Report of VE490, Undergraduate Research Program
Data Mining and Machine Learning for Intelligent Tutoring Systems

Shangquan SUN[1], **Paul WENG**[2],

[1] SJTU-Michigan Joint Institute, Shanghai Jiao Tong University

[2] SJTU-Michigan Joint Institute, Shanghai Jiao Tong University

527112517@sjtu.edu.cn, paul.weng@sjtu.edu.cn

Abstract

Based on a series of studies of human psychology and human learning, various kinds of intelligent tutoring systems (ITS) have been developed to help users learn and strengthen users' memory. After being exposed to information for multiple times with intervals of time since last exposure, human will be able to capture the information efficiently, which has been proved and named as the concept of spaced repetition (SR). Plenty of flashcard apps, one simple form of ITS, have been established according to this concept. Among them, Mnemosyne is one of the most prevalent and well-organized ones and a log data was made available from its official website. Current flashcard apps just incorporate a group of fixed schedules to organize those intervals before reviewing a previously learnt object, which, however, is quite unreasonable and inefficient because different individuals will have diverse ability of learning and learning characteristics. Thanks largely to rapid improvement of machine learning, we can adapt the schedule to each individual by training a machine learning model, which will potentially enhance the efficiency of a flashcard app.

1 Introduction

Nowadays a large number of people tend to use a flashcard application to learn something new. There are a great deal of flashcard applications, such as Anki [1], Mnemosyne [2], or duolinguo [3]. Figure 1 shows two screenshots of the interface of Mnemosyne [2]. For example, one Chinese user is learning Toefl. When he or she meets an object "enzyme", he or she can firstly try to recall the answer or Chinese interpretation of the object and then click "Show answer" to check whether his or her memory is correct or not. The next step is vital, which is to respond with a grade in order to inform the app of how well he or she can memorize the object. Typically, there are six grades varying from 0 to 5. The higher the responded grade is, the better the user can master the new knowledge. By knowing this response, the app will organize the interval before the next review so that the user's memory can be reinforced.

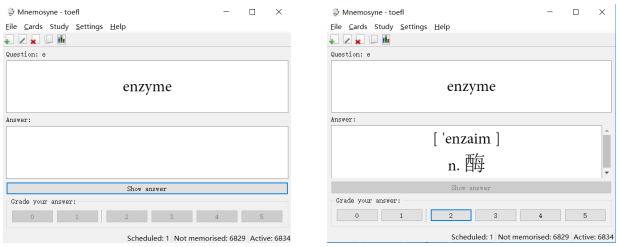


Figure 1: Two screenshots of the user interface of one flashcard app, Mnemosyne. After one user meets an item, a button, "Show answer" should be clicked. After clicking "Show answer", the user should choose a response ranging from 0 to 5 to evaluate how well he or she can recall the item. [2]

However, current flashcard applications utilize a system that generates a series of fixed scheduled intervals before subsequent reviews. As a result, some learners complain about such an awkward system, since each learner will have his or her own learning ability and characteristic. Therefore, to develop a system that is adaptable to every user and the contents to learn is extremely important.

1.1 Description of Mnemosyne [2]

In the Mnemosyne app, a user can respond to an item by inputting an evaluation of how well he or she could recall the item. The evaluation is an integer between 0 to 5. According to the definition, a response of 0 or 1 means the user has a

difficulty in memorizing the item, while a value between 2 to 5 means the user can recall the item. The system will organize the order of each item appearing on screen based on a non-adaptable algorithm. Hence, the interval since last repetition is selected from a group of fixed numbers, for instance, more than twelve hours, over one whole day and so on. Fortunately, a log data is available and contains fifteen features for each event or each operation [5]. The data included in the project contains nearly 120,000,000 events.

1.2 Related Work

The very first study about the relation between human learning and the times of exposures could be traced back to the work of Ebbinghaus in 1913 [7], which shows that whether one could capture a knowledge depends on two factors, one is reinforcement, multiple times of reviewing one object, and delay, interval before a next review. Later Spitzer established an experiment and justified a technique of tuning the two factors to effectively improve students' memorization [10]. The technique was called as Spaced Repetition (SR).

Then the Rasch Model was developed and illustrated that various results and efficiencies of cognitive process including memorization are related to examinees' abilities as well as the difficulty of object [8]. In the Rasch Model, the probability of user's responses will be predicted as a logistic function of one pair of person and object parameters. A formulaic expression of the Rasch Model can be written as $P[recall] = \phi(\theta - \beta)$, where θ is user ability and β is item difficulty. Based on the general idea of the Rasch Model, we could preliminarily anticipate that individuals have diverse capabilities of learning and memorizing and thus it is necessary to generate an adaptive model for each learner or flashcard user.

What's more, a group of researchers from Cornel University have compared the performances of multiple mathematical models, including the Rasch Model [8], that predict users' responses in the Mnemosyne app [9]. The result turns out to be that the Rasch Model performs the best.

1.3 Our Contributions

In this work, multiple classical machine learning algorithms have been utilized. Classical algorithms include Random Forest, Decision Tree, Logistic Regression, SVM (Support Vector Machine) and KNN (k-Nearest Neighbor) [6].

Our first work is to apply the machine learning algorithms to prediction of users' responses so that we can know the likelihood of users' recall based on given information of previous events. By trying multiple intervals, we can acquire a value of appropriate timing to show one object next time. Hence, the app can arrange the order of showing series of cards to the users more intelligently. As a result, a better designed flashcard app can be developed and adapted to different individual users.

Also, a condition exists that there will be insufficient amount of samples for some new users to train a model of predicting appropriate intervals. for those users who do not have adequate number of operations yet to generate an acceptable prediction model, we develop an algorithm involving unsupervised learning to cluster the user with some other old users with similar learning characteristics and domains, and then use such a group of events to obtain a more accurate prediction model. Hopefully, we expect a group of events from multiple users can lead to a relatively higher accuracy of prediction. The algorithm is based on Rasch cognitive model [8]. We set two groups of vectors of parameters, θ and β, to respectively represent the characteristics of users and objects, so that for each user i and item j, we have two vectors of parameters, θ_i and β_j, to depict them respectively. As a result, the new users' data could be combined with those from similar users to generate a better model.

2 Data Cleaning and Processing

In general, there are four steps, Data cleaning, Data processing, Training, and Validation, which are illustrated in Figure 2. Since there are many invalid or missing data in the log file, we need to firstly clean them off. After that, the left valid data must be processed so that they can be utilized for training and validation. After training models, we should test the performance of the models.

2.1 Data Source and Data Cleaning

The log data for the project is available on Sourseforge [5] as a database file. In the file, there are about 120,000,000 events included and 15 features. They are identity of user, event, timestamp, identity of object, grade, easiness of object, repetition times of acquisition, repetition times of review, times of forget, repetition times of acquisition since last forget, repetition times of review since last forget, scheduled interval, actual interval, thinking time and timestamp of next repetition, which can be seen in Table 1.

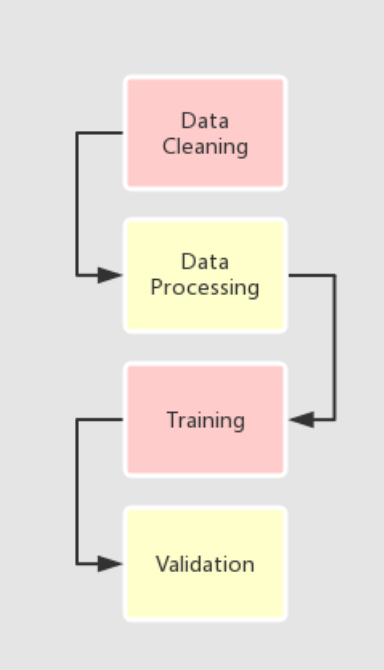


Figure 2: Flowchart of the project.

Variable	Description
user_id	autonomous user ID
event	9 for repetition
timestamp	timestamp
object_id	card ID
grade	grade
easiness	item difficulty factor
acq_reps	repetition times of acquisition
ret_reps	repetition times of review
lapses	times of forget
acq_reps_since_lapse	repetition times of acquisition since last forget
ret_reps_since_lapse	repetition times of review since last forget
scheduled_interval	schedualed review timestamp interval
actual_interval	actual review timestamp inerval
thinking_time	time of thinking the answer
next_rep	timestamp of next repetition

Table 1: The description of 15 variables included in the log file of Mnemosyne. [5]

However, most samples are invalid. Some of them contain missing data and some of them are results of invalid or supervisors' operations. Only those samples with $event = 9$ are from valid operations of users. Therefore, we filter all the samples with missing data and invalid operations.

After this step, about 49,000,000 samples are left. Also, there are 10,737 users involved in the data. Among them, more than 900 users have used the application for more than 10,000 events, which is quite abundant for us to train models.

2.2 Dimension Adjustment

In this project, there are fifteen features for one event, while four of them are unhelpful to us to train our models. They are user ID, event, object ID and scheduled interval. Note that scheduled interval is generated by the algorithm of the current flashcard app. it is exactly what we want to improve and thus cannot be used in prediction. In addition, object ID is just a text to identify each card and thus useless in this part. However, user ID and event are used to select data, which has been discussed in Section 2.1. After selecting desired data samples, we remove these four objects and then obtain ten features, such as timestamp, number of reviews, number of repetition and so on, and grade as the label.

At the same time, we anticipate that combining one event with another one or two previous events with the same user and item as one sample, so that one sample can incorporate more sorts of information, which can possibly increase the performance of model.

As a result, we adapt data into thirty dimensions by using the two previous events for a given item.

2.3 Normalization and Standardization

Normalization and Standardization are two main ways of mapping data within a large range into data between a narrow bandwidth. Two equations are listed as followed.

$$\begin{cases} DataNormalized = \frac{data - data.mean}{data.standard_deviation}, \\ DataStandardized = \frac{data - data.min}{data.max - data.min}. \end{cases} \quad (1)$$

In this project, we mainly use Standardization so that a distribution of features could be maintained and data could be mapped into a range between 0 to 1.

2.4 Label Classification

According to definition, a response of 0 or 1 means that the user has a difficulty in memorizing the item, while one

between 2 to 5 means the user can recall the item. Therefore, for binary-class prediction model, we represent grade between 2 to 5 as Class 1, while grade of 0 or 1 as Class 0. For multi-class prediction, we just set the values of grade as labels.

2.5 F_2 Score

F_2 Score is to calculate the goodness of the Binary Classification Model. Compared to F_1 Score, F_2 Score values more on Recall than Precision, since, in our case, we pay more attention on the case where actual class is Class 1.

In Table 2 and Equation 2, we can notice that Precision is the likelihood that how many predicted relevant results are correct among all the positive predictions, while Recall is the fraction of relevant and correct predictions over the total amount of actual positive instances.

	Actual Positive	Actual Negative
Positive Prediction	True Positive	False Positive
Negative Prediction	False Negative	True Negative

Table 2: Illustration of Precision and Recall. [12]

$$\begin{cases} Precision = \dfrac{true\ positive}{true\ positive + false\ positive}, \\ Recall = \dfrac{true\ positive}{true\ positive + false\ negative}. \end{cases} \quad (2)$$

$$F_m\ Score = (1 + m^2) \times \frac{precision \times recall}{m^2 \times precision + recall}. \quad (3)$$

In this project, we value more on Recall than Precision and thus use $F_2\ Score$.

2.6 Cross Validation

After fetching data, the next step is to separate samples into training set and test set. In training and testing, we choose 5-fold validation system, which means to train models with four-fifth of all data and then test the models with the rest of data. This procedure will repeat for five runs. The final results of accuracy will be the mean value of the five accuracies from the five runs.

However, if we just randomly divide the data set, there will be a case that some events in the future are included in the training set, while some events in the past are in the test set. It is illogical to predict bygone events by future events.

So to prevent this, we should firstly sort all the samples

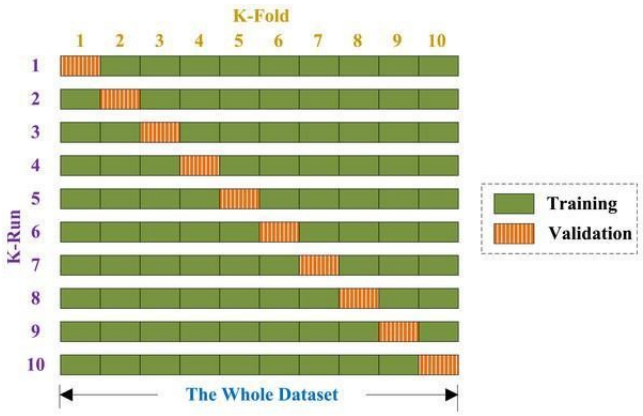


Figure 3: K-Fold Cross Validation. [4]

in chronological order and then select some past events into the training set and future events into the test set. In practical experiment, we found the method demonstrated in Figure 4 could result in a smallest variance of resulting accuracies.

Finally, we shuffle training set and test set separately so

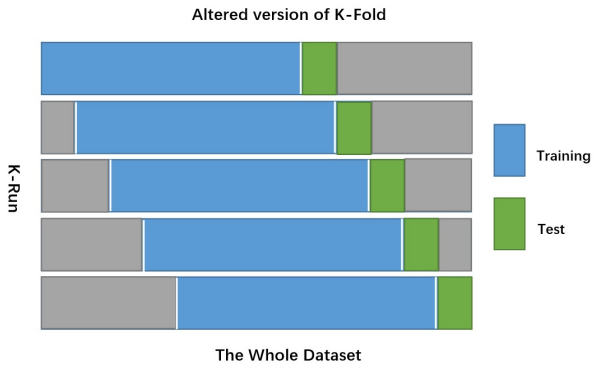


Figure 4: Altered version of K-fold, where K=5.

that all the samples could be in a random order. Typically, we select twelve users who have used the application for more than 10,000 valid operations. After doing all the steps of data cleaning and processing, we could make use of these data to train our models.

3 Experimental Results

3.1 Binary-class Prediction

In binary-class prediction part, we have tried four machine learning algorithms. They are in order Decision Tree, KNN (k-Nearest Neighbor), Logistic Regression and SVM (Support Vector Machine) [6].

Tuning parameters

Generally, we utilize sklearn, a specific module in Python containing almost all machine learning functions. And the parameters of the functions we tune in this part are the most prevalent ones.

For the Decision Tree algorithm, we have tried both "gini" and "entropy" for "criterion", as well as both "best" and "random" for "splitter". After validation, the results turn out that setting "criterion=entropy" and "splitter=best" is a better choice.

For KNN (k-Nearest Neighbor), we set "algorithm" as "auto", since the function will automatically determine the most appropriate one based on the data. Then we try both "uniform" and "distance" for "weights". Results show that the default one, "uniform", is better.

For Logistic Regression, we could choose two norms, $l1$ and $l2$. After validation, it is found that $l2$ could lead to a higher accuracy.

For SVM (Support Vector Machine), we have tested "probability" with options of "True" and "False". The results turn out that setting "probability" to be "False" is better. For "gamma", we just set it as "auto", so that the function will automatically decide it.

Result Analysis

After determining the parameters, we again use those data to train machine learning models and test the performances of the models by cross validation. A plot of bar chart of the average accuracies is shown as Figure 5. In addition, the standard deviation of accuracies over 5 runs of cross validation could be calculated and the results can be demonstrated in Figure 5.

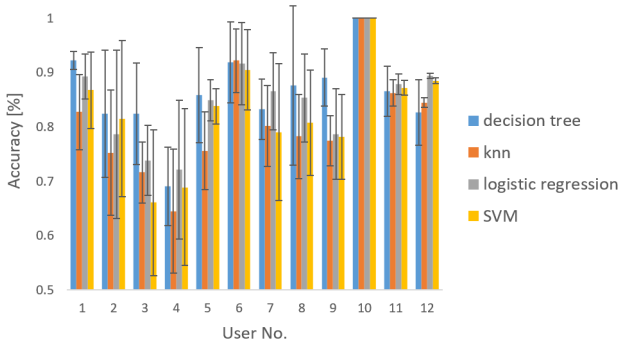


Figure 5: Bar chart of accuracy versus user for four binary-class algorithms, Decision Tree, KNN, Logistic Regression and SVM.

Also, we can obtain an average F_2 $Score$ for each user and each model. Hence, another line chart of F_2 $Score$ versus user for the four algorithms can be plotted as Figure 6.

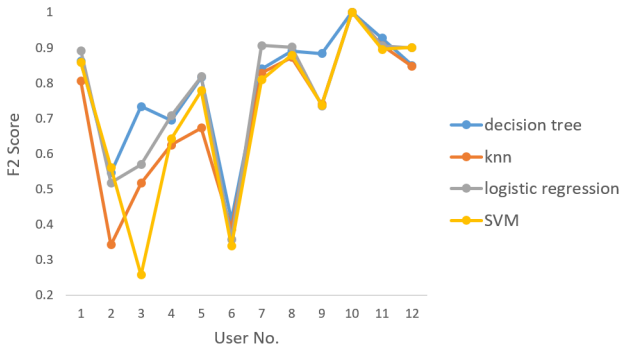


Figure 6: Line chart of F_2 $Score$ versus user for four binary-class algorithms, Decision Tree, KNN, Logistic Regression and SVM.

From Figure 6 and Figure 5, we can make five observations:

1. Since the data come from operations of users, there are a large difference of performances among users even for identical algorithms. For example, User No.10 responded to the flashcard application always with high grades, which means that he or she had already mastered all the knowledge or he or she just blindly responded positively all the time. Conversely, User No.4 and User No.3 did not follow a predictive way when responding. Therefore, all the four models perform badly for the two users.
2. We could find that generally Figure 5 and Figure 6 have a quite similar trend and shape except the case of User No.6. The accuracies for this user are very high while the F_2 $Scores$ are extremely low. The reason may be a large precision and a relatively small recall, which shows an uncertainty of the models.
3. Those models with high accuracies tend to have a smaller standard deviation. This means that for some users, the models will have extremely great results with not only a high accuracy, a high F_2 $Score$, but also a low standard deviation of accuracies. And there are no significant diversities among the cases of four algorithms.
4. Generally speaking, Decision Tree performs the best, almost for all users, the performances and F_2 $Scores$ are the highest. And Logistic Regression has the second best

performance. However, there are no considerable differences between SVM and KNN.

5. In general, for every user, we could generate a model whose accuracy could be higher than 70%. For most users, we could give a model with 80% accuracy. Moreover, Decision Tree model could lead to a model with 90% for four users out of twelve users. This is an acceptable result, especially as they could be improved with an ensemble techniques.

3.2 Multi-class Prediction

Again, we choose the previous twelve users who have used the application for more than 10,000 valid operations. After doing all the steps of data cleaning and processing, we use these data to train our models. In the multi-class prediction part, in addition to the previous four machine learning algorithms, we also try Random Forest which is a complicated version of Decision Tree [6].

Tuning parameters

We still utilize sklearn to train our models.

For Decision Tree algorithm, we also try both "gini" and "entropy" for "criterion", and both "best" and "random" for "splitter". After validation, the results turn out that setting "criterion=entropy" and "splitter=best" is the best choice.

For KNN (k-Nearest Neighbor), we set "algorithm" as "auto", since the function will automatically determine the most appropriate one based on the data. Then we try both "uniform" and "distance" for "weights". Results show that the default one, "uniform", is better.

For Logistic Regression, we could choose two norms, $l1$ and $l2$. After validation, it is found that $l1$ could lead to a higher accuracy, since it could generate more sparse solutions.

For SVM (Support Vector Machine), we have tested both LinearSVC and SVC. It turns out that the former one is better. Then we try both a default value and zero for $random\_state$. The result is $random\_state = 0$ is better.

For Random Forest, we have tested criterion, "n_estimators", "oob_score", "max_depth" and "max_features". As a result, the parameters with "n_estimators = 75", "oob_score = True", "criterion = 'entropy'", "max_depth = None" and "max_features = None" performs better.

Result Analysis

After determining the parameters, we again use those data to train machine learning models and test the performances of the models by cross validation. A plot of bar chart of the average accuracies is shown as Figure 7. In addition, the standard deviation of accuracies over 5 runs of cross validation could be calculated and the results can be demonstrated in Figure 7.

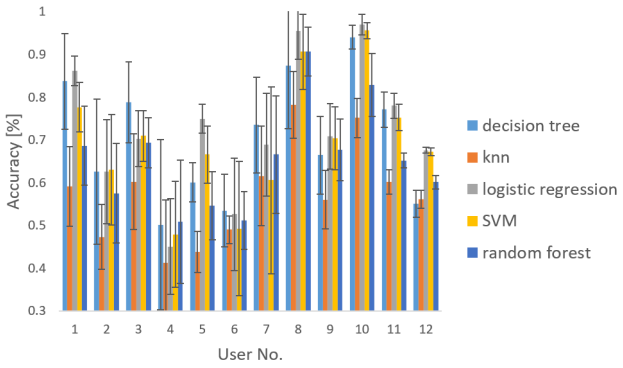


Figure 7: Bar chart of accuracy versus user for five multi-class algorithms, Decision Tree, KNN, Logistic Regression, SVM and Random Forest.

From Figure 7, we can make four observations:

1. There are large variances for one group of accuracies of one model for different users. For example, the accuracies of User No.10 and User No.8 could be about 95%, while those of User No.6 and User No.4 could be only nearly 50%. This perhaps is the result of users' various responses. Some may be easily predictable and others may not.

2. The bar chart of accuracies versus user for all five models have similar characteristics and shapes, which means that every user's samples have a degree of difficulty in using them to predict an appropriate model for all the algorithms. So for some users, all the models could not generate acceptable accuracies, while for other users, all those models have excellent performances.

3. The models of Logistic Regression, Linear SVM, and Decision Trees have the highest accuracies almost for every user. The differences among them are very slight and we couldn't tell which one is the best. Then the Random Forest model is slightly secondary to them. The

model of KNN has the worst performance. However, some results of Linear SVM and Decision Tree have extremely large standard deviation, which means the performances of the two models are not very stable.

4. We can find that in general those cases with high average accuracies tend to have small variances of accuracies. For example, for users from User No.1 to User No.6, the tendency is obvious that when accuracies of all five models are relatively high, their standard deviations will be comparatively small. This shows that those models with better performances will also be quite stable.

4 Clustering Algorithm

4.1 Generation of person and item Parameters.

In this part, we develop an algorithm that can generate a parameter vector adaptable to each user and each item. The algorithm is slightly similar to Logistic Regression and based on Rasch cognitive model [8]. Firstly, we denote the sigmoid function as

$$\Phi(x) = \frac{1}{1+e^{-x}}$$

.

Then suppose we have a group of users and objects that the users have encountered. There are m users and n objects involved. Therefore, for one specific user, User No.i, and a specific object, Object No.j, there will be a series of samples indicating all events of the user meeting the object, which are shown as $X_{i,j}$ in Figure 8. Then we define θ_i of dimension $N = 11$ to be a feature vector for user No.i and β_j of dimension $N = 11$ to be a feature vector for object No.j. To set $N = 11$ is because there are 11 dimensions for one sample.

Since the function value of a sigmoid function can be interpreted as the likelihood or mathematical expectation of a label Y_i given X_i: $\mathbb{E}[Y_i|X_i] = p_i = \Phi((\theta_i - \beta_j)X_i)$ [11]. Then we can derive a probability distribution as $P(Y_i = Y|X_i) = p_i^Y(1-p_i)^{(1-Y)}$. Based on this, we establish an objective function as Equation 4, where $\eta_{k,i,j}$ is the grade of the k-th event among the events of User No.i and Object No.j. And we could obtain Equation 5 which is an intermediate step of deriving a cost function as shown in Equation 6.

$$\max_{\theta,\beta} \prod \Big[\Phi((\theta_i - \beta_j)X)^{Y\{\eta_{k,i,j}\geq 2\}} * (1-\Phi((\theta_i-\beta_j)X))^{Y\{\eta_{k,i,j}<2\}}\Big] \tag{4}$$

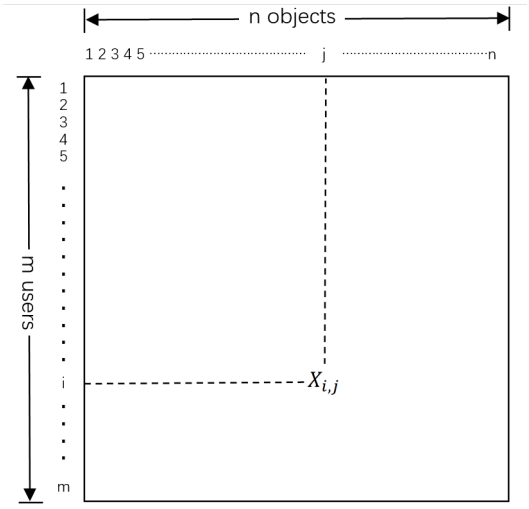


Figure 8: Sample matrix with m users and n objects. Each conjuncture point of User No.i and Object No.j, $X_{i,j}$, means all events such that the user responds to the object in chronological order.

$$L_{\theta_i,\beta_j}(X,Y) = \prod(\Phi((\theta_i-\beta_j)X_{i,j}))^{Y_{i,j}}(1-\Phi((\theta_i-\beta_j)X_{i,j}))^{1-Y_{i,j}} \tag{5}$$

$$F(\theta,\beta) = \log L_{\theta_i,\beta_j}(X,Y) - \lambda(\|\theta\|_2^2 + \|\beta\|_2^2) \tag{6}$$

Then what we do is to apply gradient descent to the cost function, $F(\theta,\beta)$ until , which is illustrated as Equation 7. The gradient descent is feasible because the function $F(\theta,\beta)$ is convergent. This can be derived from the fact that the norm of the function is two quadratic functions and the other term is also a convex function.

$$\begin{cases} \theta_i^{(t)} \leftarrow \theta_i^{(t-1)} + \alpha\dfrac{\partial F(\theta,\beta)}{\partial\theta_i} \\ \beta_j^{(t)} \leftarrow \beta_j^{(t-1)} + \alpha\dfrac{\partial F(\theta,\beta)}{\partial\beta_j} \end{cases} \quad i \leq m,\ j \leq n \tag{7}$$

After getting the matrix of $\theta \in \mathcal{R}^{12\times 11}$ containing feature parameters for the 12 users, the next step is to apply K-Means algorithm to the matrix. Since the parameters could reflect each user's learning characteristic, they could be used to determine similarity among the group of users and thus to divide them into several subgroups.

4.2 Results of Clustering

We select 12 users and 95 objects, namely $m = 12$ and $n = 95$. After running the previous algorithm, we obtain

parameter matrix, θ and apply it to the K-Means algorithm to separate the 12 users into 5 groups. Choosing the value of 5 is because it is more likely to divide them into the subgroups of 2 to 3 similar users. The result of cluster is shown in Figure 9.

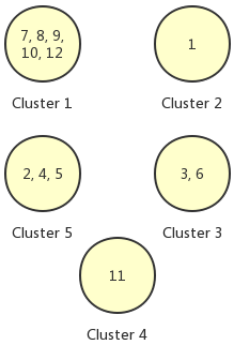
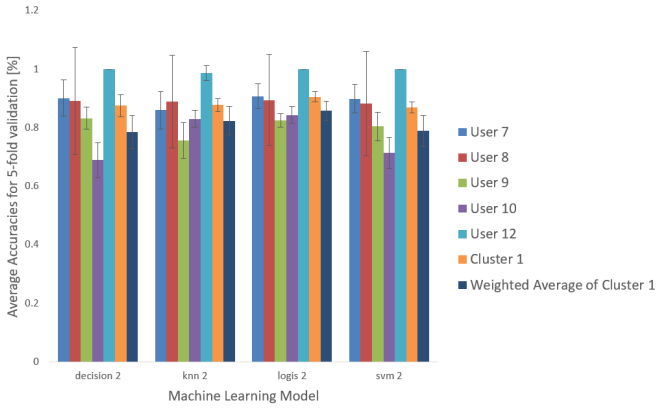


Figure 9: Result of clustering 12 users into 5 groups by the clustering algorithm.

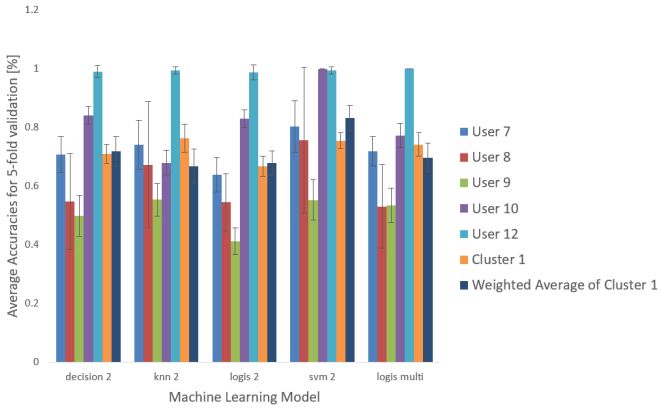
Since Clusters No.2 and No.4 only contains one user, we will only focus on testing other clusters. By inputting a combination of all samples in one cluster into previous codes to train prediction models, we could obtain the accuracy of models derived from the cluster and thus test whether the result of a cluster is proper or not. Specifically, we plot Figure 10, Figure 11 and Figure 12 for Cluster No.1, No.3 and No.5.

Based on Figure 10, Figure 11 and Figure 12, we could have several observations:

1. For Cluster No.3 and Cluster No.5, since one of the users in the clusters have a considerably large number of events, the result of the combination of the clusters will be quite close to the performance of the user with large samples. Despite this, the accuracies are very high which means the result for the clusters is acceptable.
2. For Cluster No.1, we could find that although there are relatively large differences among users in Cluster No.1, the results of the combination of the cluster are generally average values of all the users in the cluster.
3. Then the performance of Cluster No.1 and the weighted average values of the performances of all the five users in Cluster No.1 can be compared. In Figure 10.(a), we



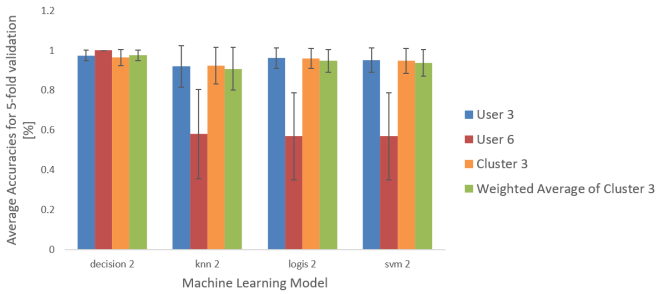
(a) The figure on the top is showing the cases of binary-class prediction.



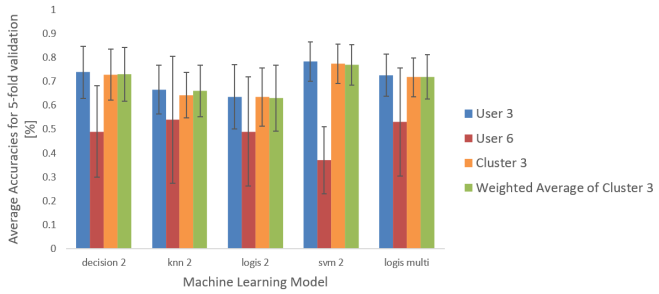
(b) The figure on the bottom indicates the cases of multi-class prediction.

Figure 10: Comparisons of the performances of the nine models given the input data of User No.7, No.8, No.9, No,10 and No.12, as well as the input data of Cluster No.1. In addition, the weighted average values of five users' results is included for comparison.

can clearly see that each binary-class prediction model for Cluster No.1 generates a higher accuracy than the average accuracy of those accuracies from the five users. In addition, The standard deviations of the accuracies of binary-class model for Cluster No.1 are always relatively smaller. These are strong evidences to show that the result of clustering the 12 users is reasonable. In Figure 10.(b), we can find that each multi-class prediction model for Cluster No.1 generates a very close accuracy to the average accuracy of those accuracies from the five users. In addition, The standard deviations of the accuracies of multi-class model for Cluster No.1 are also quite close to those of the average values of the results of the five users. These help to show that the result of cluster is



(a) The figure on the top is showing the cases of binary-class prediction.



(b) The figure on the bottom indicates the cases of multi-class prediction.

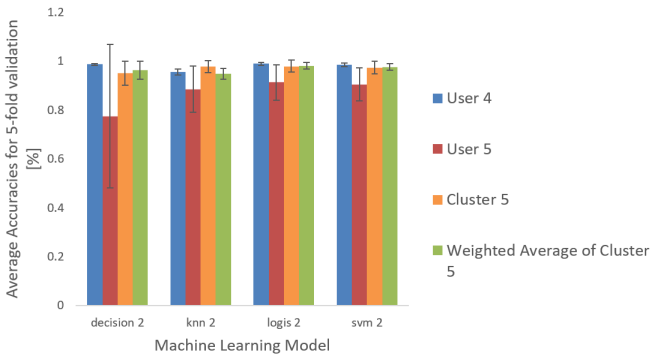
Figure 11: Comparisons of the performances of the nine models given the input data of User No.3 and No.6, as well as the input data of Cluster No.3. In addition, the weighted average values of two users' results is included for comparison.

acceptable.

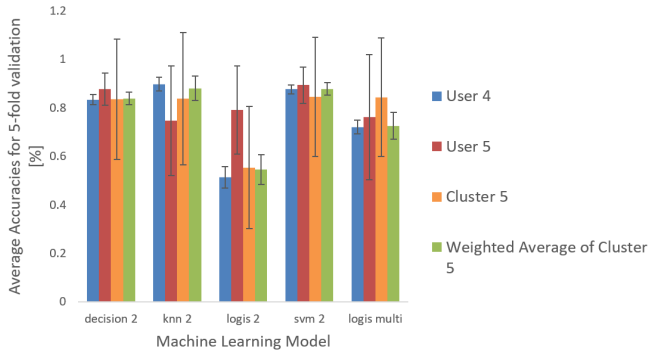
4. From the observation of Figure 11, there is an optimistic phenomenon that the performances of all the nine models for Cluster No.3 are extremely close to the weighted average accuracies of the 2 users in Cluster No.3. Besides, the standard deviations of both two cases are also very approximate to each other for every model. This is also a helpful proof of the expectation that the cluster algorithm works well.

5. From Figure 12, we can find that although the results of Cluster No.5 and the average accuracies of the users in Cluster No.5 are close to one another for each model, the accuracies of five multi-class prediction models for Cluster No.5 always have large standard deviations, which means the performances become more unstable.

Although a combination of samples from similar users cannot lead to a considerable improvement of performances of models, it could help us to train a model for prediction for those users with a small number of events and the result of



(a) The figure on the top is showing the cases of binary-class prediction



(b) The figure on the bottom indicates the cases of multi-class prediction.

Figure 12: Comparisons of the performances of the nine models given the input data of User No.2 and No.4, as well as the input data of Cluster No.5. In addition, the weighted average values of two users' results is included for comparison. Since most responses from User No.2 are positive, sklearn cannot train model with this and thus plot of the user is unavailable.

the clustering algorithm is generally acceptable

5 Conclusion and Future Work

5.1 Conclusion

Initially, there are two objectives in this project. The first one is to predict how well one user will respond to an item by a model trained with the information of previous events of the user. When predicting this, we have tried both binary-class prediction and multi-class one. The other goal is to predict an appropriate interval since last review, which is the most effective timing for next review. The second one is much more difficult than previous one, because of a large range of interval and also limited number of features. An interval could be distributed from tens to millions, which will cause the result

of prediction to possess a considerable error. After trying this, we found a very low accuracy. Therefore, we just focus on the first objective in the project. And in practical, we could use a model of predicting users' responses to predict appropriate intervals by applying various proposed events with intervals in the models. Therefore, the second difficult goal could also be accomplished by the solution to the first objective.

After experiments of training and testing models, we can successfully obtain classical machine learning models that are able to predict users' responses with an input of the target event. The best performances of the five models can vary from 65% to 95%, which is an acceptable result.

The second section in this project is to find some similar users for a new user so that the prediction models can be trained with the samples from a union of the similar users, since a new user does not have sufficient number of operations in Mnemosyne. Based on the results of applying the prediction models to a union of all the samples of the users in one cluster, we can find the result of cluster is generally reasonable and namely, the cluster algorithm derived by us can work basically well.

5.2 Future Work

In this project, we just try a part of all machine learning models and we only tune a small part of parameters for each model. In a future work, more machine learning models can be applied and in addition more parameters can be tried more carefully. What's more, a deep learning model, such as back propagation neural network [6], can be designed to have a better performance of prediction.

For the cluster algorithm, we have just tried a small set of users and objects when testifying the algorithm due to a limited capability of our computers. However, if we can test the results of clustering a larger group of users with multiple prediction models, a more comprehensive conclusion can be drawn. There may also be some improvement of the mathematical procedure in the algorithm. For example, we can try the Spectral Clustering algorithm instead of the K-Means [6].

References

[1] https://apps.ankiweb.net/, title = anki.

[2] https://mnemosyne-proj.org/, title = mnemosyne.

[3] http://www.duolingo.cn/, title = duolinguo.

[4] 使用光敏電阻和樾器宛習來判斷會議室狀慴-2. https://chtseng.wordpress.com/2016/09/07/使用光敏電阻和機器學習來判斷會議室狀態-2/. Accessed May 10, 2018.

[5] Mnemosyne project. http://sourceforge.net/projects/mnemosyne-proj/?source=typ_redirect. Accessed May 10, 2018.

[6] BISHOP, C. M. Pattern recognition and machine learning.

[7] EBBINGHAUS, H. Memory: A contribution to experimental psychology. *Annals of neurosciences 20*, 4 (1913), 155.

[8] OSECKỲ, P. Rasch, georg. probabilistic models for some intelligence and attainment tests.

[9] REDDY, S., LABUTOV, I., BANERJEE, S., AND JOACHIMS, T. Unbounded human learning: Optimal scheduling for spaced repetition. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016), ACM, pp. 1815–1824.

[10] SPITZER, H. F. Studies in retention. *Journal of Educational Psychology 30*, 9 (1939), 641.

[11] WIKIPEDIA CONTRIBUTORS. Logistic regression — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Logistic_regression&oldid=841162954, 2018. [Online; accessed 18-May-2018].

[12] WIKIPEDIA CONTRIBUTORS. Precision and recall — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Precision_and_recall&oldid=835396495, 2018. [Online; accessed 17-May-2018].