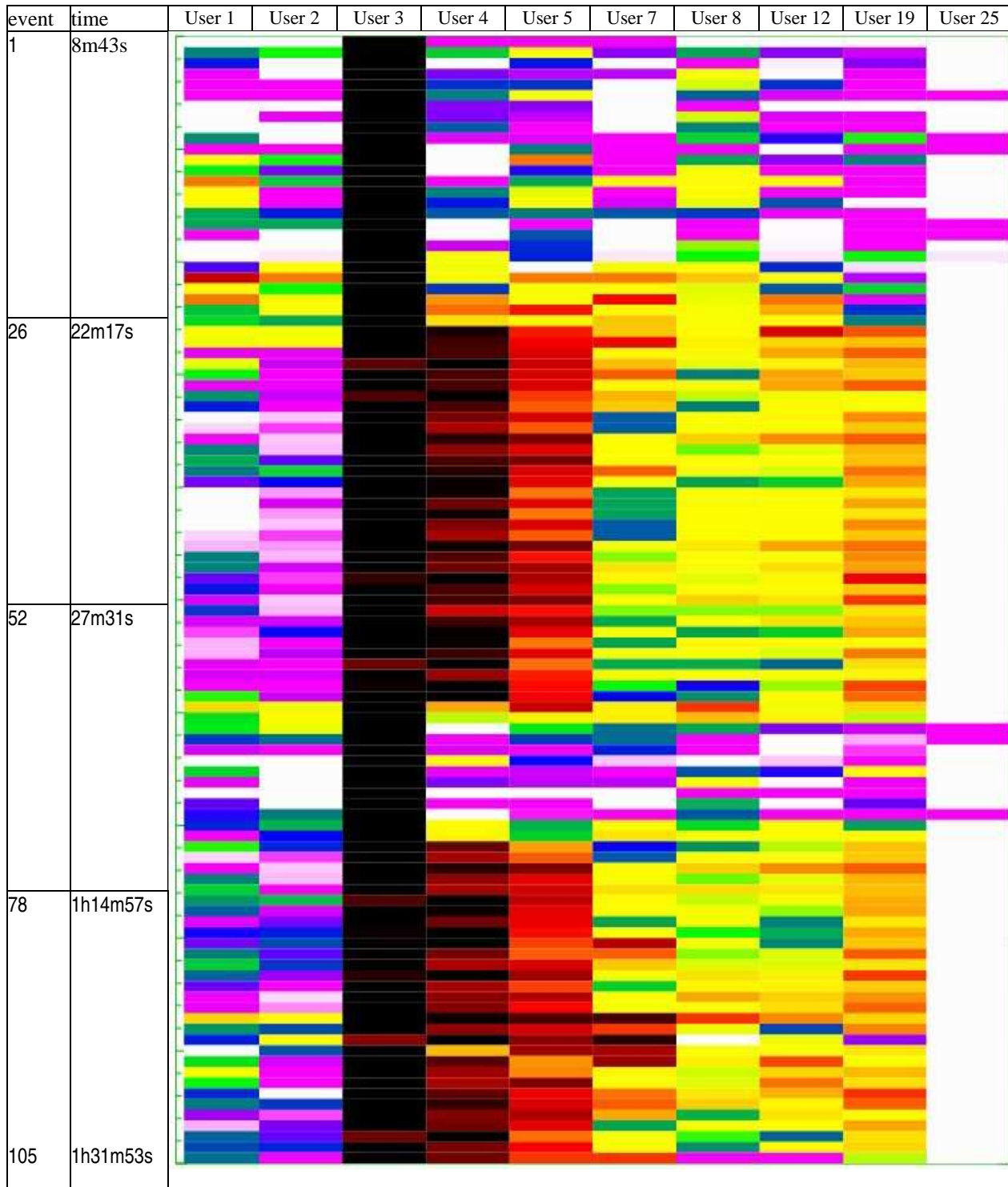


Figure 203: Heat map for testing session 1196 (User 25) with degrees of match to 10 Users. Black color indicates no match, white color indicates match.

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = cov. ratio Evaluation of Disjunction = max

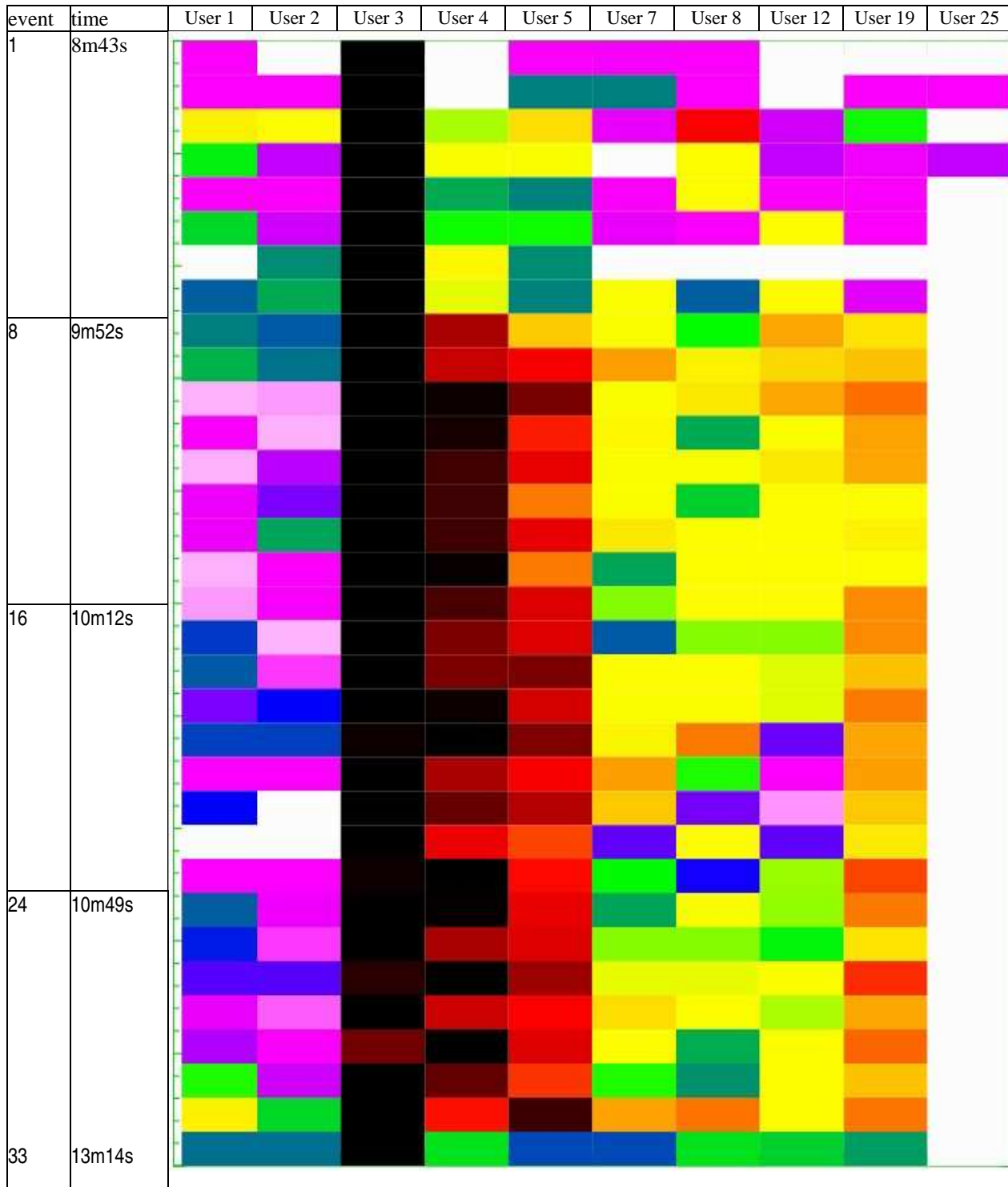


Figure 204: Heat map for testing session 1197 (User 25) with degrees of match to 10 Users. Black color indicates no match, white color indicates match.

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = cov. ratio Evaluation of Disjunction = max

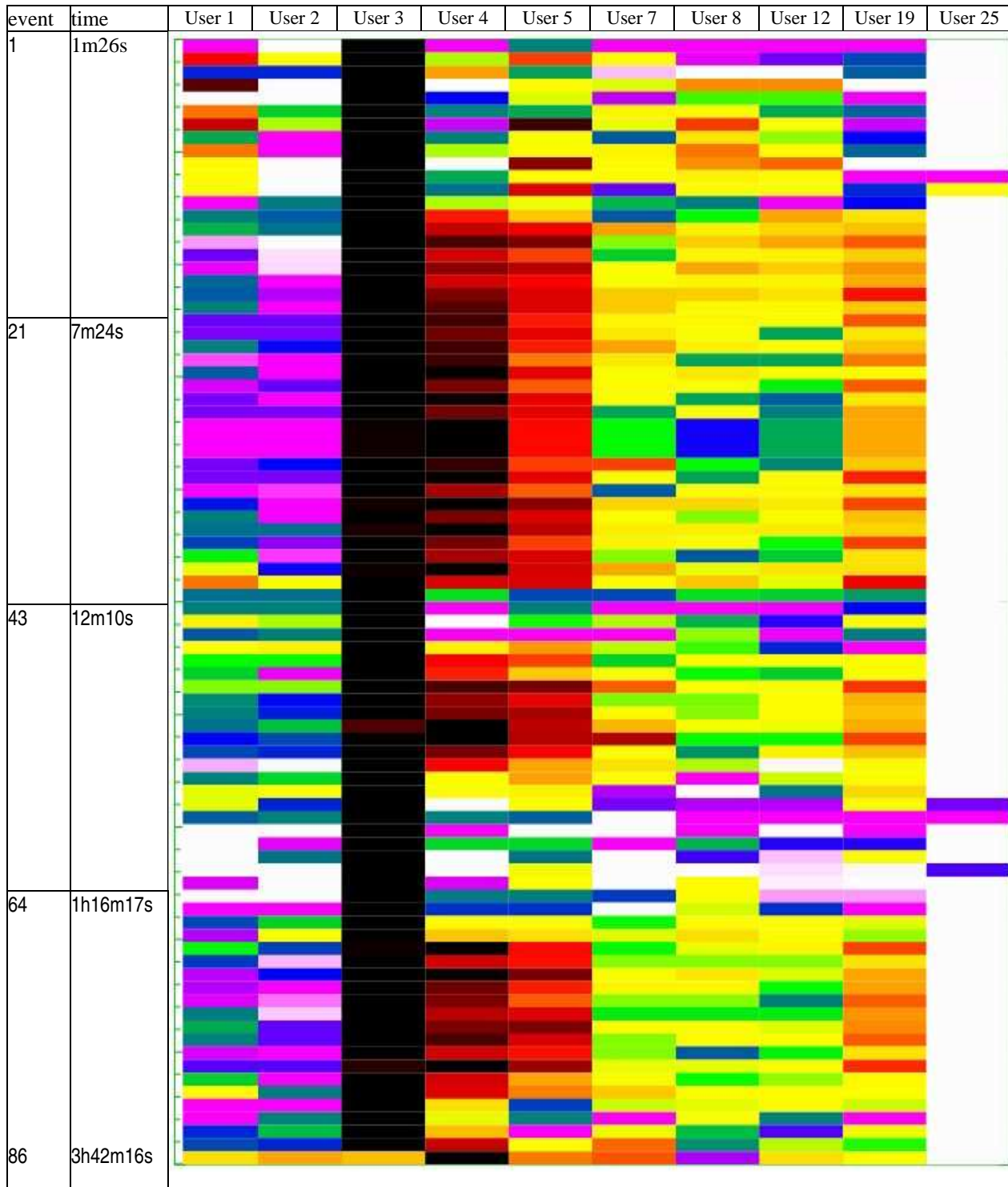


Figure 205: Heat map for testing session 1198 (User 25) with degrees of match to 10 Users. Black color indicates no match, white color indicates match.

Source Data: window records

Training Dataset: Discretization: Dis-3 **Filtering:** not filtered

Testing Dataset: Discretization: Dis-3 **Filtering:** not filtered

Learning Parameters: ambiguity = ignore-for-learning trim = optimal mode = tf Char. descr,

Testing Parameters: Evaluation of Conjunction = cov. ratio Evaluation of Disjunction = max

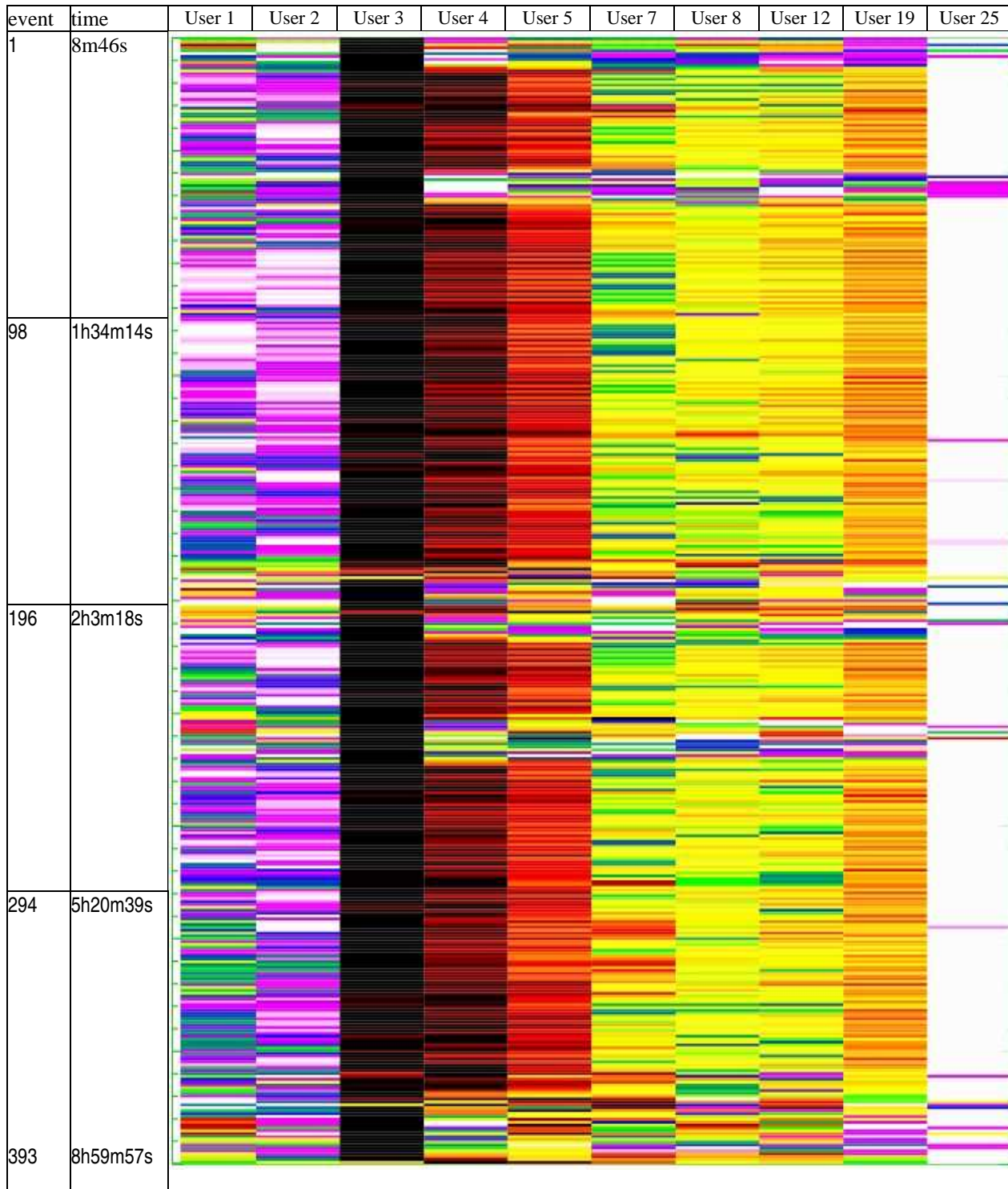


Figure 206: Heat map for testing session 1199 (User 25) with degrees of match to 10 Users. Black color indicates no match, white color indicates match.

A publication of the *Machine Learning and Inference Laboratory*
School of Computational Sciences
George Mason University
Fairfax, VA 22030-4444 U.S.A.
<http://www.mli.gmu.edu>

Editor: R. S. Michalski
Assistant Editor: K. A. Kaufman

The *Machine Learning and Inference (MLI) Laboratory Reports* are an official publication of the Machine Learning and Inference Laboratory, which has been published continuously since 1971 by R.S. Michalski's research group (until 1987, while the group was at the University of Illinois, they were called ISG (Intelligent Systems Group) Reports, or were part of the Department of Computer Science Reports).

Copyright © 2005 by the Machine Learning and Inference Laboratory.