Final Report

PDSecretary: A Mobile App for Recommending Exercise to Parkinson’s Disease Patients

WAN FANG        Dr. Wu Chuan

3035232638        Supervisor
Abstract

Parkinson’s disease is a chronic neurodegenerative disease that affects the central nervous system. In 2015, approximately 6.2 million people worldwide suffered from Parkinson's disease with about 117,000 deaths. Parkinson's disease usually occurs in elderly people over the age of 60, whereas about 1% of the elderly suffer from the disease. Parkinson's disease is currently incurable, and physicians usually recommend patients with Parkinson's disease to take exercises. There is evidence that under the guidance of a physiotherapist, the patient's motor symptoms, emotional intelligence, daily activities and quality of life can be significantly improved compared to self-training at home.

This project aims to implement an easy-to-use Android app, PDSecretary for personalized recommending exercise for Parkinson’s disease patients and make their lives more convenient. There are various functionalities planned to be implemented, such as reminding medicine taking and connecting to a wearable device to gather more activity data for a better recommendation.

The whole system, including the recommendation algorithm, backend server as well as database, and the front-end Android application have been implemented throughout the year. The first focus of this paper is on the recommendation system, which should be medically-recommended, personalized and interactive. There is performance analysis after the recommendation algorithms are developed. Also, the Android application written by Android Studio and database design will be introduced. The next step of the project is to refine the UI and UX of the client application and conduct a performance test among potential users. This final report presents the background, the approach, the result, the problems encountered, and the future works of the project.
Acknowledgement

I want to present my greatest appreciation and gratitude to Dr. Wu Chuan for supervising and helping me with the entire project. This is the first project of practical computer science I have ever participated in, and Dr. Wu has been offering support and guidance in every aspect throughout. Also, I sincerely thank Doctor Gu Mei for her medical advices relate to this project.
Contents

Abstract ............................................................................................................................................. 2
Acknowledgement ............................................................................................................................ 3
List of Figures ...................................................................................................................................... 6
Abbreviation ....................................................................................................................................... 7
1. Introduction ...................................................................................................................................... 8
   1.1 Background & Current Status ....................................................................................................... 8
   1.2 Problems ..................................................................................................................................... 8
   1.3 Objective ..................................................................................................................................... 10
   1.4 Scope & Outline of report ........................................................................................................... 12
2. Related Work Review ..................................................................................................................... 12
3. Methodology .................................................................................................................................... 15
   3.1 System Architecture Design ....................................................................................................... 15
   3.2 Recommendation Algorithm Design ........................................................................................... 16
      3.2.1 User-based CF & Item-based CF .............................................................................................. 18
      3.2.2 Content-based Filtering ........................................................................................................... 19
      3.2.3 Latent Factor Model ................................................................................................................ 19
      3.2.4 Personalized PageRank ............................................................................................................. 20
      3.2.5 Item2vec ................................................................................................................................ 21
   3.3 Recommendation Algorithm Experiment .................................................................................... 23
      3.3.1 Experiment Set Up & Procedure .............................................................................................. 23
      3.3.2 Experiment Results & Discussion ............................................................................................ 24
      3.3.3 Algorithm Selection ................................................................................................................ 27
   3.4 Medical Adjustment of Recommendation Algorithm ................................................................... 29
   3.5 Dynamic Adjustment of Recommendation Algorithm ................................................................... 33
   3.6 Architecture of Recommendation System ................................................................................... 34
   3.7 UI Design .................................................................................................................................... 35
   3.8 Database Design .......................................................................................................................... 36
4. Final Result ...................................................................................................................................... 39
   4.1 Result of Android App ................................................................................................................ 39
      4.1.1 Login Page & Registration Page .............................................................................................. 39
4.1.2 Main Page .............................................................................................................. 41
4.1.3 Recommended Video Page .................................................................................. 42
4.1.4 Reminder Page .................................................................................................... 43
4.1.5 Account Page ...................................................................................................... 45
4.2 Result of Recommendation System ......................................................................... 46
  4.2.1 Procedure .......................................................................................................... 46
  4.2.2 Performance Analysis ....................................................................................... 47
5. Future Planning & Conclusion .................................................................................... 49
6. References ................................................................................................................ 50
### List of Figures

1. Figure 1. Survey result of the effectiveness of rehabilitation ........................................9
2. Figure 2. Survey result of self-regulation and interests of self-rehabilitate patients ......10
3. Figure 3. Screenshot of Spring Ring Doctor ...............................................................13
4. Figure 4. Screenshot of PD Treasury ........................................................................14
5. Figure 5. System Architecture Design .....................................................................15
6. Figure 6. Formula for cosine similarity and two upgrade versions for item-based CF .....19
7. Figure 7. Formula for cosine similarity and two upgrade versions of user-based CF .....19
8. Figure 8. CBOW and skip-gram visualization .........................................................22
9. Figure 9. Item2vec workflow ..................................................................................23
10. Figure 10. Accuracy for algorithm movies100K ......................................................26
11. Figure 11. Accuracy for algorithm on movies1M .....................................................26
12. Figure 12. Accuracy for hybrid approach on movies100K ........................................28
13. Figure 13. Accuracy for hybrid approach on movies1M ..........................................28
14. Figure 14. Example of questions and score sheet of MDS-UPDRS .........................32
15. Figure 15. Architecture of Recommendation System ..........................................34
16. Figure 16. Item Table of MySQL Database of PDSecretary .....................................37
17. Figure 17. User Table of MySQL Database of PDSecretary ....................................38
18. Figure 18. Click Table of MySQL Database of PDSecretary .....................................38
19. Figure 19. Screenshot of Login Page ......................................................................39
20. Figure 20. Screenshot of Registration Page .............................................................40
21. Figure 21. Screenshot of Main Page .....................................................................41
22. Figure 22. Screenshot of Video List Page ...............................................................42
23. Figure 23. Screenshot of Reminder Pages ..............................................................44
24. Figure 24. Screenshot of Account Pages ...............................................................45
25. Figure 25. Percentage of liked exercise among recommendation ............................47
26. Figure 26. Percentage of liked exercise among top5 recommendation .................48
ABBREVIATIONS

PD: Parkinson Disease
CF: Collaborating Filtering
LFM: Latent Factor Model
PR: PageRank
UPDRS: unified Parkinson's disease rating scale
AWS: Amazon Web Service
1 Introduction

1.1 Background & Current Status
As a degenerative and chronic pathological change of the central nervous system, Parkinson’s disease (PD) may cause motor and cognitive impairments. Its symptoms widely include tremor, postural instability, rigidity, bradykinesia, and changes in memory, language, visuospatial capacity (Souza et al., 2011). Besides the motor and cognitive symptoms of PD, anxiety has a high prevalence among patients and can highly influence daily activities for patients with PD. Approximately, 50% of PD populations have relevant anxiety issues (Rutten et al., 2015).

PD treatment involves three modalities: medical, surgical, and rehabilitation. Among these treatments, rehabilitation has been playing an increasingly important role since neither drugs nor surgery can prevent disease progression. Therefore, continuous treatment is often needed in order to assist cognitive functions. (David et al., 2015; Goodwin, Richards, Taylor, Taylor, & Campbell, 2008; Pompeu et al., 2014). However, rehabilitation is a long-term process and normally unsupervised by professional medical personnel. Therefore, doctors cannot know patients’ situation promptly and patients cannot obtain suitable advice about physical exercise from doctors.

1.2 Problems
Regular physical exercise is beneficial for patients to maintain or improve their athletic ability, flexibility, strength, walking speed and quality of life. Physical exercise can also be combined with physical therapy. However, there is evidence that under the guidance of a physiotherapist, the patient's motor symptoms, emotional intelligence, daily activities, and quality of life can be significantly improved compared to self-training at home. As shown in figure 1, only 30 (44%) of the 68 questionnaires collected indicated that exercise showed a significant improvement in the condition. Of the 18 patients who went to rehabilitation centers or hospitals for rehabilitation, the proportion that
feels improvement was also much higher than those who had self-rehabilitation at home (77% vs 32%).

![Survey result of the effectiveness of rehabilitation](image)

Figure 1. Survey result of the effectiveness of rehabilitation

The above results explain the inefficiency of self-rehabilitation to some extent. This inefficiency is due to a lack of science and timely guidance. Since it is not possible to communicate with experts at any time and place, self-exercise is mostly searched, judged and implemented by the patient. Choosing the right exercise program according to the condition also requires high professionalism and experience. It is difficult to guarantee the quality of the exercise program selected by the patient. In addition, patients often do not plan for exercise in a reasonable, long-lasting, and planned manner based on the amount of daily exercise. This also leads to a lack of consistency in the training schedules arranged by the patients themselves.

In addition, according to the questionnaire, it is always difficult to form a habit by exercising on your own. Patients are less enthusiastic about self-training. As
shown in Figure 2, 76% of patients who had recovered at home indicated that they had experienced an unplanned exercise program. 54% of patients expressed a lack of interest in rehabilitation programs.

![Figure 2. Survey result of self-regulation and interests of self-rehabilitate patients](image)

### 1.3 Objective

According to previous problems, three main objectives should be achieved. First, based on patient needs, a mobile health application (App) is developed, its major function is to recommend exercise to patients. Such a recommendation should be medically recommended. Therefore, those recommendations may replace advice from experts. Second, the recommendation system needs to be personalized. In this way, the recommendation may better meet the interests of users. Third, instead of being static, the recommendation system should be dynamic in terms of making the adjustments based on users’ steps count or exercise accomplished. At the same
time, its design features are close to the daily life of Parkinson's patients, such as taking medication reminders.

1.4 Scope & Outline of Report

In this project, a native Android application is developed instead of iOS or a cross-platform application like React Native. On one hand, according to the survey I conducted, the majority (86.8%) of PD patients use Android phone. This may because all of the participants come from Hong Kong or mainland China. Therefore, I decide to choose Android OS as a target platform. On the other hand, native android has better user experience and response time than a web-based application like React Native. Also, since I already mastered Java, it should be much more efficient to develop native android than learning a new programming language.

PDSecretary has four main functions, including exercise recommending, notification reminders, exercise recording and visualization panel. This app allows patients to first to fill a questionnaire for further medical recommendation. The main interface is a feed and users can receive the recommendation from our model. Also, each recommended item has a detailed description and video as the instruction which can be view in PDSecretary. Exercise and medicine taking can be added to notification reminder through the app and PDSecretary will send notification for specified events. PDSecretary may also automatically send the notification to remind users to do exercise according to the user’s steps count and exercise completed rate. Those steps count and the completed rate will be shown in visualized representation in order to encourage users to exercise more.

The projects has four main contributions. Firstly, it introduces the methodology and design of the recommendation system of PDSecretary. Some mature and reliable recommendation algorithm will be summarized. For these algorithms,
a simple comparison test is performed using an open source dataset and the experimental results are used to determine which algorithm to use. Then, more medical-based and personalized modification will be explained to sketch the whole design of the architecture. Secondly, the rest of design including server, database, front-end application and UI will be introduced. Thirdly, it will present runtime result of the application as an illustration for finalized functionalities and performance of the recommendation system. At last, the results of a conducted usability test will be discussed and further suggestions about future work will be raised base on the test.

The remainder of this paper proceeds as follows. First, to show the significance of this paper, we start with a survey of related work in the current market. Then, presenting the system architecture design, user interface design, database design and algorithm design. Next, a discussion for the result with some testing data for a better understanding of its strength and limitations will be provided. Lastly, the paper closes with our thoughts and a future plan for further improvement.

2 Related Work Review
There are some existing solutions to the problem in the market. After conducting a market survey, I found out that mobile application used by PD patients can be classified into two categories. These two categories can be represented by two typical apps: Spring Ring Doctor (春雨医生), PD Treasury (帕宝袋).
The first type of widely used apps has the same feature and very similar functionalities as Spring Ring Doctor shown in figure 3. This type of apps has rich contents in different forms like video, picture and so on. Some of them even implement relatively advanced recommendation system. Patients can directly contact doctors on the platform and exchange information like a forum. However, such apps do not specifically design for PD. Normally, it is for general health-related purpose so that a larger potential user group can be achieved. Besides, those apps are mostly commercial apps. Therefore, they have to maintain their business model by either selling drugs or medical instruments or running ads, which affect user experience and may cause confusion for PD patients due to their cognitive impairments. Also, the information that relates to rehabilitation is only a very small portion of its content.
The second kind of PD apps are apps developed by different NGOs. The most typical one is PD Treasury shown in figure 4. Due to their non-profit essence, these apps are completely free and have no ads. Furthermore, as PD Treasury which was designed by Hong Kong Parkinson Disease Foundation, such apps normally target at PD or at least target at age-related or mental-related diseases. Although the app has several advantages, the flaws are also obvious. Comparing with the previous type, these apps are too rudimentary and static. Taking PD Treasury as an example, no recommendation system or interactive functionalities implemented. The whole app is more like a bulletin board to show all collected information. This may due to the limited budget for NGOs.
3 Methodology

3.1 System Architecture Design

Figure 5. System Architecture Design

The system is implemented with a 3-tier architecture design (Figure 5). Three-tier architecture typically comprise a client tier, a business or data access tier, and a data tier. The client layer is the front-end Android app, which contains UI part of our application. This layer is used for the design purpose where data is presented to the user and input is taken from the user. For example, recommended exercise videos play in this layer and all user click histories are collected also.

All business logic written in the business layer, like validation of data, calculations, data insertion and so on. This acts as an interface between Client layer and Data Access Layer. This layer is also called the intermediary layer helps to make communication faster between client and data layer. For PD Secretary, this layer composes of a web server and an AWS EC2. The web server this project uses is the HKUCS personal web server (http://i.cs.hku.hk/~username), which assigned to each student. The web server stores all contents to be shown
including video, picture and logs. By using PHP script, the server can behave as a middleware to handle clients request and update database promptly. Since the storage requirement and request handling requirement are not heavy for PDSecretary, HKUCS web server is enough for this task and reduce configuration time comparing with other web server. However, HKUCS web server cannot handle computation of recommendation system module. Amazon EC2 is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2’s simple web service interface allows us to obtain and configure capacity with minimal friction. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing us to quickly scale capacity, both up and down, as computing requirements change. Therefore, we instantiate an EC2 instance and put the recommendation system model on it. By setting up communication between web server and EC2, the business lever can fulfill all functional requirements including recommendation.

In data access layer, actual database comes in the picture. Data Access Layer contains methods to connect with database and to perform insert, update, delete, get data from database based on our input data. The data to be stored are user profile, item profile and user click history. Detailed of database design will be introduced later.

By choosing this 3-tier architecture approach, we can focus more on the Android application development as there is much less setup needed for the backend. Besides, it increase the scalability since each tier can scale horizontally and modification won’t affect other modules, which is greatly convenient for algorithm testing.

### 3.2 Recommendation Algorithm Design

The recommendation system algorithm is usually an implementation of a certain type of recommendation model that is responsible for obtaining data,
such as user preferences and descriptions of recommendable items, and predicting which options a given user group will be interested in. Among them, content-based recommendation algorithms, item-based as well as user-based collaborative filtering algorithms, latent factor model, personalized pagerank and item2vec are algorithms that are investigated the most and are the most mature.

However, the algorithms above are only general recommendation algorithms. They may solve the problem that how to recommend personal-prefer items. For this project, a personalized recommendation is surely important but not all. Not only should it recommend interesting items but also medically right items. As a result, the recommendation system should be medically adjusted.

Moreover, another objective has been raised in the objective part. Our recommendation system should be dynamic and interactive. By tracking the user’s steps count and exercises amount, the recommendation system should adjust according to the user’s personal condition. For example, if the user has already reached the required amount of exercise, only light exercises should be recommended.

The remainder of this subsection will first introduce the widely used algorithms mentioned above. Then, an experiment was conducted to compare the performance of algorithms. Based on their performance, several algorithms will be selected to compose a hybrid model. A regression experiment in the same set up will verify its performance. Furthermore, the medical and dynamic adjustment will be introduced. Finally, we will brief the architecture of recommendation system and discuss how different models integrate with each other.
3.2.1 User-based CF & Item-based CF

The recommendation list for target user was created based on user-based CF algorithm according to the view of other users. The logics are if some users rate some items at similar ratings, they will also rate other items very similarly. CF recommendation system is a method of using statistical techniques to look for the nearest neighbors of the object user. Basing on the item rating rated by the nearest neighbors, it will predict how the object user rate the items in order to offer a corresponding recommendation list. On the other hand, Collaborative Filtering component which utilizes a neighborhood-based algorithm firstly chooses a subset of users according to their similarity to the active user, and the use a weighted combination of their ratings to make predictions for the users that keep active.

The following shows the steps of the user-based CF:
1. Users are grouped according to their rated or liked items.
2. The object users are weighted in regards to resemblance with the active user.
3. Pick up ‘n’ active users who have the greatest similarity.
4. Calculate a forecast based on a weighted combination.

To be noticed that, the similarity between two users is computed based on cosine similarity. Besides the original cosine similarity, two upgraded versions have been developed. The first one integrates penalty for the over-popular item. These items can interfere the correlation between uncorrelated users. The second one considers time penalty, which focuses more on recent user behaviors.
Item-based CF follows the same idea as user-based CF. The main difference is that instead of representing the user in terms of items, an item is represented in terms of users. Then, similarities among items can be derived. Also, we have upgraded the cosine similarity by considering penalty for over-active user and time penalty.

3.2.2 Content-based Filtering

The function of content-based filtering is to recommend items that are identical to the items the user preferred to in the previous, which is different compared with collaborative filtering. This approach is based on deriving the similarity between items based on their content, such as title and description, but not how people use them. In this method, the abundant information depicting different items is supposed to be available in the form of a feature vector (y). Those various feature vectors are responsible for constructing a model of user preferences. According to the recommendations that can be generated, a great variety of machine learning techniques like support vector machines and information retrieval are capable of creating a complete user model.

3.2.3 Latent Factor Model

LFMs are parameterized modeling techniques which compute and produce a lower dimension model on the basis of the rating data, and then use it for
follow-up processing. It is constructed on the assumption that if the latent factors are the only features that govern a user’s rating for an item. For instance, referring to movie recommendations, the latent factors include language, genre, alike and director. Even if the abstraction focused by the latent factors are always domain specific, LFM have no requirement of explicit knowledge of the linked users and items, as well as the area of recommendation. They intend to retrieve the characteristics of the items and the preference prototype of the users in light of the latent factors derived from the rating data itself.

LFM models both users and items as vectors characterizing their association with each of the latent factors. Rethink the case of movie recommendations, generally, a movie can be defined in two ways. One is the features of the movie itself, such as cast, languages and its genre (i.e. thriller or drama), another way is the user affinities to these features, that is, whether the users like the movie or not. Therefore, a single item can be illustrated as a vector enumerating the degree of holding of various features, whereas a user can be modeled as a vector that denotes the affinity about the related features. In this case, the user rating for an item is regarded as a predictor of the extent of alignment of the representations of their corresponding latent factors. It is modeled as an inner product between their respective latent factor vectors.

3.2.4 Personalized PageRank

Personalized PageRank has two different definitions and implementations. One is based on a random walk on a bipartite graph. The other one is about matrix transformation. Although the graph-based implementation is the original version and easier for understanding, its requirements to traverse the graph until convergence is too inefficient in practice. Therefore, the matrix implementation is used here.
Personalized PageRank recursively models the importance of nodes on a directed graph. At a high level, given a start node $s$ whose point of view we take, we say that $s$ is important, and in addition, we say a node is important if its in-neighbors are important. To keep the importance scores bounded, we normalize the importance given from a node $u$ to a node $v$ though an edge $(u, v)$ by dividing by $u$’s out-degree. In addition, we choose a decay term $\alpha$, and transfer a fraction $1 - \alpha$ of each node $u$’s importance to $u$’s out-neighbors. Formally, given normalized edge weight matrix $W$, the Personalized PageRank vector $\pi_s$ with respect to source node $s$ is the solution to the recursive equation. If $s$ is a distribution (viewed as a column vector where entry $s[i]$ is the weight $s$ gives to node $i$) the recursive equation becomes
\[
\pi_s = \alpha s + (1 - \alpha)\pi_s W
\]

### 3.2.5 Item2vec

Word2Vec is a tool that creates an Artificial Neural Network (ANN) able to learn on input corpora. This tool creates vectors from words, learns what words are frequently followed by others and calculates similarities between them. Similarities may be used for many different purposes, i.e. in Natural Language Processing (NLP) they may be used for improving machine translation. Word2Vec can use one of two techniques: continuous bag-of-words or skip-gram.
CBOB and skip-gram are neural architectures that describe how to learn the model. They have different working principles, which are seen in Figure 1. CBOB tries to predict one word from context and Skip-gram tries to predict context by word. Skip-gram can be trained with negative sampling, a mechanism that presents negative samples created in a random way (same as the training samples, but all are negative and not included in training) to the model.

The workflow of Word2Vec model looks like this:
Step: 1. Corpus reading, counting how often words appear in the text,
Step: 2. Words sorted by word frequency in the corpus,
Step: 3. From vocabulary, producing negative samples,
Step: 4. Sentence by sentence corpus reading into the model and sub-sampling (the process of eliminating most frequent words from analysis),
Step: 5. Algorithm goes through the sentence with a specified window (model parameter), which is a number of words algorithm takes into account around the current word,
Step: 6. Using Feed-forward Neural Network where connections do not form a cycle between the units

Item2Vec creates vectors of items and learns on input in the same way as Word2Vec. In this thesis the Item2Vec model produces recommendations by learning on the history of what users liked, taking into account context (another user’s rated movies). The whole workflow can be shown in the following figure.

![Item2vec workflow](image)

**Figure 9. Item2vec workflow**

### 3.3 Recommendation Algorithm Experiment

#### 3.3.1 Experiment Set Up & Procedure

In this paper, an experiment has been conducted on two MovieLens datasets, MovieLens100K and MovieLens1M. MovieLens100K describes 5-star rating and free-text tagging activity from MovieLens, a movie recommendation service. It contains 100836 ratings and 3683 tag applications across 9742 movies. These data were created by 610 users between March 29, 1996, and September 24, 2018. While MovieLens1M contain 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users. Both datasets include ratings from each user for some movies as well as genres for each movie.
All algorithms mentioned in the methodology part have been implemented in Python. If the rating is < 4.0, the movie will be regarded as disliked by the corresponding user. The output of algorithm is top N (N = 5, 10, 15 ..., 45, 50) movies that are most likely to have ratings ≥ 4.0. Datasets are randomly separated into a training set and the testing set with roughly 2:1 quantity ratio for each user. Each algorithm runs on training set first and then compute accuracy on the testing set. The accuracy is defined as the total number of movies predicted right divided by the total number of movies that are both recommended by algorithms and rated by users.

After comparing all the individual algorithms, selecting the algorithm applied in the hybrid model based on the experimental results. The specific combination method uses a weighted average, that is, using the normalized offline accuracy of the algorithm on the test set to determine the number of items recommended by the algorithm. Different algorithms recommend the same item to be deduplicated. The combined hybrid model is subjected to a regression experiment to check the actual performance.

### 3.3.2 Experiment Results & Discussion

First, when comparing item CF and its two upgraded versions, we found that the item CF with time penalty performed worse, while the other two versions performed better. In addition, the item CF with penalty for over-active user performs better than a small dataset on a larger dataset. The original version does not reflect this growth trend. It is speculated that the over-active user is sparse on a small data set, and the effect of penalty cannot be better reflected. For user CF, the three versions perform very poorly. Accuracy is below 0.7 in both large data sets and small data sets.

The Content-based model shows that on the small dataset, with the increase in the number of recommended TopNs, accuracy is extremely
unstable, ranging between the lowest 57.4% and the highest 80%. On the big dataset, it shows a consistently high accuracy with higher than 80%. The reason for this difference is the way to build the user profile in the implementation. Since the original data set does not give specific information for each user, I can only use the simple counting method to count the tendency of each user for a different genre. Therefore, on small data sets, this method of building user profiles is too simple and accuracy is comparatively low. In the actual application, more user information can be collected to overcome such cold start problem.

LFM and personalized pagerank show high accuracy and stability. The Item2vec algorithm performs quite poor on small datasets, and the overall accuracy is below 0.7. However, on big data sets, although accuracy has not reached a very high level, it has made great progress. The average accuracy is over 0.75 and is quite stable. This is because item2vec is different from the previous shallow model, and item2vec uses a neural network, so the number of trainable parameters is greatly increased. On small data sets, only almost 600 corpora can be used for training after processing. Corpus not only limits the features that the model learns, but it also leads to the lower accuracy. With the accumulation of users, the performance of the model has also improved.
Figure 10. Accuracy for algorithm movies100K

Figure 11. Accuracy for algorithm on movies1M
3.3.3 Algorithm Selection

According to the experimental situation, LFM algorithms with greater performance is selected, and the user CF, as well as item CF, are eliminated. Despite of good performance of personalized pagerank, its graph based nature causes slow runtime. Although by using matrix simplification technique, time complexity has decreased, it is still slower than others. Since the recommendation system still has medical and dynamic module, in order to achieve a real-time response, personalized PR have to be given up.

In addition, although item2vec shows a certain degree of growth, the number of users required for its high accuracy is very high and is not suitable for recommending the initial scene of the system, so item2vec is also eliminated. Content-based filtering has problems with small data sets, but this kind of cold start problem is easier to overcome with other methods of constructing user profiles, so content-based filtering is also listed as a candidate. Therefore, hybrid model is the comprehensive model of the above selected algorithms.

In order to verify the performance of hybrid approach, a regression experiment was conducted. According to the regression experimental results, we can clearly see that the hybrid approach has better performance and stability on both large data sets and small data sets. On movieLens100K, the average accuracy is 0.77, and on movieLens1M, the average accuracy is 0.84. Therefore, the hybrid model by combining itemcf and content-based filtering is effective.
Figure 12. Accuracy for hybrid approach on movies100K

Figure 13. Accuracy for hybrid approach on movies1M
3.4 Medical Adjustment of Recommendation Algorithm

In this recommendation system, medical adjustment means that the application is required to recommend an exercise program medically suitable for patient’s condition. In other words, it is to make a prediction problem. In reality, each exercise has a different degree of matching to different patients and is therefore closer to a regression problem. However, after consulting with relevant doctors, it is found that there is currently no quantitative research on the conditional relief of sports in the medical community. Therefore, to increase the feasibility, we simply divide each movement into appropriate and inappropriate. Immediately, the whole problem is reduced to a two-class problem.

There are two general ideas for solving such two-class problems. One is big data approach, which forms a training set through data related to classification, and then trains the classifier through some supervised or unsupervised method. Finally, test the verification on the test set. This type of method is reproducible, extensible, accurate and recalled. It belongs to a traditional machine learning method. However, in practical projects, this method has encountered a very large difficulty, that is, insufficient data. Since Parkinson's is not a common disease, the number of patients is not much, and the majority of the patients are elderly people, the difficulty and cost of data collection are very large. In addition, the training classifier needs to collect a wide range of data, often involving personal privacy such as physical health, so data collection is not smooth. In the end, only 68 questionnaires were collected through an online questionnaire. Such data volume intelligence is used for market research and cannot be used for machine learning. In addition, in the public data set, the two highest quality data sets are ‘Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection’ and ‘A comparative analysis of speech signal processing algorithms for Parkinson's disease classification and the use of the tunable Q-factor wavelet transform’. However, such public data sets are mostly purely medical uses, and there is no information related to
motion and rehabilitation in the data set, and thus they are also difficult to be utilized.

The second idea is the knowledge-based approach. This method is to use the model to transform the expert's empirical knowledge into a model or strategy to judge the classification. This method is simple and easy, but the scalability is poor. In addition, the applicability of new areas with less human knowledge reserves is not good. Through information about medical experts, we found that during the hospital's diagnosis and treatment, doctors recommend sports to patients normally based on experience, and rely heavily on medical questionnaire. Such questionnaire is used to follow the longitudinal course of Parkinson's disease. Clinicians and researchers alike use the questionnaire to follow the progression of a person's Parkinson's disease. Scientific researchers use it to measure benefits from a given therapy in a more unified and accepted rating system. Neurologists also use it in clinical practice to follow the progression of their patients' symptoms in a more objective manner. Such scales can be quantified, classified, and the patient's condition recorded. Through the scale, you can clearly understand the severity of the patient's condition and the main symptoms of Parkinson.

The UPDRS is the most commonly used scale in the clinical study of Parkinson's disease. The Unified Parkinson’s Disease Rating Scale (UPDRS) was originally developed in the 1980s and included two major rating scale, Hoehn and Yahr scale and Schwab and England activities of daily living scale. The UPDRS is made up of these sections:

Part I: evaluation of mentation, behavior, and mood
Part II: self-evaluation of the activities of daily life (ADLs) including speech, swallowing, handwriting, dressing, hygiene, falling, salivating, turning in bed, walking, and cutting food
Part III: clinician-scored monitored motor evaluation
Part IV: complications of therapy
Part V: Hoehn and Yahr staging of the severity of Parkinson's disease
Part VI: Schwab and England ADL scale

By filling the rating scale, scores for different symptoms and body condition will be derived. Following the UPDRS scores over time provides insights into the patient's disease progression. For instance, if the patient’s symptoms started with a slight tremor, so his motor score would have been less than 10. The version we used is a revision of the UPDRS, known as the MDS-UPDRS published by the Movement Disorder Society (MDS) in 2007. MDS-UPDRS overcome two major limitations include the lack of consistent anchor among subscales and the low emphasis on the non-motor features of PD. The modified UPDRS retains the four-scale structure with a reorganization of the various subscales. The scales are now titled:
(1) non-motor experiences of daily living (13 items)
(2) motor experiences of daily living (13 items)
(3) motor examination (18 items),
(4) motor complications (six items)
Each subscale now has 0-4 ratings, where 0 = normal, 1 = slight, 2 = mild, 3 = moderate, and 4 = severe. Due to the purpose of our project, we mainly use section 2 and 3, which are motor experiences of daily living and motor examination. Example of questions and score sheet are shown in Figure 10.
<table>
<thead>
<tr>
<th>Part II</th>
<th>3.13</th>
<th>Posture</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Speech</td>
<td>3.14</td>
</tr>
<tr>
<td>2.2</td>
<td>Saliva and drooling</td>
<td>3.15a</td>
</tr>
<tr>
<td>2.3</td>
<td>Chewing and swallowing</td>
<td>3.15b</td>
</tr>
<tr>
<td>2.4</td>
<td>Eating tasks</td>
<td>3.16a</td>
</tr>
<tr>
<td>2.5</td>
<td>Dressing</td>
<td>3.16b</td>
</tr>
<tr>
<td>2.6</td>
<td>Hygiene</td>
<td>3.17a</td>
</tr>
<tr>
<td>2.7</td>
<td>Handwriting</td>
<td>3.17b</td>
</tr>
<tr>
<td>2.8</td>
<td>Doing hobbies and other activities</td>
<td>3.17c</td>
</tr>
<tr>
<td>2.9</td>
<td>Turning in bed</td>
<td>3.17d</td>
</tr>
<tr>
<td>2.10</td>
<td>Tremor</td>
<td>3.17e</td>
</tr>
<tr>
<td>2.11</td>
<td>Getting out of bed</td>
<td>3.18</td>
</tr>
<tr>
<td>2.12</td>
<td>Walking and balance</td>
<td>3.19</td>
</tr>
<tr>
<td>2.13</td>
<td>Freezing</td>
<td>Did these movements interfere with readings? No Yes</td>
</tr>
<tr>
<td>3a</td>
<td>Is the patient on medication? No Yes Hoehn and Yahr Stage</td>
<td></td>
</tr>
<tr>
<td>3b</td>
<td>Patient's clinical state On Off Part IV</td>
<td></td>
</tr>
<tr>
<td>3c</td>
<td>Is the patient on Levodopa? No Yes 4.1 Time spent with dyskinesias</td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>Speech</td>
<td>4.2</td>
</tr>
<tr>
<td>3.2</td>
<td>Facial expression</td>
<td>4.3</td>
</tr>
<tr>
<td>3.3a</td>
<td>Rigidly--Neck</td>
<td>4.4</td>
</tr>
<tr>
<td>3.3b</td>
<td>Rigidity--Foot</td>
<td>4.5</td>
</tr>
<tr>
<td>3.3c</td>
<td>Dystonia</td>
<td>4.6</td>
</tr>
</tbody>
</table>

### 2.12 WALKING AND BALANCE

Over the past week, have you usually had problems with balance and walking?

0: Normal: Not at all (no problems).

1: Slight: I am slightly slow or may drag a leg. I never use a walking aid.

2: Mild: I occasionally use a walking aid, but I do not need any help from another person.

3: Moderate: I usually use a walking aid (cane, walker) to walk safely without falling. However, I do not usually need the support of another person.

4: Severe: I usually use the support of another persons to walk safely without falling.

Figure 14. Example of questions and score sheet of MDS-UPDRS
When the user first uses the app, he or she will be required to complete this scale rating. Rating for each user will be stored and send to the server, each time apps request a new recommendation. With detailed rating about patient’s condition such as walking and balance, swallowing and chewing and so on, it is very easy to judge which kind of physical ability need to be strengthened.

In order to cooperate with MDS-UPDRS, 13 tags of exercise have been designed according to the rating scale. Each tag means the exercise is able to enhance patient’s ability relating to the content of the tag. The design process is under consultation and supervision of medical experts. After that, prepared exercise videos are provided to those experts and asking them to tag exercise based on the established tagging system.

3.5 Dynamic Adjustment of Recommendation Algorithm
For PD Secretary, the most ideal situation for dynamic recommendation is to track the user's health and movement, and then adjust the recommendation at any time. However, in reality, on the one hand, the mobile phone lacks a sensor that senses relevant information, and on the other hand, using Bluetooth or other techniques to connect an external device brings too much work. Under the trade-off, the dynamic adjustment can be completed by replacing the amount of exercise with the number of steps per day of the user. After the user completes the basic information and UPDRS, the user can be divided into mild, moderate and severe patients according to the rating scale. Mild patients 0-17 points, moderate patients 18-35 points, severe patients 36-52 points. According to the recommendation of relevant experts, mild patients should maintain the frequency of normal people's exercise, that is, exercise for 1 hour, 6000 steps. The recommended number of steps for moderate patients is 4000 steps, and for patients with severe cases they should walk 2000 steps. In addition, each exercise can be divided into mild exercise, moderate exercise and heavy exercise depending on the amount of exercise.
When making a recommendation, the app passes the number of steps the user has taken to the server. The completion rate $w$ is obtained by using the number of steps taken divided by recommended steps. When $w < 0.33$, heavy exercises are extracted from the recommendation list, placing at the top of the recommendation list. The order between the heavy exercises are maintained the same. When $0.33 < w < 0.66$, moderate exercises are extracted and placed in the recommendation. While $0.66 < w < 1$, extract the mild exercises motion from the recommendation list. If $1 < w$, heavy exercises are extracted from the recommendation list and placed at the end of the recommendation list.

### 3.6 Architecture of Recommendation System

The recommendation system of PDSecretary is mainly composed of three modules, namely medical module, personalization module and dynamic module. At the same time, the recommendation system uses a hierarchical architecture, as shown in Figure 15, which can be divided into three layers.

![Figure 15. Architecture of Recommendation System](image)

The Match layer corresponds to the medical module. Through the UPDRS filled in by the user, the exercise type tag suitable for the user can be known, and the exercise of the corresponding tag is all recalled, passing to the next layer as a candidate set.
The Rank layer corresponds to the personalization module, which is the hybrid model that combines content-based filtering and LFM. The combination of the two algorithms is shown in the following formula:

\[ r = w_1c + w_2lfm \]

The value of output of \( c \) and \( lfm \) is ranged between 0 and 1. The closer the value to 1, the more likely a user clicks. \( w_1 + w_2 = 1 \), both are the weights for adjusting the output. Since the LFM needs to accumulate user clicks, it has a cold start problem, so the weight of the LFM is lowered in the early stage. According to the test, \( w_1 \) is between 0.8-0.9, \( w_2 \) is between 0.1-0.2, the effect is better. Finally, sort all items according to \( r \) and output a list of top 20 to the strategy layer.

The Strategy layer corresponds to the dynamic module. After receiving the sorted list, the module calculates the recommended number of steps based on the user's information, and then adjusts the recommended list result by using the number of steps and recommended steps. Finally output the result.

### 3.7 UI Design

User interface design is important for the application. Better user interfaces can result in better user experience and user satisfaction. This application mainly serves elderly people. Thus, the biggest design principle we should follow is recognition rather than recall from Nielsen 10 heuristics.

According to the mobile app design trend in 2018, the material design is widely used because of its simplicity and conciseness. To deliver messages to the users effectively and display information clearly, Card UI designs are used. This design aims to split chunks of information into small chunks so that user can digest the information easily.
Simple icon instead of a complex graphic are used so that users can easily remember the main functions of the application. Furthermore, when we conducted a usability test with several elderly people using the prototype, we found out that they are somehow unfamiliar with modern app design convention. For example, when they are dealing with feed, nearly half of them did not realize pull the feed at top can refresh and load more items. Therefore, we change our design from the feed to the simple button-window model.

Research done by color psychology found that clam color like green or blue provides a more calm and restful environments to the users. Using these two colors can allow users to focus a certain amount of time. Therefore, the color scheme for PDSecretary is sky-blue, which aims to provide a peaceful environment for users.

3.8 Database Design
In this project, data are stored in the MySQL database. The reason why we choose the relational database is its adaptability with our recommendation algorithm. Non-relational database store and retrieve data that is modelled in means other than the tabular relations. This makes NoSQL more free and convenient but less regularized and standard. The recommendation algorithm module includes LFM which is a matrix factorization method based on the user-item matrix. The tabular structure of MySQL is easy to formulate the user-item matrix and maintain user click history. Thus, MySQL is adopted.

There are 3 tables constructed for PDSecretary. User table, item table and click table. Item table stores item profile, including id, title, description and the total number of click. Also, the tags of each item are stored. For example, if speech = 1, it means this exercise has a speech tag and may enhance patients’ ability to speak. All tags are defined and labelled by experts.
Figure 16. Item Table of MySQL Database of PDSecretary

User table stores the user’s personal information and user profile. Personal information is collected during registration. User profile is constructed gradually. Number of each tag means user’s preference of each tag. Sum of all preference is 1. For example, if speech=0.05 and swallow=0.03, this user should prefer speech related exercise according to our recommendation system.
Click table is used to store user click history. This table is a relation table, recording which user click which item at what time. Such history will be used for user-item matrix in LFM and maintaining user profile in user table.

Figure 17. User Table of MySQL Database of PDSecretary

Figure 18. Click Table of MySQL Database of PDSecretary
4 Final Result
This section will focus on the following major result – the Android application, the algorithm performance analysis.

4.1 Result of Android App
After the research on related-works, we have defined several use cases that will be included in our first release. The Android app will be introduced in terms of pages.

4.1.1 Login Page & Registration Page

![Login Page Screenshot](image)

Figure 19. Screenshot of Login Page

The login page requires user enter user name and password. Wrong user name and password will be denied.
The registration page requires user to enter personal information and complete part of UPDRS for future recommendation. Also, user can select a photo from album or take one with camera as the avatar.
4.1.2 Main Page

![Screenshot of Main Page]

The Main page is the first page after user login successfully. At the top of THE Main page is an auto-play banner. Up to now, since PDSecretary is not commercialized or published, the banner is not clickable. In the future, advertisementS or other contents can be placed in the banner.

Next to the banner is a text view showing how many exercises recommended to the user. By clicking the text view, user can jump to video list page.

Below the recommended exercise are titles of the latest news related to PD. These news are not dynamically recommended. They are manually selected and updated on server. User can jump to the webview to skim the news.

Figure 21. Screenshot of Main Page
4.1.3 Recommended Video Page

Figure 22. Screenshot of Video List Page

Video list page holds all recommended exercises. By clicking the play button, the exercise video will start to play. User can press pause or scroll until the video is out of view to stop the video. Besides, each click here will be recorded and send to server for later recommendation.
4.1.4 Reminder Page
Users can set reminders on the Reminder page, including reminder time, ringing mode, repeat mode and ringtone. The reminders that have been set can be deleted, closed and modified. When the reminder time arrives, the reminder page will appear and the phone will vibrate or ring. Users could press for a while to close the reminder. There is also a delay function: by clicking the alarm clock in the lower left corner, the reminder could be delayed for five minutes.
4.1.5 Account Page

Figure 24. Screenshot of Account Pages
Users could change the password and view and modify the information that they filled in at the time of registration on the Account page. They can also view the number of steps and set the exercise plan on this page. The number of steps can only be updated in a mobile phone with a step sensor. The training plan setting include the setting of how many steps to take each day. If the user does not reach the specified number of steps, they will be reminded regularly. In addition, the users can also view the previous number of steps per day.

4.2 Result of Recommendation System

Because the PDSecretary recommendation system contains multiple modules, the results of each recommendation are less interpretable. In addition, the pros and cons of the recommendation results often require subjective judgment, so it is difficult to objectively test the performance of the recommendation system. To evaluate the recommendations, we invited three Parkinson patients to perform a recommendation system satisfaction test. The three patients were between 70-80 years old, including two moderate Parkinson, and one mild Parkinson. The number of steps can be counted by their mobile phones. The test steps and results are as follows.

4.2.1 Procedure

First, the tester is asked to fill out a complete UPDRS to determine the condition and the type of exercise. The tester is asked to install the app the day before the test so that the number of steps can be accurately counted on the day of the test. After downloading the app, the user is required to sign up for the login, but do not enter the video list page. During the test, the user first opens the app, and then enters the recommended video page and clicks on the video of interest, and finally exits. They repeat this procedure for 3 times.
Then the user enters the video list to determine whether the recommended sports item is liked or disliked. They clicks on any favorite video, exits and enters again, also repeats this procedure for three times. Counting the proportion of videos that the user likes in all recommended videos, and then counting the proportion of videos the user likes in top 5 recommended videos.

The baseline recommendation algorithm that against PDSecretary performs the following operations. First determining the appropriate sport's tag based on the UPDRS filled out by the tester. Next, in all sports, matching the corresponding tag and sorting by the number of matched tags. Then, taking top10 video for recommendation. Lastly, calculating the proportion of favorite videos in the recommended videos.

4.2.2 Performance Analysis

![Percentage of liked exercise among recommendation](image)

Figure 25. Percentage of liked exercise among recommendation
It can be found that two of the three testers believe that the recommendation from PDSecretary is better, and the average PDSecretary satisfaction is about 9% higher than the baseline. This situation can be roughly explained. After the PDSecretary incorporates the personalized recommendation algorithm, it can improve the recommendation matching degree to a certain extent compared with the simple dependence on medical knowledge recommendation.

![Percentage of liked exercise among top5 recommendation](image)

**Figure 26. Percentage of liked exercise among top5 recommendation**

However, in the recommendation of top 5, the proportion of the favorite video is reversed. Two of the three participants thought that PDSecretary was not as good as baseline, and the average was 13% worse. According to the number of steps of the three testers and the background log analysis, the two participants did not complete 1/3 of the recommended steps on the day, therefore, according to the adjustment of the strategy layer, the heavy exercises will be placed at the forefront of the recommendation list. Relatively speaking, users are less interested in heavy exercises. In summary, the strategy layer of the recommendation system needs further improvement.
5 Future Planning & Conclusion

The aim of this project is to implement an easy-to-use mobile app for personalized recommending exercise for Parkinson disease patients and accommodate their lives. Many functionalities have been implemented, e.g. reminder, counting steps to learn more activity data for the better recommendation.

In terms of design, according to the feedback given by the user during the performance test, the user wants to add more useful small functions, such as: retrieve the password, set the font size, and so on. In addition, the user also express that some of the buttons in the reminder page is not clear enough and ask for text guidelines.

Algorithmically, according to the results of the performance test, the next step is to improve the dynamic module. In terms of data acquisition, the application can try to connect the smart equipment via Bluetooth to get more detailed data including sleep information, heart rate and so on. In addition, by obtaining detailed data, it is possible to consider training the machine model for recommendation.

The most important point lies on the content. PDSecretary can now recommend 29 items collected manually. The limited item largely restricts the performance of the recommendation system and the quality of the videos themselves are not high. If there is a large amount of high-quality content afterwards, and the accumulated user volume can make the LFM algorithm overcome the cold-start problem, thus the performance of the entire recommendation system will be improved.

Six months after submitting the project plan, we have determined the system architecture, the basic features of the application, and basic user interface of the application. And finally implemented the whole system. Although there are some setbacks, such as the algorithm is yet far from perfect, our team will continue to research more on this topic.
6 References


