

Data Mining Survey - ver 1.1009

Zheng Zhu

Department of Computer Science and Information Systems,
Birkbeck College, University of London

1 What is data mining

Data mining [1] (the analysis step of the Knowledge Discovery in Databases process, or KDD), a relatively young and interdisciplinary field of Computer Science, is the process of extracting patterns from large data sets by combining methods from statistics and artificial intelligence with database management.

The data mining process models may be defined three phase:

- Pre-processing
- Data mining
- Results validation

1.1 Pre-processing

Before data mining algorithms can be used, a target data set must be assembled. As data mining can only uncover patterns already present in the data, the target dataset must be large enough to contain these patterns while remaining concise enough to be mined in an acceptable timeframe. A common source for data is a datamart or data warehouse. Pre-process is essential to analyse the multivariate datasets before data mining. The target set is then cleaned. Data cleaning removes the observations with noise and missing data.

1.2 Data Mining

Data mining commonly involves four classes of tasks:

- Association rule learning - Searches for relationships between variables. For example a supermarket might gather data on customer purchasing habits. Using association rule learning, the supermarket can determine which products are frequently bought together and use this information for marketing purposes. This is sometimes referred to as market basket analysis.
- Clustering - is the task of discovering groups and structures in the data that are in some way or another “similar”, without using known structures in the data.
- Classification - is the task of generalizing known structure to apply to new data. The output space for classification is one of a finite set of discrete values. For example, an email program might attempt to classify an email as legitimate or spam. Common algorithms include decision tree learning, nearest neighbor, naive Bayesian classification, neural networks and support vector machines.

- Regression - Attempts to find a function which models the data with the least error. The output of regression is a real value.

1.3 Results validation

The final step of knowledge discovery from data is to verify the patterns produced by the data mining algorithms occur in the wider data set. Not all patterns found by the data mining algorithms are necessarily valid. It is common for the data mining algorithms to find patterns in the training set which are not present in the general data set. This is called overfitting. To overcome this, the evaluation uses a test set of data on which the data mining algorithm was not trained. The learned patterns are applied to this test set and the resulting output is compared to the desired output. For example, a data mining algorithm trying to distinguish spam from legitimate emails would be trained on a training set of sample emails. Once trained, the learned patterns would be applied to the test set of emails on which it had not been trained. The accuracy of these patterns can then be measured from how many emails they correctly classify. A number of statistical methods may be used to evaluate the algorithm such as ROC curves.

If the learned patterns do not meet the desired standards, then it is necessary to reevaluate and change the pre-processing and data mining. If the learned patterns do meet the desired standards then the final step is to interpret the learned patterns and turn them into knowledge.

2 Current survey of data mining

Early pattern identification includes the Bayes's theorem. Modern data mining methods are aided by the computer technologies. Since 1950s, perceptron, neural networks, genetic algorithms, decision trees and support vector machines have been developed. The data mining area is fast moving.

The highlights of the current survey of the data mining are shown below [2]

- **FIELDS and GOALS:** Data miners work in a diverse set of fields. CRM / Marketing has been the Number 1 field in each of the past four years. Fittingly, “improving the understanding of customers”, “retaining customers” and other CRM goals are also the goals identified by the most data miners surveyed.
- **ALGORITHMS:** Decision trees, regression, and cluster analysis continue to form a triad of core algorithms for most data miners. However, a wide variety of algorithms are being used. This year, for the first time, the survey asked about Ensemble Models, and 22% of data miners report using them.
- **MODELS:** About one-third of data miners typically build final models with 10 or fewer variables, while about 28% generally construct models with more than 45 variables.
- **TOOLS:** After a steady rise across the past few years, the open source data mining software R overtook other tools to become the tool used by more data

miners (43%) than any other. STATISTICA, which has also been climbing in the rankings, is selected as the primary data mining tool by the most data miners (18%). Data miners report using an average of 4.6 software tools overall. STATISTICA, IBM SPSS Modeler, and R received the strongest satisfaction ratings in both 2010 and 2009.

- TECHNOLOGY: Data Mining most often occurs on a desktop or laptop computer, and frequently the data is stored locally. Model scoring typically happens using the same software used to develop models. STATISTICA users are more likely than other tool users to deploy models using PMML.
- CHALLENGES: As in previous years, dirty data, explaining data mining to others, and difficult access to data are the top challenges data miners face, 2010 year data miners also shared best practices for overcoming these challenges. Read about their experiences overcoming these challenges.
- FUTURE: Data miners are optimistic about continued growth in the number of projects they will be conducting, and growth in data mining adoption is the number one "future trend" identified. There is room to improve: only 13% of data miners rate their company's analytic capabilities as "excellent" and only 8

3 Top 10 Data Mining Algorithms

A poll had been conducted on a international data mining conference regarding to the top 10 algorithms in data mining shown in the below table:

	Algorithms	Category
1	C4.6	Classification
2	K-means	Clustering
3	SVM	Statistical Learning
4	Apriori	Association Analysis
5	EM	Statistical Learning
6	Page-Rank	Link Mining
7	Adaboost	Ensemble Learning
8	KNN	Classification
9	Naive Bayes	Classification
10	CART	Classification

Table 1: Top 10 Data Mining Algorithms

4 Data mining algorithms and their applications

Before we go through the details, we will first introduce the feature types of the instances that can be used as input. In general, there are four types of feature values at the measurement level.

- Nominal. The value of a nominal instance is one of a different collection of names; i.e., nominal values provide only information that can distinguish one instance from another. A class label is one example of a nominal feature.
- Ordinal. The value of an ordinal feature provides information to the order of an instance. Document rank from a search engine is one example of an ordinal feature.
- Interval. For interval features, the difference between values are meaningful. Calendar dates are an example of an interval feature.
- Ratio. For ratio features, both differences and ratios are meaningful. Term frequency in a document is a form of ratio data.

Association rules [3] are an important class of regularities in data. Its objective is to find all co-occurrence relationships. One example of association rules is the market basket data analysis, which aims to discover how items purchased by customers in a supermarket are associated. Apriori algorithm [4] is widely used to solve such problem.

However, association rules do not consider the order of transactions. In practice, it may be interested to know whether people buy some items in sequence. Sequence pattern mining was developed to solve this problem. Two popular algorithms to solve this problem are GSP [5] and prefixSpan [6]. It is used in web usage mining to find the visitor's navigation pattern in one web site so that it is possible for the web site to optimize its structure. It also applies to linguistic or language patterns mining [7].

Supervised learning is one of the most successful research topics in practice. It is used in almost every domain. It learns from data, which are collected in the past and represent past experiences to produce a classification function, this classification function can be used to predict the future data. The data are described by a set of attributes. It also has a special target attribute, the class attribute. The class attribute has a set of discrete values.

A machine learning problem can be characterised by the following components:

- The input space contains instances or objects under investigation: The objects in our context are usually documents. They can be represented by a vector, where each element of the vector indicates one feature of the object. The data in the input space cannot usually feed into the learning machine algorithm directly and preprocessing is required to achieve a desired performance.
- The output space contains the learning target with respect to the input objects. For example, given a web page, the output space can be either yes or no indicating whether or not it belongs to *sports* category. However, the output space is not limited to a single decision or a scalar number, rather it may have structure.
- The hypothesis space defines the function space for mapping the input to the output. In order to find the optimal hypothesis, training data is necessary to tune the model. One principle approach to evaluate tuning is to define a

loss function (or equivalently a likelihood function¹) based on the predicted output in comparison to the true output. Thereafter optimisation methods are applied to find the optimal results. However, empirical results show that if we only optimise a model for the training data, the tuned model can work perfectly for training data, yet make poor predictions for the test data; this is called *over-fitting*. To achieve a better generalization ability, regularisation [8] may be adopted into the loss function. The main idea of regularisation is to introduce the trade-off between the empirical loss and model complexity to the problem. There are at least three different ways of regularization: 1) Explicitly via constraints on model complexity. 2) Implicitly through incremental building of the model. 3) Implicitly through the choice of robust loss functions. Another approach is called the Bayesian model, in this case, loss minimisation corresponds to the likelihood function and the regularisation can be regarded as a prior on the model. One advantage of the Bayesian model is that it can support nonparametric inference [9], thus the model is less restricted than a parametric model.

The underlying goal is to produce a classification function which predict the class of unseen data as accuracy as possible. Association rule can also be used to classification, which is find the strong correlation between the class attribute and the data attribute. There are other classification algorithms [10], i.e., decision tree, Naive Bayesian, KNN, logistic regression, neural network, CART, support vector machines, if the class attribute is a sequence pattern, hidden markov model can be used.

If there is no class attribute, we can still find the similar data instance based on its attribute. Clustering or exploratory data analysis is one technology to explore the data to find some intrinsic structures in the data. The goal of clustering is to make use of data similarity/distance to separate a finite, unlabeled data set into a finite, discrete set of hidden groups or clusters.

Some proximity measures are commonly used in clustering. Perhaps the most commonly used distance metric is the Euclidean distance. However, there are other distance metric available, i.e, Edit distance [11], Kullback-Leibler divergence [12] and Cosine similarity [13] so on.

Clustering algorithms [14] can be categorised as flat clustering (as in K-means) or hierarchical clustering (as in agglomerative hierarchical clustering).

The K-means algorithm [15] is one of the best known and most popular clustering algorithms. It seeks an optimal partition of the data into K clusters by, for example, minimising the sum of the squared distance between each data point to the mean of the corresponding data points in its cluster. It involves an iterative optimisation procedure to seek an optimal solution, which often leads to a local minimum rather than a global one.

Agglomerative clustering belongs to the category of hierarchical clustering. It starts with n clusters, each of which includes exactly one data point, then a series of merge operations is carried out until all clusters are merged into

¹ In Gaussian distribution, maximising the likelihood is the same as minimising the loss function

the same cluster. The merge operation occurs between two clusters with the highest intercluster similarity. Finally it collapses down to as many clusters as requested. This kind of clustering is widely used in document clustering and other IR applications [16].

Although clustering is different from classification, it is possible to incorporate clustering into classification to enhance its performance, e.g. K-means classifier [17].

Clustering is widely used in CRM to find the similar users, in web results mining, clustering is adopted to group the similar results together to improve the users' search experience.

Unlike classification, to evaluate cluster validity is problematic due to the absence of ground truth knowledge. In this case, it is possible to use the objective function to evaluate the clustering quality, i.e., the objective function is the one that should be minimised by the clustering algorithm.

5 Other topics on data mining

In addition to the algorithms we discussed before, there are other algorithms, i.e., semi-supervised learning [18], reinforcement learning [19] and so on.

Semi-supervised learning is to make use of both labeled and unlabeled data for training as in many cases labeling data is time-consuming and expensive. Some algorithms include self-learning, Co-learning, semi-supervised support vector machine. This learning method can be used almost every area supervised learning can be applied, i.e., natural language processing, text mining.

Reinforcement learning is concerned with how an agent ought to take actions in an environment so as to maximize some notion of cumulative reward by taking a sequence of decisions. One classic algorithm is Markov decision process. The applications include robot navigation, games.

6 AI in Education

AI technology has been widely used in Education. All data mining tasks are adopted in intelligent systems and cognitive science for education purpose. From previous section, we know that the learning model can be categorized as either predictive model (also known as "supervised learning") or descriptive model (referred to as "unsupervised learning"). We will review some of applications below from the two aspects.

Data mining can be applied as predictive model. One application is to predict a student's future performance. Ryan et. al [20] make use of linear model to predict future transfer of learning. Their results show that using such model, it can obtain higher correlation of a transfer post-test and the linear model than Bayesian knowledge tracing, which is the baseline in their experiment. Gian-Marco Baschera et. al [21] leveraged dynamic Bayesian network to predict focused, receptive states and forgetting in spelling learning environment. Regression analysis demonstrates the advantages of feature processing for engagement

modeling. To mimic causal learning, Conscious Emotional Learning Tutoring System (CELTS) used a sequential rule mining algorithms to extracts frequently occurring events from its past experience; each event is associated with either positive or negative emotions. The CELTS system having a causal memory is an effective way to understanding the cause of learner’s mistakes. Some advanced machine learning algorithms are also used in AIED. Peter Hastings et. al [22] use regular expression, latent semantic analysis and support vector machines to evaluate the student essays to documents model. Their results showed that pattern matching approach outperform others algorithms in general. Moreover, video retrieval was also studied in [23] for the purpose of supporting human assessing the learning environment. Multiple Instance learning has been applied to this problem. In narrative-centered learning environment, it is necessary to make early predictions about how effectively students will utilize learning resources, Lucy R Shores et. al [24] adopted naiveBayes, Decision Tree and SVM to build the predictive model and the results showed that support vector machine and naive Bayes models offer considerable promise for prediction.

Data mining algorithm can also be used as descriptive model. Ari Bader-Natal et. al [25] use a generalized linear mixed model to model the effectiveness of nine activities within a self-directed learning environment and the coefficients can indicate certain characteristic of the corresponding activity. Ilya M Goldin and Kevin D. Ashley [26] developed and evaluate hierarchical Bayesian models relating instructor scores of student essays to peer scores based on two peer assessment rubrics. Again we can gain insight by looking at the estimated parameters.

Other research investigate the correlation between several features. Kate Forbes-Riley and Diane Litman studied the correlation between the disengagement label and six different labels of disengagement type and found that the individual types of disengagement correlate differently with learning [27]. Joseph F. Grafsgaard et. al demonstrate the relation between facial expressions and both dialogue and task context. [28].

7 Descriptive Statistics of the Dataset

Our database can be divided into three sessions. They are collected on Dec 08, 2010, 27 Jun, 2011 and 29 Jun, 2011.

7.1 Model Length

For session 1, there are totally 91 models. We define the *model length* as the number of event indicators occurred in the model. The minimum model length is 4 and the maximum model length is 458. The mean of model length is 100.54 and the standard deviation of it is 99.75. On average, each user generated 3.37 models.

The model length is shown in figure 1. The index in figure 1(b) is corresponding to model id.

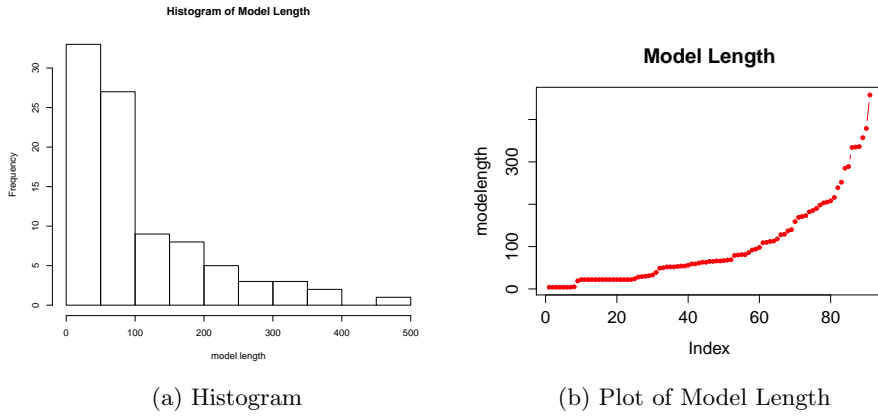


Fig. 1: the Model Length in Session 1

For session 2, there are totally 67 models. The minimum model length is 2 and the maximum model length is 1648. The mean of model length is 204.43 and the standard deviation of it is 259.12. On average, each user generated 2.72 models.

The plot of the model length is shown in figure 2.

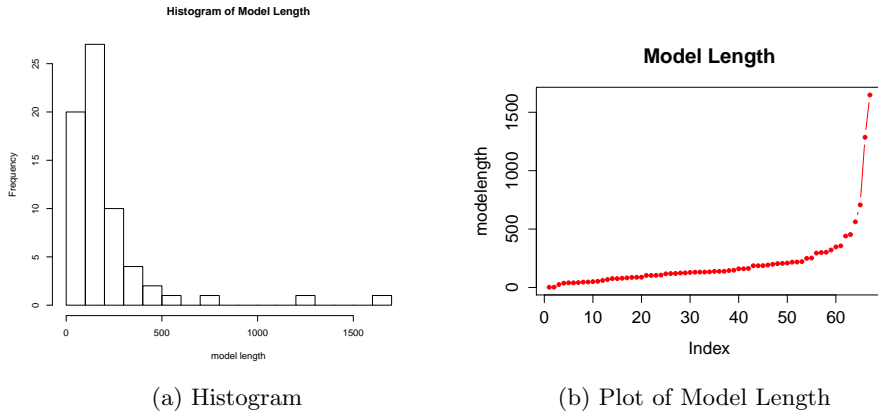


Fig. 2: the Model Length in Session 2

For session 3, there are totally 33 models generated². The minimum model length is 2 and the maximum model length is 1340. The mean of model length is

² There are 3 more models generated, however, such models do not contain the event indicator type id, so here we exclude them

398.67 and the standard deviation of it is 315.05. On average, each user generated 1.50 models.

The plot of the model length is shown in figure 3.

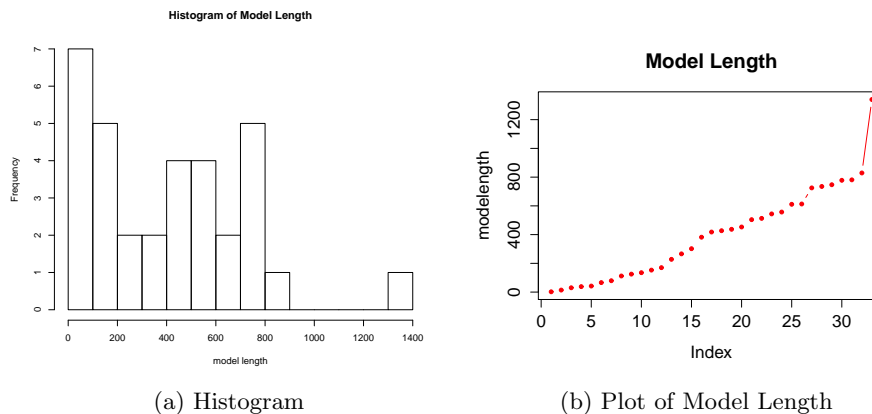


Fig. 3: the Model Length in Session 3

7.2 Model Length for Achieved Task Goal

We also can consider the model containing the achieved task goal as valid model no matter which task goal achieved and treat other model as noisy. Therefore we can remove some models. For session 1, there are 26 models, the minimum model length is 24 and maximum model length is 458, the mean is 213 and stand deviation is 108.5. Figure 4 is the plot of the valid model length in session 3. For session 2, there are 66 models in total, the minimum model length is 3 and maximum model length is 1648, the mean is 207.5 and stand deviation is 260. Figure 5 is the plot of the valid model length in session 3. For session 3, there are 22 models, the minimum model length is 66 and maximum model length is 1340, the mean is 515 and stand deviation is 307. Figure 6 is the plot of the valid model length in session 3.

Amongst the above models, there are 113 models achieved task goal 1 and 105 models achieved task goal 1 and 2. Note there is one model (model id 206264) got only task goalid 3 achieved, therefore, if we sum up the models from each session, we will have 114 models rather than 113 models. The distribution of it is shown in Figure 7.

There are 64 models achieved task goal 1,2 and 3 and 44 models achieved task goal 1,2,3 and 4. The distribution of it is shown in Figure 8.

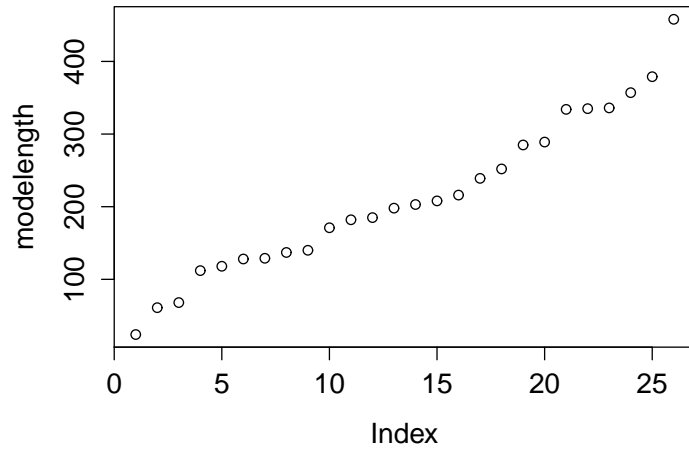


Fig. 4: Goal Achieved Model Length in Session 1

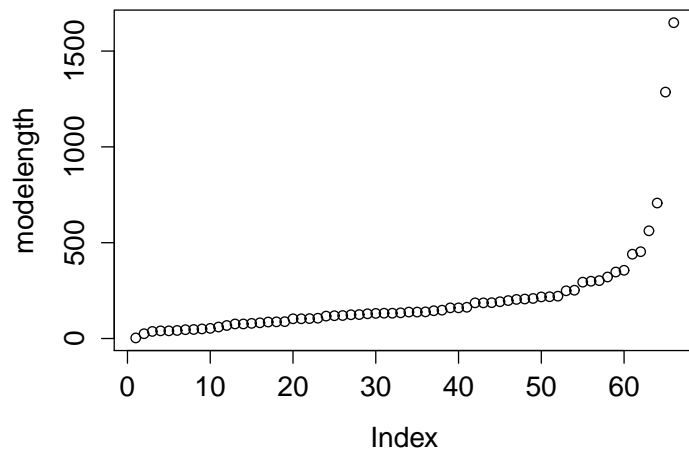


Fig. 5: Goal Achieved Model Length in Session 2

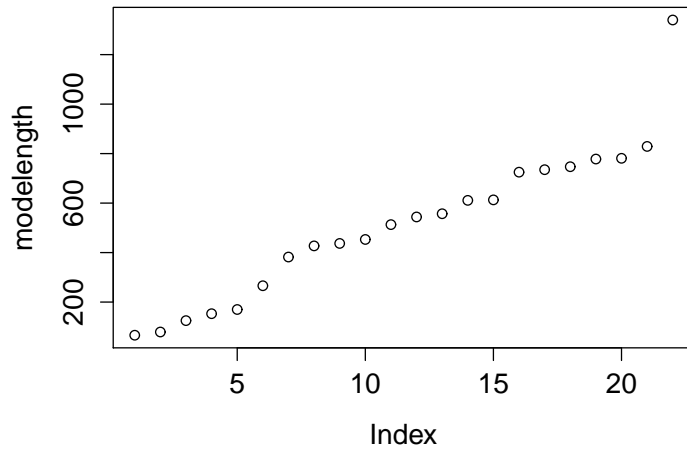
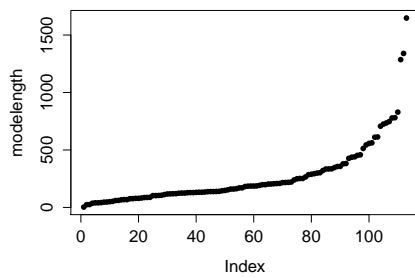
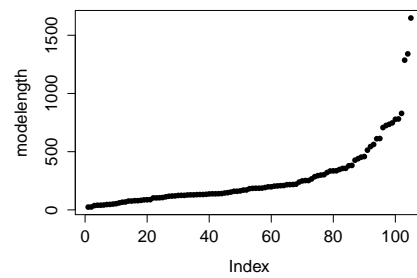


Fig. 6: Goal Achieved Model Length in Session 3

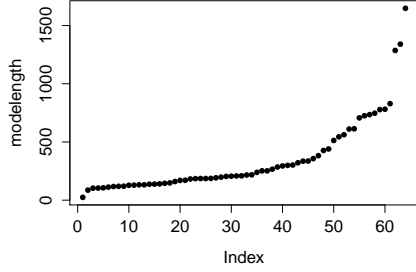


(a) Length of the Models Achieved Task Goal 1

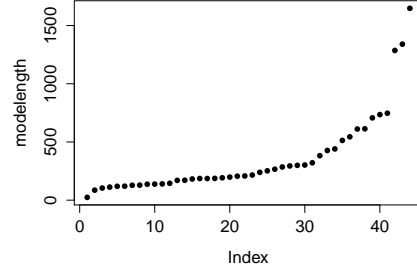


(b) Length of the Models Achieved Task Goal 1,2

Fig. 7: Length of the Models Achieved Task Goal



(a) Length of the Models Achieved Task Goal 1,2,3



(b) Length of the Models Achieved Task Goal 1,2,3,4

Fig. 8: Length of the Models Achieved Task Goal

7.3 Descriptive Statistics for Indicator Type

We will look at indicator type i individually. First, we only consider the frequency of the indicator type in one model and model frequency for indicator type i . Here model frequency for indicator type i is defined as the proportion of the number of models containing indicator type i . Therefore indicator type frequency is a local criteria of the popularity of the indicator type in one model and it can be greater than 1. Model frequency for indicator type is the global criteria and indicates the commonness of the indicator type in all models. Model frequency is a value between 0 and 1. 0 means the indicator type did not occur in any model and 1 means the indicator type occur in every model. Table 2 presents the mean of indicator frequency (IF) and model frequency (MF) for three sessions (S1 stands for session 1, S2 stands for session 2 and so on).

Indicator Type	IF(S1)	IF(S2)	IF(S3)	MF(S1)	MF(S2)	MF(S3)
1001	0.31	0	0	0.18	0	0
1002	3.25	0	0	0.56	0	0
1003	17.3	5.75	24.39	0.74	0.72	0.82
1004	0.25	41.8	124	0.11	0.24	0.55
1005	0.04	1.49	2.48	0.01	0.30	0.48
1006	9.38	7.69	15.1	0.56	0.79	0.82
1007	0.92	0.73	1.36	0.26	0.27	0.52
1008	0.46	0.33	0.24	0.14	0.15	0.12
1009	0.14	0.1	0.3	0.09	0.06	0.18
1010	18.3	9.97	30.2	0.66	0.91	0.85
1011	1.64	1.18	3.67	0.48	0.48	0.76
1014	7.1	37.2	52.6	1	0.99	0.94

Continued on next page

Indicator Type	IF(S1)	IF(S2)	IF(S3)	MF(S1)	MF(S2)	MF(S3)
1015	19.8	11.9	34.5	0.74	0.88	0.85
3002	0.05	0	0.24	0.02	0	0.06
3006	0	0.01	0.06	0	0.01	0.06
3007	0	0.06	0.42	0	0.04	0.27
3008	0	1.51	1	0	0.43	0.24
3009	0	3.24	7.06	0	0.87	0.82
5001	1.54	5.18	4.21	0.27	0.99	0.64
5002	0.24	1.13	0.79	0.08	0.28	0.21
5003	0.89	23.88	22.94	0.22	0.93	0.76
5004	0.01	0.13	0.09	0.01	0.12	0.09
6001	18.22	49.73	72.27	1	1	1
6002	0.54	1.03	0.45	0.18	0.28	0.24
6003	0.20	0.43	0.06	0.13	0.15	0.06

Table 2: Event Indicator Frequency and Model Frequency

The Table 3 shows the state indicator frequency and model frequency for state indicator in the sessions.

Indicator Type	IF(S1)	IF(S2)	IF(S3)	MF(S1)	MF(S2)	MF(S3)
2002	2.57	6.70	6.44	1.00	1.00	1.00
2003	1.82	1.90	2.75	1.00	1.00	1.00
2004	2.82	1.48	5.42	1.00	1.00	1.00
2006	2.22	1.61	3.5	1.00	1.00	1.00
2007	0	4.36	4.22	0	1.00	1.00
2008	0	1.82	1.75	0	1.00	1.00
2013	0	7.64	1.94	0	0.91	0.75
4002	1.97	1.52	2.06	1.00	1.00	1.00
4003	1.31	1.51	1.50	1.00	1.00	1.00
4004	1.23	1.54	1.42	1.00	1.00	1.00
4005	0.01	0	0.14	0.01	0	0.03
4006	0.43	0.40	0.56	0.21	0.31	0.25
4007	0.36	2.09	1.39	0.21	0.78	0.5
4008	2.76	1.96	3.53	1.00	1.00	1.00
4009	0.20	1.73	2.44	0.09	0.75	0.61
4010	2.46	4.84	6.28	1.00	1.00	1.00
4012	0	1.39	1.53	0	1.00	1.00
4014	1.32	2.13	2.64	0.44	0.73	0.28

Table 3: State Indicator Frequency and Model Frequency

Each model consists of various indicator types and each indicator type belongs to one of four status, negative (-1), neutral (0), positive (1) and intervention related (2). Note the intervention related (2) includes indicator types of feedback

request, intervention generate and intervention show. Therefore, each model is one sequence of the combination of the four status. For every session composed of many models, we can obtain the number of those status. The result is displayed in figure 9. The heights of the bars in figure 9 correspond to the conditional

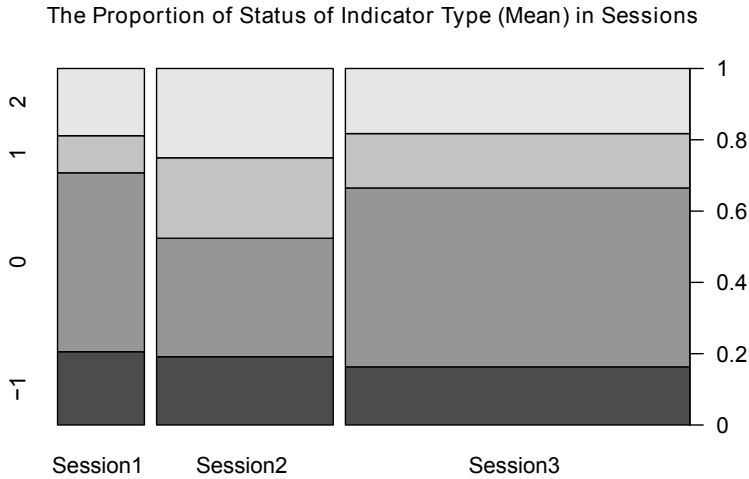


Fig. 9: The Proportion of the Number of Status in Sessions

relative frequencies of status in every session and the widths of the bars corresponds to the relative frequencies of session. We can see that the number of indicators grows along the time, this may be due to the fact that the student are more familiar with the system and produce more interaction with the system. Moreover, the number of negative indicator type decreases.

Similarly, we can plot the figure to compare the distribution of status of state indicators across the different session. The result is shown in Figure 10.

If we ignore the intervention related indicators, the result is show in figure 11.

The distribution of intervention related indicators is shown in figure 12. figure 9.

Each intervention generated indicator is associated with feedback strategy. The feedback can be non feedback,...(Sergio/Manolis, can you fill in other feedback strategy). Here we present the result of feedback strategy constituent for intervention generated indicator by each session in Figure 13 and Table 4.

Feedback Strategy	Session 1	Session 2	Session 3
0	19	0	25
2	4	52	2
3	5	17	5

Continued on next page

Feedback Strategy	Session 1	Session 2	Session 3
6	331	448	308
7	20	40	23
11	10	24	2
12	11	0	4
14	0	657	405
900	0	47	14
999	1246	2038	1596
1000	1	9	1

Table 4: Feedback Strategy Constituent for Intervention Generated Indicator

We can also drill down to the constituent of the models in different sessions, which is shown in table 5.

Indicator Type	Session 1	Session 2	Session 3
1001	28	0	0
1002	296	0	0
1003	1575	385	805
1004	23	2803	4101
1005	4	100	82
1006	854	515	497
1007	84	49	45
1008	42	22	8
1009	13	7	10
1010	1663	668	996
1011	149	79	121
1014	646	2490	1736
1015	1798	794	1138
3002	5	0	8
3006	0	1	2
3007	0	4	14
3008	0	101	33
3009	0	217	233
5001	140	347	139
5002	22	76	26
5003	81	1600	757
5004	1	9	3
6001	1658	3332	2385
6002	49	69	15
6003	18	29	2

Table 5: Session Constituent of Indicator Types

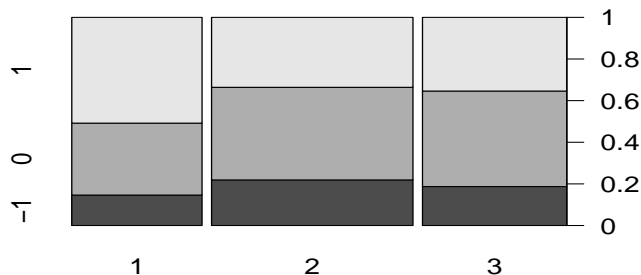


Fig. 10: The Proportion of the Number of Status for State Indicators in Sessions

7.4 Transition Matrix for Indicator Type

A sequence of indicator types forms one model. Some patterns may exist in such sequences. Here we look at the transition matrix, which is used to describe the transitions of a Markov chain. Given a finite indicator space, $P_{ij} = P(j|i)$ is the probability of moving from i to j in one time step and it is the element at row i and column j . It is usually normalized by row, quantifying the transition probability from indicator i to any other indicator. We can also normalize the matrix by column; in this case, it measures the incoming probability to state j from various other states. We add to artificial points to the systems. The notation s means the start point, which is the first indicator before users interact with the system. The notation e is the end point, which is the last indicator for every model.

The transition matrix are shown from Figure 14 to Figure 19.

The indicators associated with one circle indicate that there is some loops for the indicator. The thickness of the line indicates the value of a transition probability. The thicker the line is, the higher the probability will be. The red lines are associated with the probability less than 0.2. The black lines are associated with the probability greater or equal to 0.2.

We would like to know whether there is a statistically significant difference between the transition in different sessions. Here we run the 2 sample t test. The procedure is as follows: Given a transition from a to b , and assume we have 33 students for each session. Then for every student, we can count the number of times such a transition happens by session as N_i^j , where i is the student index and j is the session number. Therefore, we can form one vector for each session j with i from 1 to 33, the element of the vector is the number of times such a transition happens.

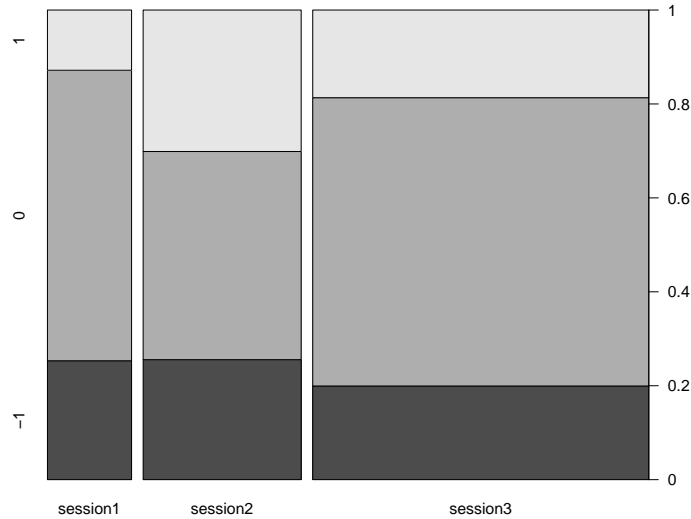


Fig. 11: The Proportion of the Number of Status without Intervention in Sessions

by one user and the length of the vector is the 33. The result of the significant test is shown in the spreadsheet.

For session 1 and 2 and indicator 6001, the significant difference occurred on $\langle 6001, 5003 \rangle$, the transition probability for the session 1 is 0.009 and for session 2, it becomes 0.157. Note in session 3, the probability is 0.105. For $\langle 6001, 1014 \rangle$, there is no significant difference between session 2 and session 3, but it exists in session 1 and session 2,3. For session 1, the transition probability is 0.278, while for session 2 and 3, they jumped to 0.472 and 0.493.

7.5 Additional Information: Duration between the Last Feedback Strategy and Non Feedback Strategy

One issue we may be interested in is related to intervention generation occurrence (IGO). For each IGO, there is one feedback strategy associated with it. Feedback strategy value 999 means there is no feedback available now. (Sergio, it is correct?) Other numbers are corresponding to different feedback strategies. First, we would like to find the duration of IGO which is the last not 999 feedback strategy to the IGO which is 999 feedback strategy. Note not every model has such pattern, some models may end up with non 999 feedback strategy, other models may not exist non 999 feedback strategies. However, we will focus on such pattern first.



Fig. 12: The Proportion of the Number of Intervention Indicators in Sessions

For session 1, the mean value of the duration is 13.18 seconds and standard deviation is 12.53. For session 2, the mean is 12.79 seconds and the standard deviation is 12.65. For session 3, the mean is 15.85 seconds and the standard deviation is 12.97.

Figure 20 shows the boxplot of the duration for the three sessions. We can see that for session 3, there are less outlier, which may indicate that the users' performance became more normal as they got familiar with the system compared to the previous sessions. The mean of duration increases slightly.

	Pattern Number	Model Number
Session 1	112	91
Session 2	110	67
Session 3	90	33

Table 6: Pattern/Model Number for Each Sessions

Table 6 implies that such pattern occurred more when the users get to understand how to interact with the system.

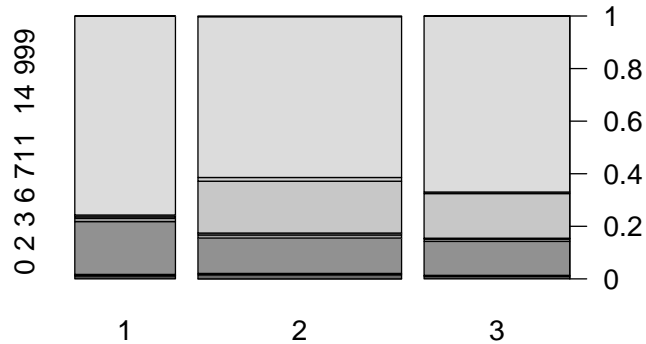


Fig. 13: Feedback Strategy Proportion for Intervention Generated Indicator

7.6 Frequent Sequential Pattern Mining

We may be interested in some sequential patterns which imply users' common interaction patterns to Migen system. Here we use SPADE algorithm [29] to mining the frequent sequential pattern, SPADE is a very efficient algorithm for the task. The toolkit we use is a package call `arulesSequence` for R <http://cran.r-project.org/web/packages/arulesSequences/index.html>.

We set the min support to be 0.7. If the min support is set too low, we may obtain a large result set. If it is set too high, we may not mine any pattern. So we set it to 0.7 based on our experiences. And any pattern whose support is greater than 0.7 should be detected by SPADE. We can find various length of patterns and here I only report the patterns with longest length, any subsequence of them satisfies our support constraints.

In Session 1 dataset. The longest patter is $\langle l, m, w, c \rangle^3$. The support is 0.74. There are totally 15 pattern found including 4 one item patterns.

In Session 2 dataset, there are 319 patterns found. The longest patterns have 7 items. They are $\langle l, m, r, s, u, w, j \rangle$ with support 0.85, $\langle l, m, r, s, u, w, f \rangle$ with support 0.79, $\langle m, r, s, u, w, f, j \rangle$ with support 0.79, $\langle l, r, s, u, w, f, j \rangle$ with support 0.79, $\langle l, m, s, u, w, f, j \rangle$ with support 0.79, $\langle c, l, m, s, u, w, j \rangle$ with support 0.72, $\langle l, m, r, u, w, f, j \rangle$ with support 0.79, $\langle l, m, r, s, w, f, j \rangle$ with support 0.79 and $\langle l, m, r, s, u, f, j \rangle$ with support 0.79.

³ The meaning of the alphabet is shown in the appendix.

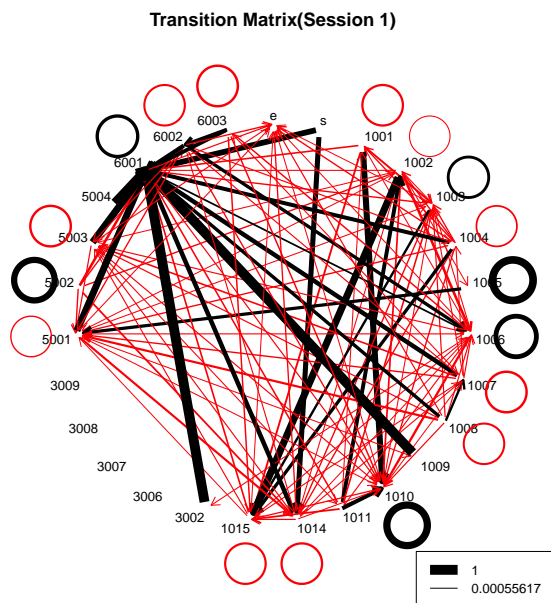


Fig. 14: Transition Matrix(Session 1)

We use the Session 3 dataset. There are total 263 patterns we found. The length of the longest pattern is 7. They are $\langle c, l, m, u, w, f, j \rangle$ and $\langle c, l, m, r, w, f, j \rangle$. Their support is 0.73.

We can find that certain pattern or subsequence pattern occurred across the session.

7.7 Sequence Clustering

If we can segment the users according to their intervention patterns, our corresponding instruction can be more useful to particular groups. Therefore, we did some clustering on the models.

We used the data from session 3. We know that session 3 contains the most up to date data and the quality of the data is better than the other two sessions.

The original sequential data consists of indicator type id. We first map those type id to alphabet. The indicators in Table 2 are mapped to a, b, \dots, y pairwise. Then each models are translated to the alphabet accordingly. Each model is associated with one user.

The key point is how to measure the distance or similarity between two models. It may depend on our task. Here I use edit distance. However, it does not mean that the edit distance can achieve our goal. One issue is if for two models with different model length, even the indicator types in shorter model

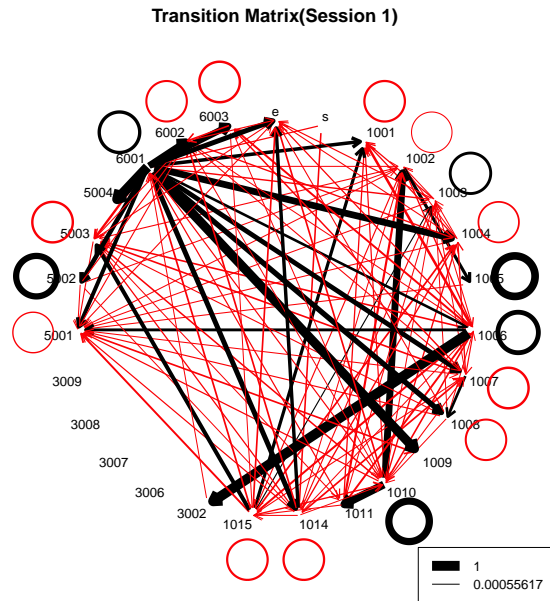


Fig. 15: Transition Matrix Normalized by Column(Session 1)

occur in the long model in the same order, the edit distance can be large as it need many insert operation. If only we can find some way to evaluate the user similarity, we can find one proper distance matrix or combination of distance matrix.

Given 33 models, we can obtain a distance matrix with 33 rows and 33 columns. Based on the distance matrix, we can make use of standard hierarchical cluster analysis, i.e., Initially, each object is assigned to its own cluster and then the algorithm proceeds iteratively, at each stage joining the two most similar clusters, continuing until there is just a single cluster. The result is shown in figure 21.

Note that some users (for example, SB1997724878) generated more than one models and each model has different similarity to other models. So the users occurred more than once in the figure. If we want to avoid such case, we can either take the shortest link or average link as the distance.

A Indicator Type Names

The indicator types colored in red are the event indicator types which did not occur in online database.

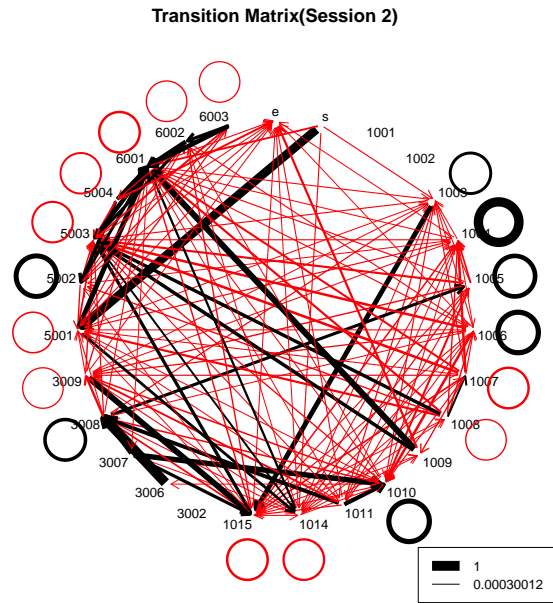


Fig. 16: Transition Matrix(Session 2)

Indicator Type	Alphabet	Indicator Type Name
1001	a	Building block made
1002	b	Pattern made
1003	c	Tile placed
1004	d	Unlocked Number changed
1005	e	Model Rule modified
1006	f	Number created(in any context)
1007	g	Number Unlocked(in any context)
1008	h	Number Locked(in any context)
1009	i	Number Named
1010	j	A shape deleted
1011	k	All shapes deleted
1012		Task done
1013		Start task
1014	l	Correct Local rule created
1015	m	Incorrect Local rule created
1016		Correct Model Rule created
1017		Incorrect Model Rule created
2001		All Local Allocations correct

Continued on next page

Indicator Type	Alphabet	Indicator Type Name
2002		Model is animated
2003		Inactive student
2004		Model consists only of Single Tiles
2005		Model consists only of Patterns
2006		Model has at least one Pattern
2007		Model has at least one overlap not corrected with Negative Tiles
2008		All negative tiles used
2009		All variable named
2010		One or more variable named
2011		One or more constant named
2013		Last shape modified
3001		Construction evaluation
3002	n	Activity document answer right
3003		Activity document answer wrong unknown reason
3004		Activity document answer wrong scaling
3006	o	Building block make plausible
3007	p	Building block make implausible
3008	q	Pattern made plausible building block
3009	r	Pattern make implausible building block
4001		Task pattern built with unit tiles
4002		Plausible building block in use
4003		Animated unmessable pattern
4004		Animated apparent solution pattern
4005		Rhythm detected
4006		Spurious Title
4007		Pattern structure general no shape detected to messup
4008		Correct general allocation
4009		Pattern coloured general
4010		Apparent solution on canvas
4011		Right amount unlocked number
4012		Too many unlocked numbers
4013		No unlocked number
4014		Incorrect local allocation detected
5001	s	Goal checked by system
5002	t	Goal checked by student
5003	u	Goal unchecked by system
5004	v	Goal unchecked by student

Continued on next page

Indicator Type	Alphabet	Indicator Type Name
6001	w	Intervention generated
6002	x	Intervention shown
6003	y	Feedback requested

Table 7: Indicator Type Name

B Data Analyst using R

R is a language and environment for statistical computing and graphics. R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering and etc) and graphical techniques, and is highly extensible. We use R to analyze the indicator type data in this report. Here I will simply describe how to carry out such operation.

R can be downloaded from <http://www.r-project.org/>.

Our data is stored in Oracle Database. The first thing we should do is connect directly to the Oracle database from R. There are two sets of database interfaces available in R:

- RODBC. It allows R to fetch data from ODBC (Open DataBase Connectivity) connections. ODBC provides a standard interface for different programs to connect to database.
- DBI. The DBI package allows R to connect to database using native database drivers or JDBC drivers.

I will explain RODBC as We used RODBC here.

B.1 RODBC

Before we use RODBC, we should install RODBC package to R. A quick way to install RODBC is to use *install.packages* function:

```
> install.packages("RODBC")
```

This command will install the package into your computer. If you want to use any function in that package, you should load it in R first:

```
> library(RODBC)
```

Next we need install the drivers from source. The Oracle ODBC driver can be found at <http://www.oracle.com/technetwork/testcontent/index-087892.html>. Download and run the installer to make sure that drivers are installed.

Then we need configure a DSN for the database. Go to the “User DSn” tab on ODBC Data Source Administrator application (Administrative Tools under Control Panels) and click the “Add...” button. Select the Oracle ODBC driver and click Finish.

We will be prompted for configuration information about Data Source Name, Description, TNS Service Name and User ID.

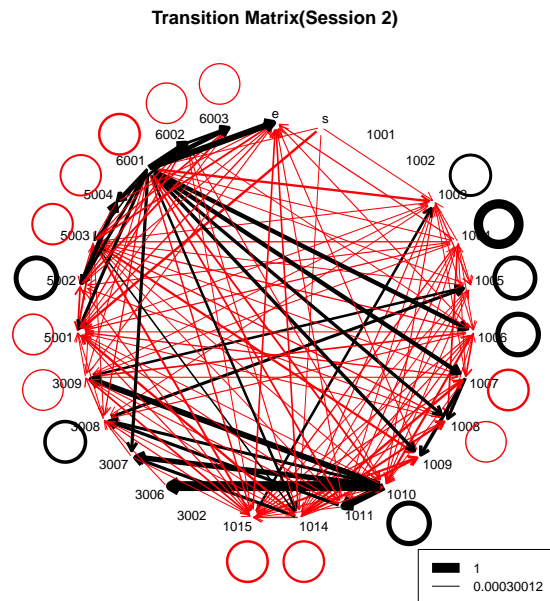


Fig. 17: Transition Matrix Normalized by Column(Session 2)

- Data Source Name - the name used to identify the data source to ODBC. For example, "migen". You must enter a Data Source Name.
- Description - a description or comment about the data in the data source. For example, "Hire date, salary history, and current review of all employees." The Description field is optional.
- TNS Service Name - the location of the Oracle database from which the ODBC driver will retrieve data. This is the same name entered in configuring network database services using the Oracle Net Configuration Assistant (NETCA). The TNS Service Name can be selected from a pulldown list of available TNS names. For example, "orcl". You must enter a TNS Service Name.
- User ID - the user name of the account on the server used to access the data. For example, "scott". The User ID field is optional.

You should be able to access the database through ODBC.

```
> library(RODBC);
> migen <- odbcConnect("migen", uid = "system", pwd = "Migen0501"
, believeNRows = FALSE)
```

The `odbcConnect` is used to open connection to ODBC databases. For detailed explanation about this command, you can use `"?odbcConnect"` to get support from help documents.

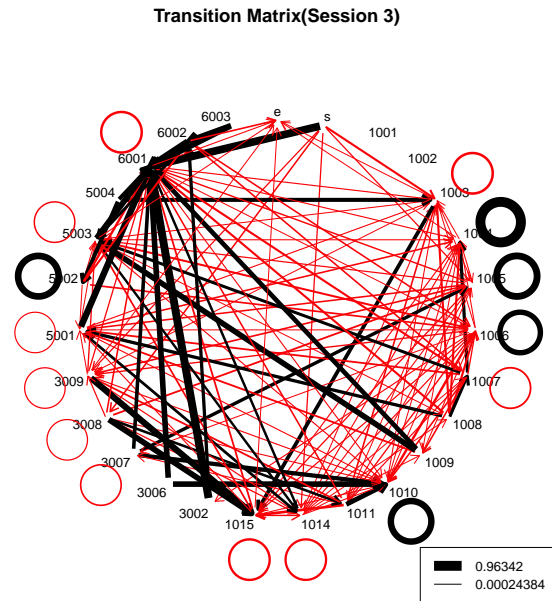


Fig. 18: Transition Matrix(Session 3)

Once we can access the ODBC databases, we can submit an SQL query to it and retrieve the results. For example,

```
data1 <- sqlQuery(migen, "select session_id, status from appstage3
where event_state = 0", errors = TRUE, max = 0, bufsize = 20000,
believeNRows = FALSE);
```

Here *migen* is the connection handle as return by *odbcConnect*. We can see the second argument is the SQL which select two column *session_id* and *status* from our stage area table *appstage3* and we specify the indicator is event by setting *event_state* = 0.

Now we are ready to analyse the data. We can use Cross tabulation to obtain insight of the data as follows:

```
tdata <- table(data$SESSION_ID, data$STATUS)
```

We have used this approach in this report very often.

R supports set operations, i.e., set union, intersection, difference. For example, *union(x,y)*, *intersect(x,y)*, *setdiff(x,y)* achieve those function. Here *x,y* are vectors containing a sequence of items with no duplicate values.

R also supports control structures like *if*, *while* and *for* which is widely adopted in other programming languages to implement some business logic.

We use spine method in package *vcd* to plot the Figure 9. Plotweb method in package *diagram* is used to plot the transition matrix. *cspade* method in *arulesSequences* package is used to mining the sequential pattern.

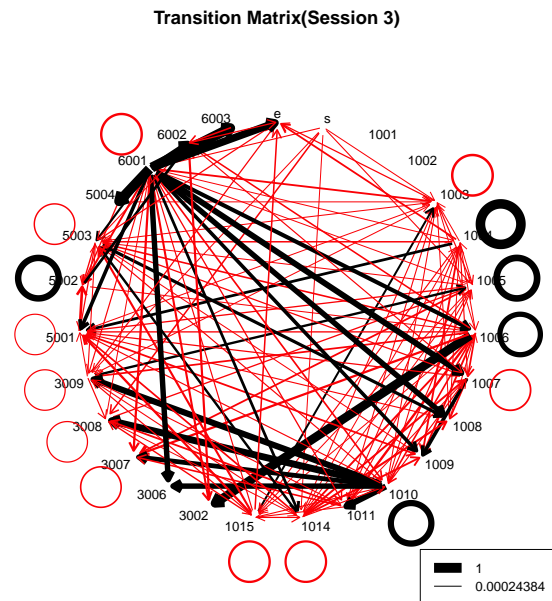


Fig. 19: Transition Matrix Normalized by Column(Session 3)

References

- [1] : Data mining. http://en.wikipedia.org/wiki/Data_mining
- [2] : Data mining survey. <http://www.rexeranalytics.com/Data-Miner-Survey-Results-2010.html>
- [3] Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. *SIGMOD Rec.* **22** (June 1993) 207–216
- [4] Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: *Proceedings of the 20th International Conference on Very Large Data Bases. VLDB '94*, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1994) 487–499
- [5] Srikant, R., Agrawal, R.: Mining sequential patterns: Generalizations and performance improvements. In: *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology. EDBT '96*, London, UK, Springer-Verlag (1996) 3–17
- [6] Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.C.: Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In: *Proceedings of the 17th International Conference on Data Engineering*, Washington, DC, USA, IEEE Computer Society (2001) 215–226
- [7] Liu, B.: *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data*. Springer (2011)
- [8] Cherkassky, V., Mulier, F.: *Learning from Data - Concepts, Theory, and Methods*. Wiley-Interscience, Hoboken, NJ, USA (2007)

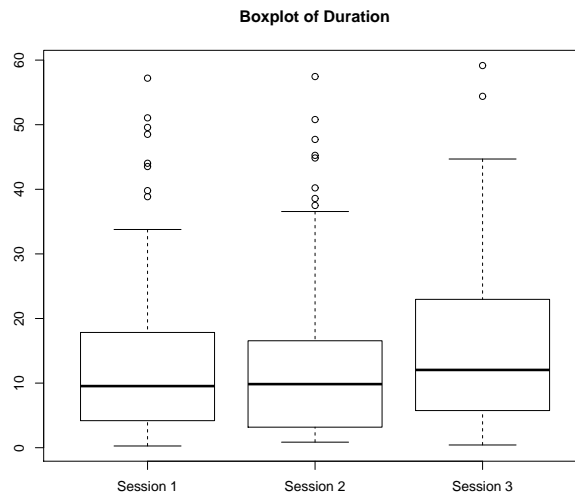


Fig. 20: The Boxplot of Duration for 3 Sessions

- [9] Fernando, P.M., Quintana, O.A.: Nonparametric bayesian data analysis. *Statistical Science* **19** (2004) 95–110
- [10] Hastie, T., Tibshirani, R., Friedman, J.H.: *The Elements of Statistical Learning*, Second Edition. Springer, New York, NY, USA (2009)
- [11] Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification* (2nd Edition). Wiley-Interscience, New York, NY, USA (2000)
- [12] Bishop, C.M.: *Pattern Recognition and Machine Learning*. Springer, New York, NY, USA (2006)
- [13] Christopher D. Manning, P.R., Schütze, H.: *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA (2008)
- [14] Xu, R., Wunsch, D.: *Clustering*. Wiley-IEEE Press (2008)
- [15] Forgy, E.W.: Cluster analysis of multivariate data: efficiency vs interpretability of classifications. *Biometrics* **21** (1965) 768–780
- [16] Willett, P.: Recent trends in hierarchic document clustering: a critical review. *Journal of Information Processing and Management* **24**(5) (1988) 577–597
- [17] Hastie, T., Tibshirani, R., Friedman, J.H.: *The Elements of Statistical Learning*, Second Edition. Springer, New York, NY, USA (2009)
- [18] Xiaojin Zhu, Andrew B Goldberg, R.B., Dietterich, T.: *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers (2009)
- [19] Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*. MIT Press (1998)
- [20] de Baker, R.S.J., Gowda, S.M., Corbett, A.T.: Towards predicting future transfer of learning. [30] 23–30
- [21] Baschera, G.M., Busetto, A.G., Klingler, S., Buhmann, J.M., Gross, M.H.: Modeling engagement dynamics in spelling learning. [30] 31–38

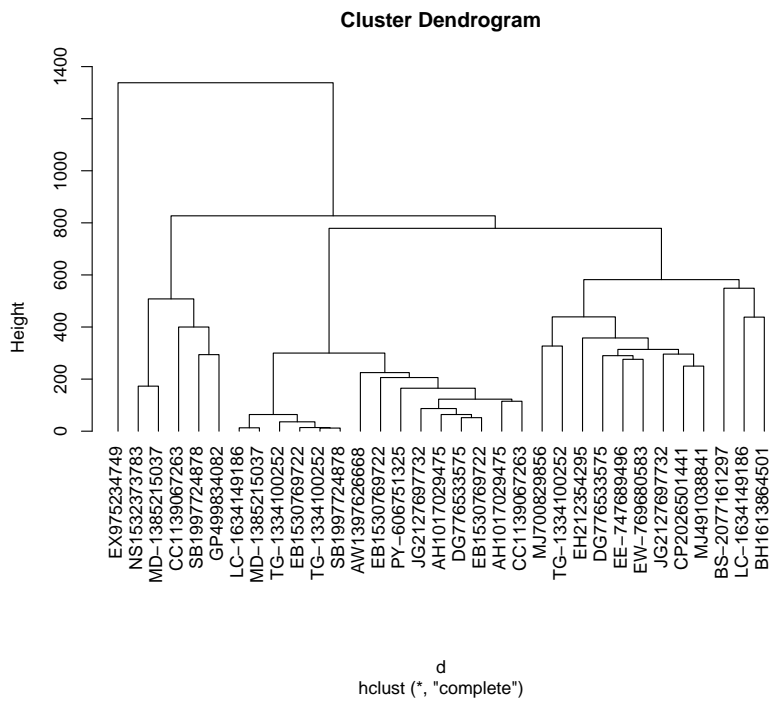


Fig. 21: The Clustering of Session 3

- [22] Hastings, P., Hughes, S., Magliano, J., Goldman, S., Lawless, K.: Text categorization for assessing multiple documents integration, or john henry visits a data mine. [30] 115–122
- [23] Qiao, Q., Beling, P.A.: Classroom video assessment and retrieval via multiple instance learning. [30] 272–279
- [24] Shores, L.R., Rowe, J.P., Lester, J.C.: Early prediction of cognitive tool use in narrative-centered learning environments. [30] 320–327
- [25] Bader-Natal, A., Lotze, T., Furr, D.: A comparison of the effects of nine activities within a self-directed learning environment on skill-grained learning. [30] 15–22
- [26] Goldin, I.M., Ashley, K.D.: Peering inside peer review with bayesian models. [30] 90–97
- [27] Forbes-Riley, K., Litman, D.J.: When does disengagement correlate with learning in spoken dialog computer tutoring? [30] 81–89
- [28] Grafsgaard, J.F., Boyer, K.E., Phillips, R., Lester, J.C.: Modeling confusion: Facial expression, task, and discourse in task-oriented tutorial dialogue. [30] 98–105
- [29] Zaki, M.J.: Spade: An efficient algorithm for mining frequent sequences. *Mach. Learn.* **42** (January 2001) 31–60
- [30] Biswas, G., Bull, S., Kay, J., Mitrovic, A., eds.: *Artificial Intelligence in Education - 15th International Conference, AIED 2011, Auckland, New Zealand, June 28 - July 2011*. In Biswas, G., Bull, S., Kay, J., Mitrovic, A., eds.: *AIED*. Volume 6738 of *Lecture Notes in Computer Science.*, Springer (2011)